Chekhov's Guns and Damocles' Swords in Real-World Information Security: Post-Quantum Cryptography, Internal Attacker and the Random Oracle

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik des
Karlsruher Instituts für Technologie (KIT)
genehmigte
Dissertation

von

Wasilij Beskorovajnov

aus Novosibirsk (UdSSR)

Tag der mündlichen Prüfung: 14. Januar 2026
1. Referent: Prof. Dr. rer. nat. Jörn Müller-Quade
2. Referent: Prof. Dr.-Ing. Tibor Jager

Acknowledgements

I would like to thank Prof. Dr. Jörn Müller-Quade for sparking my interest in theoretical computer science during my early bachelor's studies, and for continuing to nurture it throughout the whole journey, from completing my bachelor's degree to finishing my master's. During my studies, Prof. Dr. Jörn Müller-Quade repeatedly demonstrated an impressive—and, to a master's student, seemingly inexhaustible—depth of knowledge. I would therefore like to thank Prof. Dr. Jörn Müller-Quade once again for taking on the responsibility of supervising a knowledge-hungry student, and for guiding, challenging, inspiring and refuting my ideas countless times throughout my research journey. I would like to thank all of my co-authors with whom I had the opportunity to spend long evenings, since it often feels like an unwritten rule, from studying for a bachelor's degree to submitting a conference paper, that you inevitably end up waiting until the very last minute to submit. A big thanks to Dr. Rebecca Schwerdt, Astrid Ottenhues, Sarai Eilebrecht, Roland Gröll, Prof. Dr. Thorsten Strufe, Dr. Alexander Koch, Dr. Gunnar Hartung, Dr. Felix Dörre, Robert Brede, Maximillian Müller and Yufan Jiang. I would like to extend a special thank you to Laurin Benz for joining me on the memorable adventure to my first conference, PKC 2023 in Atlanta, and for experiencing American culture firsthand together.

My deepest thanks go to my family, and most notably to my wife, Marie Weihnacht, who has been, and still is, the undisputed rock amid the turbulence of my endeavors. I also cannot leave unmentioned how my daughter, Liliana Weihnacht, has filled my life with countless moments of happiness that helped me forget the hard parts of the many setbacks during my research, and has taught me to push my limits in multitasking and negotiation. Finally, I would like to give a special thank you to my parents, Alla Aleksandrovna and Igor Wasiljewitsch Beskorovajnov and my sister Alina Beskorovajnov, who taught me how to overcome seemingly impossible challenges and the meaning of family. Without their decision to emigrate to Germany, my life would have been completely different.

Last but not least, I would like to thank my colleagues at the FZI Forschungszentrum Informatik for helping me grow, both as a professional and as a researcher, and for teaching me how impulses from industry influence research.

Abstract

This thesis addresses critical threats in real-world cryptographic applications that may not be immediately apparent but can materialize at any time. The author identifies and tackles three such threats that challenge the security of modern cyber-physical system networks.

First, the Quantum Computer threat is examined, highlighting the necessity for cryptographic assumptions that are quantum-safe, meaning they remain secure against both classical and quantum computers. The author's work addresses these challenges through a series of advancements that incorporate quantum-safe assumptions where possible.

Second, the thesis addresses the internal attacker threat, which has often been relegated to secondary importance in the industrial security. However, the author argues that neglecting protections against internal attackers undermines overall security, as internal components can and do fail. This is exemplified through the design of secure protocols for applications like contact tracing, where the risk of nationwide surveillance by internal attackers is mitigated through an honest-but-curious model. Additionally, the thesis presents an outsourced computation protocol that allows secure processing of sensitive data, even when the server may collude with clients, formally revisiting a paradigm for outsourced computation that protects personal data from internal threats.

Lastly, the Random Oracle threat is explored, which questions the reliability of using the random oracle model in cryptographic constructions. While the random oracle methodology has been instrumental in real-world cryptography, it is known to lose its security guarantees when replaced by hash functions in theory. In response, the thesis presents a secure channel protocol that avoids the need for random oracles, providing a more formally sound approach to secure communication. Though this protocol is less efficient than its random oracle-based counterparts, it achieves substantial improvements over previous non-random oracle constructions.

Overall, this thesis contributes to the development of cryptographic solutions that address unaddressed threats in real-world applications, aiming to enhance both the security and practical applicability of cryptographic systems in the face of evolving challenges.

Zusammenfassung

Diese Arbeit behandelt kritische Bedrohungen in realen kryptografischen Anwendungen, die möglicherweise nicht sofort offensichtlich sind, aber jederzeit auftreten können. Der Autor identifiziert und geht auf drei solche Bedrohungen ein, die die Sicherheit moderner cyber-physischer Systemnetzwerke in Frage stellen.

Zunächst wird die Bedrohung durch Quantencomputer untersucht und die Notwendigkeit für kryptografische Annahmen hervorgehoben, die quantensicher sind, d.h. sie bleiben sowohl gegenüber klassischen als auch gegenüber Quantencomputern sicher. Die Arbeit des Autors geht auf diese Herausforderungen durch eine Reihe von Fortschritten ein, die wo möglich quantensichere Annahmen einbeziehen.

Zweitens wird in der Arbeit die Bedrohung durch interne Angreifer behandelt, die oft in der industriellen Sicherheit zweitrangig behandelt wurde. Der Autor argumentiert jedoch, dass das Vernachlässigen von Schutzmaßnahmen gegen interne Angreifer die Gesamtsicherheit untergräbt, da interne Komponenten kompromittiert werden können. Dies wird durch die Entwicklung sicherer Protokolle für Anwendungen wie Kontaktverfolgung verdeutlicht, bei denen das Risiko einer landesweiten Überwachung durch interne Angreifer durch ein ”honest-but-curious” Modell gemindert wird.

Darüber hinaus präsentiert die Arbeit ein Protokoll für ausgelagerte Berechnungen, das es ermöglicht, sensible Daten sicher zu verarbeiten, auch wenn der Server mit Clients kolludieren kann. Dies geschieht durch eine formale Neubetrachtung eines Paradigmas für ausgelagerte Berechnungen, das den Schutz personenbezogener Daten vor internen Bedrohungen gewährleistet.

Schließlich wird die Bedrohung durch den Zufalls-Orakel-Ansatz (Random Oracle) untersucht, der die Zuverlässigkeit des Einsatzes des Random Oracles in kryptografischen Konstruktionen diskutiert. Obwohl die Random Oracle Heuristik in der realen Kryptografie von unschätzbarem Wert ist, ist auch

bekannt, dass sie ihre Sicherheitsgarantien verliert, wenn sie durch Hash-Funktionen ersetzt wird. Als Reaktion darauf präsentiert die Arbeit ein sicheres Kommunikationsprotokoll, das den Bedarf an Zufalls-Orakeln vermeidet und einen formal fundierteren Ansatz für sichere Nachrichtenübertragungen bietet. Obwohl dieses Protokoll weniger effizient ist als seine Pendants, die auf dem Zufalls-Orakel basieren, erreicht es wesentliche Verbesserungen gegenüber früheren nicht-Zufalls-Orakel-Konstruktionen.

Insgesamt trägt diese Arbeit zur Entwicklung kryptografischer Lösungen bei, die unbehandelte Bedrohungen in realen Anwendungen angehen und darauf abzielen, sowohl die Sicherheit als auch die praktische Anwendbarkeit kryptografischer Systeme angesichts der sich entwickelnden Herausforderungen zu verbessern.

Contents

List of Figures

List of Tables

Part I.

Introduction & Motivation

1. Introduction

Throughout the author's scientific work, there have been numerous discussions about what defines *real-world cryptography*. These discussions arose because the author often claimed to work on solutions aimed at addressing real-world cryptographic challenges. After many debates, we concluded that the term is highly subjective and largely influenced by personal opinions.

In what follows, the author will humbly summarize the result of this discussion, which should serve, at the same time, as an introduction for the reader to the topic of real-world information security, where cryptography plays a pivotal role.

The author's impression during the discussions was that no single criterion definitively distinguishes a cryptographic construction as *not real-world* cryptography, i.e., as being of *purely academic interest*. And even then: Can *"non-real-world"* be equated with *"purely of academic interest"*? Furthermore, even if a particular construction is initially categorized as *"of academic interest"*, as the author and the author's colleagues did with Bitcoin many years ago, others may hold different perspectives. Indeed, Bitcoin eventually transitioned into widespread, real-world use despite the author's initial assessment.[1] To the author, it seems that the question itself is flawed. But up to this point in time, the author has not been able to re-formulate it in a more fitting way.

The following examples are used by the author to highlight the diverse applications of cryptographic mechanisms, focusing on areas where the author has made scientific contributions. These examples will demonstrate why the author considers them to constitute real-world cryptography and will put the author's work into perspective. Then, in the motivation section, the author will explain why concepts such as internal adversaries, quantum-computer-assisted cryptanalysis, and random oracles pose ambiguous threats to real-world applications.

[1] Though perhaps not in the way it was promised by many.

Ingenious Inventions from the Academia The work of [308] addresses the intriguing "Millionaire's Problem," wherein two parties seek to ascertain the wealthier individual without revealing the precise amount of their wealth. While this problem may seem straightforward, it holds profound implications for secure multiparty computation (MPC), enabling the secure execution of intricate mathematical functions among multiple parties.

The applications of MPC are highly diverse, encompassing secure evaluation of statistics or data mining without divulging personal information, see e.g., [275], confidential auctions with no reliance on third-party trust, see e.g., [54], and training neural networks without exposing underlying input data, see e.g., [191]. Despite its myriad possibilities, MPC has not yet permeated the mainstream market. The challenges predominantly stem from technological prerequisites and the cost-benefit ratio, which have not yet reached a threshold for widespread adoption.

MPC stands out as a compelling example highlighting that the realm of cryptology academia is rich with cryptosystems ingeniously designed with pure mathematical applications in mind. The widespread adoption of these cryptosystems hinges on the integration of these mathematical applications, which, to a large extent, are still non-existent in the real world.

At this juncture, it is crucial to highlight that many of these cryptographic inventions have reached a notable level of maturity, undergoing thorough security assessments through successive studies. In 2020, the Federal Office for Information Security (BSI) issued technical guidelines [70] for the implementation of online voting. These guidelines endorse the use of advanced cryptographic techniques such as homomorphic encryption and mixnets.

Additionally, the 'Privacy Enhancing Cryptography' (PEC) project [227] by the U.S. National Institute for Standardization and Technology (NIST) serves as a comprehensive resource, presenting numerous examples of advanced cryptographic techniques. These include zero-knowledge proofs, secure multiparty computation, (fully) homomorphic encryption, functional encryption, group and ring signatures, as well as Private Set Intersection (PSI) and Private Information Retrieval (PIR). In the chapter about the motivation of the internal attacker, i.e., Chapter 4, the author describes the underlying real-world information security challenge that these methods ultimately seek to solve, despite their currently not being widespread.

In a joint work with Sarai Eilebrecht, Yufan Jiang, and Prof. Dr. Jörn Müller-Quade, the author conducted research [42] to make significant progress toward bringing special MPC, i.e. Secure Function Evaluation (SFE), protocols into the real world. In our case study, we augmented a tutoring-service matching platform with Homomorphic Encryption (HE), yielding a variant of the Outsourced Private Set Intersection (OPSI) protocol. We also examine this protocol from a more general perspective, i.e., from the perspective of private equality-filtering algorithms, and demonstrate its merits relative to other outsourced-computation protocols for equality filtering. Our main contribution is an abstraction of this protocol that yields a two-round, two-party, sender-private secure function evaluation (SFE) protocol based on homomorphic encryption, with the requirement that the sender always encrypts its input.

What Can We Do with Blockchain? The choice of words in the title of this paragraph may be considered unscientific. However, the author of the thesis has encountered this question frequently enough. And as of today, it remains unanswered.

The blockchain is a cryptographic invention that undoubtedly stands out from the previous examples. Unlike secure multiparty computation, the blockchain has already found a firm place in the market. However, despite its size, the blockchain market only marginally affects everyday life. Despite more than a decade of intensive efforts, the successful integration of blockchain concepts into everyday applications has not been achieved. The reasons for these failures are undoubtedly diverse and application-specific in each case. After our in-depth examination of this issue in 2020 through a study (Project 374) [72] for the Federal Office for Information Security (BSI), a common pattern has crystallized in many of the applications that have since been discontinued. Apart from many of these applications being linked to blockchain without sufficient conceptual development, the core misunderstanding in blockchain applications lies in a fundamental misconception. Professional cryptographic applications are designed with a well-defined adversary model in mind, whereas most blockchain applications are designed around the blockchain itself, without considering any adversarial attack surface. This neglect results in applications that are secure in isolation—meaning no adversary can directly manipulate the application—but the actual adversary remains unaddressed outside of the application, continuing their attacks. Ex-

amples of such applications range from supply chain protections to attempts at solving fundamental national economic problems.

Security and Privacy Concerns of Real-World Technologies On March 11, 2020, following more than 118,000 COVID-19 cases in 114 countries and 4,291 deaths, the World Health Organization (WHO) declared[2] COVID-19 a pandemic. One of the initial measures adopted was contact tracing, as the highly infectious disease primarily spread through aerosols in close proximity. Tracing and preemptively warning contacts of an infectious person were anticipated to reduce subsequent infections. Overall, the effectiveness of contact tracing depends on various factors related to the disease and the social and cultural context of the population. In this work we concentrate on the technologies securing these methods and thus the discussion around the effectiveness of these methods is considered beyond the scope of this thesis. Interested readers can find relevant studies in the literature, for example, [222].

The initial technology employed for contact tracing involved paper lists documenting individuals who had visited specific locations. As is typical with paper lists, this method proved inefficient and, consequently, ineffective in tracing a rapidly spreading and highly infectious disease. The search for a digitized and automated solution commenced shortly after the WHO's pandemic declaration, with the industry proposing the first digital contact tracing solutions. In Europe, these initial solutions relied on a centralized server that collected and retained comprehensive information. The cryptographic community's response was nearly instantaneous, revealing a significant array of privacy and security concerns associated with such centralized solutions, see e.g., [293].

Simultaneously, numerous cryptology researchers, including the author of this thesis and colleagues Dr. Alexander Koch, Dr. Gunnar Hartung, Felix Dörre, Prof. Jörn Müller-Quade, and Prof. Thorsten Strufe, began exploring alternative solutions. Roughly one month after the initial centralized proposal, we published our preliminary work on eprint, which later evolved into the comprehensive version documented in [47]. A condensed version was also presented in the proceedings of Asiacrypt 2021 [48].

[2] `https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020`

At the time of writing this thesis, the WHO had already downgraded the global emergency status of COVID-19 [303], and many measures, including digital contact tracing solutions, were lifted.

While our solution was not among those implemented, the ultimately adopted digital contact tracing solution [286] was nationally implemented in Germany with numerous features addressing the security and privacy concerns of the initial centralized variant, though not all possible ones.

In conclusion, digital contact tracing effectively supplanted pen-and-paper technology, showcasing how cryptographic mechanisms can address privacy and security concerns. Thus, it serves as an exemplary case in this thesis for the application of cryptology in the real world.

2. Motivation & Contribution

2.1. Motivation

The title of this thesis introduces two metaphors that reflect the real-world challenges cryptology faces today. Since the interpretation of these metaphors is highly subjective, we do not focus on delineating their differences and instead use them interchangeably. In doing so, we refer to threats to real-world applications that the author has addressed through cryptographic scientific work. These threats form the core motivation for the author's research.

Chekhov's Gun is a narrative principle from Anton Pavlovich Chekhov that requires every narrative detail, such as objects, to become relevant at some point in the story. It is often used metaphorically for potentially dangerous objects, such as a gun, that will materialize as a source of danger in the future.

Similarly, the Sword of Damocles is an ancient moral parable that symbolizes the constant threat of impending danger, particularly for those in positions of power or privilege. It originates from the story of Damocles, who, while seated on a king's throne, realized that a sword hung above his head by a single thread, illustrating the precariousness of his situation. In literature and storytelling, the 'Sword of Damocles' is often used as a metaphor for looming threats or anxieties that create tension, even if the danger does not immediately materialize.

These metaphors are frequently used in real-world contexts across a variety of situations and are more applicable to the field of cryptology now than ever before. The author will denote, from his point of view, which metaphor best describes each specific threat. Ultimately, it is left to the reader to disagree and decide which metaphor best captures each threat.

Cryptography today stands as one of the fundamental cornerstones in our daily lives. Despite its transparent nature to end-users and often remaining concealed even from software development teams, it resides deeply within security mechanisms. The properties of cryptographic schemes, including assumptions, the underlying adversary model and security definitions, play the main role. The assumptions themselves do not prove security. The reductionist proof and the model thereof (which includes hidden assumptions as well) are regarded as such a proof, though assumptions are naturally not provable by design. Their main role in cryptography is to be as simple, truthful, and plausible as possible, providing a foundation to which complex security statements can be reduced. Unfortunately, there is no reliable way to determine the truthfulness of an assumption in the long term. As a consequence, cryptographic assumptions will continue to be mere Chekhov's Guns and Damocles' Swords. Therefore, cryptanalysis, which continually questions the validity of such assumptions, is an essential counterpart to cryptography.

Cryptanalysis has so far shed light on many different assumptions, leading to various outcomes. Some assumptions have prevailed since the 1980s until today, such as the Goppa code indistinguishability assumption used within the McEliece cryptosystem [208]. Other assumptions have been broken within a decade, such as certain variants of problems in the context of supersingular isogenies, in particular those underlying SIDH/SIKE with auxiliary points [96, 242, 253]. A related example is the Rainbow multivariate signature scheme: proposed in 2005 and later selected as a finalist in the NIST post-quantum standardization project, it withstood public scrutiny for more than fifteen years before a practical key-recovery attack in 2022 invalidated the concrete parameter sets submitted to NIST [49, 129]. As in the case of supersingular isogenies, this does not rule out the broader design paradigm altogether, but it demonstrates how specific instantiations can fail once new cryptanalytic techniques emerge. Other outcomes, fortunately, do not invariably result in severe consequences, as many cryptanalytic outcomes are primarily academic or theoretical in nature. One notable example pertains to the awareness discussions surrounding backdoor presumptions in curve selection for elliptic curve cryptography by various standardization bodies, as detailed in [36]. As of today, none of these presumptions have been substantiated as true.

Therefore, cryptology is a never ending cycle of developing cryptographic fundamentals that are scrutinized by cryptanalysis and either end up being discarded as untruthful or adapted to be scrutinized again. The focus of this thesis addresses several special categories of assumptions. The first is

the internal attacker, which can be regarded as an implicit assumption of the adversary behaving in a special way, i.e. compromising parties. The second category describes the challenge of a large pool of cryptographic assumptions being vulnerable to quantum computer assisted cryptanalysis and the migration of cryptographic protocols to quantum-safe assumptions. Finally, we revisit the random oracle methodology and its challenges today.

Internal Attacker The current state of information security in the international industry faces a myriad of problems that require attention. Numerous academic and industrial resources are available that cover most of these issues, and while many of them are not directly related to cryptology, they are equally important because the overall security of an information system is only as strong as its individual security mechanisms. One such problem is the looming threat from internal attackers. In today's world where state-level attackers become more prevalent, the risks of penetrating network defenses or impersonating/intimidating or bribing employees grow higher. This problem currently lacks a solution in the industry because the industrial state of the art unanimously considers only organizational security measures that are highly privacy-invasive for employees, as is recommended in [71, 110]. The author of this thesis has conducted two case studies that consider this attacker. One is a secure contact tracing protocol [47, 48] for warning potentially COVID-19 infected individuals and the other is a protocol for processing encrypted data [42] with the help of Homomorphic Encryption. Last but not least, the author has also formally analyzed End-to-End-Encryption protocols based on Short Authentication Strings in [41]. These protocols remain provably secure in case the systems were affected by an internal adversary attack penetrating the server infrastructure.

Building on prior work in privacy theory and policy, the author understands privacy not as a luxury or optional add-on, but as a necessary precondition for core individual and democratic freedoms. Cohen argues that privacy is indispensable for the development of the "networked self" and for meaningful participation in cultural and political life [111], while Nissenbaum's theory of contextual integrity explains how appropriate information flows are essential for trust within specific social contexts [224]. International human-rights instruments, such as the United Nations General Assembly resolution on the right to privacy in the digital age [290], likewise recognize privacy as a prerequisite for the exercise of other fundamental rights. Empirical studies

by Penney on chilling effects show that surveillance and the loss of privacy can discourage individuals from reading about, and speaking on, sensitive or controversial topics online [237]. In this sense, privacy is necessary to be able to read in private, to converse in private, and to doubt, experiment, and form opinions in private.

At the time of writing this thesis, the privacy demand has not yet gained sufficient prominence in the awareness of the international broad audience. It is often viewed as a "nice-to-have" rather than a critical necessity, and overall, it does not seem to be in high demand. For instance, people continue to use the dominant Google Analytics[1] toolset, sending countless pieces of information in plain text to Google's servers. Similarly, there seems to be little to none concern about survey responses being stored in plaintext on servers. Meanwhile, ChatGPT is emerging as a "data kraken" potentially surpassing Google, Microsoft, and Apple combined in its capacity to collect personal, sensitive, and confidential data. The prevailing rationale appears to be: *"Big companies already know everything about me, so why bother?"*[2] This mindset is highly reminiscent of the state of affairs one or two decades ago regarding end-to-end encryption. At that time, the common sentiment was: *"I have nothing to hide, so I don't need these tools."*[3] However, despite this initial resistance, technological advancements have prevailed. Today, end-to-end encryption is a standard feature in instant messaging applications, widely integrated into products across the board; for instance, Signal and WhatsApp deploy end-to-end encryption by default for personal messaging, using the Signal protocol as their underlying cryptographic design [270, 302].

Consequently, it is the author's utmost belief, hope, and motivation that protective measures against internal attackers will eventually gain similar acceptance and become part of the standard repertoire of security measures worldwide.

Post-Quantum Cryptography Cryptographic failures have a drastically larger effect on the industry as a whole than, say, a weak password policy within a

[1] `https://tagmanager.google.com/?hl=de`

[2] (That is, until there's a data breach, a political shift, or an employee abuses their access—but it'll never happen to me, right?)

[3] (That is, until I do need them, and then using end-to-end encryption makes me look suspicious.)

company. Failing to address these issues in a timely manner may result in catastrophic outcomes for the company and its clients.

One such looming catastrophic failure, which, speaking metaphorically, is undoubtebly a Damocles' Sword, is the seminal cryptanalytic work by Peter Shor [267], which introduced an algorithm for quantum computers that effectively breaks the intractability assumptions of factoring and the discrete logarithm. However, 22 years later, in 2016, the U.S. National Institute of Standards and Technology (NIST) called for a post-quantum cryptography standardization effort and received more than 80 new cryptographic constructions for Key Encapsulation Mechanisms (KEMs) and digital signatures in order to replace widely used cryptographic schemes based on the factoring and discrete logarithm assumptions. In 2024, NIST has published the final standards resulting from this effort, which include the Module-Lattice-Based Key-Encapsulation Mechanism Standard (FIPS 203 ML-KEM) [226], the Module-Lattice-Based Digital Signature Standard (FIPS 204 ML-DSA) [225], and the Stateless Hash-Based Digital Signature Standard (FIPS 205 SLH-DSA) [228]. All of the author's works presented in this thesis are developed using cryptographic primitives based on presumed quantum-safe assumptions, which are also the foundation of these standardized schemes. Moreover, the author has implemented a dedicated post-quantum secure end-to-end encrypted file transfer CLI[4] with authentication based on Short Authentication Strings and provided formal analysis of the security in the universal composability framework in [41].

Random Oracle Another theoretical catastrophic failure, which, however, has not yet materialized as catastrophic in the real world, is the uninstantiability of the Random Oracle (RO). Introduced by [30], the idea is to model a cryptographic hash function as a truly random function, represented by the Random Oracle. However, [82] critiqued this concept, arguing that no cryptographic hash function can meet this definition and that the proven security arguments were incomplete. Consequently and until today, proofs of security that treat a hash function as an Random Oracle are not entirely formally sound. This implies that cryptosystems proven secure under the random oracle model may lose their guaranteed security when the Random Oracle is replaced with an actual cryptographic hash function. It is important

[4] `https://github.com/collapsinghierarchy/noisytransfercli`

to note that, as of now, no real-world attacks exploiting this inconsistency between the model and the implementation of a hash function have been documented. The RO serves as a candidate for the Chekhov's Gun metaphor because it appears in almost every cryptographic construction, such as the standardized post-quantum schemes ML-KEM [226] and ML-DSA [225], yet its full consequences are still unknown.

The author of this thesis conducted research to explore alternative ways of constructing provably secure channels without relying on the RO assumption. In collaboration with Rebecca Schwerdt, Astrid Ottenhues, Roland Gröll, and Prof. Jörn Müller-Quade, the author embarked on devising a new security definition for public key encryption. This definition, in conjunction with authenticated channels, constructs secure channels without the need for a RO and is weaker than adaptive chosen ciphertext attack (CCA2) security. The results of this work were published in the proceedings of PKC 2022 [46] and the full version is available on the IACR Eprint Archive [43]. Subsequently, Sarai Eilebrecht and Laurin Benz joined the team, and the security notion was adapted to fit into the hybrid encryption paradigm. This adaptation was published in the proceedings of PKC 2023 [33] and the full version is available on the IACR Eprint Archive [266].

In addition, the author has explored the construction of efficient CCA2 Double receiver encryption (DRE) schemes, which are used for applications beyond secure channels, in collaboration with Laurin Benz, Sarai Eilebrecht, Roland Gröll, and Prof. Jörn Müller-Quade. The results of this work were published in the proceedings of the 2024 IACR International Conference on Public-Key Cryptography [32]. However, we exclude these results from this thesis for brevity reasons, mentioning them here only for the sake of completeness.

Subsequently, Robert Brede, under the supervision of Wasilij Beskorovajnov and Laurin Benz, applied all the prior results [32, 43, 266] in his master's thesis [68] to construct a Ring-LWE-based key encapsulation mechanism (KEM). This KEM was employed to build a practical, real-world secure channel protocol without relying on random oracles.

2.2. Contribution

The contributions of this thesis are the results of the author's efforts to address threats to real-world applications that may not seem imminent now but can materialize at any time, as illustrated by the metaphors of Damocles' sword and Chekhov's gun.

The author addresses three such threats:

The Quantum Computer Threat The first overarching threat to every cryptographic construction is the advent of quantum computers. This necessitates the use of cryptographic assumptions that are presumably post-quantum, or quantum-safe—that is, assumptions under which no polynomial-time algorithm exists to break them, whether on quantum or classical computers. Unfortunately, it has not always been possible to rely exclusively on such assumptions due to the current state of certain cryptographic constructions. For example, anonymous credentials, such as those in [75], which are used as building blocks in the construction of a contact tracing protocol [48], are not yet available under quantum-safe assumptions. Other than that, the remaining publications [32, 33, 41, 42, 46, 68] do not suffer from this issue.

The Internal Attacker Threat The second threat arises from an internal attacker. Protections against internal attackers are often considered secondary by security practitioners. However, as argued earlier, overall security depends on addressing every detail, and failing to consider internal attackers effectively assumes that internal server components can always be trusted—a premise that has been proven untrue on numerous occasions.

The author's focus on enhancing real-world applications with protections against internal attackers primarily targeted scenarios where such threats could lead to massive breaches or misuse. The first such application was the contact tracing protocol, where the greatest misuse risk is nation-wide surveillance. To mitigate this, a secure protocol [48] was devised under the assumption that central servers are not trustworthy, i.e., they are honest-but-curious.

Additionally, the author identified that many data collection applications and web services gather significant amounts of sensitive data, often to com-

pute simple statistics such as absolute frequencies, standard deviations, and variances. These services are highly vulnerable to breaches with potentially severe consequences. To protect such services from internal attackers, the author devised an outsourced computation protocol [42] that eliminates the need to trust servers to act honestly or for clients to come online after submitting their inputs. This work introduced a third paradigm for outsourced computation. With all three paradigms now available, it is expected that almost any computation on personal data can be securely realized in real-world applications, even in the presence of internal adversaries.

Last but not least, the author identifies end-to-end encryption as one of the simplest and most effective defenses against such an adversary and, accordingly, presents a formal model of short-authentication-string–authenticated end-to-end encryption for information transfer within the Universal Composability (UC) framework [41].

The Random Oracle Threat The third threat is more ambiguous and less tangible: the random oracle methodology. While it has been instrumental in many cryptographic advancements, it remains the subject of ongoing debate. The foundational critique stems from [82], which presented a pathological example of a cryptographic construction that is secure under the random oracle model but loses security guarantees when the random oracle is replaced by a hash function. To date, these results remain primarily of academic interest.

However, this critique has spurred research into cryptographic constructions that explicitly avoid modeling hash functions as random oracles, thereby ensuring formal soundness. The author observed that nearly all such constructions, beyond basic primitives, rely on secure channels at some point. This motivated the development of a secure channel protocol that does not rely on random oracles, building on a long line of research [32, 33, 46, 68].

While the secure channel protocol is still significantly less efficient than its counterparts that use random oracles, it achieves improvements over previous constructions that avoid random oracles.

2.3. Structure

The structure of this thesis is as follows.

Chapter 2: Preliminaries In this chapter, we will delineate all preliminary definitions, theorems, and lemmatas essential for the ensuing chapters.

Chapter 3: Internal Attacker We will begin this chapter with a comprehensive introduction discussing the motivation and current state of the art in protective measures against internal attackers in real-world scenarios. Following this introduction, we will present the results of two joint research efforts [42, 48] conducted by the author and his colleagues, aimed at developing secure applications using cryptographic tools that remain secure even in the presence of internal adversaries, i.e., attackers who have access to the internal state and, consequently, sensitive information of the parties involved in these applications.

Chapter 4: Random Oracle This chapter is dedicated to the author's efforts to explore ways of constructing cryptographic schemes without relying on the random oracle methodology. To properly motivate this research, the chapter begins with a summary of the current state of the debate surrounding the random oracle methodology, including arguments for and against modeling a hash function as a random oracle. Subsequently, we will present two works [33, 46], resulting from joint research by the author and his colleagues, which focus on constructing cryptographic encryption primitives to realize a secure channel without random oracles.

Chapter 5: Conclusion and Future Work In this concluding chapter, the author summarizes the main contributions of the thesis and revisits the guiding questions concerning internal attackers, post-quantum security, and the random oracle methodology. The author reflects on how the case studies and constructions developed in the previous chapters fit into the broader landscape of real-world cryptography and what their limitations are. Finally, the author presents an outline of several directions for future research, highlighting open problems and promising avenues for improving both the theoretical foundations and the practical deployability of the proposed approaches.

Part II.

Contents

3. Preliminaries

In this section, we recap the notations and definitions including frameworks needed throughout this thesis.

3.1. Notation

Throughout this work we will use the following notation.

- Sets and Elements: We denote sets in upper case (e. g. the set M) and elements of a given set in lower case (e. g. $m \in M$)
  - We denote as $\mathcal{CT}$ as a special set for the ciphertext space and $c \in \mathcal{CT}$ as a ciphertext.
  - We denote as $\mathcal{MS}$ as a special set for the message space and $m \in \mathcal{MS}$ as a plaintext.
- For a natural number $n \in \mathbb{N}$, we write $[n]$ for the set $\{0, \cdots, n-1\}$.
- Vectors and Components: Vectors are written in lower-case italic sans-serif (e. g. $\mathsf{u} \in M^n$) and its i-th coefficient is denoted as $\mathsf{u}[i]$
- Matrices are written in upper-case italic sans-serif, e. g. $\mathsf{G}, \mathsf{U}, \mathsf{A}$
- Sampling from a Distribution: We denote $x \leftarrow_{\$} X$ as x being sampled uniformly random from X and $y \leftarrow Y$ as y being sampled according the distribution Y, e. g $\mathsf{e} \leftarrow \mathcal{B}_\theta$ for e being sampled according to the Bernoulli distribution $\mathcal{B}_\theta$
- Security Parameter: We denote the security parameter as λ
- Algorithms are written in italic, e. g., *Alg* or *B*.

  - Special Algorithms: For the special algorithms that we call *adversaries, challengers, oracles, distinguishers, environment*, we denote $\mathcal{A}, C, O, \mathcal{D}, \mathcal{Z}$.
- We denote as $wgt(x)$ the hamming weight of x.

Negligibility We say a function $negl(\cdot)$ is negligible if for every positive value $c \in \mathbb{N}$ and all sufficiently large $\lambda \in \mathbb{N}$ it holds that $negl(\lambda) < \lambda^{-c}$.

Distribution Ensemble A distribution ensemble $X = \{X(\lambda, a)\}_{a \in D, \lambda \in \mathbb{N}}$ is an infinite sequence of random variables indexed by $a \in D$ for some domain D and $\lambda \in \mathbb{N}$.

Computational indistinguishability. Let

$$X = \{X(\lambda, a)\}_{\substack{\lambda \in \mathbb{N} \\ a \in \mathcal{D}}} \quad \text{and} \quad Y = \{Y(\lambda, a)\}_{\substack{\lambda \in \mathbb{N} \\ a \in \mathcal{D}}}$$

be distribution ensembles. We say that X and Y are *computationally indistinguishable*, denoted by $X \overset{c}{\approx} Y$, if for every non-uniform polynomial-time algorithm D there exists a negligible function $negl(\cdot)$ such that for every $\lambda \in \mathbb{N}$ and every $a \in \mathcal{D}$,

$$\left|\Pr[D(X_{\lambda,a}) = 1] - \Pr[D(Y_{\lambda,a}) = 1]\right| \leq negl(\lambda).$$

3.2. Coding Theory

In this thesis we will be concerned only with binary linear codes.

Definition 1 (Binary Linear Code) A *binary linear code* C of length n over the finite field $\mathbb{F}_2$ is a vector subspace of $\mathbb{F}_2^n$. It is parameterised by:

- n, the *length* of the code (codewords);
- k, the *dimension* of the code;
- d, the *minimum distance* of the code (codewords).

Therefore, we say that a binary linear code C is an $[n, k, d]_2$ code.

Definition 2 (Codeword) Let C be a linear code. A vector $c \in C$ is called a *codeword* of C.

Definition 3 (Generator Matrix) A *generator matrix* $G \in \mathbb{F}_2^{k \times n}$ for a *linear code* C is a matrix, which rows form a basis of C.

The process of *applying a code* to a message $m \in \mathbb{F}_2^k$ is carried out by $mG = c$.

Definition 4 (Dual Code) The *dual code* $C^\perp$ of a linear code C is defined by $C^\perp = \{x \in \mathbb{F}_2^n | \langle x, c \rangle = 0, \forall c \in C\}$.

Definition 5 (Parity Check Matrix) A *parity check matrix* $H \in \mathbb{F}_2^{m \times n}$ for a *linear code* C is a matrix, which rows form a basis of the dual code $C^\perp$

From the definition of a parity check matrix it therefore follows that $\forall c \in C : Hc^T = 0$, which is used extensively in most code-based cryptosystems.

Definition 6 (Goppa Code [233]) Let $L = (\alpha_1, \cdots, \alpha_n)$ with $\alpha_i \in \mathbb{F}_{2^m}$ and $g(x) \in \mathbb{F}_{2^m}[X]$ be a monic polynomial of degree t with $g(\alpha_i) \neq 0$ for $i = 1, \cdots, n$, then the linear code,

$$\Gamma(L, g) := \left\{ c \in \mathbb{F}_{2^m}^n : R_c(x) := \sum_{n=1}^{n} \frac{c_i}{x - \alpha_i} \equiv 0 \mod g(x) \right\}$$

is a Goppa code of length n for a Goppa polynomial $g(x)$ with *support* L.

Goppa Code Parity Check Matrix H For goppa codes the the parity check matrix $H \in \mathbb{F}_2^{tm \times n}$ of the goppa polynomial $g(x)$ with degree t and support L is computed in the following way:

$$H = \begin{pmatrix} g(\alpha_1)^{-1} & g(\alpha_2)^{-1} & \cdots & g(\alpha_n)^{-1} \\ g(\alpha_1)^{-1}\alpha_1 & g(\alpha_2)^{-1}\alpha_2 & \cdots & g(\alpha_n)^{-1}\alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ g(\alpha_1)^{-1}\alpha_1^{t-1} & g(\alpha_2)^{-1}\alpha_2^{t-1} & \cdots & g(\alpha_n)^{-1}\alpha_n^{t-1} \end{pmatrix}$$

Goppa Code Generator Matrix G Given the parity check matrix H of a goppa code, we can compute the generator matrix G by solving the following equation: $HG^T = 0$, with the constraint that G must be a full rank matrix.

Dimension, Length and Minimal Distance of a Goppa Code $\Gamma(L, g)$ [23] The length of the goppa code is n. The dimension is given by $k \geq n - mt$, where $deg(g) = t$ and m is the length of the support elements, i.e. $\alpha_i \in \mathbb{F}_{2^m}$. The minimal distance is given by $d \geq t + 1$ and in case of an *irreducible binary goppa code* the distance can be improved up to $d \geq 2t + 1$, which then is able to correct at least t errors.

3.3. Code-Based Cryptography

The previous coding theory definitions can be used to construct the *textbook* McEliece public-key encryption (PKE) cryptosystem (originally published in [208]).

Definition 7 (McEliece Cryptosystem) The original McEliece cryptosystem is defined by the the following three algorithms (GEN, ENC, DEC)

Algorithm 1 $GEN(1^\lambda)$

1: Choose (based on λ) m, n, the support L and an irreducible Goppa-polynomial $g(x)$ with $deg(g) = t$ from $\mathbb{F}_{2^m}[X]$
2: Compute the $k \times n$ generator matrix G
3: Sample an invertible $k \times k$ matrix S
4: Sample a random $n \times n$ matrix P
5: Compute the scrambled generator matrix $G' = SGP$
6: $\hookrightarrow$ sk $:= (S, P, L, g)$ and pk $:= (G', t, m)$

For the sake of simplicity we assume that the message $m \in \mathcal{M}$ has length k, i.e. $\mathcal{M} := \mathbb{F}_2^k$.

Algorithm 2 $ENC(\text{pk}, m)$

1: Sample a random error vector $e \in \mathbb{F}_2^n$ with $wgt(e) = t$.
2: $c = mG' + e$
3: $\hookrightarrow c$

The cryptosystem from definition 7 is only one-way under chosen plaintext attack (OW-CPA) secure but it is not deterministic. For indistinguishability

Algorithm 3 $DEC(\mathsf{sk}, c)$

1: Parse sk as (S, P, L, g)
2: $c' = cP^{-1} = mSG + e'$ with $e' := eP^{-1}$
3: Apply an efficient decoding algorithm on c' in order to get $m' = mS$
4: $\hookrightarrow m = m'S^{-1}$

under chosen plaintext attack (IND-CPA) security one has to use a randomized version of it. In the randomized version the Encryption algorithm takes in addition to an $m \in \mathcal{M}$ also a concatenated randomness $r \in \mathbb{F}_2^l$ such that $ENC(\mathsf{pk}, [r|m]) = [r|m]G' + e = rG'_1 + e + mG'_2$. The key generation and the Decryption procedure can be adapted in a straight forward way. The proofs for the randomized version to achieve IND-CPA security and correctness can be found in [230]. We will use this cryptosystem to construct an IND-CPA secure DRE. In order to prove that our cryptosystem achieves this security we will need the following standard intractability assumptions.

Definition 8 (Indistinguishability Assumption for Goppa Codes) Let D be a probabilistic algorithm. For every $n \in \mathbb{N}$, we define

$$\mathsf{Adv}^{ind}_{D,G}(n) = \mathsf{Pr}\Big[(D(G,t) = 1|(G,t),\mathsf{sk}) \leftarrow GEN(1^n)\Big] - \mathsf{Pr}\Big[D(U,t) = 1|U \leftarrow_{\$} U_{l\times n}\Big]$$

Also we define the advantage function of the problem as follows. For any ω,

$$\mathsf{Adv}^{ind}_{G}(n,\omega) = \max_{D}\Big\{\mathsf{Adv}^{ind}_{D,G}(n)\Big\} \tag{3.1}$$

where the maximum is over all D with time-complexity ω. We say G is indistinguishable if, for every polynomially bounded ω and every sufficiently large n, $\mathsf{Adv}^{ind}_{G}(n,\omega)$ is negligible.

The assumption in definition 8 states that the *scrambling* process in Step 5 of definition 7 produces a matrix G' that is indistinguishable from a random binary matrix. Equivalently, this asserts that *scrambled* irreducible binary Goppa codes are indistinguishable from random linear codes. Considerable effort has since been devoted to relaxing this assumption to other code families—most notably Reed–Solomon codes. The motivation was to reduce the size of the Goppa generator or parity-check matrix and, hence, the public

key. Unfortunately, most of these proposed relaxations were later shown to be invalid such as the case with Reed–Solomon codes [114].

In order to state that our McEliece ciphertext elements are pseudorandom we will additionally need the LPN assumptions from definition 9 and definition 10. This is easy to see given that the McEliece ciphertext is of the form $c = mG' + e$ and with definition 8 we can safely replace G' with a uniformly random binary matrix $U_{l\times n}$ such that the ciphertext now becomes $c = mU + e$. Given that e is a sample from a Bernoulli distribution the ciphertext becomes effectively an LPN sample from definition 9 and definition 10 and thus is pseudorandom by assumption.

Definition 9 (LPN Search Problem (LPNSP)) Let s be a random binary string of length l. We consider the Bernoulli distribution $\mathcal{B}_\theta$ with parameter $\theta \in (0, \frac{1}{2})$. Let $Q_{s,\theta}$ be the following distribution:

$$\{(a, \langle s, a\rangle \oplus e) | a \leftarrow_\$ \{0,1\}^l, e \leftarrow \mathcal{B}_\theta\}$$

For an adversary $\mathcal{A}$ trying to discover the random string s, we define its advantage as

$$\mathsf{Adv}^{LPN_\theta}_{\mathcal{A}}(l) = \Pr\left[\mathcal{A}^{Q_{s,\theta}} = s | s \leftarrow_\$ \{0,1\}^l\right]$$

The LPN_θ is hard if the advantage of all PPT adversaries $\mathcal{A}$ that make a polynomial number of oracle ($Q_{s,\theta}$) queries is negligible.

Definition 10 (LPN Distinguishing Problem (LPNDP)) Let s and $Q_{s,\theta}$ be as in 9. Let $\mathcal{A}$ be a PPT adversary, whose distinguishing advantage between $Q_{s,\theta}$ and the uniform distribution $\mathcal{U}_{l+1}$ after issuing at most q queries is defined as follows

$$\mathsf{Adv}^{LPNDP_\theta}_{\mathcal{A}}(q, l) = \left|\Pr[\mathcal{A}^{Q_{s,\theta}} = 1 | s \leftarrow_\$ \{0,1\}^l] - \Pr[\mathcal{A}^{\mathcal{U}_{l+1}} = 1]\right|$$

3.4. Lattice-Based Cryptography

We assume that the reader is familiar with the algebraic concepts of field and ring extensions. We also assume that the general and well-known theorems of algebraic number theory concerning lattices and rings of integers are known. Otherwise, we refer to standard textbooks such as Marcus' *Number Fields*

[206] and the volume edited by Cassels and Fröhlich [94], as well as to surveys on lattice-based cryptography and the Learning with Errors (LWE) problem [213, 247].

In addition to code-based cryptosystems we will also describe constructions of lattice-based cryptosystems. The central assumption of such cryptosystems is the LWE assumption [246] and its variations.

Definition 11 (Search Learning with errors (LWE) Problem [246]) Let $q \geq 2$ be a prime and let $\chi : \mathbb{Z}_q \to \mathbb{R}^+$ be some probability distribution, i.e. the error distribution on $\mathbb{Z}_q$. Further, let n be an integer and $s \in \mathbb{Z}_q^n$ be a vector and $A_{s,\chi}$ be the following distribution:

$$\{(a, \langle s, a\rangle \oplus e) | a \longleftarrow_{\$} \mathbb{Z}_q^n, e \leftarrow \chi\}$$

For an adversary $\mathcal{A}$ trying to discover the random string s, we define its advantage as

$$\mathsf{Adv}_{\mathcal{A}}^{LWE_{q,n,\chi}}(n) = \mathsf{Pr}\left[\mathcal{A}^{A_{q,\chi}} = s | s \longleftarrow_{\$} \mathbb{Z}_q^n\right]$$

The $LWE_{q,n,\chi}$ is hard if the advantage of all PPT adversaries $\mathcal{A}$ that make a polynomial number of oracle ($A_{q,n,\chi}$) queries is negligible.

In [246], Regev presented a quantum reduction showing that solving the average-case LWE problem is at least as hard as solving certain worst-case lattice problems, i.e. the (approximate) Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). Moreover, he provided a reduction demonstrating that an oracle for random LWE instances can be used to solve these worst-case lattice problems. These results furnish strong evidence for the plausibility of the LWE assumption and thus they are one of the most promising foundations nowadays for post-quantum cryptography candidates.

The stated search variant of the LWE problem can be reduced to solving the decision variant, which is omitted here. In [20, Lemma 2] it was proved that the problem NLWE is equivalent to the standard form of decision LWE where the LWE secret s is sampled from $\mathbb{Z}_q^n$ instead of χ^n. Therefore we present here only the normal form variant of the decision LWE problem.

Definition 12 (Normal form LWE (NLWE) Problem [63]) Let $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$ be integers and χ be an error distribution. The advantage

of a PPT adversary $\mathcal{A}$ for the (normal-form) $\mathsf{NLWE}_{n,m,q,\chi}$ problem, denoted by $\mathsf{Adv}^{\mathsf{NLWE}_{n,m,q,\chi}}_{\mathcal{A}}(\lambda)$, is defined as

$$\mathsf{Adv}^{\mathsf{NLWE}_{n,m,q,\chi}}_{\mathcal{A}}(\lambda) \coloneqq \left|\mathsf{Pr}[\mathcal{A}(A, s^\top A + e^\top) = 1] - \mathsf{Pr}[\mathcal{A}(A, b^\top) = 1]\right|,$$

where $A \hookleftarrow\$\ \mathbb{Z}_q^{n\times m}, s \leftarrow \chi^n$ and $e \leftarrow \chi^m$. The $\mathsf{NLWE}_{n,m,q,\chi}$ problem is hard if $\mathsf{Adv}^{\mathsf{NLWE}_{n,m,q,\chi}}_{\mathcal{A}}$ is negligible in λ for all PPT adversaries $\mathcal{A}$

Furthermore, this work also considers the ring variant Ring-LWE (RLWE), where one replaces the vector space $\mathbb{Z}_q^n$ by the canonical-embedded quotient ring $H_q \coloneqq H_K/qH_K$ for a number field K. In this setting, the underlying lattices are ideal lattices arising from (fractional) ideals of the ring of integers, viewed in Euclidean space via the canonical embedding. More specifically, we use the non-dual version, in which the secret is sampled in R instead of its dual $R^\vee$. The RLWE distribution in the canonical embedding is defined as follows.

Definition 13 (RLWE Distribution) Let H_K be the canonical embedded ring of integers for a number field K. For a modulus q, distribution χ on $H_q \coloneqq H_K/qH_K$, and a secret $s \in H_q$, the RLWE distribution $A_{s,q,\chi}$ is defined as sampling the value a uniformly from H_q, $e \leftarrow \chi$ and outputting $(a, as + e) \in H_q \times H_q$.

Based on this distribution, there are two versions of the problem, decision-RLWE and search-RLWE. The decision-$\mathsf{RLWE}_{q,\chi}$ asks to distinguish between a uniform distribution on $H_q \times H_q$ and $A_{s,q,\chi}$, whereas the search-$\mathsf{LWE}_{q,\chi}$ asks to find s, given samples from $A_{s,q,\chi}$. In its standard form, s is drawn from a uniform distribution and then fixed. In contrast, in the normal $\mathsf{RLWE}_{q,\chi}$ ($\mathsf{NRLWE}_{q,\chi}$), s is chosen from χ and then fixed. (cf. [204]) The advantage of an attacker $\mathcal{A}$ against the decision-NRLWE (d-NRLWE) is its advantage of distinguishing, formally

$$\begin{aligned} Adv^{d-NRLWE_{q,\chi}}_{\mathcal{A}}(\kappa) = |Pr\left[\mathcal{A}(a,b) = 1 \mid (a,b) \leftarrow \mathcal{U}(H_q \times H_q)\right] \\ -Pr\left[\mathcal{A}(a,b) = 1 \mid (a,b) \leftarrow A_{s,q,\chi}, s \leftarrow \chi\right]|. \end{aligned}$$

On the contrary, in the search-NRLWE (s-NRLWE), the advantage of an attacker $\mathcal{A}$ is its probability of finding s, formally

$$Adv^{s-NRLWE_{q,\chi}}_{\mathcal{A}}(\kappa) = Pr\left[\mathcal{A}(a,b) = s \mid (a,b) \leftarrow A_{s,q,\chi}, s \leftarrow \chi\right].$$

The d-NRLWE$_{q,\chi}$ (or s-NRLWE$_{q,\chi}$) assumption states that for every PPT attacker $\mathcal{A}$, the advantage $adv_{\mathcal{A}}^{d-NRLWE_{q,\chi}}(\kappa)$ (or $adv_{\mathcal{A}}^{s-NRLWE_{q,\chi}}(\kappa)$) is negligible in κ. If the lattice Λ is clear from the context, we write d-NRLWE$_{q,r}$ for $r \in \mathbb{R}$ as shorthand for d-NRLWE$_{q,\chi}$ with $\chi = \mathcal{D}_{\Lambda,r}$.

LWE Trapdoor Function The main ingredient for lattice-based constructions will be the trapdoor mechanism from [211]. The trapdoor function is defined as $f_{a,e}(s) = as + e \mod q$, i.e. the function of the LWE samples. The authors defined trapdoors over $\mathbb{Z}^n$. As this work uses them over a ring of integers R of a number field K, we adapt the definition to this setting.

Definition 14 (g-trapdoor, adapted from [211]) Let R be the ring of integers of a number field K and a modulus q. Let $m > \omega \geq 1$ be integers, $a \in R_q^m$ and $g \in R_q^\omega$. A g-trapdoor $T \in R^{\omega \times (m-\omega)}$ is a matrix for which $(T, I)a = gh$ for some invertible $h \in R_q$. We refer to h as the tag or label of the trapdoor.

The vector g is called *gadget vector*. It is chosen in such a way that inverting $gs + e$ is efficient as is shown below. Let

$$g = \begin{pmatrix} 1 \\ 2 \\ 4 \\ \vdots \\ 2^{k-1} \end{pmatrix} \in R_q^k, k = \lceil \log q \rceil.$$

For a prime $q = \sum_{i=1}^{k} q_i 2^{i-1}$, the following matrix B_g is a basis of the lattice $\Lambda_q^\perp(g)$ [211].

$$B_g = \begin{pmatrix} 2 & & & & & & q_1 \\ -1 & 2 & & & & & q_2 \\ & -1 & & & & & q_3 \\ & & \ddots & \ddots & & & \vdots \\ & & & -1 & 2 & & q_{k-1} \\ & & & & -1 & & q_k \end{pmatrix} \in R_q^{k \times k} \tag{3.2}$$

Hereby, each $q_i \in \{0, 1\}$ is interpreted as constant polynomial.

Inverting the Trapdoor Function Inverting the LWE function $f_{g,e}(s) = gs + e \mod q$ works, if $e \in \mathcal{P}_{1/2}(q \cdot B^{-T})$, where B is either B_g or its Gram-Schmidt orthogonalization [195, 211]. Our student Robert Brede provided more details on this inversion process and we refer the interested reader to his master's thesis [68]. Also Micciancio and Peikert [212] and Lai, Cheung, and Chow [195] provide a complete overview over all processes regarding the inversion and the generation of the LWE trapdoor function. Based on the inversion of $gs + e$, the $Invert_g$ function recovers s from $c = as + e$ with the help of a g-trapdoor T with tag h. (cf. algorithm 4)

Algorithm 4 $Invert_g(c, T, h)$

1: $b = (T, I) \cdot c = (T, I) \cdot a \cdot s + (T, I) \cdot e = g \cdot h \cdot s + (T, I) \cdot e = g \cdot h \cdot s + \hat{e}$
2: Calculate $\hat{s}$ with $b = g \cdot \hat{s} + \hat{e}$ for some small $\hat{e}$ as described in [68]
3: $s = h^{-1}\hat{s}$
4: $\hookrightarrow s$

The $Invert_g$ function returns the correct s, if the error $\hat{e}$ is sufficiently small (cf. [68, 195, 211]) Note that the $Invert_g$ function only calculates the secret s. The error e can then be calculated as $e = c - as$. To generate an a with a corresponding g-trapdoor T with tag h, choose $a' \in R_q^\omega$, a tag h and a trapdoor T and set $a = (a', hg - Ta')$.

3.5. Cryptographic Schemes

The previous sections examined only the foundations of exact cryptographic constructions. In this section, we would like to examine the formal definitions of several cryptographic primitives that are agnostic to specific mathematical assumptions. The according formal security definitions can be found in the next section section 3.7.

3.5.1. Public-Key (asymmetric) Cryptography

Definition 15 (Rerandomizable PKE Σ_R [243]) We define the syntax for rerandomizable encryption $\Sigma_R = (GEN, ENC, DEC, ReRand)$ in the following way.

- *GEN* is a randomized algorithm which outputs a public key pk and a corresponding secret key sk.
- *ENC* is a randomized Encryption algorithm which takes a plaintext $\mathsf{m} \in \mathcal{M}$ and a public key and outputs a ciphertext.
- *ReRand* is a randomized algorithm which takes a ciphertext and outputs another ciphertext.
- *DEC* is a deterministic Decryption algorithm which takes a private key and a ciphertext, and outputs either a plaintext or an error indicator $\perp$.

Definition 16 (Correctness of Σ_R [243]) For the property of correctness to hold we require Σ_R to satisfy the following conditions.

- $\forall \mathsf{m} \in \mathcal{M} : \mathit{DEC}(\mathsf{sk}, \mathit{ENC}(\mathsf{pk}, \mathsf{m})) = \mathsf{m}$
- For every independently chosen $(\mathsf{pk}', \mathsf{sk}') \leftarrow \mathit{GEN}(1^\lambda)$, the sets of honestly generated ciphertexts under pk and pk′ are disjoint, with overwhelming probability over the randomness of *GEN*.
- For every plaintext m and every (honestly generated) ciphertext $c \leftarrow \mathit{ENC}(\mathsf{pk}, \mathsf{m})$, the distribution of $\mathit{ReRand}(\mathsf{c})$ is identical to $\mathit{ENC}(\mathsf{pk}, \mathsf{m})$.
- For every (purported) ciphertext c and every $c' \leftarrow \mathit{ReRand}(c)$, we must have $\mathit{DEC}(\mathsf{sk}, \mathsf{c}') = \mathit{DEC}(\mathsf{sk}, \mathsf{c})$

To fulfill the second correctness property, one can include the public key in all ciphertexts and copy/check it during rerandomization/decryption. This ensures that a ciphertext can only be decrypted under at most one key pair.

Definition 17 A homomorphic encryption scheme consists of a tuple of four probabilistic polynomial time (PPT) algorithms (*GEN*, *ENC*, *DEC*, *Eval*)

- The key generation algorithm $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathit{GEN}(1^\lambda)$ takes the security parameter λ and produces a random key pair (pk, sk), where sk denotes the secret key for decrypting a ciphertext that is held private by the ciphertext receiver and pk denotes the public key that is used to produce a ciphertext of a given message.
- The probabilistic encryption algorithm $\mathsf{c} \leftarrow \mathit{ENC}(\mathsf{pk}, m; r)$ takes a message $m \in M$ from the message space M and a public key and produces a ciphertext $\mathsf{c} \in \mathcal{CT}$ from the ciphertext space $\mathcal{CT}$ using randomness r.

- The evaluation algorithm $\mathsf{c}_o \leftarrow \mathit{Eval}(C, ek, \mathsf{c}_1, \ldots, \mathsf{c}_n)$ takes a circuit C, an evaluation key ek and several ciphertexts $\mathsf{c}_1, \ldots, \mathsf{c}_n$ and evaluates the circuit on the given ciphertexts, resulting in a output ciphertext c_o.
- The deterministic decryption algorithm $m = \mathit{DEC}(\mathsf{sk}, \mathsf{c})$ takes a ciphertext $\mathsf{c} \in \mathcal{CT}$ from the ciphertext space $\mathcal{CT}$ and computes the message corresponding to c.

We require the scheme to be correct. That is, for every message $m \in M$ out of the message space and every key pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathit{GEN}(1^\lambda)$ we require that

$$C(m_1, \ldots, m_n) = \mathit{DEC}(\mathsf{sk}, C(\mathit{ENC}(\mathsf{pk}, m_1), \ldots, \mathit{ENC}(\mathsf{pk}, m_n))$$

for a given circuit C.

We will later rely on a primitive that, informally, "ties together" ciphertexts under different public keys. DRE encrypts a plaintext to two ciphertexts using two different public keys with the guarantee, that these ciphertexts decrypt to the same plaintext.

Definition 18 ([46]) A double receiver encryption scheme consists of three PPT algorithms $(\mathit{GEN}, \mathit{ENC}, \mathit{DEC})$

- The key generation algorithm $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathit{GEN}(1^\lambda)$ takes the security parameter λ and produces a random key pair (pk, sk), where sk denotes the secret key for decrypting a ciphertext which is held private by the ciphertext receiver and pk denotes the public key that is used to produce a ciphertext of a given message.
- The probabilistic encryption algorithm $\mathsf{c} \leftarrow \mathit{ENC}(\mathsf{pk}_1, \mathsf{pk}_2\, m; r)$ takes a message $m \in M$ from the message space M and the public keys of two receivers and produces a ciphertext $\mathsf{c} \in \mathcal{CT}$ from the ciphertext space $\mathcal{CT}$ using some additional randomness r.
- The deterministic decryption algorithm $m = \mathit{DEC}(\mathsf{sk}_i, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{c})$ takes a ciphertext $\mathsf{c} \in \mathcal{CT}$ from the ciphertext space $\mathcal{CT}$ and computes the message corresponding to c and the public keys of the intended two receivers.

Overall it can be summarized as follows:

$$\begin{aligned} \mathit{GEN}: &\quad 1^\lambda \mapsto (\mathsf{sk}, \mathsf{pk}) \\ \mathit{ENC}: &\quad (\mathsf{pk}_1, \mathsf{pk}_2, m) \mapsto \mathsf{c} \\ \mathit{DEC}: &\quad (\mathsf{sk}_i, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{c}) \mapsto m \text{ where } i \in \{1, 2\} \end{aligned}$$

We will sometimes use the notation $(\mathsf{sk}^R, \mathsf{pk}^R)$ and $(\mathsf{sk}^S, \mathsf{pk}^S)$ for the key pairs of two independent users, analogous to the notation $(\mathsf{sk}_1, \mathsf{pk}_1)$ and $(\mathsf{sk}_2, \mathsf{pk}_2)$.

In addition, we will require the function $\mathsf{f}_{\mathrm{Key}}$, which checks if the key pair $(\mathsf{sk}, \mathsf{pk})$ is well-formed.

$$\mathsf{f}_{\mathrm{Key}}: \quad (\mathsf{sk}, \mathsf{pk}) \mapsto \begin{cases} \mathit{true} \\ \mathit{false}. \end{cases}$$

This function plays a special role in the construction of a secure channel from DRE. This function returns *true* if the given tuple $(\mathsf{sk}, \mathsf{pk})$ is a valid output of the *GEN* algorithm. As far as our research suggests, this function is admissible in every DRE scheme we are aware of at this point in time.

3.5.2. Secret-Key (symmetric) Cryptography

Definition 19 (Pseudorandom Generator PRG [181]) Let l be a polynomial and let G be a deterministic polynomial-time algorithm such that for any n and any input $\mathsf{s} \in \{0,1\}^{\mathsf{n}}$ the result $\mathsf{G}(\mathsf{s})$ is a string of length $l(\mathsf{n})$. We say that G is a pseudorandom generator if the following condition hold:

1. **Expansion**: For every n it holds that $l(\mathsf{n}) > \mathsf{n}$
2. **Pseudorandomness**: For any PPT algorithm D, there is a negligible function negl such that

   $$|Pr[\mathsf{D}(\mathsf{G}(\mathsf{s})) = 1] - Pr[\mathsf{D}(\mathsf{r}) = 1]| \leq \mathsf{negl}(\mathsf{n})$$

   where the first probability is taken over uniform choice of $\mathsf{s} \in \{0,1\}^{\mathsf{n}}$ and the randomness of D, and the second probability is taken over uniform choice of $\mathsf{r} \in \{0,1\}^{l(\mathsf{n})}$ and the randomness of D.

Definition 20 (SKE) A secret-key encryption (SKE) scheme is a pair of PPT algorithms $SKE = (GEN_{SKE}, ENC_{SKE}, DEC_{SKE})$ with key space $\mathcal{K}_{\text{ske}}$ and ciphertext space $\mathcal{CT}_{\text{ske}}$ with:

- The key generation algorithm $\mathsf{dk} \leftarrow GEN_{SKE}(1^\lambda)$ takes the security parameter λ and produces a symmetric key $\mathsf{dk} \in \mathcal{K}_{\text{ske}}$.
- $\mathsf{c} = ENC_{SKE}(\mathsf{dk}, m)$: The deterministic *encryption algorithm* takes as input a key $\mathsf{dk} \in \mathcal{K}_{\text{ske}}$ and a message m, and outputs a ciphertext c.
- $m = DEC_{SKE}(\mathsf{dk}, \mathsf{c})$: The deterministic *decryption algorithm* takes as input a key $\mathsf{dk} \in \mathcal{K}_{\text{ske}}$ and a ciphertext c, and outputs a message m' (which may be $\perp$)

We require that for all $\mathsf{dk} \in \mathcal{K}_{\text{ske}}$ it holds that

$$m = DEC_{SKE}(\mathsf{dk}, ENC_{SKE}(\mathsf{dk}, m)) .$$

Definition 21 (MAC) A message authentication code (MAC) scheme MAC with key space $\mathcal{K}_{\text{mac}}$ consists of the three algorithms $(GEN_{MAC}, SIGN_{MAC}, VER_{MAC})$, where

- The key generation algorithm $\mathsf{mk} \leftarrow GEN_{MAC}(1^\lambda)$ takes the security parameter λ and produces a symmetric MAC key $\mathsf{mk} \in \mathcal{K}_{\text{mac}}$.
- $\sigma \leftarrow SIGN_{MAC}(\mathsf{mk}, m)$: The randomized *signing algorithm* takes as input a signing key $\mathsf{mk} \in \mathcal{K}_{\text{mac}}$ and a message m, and outputs a tag σ.
- $b = VER_{MAC}(\mathsf{mk}, m, \sigma)$: The deterministic *verification algorithm* takes as input a signing key $\mathsf{mk} \in \mathcal{K}_{\text{mac}}$, a message m and a tag σ, and outputs $b = 1$ if $\sigma \leftarrow SIGN_{MAC}(\mathsf{mk}, m)$ and $b = 0$ otherwise.

Definition 22 (Hash Function) A hash function is a function $\mathsf{H} : \{0,1\}^* \rightarrow \{0,1\}^\ell$ mapping bit strings of any length to bit strings of a fixed length ℓ.

3.5.3. Hybrid Cryptography (KEM-DEM)

Initially introduced in [115], the KEM-DEM framework is a specialized form of hybrid encryption that combines the advantages of both public-key and symmetric encryption. The symmetric encryption component allows for the

encryption of long plaintexts, while the KEM component, which asymmetrically encrypts the symmetric key, ensures secure key distribution without the challenges associated with symmetric encryption's key distribution problem. With KEMs, the terminology differs somewhat from that of regular PKE schemes. We are no longer *encrypting* the symmetric key, but rather *encapsulating* it. Therefore, the ciphertexts are now called *encapsulations*.

Definition 23 (KEM) A key encapsulation mechanism (KEM) is given by a set of three PPT algorithms (GEN, ENC, DEC) with

- The key generation algorithm $(\mathsf{pk}, \mathsf{sk}) \leftarrow GEN(1^\lambda)$ takes the security parameter λ and produces a random key pair $(\mathsf{pk}, \mathsf{sk})$, where sk denotes the secret key for decapsulating which is held private by the receiver and pk denotes the public key that is used to produce an encapsulation and a symmetric key.
- The probabilistic encapsulation algorithm $(\mathsf{c}, \mathsf{k}) \leftarrow ENC(\mathsf{pk}; r)$ outputs a public key and produces an encapsulation $\mathsf{c} \in \mathcal{CT}$ from the encapsulation space $\mathcal{CT}$ and a fresh symmetric key k using some additional randomness r.
- The deterministic decapsulation algorithm $\mathsf{k} = DEC(\mathsf{sk}, \mathsf{c})$ takes an encapsulation $\mathsf{c} \in \mathcal{CT}$ from the encapsulations space $\mathcal{CT}$ and computes the encapsulated symmetric key k.

Overall it can be summarized as follows:

$$GEN : 1^\lambda \mapsto (\mathsf{sk}, \mathsf{pk}), \qquad ENC : \mathsf{pk} \mapsto (\mathsf{k}, \mathsf{c}), \qquad DEC : (\mathsf{sk}, \mathsf{c}) \mapsto \mathsf{k}$$

such that the correctness property holds, i.e. $\mathsf{k} = DEC(\mathsf{sk}, C)$ whenever $(\mathsf{sk}, \mathsf{pk}) \leftarrow GEN(1^\lambda)$ and $(\mathsf{k}, \mathsf{c}) \leftarrow ENC(\mathsf{pk})$.

The data encapsulation mechanism (DEM) part is symmetric in nature and is formalized as follows.

Definition 24 (DEM) A data encapsulation mechanism (DEM) is given by a set of two PPT algorithms $(ENC_{\mathrm{DEM}}, DEC_{\mathrm{DEM}})$ with

- $ENC_{\mathrm{DEM}}(\mathsf{k}, m)$: The deterministic *data encryption algorithm* takes as input a key $\mathsf{k} \in \mathcal{K}_{\mathrm{DEM}}$ and a message m, and outputs a ciphertext c.

- $DEC_{\mathrm{DEM}}(\mathsf{k}, \mathsf{c})$: The deterministic *decryption algorithm* takes as input a key $\mathsf{k} \in \mathcal{K}_{\mathrm{DEM}}$ and a ciphertext c, and outputs a message m' (which may be the special symbol $\perp$)

Overall it can be summarized as follows:

$$ENC_{\mathrm{DEM}} : (\mathsf{k}, m) \mapsto \mathsf{c}, \qquad DEC_{\mathrm{DEM}} : (\mathsf{k}, \mathsf{c}) \mapsto m$$

such that $\mathrm{m} = DEC_{\mathrm{DEM}}(\mathsf{k}, \mathsf{c})$ whenever $c \leftarrow DEC_{\mathrm{DEM}}(\mathsf{k}, m)$ (correctness).

The KEM-DEM framework comes in two flavors which slightly differ in the combination of the KEM and DEM. One construction–which we call *single-message* communication– generates a fresh symmetric key for each encryption of a message. Oftentimes such keys are also called *ephemeral*. They are by nature short-lived in contrast to *static* keys that might be arbitrarily long-living, which are used in order to encrypt with the same key across multiple sessions. The ephemeral variant is the original definition of the KEM-DEM framework and intuitively yields a (hybrid) PKE scheme (GEN', ENC', DEC') where $GEN' \equiv GEN$ and

$ENC'(\mathsf{pk}, m)$:
- $(\mathsf{k}, \mathsf{c}_1) \leftarrow ENC(\mathsf{pk})$.
- $\mathsf{c}_2 \leftarrow \mathrm{DEM}.ENC(\mathsf{k}, m)$.

↪ Return $(\mathsf{c}_1, \mathsf{c}_2)$.

$DEC(\mathsf{sk}, (\mathsf{c}_1, \mathsf{c}_2))$:
- $\mathsf{k} \leftarrow DEC(\mathsf{sk}, \mathsf{c}_1)$.
- $m \leftarrow \mathrm{DEM}.DEC(\mathsf{k}, \mathsf{c}_2)$.

↪ Return m.

3.5.4. Deterministic Public-Key Encryption

Deterministic public-key encryption (D-PKE) replaces the randomness of standard (probabilistic) PKE by requiring the encryption algorithm to be deterministic. Formally, a PKE scheme $\Pi = (GEN, ENC, DEC)$ is called *deterministic* if ENC is a deterministic function, i.e. for every public key pk and every plaintext m, repeated encryptions of m under pk always yield the same ciphertext. This property is attractive for applications such as encrypted databases or key-wrap mechanisms, but it immediately rules out standard semantic security IND-CPA. An adversary can always test equality of plaintexts by testing equality of ciphertexts.

To obtain meaningful security for D-PKE, one restricts attention to *high-entropy* message distributions, following the entropic-security paradigm of Dodis and Smith [130] and subsequent work on deterministic encryption [27, 31, 154]. The basic measure of uncertainty is *min-entropy*.

Definition 25 (Min-Entropy) Let X be a distribution over bitstrings. We say that X has (pointwise) min-entropy ν if, for every $k \in \mathbb{N}$ and every $x^\star \in \{0,1\}^*$,

$$\Pr\left[x = x^\star \mid x \leftarrow X\right] \leq 2^{-\nu(k)}.$$

We say that X has *high* min-entropy if $\nu(k) \in \omega(\log k)$.

Intuitively, min-entropy bounds the maximum point probability of the distribution. No single plaintext occurs with probability more than $2^{-\nu(k)}$. A uniform distribution over $\{0,1\}^n$ has min-entropy n. By contrast, natural-language plaintexts usually have low min-entropy because many strings are extremely unlikely or impossible.

Multi-message security. A key insight of Bellare, Boldyreva, and O'Neill [27] and Bellare et al. [31] is that, for D-PKE, security for a single ciphertext does not in general imply security for multiple ciphertexts, even when each message individually comes from a high min-entropy distribution. Consequently, if a public key is intended to be used more than once, security definitions for deterministic encryption must inherently consider *vectors* of plaintexts rather than single messages.

Given a public key pk and a vector of messages

$$x = (x[1], \ldots, x[\ell]),$$

we write

$$c \leftarrow \mathit{ENC}(\mathsf{pk}, x)$$

for component-wise encryption, i.e. $c[i] := \mathit{ENC}(\mathsf{pk}, x[i])$ for all $1 \leq i \leq \ell$. Security then asks that encryptions of two such vectors x_0, x_1 drawn from high min-entropy message spaces are indistinguishable, under syntactic restrictions on the adversary that prevent it from trivially encoding information into message lengths or equality patterns.

Single-receiver vs. multi-receiver security. For randomized PKE under standard IND-CPA or IND-CCA notions, it is well known that security for a single honestly generated key pair lifts to security for any polynomial number of independently generated key pairs via a simple hybrid argument. The multi-user advantage is at most a polynomial factor larger than the single-user advantage.

Bellare, Dowsley, and Keelveedhi [29], however, showed that this implication can fail dramatically for deterministic schemes. There exist D-PKE schemes that are secure for a single user but become insecure as soon as an adversary is given ciphertexts under two different public keys. This motivates a *multi-receiver* security notion, called mIND-DE in [29, 31], where an adversary obtains encryptions of high-entropy message matrices across many users at once and must distinguish between two such matrices.

In the notation adopted later in this chapter, an mIND-DE adversary

$$A = (A_c, A_m, A_g, nm(\cdot), \nu(\cdot))$$

first chooses common state st via A_c, then outputs two message matrices (M_0, M_1) via A_m, where each matrix has $nm(k)$ rows (messages per user) and $\nu(k)$ columns (users). The experiment generates independent key pairs for each user and encrypts the i-th row of M_b under the i-th public key for a hidden bit b. Finally, A_g must guess the challenge bit b. A high-min-entropy condition requires that no fixed plaintext appears with probability greater than $2^{-\nu(k)}$ across all entries of M_b. Syntactic legitimacy conditions prevent trivial encodings via lengths or equality patterns.

IND-DE and mIND-DE. These ideas are formalized via two game-based notions.

Definition 26 (IND-DE Security) Let $\Pi = (GEN, ENC, DEC)$ be a deterministic PKE, and let $I = (I_c, I_m, I_g)$ be an IND-DE-adversary. Here, I_c outputs a common state string st. The algorithm I_m outputs two message vectors (x_0, x_1) of equal length and with identical length patterns. The algorithm I_g receives a public key pk, the component-wise encryption of one of these vectors, and st, and must guess which vector was encrypted. The message generator I_m is required to produce high-min-entropy message vectors. Legitimacy conditions restrict the use of vector length and equality patterns via a fixed reference ensemble [31].

The IND-DE advantage of I against Π is defined by

$$\mathcal{A}^{\mathsf{IND\text{-}DE}}_{I,\Pi}(k) := \left|\Pr[\mathsf{Exp}^{\mathsf{IND\text{-}DE}}_{I,\Pi}(k) = 1] - \tfrac{1}{2}\right|,$$

We say that Π is IND-DE-secure if $\mathcal{A}^{\mathsf{IND\text{-}DE}}_{I,\Pi}(k)$ is negligible for every legitimate high-min-entropy adversary I with trivial common state I_c.

Definition 27 (mIND-DE Security) Let $\Pi = (GEN, ENC, DEC)$ be a PKE, and let $A = (A_c, A_m, A_g; A_{\mathsf{nm}}(\cdot), A_v(\cdot))$ be a mIND-DE-adversary as in Definition 10. Here, A_c outputs a common state string st, and A_m outputs two message matrices (M_0, M_1) of dimensions $A_{\mathsf{nm}}(k) \times A_v(k)$, where each column corresponds to a user and each row to a message encrypted under that user's public key. The experiment generates $A_v(k)$ independent key pairs, encrypts the i-th row of M_b under the i-th public key for a hidden bit b, and gives all resulting ciphertexts together with st to A_g, which must guess b. As in the single-receiver case, A_m must output high-min-entropy messages. Additional legitimacy conditions restrict trivial encodings via lengths and equality patterns.

The mIND-DE advantage of A against Π is defined by

$$\mathcal{A}^{\mathsf{mIND\text{-}DE}}_{A,\Pi}(k) := \left|\Pr[\mathsf{Exp}^{\mathsf{mIND\text{-}DE}}_{A,\Pi}(k) = 1] - \tfrac{1}{2}\right|,$$

We say that Π is mIND-DE-secure if $\mathcal{A}^{\mathsf{mIND\text{-}DE}}_{A,\Pi}(k)$ is negligible for every legitimate high-min-entropy adversary A in the considered class (for example, with a trivial common state and a fixed number of users).

The line of work by Bellare et al. [27, 29, 31], together with the entropic-security framework of Dodis and Smith [130] and the unified construction of Fuller et al. [154], show that these min-entropy-based indistinguishability notions are both achievable and robust. They admit generic constructions from standard building blocks (such as trapdoor functions with hardcore bits) and support natural composition and multi-user reasoning. In particular, the $\mathsf{mIND\text{-}DE}_2$ security of the *Encrypt-with-Hardcore* deterministic PKE of Fuller et al. underlies the security proof of our detDKEM and, ultimately, of the sender-binding key encapsulation mechanism (SB-KEM) constructions built from DRE in Section 5.3.2.

3.6. Anonymous Authentication

In Section 4.3.4.3 we employ an periodic n-time anonymous authentication scheme in order to protect against sybil attacks.

Definition 28 (Periodic n-time Anonymous Authentication [75]) The algorithms $\Sigma_{\text{tok}} = (GEN_{\mathcal{I}}, GEN_{\mathcal{U}}, OBTAIN, SHOW, IDENTIFY)$ are defined as follows:

- $GEN_{\mathcal{I}}(1^k)$ is the key generation algorithm of the e-token issuer $\mathcal{I}$. It outputs a key pair $(\mathsf{pk}_{\mathcal{I}}, \mathsf{sk}_{\mathcal{I}})$.
- $GEN_{\mathcal{U}}$ creates the user's key pair $(\mathsf{pk}_{\mathcal{U}}, \mathsf{sk}_{\mathcal{U}})$ analogously.
- $OBTAIN(\mathcal{U}(\mathsf{pk}_{\mathcal{I}}, \mathsf{sk}_{\mathcal{U}}, n), \mathcal{I}(\mathsf{pk}_{\mathcal{U}}, \mathsf{sk}_{\mathcal{I}}, n))$ is a protocol between a user $\mathcal{U}$ and an issuer $\mathcal{I}$. At the end of this protocol, the user $\mathcal{U}$ obtains an e-token dispenser D, usable n times per time period.
- $SHOW(\mathcal{U}(D, \mathsf{pk}_{\mathcal{I}}, t, n), \mathsf{V}(\mathsf{pk}_{\mathcal{I}}, t, n))$ is a protocol between a user $\mathcal{U}$ and a verifier V. The verifier outputs a token serial number (TSN) S and a transcript τ. The user's output is an updated e-token dispenser D'.
- $IDENTIFY(\mathsf{pk}_{\mathcal{I}}, S, \tau, \tau')$. Given two records (S, τ) and (S, τ') output by honest verifiers in the *SHOW* protocol, where $\tau \neq \tau'$, computes a value $s_{\mathcal{U}}$ that can identify the owner of the dispenser D that generated the TSN S.

There are several properties that an e-token dispenser Σ_{tok} is required to satisfy. Again we will state the simplified variant that applies to our setting with a single verifier V instead of multiple.

Definition 29 (Soundness of Σ_{tok} [75]) Given an honest issuer $\mathcal{I}$, a honest verifier is guaranteed that it will not accept more than n e-tokens from a single e-token dispenser in a single time period. Let $\mathcal{E}$ be a knowledge extractor that executes u *OBTAIN* protocols with all adversarial users and produces functions, $f_1, \cdots, f_u$ with $f_i : \mathbb{T} \times \mathbb{I} \rightarrow \mathbb{S}$, where $\mathbb{I}$ is the index set $[0, \cdots, n-1]$, $\mathbb{T}$ is the domain of the time period identifiers and $\mathbb{S}$ is the domain of TSN's. Running through all $j \in \mathbb{I}$, $f_i(t, j)$ produces all n TSNs for dispenser i at time $t \in \mathbb{T}$.

We require that for every adversary the probability that an honest verifier will accept S as a TSN of a *SHOW* protocol executed in time period t, where $S \neq f_i(t, j), \forall 1 \leq i \leq u$ and $\forall 0 \leq j < n$, is negligible.

Definition 30 (Identification of Σ_{tok} [75]) There exists an efficient function ϕ with the following property.

Suppose the issuer and verifiers $\mathsf{V}_1, \mathsf{V}_2$ are honest. If V_1 outputs (S, τ) and V_2 outputs (S, τ') as the result of *SHOW* protocols, then $\mathit{IDENTIFY}(\mathsf{pk}_{\mathcal{I}}, S, \tau, \tau')$ outputs a value $s_{\mathcal{U}}$, such that $\phi(s_{\mathcal{U}}) = \mathsf{pk}_{\mathcal{U}}$, the violator's public key.

By saying that a user has *reused* an e-token, we mean that there exists $(S, \tau), (S, \tau')$ that are both output by honest verifiers.

Definition 31 (Anonymity of Σ_{tok} [75]) The adversary, acting as the issuer, may run many *OBTAIN* protocols with many honest users. Then this adversary may invoke *SHOW* protocols with users of his choice, up to n times per time period with the same user.

The adversary should not be able to distinguish whether he is indeed interacting with real users or with simulator $\mathcal{S}$ that predents to be real users without knowing anything about them, including which users it is supposed to be at any point in time, and without access to any secret or public key, or the user's e-token dispenser D.

3.7. Encryption Security Definitions

The works [103, 197] have shown that outsourced computation based on both approximate and exact lattice-based FHE schemes cannot in general be reduced to the IND-CPA security, as defined in [163], of these schemes. Instead, the authors introduced the IND-CPA$^{\text{D}}$ security.

Definition 32 (IND − CPA$^{\text{D}}$ Security [197]) A public-key homomorphic (possibly approximate) encryption scheme with plaintext space $\mathcal{M}$ and ciphertext space $\mathcal{CT}$ is defined by $E = (\mathit{GEN}, \mathit{ENC}, \mathit{DEC}, \mathit{Eval})$. $\mathrm{Expr}_b^{indcpa^D}[\mathcal{A}]$ denotes an experiment, parameterized by a bit $b \in \{0, 1\}$ and involving an efficient adversary $\mathcal{A}$ that is given access to the following oracles, sharing a common state $S \in (\mathcal{M} \times \mathcal{M} \times \mathcal{CT})^*$ consisting of a sequence of message-message-ciphertext triplets:

- An encryption oracle $O_{ENC}(m_0, m_1)$ that, given a pair of plaintext messages m_0, m_1, computes $c \leftarrow ENC_{\mathsf{pk}}(m_b)$, extends the state

$$S := [S; (m_0, m_1, c)]$$

  with one more triplet, and returns the ciphertext c to the adversary.

- An evaluation oracle $O_{Eval}(g, J)$ that, given a function $g : \mathcal{M}^k \to \mathcal{M}$ and a sequence of indices $J = (j_1, \ldots, j_k) \in \{1, \ldots, |S|\}^k$, computes the ciphertext $c \leftarrow Eval_{\mathsf{pk}}(g, S[j_1].c, \ldots, S[j_k].c)$, extends the state

$$S := [S; (g(S[j_1].m_0, \ldots, S[j_k].m_0), g(S[j_1].m1, \ldots, S[j_k].m_1), c)]$$

  with one more triplet, and returns the ciphertext c to the adversary, Here and below $|S|$ denotes the number of triplets in the sequence S, and $S[j].m_0$, $S[j].m_1$ and $S[j].c$ denote the three components of the j-th elememt of S.

- A decryption oracle $O_{DEC}(j)$ that, given an index $j \leq |S|$, checks whether $S[j].m_0 = S[j].m_1$, and if so, returns $DEC_{\mathsf{sk}}(S[j].c)$ to the adversary. (If the check fails, a special error symbol $\perp$ is returned.)

The experiment is defined as

$$\begin{aligned}
&\mathrm{Expr}_b^{\mathrm{indcpa}^D}[\mathcal{A}](1^\kappa): \quad (sk, pk, ek) \leftarrow GEN(1^\kappa)\\
&\qquad\qquad\qquad\qquad\quad S := []\\
&\qquad\qquad\qquad\qquad\quad b' \leftarrow \mathcal{A}^{O_{ENC}, O_{Eval}, O_{DEC}}(1^\kappa, \mathsf{pk}, \mathsf{ek})\\
&\qquad\qquad\qquad\qquad\quad \mathrm{return(b')}
\end{aligned}$$

The advantage of adversary $\mathcal{A}$ against the IND − CPA^D security scheme is

$$\begin{aligned}
\mathrm{Adv}_{\mathrm{indcpa}^D}[\mathcal{A}](\kappa) = |&\Pr\{\mathrm{Expr}_0^{\mathrm{indcpa}^D}[\mathcal{A}](1^\kappa) = 1\}\\
&-\Pr\{\mathrm{Expr}_1^{\mathrm{indcpa}^D}[\mathcal{A}](1^\kappa) = 1\}|,
\end{aligned}$$

where the probability is over the randomness of $\mathcal{A}$ and the experiment. The scheme $\mathcal{E}$ is *IND-CPA^D*-secure if for any efficient (probabilistic polynomial time) $\mathcal{A}$, the advantage $\mathrm{Adv}_{\mathrm{indcpa}^D}[\mathcal{A}]$ is negligible in κ.

Although a DRE scheme is also a PKE scheme, its syntax differs and therefore requires a reformulation of the IND-CPA security definition.

Definition 33 (IND-CPA DRE) A DRE scheme is said to be indistinguishable under chosen plaintext attack, i.e. is IND-CPA secure, if for all PPT algorithms $\mathcal{A}$ win the IND-CPA DRE game in Figure 3.1 with probability that is at most negligibly more than one half.

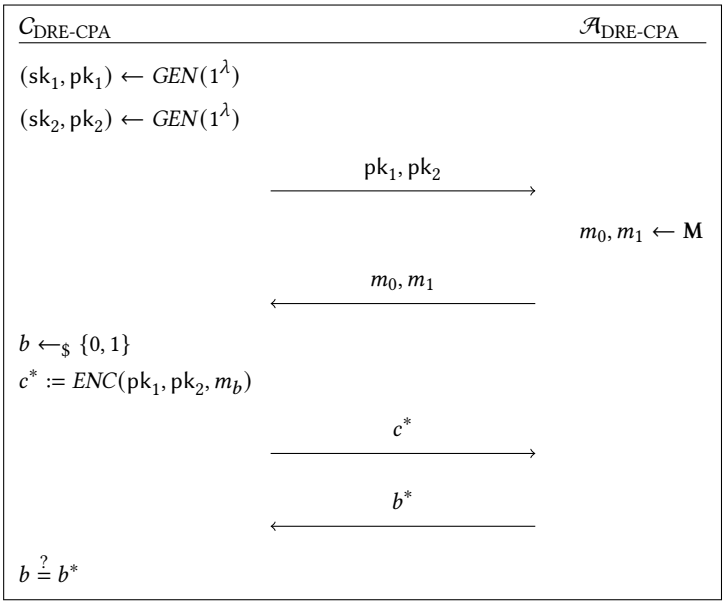

Figure 3.1.: The IND-CPA DRE Game.

In addition to IND-CPA security, soundness is usually required for a DRE scheme to be usable in more complex constructions. This was one of our main motivations in [32], where we showed that this property is surprisingly often ignored in the literature.

Definition 34 (Soundness for DRE [109]) Given an experiment $\mathsf{Exp}^{\text{dre-sound}}_{\mathcal{E},\mathcal{A}}$ as defined in Figure 3.2 for a DRE scheme $\mathcal{E}$ and a PPT algorithm $\mathcal{A}$. The advantage of $\mathcal{A}$ is

$$\mathsf{Adv}^{\text{dre-sound}}_{\mathcal{E},\mathcal{A}}(\lambda) := \mathsf{Pr}[\mathsf{Exp}^{\text{dre-sound}}_{\mathcal{E},\mathcal{A}} = 1].$$

$\mathcal{E}$ satisfies soundness if for any $\mathcal{A}$, we have that $\mathsf{Adv}^{\text{dre-sound}}_{\mathcal{E},\mathcal{A}}$ is negligible in λ.

$\mathsf{Exp}^{\text{dre-sound}}_{\mathcal{E},\mathcal{A}}$

1. $(\mathsf{sk}_S, \mathsf{pk}_S) \leftarrow \mathit{GEN}(1^\lambda); (\mathsf{sk}_R, \mathsf{pk}_R) \leftarrow \mathit{GEN}(1^\lambda)$
2. $c \leftarrow \mathcal{A}(1^\lambda, \mathsf{sk}_S, \mathsf{pk}_S, \mathsf{sk}_R, \mathsf{pk}_R)$
3. Return 1 if $\mathit{DEC}(\mathsf{sk}_R, \mathsf{pk}_S, \mathsf{pk}_R, c) \neq \mathit{DEC}(\mathsf{sk}_S, \mathsf{pk}_S, \mathsf{pk}_R, c)$, else return 0.

Figure 3.2.: The DRE Soundness Game.

Because the IND-CPA security does not address any malleability attacks it is not advisable to use this security definition as-is. In order to adress these attacks the indistinguishability under adaptive chosen ciphertext attack (IND-CCA2) security was introduced. The usual IND-CCA2 security is somewhat different for a DRE scheme than it is defined in Definition 35. However, when a DRE scheme already satisfies the soundness property from Definition 34 then the definition of IND-CCA2$_{\text{DRE}}$ collapses to the standard definition of IND-CCA2.

Definition 35 (IND-CCA2$_{\text{DRE}}$) A DRE scheme is said to be indistinguishable under adaptive chosen ciphertext attack (IND-CCA2$_{\text{DRE}}$ secure), if for all PPT algorithms $\mathcal{A}$ wins the IND-CCA2$_{\text{DRE}}$ game in Figure 3.3 with probability at most $\frac{1}{2} + \mathit{negl}(\lambda)$, i. e.,

$$\mathsf{Adv}^{\text{IND-CCA2}_{\text{DRE}}}_{\text{DRE},\mathcal{A}}(\lambda) := \left| \Pr[\mathsf{Exp}^{\text{IND-CCA2}_{\text{DRE}}}_{\text{DRE},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right| \leq \mathit{negl}(\lambda)\,.$$

Definition 36 (OT-IND) A SKE scheme *SKE* is said to be one-time indistinguishable (OT-IND secure), if for all PPT algorithms $\mathcal{A}$ wins the OT-IND game in Figure 3.4 with probability at most $\frac{1}{2} + \mathit{negl}(\lambda)$, i. e.,

$$\mathsf{Adv}^{\text{OT-IND}}_{\mathit{SKE},\mathcal{A}}(\lambda) := \left| \Pr[\mathsf{Exp}^{\text{OT-IND}}_{\mathit{SKE},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right| \leq \mathit{negl}(\lambda)\,.$$

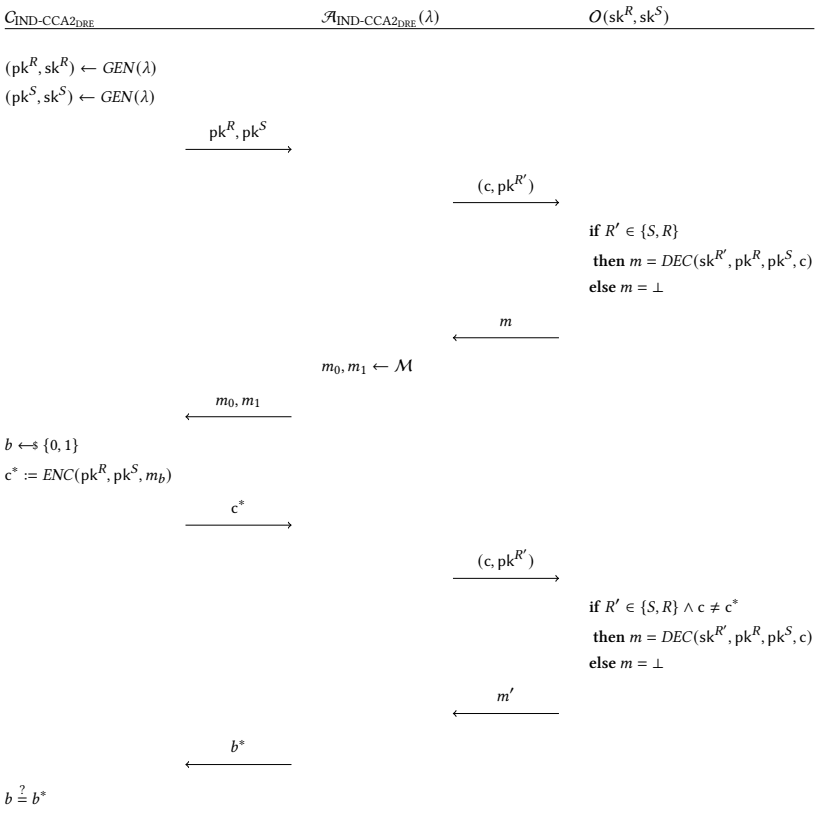

Figure 3.3.: Depiction of the IND-CCA2$_{\text{DRE}}$ game

Definition 37 (Collision Resistance) A family of hash functions $\{\text{H}_k\}_{k \in K}$ is said to be collision resistant (CR) if for all PPT algorithms $\mathcal{A}$ the advantage $\text{Adv}^{\text{CR}}_{\text{H},\mathcal{A}}(\lambda)$ is negligibly small, where

$$\text{Adv}^{\text{CR}}_{\text{H},\mathcal{A}}(\lambda) := \Pr[x \neq x' \text{ and } \text{H}(x) = \text{H}(x') | k \leftarrow\!\!\$\ K, (x, x') \leftarrow \mathcal{A}(\lambda, \text{H}_k)] .$$

We need to look at keyed hash functions, as for every fixed hash function there exists an adversary with a collision hard coded by the pigeonhole principle. Still, in a slight abuse of notation we will speak of a “collision resistant hash

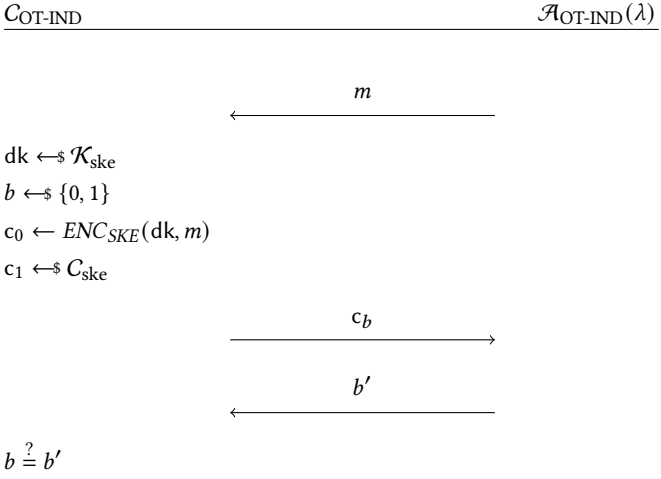

Figure 3.4.: Depiction of the OT-IND game.

function", by which we mean a function sampled uniformly from a collision resistant hash function family.

Definition 38 (OT-SUF) A message authentication code scheme *MAC* is said to be one-time strongly unforgeable (OT-SUF secure), if for all PPT algorithms $\mathcal{A}$ wins the OT-SUF game in Figure 3.5 with at most negligible probability, i. e.,

$$\mathsf{Adv}^{\text{OT-SUF}}_{MAC,\mathcal{A}}(\lambda) := \mathsf{Pr}[\mathsf{Exp}^{\text{OT-SUF}}_{MAC,\mathcal{A}}(\lambda) = 1] \leq negl(\lambda)\,.$$

We use a key derivation function $KDF : \mathcal{K} \to \{0,1\}^n$ with key-space $\mathcal{K}$ in order to generate the *SKE*- and *MAC*-keys from a short seed. We require for a *KDF* function to be IND secure according to Definition 39.

Definition 39 (IND KDF [63]) A key derivation function *KDF* is said to be IND secure if for all PPT algorithms $\mathcal{A}$ the advantage $\mathsf{Adv}^{\text{IND}}_{KDF,\mathcal{A}}(\lambda)$ is negligibly small, where

$$\mathsf{Adv}^{\text{IND}}_{KDF,\mathcal{A}}(\lambda) := |\mathsf{Pr}[\mathcal{A}(\lambda, KDF(\mathsf{k})) = 1] - \mathsf{Pr}[\mathcal{A}(\lambda, r) = 1]|$$

for $\mathsf{k} \leftarrow\!\$\ \mathcal{K}$ and $r \leftarrow\!\$\ \{0,1\}^n$.

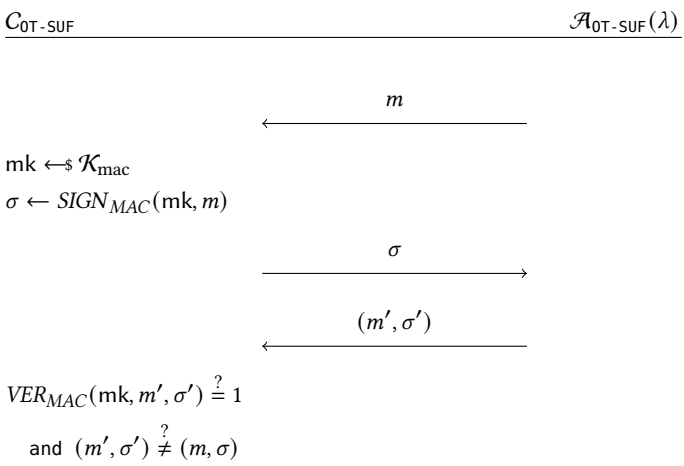

Figure 3.5.: Depiction of the OT-SUF game.

3.8. Real/Ideal Simulation Paradigm

Loosely speaking, the so-called standalone security is a security framework where the security of a protocol is proved by providing a simulated protocol transcript where the simulator constructing the transcript only gets the inputs of a subset of the parties (more detailed, it gets the inputs of the corrupted parties) and showing that the simulated protocol execution is computationally indistinguishable from a real-world protocol execution between the actual parties. Put more formally, we set up two different worlds, called the real world, where the actual protocol is computed by the parties and a subset of the parties is (either semi-honestly or maliciously) controlled by an adversary, and the ideal world, where the same function is computed by an incorruptable Turing machine, called the ideal functionality interacting with a simulator which is simulating the real-world execution by getting the inputs of the adversarially controlled parties and the function's output and generating a protocol transcript (consisting of the simulated messages that the simulated parties exchanged during the protocol execution) that is supposed to be indistinguishable from the transcript of the real-world execution. In order to prove the stated indistinguishability of the transcripts, we have a polynomially bounded interactive Turing machine, called the distinguisher, which gets all parties' inputs, the output and the transcript (which we call the distinguisher's

view) and decides whether the transcript is generated by either the real-world interaction between the computing parties or by the simulator interacting with the functionality. Stated as a definition, we require the following to hold:

Definition 40 Let π be a n-party protocol on inputs $x = (x_1, \ldots, x_n) \in X^n$ and $\mathcal{F}$ a functionality. We say that π standalone securely realizes $\mathcal{F}$ if for all polynomially bounded adversaries $\mathcal{A}$ a simulator $\mathcal{S}$ exists such that:

$$IDEAL_{\mathcal{F},\mathcal{S}}(\lambda, x)_{\lambda\in\mathbb{N},x\in X^n} \overset{c}{\approx} REAL_{\pi,\mathcal{A}}(\lambda, x)_{\lambda\in\mathbb{N},\mathbf{x}\in X^n},$$

where $REAL_{\pi,\mathcal{A}}(\lambda, x)_{\lambda\in\mathbb{N},x\in X^n}$ is $\mathcal{A}$'s view of the real-world protocol execution and $IDEAL_{\mathcal{F},\mathcal{S}}(\lambda, x)_{\lambda\in\mathbb{N},x\in X^n}$ is $\mathcal{S}$'s view in the ideal world when interacting with $\mathcal{F}$.

3.8.1. Universal Composability Framework

The UC framework [77, 79] additionally introduces an interactive Turing machine $\mathcal{Z}$, called the environment, which has to distinguish the real world execution from the ideal world simulation by actively communicating with the adversary during the function computation. Additionally, we define the environment's view with an adversary and a function as the input, the exchanged messages between the corrupted parties and adversary together with the outputs. This restricts the simulation-based proof since rewinding of the adversary is not possible in this case. For a formal statement, we recite the simulation paradigm of the UC framework by Canetti.

Definition 41 ([77]) Let π be a n-party protocol on inputs $x = (x_1, \ldots, x_n) \in X^n$ and $\mathcal{F}$ a functionality. We say that π UC-securely realizes $\mathcal{F}$ if for all polynomially bounded adversaries $\mathcal{A}$ a simulator $\mathcal{S}$ exists such that for every polynomially bounded environment $\mathcal{Z}$:

$$IDEAL_{\mathcal{F},\mathcal{S},\mathcal{Z}}(\lambda, x)_{\lambda\in\mathbb{N},x\in X^n} \overset{c}{\approx} REAL_{\pi,\mathcal{A},\mathcal{Z}}(\lambda, x)_{\lambda\in\mathbb{N},x\in X^n},$$

where $REAL_{\pi,\mathcal{A},\mathcal{Z}}(\lambda, x)_{\lambda\in\mathbb{N},x\in X^n}$ is the environment's view when interacting with $P_1, \ldots, P_n$ and $\mathcal{A}$ and $IDEAL_{\mathcal{F},\mathcal{S},\mathcal{Z}}(\lambda, x)_{\lambda\in\mathbb{N},x\in X^n}$ is $\mathcal{Z}$'s view when interacting with $\mathcal{F}$ and $\mathcal{S}$.

Additionally, the Universal Composition framework has the property that any UC-secure protocol can be securely (concurrently) composed with any other UC-secure protocol. This is the universal composition theorem, which is rephrased here without proof.

Theorem 1 (UC composition theorem, see [77]) Let $\mathcal{F}$ and $\mathcal{G}$ be two ideal functionalities. Further let ρ be a protocol that securely UC-realizes $\mathcal{G}$ and let π be a protocol that securely UC-realizes $\mathcal{F}$. Then the composed protocol ρ^{π} securely UC-realizes $\mathcal{G}$.

We refer the proof to [77].

3.8.2. Ideal Functionalities and Protocols

Rebecca Schwerdt has re-formulated the ideal functionality for a key registration with knowledge (KRK) in Beskorovajnov et al. [45]. In our subsequent work for outsourced computation [42] we have continued to use this functionality.

Another common ideal functionality is $\mathcal{F}_{AUTH}$ shown in fig. 3.7. It defines authenticated communication, meaning that parties can send messages and the receiving party knows the sender ID. This stops an adversary from sending messages in the name of other parties. However, anyone and especially the adversary can read any sent messages. The adversary can block any message and send it in his own name.

Canetti [80] showed how to realize $\mathcal{F}_{AUTH}$ with a signature scheme and a certificate authority (CA). For this, they introduced the ideal functionality $\mathcal{F}_{CERT}$, which provides signatures that are bound to the sender. It is depicted in fig. 3.8. The Protocol $\pi_{AUTH}^{\mathcal{F}_{CERT}}$, which realizes $\mathcal{F}_{AUTH}$ using $\mathcal{F}_{CERT}$ is depicted in fig. 3.9. Intuitively, it uses $\mathcal{F}_{CERT}$ to sign each message when sending and verify the authenticity upon receiving. This puts the main task of verifying the authenticity of the signature onto the protocol realizing $\mathcal{F}_{CERT}$. For this, the authors constructed the protocol $\pi_{CERT}^{\mathcal{F}_{CA}}$. In their work, this protocol uses the ideal functionalities $\mathcal{F}_{SIG}$, which realizes signatures and $\mathcal{F}_{CA}$ for the CA. The former is replaced with an EUF-CMA secure signature scheme, which was proven to realize the needed ideal functionality [80]. The latter is depicted in fig. 3.10 and will remain an ideal functionality throughout this work. It

Functionality $\mathcal{F}_{\text{KRK}}$

Provides: Key registration with knowledge.
Parameters:

- Function $\mathtt{f}_{\text{Key}} : (\mathsf{sk}, \mathsf{pk}) \mapsto \begin{cases} \textit{true}, & \text{well-formed key pair} \\ \textit{false}, & \text{otherwise} \end{cases}$

State:

- Function $p_{\text{Reg}} : \textit{mid} \mapsto (P, \mathsf{sk}, \mathsf{pk})$ of pending registrations.
- Function $p_{\text{Ret}} : \textit{mid} \mapsto (P_i, P_j)$ of pending retrievals.
- Set $\mathcal{R}$ of registered tuples $(P, \mathsf{sk}, \mathsf{pk})$.

Behaviour:

- Upon receiving $(\textit{register}, \mathsf{sid}, \mathsf{sk}, \mathsf{pk})$ from a party P, draw fresh $\textit{mid}$, send $(\textit{register}, \mathsf{sid}, \textit{mid}, P, \mathsf{pk})$ to the adversary $\mathcal{A}$ and append $\textit{mid} \mapsto (P, \mathsf{sk}, \mathsf{pk})$ to p_{Reg}.
- Upon receiving $(\textit{register ok}, \mathsf{sid}, \textit{mid})$ from the adversary $\mathcal{A}$, retrieve $(P, \mathsf{sk}, \mathsf{pk}) := p_{\text{Reg}}(\textit{mid})$, check
  - $\mathtt{f}_{\text{Key}}(\mathsf{sk}, \mathsf{pk}) = \textit{true}$
  - $\nexists\, \mathsf{sk}', \mathsf{pk}' : (P, \mathsf{sk}', \mathsf{pk}') \in \mathcal{R}$
  - $\nexists\, P', \mathsf{sk}' : (P', \mathsf{sk}', \mathsf{pk}) \in \mathcal{R}$

  and append $(P, \mathsf{sk}, \mathsf{pk})$ to $\mathcal{R}$ if all checks were successful.
- Upon receiving $(\textit{retrieve}, \mathsf{sid}, P_i)$ from a party P_j, draw fresh $\textit{mid}$, send $(\textit{retrieve}, \mathsf{sid}, \textit{mid}, P_i, P_j)$ to the adversary $\mathcal{A}$ and append $\textit{mid} \mapsto (P_i, P_j)$ to p_{Ret}.
- Upon receiving $(\textit{retrieve ok}, \mathsf{sid}, \textit{mid})$ from the adversary $\mathcal{A}$, look up $(P_i, P_j) := p_{\text{Ret}}(\textit{mid})$ and $(P_i, \mathsf{sk}_i, \mathsf{pk}_i) \in R$. If no such entry exists in $\mathcal{R}$, set $pk_i := \perp$. Send $(\textit{retrieved}, \mathsf{sid}, \mathsf{pk}_i, P_i)$ to P_j.

Figure 3.6.: Ideal Functionality $\mathcal{F}_{\text{KRK}}$ [45]

Functionality $\mathcal{F}_{AUTH}$

Provides:
Single-receiver single-message single-sender authenticated message transfer with constant message size.
State:

- $m \leftarrow \perp$; $R \leftarrow \perp$; .

Behavior:

- Upon invocation with input (*send*, *sid*, *R*, *m*) from some party S, send backdoor message (*send*, *sid*, *S*, *R*, *m*) to the adversary $\mathcal{A}$.
- Upon receiving (*send ok*, *sid*) from adversary $\mathcal{A}$: If not yet generated output, then output (*sent*, *sid*, *S*, *R*, m) to R
- Ignore all further inputs

Figure 3.7.: The Ideal $\mathcal{F}_{AUTH}$ Functionality

defines the ability to register a public key at a trusted authority and retrieve the public key associated with a party.

Figure 3.11 shows the resulting protocol $\pi_{CERT}^{\mathcal{F}_{CA}}$. The idea is to store the verification key of the sending party in the CA. The receiver can then retrieve the verification key of the sender and thus verify that the signature of the message is not only valid but also belongs to the sender ID.

The protocol π_{SIG} realizing $\mathcal{F}_{SIG}$ is adapted to allow for multiple messages being signed by the same signing key. Without this adaptation, a new signing key would be generated for each message, being highly inefficient. Canetti and Rabin [86] showed that this can be achieved by creating the new protocol $\hat{\rho}$, which simulates multiple instances of π_{SIG}. Instead of receiving a single session identifier (sid), which changes with each call to π_{SIG}, $\hat{\rho}$ receives two session identifiers. The sid, which determines the instantiation of $\hat{\rho}$ and the subsession identifier (ssid), which indicates which instantiation of π_{SIG} should be called. As $\hat{\rho}$ is only instantiated once throughout the whole protocol, the sid is constant and not really used. Therefore, it is omitted throughout

Functionality $\mathcal{F}_{CERT}$

Provides:
Signatures that are bound to the parties.
Behavior:

- Upon receiving (**Sign**, $sid = (S, sid')$, m) from some party S, send (**Sign**, sid, m) to the adversary. Upon receiving (**Signature**, sid, m, σ) from the adversary, verify that no entry $(m, \sigma, 0)$ is recorded. If it is, output an error message to S and halt. Else, output (**Signature**, sid, m, σ) to S and record entry $(m, \sigma, 1)$.
- Upon receiving a value (**Verify**, sid, m, σ) from some Party P, hand (**Verify**, sid, m, σ) to the adversary. Upon receiving (**Verified**, sid, m, ϕ) from the adversary, do:
  1. If $(m, \sigma, 1)$ is recorded then set $f = 1$.
  2. Else, if the signer is not corrupted and no entry $(m, \sigma', 1)$ for any σ' is recorded, then set $f = 0$ and record the entry $(m, \sigma, 0)$.
  3. Else, if there is an entry (m, σ, f') is recorded, then set $f = f'$.
  4. Else, set $f = \phi$ and record the entry (m, σ', ϕ).

  Output (**Verified**, sid, m, f) to P.

Figure 3.8.: The Ideal $\mathcal{F}_{CERT}$ Functionality [80]

this work for better readability and the given sid_{AUTH}, which changes, is the identifier to indicate which instantiation of π_{SIG} should be called. For multiple sid_{AUTH}, $\hat{\rho}$ does not really instantiate new instances of π_{SIG}, as this would lead to many signing keys. Instead, the same signing key can be used if the sid_{AUTH} is signed in addition to the message [86].

Another important fact is that the protocols using $\pi_{AUTH}^{\mathcal{F}_{CERT}}$ have to check, that the same sid_{AUTH} is not used twice to prevent replay attacks.

Canetti [80] described two methods to ensure this. One is to keep a list of used sids and compare with the list when receiving a message. The other option

Protocol $\pi_{AUTH}^{\mathcal{F}_{CERT}}$

Provides:
Single-receiver single-message single-sender authenticated message transfer with constant message size.
Behavior of Party P:

- Upon receiving $(\textbf{Send}, sid, B, m)$, set $sid' = (P, sid)$, $m' = (m, B)$ and send $(\textbf{Sign}, sid', m')$ to $\mathcal{F}_{CERT}$. Upon receiving $(\textbf{Signed}, sid', m', s)$, send (sid, P, m, s) to B.
- Upon receiving (sid, B, m, s), set $sid' = (B, sid)$, $m' = (m, P)$ and send $(\textbf{Verify}, sid', m', s)$ to $\mathcal{F}_{CERT}$ to obtain $(\textbf{Verified}, sid', m', s, f)$. If $f = 1$, then output $(\textbf{Sent}, sid, P, B, m)$ and halt. Else halt with no output.

Figure 3.9.: The Protocol $\pi_{AUTH}^{\mathcal{F}_{CERT}}$ Realizing $\mathcal{F}_{AUTH}$

Functionality $\mathcal{F}_{CA}$

Provides:
Mapping from identities to verification keys.
Behavior:

- Upon receiving $(\textbf{Register}, sid, vk)$ from party P, send $(\textbf{Registered}, sid, vk)$ to the adversary. Upon receiving **ok** from the adversary, and if $sid = P$ and this is the first request from P, then record the pair (P, vk)
- Upon receiving a message $(\textbf{Retrieve}, sid)$ from party P', send $(\textbf{Retrieve}, sid, P')$ to the adversary, and wait for an **ok** from the adversary. Then, if there is a recorded pair (sid, vk) output $(\textbf{Retrieve}, sid, vk)$ to P'. Else output $(\textbf{Retrieve}, sid, \perp)$ to P'.

Figure 3.10.: The Ideal Functionality $\mathcal{F}_{CA}$

Protocol $\pi_{CERT}^{\mathcal{F}_{CA}}$

Provides:
Signatures that are bound to the parties.
Parameters:

- EUF-CMA secure signature scheme $\Sigma = (Gen, Sign, \text{Vfy})$

State of Party P:

- Keypair $(vk, sk) \in (PK, SK)$ of own credentials
- Function $f_{PK} : P \rightarrow PK$ of known public keys

Behavior of Party P:

- Upon receiving $(\textbf{Sign}, sid = (P, sid_{AUTH}), \text{m})$
  1. If (vk, sk) does not exist, draw $(sk, vk) \leftarrow Gen(1^\kappa)$, and send $(\textbf{Register}, P, vk)$ to $\mathcal{F}_{CA}$.
  2. Create $\sigma \leftarrow Sign(sk, (m, sid_{AUTH}))$. Output $(\textbf{Signature}, sid, m, \sigma)$
- Upon receiving $(\textbf{Verify}, sid = (P', sid_{AUTH}), m, \sigma)$ check if $f_{PK}(P')$ exists. If it does not send $(\textbf{Retrieve}, P')$ to $\mathcal{F}_{CA}$ to obtain response $(\textbf{Retrieve}, P', vk)$. If $vk = \perp$ output $(\textbf{Verified}, sid, m, 0)$. Else set $f_{PK}(P') = vk$ and output $(\textbf{Verified}, sid, m, \sigma, \text{Vfy}(pk, (m, sid_{AUTH}), \sigma))$.

Figure 3.11.: The Protocol $\pi_{CERT}^{\mathcal{F}_{CA}}$ Realizing $\mathcal{F}_{CERT}$ using an EUF-CMA Secure Signature Scheme

is to negotiate the usable sids at the beginning of the protocol. Concretely, the parties draw nonces at beginning and share them with each other. These shared nonces are then used one after the other. This reduces the amount of stored data but introduces a chance of error [80]. Throughout this work, we will assume there is a mechanism in place that checks for duplicates and rejects them, when we write *fresh* sid_{AUTH}.

4. Internal Attacker

A company is a complex network of information systems and people. The state-of-the-art approach to securing such enterprise systems has evolved from the need to protect internal processes against externally intruding adversaries. Threat assessments [73, 126, 139, 140, 281, 288] for 2024–2025 still highlight ransomware and other malware as leading attack vectors against internal networks. This vector is one of the oldest, and as a result, many technologies already exist to help companies defend against it.

Unfortunately, these measures become ineffective once the perimeter is breached and an adversary can passively eavesdrop on or exfiltrate data or, in the worst case, manipulate it. We refer to such an adversary as "internal" and do not distinguish between a rogue employee, an external intruder who has breached the perimeter, or an external actor bribing or extorting an employee. From an information-security standpoint, the protective measures against all of these are the same. Moreover, once an attack occurs, it is generally unrealistic to assume that one can reliably tell them apart. Any discernibility may be achieved only after the fact, by which time the attack has already occurred. This thesis on the other hand focuses on pre-emptive measures.

ENISA reports note that attacks by internal adversaries appear in smaller numbers than other threats, though likely with substantial underreporting. For several years, the European Union Agency for Cybersecurity (ENISA) has repeatedly stated in its annual threat reports that "[…] organisations remain reluctant in sharing details of these incidents." ([138, 139]). In the 2025 report [140], the issue is scarcely mentioned. Nevertheless, a figure of 0.8% of incidents is reported.

One important note is in order. Given the apparently low figures, one might mistakenly conclude that investments in insider-threat protections are not economically justified. This is misguided for at least two reasons. First, substantial underreporting likely depresses the figures. One plausible driver

is reputational risk: admitting that a rogue insider employee facilitated data exfiltration may be seen as more damaging than attributing the incident to human error—such as an employee clicking a phishing email, which can occur even in well-trained organizations. Second, as noted earlier, attribution is hard: even after the fact, it is often impossible to determine whether a breach stemmed from a rogue, bribed, or extorted employee, or from a vulnerability or misconfiguration that let an external attacker breach the perimeter and possibly impersonate an employee. Moreover, many compromises begin with phishing, leaving the internal network only a single click away. It, thus, remains unclear what fraction of such incidents should be classified as insider attacks. As a result, current statistics likely understate both the prevalence of insider threats and the total damage they cause.

The current solution space for these kind of threats usually covers many different internal surveilling measures that are privacy-invasive to the company's employees, see e.g. [110]. Other solutions revolve around the informal concept of *Zero Trust*, which is a security "paradigm" that assumes no user or device, whether inside or outside the network, can be trusted by default, requiring continuous verification and strict access controls for all resources.

However, a provably secure solution that avoids privacy-invasive measures remains elusive in industry. This is surprising from the standpoint of provable security and cryptographic modeling, where even the simplest threat models typically allow for any party to be corrupted at some point, aside from notable cases that rely on trusted third parties (e.g., a public-key infrastructure (PKI) with a root certification authority (CA)). Protecting against these kinds of threats is hardly a new motivation in the cryptographic literature: work from the 2000s (e.g., [156, 165, 166]) already discussed attacks by internal users, and the idea goes back even further. As early as 1978, Rivest, Adleman, and Dertouzos proposed early approaches to such threats [250]. Today is no different. Consequently, we now have nearly five decades of cryptographic research on such protections.

From the author's perspective, it appears that when a product is primarily based on a cryptographic protocol, protections of some degrees against internal threats within the participating parties are usually in place. The most notable example is the Signal[1] instant messaging application, which uses an ingenious double-ratcheting protocol to quickly recover a client from a

[1] `https://signal.org/de/`

potential breach, as analyzed in [52, 117]. Other end-to-end-encrypted file-transfer tools also mitigate, at a minimum, the attack vector of a compromised back-end server, for example, croc[2], noisytransfer[3], rustytransfer[4] or magic wormhole[5]. However, adapting cryptographic methods to products that extend beyond simple applications is a non-trivial task. Currently, multi-party computation (MPC) appears to be the most versatile tool for achieving such protections, but, as mentioned in the introduction, it suffers from significant overhead due to an overly restrictive adversary model. While an overly restrictive adversary model may seem advantageous, in many cases it goes far beyond what is actually critical within a company's network. This effectively results in unnecessarily large overhead, which hinders MPC solutions from being widely adopted.

From the author's perspective, modeling an internal adversary should begin with the simplest cases. Protective measures derived from these cases will naturally generalize to more complex scenarios and can often be used as is in other settings. The most basic, though often uninteresting from a cryptographic standpoint, is an honest employee who unknowingly or mistakenly performs actions that lead to information leakage. This may occur due to misconfiguration or simply not being aware of data protection measures in place, and thus unknowingly bypassing them.

Separation of Duties One may wonder how cryptographic measures can address the benign misbehaviour. We will demonstrate that the concept of *separation of duties*, when incorporated into a cryptographic protocol—such as specific variants of MPC protocols or a contact tracing protocol—provides this kind of protection as a byproduct. This concept revolves around distributing distinct processing steps of an application across different systems, operated by separate individuals or (sub-)organizations. This inherently reduces risk of complete information disclosure against individual failures by adhering to the principle of not keeping all eggs in one basket.

[2] https://github.com/schollz/croc
[3] https://github.com/collapsinghierarchy/noisytransfercli
[4] https://github.com/collapsinghierarchy/rustytransfer
[5] https://github.com/magic-wormhole/magic-wormhole

Passive Confidentiality (i.e. "Honest-but-Curious" Security) To strengthen defenses against internal adversaries, one may now consider an employee who knowingly breaches security measures with malicious intent, without actively manipulating any systems. Relying solely on the separation of duties concept will not be sufficient, as such an employee can still leak information—albeit only the information to which they have access. To ensure confidentiality, encryption methods are required, but this task becomes non-trivial due to the challenges of key management. Simply encrypting data-in-use shifts the internal adversary's focus from breaching the information itself to breaching the decryption key. This is because, by the nature of data-in-use, classically encrypted data must eventually be decrypted to be further processed, and therefore, the decryption key must be accessible and used frequently. This is also why protecting data-at-rest against internal adversaries is relatively easier—the decryption key can be kept out of the adversary's reach, as it is used infrequently. However, we can adapt the separation of duties concept to the key management of data-in-use encryption. This would result in one party holding the decryption key and another party processing the encrypted data, i.e., the data-in-use. Unfortunately, this approach is not sufficient, as it still requires the encrypted data to be processable. Ideally, data-in-use would remain constantly encrypted during processing, with only the final result being decrypted and forwarded to the next processing step, which at that stage might not even require further protection. The challenge of processing encrypted data was conceptually addressed by the seminal work of [250], which introduced homomorphic encryption. Since then, various encryption schemes have been developed, allowing different degrees of processing on encrypted data. To name a few:

- Fully Homomorphic Encryption (FHE), introduced by [159], allows arbitrary computations on encrypted data but involves significant computational overhead.
- Somewhat Homomorphic Encryption (SHE) [291] permits limited operations on encrypted data and represents a practical trade-off between functionality and efficiency.
- Deterministic Encryption [28] enables exact search capabilities on encrypted data by generating the same ciphertext for the same plaintext across different encryptions, which is useful for efficient indexing, albeit with reduced semantic security.

- Searchable Encryption [271] facilitates searching encrypted data without revealing the plaintext, a key solution for secure outsourced storage.
- Functional Encryption [57] allows specific computations on encrypted data based on predefined access rights, ensuring selective data access.
- Re-randomizable Encryption [83] adds an additional layer of security by allowing ciphertexts to be randomized without changing the underlying plaintext.

Digital Operations Resilience Act (DORA [146]) Currently, forthcoming European regulations in the financial sector explicitly suggest a need for cryptographic measures to protect so-called data-in-use. See, for example, Article 6 *Encryption and Cryptographic Controls*

"2. The policy on encryption and cryptographic controls shall be designed based on the results of approved data classification and ICT risk assessments, and shall include the following elements: [...] (b) rules for the encryption of data-in-use, where necessary.[...]"[146]

As already mentioned, traditional encryption methods—such as those that satisfy strong security standards like chosen-ciphertext attack (CCA) —are not suitable for encrypting data-in-use, as they render the data unusable for subsequent processing. Thus, it appears that these regulations may be encouraging the use of advanced techniques like homomorphic encryption (HE) or multi-party computation (MPC) for encrypting data-in-use, thereby strengthening financial systems against insider threats. The final regulations are expected to take effect in January 2025.

Pseudonymization Alternative means of protecting personal data-in-use are pseudonymizations. The current regulation and best-practices landscape provides plenty of resources (c.f. [137, 141, 142, 143, 145]). Pseudonymization is a data protection technique that replaces identifiable information within a dataset with pseudonyms and de-pseudonymizes them when necessary for processing. This process reduces the risk of re-identification while allowing the data to remain useful for analysis and processing. According to the GDPR Article 4(5), pseudonymization is defined as:

"The processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organizational measures to ensure that the personal data are not attributed to an identified or identifiable natural person."[144, GDPR Article 4(5)]

There are also other definitions available from ISO/IEC 20889 [174] and ISO 25237 [173] standards. However, all definitions share the essence of protecting personal data while keeping it usable, which effectively means that they are data-in-use protections against internal adversaries as well. To emphasize this observation we refer to paragraph 42 of the guidelines on pseudonymization from the European Data Protection Board (EDPB):

"If a controller or processor wants to use pseudonymisation to reduce confidentiality risks from some or all unauthorised third parties, they will include those third parties in the pseudonymisation domain and assess the means they are reasonably likely to use for attribution. Relevant third parties include not only cyber-crime actors, **but also employees or maintenance service providers acting in their own interests rather than on instructions from the controller**. Taking due account of contextual elements and the circumstances at hand, it is recommended to consider both actions in good faith, and those executed with criminal intent. "[137, Paragraph 42]

Although the regulatory definitions are valuable, they are not formal in the sense of provable security; they require thorough risk analysis and familiarity with state-of-the-art pseudonymization techniques for effective implementation, which can be categorized in the following way:

- **Pseudonymization**: Involves replacing identifiable data with pseudonyms using symmetric cryptographic techniques.
- **Delegated Pseudonymization**: Involves outsourcing delegation of the pseudonymization process to someone else. Typically, the data subjects themselves or a third party holding the data. These methods involve the use asymmetric cryptographic techniques. Usually, these techniques are part of the Privacy Enhancing Technology (PET) or the Privacy Enhancing Cryptography (PEC) [227].

Thus, the previously mentioned encryption schemes, such as FHE, SHE, deterministic encryption, functional encryption and so on, are considered pseudonymization techniques in the industrial state of the art. Therefore, we arrive at the same cryptographic foundation when considering pseudonymization as it is the case with the data-in-use protections in DORA [146].

A Note on Privacy and Security At the start of this chapter, we argued that protections against internal attackers are security measures, whereas pseudonymization methods are usually portrayed as privacy measures. We contend, however, that pseudonymization techniques are precisely the same mechanisms used to protect data-in-use against internal threats. This overlap exists because measures that safeguard individuals' personal data also serve as security controls for the organization holding that data, defending it from both external and internal adversaries. Consequently, if data processing is not secure against internal attackers, privacy cannot be achieved. In other words, privacy for data subjects are security measures for the data controller.

Outlook on Privacy & Security in the Real World Despite significant advances in regulatory frameworks and best-practice recommendations, privacy-enhancing methods have yet to achieve broad industrial adoption. Waldman [295] offers several plausible explanations for why privacy—and, consequently, the techniques designed to protect it—remain niche concerns (if acknowledged at all) in industry. As laypersons in the field of privacy law, we are not positioned to offer a definitive assessment of these legal arguments, nor to propose specific amendments to the General Data Protection Regulation to address this lack of methodological uptake. Instead, the primary goal of this chapter is to examine the technology stack of privacy-enhancing techniques and to evaluate their efficiency and effectiveness with regard to privacy definitions from the academic literature. And thus, showing feasibility at least on the technological level.

4.1. The Non-Collusion Assumption

To analyze constructions adhering to the separation of duties principle using methods from provable security, an assumption must be made. In our case, we adopt the non-collusion assumption between two or more separated servers.

This weakens the previously stated adversary model slightly, as we exclude internal adversaries from colluding across separated components.

In the following we would like to discuss why this relaxation is justified and how to harden systems following the separation of duties concept without violating the non-collusion assumption. Deploying solutions that do not address the hardening aspect of the non-collusion assumption leads to contemporary examples of "privacy washing", as highlighted by the recent Dagstuhl Seminar [1]. One prominent, contemporary example of this assumption being insufficiently realized, where the risk of a violation is reasonably high, is Worldcoin's MPC concept [269], in which two shares of a secret-sharing scheme are held within the same organization.

Examples of non-collusion assumptions between parties are plentiful, with the most prevalent being the non-collusion assumption between a server and its clients, as we describe in a small exemplary study on outsourced PSI (OPSI) in Appendix A.1.

Examples of works using explicitly the non-collusion assumption between two servers is the Information-Theoretic Private Information Retrieval (IT-PIR) by [108], the contact tracing protocol ConTra Corona by [48] or the multi-cloud storage by [273]. We would like to emphasize it again, such assumptions create a potential single point of failure (SoF), making it essential to assess the risk and either accept it (if, for example, it is justified to be sufficiently small) or implement preventive measures to either eliminate the risk of exploitation or significantly increase the difficulty of exploiting it at scale.

In the following, we will not consider an outsider adversary that might penetrate the networks of all operators at once and thereby violate the non-collusion assumption. This is because such an adversary is already addressed by other means, and accounting for this type of adversary would overcomplicate the analysis. In addition to leaking information, the risk of an external adversary manipulating data is reasonably high, and additionally strenghtened measures must be considered if such an adversary gains access to internal systems. Therefore, we focus on attacks that violate the non-collusion assumption from the perspective of an insider attacker corrupting different parties.

The first category refers to attacks involving organizations that operate the separated servers and knowingly collude with each other—often under the instruction of a C-level executive—to share data. The motivation for this level

of corruption can vary, as described in detail for the public sector by [175]. From our perspective, the most important factor driving such collusion is the potential cost to the organizations if the collusion is uncovered, which almost directly translates to direct cost to the C-level executives. Therefore, we propose examining the potential minimal costs that organizations operating the separated servers may incur if their collusion is uncovered, in order to assess the risk of violating the non-collusion assumption due to organizational C-level corruption. In order to increase the chances of uncovering such collusions, and thus violations, the organisations must employ proper whistleblowing mechanisms.

If we exclude C-level corruption within an organization, the only remaining threat to the non-collusion assumption comes from individual insiders with access to data processed by the servers. Their motivations can vary and may include misuse of sensitive information by personnel within the operating organization. Although the honest-but-curious adversary model accounts for such attackers, its guarantees break down when these adversaries collude. In the specific use cases considered, we find it difficult to identify realistic scenarios in which two individuals working for different operators—without any prior acquaintance—would collude to misuse data for personal gain. Even if such individuals already knew each other before the protocol's execution, it is still hard to imagine a realistic scenario in which they would deliberately take jobs at different organizations with the intent of later colluding to misuse sensitive data. We therefore treat these as corner cases: not impossible, but difficult to assess reliably. Moreover, we find that reasoning about and modeling this kind of collusion cannot be done in general. It depends strongly on the specific organizations and their employees operating the systems.

We propose to first examine the prerequisites of such an attack and then implement countermeasures that can thwart them. A key requirement is the possibility of a passive attack on the individual level prior to the collusion taking place. While our protocols protect against information leakage when attacks happen on the individual level, additional countermeasures—some of which may be organizational in nature—could be implemented to further prevent this type of attack. It is important to distinguish between protecting against information leakage during an attack and preventing or tracing the attack itself. Preventive measures typically involve authentication or authorization mechanisms, while tracing measures require secure logging and effective monitoring of those logs. However, when these measures become surveillance-oriented, organisations may encounter the same issue

exhibited by current "Zero Trust" solutions: they can be highly privacy-invasive towards employees. It is therefore a trade-off organisations must make when accounting for protections against individual collusion across different organisations.

Finally these measures are highly dependent on the specific circumstances of the operators. We hope that future research will explore such case studies to develop practical guidelines for further strengthening protection in the context of non-collusion assumptions and thus avoiding the "Privacy Washing" issues.

4.2. A Formal Treatment of Homomorphic Encryption Based Outsourced Computation in the UC Framework

This section is based on a joint work with Sarai Eilebrecht, Yufan Jiang, and Prof. Dr. Jörn Müller-Quade [42]. At first this work was submitted to PKC2025, where it was declined. After a major revision by the authors, this work was submitted to ACNS 2026 and was declined again. Currrently, this work is submitted to the Volume 3 Issue 1 of the IACR CiC journal under the title "A UC Secure Transformation for Multi Client, One Shot, Reusable HE Based Outsourced Computation".

Privacy-preserving computation based on homomorphic encryption (HE) and secure function evaluation (SFE) has matured since the early work of Rivest et al. [250], and multi-party computation (MPC) has enabled applications ranging from privacy-preserving analytics [275] to secure auctions [54] and private machine learning [191]. Yet, deployment in real-world web-service settings remains limited. Our hypothesis is that prevailing outsourcing models do not match the constraints of operational realities. Therefore, in this work, we focus on outsourced computation for web services with the following requirements:

- Dynamic and initially (pre-computation) unknown set of clients.
- One-shot clients: a client contributes once and can remain offline thereafter.

- Minimal input-providing client work: client work should be roughly the cost of running the encryption algorithm.
- Reusable setup: Input is provided once and can be reused across multiple computations, without requiring the input-providing client to remain available.
- Decoupled compute and output: computation can complete while the initiator is offline, and results are retrievable later.

HE-based outsourced MPC started with multi-key HE [200, 202] and continued with MPHE/rdMPHE-style systems [17, 89, 105, 194, 219, 220, 221, 223]. These approaches typically require for the output recovery a collaborative decryption and/or key switching by multiple key holders, so a threshold of input parties must return online. This conflicts with the web-service setting we target, where input contributors that do not require any output should be able to go offline permanently (in the worst case, everyone except a single designated receiver) after uploading, and only designated receivers later retrieve results.

A second line of work, HE-based outsourcing under server–server non-collusion, relaxes the requirement that clients come back online by moving decryption assistance to a second, non-colluding server [69, 241, 282, 296, 297]. This is the closest conceptual starting point for our setting, since it enables one-shot input clients and on-demand output delivery without requiring other input parties to return online; however, existing proposals rely on non-standard HE constructions. Outsourcing has also been studied extensively via server-aided garbled circuits and via secret-sharing/SPDZ-style frameworks, but these approaches either hinge on trust restrictions that we deem implausible in open web services and/or require protocol-specific input/output interaction rather than an encrypt-only upload interface (see Section 4.2.1) [53, 121, 122, 123, 176, 178, 179, 217, 304].

By contrast, a server–server non-collusion assumption posits two dedicated providers that do not collude with each other. We view this as more realistic and controllable in practice, for example through legal and organizational separation of duties between a data holder and a data processor. This assumption has appeared in various settings, for example in PIR and multi-cloud systems [108, 273], in domain-specific services [48, 309], and in outsourcing frameworks that build on the non-collusion model of [178], such as [90, 92].

Nevertheless, HE-based outsourced computation remains worth studying in its own right because it builds upon a long line of optimization research on standard schemes and because there is a large body of HE-based protocols and applications that would directly benefit from outsourcing. In many realistic deployments, inputs are already produced or stored under HE encryption (or must remain encrypted end-to-end), and converting them into secret shares would introduce additional interaction, additional trust and setup assumptions, and protocol-specific client logic that we seek to avoid. Finally, HE enables outsourcing in settings where the set of computing servers changes over time, since encryption does not require distributing input-dependent state to a fixed server set in advance. Taken together, HE-based outsourced computation is a worthwhile primitive for our target interface, but a naïve single-server instantiation is insecure once decryption is delegated: placing decryption capability under a single secret key at the server creates a single point of failure. In standard single-key HE, the party that holds the secret key is both the input owner (who encrypts) and the decryptor (who returns outputs), so avoiding non-collusion style trust restrictions typically forces clients to participate as decryptors even if they do not wish to receive an output.

If input-providing clients are to act only as contributors and may then go offline, then some other party must assume the role of the decryptor. This decryptor cannot simply be the calculating server, since that would amount to giving the calculating server unilateral decryption capability. Instead, one is led to introduce a separate decryption service which can decrypt designated results without any single entity learning the full secret key. Otherwise, one merely trades one trust restriction for another.

This motivates a step-wise approach. We begin with the simplest setting consisting of one computing server, one decryption server, and a single-key HE scheme, and we assume that these two servers do not collude (or, in the terminology of [178], do not cooperate). Subsequent extensions can then replace this trust restriction by threshold-style decryption mechanisms in a provable way. To capture the full range of outsourcing architectures, our model allows the Decryptor (and thus the secret key) to be part of the initiating party (that actually expects to receive the output), matching the prevailing single-key outsourced-computation pattern. Thus, our approach generalizes single-key HE outsourcing and subsumes multiple deployment options.

A Note on Concurrent Work: Notable concurrent work by Scafuro and Verber proposes a server aided MPC framework and includes an FHE based variant with distributed key generation and distributed decryption among servers [261]. Their approach, however, relies on clients contributing inputs via verifiable secret sharing and signatures, rather than via a ciphertext only upload interface. This share-native input layer also makes approximate-arithmetic HE instantiations, such as CKKS, less straightforward, since VSS reconstruction is inherently exact algebraic and bit-level operation. Separately, their work does not discuss noise distribution attacks and related countermeasures, such as those captured by IND-CPA$^{\mathrm{D}}$ style concerns, whereas we explicitly account for these issues and discuss mitigation directions in our setting. Last but not least, the FHE-based outsourced protocol does not account for any client-side postprocessing by the output recipient, which effectively means that protocols requiring such postprocessing (e.g., the zero-test protocol in Section 4.2.11) are not instantiable within their framework.

In addition, we view it as an important direction for future work to strengthen our model by replacing the current non collusion requirement with a threshold style assumption via a robust distributed decryption procedure. Concretely, we plan to incorporate an idealized distributed decryption functionality along the lines of [315] (which is another concurrent work), and to realize it with an appropriate protocol among the servers, so that correctness and privacy can be maintained as long as fewer than a threshold number of servers collude.

Our approach. We study outsourced computation in a single-key setting with two non-colluding parties: a Calculator and a Decryptor. At a high level, functions of the form $f(y, \mathbf{x}) = f_2\big(y, f_1(\mathbf{x})\big)$ are realized by evaluating f_1 homomorphically over the encrypted inputs at the Calculator and then running a two-round, one-sided, sender-private 2-party SFE protocol f_2 between the Calculator (sender) and the Initiator (receiver). The Decryptor holds the secret key, decrypts a masked interim value, and stores it until the Initiator retrieves it. Clients produce and upload only ciphertexts. Initiators add a fresh one-time mask per round. The setup is reusable within a key epoch, the evaluation function is fixed within that epoch, and computation and output retrieval are decoupled.

Overall out goal is to develop a framework that yields UC secure outsourced computation protocols from HE without requiring the protocol designer to prove from scratch, each time, that a concrete protocol UC realizes a target

ideal functionality. In our approach, the protocol designer only needs to prove that the underlying HE based protocol realizes a simple two party functionality in the standalone model, and our transformation then directly yields a UC secure outsourced computation protocol.

Contributions.

- We formulate an ideal functionality for non-interactive outsourced computation with dynamic, one-shot clients and reusable setup, and we give a construction from black-box HE in a two non-colluding servers model.
- We prove that the protocol UC-realizes the functionality in the $\mathcal{F}_{\text{KRK}}$-hybrid under semi-honest, static corruption, assuming no server–server collusion. The construction is modelled with sender privacy (the sender contributes inputs only as ciphertexts).[6] Additionally, we discuss an adaptation of the protocol (based on NIZKs) with security against active/malicious adversaries.
- We demonstrate practicality with a simple homomorphic private equality algorithm.

Organization. In Section 4.2.2 we present the functionality and the generic protocol, together with the security theorem. In Section 4.2.11 we discuss instantiation choices and engineering aspects, including homomorphic equality filtering and we report on our evaluation.

4.2.1. Related Work

Due to our threat model considering non-collusion between at least "honest-but-curios" parties, we exclude works on outsourced computation like [305] that rely on fully trusted parties from the following analysis. Frameworks such as [263] that achieve even stronger security notions such as verifiability are also out of scope.

[6] We are confident that the sender privacy is not a strict requirement and future extensions will incorporate a variant, where the calculator can store inputs in plaintext.

HE-based Outsourced MPC. López-Alt et al. [200] proposed a construction of outsourced secure computation using a key-homomorphic encryption scheme. This was the first work in which clients' online work on inputs is essentially just encryption, with additional (typically lightweight) interaction deferred to setup and the output phase. This construction was refined in [202], where they defined a new type of homomorphic encryption scheme, called multi-key HE (MKHE). Building on this, a new line of research was established, e. g. [17, 221, 223]. At a high level, MKHE enables evaluation over ciphertexts encrypted under multiple independent public keys: each input party generates its own key pair, encrypts its input under its own public key, and outsources the ciphertexts to the computing parties, who homomorphically evaluate to obtain an encrypted result under a combined key. Since the decryption key material is distributed across the input parties, the output is typically recovered via a joint decryption and/or a (collective) key-switch/re-encryption step so that the final ciphertext is decryptable under the receiver's key. A significant drawback of standard MKHE is that both ciphertext size and homomorphic evaluation complexity typically grow with the number of input parties, whereas MPHE frameworks utilize a collective public key to ensure that ciphertexts and computations remain "single-key sized" regardless of the group size. In parallel, several works introduced multi-party HE (MPHE), where parties first jointly set up a collective public key and hold the corresponding secret key in shared form (so ciphertexts and evaluation can remain essentially "single-key sized"), at the cost of a more rigid user structure tied to a fixed group of parties (limited dynamicity/ciphertext reusability). Such MPHE-based frameworks and optimizations include, e.g., [194, 219, 220].

Cheon et al. [105] formalize the MKHE/MPHE distinction and propose reusable dynamic MPHE (rdMPHE), which relaxes MPHE by allowing a small precomputation interaction among key owners while keeping evaluation and ciphertext sizes independent of the number of parties, and additionally supports both dynamic enrollment of new parties and ciphertext reusability across different contexts. A recent UC-framework analysis of these approaches is also available [89]. However, rdMPHE still requires a collaborative decryption procedure (i.e., partial decryptions from multiple key holders), and does not natively support a setting where all input parties go offline permanently and only a single receiver later comes online to obtain the result. It is an interesting direction for future work to extend our outsourced computation framework to work with a distributed decryptor functionality that may be realized with

the help of rdMHE in order to weaken the non-collusion assumption between a single calculator and decryptor to a threshold corruption model of multiple decrypting parties (that are decoupled from the input parties in order to remove the burden on input contributing parties). A recent notable work in this direction is [315], which defines distributed decryption via an ideal functionality. In future work, we plan to extend our framework with this functionality, enabling us to outsource HE-based computation using the schemes discussed in this paragraph while meeting all of our deployment requirements and replacing the non-collusion assumption with threshold-based security.

HE-based Outsourced MPC with Non-Collusion. The requirement that (all or a threshold of) input parties must come online again for a collaborative decryption step is relaxed in [241]. There, the single-server setting is extended to a two-server setting: clients still encrypt their inputs under self-generated key pairs, but they no longer need to participate in a joint decryption procedure. This line of work is the closest to ours. Their requirements largely match those stated in our introduction, except that they (i) allow each client to use its own key pair and (ii) rely on (seemingly) stronger non-collusion assumptions, namely that the two servers do not collude.[7]In [241], Peter et al. give a two-server construction based on the BCP cryptosystem [69]. BCP is additively homomorphic and supports a double-trapdoor decryption mechanism, which induces a master secret key tied to the public parameters. Building on [241], Wang et al. [296, 297] study computation over cloud data encrypted under multiple independent keys and propose a two-cloud setting with two non-colluding servers. The main difference is that [241] relies on BCP's master trapdoor: the second server can decrypt (blinded) ciphertexts and re-encrypt, whereas Wang et al.'s two-cloud approach is PRE-based and aims to avoid explicit server-side decryption while focusing primarily on polynomial/arithmetic computations. Subsequently, [282] proposes a two-server solution in a similar spirit, improving the design by avoiding explicit server-side decryption via proxy re-encryption with a different HE cryptosystem.

The main drawback of these approaches with respect to our motivation is that, although they satisfy our functional requirements for secure outsourced

[7] Our initial analysis suggests that the construction may still provide passive security against client–server collusion, but it breaks down if the two servers collude.

computation, they rely on non-standard HE cryptosystems that do not benefit from the extensive optimization efforts and security scrutiny available for widely deployed schemes such as BGV, BFV, CKKS, TFHE, and related variants.

Server-Aided MPC Non-collusion assumptions also appear in outsourcing frameworks based on garbled circuits. Kamara et al. [178, 179, 216, 217, 304] study server-aided outsourcing of secure computation, where an untrusted cloud assists in executing a secure computation (e.g., via garbled circuits), under an explicit restriction on which parties may collude, and formalize non-colluding adversaries. Most notably, these frameworks typically rely on a server–client non-collusion assumption, i.e., the assisting cloud/server is assumed not to collude with the input-providing parties. We deem this more dangerous in a web-service scenario than a server–server non-collusion assumption: in the latter, the main risk stems from organization-level collusion that can (at least in principle) be discouraged by contractual separation, audits, legal liability, and whistleblowing mechanisms, whereas server–client non-collusion can be violated by a single rogue administrator or by compromise of the service operator. Follow-up works such as [90, 91, 92, 93] refine the server-aided GC model and often shift the trust restriction to a server–server non-collusion assumption (e.g., that the application server and the assisting cloud do not collude). Further works study alternative points in the design space (e.g., efficiency and different threat models) within the server-aided setting [53]. However, many of these outsourcing protocols are designed primarily for the two-party case. Related GC-based MPC work lifts the two-party restriction (e.g., [299]), but these protocols still require non-trivial participation by all parties during preprocessing/input handling and output opening.

While these approaches reduce online work for a constrained party, clients still must participate in a protocol-specific input-delivery and output-retrieval procedure (e.g., OT/garbled-label interaction and later output decoding/-consistency checks in outsourced-GC systems), rather than performing an encrypt-only, one-shot upload and remaining offline until result retrieval (except, possibly, for a designated output recipient). Hence they do not meet our “encrypt-only and go offline” client requirement.

Linear Secret Sharing based Outsourced MPC The "two non-colluding computation servers" paradigm has also been explored outside the GC setting, in arithmetic secret-sharing–based outsourced MPC, already in the framework of Veugen et al. [294]. Their construction supports many lightweight clients via input-independent preprocessing and two non-colluding computation servers, and provides strong integrity (even against a malicious server) using SPDZ-style authenticated secret sharing with efficient finite-field MAC/tag checks.

Concurrently, Jakobsen et al. [176] provided a generic outsourcing layer in which a dynamic (and initially unknown) set of lightweight clients can contribute inputs with minimal local work and later obtain outputs, while the heavy computation is carried out by a fixed set of worker servers running an underlying MPC protocol. In contrast to the two-server non-collusion model, their security is based on a threshold-style assumption that at least one worker server remains honest. After submitting their shares, input clients need not remain online during computation; in particular, only those clients that wish to receive an output must come online to retrieve it and remove their mask. The authors do not claim share reuse across multiple computations, and we do not rely on such a claim here. Nevertheless, our initial analysis suggests that keeping long-lived input shares across computations may remain secure provided that fresh masking randomness is used for each computation (as in our protocol), though a formal treatment is left for future work.

Likewise, the SPDZ line of protocols [121, 122, 123] (and SPDZ-based outsourced-I/O variants) supports reusable setup via offline preprocessing and enables computations to proceed independently of whether input-providing clients remain online, with results delivered on demand. At the same time, the efficiency and malicious-security mechanisms that make these approaches attractive, e.g., authentication / MAC-style consistency checks and related protections—are intrinsically tied to secret-shared representations of values over a finite field (or ring). Consequently, these techniques "fit best" when the worker world computes on authenticated shares (as in SPDZ-style arithmetic MPC), whereas ciphertext-native instantiations (e.g., multiparty/multikey HE, and especially approximate-number schemes) do not directly expose the same share structure that these protections rely on.

Therefore, despite the generality claimed in [176], extending these outsourcing techniques to HE based outsourced computation, particularly to approximate arithmetic settings such as CKKS, is non trivial. Moreover, follow up

work in this line, such as [151, 231], has so far not provided HE based instantiations from which to draw firm conclusions. A notable exception is the concurrent work [261] on server aided MPC, where the authors present two protocol variants. One is based on MPC among multiple servers, similarly to [176], and the other is based on FHE. In particular, they show that the secret sharing approach can in fact be combined with FHE. If this line of research is to be compared to ours, then one should compare it in its simplest form, i.e., with a two-worker setup, which effectively translates the threshold honest-majority assumption into a server–server non-collusion assumption. Such a comparison would also currently fall short with respect to malicious-security countermeasures. These appear to be a major advantage of LSS-based outsourced MPC frameworks, whereas we do not (yet) describe corresponding countermeasures in our setting. In our case, such countermeasures would likely rely on methods like NIZKs or SNARKs, which are substantially heavier primitives than the MAC-based mechanisms used in this line of research. Alternatively, one could extend our work to a more robust decryption procedure, such as in [315], which would effectively eliminate the non-collusion assumption. A notable concurrent work that likewise considers FHE-based outsourced computation with a similar distributed decryption procedure, and that simultaneously meets our target criteria under stronger security guarantees, is [261]. However, in that work the inputs are contributed by input clients via verifiable secret sharing, rather than being directly encrypted. We leave a full analysis and comparison to future work, after extending our framework in this direction.

4.2.2. Formal Modeling of Outsourced Computation

In this section, we present the construction for non-interactive outsourced computation based on black-box homomorphic encryption. To this end we firstly define the according ideal functionality together with a formal description of a real protocol. Eventually, we show a concise proof sketch for the security within the UC framework. The full proof can be found in [42].

4.2.3. Ideal Functionality

For our outsourced secure function evaluation framework, we define a functionality viewed in Figure 4.1 that matches the interactions of participating

Functionality $\mathcal{F}_{\text{OutComp}}$

Setup The functionality is parameterized with an (m+1)-ary function $f : X^{m+1} \to Y$, and empty lists $L_{P,X}$ and $L_{P,Y}$. The functionality interacts with n initiators $\mathsf{P}_{\mathsf{Y,j}}$ for $j \in [n]$, m input parties $\mathsf{P}_{\mathsf{X,i}}$ for $i \in [m]$, the calculator server S_{C}, the decryptor server S_{D} and the simulator $\mathcal{S}$. Additionally, initialize an empty list $R_{X,Y} = \emptyset$ for registered clients

Registration of Client $\mathsf{P}_{\mathsf{Y,j}}$ or $\mathsf{P}_{\mathsf{X,i}}$: Whenever a client C sends a message $(\text{register}, \text{sid}, \text{pid})$ check whether pid is already in the list $R_{X,Y}$ of registered parties. If not, send back ok and add pid to $R_{X,Y}$, else send $\perp$.

Outsourcing of $\mathsf{P}_{\mathsf{X,i}}$'s input: Whenever receiving the message $(\text{outsource}, \text{sid}, x_i, \mathsf{P}_{\mathsf{X,i}})$ from an input party $\mathsf{P}_{\mathsf{X,i}}$, add the entry $(\mathsf{P}_{\mathsf{X,i}}, x_i)$ to the list of outsourced input parties' entries $L_{P,X}$. If there is already an existing entry $(\mathsf{P}_{\mathsf{X,i}}, x_i')$ for the given input, replace the existing input x_i' with the new input x_i. Additionally send the message $(\text{input}, \mathsf{P}_{\mathsf{X,i}})$ to $\mathcal{S}$

Outsourcing of $\mathsf{P}_{\mathsf{Y,j}}$'s input: Whenever receiving the message $(\text{outsource}, \text{sid}, y_j, \mathsf{P}_{\mathsf{Y,j}})$ from an initiator client $\mathsf{P}_{\mathsf{Y,j}}$, add the entry $(\mathsf{P}_{\mathsf{Y,j}}, y_j)$ to the list of outsourced input parties' entries $L_{P,Y}$. If there is already an exisiting entry for the given input, replace the existing input with the new input. Send the message $(\text{input}, \mathsf{P}_{\mathsf{Y,j}})$ to $\mathcal{S}$

Protocol Computation: Upon receiving $(\text{start}, \text{sid}, \text{ssid})$ from an initiator $\mathsf{P}_{\mathsf{Y,j}}$, send $(\text{ssid}, \mathsf{P}_{\mathsf{Y,j}})$ to notify $\mathcal{S}$. If $\mathcal{S}$ returns $(\text{ssid}, \mathsf{P}_{\mathsf{Y,j}})$, send a notification $(\text{ssid}, \mathsf{P}_{\mathsf{Y,j}})$ to S_{C} and S_{D}. Upon receiving $(\text{ready}, \text{ssid})$ from S_{C} and S_{D}, retrieve all $(x_i, \mathsf{P}_{\mathsf{X,i}})$ for $i \in [m]$ from $L_{P,X}$ and $(y_j, \mathsf{P}_{\mathsf{Y,j}}) \in L_{P,Y}$:

- If some $(x_i, \mathsf{P}_{\mathsf{X,i}})$ has not been stored yet, send $(\text{output}, x_i, \text{fail})$ to the initiator $\mathsf{P}_{\mathsf{Y,j}}$, S_{C} and S_{D}.
- Else, compute $z \leftarrow f(y_j, \{x_i\}_{i \in [m]})$ and store $(\text{sid}, \text{ssid}, z)$. If S_{D} and $\mathsf{P}_{\mathsf{Y,j}}$ are corrupted, send (result, z) to $\mathcal{S}$, otherwise, send $(\text{result}, \text{ssid})$ to $\mathcal{S}$.

Protocol output: Upon receiving $(\text{output?}, \text{sid}, \text{ssid})$ from $\mathsf{P}_{\mathsf{Y,j}}$, check whether some entry $(\text{sid}, \text{ssid}, z)$ is stored. If this is the case, forward the message to $\mathcal{S}$, else output $(\text{sid}, \text{ssid}, \perp)$ to $\mathsf{P}_{\mathsf{Y,j}}$. When getting answer ok from $\mathcal{S}$, send $(\text{output}, \text{ssid}, z)$ to the initiator $\mathsf{P}_{\mathsf{Y,j}}$ and if $\mathsf{P}_{\mathsf{Y,j}}$ is corrupted additionally to $\mathcal{S}$; else send $(\text{output}, \text{ssid}, \perp)$ to the initiator $\mathsf{P}_{\mathsf{Y,j}}$.

Figure 4.1.: Outsourced Computation Functionality $\mathcal{F}_{\text{OutComp}}$.

parties. This includes the ability of clients to register to the protocol after setting up the protocol and model the clients' actions as separate phases from the actual computation rounds. Furthermore, we enable the reusability of client's outsourced inputs across multiple computation rounds in order to avoid unnecessary communication between clients and servers. Therefore we split the standard outsourced secure function evaluation functionality into several distinct phases:

Registration. First, we integrate a registration phase to the functionality. This phase is designed for new clients to join a fresh computation round without setting up a new protocol instance.

Outsourcing. The next two phases are the outsourcing phases for initiators' and clients' inputs. Those are modeled as separate phases apart from the actual computation phase.

Computation. In the computation phase, the ideal functionality $\mathcal{F}_{\mathsf{OutComp}}$ receives a trigger from the initiating party $\mathsf{P}_{\mathsf{Y,j}}$ and computes the function f over the outsourced inputs of all input clients $\mathsf{P}_{\mathsf{X,i}}$ and the initiator $\mathsf{P}_{\mathsf{Y,j}}$.

Output. The initiator $\mathsf{P}_{\mathsf{Y,j}}$ may trigger the functionality $\mathcal{F}_{\mathsf{OutComp}}$ at any time and query the computation result, which will be delivered to the client once it is ready. In particular, only the initiating client receives the output, while all other clients can remain offline indefinitely.

We decouple computation from output to model clients that aren't always online. In practice, a client can submit a request and go offline, then later reconnect and ask the server whether the computation has finished to retrieve the result.

4.2.4. Protocol Description

The idea of our general outsourced multi-party protocol is to compose an arithmetic circuit (noted as f_1) over the inputs of all input clients, which can be implemented by a somewhat homomorphic encryption scheme, with a general function (noted as f_2) that can be possibly invoked by a one-sided two-round standalone two-party sender-private SFE protocol Π_{2PC} applying homomorphic encryption (a general construction of the two-party sender-private SFE protocol can be seen in Figure 4.4). The overall function f is thus decomposed into $f := f_1 \circ f_2$, where f_1 is secretly computed by S_{C} only and f_2

Protocol Π_{OutComp} – Inputs/Outputs

Private inputs: Each initiator $\mathsf{P}_{\mathsf{Y,j}}$ has input y_j, where $\mathcal{P}_{\mathsf{Y}} = \{\mathsf{P}_{\mathsf{Y},0}, ..., \mathsf{P}_{\mathsf{Y},n-1}\}$. Each input party $\mathsf{P}_{\mathsf{X,i}} \in \mathcal{P}_{\mathsf{X}}$ has input x_i, where $\mathcal{P}_{\mathsf{X}} = \{\mathsf{P}_{\mathsf{X},0}, ..., \mathsf{P}_{\mathsf{X},m-1}\}$.
Public inputs: Public parameters, an arithmetic circuit f_1 and a stand-alone secure two-round one-sided two-party protocol Π_{2PC} realizing a function $f_2 : X \times Y \rightarrow Z$.
Outputs: At any round q, $\mathsf{P}_{\mathsf{Y,j}}$ outputs $z_q \leftarrow f_2(y_j, f_1(x_1, \cdots, x_m))$.

Figure 4.2.: Outsourced Computation Protocol Π_{OutComp}: Inputs/Outputs

during the execution of protocol Π_{2PC} between the virtual parties $(\mathsf{P}_{\mathsf{X,i}}, \mathsf{S}_{\mathsf{C}})$ and $(\mathsf{P}_{\mathsf{Y,j}}, \mathsf{S}_{\mathsf{D}})$. We discuss variants of f_1 and f_2 and their impact on security and realizability later in Section 4.2.7.

The reason why the required two-party protocol is specifically an SFE but not a more general MPC protocol is due the fact that the computation of the underlying protocol should be executed within one single computation phase and cannot be split over several distinct phases, such as commitments.

Compared to the functionality $\mathcal{F}_{\mathsf{OutComp}}$, the protocol Π_{OutComp} shown in Figure 4.3 has an additional server setup phase for the decryption server S_{D} to generate the key pair and sharing the generated public key with the calculation server S_{C}. In this phase, the decryptor server generates its own key pair and registers it to the hybrid functionality $\mathcal{F}_{\mathsf{KRK}}$, which is defined according to [45] and can be found in Figure 4.11, where the key pair is stored and the public key is distributed to the calculator and each new registered client.

In the client registration phase, each new client invokes $\mathcal{F}_{\mathsf{KRK}}$ to receive the decryptor's public key. In the outsourcing phase, an input client encrypts its input using the decryptor's public key $c_{x,j} = ENC(\mathsf{pk}, x_j)$ and sends the ciphertext to the calculator. Additionally, the initiator samples a one-time pad r and sends it together with the encrypted input to the Calculator.[8]

[8] The Calculator is permitted to know r. The mask must be hidden only from the Decryptor.

Protocol Π_{OutComp} – Execution

Private Inputs, Public Inputs and Outputs are depicted in Figure 4.2.

Initialization of servers $\mathsf{S_C}$ and $\mathsf{S_D}$: $\mathsf{S_D}$ generates a fresh random key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathit{GEN}(1^{\lambda})$ and invokes $\mathcal{F}_{\mathsf{KRK}}$ (Figure 3.6) with $(\mathsf{register}, \mathsf{sid}, \mathsf{pk}, \mathsf{sk})$ in order to register its generated key pair. $\mathsf{S_C}$ invokes $\mathcal{F}_{\mathsf{KRK}}$ with $(\mathsf{retrieve}, \mathsf{sid}, \mathsf{S_D})$ in order to receive $\mathsf{S_D}$' public key pk.

Registration of Client $\mathsf{P_{Y,j}}$ or $\mathsf{P_{X,i}}$: Whenever a new client wants to register itself, it invokes $\mathcal{F}_{\mathsf{KRK}}$ with $(\mathsf{retrieve}, \mathsf{sid}, \mathsf{S_D})$ to get the public key pk of $\mathsf{S_D}$.

Outsourcing of $\mathsf{P_{X,i}}$'s input: Whenever an input client $\mathsf{P_{X,i}}$ wants to outsource its input x_i, it encrypts its input using $\mathsf{S_D}$'s pk to compute $\mathsf{c}_{x,i} = \mathit{ENC}(\mathsf{pk}, x_i)$ and sends $(\mathsf{sid}, \mathsf{pid}, \mathsf{c}_{x,i})$ to $\mathsf{S_C}$.

Outsourcing of $\mathsf{P_{Y,j}}$'s input: Whenever an initiating client $\mathsf{P_{Y,j}}$ wants to outsource its input y_j, it encrypts its input using $\mathsf{S_D}$'s pk to compute $\mathsf{c}_{y,j} = \mathit{ENC}(\mathsf{pk}, y_j)$. Then it generates a random mask $r_j \leftarrow \mathcal{M}$ and encrypts it using $\mathsf{S_D}$'s pk as $\mathsf{c}_{r,j} = \mathit{ENC}(\mathsf{pk}, r_j)$. Finally, it sends $(\mathsf{sid}, \mathsf{pid}, \mathsf{c}_{y,j}, \mathsf{c}_{r,j})$ to $\mathsf{S_C}$.

Protocol computation:

1. Whenever $\mathsf{S_C}$ receives a message $(\mathsf{start}, \mathsf{sid}, \mathsf{ssid})$, it checks whether ssid is stored yet. If this is the case, it outputs $\perp$ to $\mathsf{P_{Y,j}}$. Otherwise, it stores ssid and computes $\mathsf{c}_a \leftarrow f_1(\{x_i\}_{i\in[m]})$. Afterwards, it follows the sender's instructions (i.e. step 2) of the two-party protocol Π_{2PC}: on the values $\mathsf{c}_{y,j}$ and c_a in order to retrieve $\mathsf{c}_{a'}$ and masks the computation result $\mathsf{c}_{d'} = \mathsf{c}_{a'} + \mathsf{c}_{r,j}$
2. $\mathsf{S_C}$ sends $(\mathsf{sid}, \mathsf{ssid}, \mathsf{c}_{d'})$ to $\mathsf{S_D}$, which decrypts $d' \leftarrow \mathit{DEC}(\mathsf{sk}, \mathsf{c}_{d'})$ and stores $(\mathsf{sid}, \mathsf{ssid}, d')$.

Output:

1. Whenever $\mathsf{P_{Y,j}}$ requests $\mathsf{S_D}$ $(\mathsf{output?}, \mathsf{sid}, \mathsf{ssid})$, $\mathsf{S_D}$ checks whether some $(\mathsf{sid}, \mathsf{ssid}, d')$ is stored for given $(\mathsf{sid}, \mathsf{ssid})$. If this is the case, it sends $(\mathsf{sid}, \mathsf{ssid}, d')$ to $\mathsf{P_{Y,j}}$. Else $\mathsf{S_D}$ sends $(\mathsf{sid}, \mathsf{ssid}, \mathsf{NotFinished})$ to $\mathsf{P_{Y,j}}$.
2. $\mathsf{P_{Y,j}}$ unmasks d' by computing $d = d' - r$, follows step 3 in Π_{2PC} on value d and outputs z.

Figure 4.3.: Outsourced Computation Protocol Π_{OutComp}: Execution

Each computation round is invoked by an initiator who sends the message (start, sid, ssid) to the calculator, where ssid is a fresh sub-session id. Then, the calculator evaluates the arithmetic circuit f_1 over the input clients' inputs $c_a = f_1(c_{x,1}, ..., c_{x,n})$ and follows the instructions of two-party protocol Π_{2PC} for the sender (i. e. step 2 in Figure 4.4) in order to compute the encrypted interim result, which is forwarded to the decryptor afterwards. The decryptor decrypts the encrypted interim result and stores it for the result retrieval invoked by the initiator.

In the output phase, the initiator sends the sub-session id ssid to the decryptor to query the result. The decryptor goes through the stored interim results and checks whether the computation labeled with the given *ssid* is already finished. If not, the decryptor outputs *notfinished* and the initiator has to talk to the decryptor at a different time point. Otherwise, the decryptor sends the decrypted (masked) interim result to the initiator, who removes the masking one-time pad and follows the last step of the two-party computation protocol (i. e. step 3 in Figure 4.4) to compute the final result.

Underlying Two-Party Protocol. The underlying two-party protocol as shown in Figure 4.4 used in our outsourced protocol is a non-interactive one-sided two-party sender-private function evaluation over two inputs, which realizes a simple functionality described in Figure 4.5. The only primitive required within our general construction is an *IND-CPA* secure somewhat homomorphic encryption scheme.[9] The general protocol structure works as follows: as a global setup, the receiver (or client) has a key pair of a somewhat homomorphic encryption scheme, the sender (or server) has the respective public key of the HE scheme. The global setup for both parties consist of the public parameters of the homomorphic encryption scheme and a composition $f = f_R(\mathbf{y}, f_S(\mathbf{y}, \mathbf{x}))$ of the function f to be computed on.

Both functions f_S and f_R should be computable by the homomorphic encryption scheme and represent the computation instructions of both parties: f_S is the arithmetic circuit over the encrypted inputs $c_{\mathbf{x}}$ and $c_{\mathbf{y}}$ computed by the sender S and f_R represents the postprocessing done by the receiver R over the decrypted interim result $\mathbf{z}'$ and its input $\mathbf{y}$ in order to compute the protocol's result z. Additionally, the inputs of both parties are vectors of the

[9] $\text{IND} - \text{CPA}^{\text{D}}$ becomes important only when the Π_{2PC} is used in Π_{OutComp}. For the standalone security $\text{IND} - \text{CPA}$ suffices.

Protocol Π_{2PC}

Private inputs: The sender party S has input x and auxiliary information pk for a homomorphic encryption scheme HE and the receiver party R has input y and has auxiliary information (sk, pk).
Public inputs: Public parameters and an $IND\text{-}CPA^D$ secure somewhat homomorphic encryption scheme $HE = (GEN, ENC, DEC)$ and a set of functions $f, f_S, f_R : Y \times X \rightarrow Z$ with $f = f_R(y, f_S(y, x))$ computable by HE
Outputs: R outputs $\mathbf{z} = f(x, y)$.

Protocol computation:

1. R encrypts its input $\mathrm{c}_y = ENC(\mathrm{pk}, y)$ and sends $(\mathrm{pk}, c_\mathrm{y})$ to S.
2. When S receives a message $(\mathrm{pk}, \mathrm{c}_y)$ from R, S encrypts its input vector $\mathrm{c}_x = ENC(\mathrm{pk}, x)$ and computes $\mathrm{c}_s = f_S(\mathrm{c}_y, \mathrm{c}_x)$. S then sends c_s to R.
3. When R receives the ciphertext c_s, it decrypts c_s to $z' = DEC(\mathrm{sk}, \mathrm{c}_s)$. Then it evaluates c_s to $z = f_R(z', y)$ and outputs z at the end.

Figure 4.4.: The standalone secure one-sided sender-private 2-party SFE protocol Π_{2PC}

same size. In the first step, the receiver encrypts their input using the public key $c_\mathrm{y} = ENC(\mathrm{pk}, \mathbf{y})$ and sends the encrypted message to the sender. The sender then encrypts their own input $c_\mathbf{x} = ENC(\mathrm{pk}, \mathbf{x})$ and evaluates a given arithmetic circuit $c_\mathbf{s} = f_S(c_\mathrm{y}, c_\mathbf{x})$ over the two encrypted inputs and sends the interim result back to the receiver. The receiver is then able to decrypt the interim result $\mathbf{z}' = DEC(\mathrm{sk}, c_\mathrm{S})$. Depending on the evaluated function, the receiver might have to do some post-processing over the interim result in order to get the function's result. In this case, the client computes a local function $\mathbf{z} = f_R(\mathbf{z}', \mathbf{y})$ over the interim result $\mathbf{z}'$ and its local input and outputs the given result as the protocol's result.

Sender-Private Property. This property stems from the sender contributing its inputs only in an encrypted form. In general, there are one-sided two-party SFE protocols which do not require the sender to encrypt their inputs, such

Functionality $\mathcal{F}_{2PC}$

The functionality is parameterized by a function over two parameters $f : X \times Y \to Z$. The functionality interacts with two parties, the sender S and the receiver R.

Execution

- When getting input set X from S and input set R from R, compute the result $z = f(x, y)$ and output z to R.

Figure 4.5.: Two-Party Functionality $\mathcal{F}_{2PC}$.

as in [98]. However, this additional encryption step on the sender's side is required since in outsourced protocols, the dataset stored in S_C needs to be protected due to the privacy of the clients. If such a requirement is in place, which is usually the case for outsourced computation, then protocols such as [98] cannot be used as is. Every such non-interactive one-sided two-party sender-private secure function evaluation protocol can be used as an underlying protocol for our general outsourced construction, if we make a small modification: The key pair used for the homomorhic encryption scheme must be a global setup across multiple rounds (in a single session). This modification does not affect the standalone security of the underlying protocol, but is merely needed for the proper security reduction.

4.2.5. Security Analysis

In this section, we provide the Theorem 2 and introduce the proof idea. Since the whole communication of all parties is over ciphertexts using the somewhat homomorphic encryption scheme—except the communication between the decryptor and the initiator—the simulation is straightforward since the simulator is either able to extract the correct inputs (hence we have to require to be in the $\mathcal{F}_{KRK}$-hybrid model) or to simulate a legitimate ciphertext message due to the IND-CPA^D security of the homomorphic encryption scheme. The communication between the decryptor and the initiator is also simulatable since the message is a legitimate (decrypted) message of the protocol Π_{2PC} that standalone securely realizes f_2 where initiator's encrypted

input and the output of f_1 can be viewed as the the encrypted inputs of Π_{2PC}.

Having the composition of f using f_1 and f_2 in mind, we are able to show how one execution round of our protocol $\Pi_{OutComp}$ can be transformed into a standalone one-sided two-party sender-private protocol Π_{2PC}: Merge the input clients $P_{X,i}$ and the calculator S_C into one party $(P_{X,i}, S_C)$ and view the output encrypted of f_1 (which is a circuit evaluation solely over the input clients) as one encrypted input of f_2. Also merge the decryptor S_D and the initiator $P_{Y,j}$ into one party $(S_C, P_{Y,j})$. The result is the computation of f_2.

Theorem 2 Assume a one-sided two-round two-party sender-private SFE protocol Π_{2PC} standalone-securely realizing a two-party functionality f_2 (viewed in Figure 4.5 that bases solely on a somewhat homomorphic encryption scheme, an arithmetic circuit c_f realizing a function f_1 using an IND-CPA^D secure somewhat homomorphic encryption scheme $HE = (GEN, ENC, DEC)$ and a function f in $\mathcal{F}_{OutComp}$ is defined as $f = f_1 \circ f_2$. Then the protocol $\Pi_{OutComp}$ realizes the functionality $\mathcal{F}_{OutComp}$ in the $\mathcal{F}_{KRK}$-hybrid model with static corruption in the presence of a semi-honest adversary, assuming that the servers S_C and S_D do not collude.

The full proof was done by Sarai Eilebrecht and Yufan Jiang in [42] and thus we show here only a sketch and refer the interested reader to the original publication.

Proof In the case of a corrupted Initiator the simulator is able to setup a key pair on its own by simulating $\mathcal{F}_{KRK}$, extract the initiator's (modified) input using the self-generated secret key and compute the simulation of Π_{2PC} in order to compute a valid message coming from the decryptor. Note that even if multiple initiators are corrupted, they are not able to learn more than their inputs and the respective outputs, since each computation and output phase is computed sequentially, which does not conflict with the security of the underlying standalone secure two-party protocol Π_{2PC}.

In the case of a corrupted input client, the simulator is able to generate its own key pair. Thus, $\mathcal{S}$ can extract the (modified) inputs sent by the environment. Note that even if several input clients are corrupted, the corrupted parties cannot learn more than their given inputs.

In the case of an initiator and input clients being corrupted, the simulator is able to use a self-generated key pair of the HE scheme by simulating the

hybrid functionality $\mathcal{F}_{\text{KRK}}$ and thus able to extract the (modified) inputs of the corrupted client parties. By the standalone security of the underlying protocol Π_{2PC}, the simulator is able to follow Π_{2PC}'s simulation in order to generate an appropriate message sent by the decryptor to the corrupted initiator. Again, since the given phases are computed only sequentially, the security of the construction can be reduced to the standalone security of Π_{2PC}, even if multiple client parties are corrupted.

In the case of the S_C, initiator and input client being corrupted, the simulator is able to extract the corrupted parties' inputs again by simulating $\mathcal{F}_{\text{KRK}}$ and generating a key pair on its own on behalf of the simulated $\mathcal{F}_{\text{KRK}}$. Since the corrupted calculator receives all encrypted inputs of all clients (including the honest ones), the simulator has to produce the encrypted inputs of the honest clients. Due to the IND-CPA$^{\text{D}}$ security of the HE scheme, $\mathcal{S}$ can easily fake those messages by generating some random ciphertexts. If additionally some initiators are corrupted, the simulator also is able to generate a valid message sent from the decryptor to the initiator by following the simulation of Π_{2PC}.

In the case of the decryptor, initiators and input clients being corrupted, the simulator is able to learn the secret key dedicated for the corrupted decryptor by simulating $\mathcal{F}_{\text{KRK}}$ and therefore getting the whole key pair for *HE*. Since $\mathcal{S}$ then holds the secret key, it is able to extract the corrupted parties' inputs and due to the standalone security of the underlying protocol Π_{2PC}, $\mathcal{S}$ is able to follow Π_{2PC} simulation instructions to generate a valid message coming from the honest calculator and to all honest initiators in each computation and output phase. Since we assume that the decryptor does not collude with the calculator, the corrupted decryptor is not able to learn the honest clients' encrypted inputs.

4.2.6. IND-CPA$^{\text{D}}$ Attacks on HE based Outsourced Computation

The works [103, 197] have shown that outsourced computation based on both approximate and exact lattice-based FHE schemes cannot, in general, be reduced to the IND-CPA security of these schemes.

IND-CPA^{D} Attacks The authors describe attacks on various implementations of CKKS [104], BGV/BFV [66, 67, 147], and TFHE [106] that result in full secret key recovery. The attacks in [197] target the differences in approximate evaluations, while those in [103] exploit the observation that the correctness of exact FHE schemes is tightly coupled with the noise distribution. This distribution can be exploited by repeatedly summing ciphertexts of zeros until the output flips to a 1. Simply put, this makes it possible to compute the exact magnitude of the noise and, subsequently, recover the secret key. To guarantee security in general, a strengthening of FHE security is therefore required, which the authors of [197] introduce as IND-CPA^{D} security. This definition can be achieved through various means, and determining which measure offers the best trade-off between security and efficiency is an ongoing debate. However, all proposed measures require control over the noise distribution during computations. A promising approach is presented by Alexandru et al. [15], who introduced application-aware homomorphic encryption. This approach suggests controlling noise based on the specific application being computed, rather than assuming worst-case noise consumption. This strategy appears particularly suitable for our outsourcing framework because the outsourced protocol is restricted to a predefined evaluation function, which cannot be altered for a specific key pair after its initial setup. Unfortunately, as discussed in [103], this approach introduces an additional challenge in the engineering of HE-based outsourced computation. Further research is needed to streamline this process and reduce its susceptibility to errors.

Mitigation Strategies for Π_{OutComp} Although IND-CPA^{D} security is generally required, there are specific scenarios in our outsourcing framework where the attack requirements, as described in [103, 197], are not met. Consequently, the attack can be thwarted through organizational measures, and IND-CPA^{D} security suffices. For exact schemes, the attack requirements correspond to the query types of the IND-CPA^{D} security game:

- (A) Decryption requests
- (B) Evaluation requests
- (C) Encryption requests

All three types of requests are required to successfully carry out the attack. In our outsourced computation framework, we have two dedicated parties: the Calculator (S_C) and the Decryptor (S_D), who are assumed not to collude

with each other. Additional parties include the Initiating Party ($P_{Y,j}$) and the Input Client Party ($P_{X,i}$). These parties can collude either with the Calculator or the Decryptor, but never with both simultaneously. In section 4.2.13, we describe our outsourced equality filtering protocol, which plausibly satisfies all the attack requirements if an adversary successfully corrupts the parties S_C, $P_{Y,j}$, and $P_{X,i}$. Therefore, only an IND-CPA^D secure HE scheme can provide protection in this case.

The first key requirement is that the Calculator must be corrupted, enabling the issuance of type (B) requests. This scenario is plausible in an honest-but-curious setting. The second requirement is that some input clients must collude with the honest-but-curious calculating server, allowing the adversary to issue encryption requests (C). This is also plausible, as previously discussed, since non-collusion between a server and a client cannot be reliably assumed in a web service scenario. The final requirement, however, can only be met if the adversary (controlling the Calculator and some input clients) also gains access to the decryption results, which are only visible to the Initiating Party. While such a scenario is plausible in protocols similar to our outsourced homomorphic equality protocol described in section 4.2.11, it is not universally applicable.

One important category of outsourced protocols in our framework that does not meet all three requirements simultaneously are protocols where the secret key resides within the Initiating Party. In this case, the Decryptor (S_D) and the Initiating Party ($P_{Y,j}$) effectively become a single party, denoted as (S_D, $P_{Y,j}$). An example of such an architecture can be found in [310]. This simplifies the protocol but also increases the need for proper authorization and authentication measures, as (S_D, $P_{Y,j}$) essentially becomes just another client among many, which, however, is assumed not to collude with S_C. Input clients ($P_{X,i}$) must then be assured that (S_D, $P_{Y,j}$) is not a fake client controlled by S_C.[10] [11] In this case, the masking procedure can be skipped. One example of such an application is an online survey or private analytics/aggregations, where the decryption key resides on the initiating party's laptop, which creates the survey/analysis and keeps the final results private.

[10] If S_D is instead a dedicated server, this assurance becomes easier to establish, as S_D can act as a well-known global entity trusted by all input clients ($P_{X,i}$).

[11] Additionally, there might be data and communication overhead if an input client's data is to be reused across multiple computations initiated by different initiators, as the same input would need to be encrypted multiple times under different (S_D, $P_{Y,j}$) public keys.

4.2.7. Choice of the Outsourced Function f

At the heart of every instantiation of Π_{OutComp} is the choice of the function f that will be computed on the encrypted inputs of the input clients $\mathsf{P}_{\mathsf{X,i}}$ and the initiating party $\mathsf{P}_{\mathsf{Y,j}}$. In Π_{OutComp}, we model the function f as a concatenation of f_1 and f_2, i. e., $f := f_2(y_j, f_1(x_1, \cdots, x_m))$:

- f_1 is an arithmetic circuit that is efficiently computable using a somewhat homomorphic encryption scheme and is solely computed on the inputs from the input clients $\mathsf{P}_{\mathsf{X,i}}$
- f_2 is a function that we require to be standalone-securely realizable by a protocol Π_{2PC} from Figure 4.4 and therefore this function also incorporates the input from the initiating party $\mathsf{P}_{\mathsf{Y,j}}$.

This way we are able to reduce the security and privacy of the overall outsourced protocol in the UC framework to the security and privacy of a simpler protocol in the standalone setting. We observe multiple cases:

1. Outsourced Arithmetic Circuit Calculation (OACC) case, where $f_2 := id$
2. Outsourced Secure Function Evaluation (OSFE) case, where $f_2 \neq id$, i. e., $\Pi_{\mathsf{2PC}}(y, x) = f_2(y, x) \neq (y, x)$
    - Simple two-party sender-private SFE: Receiver does no post-processing after the decryption step 3 in Π_{2PC} from Figure 4.4, i. e., $z = z'$.
    - Proper two-party sender-private SFE: Receiver does non-trivial post-processing, i. e., $z \neq z'$

OACC The OACC case effectively means that the initiating party, $\mathsf{P}_{\mathsf{Y,j}}$, has no input that needs to be incorporated into the computation. In this scenario, no additional proofs are required beyond the IND-CPA$^{\mathrm{D}}$ security of the employed homomorphic encryption, leading to a secure instantiation of Π_{OutComp}.

Simple OSFE The OSFE case covers all outsourced protocols where the input of the initiating party $P_{Y,j}$ must be integrated into the computation. In this context, we can make two important distinctions that may simplify the protocol instantiation process. If the only task for the the initiator $P_{Y,j}$ is to unmask or decrypt the result for the protocol to successfully terminate, i. e., $\Pi_{2PC}(y, x) = f_2(y, x)$, then the simulation will succeed for the instantiation $\Pi_{OutComp}$.

Proper OSFE The most complex case arises when the receiver is required to perform non-trivial computations after the decryption/unmasking step. We also note that [261], a concurrent work closely related to ours, models only the first category (OACC) and does not provide an extension to the second OSFE category.

In this case, one must either employ a Π_{2PC} protocol that already has a standalone simulation-based proof in the literature, or provide this proof themselves. However, we argue that the current state-of-the-art in SFE protocols is rich in primitives such as Private Set Intersection, Oblivious Transfer, Private Information Retrieval, Private Equality Tests, and many others that can be used as building blocks for a secure instantiation of $\Pi_{OutComp}$.

4.2.8. Output Privacy

The formulation of privacy requirements in SFE and MPC protocols is built around the principle that a protocol should not reveal more information than what can be inferred from the result alone. Any information about the inputs that can be inferred from the result is usually termed *leakage*. A key question that remains unresolved is how to quantify the privacy loss due to this leakage, and consequently, how to sanitize the result before publication to limit the leakage to a predetermined, controlled threshold while retaining a sufficient level of usefulness or utility in the final output.

This problem seems to be equivalent to the general challenge that has driven anonymization and privacy research for decades. While this research has yielded numerous solutions for specific datasets and use cases, a universal solution for privacy definition and a corresponding sanitization method applicable to any type of information remains elusive.

Currently, Differential Privacy (DP) [135], along with methods like the exponential mechanism from [210], may appear to offer a potential solution, as DP can be applied to a wide range of real-world scenarios where information needs to be sanitized. However, strong indications suggest that this method of quantifying privacy is not always the optimal choice (see e. g. [24]). Furthermore, even if privacy is quantified using various flavors of DP—such as ϵ-DP, (ϵ, δ)-DP, Pufferfish Privacy [183], or Blowfish Privacy [167]—it remains highly debatable how to accurately assess the actual privacy loss that may result from the chosen threshold according to the specific DP definition used (see e. g. [256]).

For these reasons, we consider extending our protocol Π_{OutComp} to achieve quantifiable output privacy as a non-trivial task that we leave as an open problem. Similar to the selection of authentication and authorization mechanisms, where different use cases (even for the same function f) require different approaches, the choice of an output privacy mechanism for sanitizing results depends heavily on the specific requirements and real-world circumstances of the application implementing the protocol.

We recommend that any implementation of an instantiation using a function f should ensure transparency to input clients about the function f that will be computed on their inputs. This won't solve the problem, but it may bolster a discussion with those directly affected, who may have a better understanding of how the output of f interferes with their privacy.

4.2.9. Choice of Authentication and Authorization

When deploying an instantiation in the real world, one will eventually face the question of how to authorize the input clients $\mathsf{P}_{\mathsf{X},\mathsf{i}}$ and the initiating clients $\mathsf{P}_{\mathsf{Y},\mathsf{j}}$. This issue is not specific to Π_{OutComp} but applies to any protocol that relies on the passive adversary model. Without proper authentication and authorization mechanisms, the system would be vulnerable to trivial denial-of-service attacks by spamming encrypted inputs. The protocol Π_{OutComp} does not include any such protections, and it is implicitly assumed—based on the chosen adversary model and party setup—that every input client and initiator participating in the protocol is authorized and trusted to behave passively (if being corrupted). This implies that authorization is presumed to occur when a party is granted access to communicate with S_C and S_D. At

first glance, this might seem as an unrealistic choice of an adversary model, but there is, from our point of view, a sound reasoning behind it.

Consider, for example, a simple survey aggregation scenario, which is one of the most straightforward instantiations of Π_{OutComp}. Imagine the participation link (i. e., communication access with S_C) is made publicly available. In such a case, many people might attempt to cause disruption by submitting contrived responses to skew the overall results. It also applies to the passive adversary model: while the security of Π_{OutComp} would be easily compromised in a public setting, in an authorized environment, we are able to thwart any remaining honest-but-curious attacks, which is a typical choice for protocols proven secure within passive adversary models. In such a setting we are working with trusted (but curious) parties.

The main takeaway is that Π_{OutComp} requires proper authentication and authorization for participating clients to ensure that only honest-but-curious clients are allowed to participate. There are plenty of methods available for this, making the choice of methods a relatively minor issue. The bigger challenge is to ensure that participants must be trusted to act in a honest-but-curious way only. If the latter challenge cannot be addressed with negligible risk, then the application intended to be realized by an instantiation of Π_{OutComp} will not work as intended.

4.2.10. Separated Duties in Outsourced Computation

Mouchet et al. [220] refer to the non-collusion assumption as the *two-clouds model*. We find this terminology misleading, as it suggests that the calculator and decryptor are limited to two separate cloud servers. In reality, a wide variety of real-world systems can execute the roles of these parties. When mapping parties S_C and S_D to their real-world counterparts, these are essentially programs running on specific technical systems within distinct organizations, managed by different individuals who may administer, interact with, or observe the execution and state of these programs. These individuals are part of a broader organizational structure, where others in different roles can influence those with direct access to the executing code and its state. All of these actors could play a significant role in undermining the non-collusion assumption as we discussed in the introduction of this chapter (Section 4.1).

As a result, we do not recommend deploying S_C and S_D within two different large organizations, as suggested by the *two-clouds model* terminology. Perhaps the most important argument is that, catching a large company misbehaving (or in our case, violating the non-collusion assumption) will, in many cases, not lead to severe consequences for the company. This can result in a potentially favorable risk/cost and merit assessment, encouraging the violation of the non-collusion assumption. Examples of such misbehavior by large companies are plentiful; for instance, consider VW knowingly manipulating emission data [283].

In response, we propose alternative architectures that, in our view, offer a more reliable foundation for minimizing the risk of collusion between S_C and S_D on organizational level. To this end we aim at maximizing the cost of knowingly violating the non-collusion assumption. From our perspective, these costs are highest when the operation of the service provided by S_C or S_D is the sole source of income for the respective organizations. For such an organization, being caught colluding with another party would result in bankruptcy.

While S_C generally needs to be a server with sufficient computational power and bandwidth, the S_D component is more flexible and can be implemented on different systems:

- The simplest implementation is a dedicated server similar to S_C.
- Another widely used option involves implementing S_D within a TEE or HSM component, which can physically reside within the same organization as S_C but be operated by a different organization.
- Additionally, S_D can be realized within e. g., browser-based clients, allowing for a simplified architecture where S_D and $P_{Y,j}$ reside within the same browser client. This effectively means that those two parties are re-formularized as one single entity $(S_D, P_{Y,j})$

The latter variant, if applicable to the use case, simplifies the protocol but also increases the need for proper authorization and authentication measures, as the combination of $(S_D, P_{Y,j})$ becomes just another client among many. Input clients $P_{X,i}$ must then be assured that $(S_D, P_{Y,j})$ are not a fake client controlled by S_C. Additionally, there might be some data and communication overhead if an input client's input should be usable for computation initiated by several initiators since the same input has to be encrypted several times under the different $(S_D, P_{Y,j})$'s public keys. In this case one can skip the

masking procedure. If S_D is a dedicated server, this assurance is more easily established, as S_D can be a well-known global entity among the input clients $P_{X,i}$.

4.2.11. Instantiations

The preceding sections developed a versatile outsourcing model that accommodates multiple deployment choices by instantiating Π_{OutComp} with different party configurations and a standalone secure 2-party subprotocol Π_{2PC}. In what follows, we instantiate Π_{2PC} with a vectorized Private Equality Test (VectorPET) and present benchmarks for the resulting outsourced, batched equality protocol.

4.2.11.1. UC-Secure Equality Filtering for Encrypted Web Services

We consider performing encrypted matching of a collection of values against a single target and producing an indicator vector of encrypted bits that can be aggregated on the client-side. This primitive underlies common web-service endpoints (e.g., key lookups, equality filters, and equality-based counts/aggregations). Encrypted DBMSs (e.g., [51, 172, 248, 313]) can realize such endpoints end to end. A key caveat, however, is that these systems are built on symmetric HE primitives and thus fall outside the scope of our outsourced protocol, which inherently requires an asymmetric (public-key) HE scheme. Their outsourcing model also differs conceptually: a single honest client owns a database, outsources it to an untrusted server, and later issues queries against that database. Accordingly, in what follows we focus on protocols with asymmetric primitives.

Formally, an HE equality operator is a non-interactive procedure that, given (batched) encryptions of many x_i and of a target a, returns a (batched) encryption of the indicators $[x_i = a]$. This operator plugs modularly into richer secure pattern matching (SPM) and search pipelines (cf. [11, 301]). There is a rich literature (e.g., [60, 101, 102, 171, 187, 188, 189, 218, 276, 310]) on designing such operators and building upon them. In Section 4.2.13 we give an overview of these methods.

One of the simplest and most straightforward solutions represents the values to be matched as integers, subtracts them, and then applies Fermat's Little

Functionality $\mathcal{F}_{\text{2P-VecPET}}$

The functionality interacts with a sender S holding a sequence $X = (x_0, \ldots, x_{n-1})$ and a receiver R holding a sequence $Y = (y_0, \ldots, y_{n-1})$, both over a common domain.

Execution

- Upon inputs X from S and Y from R, compute the indicator vector $\mathbf{e} \in \{0,1\}^n$ defined by $\mathbf{e}_i \leftarrow \left[\, x_i = y_i \,\right]$ for $i \in \{0, \ldots, n-1\}$, and output $\mathbf{e}$ to R.

Figure 4.6.: Vector Private Equality Test (VectorPET) functionality $\mathcal{F}_{\text{2P-VecPET}}$

Theorem (FLT) to the batched differences. Formally it works as follows: Pick a prime plaintext modulus p and compute slotwise,

$$\text{EQ}_{\text{FLT}}(x, a) \;=\; 1 - (x - a)^{p-1} \;\in\; \{0, 1\}\,,$$

so the result encrypts 1 iff $x = a$, else 0. There are no rotations in the equality core, and the method preserves one value per slot packing. This approach is readily implementable in most HE libraries (e.g., Lattigo, OpenFHE, HElib), as it requires only a prime plaintext modulus p and a batching condition such as $p \equiv 1 \pmod{2N}$ for the ring $R_p = \mathbb{Z}_p[X]/(X^{2N} + 1)$, which ensures that $X^{2N} + 1$ splits into linear factors modulo p. Consequently, each slot in the ciphertext holds a single integer from $\mathbb{Z}_p$, and homomorphic operations act slot-wise, enabling simple and favorable SIMD batching. While other approaches may achieve lower multiplicative depth and, in specific circumstances and parameterizations, even better running times, we leave them out of scope for this work and use the FLT approach due to its simplicity to demonstrate an instantiation of our Π_{OutComp} protocol. From this, we conclude that such an equality operator can be UC-securely used in an outsourced manner.

To instantiate Π_{OutComp} with an equality operator, we adopt the ideal functionality in Figure 4.6. We model batching as a first-class property by operating on fixed-length vectors. This formulation also gives the receiver maximal flexibility in choosing its matching mask: $\mathbf{y}$ may be a single value repeated across all slots, a periodic pattern, or any application-specific sequence. Similarly, the sender's input layout can encode arbitrary collections. Consequently, the

semantics of the indicator vector $\mathbf{e}$ adapt to these choices, for example, membership against a single target when $\mathbf{y}$ is constant, or position-wise equality against a template when $\mathbf{y}$ encodes structure. All of these scenarios can be realized in either one-shot or streaming modes by reusing sender/receiver inputs across key epochs (or rotating masks/templates per epoch) while keeping the batching interface unchanged.

The protocol can be seen in Figure 4.7. The operator EQ admits multiple instantiations. A simple "strawman" zero-test computes $(c_{\mathbf{x}} - c_{\mathbf{y}}) \cdot c_{\mathbf{r}}$ with a fresh randomizer $c_{\mathbf{r}}$. The receiver then decrypts and interprets slots that decrypt to zero as matches (the indicator vector is obtained by post-processing). Alternatively, EQ can use Fermat's Little Theorem over $\mathbb{Z}_t$, computing $\mathbf{1} - (c_{\mathbf{x}} - c_{\mathbf{y}})^{\,t-1}$, which produces encrypted $\{0, 1\}$ indicators directly.

Theorem 3 Given an IND-CPA$^{\mathrm{D}}$ secure somewhat homomorphic encryption scheme $HE = (GEN, ENC, DEC)$, the protocol $\Pi_{\mathsf{2P-VecPET}}$ securely realizes $\mathcal{F}_{\mathsf{2P-VecPET}}$ under sequential composition in the presence of a semi-honest adversary $\mathcal{A}$.

Proof To show sequential composability of $\Pi_{\mathsf{2P-VecPET}}$, it suffices to construct a simulator $\mathcal{S}$ that, against any semi-honest $\mathcal{A}$, produces a transcript indistinguishable from a real execution, given the corrupted party's input and the functionality's output. Since the adversary is passive, no rewinding is needed.

Case 1: Sender S is corrupted. The simulator $\mathcal{S}$ is given S's input $\mathbf{x}$ and no output from $\mathcal{F}_{\mathsf{2P-VecPET}}$. It must emulate R's messages to S.

Simulation.

1. $\mathcal{S}$ honestly generates a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow GEN(1^{\lambda})$.
2. $\mathcal{S}$ samples an arbitrary dummy vector $\widetilde{\mathbf{y}}$ from the receiver's input domain (e.g., uniformly at random, or the all-zero vector of the right shape), and computes $c_{\widetilde{\mathbf{y}}} \leftarrow ENC(\mathsf{pk}, \widetilde{\mathbf{y}})$. It simulates receiving $(\mathsf{pk}, c_{\widetilde{\mathbf{y}}})$ from the honest R, exactly as R would send $(\mathsf{pk}, c_{\mathbf{y}})$ in the real protocol.
3. $\mathcal{S}$ samples an arbitrary dummy vector $\widetilde{\mathbf{eq}}$ from the message space, and computes $c_{\widetilde{\mathbf{eq}}} \leftarrow ENC(\mathsf{pk}, \widetilde{\mathbf{eq}})$. It simulates sending $c_{\widetilde{\mathbf{eq}}}$ to the honest receiver interface and halts.

Protocol $\Pi_{2P-VecPET}$

Private inputs: The sender S holds a sequence $\mathbf{x} = (x_0, \ldots, x_{n-1})$ and auxiliary information pk for a somewhat homomorphic encryption scheme *HE*. The receiver R holds a sequence $\mathbf{y} = (y_0, \ldots, y_{n-1})$ and auxiliary information $(\mathsf{sk}, \mathsf{pk})$.
Public inputs: Public parameters of *HE* and a publicly specified equality operator $\mathsf{EQ}(\cdot, \cdot)$ that is evaluable under *HE*.
Output: R outputs the indicator vector $\mathbf{e} \in \{0,1\}^n$ with $\mathbf{e}_i = [\, x_i = y_i \,]$.
Protocol computation:

1. (*Receiver encryption*) R computes $c_{\mathbf{y}} \leftarrow \mathit{ENC}(\mathsf{pk}, \mathbf{y})$ and sends $(\mathsf{pk}, c_{\mathbf{y}})$ to S.

2. (*Sender evaluation*) Upon receiving $(\mathsf{pk}, c_{\mathbf{y}})$, S encrypts its input $c_{\mathbf{x}} \leftarrow \mathit{ENC}(\mathsf{pk}, \mathbf{x})$ and homomorphically computes

$$c_{\mathrm{eq}} \leftarrow \mathsf{EQ}(c_{\mathbf{x}}, c_{\mathbf{y}}),$$

   where c_{eq} encrypts the slotwise indicators $[x_i = y_i]$. S sends c_{eq} to R.

3. (*Receiver decryption*) R decrypts c_{eq} to obtain $\mathbf{e} \leftarrow \mathit{DEC}(\mathsf{sk}, c_{\mathrm{eq}})$ and outputs $\mathbf{e}$.

Figure 4.7.: Vector Private Equality Test (VectorPET) protocol $\Pi_{2P-VecPET}$

Indistinguishability. First, the simulated public key pk is distributed exactly as in the real world (honest key generation), hence is perfectly indistinguishable. Second, by *IND-CPA* security of *HE*, the ciphertext $c_{\widetilde{\mathbf{y}}} = \mathit{ENC}(\mathsf{pk}, \widetilde{\mathbf{y}})$ is computationally indistinguishable from $c_{\mathbf{y}} = \mathit{ENC}(\mathsf{pk}, \mathbf{y})$ for any real receiver input $\mathbf{y}$ of the same shape.

Since S never sees any decryptions or further messages from R in this round, the entire view of the corrupted sender (its input $\mathbf{x}$, the received $(\mathsf{pk}, c\ldots)$, and any ciphertexts it locally computes/outputs) is indistinguishable between the simulated and real executions.

This argument is agnostic to the specific algebraic form of EQ and covers any PPT, publicly evaluable equality operator under *HE* (with public evaluation/automorphism keys treated as public parameters). In particular, it applies to: (i) the "strawman" zero-test $(c_{\mathbf{x}} - c_{\mathbf{y}}) \cdot c_{\mathbf{r}}$ with fresh $c_{\mathbf{r}}$; (ii) the Fermat test over $\mathbb{Z}_t$, $\mathbf{1} - (c_{\mathbf{x}} - c_{\mathbf{y}})^{t-1}$; (iii) Frobenius/field-norm constructions over $\mathbb{F}_{p^\ell}$ that compute $N(z) = \prod_{i=0}^{\ell-1} z^{p^i}$ for $z = x - a$ and map to $\{0, 1\}$; and (iv) bitwise XNOR+AND circuits that form per-bit XNORs and aggregate them via a balanced product tree (e.g., $\prod_{i=0}^{\mu-1}(1 - (x_i - a_i)^2)$ over odd moduli). By *IND-CPA*, replacing $c_{\mathbf{y}}$ with an encryption of any dummy $\widetilde{\mathbf{y}}$ preserves indistinguishability through any such public evaluation pipeline, hence the corrupted-sender view remains indistinguishable.

Case 2: Receiver R is corrupted. Here the simulator $\mathcal{S}$ is given R's input $\mathbf{y}$ and the ideal output $\mathbf{e} \in \{0,1\}^n$ with $\mathbf{e}_i = [x_i = y_i]$. It must simulate the view of a corrupted R interacting with an honest S under the protocol $\Pi_{\mathsf{2P-VecPET}}$.

Simulation.

1. $\mathcal{S}$ honestly generates a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathit{GEN}(1^\lambda)$, computes $c_{\mathbf{y}} \leftarrow \mathit{ENC}(\mathsf{pk}, \mathbf{y})$, and sends $(\mathsf{pk}, c_{\mathbf{y}})$ to S, exactly as in the real protocol.
2. Using the ideal output $\mathbf{e}$, $\mathcal{S}$ prepares a valid output ciphertext for R: it sets $\widetilde{c}_{\mathrm{eq}} \leftarrow \mathit{ENC}(\mathsf{pk}, \mathbf{z}')$ and applies the same canonical post-processing that the honest sender applies to its evaluated ciphertext before transmission (e.g., public relinearization, modulus switching/rescaling to the designated output level, and metadata normalization). It then sends this $\widetilde{c}_{\mathrm{eq}}$ to R.

Indistinguishability. The first message $(\mathsf{pk}, c_{\mathbf{y}})$ is distributed exactly as in a real execution (honest key generation and encryption), hence it is perfectly simulated. In a real execution, the honest sender computes

$$c_{\mathrm{eq}} \;\leftarrow\; \mathsf{EQ}\big(\mathit{ENC}(\mathsf{pk}, \mathbf{x}),\; c_{\mathbf{y}}\big)$$

while the receiver would compute $\mathit{DEC}(\mathsf{sk}, c_{\mathrm{eq}}) = \mathbf{z}$ and either $\mathbf{z} = \mathbf{e}$ or the receiver has to do some post-processing to recover the $\mathbf{e}$ from $\mathbf{z}$. By correctness, the simulated $\widetilde{c}_{\mathrm{eq}}$ decrypts to $\mathbf{z}'$, so we need to show that the corrupted receiver derives the same $\mathbf{e}$ out of $\mathbf{z}'$. This depends on the choice of the equality operator *EQ*.

Applicability to concrete EQ. Let $\mathbf{z} = \mathit{DEC}(\mathsf{sk}, c_{\mathrm{eq}})$ denote the receiver-side plaintext produced by the real protocol and let Φ_{EQ} be the (public, deterministic) post-processing map that the receiver applies to obtain the indicator vector, i.e., $\mathbf{e} = \Phi_{\mathrm{EQ}}(\mathbf{z})$. The simulator chooses $\mathbf{z}'$ so that $\Phi_{\mathrm{EQ}}(\mathbf{z}') = \mathbf{e}$ and sends a canonical-form encryption of $\mathbf{z}'$.

- **Indicator-producing** EQ **(direct equality):** For Fermat-based $\mathbf{1} - (c_{\mathrm{x}} - c_{\mathrm{y}})^{t-1}$, Frobenius/field-norm, and bitwise XNOR+AND, the decrypted value *is* the indicator vector: $\mathbf{z} = \mathbf{e}$, hence Φ_{EQ} is the identity. The simulator sets $\mathbf{z}' \leftarrow \mathbf{e}$ and returns $\mathit{ENC}(\mathsf{pk}, \mathbf{z}')$ (with the same public canonicalization as the honest sender).
- **Zero-test** EQ **(strawman):** Here $\mathbf{z}$ consists of per-slot zero tests: $\mathbf{z}_i = 0$ iff $x_i = y_i$, and $\mathbf{z}_i \neq 0$ otherwise (with randomness from c_{r}). The receiver's post-processing is $\Phi_{\mathrm{EQ}}(\mathbf{z})_i = [\mathbf{z}_i = 0]$. The simulator sets $\mathbf{z}'_i \leftarrow 0$ whenever $\mathbf{e}_i = 1$, and samples $\mathbf{z}'_i$ uniformly at random from the nonzero plaintexts whenever $\mathbf{e}_i = 0$, so that $\Phi_{\mathrm{EQ}}(\mathbf{z}') = \mathbf{e}$. It then returns $\mathit{ENC}(\mathsf{pk}, \mathbf{z}')$ in the same canonical form as the honest sender's output.

In all cases, the corrupted receiver decrypts to $\mathbf{z}'$ and applies the prescribed (public) Φ_{EQ} to obtain exactly $\mathbf{e}$. Since the first message $(\mathsf{pk}, c_{\mathrm{y}})$ is identically distributed and the second message is a valid ciphertext in the same public canonical form whose decryption and post-processing yield the same $\mathbf{e}$, the simulated and real views are computationally indistinguishable.

Therefore the simulated protocol is compuationally indistinguishable from the real-world protocol execution.

Choice of Parties and the Outsourced Computation Architecture. In Π_{OutComp} we assume two non-colluding servers: the Calculator $\mathrm{S_C}$ (stores ciphertexts, computes f_1 and the public EQ) and the Decryptor $\mathrm{S_D}$ (holds sk and returns only required outputs). To reduce collusion risk, $\mathrm{S_C}$ should be run by an independent, honest-but-curious provider with sufficient compute for SIMD equality and packing. Since data remain encrypted and CPU is the primary bottleneck, $\mathrm{S_C}$ should be run on a server with sufficient computing power.

For $\mathrm{S_D}$, three deployment options are viable. (i) Operate $\mathrm{S_D}$ under a separate organization (e.g., a data controller, institutional key custodian, or regulated trust service) that maintains the decryption keys and exposes only a minimal

result-delivery interface. (ii) Embed S_D as a properly configured TEE/HSM component under S_C's operational control but with independent attestation and audit. (iii) No dedicated decryptor: collapse S_D into the initiator so that each initiator holds (sk, pk) and decrypts its own outputs locally. This removes the server–server non-collusion requirement for that workflow, reduces operational complexity, but it shifts key-management burdens to initiators and eliminates the reuse of a single decryptor across multiple initiators. On the positive side, under option (iii) the $IND\text{-}CPA^D$ attacks that rely on an adversary observing decryption outputs are no longer viable, since by design S_C never receives any decryptions and only the initiator decrypts locally. Also, the overhead of computing and storing the mask is removed. On the other hand, merging S_D with the initiator is generally unsuitable when many initiators access the same outsourced inputs, since it implies per-initiator key material, heterogeneous key epochs, and weaker global auditability/fairness across initiators.

The "no dedicated decryptor" (client-held secret key) pattern is the most common architecture in outsourced-computation works, making (iii) a familiar baseline. Our two-server outsourcing protocol cleanly subsumes (iii) as a special case, showing that the model is a generalization of prevailing designs and thus versatile for modeling a range of outsourcing scenarios.

4.2.12. Benchmarks

We begin with the most computation-light batched equality protocol, which requires only a single ciphertext multiplication but incurs communication equal to the length of the outsourced database. This places it at the low end of computation cost and the high end of communication cost. We then adopt a more computation-intensive variant from the literature, i.e., Fermat-based equality, which reduces communication to the optimal size of the equality indicator vector, trading additional multiplications for bandwidth savings. Other equality operators are left for future work.

"Strawman" Vector PET. All evaluations were performed on a server with an AMD EPYC 7763 64-Core Processor, albeit only 1 core was effectively

utilized due to missing parallelization in the benchmark. The benchmark implementation is available on GitHub.[12]. Since our protocol involves only one multiplication, we selected a simple parameter set with a ternary distribution, specifically $log_2(N) = 13$ and $log_2(Q) = 58$, based on recommendations from [13], achieving a 256-bit security level. We used a 26-bit plaintext modulus. For accommodating larger plaintext spaces that cover e. g. 256-bit hash values one will have to use a different parameterization or employ a Chinese Remainder Theorem in order to encode the hash value into smaller components.

Table 4.1.: Performance metrics for the Calculator, Decryptor, and Initiator, averaged over 100 runs. "Slots" indicates that results scale with any factorization of (clients × number of values per client). For example, 524288 slots can represent 524288 clients with one value each, or 32768 clients with 16 values each.

Slots	Calculator (w/oAggr.)	Calculator (w/Aggr.)	Decryptor	Initiator
32768	248.24ms	3.92ms	1.31ms	68µs
65536	382.02ms	7.76ms	3.25ms	133µs
131072	1.01s	16.22ms	5.22ms	299µs
262144	5.58s	35.17ms	11.87ms	453µs
524288	14.22s	66.59ms	22.34ms	928µs

Table 4.2.: Storage metrics for the Calculator and Initiator given the parameterization of $log_2(N) = 13$ and $log_2(Q) = 58$.

Slots	Storage (Calc.)	Storage (Init. mask)
32768	237.58 kB	106.5 kB
65536	475.14 kB	213 kB
131072	950.28 kB	426 kB
262144	1.9 MB	852 kB
524288	3.8 MB	1.7 MB

[12] `https://github.com/collapsinghierarchy/opsi_sepdutyhe_bench`

Discussion. Table 4.1 reports end-to-end runtimes across batch sizes ("slots"). By "w/Aggr." we mean that the Calculator homomorphically adds incoming (sparse) client ciphertexts into as few packed ciphertexts as possible before running the equality operator. This is a linear, depth-free step (no rotations[13]) that compacts sparse uploads into a small number of dense ciphertexts.

As expected, Calculator CPU time scales roughly linearly in slots, but aggregation substantially reduces the constant by shrinking the number of ciphertexts that enter the costly equality evaluation. The w/oAggr. path (no compaction) performs equality on many small, sparse ciphertexts and therefore pays much higher CPU time (and later bandwidth), whereas the w/Aggr. path pays a modest addition cost up front and then evaluates equality on a minimal set of dense ciphertexts, yielding the smaller runtimes in the table. Decryptor time remains small (one decryption per batch), and Initiator time stays close to a single encryption with minor overhead for preparing its vector and consuming the indicator. Aggregation also lowers Calculator-side storage (see Table 4.2) and reduces network traffic for result delivery, leaving bandwidth as the only potential bottleneck at extreme scales. An optimization not yet implemented is to have the initiator send only a short seed, from which the Calculator deterministically derives the mask via a PRG. This reduces both the initial communication and the initiator-side storage from $O(\text{slots})$ to $O(1)$, up to the seed length.

Fermat's Little Theorem (VectorPET). The "strawman" zero-test has two attractive properties: (i) a minimal multiplicative depth (one ct×ct multiply) and (ii) very simple HE parameterization. By contrast, the Fermat approach evaluates $1-(x-a)^{p-1}$, so the required multiplicative depth grows with the exponent. Using a generic square-and-multiply, the depth is $\lfloor \log_2(p-1) \rfloor$ in our case. A key trade-off comes from the plaintext modulus p. Larger p allows encoding larger per-slot values and can simplify equality logic, but it increases the exponent $p-1$ (and thus the multiplicative depth), forcing a larger ciphertext modulus Q and possibly a larger ring dimension N. In our evaluations we set $p = 65537 = 2^{16}+1$ (bit-length 17), so $p-1$ is a power of two and the equality core requires exactly 16 squarings with no extra multiplies. We use $\log_2(Q) \approx 660$ and $N = 32768$, i.e., a power-of-

[13] We assume each input client is informed which slots to populate. Our benchmarking code implements this placement logic.

two cyclotomic ($m = 2N$). Since $p \equiv 1 \pmod{2N}$, the plaintext CRT splits into linear factors, enabling standard SIMD batching. Hence slot counts are directly comparable to the previous benchmark.

Averaged over 50 runs on a Lenovo T14s, the overhead for a batched Fermat's Little Theorem evaluation is ≈ 5146 ms per ciphertext.[14] Consequently, unless these exponentiations are parallelized across cores, this HE evaluation dominates per-ciphertext runtime in the aggregation pipeline.

Discussion. With the above parameters, a relinearized BGV ciphertext is on the order of ∼5.5–7 MB in memory (depending on the exact RNS prime sizes). Using 7 MB as a safe upper bound, 2^{19} slots (= 524288) require 16 ciphertexts in the aggregation structure, for a total of ≈ 112 MB. On the bandwidth side, the decrypting party returns only the indicator vector $\mathbf{e}$, whose size is one bit per slot: for 2^{19} slots this is 524,288 bits = 64 KiB, compared to ∼1 MB in the "strawman" zero-test variant that returns a full-value vector.

There are alternative equality operators that may be more efficient than the canonical FLT instantiation. While some reduce multiplicative depth or the number of multiplications for a given symbol/bit width, in our regime none provides an orders-of-magnitude (e.g., $10^3\times$) speedup. The equality core remains the dominant cost unless heavily parallelized. The trade-off is clear: moving from the zero-test variant (returning $\log_2 p$-bit residues) to an indicator vector reduces result communication by a factor of roughly $\log_2 p$ (one bit per slot), at the expense of a slower EQ evaluation. Which side to favor depends on the application's constraints, compute budget and parallelism at S_C, available bandwidth/latency for result delivery, and acceptable parameter sizes. For example, streaming scenarios often have small per-window inputs but strict latency targets, which can favor the zero-test path (e.g., ≈ 66 ms on a single core to test 2^{19} slots per window). Conversely, one-shot queries over large databases from constrained devices may prefer indicator-producing variants (e.g., FLT) that cut result size to one bit per slot and thus minimize return bandwidth.

[14] Amortized over 32768 slots this is ≈ 0.15 ms per slot (for a 16-bit value).

4.2.13. Encrypted Equality Operator

Below we systematize the main asymmetric families, their multiplicative depth, SIMD behavior, and known optimizations. We then parameterize costs as a function of the value bit-length k and the array size n, focusing on (i) multiplicative depth (levels consumed by the circuit), (ii) the number of ciphertexts required to pack the inputs (and intermediate digit/symbol layouts), and (iii) the total number of ciphertext–ciphertext multiplications across the whole computation.

(A) Integer-wise *Subtract+FLT* (Fermat zero test in $\mathbb{Z}_p$). This is one of the simplest and most straightforward solutions. It is also easy to implement in most HE libraries (e.g., Lattigo, OpenFHE, HElib), as it requires only a prime plaintext modulus p over $\mathbb{Z}_p$ and otherwise places minimal constraints on the plaintext space.

Pick a prime plaintext modulus p and compute, slotwise,

$$\mathrm{EQ}_{\mathrm{FLT}}(x, a) \;=\; 1 - (x - a)^{p-1} \;\in\; \{0, 1\},$$

so the result encrypts 1 iff $x = a$, else 0. The multiplicative depth for evaluating $(x - a)^{p-1}$ via repeated squaring and a balanced product of selected powers admits the clean bound

$$D_{\mathrm{FLT}} \;=\; \left\lceil \log_2(p-1) \right\rceil \;+\; \left\lceil \log_2 \big(\mathrm{hw}(p-1)\big) \right\rceil,$$

where $\mathrm{hw}(\cdot)$ denotes Hamming weight. If $p - 1$ is a power of two (e.g., $p = 257$ or 65537), then $\mathrm{hw}(p-1) = 1$ and the combine term vanishes, yielding the exact depth $D_{\mathrm{FLT}} = \log_2(p-1)$ (e.g., 8 for 257, 16 for 65537). For general batching primes p, $\mathrm{hw}(p-1)$ is typically small (a few to a dozen), so the extra $\lceil \log_2(\mathrm{hw}(p-1)) \rceil$ term is tiny (often ≤ 3–4), which is why this depth is often summarized as $\approx \lceil \log_2(p-1) \rceil$. There are no rotations in the equality core, and the method preserves one value per slot packing. A common refinement is a digit variant: represent x in base B with $d = \lceil k / \log_2 B \rceil$ digits, apply a small-prime FLT (take $p \approx B{+}1$) per digit, and combine digits via a balanced product

$$\mathrm{EQ}_{\mathrm{digits}}(x, a) \;=\; \prod_{j=0}^{d-1} \big(1 - (x_j - a_j)^{p-1}\big),$$

cf. the digit-wise comparison/equality compositions in [218]. This trades a small increase in multiplication count for a substantial depth reduction (depth $\approx \lceil \log_2(p-1) \rceil + \lceil \log_2 d \rceil$) while retaining one value per slot. Variants in private-search (e.g.,[257, 258, 259, 260]) can be adapted accordingly.

For a k-bit domain, choose $p > 2^k$ to avoid wraparound in the equality test (and $p > n$ if indicators are summed).[15]

In the digit variant with base $B = 2^b$, set $d = \lceil k/b \rceil$ and take $p \approx B{+}1$. With polynomial degree N and batching enabled (i.e., $p \equiv 1 \bmod 2N$), the SIMD slot count is $s = N$, so packing n values requires $c = \lceil n/s \rceil$ ciphertexts (or $c \cdot d$ if digits are stored separately). The total ciphertext–ciphertext multiplications for FLT equality over all inputs are then

$$M_{\mathrm{FLT}} \;\approx\; c{\cdot}\Big(\big\lfloor \log_2(p{-}1) \big\rfloor {+} \mathrm{hw}(p{-}1) {-} 1 \Big) \quad \text{(or multiplied by } d \text{ in the digit case)},$$

all at depth D_{FLT} as above.

(B) *Subtract+Frobenius+MultTree* in $\mathbb{F}_{2^\ell}$ (field-norm zero test). This route leverages the Frobenius automorphism in a binary extension field to obtain a very shallow, word-wise equality. Work per slot over $\mathbb{F}_{2^\ell}$, set $z = x - a$, and compute the field norm

$$N(z) \;=\; \prod_{i=0}^{\ell-1} z^{2^i} \;=\; z^{2^\ell - 1},$$

then output $\mathrm{EQ}_{\mathbb{F}_{2^\ell}}(x, a) = 1 - N(z)$, which is 1 iff $x = a$ and 0 otherwise. Each map $z \mapsto z^{2^i}$ is a Frobenius (plaintext-space) automorphism and is evaluated via a ciphertext automorphism/key-switch, consuming no multiplicative depth. Eventually, only the balanced multiplication tree over the ℓ conjugates consumes depth. Hence, the per-symbol equality depth is

$$D_{\mathrm{field}} = \lceil \log_2 \ell \rceil \qquad \text{with} \qquad M_{\mathrm{field}} = \ell - 1 \;\; \text{ct}{\times}\text{ct multiplications},$$

while preserving one symbol per slot and avoiding rotations in the equality core [188, 189]. For k-bit values represented as $m = k/\ell$ symbols, combine

[15] If such a p is impractical, use a CRT-based approach: pick primes $p_1, \ldots, p_r$ with $\prod_i p_i > 2^k$ (and $\prod_i p_i > n$ if summing), evaluate the test modulo each p_i, and combine the per-modulus results (e.g., by conjunction or CRT reconstruction).

the m symbol-equalities by a balanced product, which adds $\lceil \log_2 m \rceil$ depth and $(m-1)$ multiplications, yielding a per-value depth of $\lceil \log_2 \ell \rceil + \lceil \log_2 m \rceil$. It is not a strict requirement to work over $\mathbb{F}_{2^\ell}$ for this approach. The Frobenius–norm technique generalizes to extensions $\mathbb{F}_{p^\ell}$ for any prime p: compute $N(z) = \prod_{i=0}^{\ell-1} z^{p^i} \in \mathbb{F}_p$ for $z = x - a$, and then output $1 - N(z)^{p-1}$ to collapse to $\{0, 1\}$. See, e.g., Tan et al. [171, 276].

In OpenFHE's BGV/BFV schemes with power-of-two cyclotomics, the ring has cyclotomic order $m = 2N$ for ring dimension N. Packed encoding over $\mathbb{Z}_p$ requires an m-th root of unity in $\mathbb{F}_p$, which means $m \mid (p-1)$, or equivalently $p \equiv 1 \pmod{2N}$.

When this condition holds (for example, with $N = 32768$ giving $m = 65536$ and choosing $p = 65537$), the plaintext CRT factors are all linear, so each slot is just $\mathbb{F}_p$ and the extension degree is $d = \mathrm{ord}_m(p) = 1$. In this case, the Frobenius map $z \mapsto z^p$ acts trivially on each slot, so the "depth-free Frobenius" norm trick collapses to the standard Fermat test: $1 - z^{p-1}$.

To obtain a nontrivial Frobenius action (slot field $\mathbb{F}_{p^\ell}$ with $\ell > 1$) and batching, one would need a cyclotomic modulus m where $\mathrm{ord}_m(p) = \ell > 1$. This typically requires using non-power-of-two cyclotomics, which OpenFHE's PKE schemes currently do not support to a satisfactory extent. As a result, on OpenFHE's default power-of-two rings, the per-symbol equality core is effectively the Fermat test. The same restriction applies to Lattigo's BFV/BGV packing, which also relies on power-of-two cyclotomics and thus yields slot fields of degree $\ell = 1$ over $\mathbb{F}_p$ in the batched setting. Consequently, a nontrivial Frobenius action with batching (i.e., slot fields $\mathbb{F}_{p^\ell}$, $\ell > 1$) is, in practice, attainable with *HElib*, which supports non-2^k cyclotomics and lets one choose m with $\mathrm{ord}_m(p) = \ell > 1$.

Refinement via randomized equality (Bonte–Iliashenko). A constant-depth improvement replaces the per-symbol norm product by a single zero test on a randomized linear combination of differences, thereby removing the dependence of the multiplicative depth on the pattern length M [60]. Concretely, their circuit (Alg. 2) uses

$$\begin{aligned} \mathrm{Mul} &= \left\lceil \log_2(t-1) \right\rceil + \mathrm{wt}(t-1) + d - 2, \\ \mathrm{Rot} &= \left\lceil \log_2 M \right\rceil + \mathrm{wt}(M) - 1, \\ \mathrm{Frob} &= d - 1, \end{aligned}$$

This requires a multiplicative depth of $\lceil \log_2(t-1) \rceil + \lceil \log_2 d \rceil$ (independent of M). By contrast, the vanilla field-norm approach EQ_{t^d} adds an extra $+\lceil \log_2 M \rceil$ to the depth and incurs additional ciphertext–ciphertext multiplications proportional to the binary weight of M [60, Tab. 2]. Thus, the randomized variant strictly reduces ct×ct multiplications and eliminates the M-term in depth, at the cost of a negligible (tunable) soundness error and a few moderate operations.

(C) Bitwise *XNOR+AND* (binary-circuit equality). This route implements equality at the bit level: compute a per-bit XNOR and then AND across all μ=k bits via a balanced product. Over a binary plaintext, the per-bit XNOR is linear, and the equality on μ bits is

$$\mathrm{EQ}_{\mathrm{bit}}(x, a) \;=\; \prod_{i=0}^{\mu-1} \big(1 + x_i + a_i\big)\,,$$

which yields multiplicative depth $\lceil \log_2 \mu \rceil$ for the AND tree. In the more common odd-modulus setting (BGV/BFV with batching), a standard "square trick" realizes the per-bit XNOR with one multiplication,

$$\mathrm{EQ}_{\mathrm{bit}}(x, a) \;=\; \prod_{i=0}^{\mu-1} \big(1 - (x_i - a_i)^2\big)\,,$$

so the depth becomes $1 + \lceil \log_2 \mu \rceil$. With SIMD packing and Galois automorphisms, the μ-fold product per record can be evaluated as a slotwise balanced tree using rotations, so each ciphertext pays only $\lceil \log_2 \mu \rceil$ ct×ct multiplications (in parallel for all packed records) rather than $\mu-1$ [101, 102, 187].

Parameterization and layout. Bit-slicing a k-bit value consumes k slots per record; with s slots per ciphertext, a single ciphertext holds s/k records and the input size is $c_{\mathrm{bit}} = \lceil nk/s \rceil$ ciphertexts. Per ciphertext, the multiplication count is $\approx \lceil \log_2 \mu \rceil$ for the reduction tree (plus μ squarings if the square trick is used; these are ct×ct squarings and add one to the depth budget). Rotations are required at each tree level (typically $O(\lceil \log_2 \mu \rceil)$ per ciphertext), and key-switching costs should be provisioned accordingly. If values are provided in small digits (e.g., bytes), one can apply the XNOR product per digit (depth $1 + \lceil \log_2 b \rceil$ for b=8) and then AND across the k/b digits with an additional $\lceil \log_2(k/b) \rceil$ depth, which often improves packing and reduces rotations.

Pros/cons. Bitwise XNOR+AND offers shallow, data-width–driven depth and composes cleanly with further Boolean logic; it is a canonical equality gadget in HE search frameworks [101, 102, 187]. The main trade-offs are (i) the need for bit/digit decomposition (either at input time or via a costly server-side procedure), (ii) several rotations per tree level, and (iii) higher slot pressure: k slots per record unless a digitized layout is used.

(E) Bitwise *XOR+OR* addition-only equality. SCAM [50] replaces XNOR+AND by a weighted XOR–OR that uses only additions under LWE/FHEW-style primitives, eliminating ciphertext–ciphertext multiplications. While appealing for hardware CAMs, it lies outside our ring-SIMD setting and reveals magnitude unless post-processed; we treat it as out of scope for our BGV/BFV focus.

Guidance. For single-attribute filters over large arrays with tight ciphertext budgets, integer-wise *Subtract+FLT* (A) or its digit variant often dominate: one value per slot, no rotations, depth $\approx \log_2(p-1)$ (or $\log_2(p-1) + \log_2 d$) [171, 218].

When attributes are symbol-aligned (e.g., bytes), the $\mathbb{F}_{2^\ell}$ *Frobenius+MultTree* (C) yields very low depth (e.g., 3 for ℓ=8) with clean SIMD. If equality must compose with several subsequent predicates, *XNOR+AND* (B) on small digits remains attractive due to its shallow, data-width-driven depth and fine control over the rotation/multiplication trade-off [102, 187].

If a constant-depth kernel is required, the randomized family (D) achieves that at negligible error and is backed by code [60]. Comparative discussions such as [188] corroborate these trade-offs: word-wise (field/FLT) methods typically excel when packing is paramount, while bit/digit methods shine when depth is the bottleneck and values are already small.

4.3. Contact Tracing against the Coronavirus by Bridging the Centralized–Decentralized Divide for Stronger Privacy

This section is from a joint work of the author with Dr. Alexander Koch, Dr. Gunnar Hartung, Felix Dörre, Prof. Jörn Müller-Quade, and Prof. Thorsten Strufe.

We published our work on eprint, which later evolved into the comprehensive version documented in [47]. A condensed version was also presented in the proceedings of Asiacrypt 2021 [48]. We will indicate in the relevant sections which contributions are the sole research efforts of one or more co-authors.

During the early stages of a pandemic, when a vaccine is not yet available, one of the most important interventions to contain its spread, is – besides the reduction of face-to-face encounters in general – the consequent isolation of infected persons, as well as those who have been in close contact with them ("contacts") to break the chain of infections. In phases with low case numbers of the SARS-CoV-2 pandemic, contact tracing has been used to keep case numbers in check (for a longer time). However, tracing contacts manually (by interviews with infected persons) is not feasible when the number of infections is too high. Hence, more scalable and automated solutions are needed to safely relax restrictions of personal freedom imposed by a strict lockdown, without the risk of returning to a phase of exponential spread of infections. *Digital contact tracing* using off-the-shelf smartphones is used as an additional measure that is more scalable, does not depend on infected persons' ability to recall their location history during the days before the interview, and can even track contacts between strangers.

In many digital contact tracing protocols, e.g. [18, 19, 26, 65, 87, 97, 107, 238, 251, 252, 287], users' devices perform automatic proximity detection via short-distance wireless communication mechanisms, such as Bluetooth Low Energy (BLE), and jointly perform an ongoing cryptographic protocol which enables users to check whether they have been colocated with contagious users. However, naïve designs for digital contact tracing pose a significant risk to users' privacy, as they process confidential information about users' location history, meeting history, and health condition [193].

This has sparked a considerable research effort to design protocols for privacy-preserving contact tracing, most of which revolve around the following idea: Participating devices continuously broadcast ephemeral, short-lived pseudonyms and record pseudonyms broadcast by close-by devices. When a user is diagnosed, she submits either all the pseudonyms her device used while she was contagious or all the pseudonyms her device has recorded (during the same period) to a server. The first approach is the *upload-what-you-sent* paradigm, while the second is called *upload-what-you-observed* paradigm. Users' devices are then either actively notified by the server, or they regularly query the server for pseudonyms uploaded by infected users.

Some of the designs that received the most attention are the centralized PEPP-PT proposals [239, 240], as well as the more decentralized approach of [87] and DP3T [287], which served as sketches for the subsequently proposed Apple/Google-API (GAEN) [19]. While the "centralized" approaches of PEPP-PT do not provide any privacy guarantees towards the users against the central server infrastructure [3, 4] (unless they are augmented by, e.g. mix-nets), the DP3T approach [287], as well as the similar protocol by Canetti, Trachtenberg, and Varia [87], expose the ephemeral pseudonyms of every infected user, which enables her contacts to learn whether she is infected. A detailed comparison is given in [150].

We argue that both, *protection against a centralized actor*, as well as *protection of infected users from being stigmatized for their status*[16], is important for any real-world solution. By specifying a protocol that achieves both of these goals and detailing the corresponding design choices, we aim to contribute to the ongoing discussion on privacy-preserving digital contact tracing.

4.3.1. Contribution

We propose a strong and encompassing simulation-based security notion via an ideal contact tracing functionality (in section 4.3.8) that allows us to capture the following privacy and security guarantees.

- It makes the exact leakage an attacker would gather explicit. This leakage can be described by a partially anonymized, partially pseudonymized contact graph (described and motivated in detail in section 4.3.8 and fig. 4.19), a list of positively tested and corrupted participants, and their warning status. This (minimal) leakage is inherent to BLE-based contact tracing schemes.
- It captures that the locally exchanged identifiers do change quickly (each "short-term epoch") in an unlinkable fashion, but the time of an encounter causing a warning can only be narrowed down on a more coarse-grained timescale. In other words, while observed identifiers change, e.g. every 15 minutes, a warning does only give away the day (or another globally-fixed "long-term epoch") of the encounter.

[16] See `https://coronadetective.eu` (last-accessed: 13.11.2025) for a service that detects the contacts that caused a warning for DP3T-based approaches.

- It captures the worst-case guarantees in the sense that our guarantees hold, no matter how history unfolds, people meet, move and get infected, i.e., the environment can fully control the (directed) contact graph and infection status per short-term epoch.
- It provides guarantees against not being warned despite a (BLE-detectable) risk contact with an honest user (false negatives). For this, we assume that an attacker does not jam any local communication.
- It provides guarantees against being warned without a corresponding risk contact (false positives), *unless* the user was in proximity to a corrupted user *and* a corrupted user is infected or in proximity to an infected user. (This restriction is necessary, as in any protocol not protecting against malicious replays of proximity beacons, any attacker can cause a false positive under these conditions. However, protecting against replays would require processing time and location information, which is deemed undesirable.)

As a second part, we specify a privacy-preserving contact tracing protocol that achieves this security notion. It follows the upload-what-you-observed paradigm and achieves its goals by the following mechanisms:

- We split up the identifiers into short-lived *public identifiers* (pids) used for broadcasting, and longer-lived secret identifiers used for querying for warnings (cf. section 4.3.4.1).
- We employ a strict server separation concept, where the servers (for uploading the lookup table for this split-up identifiers, for matching, and for warning queries) carry out different functions (cf. section 4.3.4.2). For reasons of complexity reduction, the ideal functionality in the main body does not include server corruptions.
- We employ strong, but anonymous anti-Sybil protections coupled to, e.g., an SMS challenge, to ensure that the guarantees cannot be circumvented by registering multiple Sybil identities (cf. section 4.3.4.3).

Additionally, we argue that our protocol is similar in efficiency to DP3T, on the side of the smartphone used, see our efficiency analysis on p. 126. While our protocol was designed with the current COVID-19 pandemic in mind, note that it can easily be generalized to perform contact tracing for other transmissible diseases and enable an effective containment in case a new virus is about to hit a population without any immunity from prior exposition.

Outline We define our informal security model for BLE-based contact tracing in section 4.3.3, the formal version is given in section 4.3.8. For this protocol, section 4.3.4 proposes a number of core security mechanisms in a modular way, which are applied to obtain our overall protocol presented in section 4.3.6. In Section 4.3.7 we discuss the computational requirements of our protocol to the infrastructure components needed for the execution. A security and privacy analysis of the protocol follows in sections 4.3.9 and 4.3.10.

4.3.2. Related Work

Canetti et al. [87] mention an extension of their protocol using private set intersection protocols in order to protect the health status of infected individuals. However, it is unclear how feasible such a solution is with regard to the computational load incurred on both, the smartphone and the server, cf. [133, P3]. Whereas DP3T [287] claims that protecting the infection status of individuals in decentralized protocols is impossible by [2, IR 1] and therefore does not address further countermeasures.

Chan et al. [97, Sect. 4.1] include a short discussion of protocols in the upload-what-you-observed paradigm, and propose a form of rerandomization of identifiers at the side of the smartphone. In this protocol, a user downloads all published identifiers and checks whether they are a rerandomization of their own identifier (requiring one exponentiation). Hence, this approach puts a regular heavy computation cost on the user's device, and is likely not practical. Bell et al. [26] propose a solution for digital contact tracing based on homomorphic equality tests, aimed at protecting the infection status. However, there the central server learns the full contact graph for infected and non-infected users alike, as all users periodically upload their observations.

Besides BLE-based approaches, there are also proposals that use GPS traces of infected individuals to discover hot spots as well as colocation, such as [34, 149]. However, there is a consensus that GPS-based approaches do not offer a sufficient spatial resolution to estimate the distance between two participants with sufficient precision.

The protocols of Garofalo et al. [155], and DESIRE [95] (another hybrid approach, constituting concurrent work), broadcast public keys and compute Diffie-Hellman shared secret upon receiving a broadcast. Both are very similar

to a proposal from Cho, Ippolito, and Yu [107]. Both constructions compute two separate hashes of a shared secret, which constitutes an encounter, and use one for reporting contacts at risk and another one for querying their status. An advantage of registering an encounter by computing a shared secret from a Non-Interactive Key Exchange is the protection against certain kinds of replay attacks as observing a public key is not enough for impersonation. The main disadvantage, is that a public key does usually not fit into a single advertisement packet and therefore additional workarounds are necessary. Also, the security model of DESIRE is different from ours, e.g. if two corrupted users would like to know whether and when they met the same honest non-infected user, they could cooperate with the DESIRE server (which can link all encounter tokens of a user together, because a user has to upload all of them at once when querying for a warning) to link both encounters. Garofalo at al. introduce a Central Health Authority server, and a matching server that has some similarities to our server pipeline.

Instead of broadcasting large public keys, the protocol Pronto-C2 by [21] broadcasts addresses, where the public keys can be retrieved from. This requires the public keys to be anonymously uploaded in advance, which is similar to the submission routine in our protocol. Pronto-C2 separates the task for authenticating app requests from the central server and leaves the task for matching and risk-computation to the smartphone, which might incur a significant workload on the smartphone. On the other hand, our protocol utilizes a dedicated party for every privacy-sensitive task, i.e. submission, matching, warning and registering, and leaves only the task of risk-computation to the smartphone. The interested reader is referred to [292] for a general discussion on hybrid approaches.

The protocol Epione by [285], as well as the protocol Catalic by [134] make use of private set intersection to improve on the privacy side.

Canetti et al. [88] introduce two protocols and also feature a universal composability (UC) modeling of contact tracing functionalities, which constitutes concurrent and independent work. While their modelling takes broad strokes by employing a global functionality for interacting with the physical world, via a set of allowable measurement functions and faking functions to the physical world, we specifically model the aspect of people being in relevant closeness to each other using a contact graph, and can hence model the leakage and e.g. relay attacks by certain operations on the graph – yielding a more easy-to-handle criterion. Moreover, only an extension of one of their

protocols, called CertifiedCleverParrot, incorporates anti-Sybil protections, but this is not modeled and proven secure in their UC setting. For an alternative modelling and analysis of security notions using game-based definitions, such as forward security, see the concurrent work of Danz et al. [124].

4.3.3. Security Model

Our main goals are *privacy*, i.e. limiting disclosure of information about participating individuals, and *security*, i.e. limiting malicious users' abilities to produce "wrong protocol outcomes", such as being warned without a (BLE-detectable) risk contact (false negatives), or not being warned despite a risk contact (false positives). For privacy, we consider the following types of private information:

- where users have been at which point in time,
- whom they have met (and when and where),
- whether a user has been infected,
- whether a user has received a warning because she was colocated with an infected user.

We have a precise analysis of which of these goals are achieved under which conditions, and refer to sections 4.3.9 and 4.3.10 for details. We refer the interested reader to [193] for a systematization of different privacy desiderata.

Modeling Time We assume time is divided into disjoint, consecutive intervals called *epochs* (or *short-term epochs*). A *long-term epoch* is the union of a fixed number of consecutive short-term epochs. Again, all long-term epochs are disjoint and consecutive. In the following, we assume each short-term epoch corresponds to a 15 minute interval, and each long-term epoch corresponds to a day. Hence, there are 96 short-term epochs in a long-term epoch, and a tuple from $\mathbb{N} \times \mathbb{Z}_{96}$ specifies a short-term epoch. (These durations are parameters, but for concreteness we describe our protocol with these parameters fixed.)

Allowing the Distinguisher to Define Reality We let the distinguisher $\mathcal{Z}$ define the physical reality for each epoch $t \in \mathbb{N} \times \mathbb{Z}_{96}$, i.e. who meets whom (defined by a contact graph G_t) and who is infected (a set of parties $\mathcal{P}_{infected,t}$). Nodes in G_t correspond to participating parties, and G_t contains an edge (P_1, P_2) if P_2 registered a contact with P_1. Since who registered a contact with whom might not be a symmetric relation (e.g. due to noise in the wireless signal), each G_t is a *directed graph*.[17] (We do not impose any restrictions on G_t or $\mathcal{P}_{infected,t}$, the environment may set these arbitrarily, even in ways that would be impossible in the physical world.) The distinguisher $\mathcal{Z}$ defines these values by sending them to a party P_{mat} (named after the ideal functionality $\mathcal{F}_{\text{mat}}$ as explained below). Each such input marks the beginning of a new short-term epoch. In the ideal experiment, this is a dummy party which forwards these inputs to $\mathcal{F}_{\text{CT}}$. In the real experiment, P_{mat} sends $\mathcal{P}_{infected}$ to $\mathcal{F}_{\text{med}}$ and G to $\mathcal{F}_{\text{mat}}$. This *hybrid* (i.e. ideal, but used in the real world to abstract from a realization of it) functionality $\mathcal{F}_{\text{mat}}$ represents the "world state" or "material world"[18], including a representation of who met whom (controlable by the environment), and a synchronized "epoch-wise" clock. This functionality is used for local broadcast and to decide which participant receives a particular public identifier pid. Here, Servers constitutes a set of centralized servers, see section 4.3.4.2.

As mentioned above, the incorruptible party P_{mat} just forwards the contact graph G and the set of infected parties $\mathcal{P}_{infected}$ to the relevant functionalities $\mathcal{F}_{\text{mat}}$ and $\mathcal{F}_{\text{med}}$ (which represents the medical professional that is informed about who is infected, and will be given in section 4.3.6 on p. 127), respectively.

Communication Channels Channels between the parties, functionalities and the servers are assumed to be confidential and authenticed (in the fitting direction). We assume the attacker does not jam any wireless communication between honest parties. (The distinguisher $\mathcal{Z}$ can emulate a suppression of broadcasts by leaving out edges in the contact graph.)

When a user, e.g. uploads data used in the protocol that should not be linked to the person (e.g. public or secret identifiers), the server can easily link

[17] This captures a relaxed notion of "proximity", as high-gain antennas could be used to register a contact, although not physically being in proximity.

[18] Internally, the author(s) humorously prefer to read the name of $\mathcal{F}_{\text{mat}}$ as "the matrix".

$\mathcal{F}_{\text{mat}}(\mathcal{P}, P_{\text{mat}}, \text{Servers})$

State:

- Current contact graph $G = (\mathcal{P}, E)$
- Current time $e = (e_{lt}, e_{st}) \in \mathbb{N} \times \mathbb{Z}_{96}$.

Set Neighborhood:

1. Receive and store directed contact graph $G = (\mathcal{P}, E)$ from party P_{mat}.
2. Increment e_{st} (in $\mathbb{Z}_{96}$). If $e_{st} = 0$, increment e_{lt} and send (*newLongTermEpoch*) to all servers, and then to all parties except P_{mat}.

Receiving Broadcasts:

1. Receive (pid) from a participant P, where pid is a public identifier.
2. Send (pid) to all P' with $(P, P') \in E$.

Figure 4.8.: Ideal Functionality $\mathcal{F}_{\text{mat}}(\mathcal{P}, P_{\text{mat}}, \text{Servers})$

Protocol of P_{mat} in the Real Setting

Update Neighborhood and Infections:

1. Receive a contact graph G and a set of infected parties $\mathcal{P}_{infected}$ from the environment $\mathcal{Z}$ as input.
2. Send G to $\mathcal{F}_{\text{mat}}$.
3. Send $\mathcal{P}_{infected}$ to $\mathcal{F}_{\text{med}}$.

Figure 4.9.: Protocol of P_{mat} in the Real Setting

these pairs with communication metadata (such as the user's IP address), which might be used to ultimately link this data to a specific individual. We therefore use an anonymous communication channel for all communication with the servers. In practice, one can communicate via publicly available

proxies that are managed by operators separate from the protocol servers. Alternatively, one might also employ the TOR onion routing network [284]. (We analyze the load that would be placed on TOR in Section 4.3.7.)

Corruption Model In the formal modeling and our security proofs – to keep the complexity of the description and proofs manageable – centralized servers are perfectly trusted. However, the protocol was designed in a way that the information leakage to the servers is still acceptable in the case of a passive (honest-but-curious) server corruption, as will be explained in section 4.3.9.1.

Regarding the users, we do only consider static corruptions, i.e. corruptions that happen at the beginning of the protocol execution. We do not distinguish between "the attacker" and corrupted, malicious, or compromised parties.

Modeling Medical Professionals Furthermore, we trust medical professionals to not disclose data regarding the users who are under their care, as is their duty under standard medical confidentiality. This is abstracted by introducing a hybrid functionality $\mathcal{F}_{\text{med}}$, which represents medical professionals who are aware about the infection status of all users. $\mathcal{F}_{\text{med}}$ is defined in section 4.3.6 on p. 127.

4.3.4. Core Security Mechanisms

We start by giving a relatively generic, abstract template of contact tracing protocols, which are characterized by send-what-you-observed upon infection. This allows us to put our core security mechanisms in context and serve as a starting point for describing them.

Generation of "Random" Identifiers. For every time period t, the user's device generates an identifier pid_t. (These identifiers can look uniformly random and are computationally unlinkable, unless they incorporate additional time/location information for replay/relay protections.)

Broadcasting and Recording. During the time period t the identifier pid_t is repeatedly broadcast so nearby participants can record it, together with the date/time (maybe involving additional postcomputation before storing).

Warning Co-located Users. When a user is tested positive, one extracts a list of all *recorded* pid' from the infected user's device (assuming that old ones are periodically deleted). The user is then given a TAN code that she can use to send this list to a central server. The server marks the respective pids as potentially infected, and then allows users to query for a given pid, answering whether it is marked as potentially infected.

4.3.4.1. Splitting of Identifiers

We propose to use, instead of just one public identifier pid that is used for both, broadcasts and warning queries, two versions of identifiers: public identifiers pid that are used for broadcasting, and a secret identifiers sid which are used to query the server for warnings. The server internally keeps a table linking sids to pids, where users can submit new entries to. This split-up of identifiers achieves better privacy, because malicious users cannot just use public identifiers they have observed to query the server for the warning status of the pids' owners.

Generation of "Random" Identifiers. For every time period t, the device generates pid_t, sid_t in a such way that one cannot efficiently derive sid_t from pid_t. Moreover, given a set of pids which are either all from the same user, or all from different users, it should not be possible to distinguish which is the case. Finally, we require that only the user to whom these ids belong can submit them, e.g. by her knowing a preimage that is used to generate both in tandem and also submitting the preimage.

Broadcasting and Recording. Proceeds as above.

Warning Co-Located Users. When an infected user sends a list of all recorded pid' as above, the server looks up the respective sids in his database of (sid, pid) tuples and marks them as potentially infected. The server then allows users to query for sids, answering whether they are marked as potentially infected.

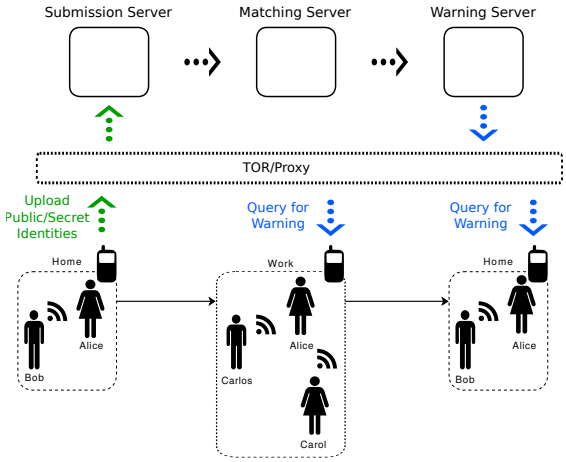

Figure 4.10.: Overview of the application's infrastructure. The figure depicts different possible scenarios: In the morning, Alice uploads her daily public/secret identifiers to the submission server, and periodically queries the warning server for warnings. Throughout the day, while she is in proximity to Bob, Carlos and Carol, the application exchanges public identifiers with their phones.

4.3.4.2. Splitting-Up the Server into a Pipeline

The change introduced in section 4.3.4.1 allows to split the process of warning co-located users into three tasks for three non-colluding servers, the submission server, the matching server, and the warning server:

- The *submission server* collects the uploaded secret and public identifiers from different users (more precisely, it receives sid and the seed for the PRG) and then computes the $(\mathrm{sid}'_i, \mathrm{pid}_i)$ pairs using the PRG with the given seed. It rerandomizes the sid'_i values another time with fresh, non-reproducible randomness (obtaining sid''_i), and stores $(\mathrm{sid}''_i, \mathrm{pid}_i)$ for a short period of time. When the submission server has a sufficient number of submissions, it shuffles them and sends them to the matching server. For ease of notation, we assume that this transaction happens at the beginning of the next long-term epoch. (We assume that enough users participate, for the batching to make sense.)
- The *matching server* collects the $(\mathrm{sid}''_i, \mathrm{pid}_i)$ pairs and stores them. Upon receiving the pids recorded by the devices of infected users,

which we call a *match request*, the matching server looks up the respective sid_i''s of all potentially infected users and sends them to the warning server.

- The *warning server* decrypts sid_i'' to recover $\mathrm{wid} \coloneqq DEC_{\mathrm{sk}_W}(\mathrm{sid}_i'')$ for all potentially infected users. It then allows to query for warning ids by the users, which we call *warning query* in the following.

For illustration, see fig. 4.10. We assume all communication between the servers uses confidential and authenticated channels. Section 4.3.9.1 contains a privacy analysis in case of compromised, honest-but-curious and partly colluding servers.

4.3.4.3. Protecting from Encounter-wise Warning Identifiers and Sybil Attacks

Our measures of having a lower resolution for the secret/warning identifiers are not yet sufficient to hide the infection against the following, more motivated attack: An attacker that is able to upload an unlimited number of sid and PRG seed values to the submission server, can change to a set of pids that belong to a different warning identifier, after each short-term epoch. Upon warning, the attacker can then deduce which of the warning identifiers have been warned, and from that deduce the exact short-term epoch the encounter happened. A simple rate-limiting on the side of the app is ineffective against malicious attackers, and a simple traffic-based rate-limiting on the side of the servers per app instance is not possible due to the anonymized communication.

Sybil Attack Moreover, the above attacker can run a so-called *Sybil attack*, i.e. creating multiple (seemingly) independent app instances. Hence, we aim to prevent this type of attack and ideally to ensure a limitation of uploads to the submission server to one per user (identifier) per day. For this, it is helpful to use a users identifier that is difficult to obtain in larger numbers, to force the adversary to invest additional resources for spawning Sybil instances. While there are a number of solutions, for concreteness, we propose to bind each app instance to a phone number (as the aforementioned user identifier) and require a registration process using an SMS challenge. (Note that this

approach does not prevent an attacker from performing a Sybil attack on lower scale, as the attacker might own multiple phone numbers.[19])

Countermeasures Binding an app to an identifiable resource (such as a valid phone number) while ensuring the user's anonymity, requires a bit of care. For this, we use the periodic n-times anonymous authentication scheme from [75]. Such a scheme Σ_{tok} allows users $\mathcal{U}$ to obtain e-token dispensers from the issuer $\mathcal{I}$, in our case the registration server, and each dispenser may only issue up to n anonymous and unlinkable e-tokens. For more than n e-tokens a user is required to obtain a fresh dispenser. A designated verifier V, in our case the submission server, is then able to verify each e-token. As each user is allowed to upload only once a day the number of dispensable tokens is set to $n = 1$, in which case the scheme from [75] solves exactly the problem from [119]. Nevertheless we decided to use the terminology from [75] for the sake of generality. In the following we show a slightly simplified version of the fully fledged definition from [75] as we do not require all of the features. The formal definition is given in Definition 28. In our setting, we choose $n = 1$ and choose as time period the long-term epoch period, i.e. the user can obtain one "e-token" per long-term epoch to upload a new sid and PRG seed to the submission server. The submission server validates the "e-tokens" and only accepts submissions with valid tokens while checking for double-spending. The token dispenser is then issued to the user during a registration process, which uses the aforementioned SMS challenges.

Registration Modeling Formally, we define the hybrid functionality $\mathcal{F}_{\text{reg}}$, which represents the party towards which parties run the registration protocol, and which keeps a list of registered parties, and is given below. This is e.g. for obtaining a token dispenser to perform the regular uploads. To keep the model simple, we do not incorporate SMS challenges into $\mathcal{F}_{\text{reg}}$. (An SMS challenge, as well as the upload TAN, might be modeled via an authenticated channel from the party, for which an adversary can break authentication by guessing. See [9] for a formalization).

[19] One might use remotely verifiable electronic ID cards instead.

$\mathcal{F}_{\text{reg}}(\mathcal{P})$

State:

- Set of registered parties and their public keys as pairs $\mathcal{RP}$.
- Issuer secret and public key for e-token dispensers $(\text{sk}_{\mathcal{I}}, \text{pk}_{\mathcal{I}})$

Registering a Party:

1. Upon $(\textit{register}, \text{pk}_{\mathcal{U}})$ from party P:
   if P is not already in a pair in $\mathcal{RP}$, store $(P, \text{pk}_{\mathcal{U}})$ in $\mathcal{RP}$, else abort.
2. Issue a new e-token dispenser for P acting as $\mathcal{U}$ by participating as $\mathcal{I}$ in the protocol $\textit{OBTAIN}(\mathcal{U}(\text{pk}_{\mathcal{I}}, \text{sk}_{\mathcal{U}}, 1), \mathcal{I}(\text{pk}_{\mathcal{U}}, \text{sk}_{\mathcal{I}}, 1))$.

Figure 4.11.: Hybrid Functionality $\mathcal{F}_{\text{reg}}(\mathcal{P})$

4.3.5. Post-Quantum Contact Tracing

In our original protocol in [47, 48], we designed the protocol using cryptographic primitives, allowing it to be constructed based on various assumptions underlying these primitives. Initially, we proposed using some constructions that are not quantum-safe. Here, we discuss how this protocol can be adapted to achieve partial quantum-safe security and outline remaining open problems.

Re-Randomizable Encryption We initially proposed employing the standard ElGamal scheme for instantiating a re-randomizable encryption scheme. This choice was due to ElGamal's conceptual simplicity and all authors being familiar with. However, our original protocol is not limited to this scheme, and any other IND-CPA-secure re-randomizable PKE scheme could be used.

Currently, standardized quantum-safe schemes are typically analyzed in their ultimate form, securing against chosen-ciphertext attacks (CCA). To our knowledge, no theorems in the literature confirm that the optimizations of these schemes, in their IND-CPA variants, would carry over to a possible re-randomization procedure. Conceptually, the IND-CPA part of ML-KEM [226]

(formerly known as CRYSTALS-KYBER) is similar to ElGamal and is often referred to as "Noisy ElGamal". ML-KEM also admits restricted homomorphic additions, similarly as McEliece does. However, its correctness depends largely on parameter selection and error growth. Since ML-KEM is highly optimized, adapting it to meet re-randomization requirements according to Definition 15 and Definition 16 while maintaining these optimizations is left for future work.

We propose using a simplified variant of the BGV scheme [67] with the simplest possible parameterization (see, for example, [13]). In this case, re-randomization can be achieved through the homomorphic addition of public encryptions of 0, making it straightforward. Correctness and security proofs follow directly from the IND-CPA security and BGV's homomorphic properties. Additionally, BGV supports ciphertext packing and SIMD execution, making future adaptations of these concepts to the matching procedures in our protocol a promising direction for future work.

E-Token Dispenser We also proposed employing an E-token dispenser construction from [75], which relies on bilinear pairings and variants of the DDH problem, and is therefore not quantum-safe. Fortunately, the authors' formulation of such a dispenser (cf. Definition 28) is agnostic to specific constructions, suggesting that many different constructions might be possible, some of which could be based on quantum-safe assumptions. However, to the best of our knowledge, only few such constructions currently exist, such as [272], which, at the time of writing, still requires additional scrutiny.

4.3.6. Separated Duty Contact-Tracing Protocol

We can now describe the full protocol. For this, let n denote the security parameter. We assume a IND-CPA secure, rerandomizable public key encryption scheme $(GEN, ENC, DEC, ReRand)$ according to Definition 15 and Definition 16. Let PRG be a secure pseudorandom generator according to Definition 19, and H be a one-way function. Finally, let $\Sigma_{\text{tok}} = (GEN_{\mathcal{I}}, GEN_{\mathcal{U}}, OBTAIN, SHOW, IDENTIFY)$ be an anonymous e-token dispenser scheme as in Definition 28. First, we will describe the overall protocol structure, followed by the full protocol description for all parties, which will be used in the real/ideal security analysis. For an illustration, see fig. 4.12.

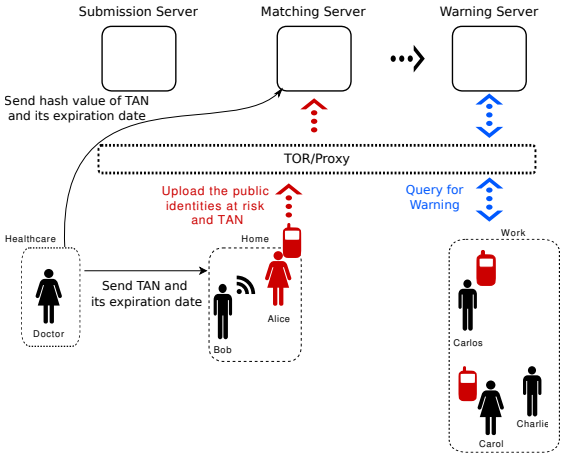

Figure 4.12.: Information flow upon issuing a warning. When the doctor is informed about a positive test, she generates a new TAN and sends it to the matching server and then communicates it to positively tested Alice. Then, using this TAN, Alice uploads all public identifiers she observed during her infectious period. The application regularly queries for its warnings to its the warning server. In the case of Carlos and Carol, who have been in contact with Alice in fig. 4.10, this check will turn out to be positive.

App Setup. When the proximity tracing software is first installed on a user's device, for anti-Sybil measures as described in section 4.3.4.3, the application proves possession of a phone number (e.g. via an SMS challenge) and obtains an e-token dispenser.

Creating Secret Warning Identifiers. For each long-term epoch, the application generates a random *warning identifier* $\mathsf{wid} \leftarrow\!\!\$\ \{0,1\}^n$

Deriving Public Identifiers. For each warning identifier wid, the app computes $\mathsf{sid} \coloneqq \mathit{ENC}(\mathsf{pk}_{\mathcal{W}}, \mathsf{wid})$, where *ENC* is the encryption algorithm of a rerandomizable, IND-CPA-secure public-key encryption scheme, and $\mathsf{pk}_{\mathcal{W}}$ is the warning server's public key. Additionally, the app chooses a random seed $\leftarrow\!\!\$\ \{0,1\}^n$ (*rerandomization seed*) per warning identifier. The app (interactively) presents an e-token τ to the submission server via an anonymous channel, and uploads $(\mathsf{sid}, \mathsf{seed})$ to the submission server via the same channel. If the e-token is invalid (or the server detects double-spending of this

e-token), the server refuses to accept (sid, seed). Both the submission server and the app compute 96 rerandomization values $r_1, \ldots, r_{96} = \mathsf{PRG}(\mathsf{seed})$, and rerandomize sid using these values, obtaining $\mathsf{sid}'_i \coloneqq \mathit{ReRand}(\mathsf{sid}; r_i)$ for $i \in \{1, \ldots, 96\}$. The ephemeral public identifiers of the user are defined as $\mathsf{pid}_i \coloneqq \mathsf{H}(\mathsf{sid}'_i)$ for all i. The app saves the public identifiers for broadcasting during the day of validity of wid. The submission server rerandomizes each sid'_i again (using non-reproducible randomness) to obtain sid''_i and stores the $(\mathsf{sid}''_i, \mathsf{pid})$ pairs and forwards them to the matching server.

Broadcasting and Recording. During each time period i, the device repeatedly broadcasts pid_i. When it receives a broadcast value pid' from someone else, it stores (e_{lt}, pid'), where e_{lt} is the current long-term epoch. Every long-term epoch, the device deletes all pid's that are old enough to no longer be epidemiologically relevant.

Performing Contact Matching. The matching server maintains a list of hash values of all TANs issued by medical professionals and all tuples it has received from the submission server, deleting each tuple after three weeks.[20] Then a user submits a list of public identifiers together with a valid TAN, the matching server marks the TAN's hash value as invalid by deleting it from its list. The server looks up the corresponding secret identifiers sid and sends them to the warning server.

Sending a Warning. When a user is tested positive, the medical personnel generates a TAN and registers it at the matching server. The user collects a list of public identifiers pid' that have been received by his device while the user was likely infectious, and sends this list together with the TAN to the matching server. The matching server checks if the TAN is valid and removes it from the pending list. Subsequently, the according sid'_i are retrieved, rerandomized into sid''_i and sent to the warning server. The warning server decrypts the received $\mathsf{wid}_i = \mathit{DEC}_{\mathsf{sk}_W}(\mathsf{sid}''_i)$ and warns all applications that query with matching wid_is.

[20] If a user A has been in contact with an infected user B, and if B takes up to three weeks to show symptoms and have a positive test result, the data retention on the matching server is sufficient to deliver a warning to A.

Protocol of the App/Users: State, Register, Upload

State:

- Current epoch $e = (e_{lt}, e_{st}) \in \mathbb{N} \times \mathbb{Z}_{96}$. The public identifiers of the current long-term epoch $(\mathsf{pid}_j)_{j\in[1,\ldots,96]}$. Current token dispenser D. Set of recorded broadcasts of pids. Current Warning identifier wid
- Let $\mathsf{pk}_{\mathcal{W}}$ and $\mathsf{pk}_{\mathcal{I}}$ be the hardwired public key of the warning server, and e-token dispenser issuer, respectively. Let $(\mathsf{sk}_{\mathcal{U}}, \mathsf{pk}_{\mathcal{U}})$ be the generated user secret/public key pair during the registration.
- Set of earlier warning identifiers (wid, k), where k is the according long-term epoch.

Register:

1. When a new party is created by the environment, it first generates a token-dispenser secret/public key pair $(\mathsf{sk}_{\mathcal{U}}, \mathsf{pk}_{\mathcal{U}})$ and then sends $(\mathit{register}, \mathsf{pk}_{\mathcal{U}})$ to $\mathcal{F}_{\text{reg}}$.
2. Obtain a token dispenser D by participating as $\mathcal{U}$ in $\mathit{OBTAIN}(\mathcal{U}(\mathsf{pk}_{\mathcal{I}}, \mathsf{sk}_{\mathcal{U}}, 1), \mathcal{I}(\mathsf{pk}_{\mathcal{U}}, \mathsf{sk}_{\mathcal{I}}, 1))$ with $\mathcal{F}_{\text{reg}}$ acting as $\mathcal{I}$. Initialize the state and run "Upload Submission".

Upload Submission:

1. Generate fresh $(\mathsf{wid}, \mathsf{seed}, \mathsf{sid})$ and the according list of $\{(\mathsf{sid}'_j, \mathsf{pid}_j)\}_{j\in[1,\cdots,96]}$. Enqueue the current (wid, e_{lt}).
2. Submit a token by participating as $\mathcal{U}$ in $\mathit{SHOW}(\mathcal{U}(D, \mathsf{pk}_{\mathcal{I}}, e_{lt}, 1), \mathsf{V}(\mathsf{pk}_{\mathcal{I}}, e_{lt}, 1))$ to the *Submission Server*, which acts as V.
3. Send $(\mathsf{seed}, \mathsf{sid})$ over the same channel to the *Submission Server*.

Scheduled Upload:

1. Upon $(\mathit{newLongTermEpoch})$ from $\mathcal{F}_{\text{mat}}$. Increment e_{lt}.
2. Dequeue outdated wids and recorded pids. Continue as in "Upload Submission".

Figure 4.13.: Protocol of the App/Users: State, Register, Upload

Protocol of the Submission Server

State:

- Current epoch e_{lt}.
- The current batch of $\{(\mathrm{sid}_j''^k, \mathrm{pid}_j^k)\}_{j\in[1,\cdots,96]}$.

Handling Submissions:

1. Verify the token by participating as V in $\mathit{SHOW}(\mathcal{U}(D, \mathrm{pk}_I, e_{lt}, 1), \mathsf{V}(\mathrm{pk}_I, e_{lt}, 1))$.
2. Detect possible double spending.
3. Receive $(\mathrm{seed}, \mathrm{sid})$ from $\mathcal{U}$.
4. Generate $\{(\mathrm{sid}'_j, \mathrm{pid}_j)\}_{j\in[1,\cdots,96]}$ with the help of seed.
5. Rerandomize the sid'_j using fresh randomness, i.e. $\mathrm{sid}''_j = \mathit{ReRand}(\mathrm{sid}'_j)$
6. Add the generated tuples (with rerandomization) $\{(\mathrm{sid}''_j, \mathrm{pid}_j)\}_{j\in[1,\cdots,96]}$ to the batch of e_{lt}.

Forwarding Submissions:

1. Upon $(\mathit{newLongTermEpoch})$ from $\mathcal{F}_{\mathrm{mat}}$.
2. Shuffle the last batch and send the complete batch to the *Matching Server* together with e_{lt}.
3. Increment e_{lt}.
4. Create a new empty batch for the new epoch.

Figure 4.14.: Protocol of the Submission Server

Retrieving Warnings The application regularly queries the warning server for the warning identifiers it has used during the last 28 days itself. This is done via an anonymous channel with proper authentication of the warning server. If the query returns that the warning identifier has been marked as at-risk, it informs the user she has been in contact with an infected person during the long-term epoch when the warning identifier was used.

Protocol of the App/Users: Broadcast, Matching, Warning

Sending Broadcasts:

1. Upon (*sendBroadcast*) from the environment.
2. Send $(\text{pid}_{e_{st}})$ to $\mathcal{F}_{\text{mat}}$ and increment e_{st}.

Recording Broadcasts:

1. Upon (pid) from $\mathcal{F}_{\text{mat}}$.
2. Enqueue (pid, e_{lt}).

Match Request:

1. Upon (*positive*) from the environment.
2. Send (*warningRequest*) to $\mathcal{F}_{\text{med}}$.
3. Receive (tan) from $\mathcal{F}_{\text{med}}$.
4. Extract the list L of all recorded/received public identifiers from the queue.
5. Send (L, tan) to the *Matching Server*.

Querying a Warning:

1. Upon $(query, t)$ from the environment.
2. Find the corresponding wid for long-term epoch t and send (wid) to the *Warning Server*.
3. Receive bit b from the warning server.
4. Output b to the environment.

Figure 4.15.: Protocol of the App/Users: Broadcast, Matching, Warning

Hybrid Functionality of Medical Professionals The medical professional is modeled by the hybrid functionality $\mathcal{F}_{\text{med}}$, formally described in Figure 4.18, which gives out a TAN to parties which are deemed infected, as given below. In a bit more detail, $\mathcal{F}_{\text{med}}$ stores a set $\mathcal{P}_{infected}$ of infected/positively tested participants as provided by the environment $\mathcal{Z}$. If such a participant $P \in \mathcal{P}_{infected}$

Protocol of the Matching Server

State:

- The current epoch e_{lt}.
- Per long-term epoch t a set $\mathcal{B}_t$ of $(\text{sid}', \text{pid})$ pairs.
- Set of TANs of pending matching requests $T_{pending}$.

Removing Outdated Information:

1. Upon (*newLongTermEpoch*) from $\mathcal{F}_{\text{mat}}$.
2. Increment e_{lt} and delete all sets $\mathcal{B}_t$ where $0 \leq t \leq e_{lt} - 14$.

Handling Submissions:

1. Receive a set of $(\text{sid}', \text{pid})$ tuples and an epoch t from the *Submission Server* and store it as $\mathcal{B}_t$.

Preparing Match Request:

1. Receive (h_{tan}) from $\mathcal{F}_{\text{med}}$ and insert (h_{tan}, e_{lt}) into $T_{pending}$.

Handling Match Request:

1. Receive (S, tan) from party P, where S is a set of pids.
2. If there is an index $t \in \mathbb{N}$ such that there is an entry $(\mathsf{H}(\text{tan}), t) \in T_{pending}$, remove this entry from $T_{pending}$, otherwise abort.
3. Let $M \coloneqq \{(\text{sid}'_l, t_l) \colon \exists \text{pid}_l \in S, t_l \in \mathbb{N}$ such that $(\text{sid}'_l, \text{pid}_l) \in \mathcal{B}_{t_l} \wedge t_l \leq t\}$.
4. Rerandomize all the $\text{sid}'_l \in M$ from the previous step and send $\{(\text{sid}''_l \coloneqq \mathit{ReRand}(\text{sid}'_l), t_l) \colon (\text{sid}'_l, t_l) \in M\}$ to the warning server.

Figure 4.16.: Protocol of the Matching Server

Protocol of the Warning Server

State:

- The current epoch e_{lt}.
- PKE key pair $(\mathrm{sk}_{\mathcal{W}}, \mathrm{pk}_{\mathcal{W}})$.
- Set $\mathcal{WL}$ of released wids and their validity epoch t.

Removing Outdated Information:

1. Upon (*newLongTermEpoch*) from $\mathcal{F}_{\mathrm{mat}}$.
2. Increment e_{lt} and delete all $(\mathrm{wid}, t) \in \mathcal{WL}$, with $0 \leq t \leq e_{lt} - 14$.

Issuing Warnings:

1. Receive a list $\{(\mathrm{sid}_l'', t_l)\}$ from the *Matching Server*.
2. Decrypt, deduplicate and add the received warning identifiers $\{(\mathrm{wid}_l = DEC_{\mathrm{sk}_{\mathcal{W}}}(\mathrm{sid}''), t_l)\}$ to $\mathcal{WL}$.

Warning Query:

1. Receive warning identifier (wid).
2. Search all finished *epoch* for wid and return 1 if a match is found, 0 otherwise.

Figure 4.17.: Protocol of the Warning Server

requests a TAN (using *warningRequest*), $\mathcal{F}_{\mathrm{med}}$ chooses a TAN, registers its hash value with the matching server and sends it to P.

4.3.7. Efficiency

The following discussion only applies to the construction of our protocol with a variant of El-Gamal based on elliptic curves and the exact pairing based construction of the periodic n−times anonymous credential scheme from [75].

$\mathcal{F}_{\text{med}}(P_{\text{mat}}, \textit{Matching Server})$

State:

- Set of infected parties $\mathcal{P}_{\textit{infected}}$.

Set Infected:

1. Receive and store the set of infected parties $\mathcal{P}_{\textit{infected}}$ from a party P_{mat}.

Handling Warning Request:

1. Upon ($\textit{warningRequest}$) from $P \in \mathcal{P}_{\textit{infected}}$.
2. Generate $\text{tan} \leftarrow\!\!\$\ \{0,1\}^{2n}$.
3. Send (H(tan)) to the *Matching Server*.
4. Send (tan) to P.

Figure 4.18.: Hybrid Functionality $\mathcal{F}_{\text{med}}(P_{\text{mat}}, \textit{Matching Server})$

Our protocol incurs computation, communication and storage cost on the smartphone, submission server, matching server and the warning server. First of all we argue that the application on the smartphone does not incur significantly larger costs than currently deployed solutions. Computation-wise, the most expensive operations, i.e. operations needed for using the token-dispenser scheme and 96 rerandomizations, have to be performed only once a day (long-term epoch). These are 12 multi-base exponentiations in the domain group of a pairing and 23 multi-base exponentiations in the target group as was shown in [75]. The remaining computations, i.e. 96 hashes for the pids and the generation of seed, wid, sid, are cost-wise similar to currently deployed solutions for contact tracing and thus the overall battery consumption and CPU load are comparable. The application has to store a constant amount of information of several kilobytes, i.e. $28 \times \text{wid}, 96 \times \text{pid}$. The only growing variable is the set of recorded/observed pids. We argue that the number of received pids will be rather small as current studies suggest, i.e. [148]. The communication comprises several small requests a day to different servers and the broadcast/reception of a pid, which we deem overall negligible.

Next, we analyze the computational cost on the submission server. Considering that the population of the EU is approximately 448 Mio. and current experience with the German contact-tracing application CWA shows that 30% of the German population have adopted the application, we may assume for further considerations 134 Mio. users in our protocol. The submission server must perform $2 \cdot 96$ rerandomizations of the sids per day and user, which means that $2 \cdot 96 \cdot 134 \cdot 10^6 \approx 2.6 \cdot 10^{10}$ rerandomizations a day or $\approx$ 300000 a second. Using the ElGamal scheme, the dominant part of the rerandomizations are two modular exponentiations or scalar multiplications if we use the ECC variant of ElGamal. For an upper bound we may use current benchmarks for the verification algorithm of ECDSA, which has two dominant scalar multiplications on elliptic curves as well. According to [38] the verification of ecdonaldp256 on an (2018) AMD EPYC 7371 with 16×3100MHz requires 425723 cycles, which means that we are able to verify $\frac{16 \cdot 3100 \cdot 10^6}{425723} \approx 116507$ signatures a second. We argue therefore that $\approx$ 300000 rerandomizations per second is a realistic requirement and the computational load on the submission server—while undeniably high—can be handled with a realistic amount of equipment.

Next, we analyze the amount of data uploaded from the users' devices to the submission server. Our estimation shows that a daily upload by our protocol is at most 240 kbit. With 138 Mio. users the submission server has to handle 33Tbit a day. By scattering uploads across the span of the day we achieve a lower bound of 0.3Gbit/s, which we deem realistic. While the server may be able to handle this amount of requests, our protocol requires that the uploads are performed through an anonymous channel. To this end one may use TOR and we argue that the EU-wide deployment of our protocol relying on TOR is within TOR's capacities. As of 2020 the advertised bandwidth of the TOR network is approx. 500 Gbit/s and the consumed bandwidth is approx. 250Gbit/s (cf. `https://metrics.torproject.org/bandwidth.html`), which is sufficient for our 0.3Gbit/s. Another important restriction of TOR is the number of active users, which currently is around 2Mio users (cf. `https://metrics.torproject.org/userstats-relay-country.html`). If our server is able to handle 0.3Gbit/s then the amount of users served per second will be 1550, which is a rather small delta to the overall number of TOR users. The latency added by using TOR is in the magnitude of seconds and has no impact on the protocol, as a warning delivered a few seconds later is acceptable. Similar considerations can be made for the matching and the warning server.

However, the costs of computation and communication are overall smaller than on the submission server and are hence tamable in the same fashion.

4.3.8. The Ideal Functionality

A Note from the Author: *The formal modeling and analysis of the security and privacy features of the contact tracing protocol was a joint effort by the author, Dr. Alexander Koch, Dr. Gunnar Hartung, Felix Dörre, Prof. Jörn Müller-Quade, and Prof. Thorsten Strufe. The modeling and analysis underwent many revisions, and it must be acknowledged that it was primarily due to the efforts of Felix Dörre, Dr. Gunnar Hartung, and Dr. Alexander Koch that all aspects of the analysis were integrated and ultimately constituted the complete proof, as found in the proceedings of Asiacrypt 2021 [48] and on the eprint server [47]. Additionally, the security and privacy analysis of the contact graph was largely the result of Felix Dörre's own research in the final stages of a version that was subsequently published. The author contributed to early versions of the analysis, which were later significantly expanded by the co-authors. Therefore, this thesis will not include the full proof of the protocol; instead, we will focus on providing a concise analysis of its security and privacy features.*

Before we are ready to state our ideal contact-tracing functionality, let us begin with several assumptions that allow us to simplify our proof and reduce complexity:

- In this section, we assume for the sake of simplicity that the servers are incorruptible. However, we discuss security against server corruption in section 4.3.9.1, and Felix Dörre provides a corresponding strengthened ideal functionality along with an adapted simulator in the appendix of [47].
- The per-day uploads are synchronous. We assume that before any pid is broadcast, all parties have made their per-day upload.[21]
- All parties, even corrupted ones, send exactly one broadcast per epoch. (The distinguisher can emulate a single corrupted party making multi-

[21] In practice, parties can make their uploads a few days ahead of time without incurring additional risk.

ple broadcasts by using additional corrupted parties with similar/equal sets of recipients.)

- For formal reasons, parties can only perform computations and broadcasts when they receive an input. Hence, we assume the environment $\mathcal{Z}$ inputs a dummy message (*sendBroadcast*) to all honest participants at the beginning of a new epoch.
- Contacts happening on the day an infected person is uploading their list do not incur immediate warnings. These are delayed until the next long-term epoch. This is also a privacy feature, ensuring that no one can learn the time of an encounter with an infected person with precision higher than a long-term epoch.

We are now ready to describe important aspects and notions used in our ideal functionality $\mathcal{F}_{\text{CT}}$, which formalizes our security and privacy guarantees: Whenever the environment $\mathcal{Z}$ starts a new short-term epoch by sending $G_i = (\mathcal{P}, E_i)$ and $\mathcal{P}_{infected}$ to $\mathcal{F}_{\text{CT}}$ (via P_{mat}), $\mathcal{F}_{\text{CT}}$ creates two derived graphs G_i' and $(\mathcal{P}, \hat{E}_i)$. G_i' is a partially anonymized, partially pseudonymized version of G_i. We let $\mathcal{F}_{\text{CT}}$ output G_i' and $\mathcal{P}_{infected} \cap \mathcal{P}_{corrupted}$ to the simulator, hence this is the information leakage of our protocol. The edge set $\hat{E}_i$ represents who will receive warnings from whom, hence the simulator's abilities to modify $\hat{E}_i$ represent the attacker's abilities to induce and suppress warnings.

Information Leakage on the Contact Graph We now describe the anonymization and pseudonymization process for G_i' in detail, cf. steps 3 to 5 in "Set Neighborhood/Infected" below. The process is exemplified by the graphs G_t and G_t' shown in fig. 4.19 (left and middle, respectively). Nodes corresponding to uncorrupted parties are renamed to a pseudonym chosen independently for each epoch (in the example, the nodes of A and C are shown as dashed). This means that an attacker cannot re-identify participants encountered earlier and hence cannot track them over time. Edges between uncorrupted parties are removed entirely (in the example the edge (A, C) is removed), hence the attacker is completely oblivious of contacts between honest parties. Edges between corrupted parties (in the example (B, D)) are preserved without modifications, since we assume they are fully controlled by the attacker and hence the attacker is completely aware of any contacts between them. Before the pseudonymization takes place, nodes corresponding to honest receivers are duplicated for each incoming edge, leaving only the outgoing edges on the

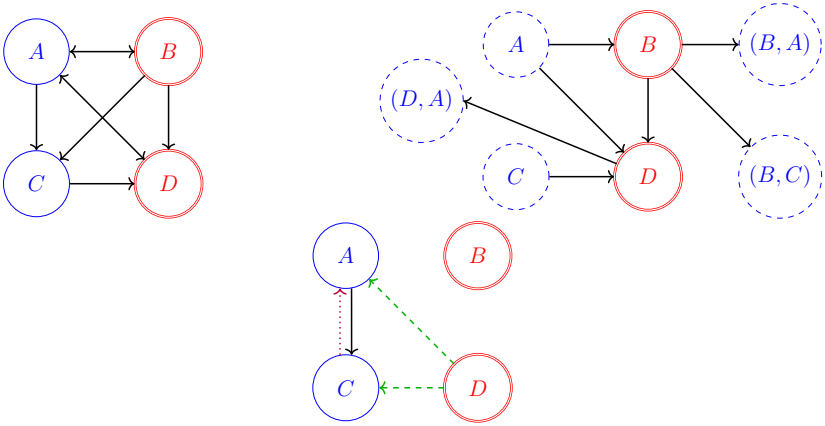

Figure 4.19.: *Left:* An example of a contact graph $G_t = (\mathcal{P}, E_t)$ with two honest parties A and C and two corrupted parties B and D. The edges indicate where a broadcast is delivered. *Middle:* The pseudonymized graph $G'_t = (Q_t, E'_t)$ of G_t as leaked by $\mathcal{F}_{\text{CT}}$ to the simulator. Dashed node borders indicate that the node name is replaced with an opaque pseudonym. *Right:* An example for $(\mathcal{P}, \hat{E}_t)$. This graph is initialized with all edges from G_t between honest parties (shown in solid black). The adversary has already inserted edges using the commands $(\textit{relay}, t, \text{pseudonymize}(C), D, B, \text{pseudonymize}((B, A)))$ as in "Replay/Relay" (shown in dotted purple) and $(\textit{sendBroadcast}, t, t, B, D)$ as in "Broadcasts From Corrupted User" (shown in dashed green). Note that warnings from honest parties are delivered *against* the direction of all the edges. So an infected A would warn C and D, an infected C would warn A and D.

original node, since corrupted senders cannot detect if they are broadcasting to the same participant. This step anonymizes edges to honest nodes. In the example the newly introduced nodes by this step are: (D, A), (B, A) and (B, C). The outgoing edges are left at their original node (for example from A), since corrupted receivers (in the example B and D) can easily detect they were in contact with the same person at approximately the same time by comparing the broadcast values. Note that this disadvantage is shared by many contact tracing protocols.

Additionally, all users of the protocol can query $\mathcal{F}_{\text{CT}}$ to check if they have received a warning, which might enable them to infer additional information about the infection status of other participants. (However, this information is inherent to all contact tracing protocols.)

Manipulation of Warnings We now discuss the attacker's ability to manipulate warnings, i.e. the attacker's options to influence $\hat{E}_i$. Note that $\hat{E}_i$ is initialized to contain all edges between honest parties (step 7 in "Set Neighborhood/Infected" below). The simulator does not have the ability to remove edges from $\hat{E}_i$, but it can introduce new edges (under certain conditions) by causing $\mathcal{F}_{\text{CT}}$ to execute "Replay/Relay" and "Broadcasts From Corrupted User".

"Replay/Relay" models a situation where a corrupted user re-broadcasts a value previously broadcast by an honest party: In this scenario – see the dotted purple edges of fig. 4.19 (right) – an honest party C broadcasted certain value during an epoch t, received by the corrupted party D. D cooperates with B and B re-broadcasts the same value in the presence of A. Hence, in our protocol, if A was infected, it would cause a warning to be delivered to C (regarding a contact during epoch t), even if those parties did not meet.

"Broadcasts From Corrupted User" models a situation, see the dashed green edges of fig. 4.19 (right), where a corrupted user B broadcasts a pid potentially uploaded by another corrupted user D, or potentially not even uploaded, yet. Broadcasting another user's pid causes warnings to be delivered to that user (D), as if D had been performing the broadcast instead of B, hence we add corresponding edges to $\hat{E}_i$. Note that the time of broadcast can be different from the long-term epoch for which the pid was (or will be) uploaded.

In addition to the ability to manipulate $\hat{E}_i$ discussed above, the attacker is able to directly send warnings in case a corrupted party is infected. $\mathcal{F}_{\text{CT}}$ enforces that the attacker can only send warnings to honest parties who have been in contact with any corrupted party during the last 14 long-term epochs and a corrupted party is infected after this encounter took place (see step 4 of "Handling Match Requests" on p. 135). The simulator is allowed to specify honest parties fulfilling these conditions (via their pseudonyms). $\mathcal{F}_{\text{CT}}$ will add these parties to the set $\mathcal{WP}$ of parties who have received a warning. When these parties next send $(query, t)$ for the corresponding long-term epoch t to $\mathcal{F}_{\text{CT}}$, $\mathcal{F}_{\text{CT}}$ will find the warning in $\mathcal{WP}$ and return 1, indicating a warning has been issued.

Having stated the formal security guarantee that we capture with this theorem, we proceed to discuss the interpretation and limitations on what we achieve exactly. For example, the extensive powers of the environment, also in determining the number and place of corrupted users, make it less clear what, e.g. our anti-Sybil protections actually achieve w.r.t. the privacy of the

$\mathcal{F}_{\text{CT}}(\mathcal{P}, P_{\text{mat}})$ – State

State:

- Current epoch $(e_{lt}, e_{st}) \in \mathbb{N} \times \mathbb{Z}_{96} =: I$.
- Set of corrupted parties $\mathcal{P}_{corrupted}$.
- Set of honest parties $\mathcal{P}_{honest} = \mathcal{P} \setminus \mathcal{P}_{corrupted}$.
- A sequence $(\mathcal{P}_{infected,i})_{i \in I}$ of sets of infected parties, i.e. the history of infected parties.
- Set of currently infected parties $\mathcal{P}_{infected}$
- A sequence of all contact graphs so-far $(G_i = (\mathcal{P}_i, E_i))_{i \in I}$, i.e. the global meeting history.
- Current contact graph $G = (\mathcal{P}, E) = G_{(e_{lt}, e_{st})}$ and its pseudonymized version $G' = (\mathcal{Q}, E')$
- Parties at risk $\mathcal{WP} \subseteq \mathcal{P} \times \mathbb{N}$, which signifies which parties have encountered a positive participant (that generated a warning) in the last 14 long-term epochs and during which long-term epochs the encounters took place.
- A sequence of edge sets $(\hat{E}_i)_{i \in I}$ on $\mathcal{P}_i$ which does some bookkeeping necessary to know who is to be warned. Let $\hat{E}$ be the edge set of the current epoch.

Figure 4.20.: $\mathcal{F}_{\text{CT}}(\mathcal{P}, P_{\text{mat}})$–State

users. We state that the e-token dispenser is meant to guarantee that not too many malicious users/Sybils exists because they are hard to create, in our formal terms this only corresponds to the guarantee that the number of daily uploads is bounded by the number of users. Hence, for real-word security we believe that we can exclude excessive Sybil attacks.

Note that this points at a larger aspect that is typical for security modeling in general, but also relevant to fully understand the scope of our modeling: Giving the environment a lot of power to shape the scenarios in which the protocols are used, is an instance of a strong worst-case modelling. By

$\mathcal{F}_{\mathrm{CT}}(\mathcal{P}, P_{\mathrm{mat}})$ – Neighborhood and Broadcast

Set Neighborhood/Infected:

1. Receive a contact graph $G = (\mathcal{P}, E)$ and a set of infected parties $\mathcal{P}_{infected}$ from party P_{mat}.
2. Add G to the global meeting history, and $\mathcal{P}_{infected}$ to the history of infected parties.
3. Set $E' = \{(P_0, P_1) \in E \mid P_0 \in \mathcal{P}_{corrupted} \vee P_1 \in \mathcal{P}_{corrupted}\}$.
4. For all $\alpha = (P_0, P_1) \in E'$ with $P_0 \in \mathcal{P}_{corrupted}$, $P_1 \in \mathcal{P}_{honest}$, replace α with $\alpha' = (P_0, \alpha)$.
5. Select a random, injective mapping $\mathrm{pseudonymize}_i \colon \mathcal{P}_{honest} \cup (\mathcal{P} \times \mathcal{P}) \to \{0,1\}^{2n}$ where $i = (e_{lt}, e_{st})$. Extend it by $\mathrm{pseudonymize}_i(P) = P$ for all $P \in \mathcal{P}_{corrupted}$. Set $E' \coloneqq \{(\mathrm{pseudonymize}_i(x), \mathrm{pseudonymize}_i(y)) \colon (x, y) \in E'\}$, i.e. rename all nodes in E'. Let Q be the set of nodes used in the set of edges E'.
6. Leak $(Q, E'), \mathcal{P}_{infected} \cap \mathcal{P}_{corrupted}$ to the adversary.
7. Let $\hat{E} \coloneqq (\mathcal{P}_{honest} \times \mathcal{P}_{honest}) \cap E$.
8. Increment e_{st} (in $\mathbb{Z}_{96}$).
9. If $e_{st} = 0$ then increment e_{lt} and delete all (P, t) pairs from $\mathcal{WP}$ where $0 \leq t \leq e_{lt} - 14$.

Send Broadcast:

1. Receive and ignore ($sendBroadcast$) from a participant P.

Broadcasts From Corrupted User:

1. Receive $(sendBroadcast, t_1, t_2, P_1, P_2)$ from the adversary, with $t_1, t_2 \in [e_{lt} - 14, e_{lt}] \times \mathbb{Z}_{96}$, $P_1, P_2 \in \mathcal{P}_{corrupted}$ (with the meaning that P_1 broadcasts in the name of (i.e. the pids registered by) P_2).
2. For each $(P_1, x) \in E_{t_1}$, add edge (P_2, x) to $\hat{E}_{t_2}$.

Figure 4.21.: $\mathcal{F}_{\mathrm{CT}}(\mathcal{P}, P_{\mathrm{mat}})$ – Neighborhood and Broadcast

$\mathcal{F}_{\mathrm{CT}}(\mathcal{P}, P_{\mathrm{mat}})$ – Replay/Relay, Matching and Warning

Replay/Relay:

1. Receive $(relay, t, P'_1, P'_2, P'_3, P'_4)$ from the adversary, where $P'_1 \in \text{pseudonymize}(\mathcal{P})$, $P'_2, P'_3 \in \mathcal{P}_{corrupted}$, $P'_4 \in \text{pseudonymize}(\mathcal{P}_{corrupted} \times \mathcal{P}_{honest})$.
2. Let $P_j \coloneqq \text{pseudonymize}_i^{-1}(P'_j)$ for $j = 1, 2, 3, 4$. (Note that $P_2 = P'_2$, $P_3 = P'_3$.)
3. If $(P_1, P_2) \in E_t$, $(P'_3, P'_4) \in E'$, let $\hat{P}_4 \in \mathcal{P}$ be the node such that $P_4 = (P_3, \hat{P}_4)$, and add the new edge $(P_1, \hat{P}_4)$ to $\hat{E}_t$.

Handling Match Requests:

1. Receive (*positive*) from party P.
2. If $P \in \mathcal{P}_{corrupted}$, skip to step 4. If $P \notin \mathcal{P}_{infected}$, return. Otherwise, continue.
3. Let $R \coloneqq \mathbb{N} \cap [e_{lt} - 14, e_{lt})$. For each epoch $i \in R \times \mathbb{Z}_{96}$ (the relevant time period), determine the set $\Delta\mathcal{WP}_i$ (new parties at risk) of nodes P' such that $(P', P) \in \hat{E}_i$. Skip to step 5.
4. Let $lastInfected_{lt} \coloneqq \max\{i \in \mathbb{N} \colon \exists j \in \mathbb{Z}_{96}, \text{ such that } P \in \mathcal{P}_{infected,(i,j)}\}$. (Let $lastInfected_{lt} \coloneqq -\infty$ if this set is empty.) Let $R \coloneqq \mathbb{N} \cap [e_{lt} - 14, e_{lt}) \cap [0, lastInfected_{lt}]$. Send (*forceWarning*) to the adversary, asking for subsets S_i of (the pseudonyms of) uncorrupted parties which have been in proximity to a corrupted party during epochs in R, i.e. $S_i \subseteq \{q \in \text{pseudonymize}_i(\mathcal{P}_{honest}) \mid \exists q' \in \mathcal{P}_{corrupted} \text{ where } (\text{pseudonymize}_i^{-1}(q), q') \in E_i\}$. After the response, set $\Delta\mathcal{WP}_i = \text{pseudonymize}_i^{-1}(S_i)$ as the set of parties that will be warned for the current epoch.
5. For each $i = (i_{lt}, i_{st}) \in R \times \mathbb{Z}_{96}$, add $\{(P', i_{lt}) \mid P' \in \Delta\mathcal{WP}_i\}$ to the list of active warnings $\mathcal{WP}$.

Handling Warning Query:

1. Receive $(query, t)$ from party P
2. Return 1 if $(P, t) \in \mathcal{WP}$, otherwise return 0.

Figure 4.22.: $\mathcal{F}_{\mathrm{CT}}(\mathcal{P}, P_{\mathrm{mat}})$ – Replay/Relay, Matching and Warning

quantifying over all environments (and implicitly over all computable "real world" scenarios of contact graphs and infection statuses), without a proper analysis of the costs and impracticalities of achieving this in the real, physical world[22], we simplify the analysis and abstract from the many scenarios that may arise in its actual use. In the light of this, we give, in the following, an interpretation of our security guarantees and a discussion of guarantees and limitations not captured by our model, in the following:

4.3.9. Privacy Analysis

For our privacy analysis, we assume corrupted users can link some public identifiers they directly observe to the real identities of the corresponding user, e.g. by accidentally meeting someone they know. This pessimistic approach yields a worst-case analysis regarding the information available to corrupted users.

Privacy of Positively Tested Participants In the ideal functionality ($\mathcal{F}_{\text{CT}}$ in section 4.3.8), the attacker is provided with $\mathcal{P}_{infected} \cap \mathcal{P}_{corrupted}$, so the infection status of honest parties is protected here. The pseudonymized contact graph is independent of the infection status. Apart from the inherent leakage about the infection status from warning queries, this models that the protocol does not introduce any additional information leaks on the infection status of honest participants. (For example, a motivated "paparazzi" attacker might take a "group testing" approach in that he tries to get near several subgroups of a larger group to later single out positively tested participants upon warning.) Note that is in contrast to DP3T, where short-term identifiers of a whole day can be linked together, upon uploading data in case of an infection.

Privacy of Warned Participants Our protocol naturally protects the privacy of warned participants and their social graph as the published warning identifier is computationally unlinkable to any information that can be recorded locally (i.e. pids), and also deciding whether some identifiers belong to the

[22] While it would be perfectly possible for an environment to use as a contact graph a fresh, and independently sampled random graph on $\mathcal{P}$ for each short-term epoch, the costs of implementing this in real time for 15 minute epochs would be quite challenging.

same user, is impossible. Thus, a wid does not help the attacker in breaking the users' privacy.

4.3.9.1. Privacy in the Case of Compromised Servers

This section presents an analysis of the privacy guarantees offered by our protocol if servers are compromised. See the appendix of [47] for the formal guarantees in case of passively corrupted servers.

Linking Public Identifiers from the Long-Term Epoch If the submission server is compromised, the attacker will be able to link different public identifiers pid to the same secret sid, and hence can link the public identifiers the user is using during the same long-term epoch. This poses a privacy threat, if the attacker additionally has observed some of the targeted public identifiers pid, which requires users colluding with the server.

Similarly, if both the matching server and the warning server are corrupted, the attacker can decrypt the sid values stored by the matching server to recover the wid value, and hence again link public identifiers to the secret identifiers sid and the respective warning identifier wid. Such an attacker that also colludes with corrupted users may be able to link public identifiers to times and places where these identifiers have been broadcast, and hence observe parts of the user's location history and track a user for up to one day. We stress that even if all servers are compromised, an attacker will not be able to link public identifiers used on different days (assuming the use of anonymous channels).

Contact Information of Infected Users Information about encounters between users is stored strictly on the user's devices. Only the meeting history, i.e. the list of encountered public identifiers, without times and places of meetings, of infected users is transmitted to the central servers.

If the attacker has compromised the matching server *and* is able to link public identifiers used on the same long-term epoch (as in the previous scenario), the attacker might be able to infer repeated meetings of the infected user, i.e. she can learn how many encounters with the same persons the infected user's device has registered within each day. If the attacker has additionally observed some of the warned public identifiers at specific times and places,

the attacker will also learn where and when the encounter took place, and hence learn parts of the location history of the infected user as well as the warned users.

Warnings Issued If the attacker has compromised the matching server, she can immediately observe the public identifiers of all users who have been colocated with infected users. If the attacker can additionally link a public identifier to a specific individual, the attacker can conclude this person has received a warning. (Note that a similar attack is possible in the DP3T protocol [287], but even without compromising a server.)

4.3.10. Security Analysis

We now analyze an attacker's ability to cause false negatives or false positives. As above, we assume central servers to follow the protocol. See the appendix of [47] for the formal guarantees in case of passively corrupted servers.

Creating False Negatives A false negative occurs when an uncorrupted user A has been in colocation with an uncorrupted infected user B but A does not receive a warning. These false negatives are not possible in our protocol. In $\mathcal{F}_{\text{CT}}$ this property is modeled by, $\hat{E}$ initially containing all edges between honest users, and during the protocol edges can only be added and never removed. (Note that we excluded jamming of the BLE signal by the adversary, as motivated in section 4.3.3.)

Only in the case of a (passively) corrupted matching server can the adversary evade these guarantees regarding false negatives. This is because a corrupted matching server will learn the TANs at the time when honest users upload their list of observed identifiers. Exactly during (in parallel to) this step, an adversary may "use up" (and thereby invalidate) this TAN (after the matching server learned it), but before the honest user's request is finished. However, note that in this case, it is evident to the honest user that the TAN has been invalidated, pointing towards a passive corruption of the matching server (which is hence incentivised to not use this attack.)

False Positives Regarding Honest Users An honest user A is subject of a false positive if she has not been colocated with an infected user, but she nonetheless receives a warning. Our security goal is to prevent false positives, unless i) A was in proximity to a corrupted user, *and* ii) the attacker is in proximity to an infected user, or has been infected themselves.

This is captured by the following fact: In order for an honest party A to be warned, the party has to be included in $\mathcal{WP}$. It can only be included in $\mathcal{WP}$, if there is an outgoing edge from A in $\hat{E}$ (warning triggered from an honest party) or there is an outgoing edge from A to a corrupted party in E (warning triggered from a corrupt party).

If A was not in proximity to a corrupted user, the attacker cannot use "Replay/Relay" to add new outgoing edges to $\hat{E}$ (as $(P'_3, P'_4) \notin E'$ in step 3, because P'_3 is corrupted and $\hat{P}_4 = A$ is not in proximity to a corrupted user) and hence cannot trigger a false warning from an honest party (unless the submission or the matching server is passively corrupted, as in this case the adversary learns otherwise unobserved pids to use for this). The attacker cannot trigger a warning for an honest that has not been in contact with a corrupt party, as step 4 of "Handling Match Requests" requires all S_i to be empty in this case (unless the submission or the matching server is passively corrupted).

If the attacker has not been in proximity to an infected user and no corrupted party has been infected, the attacker can only insert edges into $\hat{E}$ using "Replay/Relay" where the target will never be infected. So a false warning cannot be triggered from an honest party. Regarding warnings triggered from a corrupt party, $lastInfected_{lt}$ will always be $-\infty$ in step 4 of "Handling Match Requests" and parties can be added to $\mathcal{WP}$. This concludes our argument that producing a false positive for an honest user requires proximity of the attacker to both, the honest user and to an infected user (or the corrupted user is infected).

5. Random Oracle

5.1. The Random Oracle Debate

In the introduction we would like to address the debate about Random Oracles. It is left to the reader to determine the weight of each individual argument from the literature.

In 1994, Bellare and Rogaway [30] asserted that the Random Oracle paradigm allows for much more efficient constructions of cryptographic protocols, a claim that holds true in practice until today. Almost every real-world cryptographic construction actively employed in everyday applications is both constructed and proven secure in the RO Model. Maybe the most prominent example is the Fujisaki-Okamoto transformation for CCA2 secure Public-Key Encryption. This transformation is pervasive in almost every standardized Post-Quantum cryptosystem, ML-KEM [226], FrodoKEM [278], Classic McEliece [277]. Even schemes that do not rely on the FO-transformation nevertheless require the random oracle as a baseline such as RSA-OAEP [12]. Thus, the random oracle model remains an indispensable tool in the design and analysis of secure cryptographic systems.

The initiation of the debate about the RO model can be traced back to the pathological constructions of digital signatures and encryption schemes presented in [82]. While these constructions were proven secure in the RO model, their security completely collapsed when the RO was replaced by any implementation of a cryptographic hash function. Since then, Eaton and Song extended the uninstantiability results from [82] to the quantum random oracle model (QROM) as well. In the following, we will write (Q)ROM for addressing both the ROM and the QROM. Additionally, a series of works have discussed the advantages and disadvantages of the RO model [76, 82, 84, 158, 164, 182, 192, 196, 207, 214].

Unfortunately, many of these works often deviated from the core discussion of the Random Oracle, venturing into the broader debate about the interplay between the theory and practice of cryptography. While this is undoubtedly an important discussion, it complicates efforts to follow the dedicated debate on the RO model.

In the following, the author of this thesis attempts to humbly summarize the RO debate until today. The proposal is to structure the debate into two subsections: arguments for using the RO Methodology and arguments for avoiding it. Most of the arguments lack absoluteness in the sense that their weight can be diminished by counter-arguments. To the best of the author's knowledge, it is elaborated on this where possible. The final assignment of argument weight is left to the reader.

5.1.1. Arguments for Avoiding the Random Oracle

This section summarizes the arguments that may be used to justify cryptographic research that explicitly avoids the random oracle assumption. These arguments are the motivation for this thesis.

Random Oracle as the Checkov's Gun of Provable Security Provable security and the soundness of the cryptographic formalisms used therein are essential cornerstones of the art of cryptographic design. The foundational work by [82], along with the extended results from [136], demonstrated that the (Q)ROM is inherently unsound in general. This ultimately implies that security theorems relying on the (Q)ROM may be vulnerable to additional attack vectors that exploit the inconsistency between the idealized random oracle and the practical reality of hash functions, which are not truly random as the RO methodology assumes.

However, the weight of this argument is often diminished by emphasizing that the likelihood of such events is considered small, given the absence of real-world examples indicating a significant risk. This reasoning is frequently invoked in works that choose to rely on the (Q)ROM.

Adversaries with Quantum Computer Capabilities and Universal Composability The random oracle methodology enables various proof techniques that allow for concise proof statements. These statements, while potentially appearing convoluted at first glance, have withstood considerable scrutiny over the last few decades and have not revealed significant breakdowns. The most prominent use of the random oracle is to guarantee the extractability of otherwise secret information, such as plaintexts or symmetric keys in a Key Encapsulation Mechanism (KEM). By simulating the random oracle for the adversary, the simulator can gain access to information that would otherwise remain hidden due to the potentially obfuscated inner workings of the adversary.

This ultimately facilitates succinct proof techniques, which have become crucial tools, such as the Fujisaki-Okamoto transformation [168] or the Fiat-Shamir transformation for digital signatures [203]. However, employing these techniques comes at a cost. To extend proofs beyond standard theorems, such as CCA2 or existential unforgeability under chosen message attack (EUF-CMA), one must ensure that the proof in the simple ROM model remains valid in extended settings, such as the quantum random oracle model (QROM) [59] or in universally composable frameworks [76]. This often necessitates introducing additional artifacts into the model solely to align with the random oracle methodology, without necessarily contributing meaningful details relevant to real-world applications.

This criticism can, however, be mitigated by noting that overly complex proofs and models may stem more from the expressiveness of the universal composability framework or the additional formalism required to accommodate quantum computing, rather than from the inherent limitations of the ROM. Moreover, there are few, if any, indications that non-ROM proofs for ROM-avoiding schemes in these models are inherently more succinct or involve fewer artifacts. On the other hand, there are no general results suggesting that non-ROM proof techniques inherently lead to more complex proofs.

Ultimately, while the use of the ROM introduces the aforementioned challenges, further research into alternative proof techniques that avoid the ROM could potentially yield better results.

Necessary and Sufficient Security Arguments The RO is a very strong assumption in a sense that relying upon this assumption one is able to construct cryptographic schemes with security against strong adversaries. Having a truly random function, which we are able to simulate (or even adaptively program) for any adversary at our disposal many public-key encryption security definitions become trivial to achieve and albeit the modular decomposition of the Fujisaki-Okamoto transformation [153] by Hofheinz et al. [168] suggests that one may model this transformation in two distinct steps having the random oracle at our disposal seems to make this decomposition inaccurate. The reason is that even the first transformation, the T-transformation, yields a PKE scheme that is far more secure than the OW-PCVA level suggested by Theorem 3.1, resulting in cryptosystem constructions with overly strong security guarantees. For example, compare a T-transformed PKE—which, due to derandomization, loses most of its homomorphic properties—to deterministic textbook RSA, which is already OW-PCVA and remains multiplicatively homomorphic.

5.1.2. Arguments for Using the Random Oracle

This section summarizes the arguments that may be used to justify cryptographic research that uses the random oracle methodology.

Efficiency Undoubtedly, the random oracle methodology has so far yielded cryptosystems that balance both security and efficiency for real-world use. Even the new post-quantum standards, such as ML-KEM [226] and ML-DSA [225], adhere to this methodology, albeit with additional modifications, such as extending the Fiat-Shamir paradigm to handle aborts [203]. To the best of our knowledge, no result has yet demonstrated a CCA2-secure PKE scheme that matches the efficiency of its CCA2-secure counterparts. Such a result would be highly impactful, as many protocols that avoid the random oracle assumption must rely on a CCA2-secure PKE at some stage as a foundational building block for e.g. a secure channel.

Thus, while adopting the random oracle assumption in protocol design is not strictly sufficient, it is often a necessary condition when efficient and secure building blocks are required. This argument can be challenged in a manner similar to the critique of the second argument. In some ways, it becomes a self-fulfilling prophecy: if research into making non-ROM protocols more

efficient is neglected, ROM-based protocols will naturally maintain their dominance in terms of efficiency.

A notable exception are digital signatures. For digital signatures, there already exist practical standard-model alternatives that avoid random oracles and are almost as efficient as e.g. ML-DSA: stateful hash-based signature schemes such as XMSS and LMS [169, 209]. XMSS admits proofs based only on standard properties of hash functions (i.e., without number-theoretic hardness assumptions), and both XMSS and LMS are standardized by the IETF and profiled for deployment in NIST SP 800-208 [113, 169, 209].

A (Reliable) Real World Assumption No real-world attack is currently known that exploits the gap between the random oracle (RO) model and a secure cryptographic hash function. This argument is consistently repeated in almost every work that relies on the random oracle model. As demonstrated by [192], this argument holds merit, with evidence suggesting its validity. In fact, Koblitz and Menezes have shown that many attempts to remove the random oracle from ROM-based schemes have resulted in the introduction of other weaknesses that were trivially exploitable, in stark contrast to the relatively low risk associated with the fuzzy notion of a random oracle.

Moreover, deploying a product based on a theoretically secure non-ROM protocol introduces additional challenges. Many such algorithms are non-standard and lack the rigorous scrutiny of experienced cryptanalysts. Beyond the simplicity and efficiency of ROM-based cryptosystems, a critical and perhaps one of the most important arguments for relying on the random oracle model in real-world applications is the familiarity and extensive scrutiny of its proof techniques and implementation practices. These have been refined over the last three decades since the random oracle methodology was introduced.

Avoiding the random oracle methodology would necessitate developing entirely new techniques and practices, which are unlikely to be fully secure in their initial iterations. Such uncertainty is unacceptable for real-world systems. The field of cryptographic research must continue advancing in this direction, working on alternatives that may eventually achieve comparable reliability.

A notable exception to this argument is a recent work from Khovratovich et al. [182], which describes practical attacks on the RO-based Fiat-Shamir

technique.[1] The authors construct explicit circuits for which an interactive proof is sound, yet its Fiat–Shamir (FS) non-interactive version becomes unsound when you instantiate the RO with a real hash function. Concretely, they target the GKR family of interactive proofs [161] and exhibit "weak challenge" points: if the verifier's challenge equals (a function of) the circuit's true output, a cheating prover can make the verifier accept a false statement after FS is applied. They then design circuits that compute the FS challenge internally (by embedding the same hash the FS transform uses) so that this bad challenge occurs with overwhelming probability. This is not a break of FS in general, nor of common digital signatures derived from Σ-protocols. It's a targeted but practical counterexample: certain FS-flattened protocols (like GKR-style) become insecure for specially crafted circuits that smuggle in the FS hash and trigger a weak challenge.

Concluding for now, this work shows that the "random oracle paradigm" of thoughtlessly embedding an interactive proof system into the FS heuristic and assuming that everything will be fine is misleading, because this "paradigm" was flawed to begin with. It is not the first time that these kinds of protocols have undergone special scrutiny (see, e.g., [35]), and it is very likely not the last. It was already clear that special care must be taken with the FS heuristic, and this remains true. From an engineer's point of view, it is acceptable not to focus on theoretical foundations, because it should be self-evident that the foundations relied upon must be formally sound, and no one should expect engineers to handle formally unsound foundations securely. To the author, the conclusion is that now even more scrutiny is required from theoretical research to find similar results and to provide more guidance for cryptographic engineers in designing and implementing secure systems. Therefore, this work is not yet a smoking gun that Chekhov would expect to fire at some point. Nevertheless, now there is clearly ammunition lying next to the gun.

[1] Matthew Green has a notable series of blog posts about explaining the attack and it's reach in popular science terms `https://blog.cryptographyengineering.com/2025/02/04/how-to-prove-false-statements-part-1/`

5.2. Secure Message Transfer without Random Oracles

This section is based on joint works with Dr. Rebecca Schwerdt, Astrid Ottenhues, Roland Gröll, and Prof. Dr. Jörn Müller-Quade. In these works we devised a security and encryption definition that, in conjunction with authenticated channels, constructs secure channels without the need for a RO and is weaker than all previously known public-key encryption security definitions, with CCA2 being currently the most prevalent one. The results of this work were published in the proceedings of PKC 2022 [46] and in the full version [43]. Subsequently, Sarai Eilebrecht and Laurin Benz joined the team, and the security notion was adapted to fit into the hybrid encryption paradigm. This adaptation was published in the proceedings of PKC 2023 [33] and in the full version [266]. Dr. Rebecca Schwerdt dedicated her dissertation to establishing the theoretical foundation for exploring the definition hierarchy of public-key encryption security. Interested readers are encouraged to refer to her PhD thesis [265] for a comprehensive overview of the foundations, while this work provides brief security proof sketches, constructions, and essential theorems and lemmas.

Our motivation was to devise a security definition that, while weaker, remains sufficient to allow for the construction of public-key encryption schemes that may be employed in the construction of a secure channel than the previously known CCA2 constructions in the standard model, i.e. without the random oracle heuristic, making them suitable for real-world use. However, our expectations regarding practicality were not fully met during our research on this security definition.

Despite this, we did not find any arguments or indications that rule out potential practical advantages compared to the standard model CCA2 counterparts. On the contrary, Dr. Rebecca Schwerdt clearly demonstrated in [265] that sender-binding chosen plaintext attack (SB-CPA) is strictly weaker than CCA2. Indeed, our definition aligns more closely with CPA security, which motivated us to term it SB-CPA instead of SB-CCA. As a result, constructions do not need to strengthen their security to achieve extractability or non-malleability, which is required in the prevalent case of CCA2. Consequently, the definition enables trivial constructions from schemes that already achieve CCA2 security and possibly many more, yet unknown, that are weaker and otherwise not possible if CCA2 security is the only choice.

The lack of research in this direction is another indication. SB-CPA serves as a fundamental security definition for a basic encryption primitive, with foundations tailor-made to efficiently fit into the FO-paradigm. A notable example of a substantial research investment for CCA2 constructions based on FO-transformations was the NIST standardization call [2] of post-quantum cryptography.

Conclusively, despite unfulfilled expectations in the aspect of practicality, the fact that SB-CPA allows for more relaxed constructions and the non-existence of research in the foundations of SB-CPA constructions are promising indications for future work. Our works aimed to devise SB-CPA constructions that can serve as a first stepping stone towards practicable constructions without relying on random oracles.

Robert Brede [68] presented promising results in his master's thesis. The KEM comparison table showed noticeable improvements upon the previous works and albeit we were not able to reach our ultimate goal of devising an SB-CPA post-quantum construction of an SB-KEM that is as efficient as kyber/ML-KEM [226, 279], the results demonstrate that across all dimensions we are now able to construct a secure channel that is practical at least, but maybe not the most efficient one.

In this section we attempt to construct a real protocol for secure message transfer without random oracles by facilitating all of the results from Sections 5.2.2, 5.3 and 5.4 and analyze the performance dimensions of this protocol in Section 5.4.

5.2.1. Related Work

A PKE satisfying CCA2 security was already shown by Canetti in [78] to realize secure message transfer (SMT) in the universal composability (UC) framework by communicating confidentially over authenticated channels. On the other hand CCA2 was also shown by Canetti et al. [85] to be unnecessarily strong for this purpose. Hence relaxations of CCA2 came into focus. Among these relaxations is indistinguishability under replayable chosen ciphertext attack (IND-RCCA), introduced by Canetti, Krawczyk and Nielsen in [85] where they show that IND-RCCA suffices to UC-realize SMT using

[2] `https://csrc.nist.gov/projects/post-quantum-cryptography`

authenticated channels. IND-RCCA differs from CCA2 in the characteristic that the ability to generate ciphertexts, which decrypt to the same plaintext as the test ciphertext, does not help the adversary to win the game. Recently, Badertscher et al. [22] examined IND-RCCA and variations of it using the constructive cryptography framework to construct a confidential channel—a strictly weaker notion than SMT. They concluded that IND-RCCA is not sufficient to realize confidential channels when using the authenticated channel for public key transfer only. They introduce a stronger security definition to solve this problem whereas we, like the original IND-RCCA paper, assume authentication for every message transfer.

Another direction to achieve weaker security definitions is that of tag-based encryption (TBE) which was introduced by MacKenzie, Reiter and Yang [205]. They introduced the notion of tag-based non-malleability, which is nowadays known as indistinguishability under adaptive-tag weakly chosen ciphertext attack (IND-atag-CCA) security for TBE. The authors show that an IND-atag-CCA secure TBE scheme is also sufficient to realize SMT when provided with authenticated channels. A relaxation, indistinguishability under selective-tag weakly chosen ciphertext attack (IND-stag-CCA), has been shown to facilitate CCA2 constructions with the additional usage of a one-time signature scheme [58] or a message authentication code combined with a commitment scheme [56]. Both constructions are originally meant for identity based encryption (IBE), but Kiltz showed in [184] how to adapt these for the TBE setting. So far IND-stag-CCA secure TBE has not been shown, however, to directly facilitate SMT.

Let us now look at how CCA2 secure schemes can be constructed, and thus directly yielding a secure channel, without employing the RO model. The most efficient general construction paradigms nowadays are the lossy trapdoor functions by Peikert and Waters [236], the correlated products by Rosen and Segev [254] and the very similar k-repetition by Döttling et al. [131][3], the Cramer-Shoup-like constructions [116] and the adaptive trapdoor functions [186]. More efficient constructions of SMT can be built upon TBE. The—to the best of our knowledge—most efficient code-based TBE schemes nowadays are due to Kiltz [184], Kiltz, Masny and Pietrzak [185], Cheng et al. [100] and Yu et al. [311]. In their schemes, the notion of IND-stag-CCA security for TBE is required, which can be used to construct CCA2 schemes by adding

[3] In spite of being a generic paradigm this work was applied only to McEliece so far.

one-time signatures or message authentication codes and commitments as mentioned above.

Regarding both of our research questions we see that although some progress was made in previous works there is still a lot of room for improvement.

5.2.2. Sender-Binding Encryption

The construction of secure channels is one of the main goals of cryptography. Among the milestones that have been reached to this end are public-key cryptosystems by Diffie and Hellman [128], semantic security by Goldwasser and Micali [162] (today referred to as CPA), and the stronger CCA2 by Rackoff and Simon [245].

Nowadays, CCA2 secure public-key encryption (PKE) is a cornerstone of many protocols realizing secure channels for our daily life applications. One of the most typical applications is the encryption of e-mails. This is usually realized by implementations of either the S/MIME [262] or OpenPGP [74] standard. Both standards utilize a PKI and digital signatures to realize authenticated channels. Hence we see that widespread applications of secure message transfer (SMT) integrally use authenticated channels and a PKI in addition to encryption. Secure message transfer (SMT) is an abstraction of authenticated and encrypted communication in the universal composability (UC) model. How secure message transfer (SMT) can be utilized in practical real world scenarios can be seen for example in [249].

It is widely known that CCA2 is unnecessarily strong to construct SMT when authenticated channels are already present [85]. In addition many concrete CCA2 constructions either lack efficiency to be considered practical constructions or were only proven secure within the RO model. We refer to Section 5.1 for a summary of the debate around the ROM topic. However, we consider the exploration of alternatives just as important and therefore focus on constructions proven secure in the standard model in this work. Hence the following question arises:

> *What is the weakest security definition in order to establish a secure channel in the standard model if we assume existing authenticated channels?*

In an attempt to answer this question we find a non-trivial relaxation of the weakest prior notions of replayable chosen ciphertext attack (RCCA) from [85] and adaptive-tag weakly chosen ciphertext attack (atag-wCCA) from [205], which were both shown to be weaker than CCA2 and used to construct secure channels. While this work does not provide an ultimate answer to this question—i.e., we do not prove that our definition, labeled indistinguishability under sender-binding chosen plaintext attack (IND-SB-CPA), is the weakest possible and hence necessary—we show IND-SB-CPA to be sufficient in the sense that any encryption protocol satisfying this security can be used directly to UC-realize SMT using authenticated channels.

Although this is an interesting theoretic result, we argue that for more relevancy the previous question needs to be accompanied by the following:

> *Can weaker security notions lead to simpler and more efficient constructions of a secure channel in the standard model?*

In the current state of affairs, TBE is an attractive choice for constructing efficient CCA2 secure PKE in the standard model as already the weakest established TBE security notion, IND-stag-CCA, was shown by Kiltz [184] to yield a transformation to CCA2 secure PKE by adding one-time signatures for example. We show that IND-stag-CCA secure TBE does not actually require prior transformation to CCA2 secure PKE in order to construct secure channels: By deriving the new concept of sender-binding encryption (SBE) from TBE we are able to construct secure channels directly from IND-stag-CCA secure encryption. The intuition behind SBE is to tie ciphertexts not only to the receiver as with classic PKE notions, but to the sending/encrypting party as well.

Somewhat surprisingly, via IND-SB-CPA secure SBE we are also able to construct secure channels from DRE which only satisfies CPA security and soundness. CPA secure DRE was initially introduced by Diament et al. [127] to facilitate message transmission from one sender to two different receivers and allows for interesting applications such as security puzzles for denial of service countermeasures. Subsequently, Chow et al. [109] introduced the property of soundness for DRE, and proved it to be crucial for some applications such as plaintext awareness (PA). Our DRE-based protocol allows for a much simpler and more efficient encryption than IND-stag-CCA secure TBE for constructing secure channels and hence allows us to answer the second question in the positive.

One caveat of the construction via DRE is that we require an extended PKI that realizes the *KRK* functionality. This guarantees that users of the PKI have knowledge of their private keys. While this is not a common functionality of PKIs in use today, there are first protocol drafts like OTRv4[4] which utilize deniable authenticated key exchange protocols that rely on the KRK functionality. In this case those are DAKEZ and XZDH due to Unger and Goldberg [289].

5.2.3. IND-SB-CPA Security

SMT is commonly realized by combining an IND-CCA2-secure PKE scheme or an IND-atag-CCA-secure TBE scheme with authenticated channels. As highlighted in the introduction, however, both of these security notions appear unnecessarily strong and restrictive for this application.

We are not the first to make this observation (see Section 5.2.1), as prior efforts have sought to relax these security notions to facilitate SMT. For instance, the RCCA relaxation of CCA2 and the use of IND-stag-CCA-secure TBE schemes are notable examples of such attempts. These relaxations aim to strike a balance between security and efficiency by discarding certain overly stringent requirements that are irrelevant for specific use cases, like secure message transmission.

In this section, we introduce the concept of SBE and propose a new security notion, IND-SB-CPA. This notion is weaker than the IND-stag-CCA relaxation but still captures the level of security required for SMT via authenticated channels. The goal of this new framework is to eliminate redundant constraints while preserving the core security properties essential for practical deployment. Although the term SBE has not been formally defined before, all prior realizations of SMT via authenticated channels—whether based on CCA2, RCCA, atag-wCCA, or selective-tag weakly chosen ciphertext attack (stag-CCA)—essentially work by constructing a de facto SBE scheme from the underlying encryption scheme. This observation underscores the potential for a more unified and streamlined approach to defining security notions for SMT, avoiding unnecessary overhead introduced by stricter models like CCA2.

[4] https://github.com/otrv4/otrv4/blob/master/otrv4.md

Definition 42 (Sender-binding encryption (SBE)) The interface of an SBE scheme is given by a set of three PPT algorithms (Gen, Enc, Dec):

$$\begin{aligned} \mathsf{Gen}: &\quad 1^\lambda \mapsto (\mathsf{sk}, \mathsf{pk}) \\ \mathsf{Enc}: &\quad (\mathsf{pk}, S, m) \mapsto c \\ \mathsf{Dec}: &\quad (\mathsf{sk}, S, c) \mapsto m. \end{aligned}$$

We expect an SBE scheme to fulfill the notion of correctness, i.e. that whenever $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\lambda)$, then

$$m = \mathsf{Dec}(\mathsf{sk}, S, \mathsf{Enc}(\mathsf{pk}, S, m)).$$

Some remarks are in order about this use case definition of SBE.

In addition to the inputs present in any common PKE scheme, encryption and decryption algorithms use the encrypting party's ID S[5] as well. The ID of a party represents the identification information used within the system. This might be the public key itself, the party's actual name, their e-mail address etc. This does not only bind a ciphertext to the receiving party who holds the secret key and is able to decrypt the ciphertext—as any PKE scheme does—but also to the party who created the encryption.

However, binding a ciphertext to the ID of a sending/encrypting party alone does not yet yield obvious benefits. Even if a specific party ID is specified by the protocol, party IDs are public knowledge and malicious parties can insert any ID they want. SBE starts to unfold its benefit when used in conjunction with IDs that are associated with authenticated channels. This channel reliably indicates the true sender S of a message. Checking this against the sender ID bound to the received ciphertext prevents (honest sender) replay attacks, i.e., that this message was just copied from another (unwitting) sender. The terminology "sender-binding" stems from the example application of SMT via authenticated channels where this is taken to be the encrypting/sending party. Of course there might be other use cases for SBE where the encrypting party does not constitute a "sender". But throughout this paper (whenever we talk about SBE) we use R and "receiver" to denote the party owning the

[5] For the encryption mechanism we will sometimes omit the explicit input of the ID S if it is clear from the context which party S is conducting the encryption.

keys $(\mathsf{sk}_R, \mathsf{pk}_R) := (\mathsf{sk}, \mathsf{pk})$, and S and the term "sender" for the party whose ID is input on encryption and decryption.

Given the definition of an SBE scheme we still need to arrive at a meaningful corresponding security notion. With the additional key pair $(\mathsf{sk}_S, \mathsf{pk}_S)$ we also need to define how much decryption power the adversary gets for these keys in the two oracle phases. We choose this intuitively to be symmetric with the challenge keys $(\mathsf{sk}_R, \mathsf{pk}_R)$. Because this gives a weaker notion and is still enough for SMT we restrict decryption not only for the challenge tag S but for R as well.

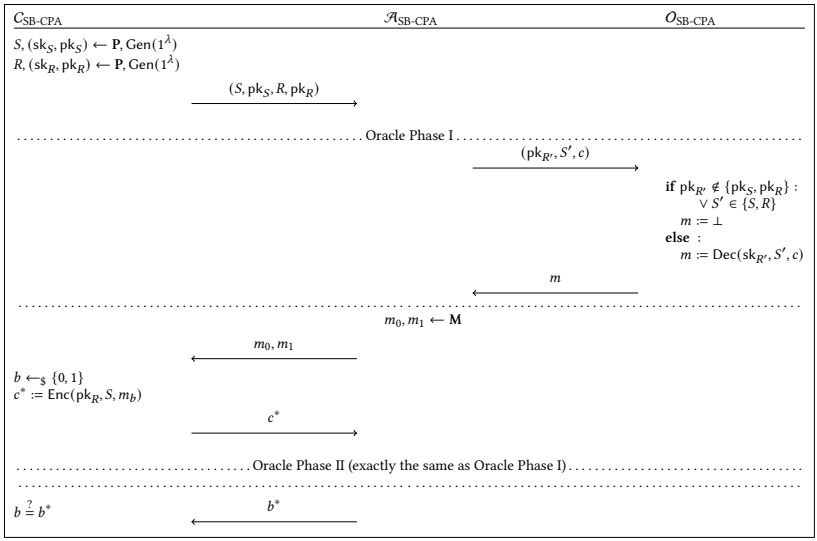

Figure 5.1.: The IND-SB-CPA Game for SBE

Definition 43 (IND-SB-CPA) An SBE scheme (Gen, Enc, Dec) satisfies IND-SB-CPA security, if and only if for any PPT adversary $\mathcal{A}_{\text{SB-CPA}}$ the advantage to win the IND-SB-CPA game shown in Figure 5.1 is negligible in λ.

Within this context of SBE, the new security notion of IND-SB-CPA has a very straight forward intuition: If it was possible to alter a ciphertext $c \leftarrow \mathsf{Enc}(\mathsf{pk}, S, m)$ to some c' which successfully decrypted under another sender ID S' (i.e. $\mathsf{Dec}(\mathsf{sk}_R, S', c') \neq \perp$), replay attacks would be possible. Let us look at this in a bit more detail. From Figure 5.1 we see that the

adversary is provided with perfect knowledge (via oracle or its own power) about any ciphertext which involves any other party than just S and R. About communication between S and R, on the other hand, the adversary learns nothing—with the natural exception that encryption only requires public knowledge and can therefore be conducted by the adversary as well. A directed version—where the adversary can additionally decrypt messages from R to S (but not from S to R)—would also naturally suggest itself. But as mentioned before our choice of a symmetric version is strictly weaker as well as sufficient for SMT construction. Having no decryption possibilities for the channel (S to R) along which the challenge ciphertext is sent justifies classifying IND-SB-CPA as some form of CPA security.

In the next section we show that IND-SB-CPA is not merely of academic interest by giving a generic example construction for IND-SB-CPA secure SBE via DRE.

5.2.4. Transformation from DRE to SBE

In this section, we generically construct an IND-SB-CPA-secure SBE scheme from DRE. Although DRE was initially designed to facilitate message transmission from one sender to two different receivers, selecting one of the receivers to act as the sender itself provides a mechanism to bind the ciphertext to the sender. This enables the construction of an IND-SB-CPA-secure SBE scheme. Combined with the use of PKIs employing KRK, this results in an encryption scheme where the sender is guaranteed to be aware of the plaintext. Without KRK, there is no assurance that the sender knows the private key corresponding to their public key, making this awareness unreliable. A possible realization of the KRK functionality is for the PKI to require a zero-knowledge proof of knowledge of the secret key during public key registration. While this may be a computationally expensive operation, it only needs to be performed once at the time of registration.

We require the underlying DRE scheme to be sound, IND-CPA secure, and compatible with the key registration functionality $\mathcal{F}_{\mathsf{KRK}}$. For the formal definition of DRE, its soundness properties, and the specification of $\mathcal{F}_{\mathsf{KRK}}$, we refer the reader to Section 3.7 and Figure 3.6, respectively.

Regarding compatibility, we require the DRE scheme to support an efficiently computable boolean function $\mathsf{f}_{\mathrm{Key}}$. On input of a (possible) key pair $(\mathsf{sk}, \mathsf{pk})$

this function decides whether the keys "belong together", i.e., whether they could have been output by the encryption scheme's key generation algorithm or might just be an unrelated pair of values:

$$\mathfrak{f}_{\mathsf{Key}} : (\mathsf{sk}, \mathsf{pk}) \mapsto \begin{cases} \mathit{true}, & (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\lambda) \\ \mathit{false}, & \text{else.} \end{cases}$$

This is necessary for the scheme to be used in conjunction with the registration functionality $\mathcal{F}_{\mathsf{KRK}}$.

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure DRE scheme which admits a function $\mathfrak{f}_{\mathsf{Key}}$. We define a new encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$:

$\mathsf{Gen}(1^\lambda)$ executed by party P:

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\lambda)$.
- Register $(\mathsf{sk}, \mathsf{pk})$ with $\mathcal{F}_{\mathsf{KRK}}^{\mathfrak{f}_{\mathsf{Key}}}$.

↪ Return $(SK, PK) := ((\mathsf{sk}, \mathsf{pk}), P)$.

$\mathsf{Enc}(PK_R, S, m) = \mathsf{Enc}(R, S, m)$ executed by party S:

- Retrieve pk_R and pk_S from $\mathcal{F}_{\mathsf{KRK}}^{\mathfrak{f}_{\mathsf{Key}}}$.

↪ Return $c \leftarrow \mathsf{Enc}(\mathsf{pk}_R, \mathsf{pk}_S, m)$.

$\mathsf{Dec}(SK_R, S, c) = \mathsf{Dec}((\mathsf{sk}_R, \mathsf{pk}_R), S, c)$ executed by party R:

- Retrieve pk_S from $\mathcal{F}_{\mathsf{KRK}}^{\mathfrak{f}_{\mathsf{Key}}}$.

↪ Return $m := \mathsf{Dec}(\mathsf{sk}_R, \mathsf{pk}_R, \mathsf{pk}_S, c)$.

Let us give some intuition about the construction before we move on to formalities. Choosing one of the receivers for DRE to be the sender itself and having them encrypt a message under its own key might seem counterintuitive at first, but has one crucial benefit: It guarantees to the other (actual) receiver that even if the sender might not have constructed the ciphertext themselves but rather copied it from somewhere else, they have knowledge about the plaintext since they are able to decrypt as well. This is guaranteed by the registration with $\mathcal{F}_{\mathsf{KRK}}^{\mathfrak{f}_{\mathsf{Key}}}$ in conjunction with the soundness property of the underlying DRE scheme.

Lemma 4 In the $\mathcal{F}_{\text{KRK}}^{\mathsf{f}_{\text{Key}}}$ hybrid model (Gen, Enc, Dec) is an IND-SB-CPA secure SBE scheme.

Proof Assuming that (Gen, Enc, Dec) is a sound DRE scheme with key function f_{Key} and assuming we have an adversary $\mathcal{A}_{\text{SB-CPA}}$ who has non-negligible success probability in winning the IND-SB-CPA game with respect to (Gen, Enc, Dec), we construct an adversary $\mathcal{A}_{\text{DRE-CPA}}$ with non-negligible success probability in winning the DRE IND-CPA game with respect to (Gen, Enc, Dec). Note that in this case, $\mathcal{A}_{\text{DRE-CPA}}$ not only fields $\mathcal{A}_{\text{SB-CPA}}$'s queries to $\mathcal{O}_{\text{SB-CPA}}$ but also plays the role of $\mathcal{F}_{\text{KRK}}^{\mathsf{f}_{\text{Key}}}$ and has therefore access to registered keys. In the reduction shown in Figure 5.2 we do not explicitly state this, but all interactions with $\mathcal{F}_{\text{KRK}}^{\mathsf{f}_{\text{Key}}}$ are handled exactly as the functionality itself would. The only exceptions are that an instantaneous *ok* is assumed whenever the functionality would ask the adversary for some permission and that in the first phase the adversary $\mathcal{A}_{\text{DRE-CPA}}$ itself "registers" the keys pk_S and pk_R for S and R respectively without providing corresponding secret keys.

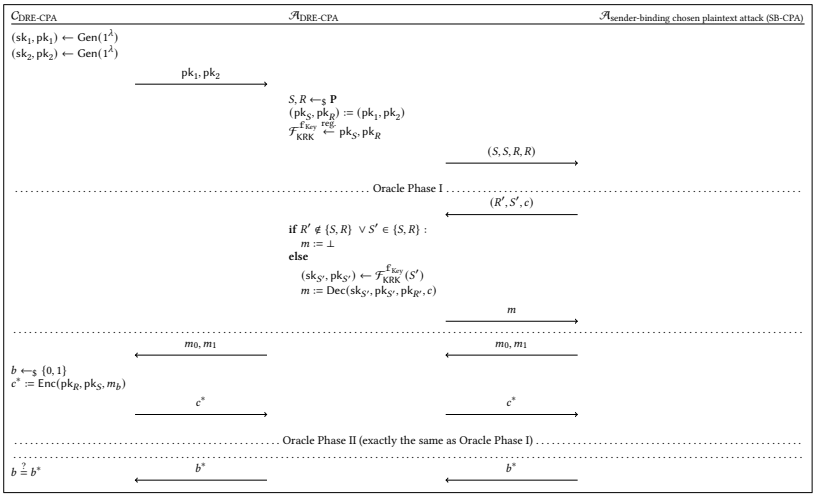

Figure 5.2.: Reduction for DRE Construction

Since $\mathcal{A}_{\text{DRE-CPA}}$ has access to the internal state of $\mathcal{F}_{\text{KRK}}^{\mathsf{f}_{\text{Key}}}$, they can look up the keys $(\mathsf{sk}_{S'}, \mathsf{pk}_{S'})$ for any oracle query (R', S', c). If no such keys have been registered, decryption of the ciphertext would result in $\perp$. If keys have

been registered, they can be used to correctly decrypt the ciphertext as the soundness of DRE (see Definition 34 for definition) guarantees

$$\mathsf{Dec}(\mathsf{sk}_{S'}, \mathsf{pk}_{S'}, \mathsf{pk}_{R'}, c) = \mathsf{Dec}(\mathsf{sk}_{R'}, \mathsf{pk}_{R'}, \mathsf{pk}_{S'}, c).$$

Hence it is no problem for $\mathcal{A}_{\text{DRE-CPA}}$ to respond with correct decryptions exactly as $O_{\text{SB-CPA}}$ would. This gives $\mathcal{A}_{\text{DRE-CPA}}$ the same non-negligible success probability as $\mathcal{A}_{\text{SB-CPA}}$. □

5.2.5. DRE Constructions from McEliece, LPN and LWE

In this section we present an efficient way to construct an IND-CPA secure and sound DRE schemes from the McEliece and LPN assumptions.

5.2.5.1. McEliece Based IND-CPA Secure DRE

Our DRE scheme can be seen as an augmentation of a construction from Kiltz et al. [185]. In this the authors propose a creative construction of a low-noise LPN-based TBE scheme, which they show to be IND-stag-CCA secure. In the appendix of [185] the authors introduce a simplified variant of their IND-stag-CCA secure construction, which is only IND-CPA secure. We use this simplified variant as a basis for our own construction. In order to establish the soundness property we add a second encryption of the randomness and exploit the randomness recovery to perform the consistency check. Moreover, we change the trapdoor mechanism to the one from the McEliece cryptosystem over Goppa codes. Note also, that this DRE scheme admits an efficiently computable function $\mathfrak{f}_{\text{Key}}$ as required for the use with $\mathcal{F}_{\text{KRK}}$ (cp. Section 5.2.4):

$$\mathfrak{f}_{\text{Key}} : ((S, P, G'), (G, C)) \mapsto \begin{cases} \mathit{true}, & G = SG'P \\ \mathit{false}, & \text{else.} \end{cases}$$

In conjunction with Theorem 5 and Theorem 6 our DRE scheme satisfies all requirements for the generic transformation to IND-SB-CPA given in Section 5.2.4. Hence we can use it to efficiently achieve SMT if combined with authenticated channels. In Appendix A.2 we provide for the sake of completeness an alternative way to construct an IND-CPA secure DRE based on similar assumptions following the k-repetition paradigm from [132] and prove its soundness as well.

Parameters:

- Let $\mathcal{G}_{n,t}$ be the family of irreducible binary Goppa-codes of length n, which can correct up to t errors with a code dimension l.
- Let $\theta = \frac{t}{n} + \epsilon$ be the Bernoulli parameter of the error for some $\epsilon > 0$.
- Let $G_2 \in \{0,1\}^{l \times n}$ be the publicly known generator matrix of a code from $\mathcal{G}_{n,t}$, where *Correct* is the according error-correcting algorithm.
- Let $M = \{0,1\}^l$ be the message space.

McEliece DRE Cryptosystem: We define (Gen, Enc, Dec) as follows.

- The key generation algorithm $\mathsf{Gen}(1^n)$ works as follows:
  - Sample a random Matrix $C \in \{0,1\}^{l \times n}$
  - Sample a generator matrix $G' \in \{0,1\}^{l \times n}$ for a code from $\mathcal{G}_{n,t}$.
  - Sample a random non-singular matrix $S \in \{0,1\}^{l \times l}$.
  - Sample a random permutation matrix $P \in \{0,1\}^{n \times n}$.
  - Set $G := SG'P$.

  $\hookrightarrow$ Return $\mathsf{pk} = (G, C, t)$ and $\mathsf{sk} = (S, G', P)$
- The encryption algorithm $\mathsf{Enc}(\mathsf{pk}_R, \mathsf{pk}_S, m)$ works as follows:
  - Parse pk_R as (G_R, C_R, t) and pk_S as (G_S, C_S, t)
  - Sample $s \leftarrow_{\$} \{0,1\}^l$
  - $e_R, e_S, e \leftarrow \mathcal{B}_\theta$
  - $c_R = s \cdot G_R \oplus e_R$
  - $c_S = s \cdot G_S \oplus e_S$
  - $c' = s \cdot C_S \oplus e \oplus m \cdot G_2$

  $\hookrightarrow$ Return $c = (c_R, c_S, c')$.
- The decryption algorithm $\mathsf{Dec}(\mathsf{sk}_R, \mathsf{pk}_S, c)$ works as follows:
  - Parse c as (c_R, c_S, c') and sk_R as (S_R, G'_R, P_R)
  - Compute $\hat{y}_R = c_R \cdot P_R^{-1} = (s \cdot S_R) \cdot G'_R \oplus e_R \cdot P_R^{-1}$

- Compute $s \cdot S_R = Correct(\hat{y}_R)$
- Compute $s = (s \cdot S_R)S_R^{-1}$
- Compute $c'_S = s \cdot G_S$
- Set the verification bit b as follows
  - Set $b = 1$ if the hamming weight of $c'_S \oplus c_S$ is smaller than t.
  - Set $b = 0$ otherwise.

↪ If $b = 0$ return $\perp$, otherwise:

- Compute $c' = s \cdot C_S$
- Correct the error from $m = Correct(c \oplus c')$, where $c \oplus c' = (m \cdot G_2 \oplus e)$.

↪ Return m.

Theorem 5 The DRE scheme (Gen, Enc, Dec) is IND-CPA secure, given that both the McEliece assumption and the learning parity with noise decisional problem (LPNDP) hold. In particular, let $\mathcal{A}$ be an IND-CPA adversary against the cryptosystem. Then there is a distinguisher $\mathcal{B}$ for Goppa codes and a distinguisher $\mathcal{D}$ for the LPNDP, such that for all $\lambda \in \mathbb{N}$

$$\mathsf{Adv}^{\mathrm{CPA}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{LPNDP}_{\theta}(3n,l)}_{\mathcal{D}}(\lambda) + 2 \times \mathsf{Adv}^{\mathrm{ind}}_{\mathcal{B}_R,G_R}(\lambda).$$

Proof **Game 1** This is the DRE IND-CPA game. The challenge ciphertext will be of the form:

$$c^* = (s \cdot G_S \oplus e_S, s \cdot G_R \oplus e_R, s \cdot C_S \oplus e \oplus m_b \cdot G_2)$$

Let $G^* := (G_S|G_R|C_S) \in \{0,1\}^{l \times (3 \cdot n)}$ and $e^* := (e_S|e_R|e) \in \{0,1\}^{3 \cdot n}$. In order to simplify the understanding of the transitions to the next games we rewrite c^* into the following form, where 0 has dimension $2 \cdot n$.

$$c^* = (s \cdot G^* \oplus e^* \oplus (\mathbf{0}, m_b \cdot G_2))$$

Game 2 Same as Game 1, except that the generator matrix G_R within the public key is replaced by uniformly random matrix $U_R \in \{0,1\}^{l \times n}$. Therefore, the receiver public key in Game 2 is $pk_R := (U_R, C_R, t)$.

Any distinguisher $\mathcal{A}_R$ distinguishing between Game 1 and Game 2 yields a distinguisher $\mathcal{B}_R$ for a random irreducible Goppa code from a random linear code. Therefore,

$$\mathsf{Adv}^{CPA}_{\mathcal{A}} \leq \mathsf{Adv}^{CPA}_{\mathcal{A},\text{Game 2}} + \mathsf{Adv}^{ind}_{\mathcal{B}_R,G_R}(\lambda)$$

Game 3 Same as Game 2, except that the generator matrix G_S within the public key is replaced by uniformly random matrix $U_S \in \{0,1\}^{l \times n}$. Therefore, the sender public key in Game 3 is $pk_S := (U_S, C_S, t)$.

Any distinguisher $\mathcal{A}_S$ distinguishing between Game 2 and Game 3 yields a distinguisher $\mathcal{B}_S$ for a random irreducible Goppa code from a random linear code. Therefore, w.l.o.g

$$\mathsf{Adv}^{CPA}_{\mathcal{A}} \leq \mathsf{Adv}^{CPA}_{\mathcal{A},\text{Game 3}} + 2 \times \mathsf{Adv}^{ind}_{\mathcal{B}_R,G_R}(\lambda) \tag{5.1}$$

Game 4 Instead of computing the challenge ciphertext as

$$c^* = (s \cdot G^* \oplus e^* \oplus (\mathbf{0}, m_b \cdot G_2))$$

the challenger chooses $c^* \leftarrow_{\$} \mathcal{U}_{3 \cdot n}$ instead. We justify this replacement by observing that $(s \cdot G^* \oplus e^*)$ is an instance of the LPNDP and therefore can be replaced by a random value $u \leftarrow_{\$} \mathcal{U}_{3 \cdot n}$. The random vector u acts as a One-Time Pad s.t. the ciphertext is transformed into a uniformly distributed random value:

$$c^* = (u \oplus (0, m_b \cdot G_2))$$

This is the challenge ciphertext used in Game 4. The advantage of the original DRE IND-CPA adversary $\mathcal{A}$ is now 0, as the succeeding probability is $\frac{1}{2}$. The indistinguishability follows from the hardness of the LPNDP.

If the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has non-negligibly different succeeding probabilities in Games 3, 4 then we can use this adversary to solve any LPNDP. To this end we can use the following distinguisher D for a given LPNDP oracle O, which is either $Q_{s,\theta}$ with $s \in \{0,1\}^l$ or $\mathcal{U}_{l+1}$ by issuing $(3 \cdot n)$ number of queries.

1. Generate the public keys U_R, U_S, C_S for $\mathcal{A}$.
   - Query the LPNDP oracle: $(a_1, b_1), \cdots, (a_{3\cdot n}, b_{3\cdot n}) \leftarrow O$
   - Set $U_R = (a_1^t, \cdots, a_n^t)$, $U_S = (a_{n+1}^t, \cdots, a_{2n}^t)$ and $C_S = (a_{2n+1}^t, \cdots, a_{3n}^t)$.
2. $(m_0, m_1) \leftarrow \mathcal{A}_1(U_S, C_S, U_R)$
3. $b \leftarrow_\$ \{0, 1\}$
4. Set the challenge ciphertext to

$$c^* = ((b_1, \cdots, b_{3\cdot n}) \oplus (\mathbf{0}, m_b \cdot G_2))$$

5. $b' \leftarrow \mathcal{A}_2(U_S, C_S, U_R, c^*)$
6. If $b' = b$ then return 1, else return 0.

If $O = Q_{s,\theta}$, then we have the same situation as in Game 3, else $O = \mathcal{U}_{l+1}$ and we have the same situation as in Game 4. Therefore:

$$\mathsf{Adv}^{CPA}_{\mathcal{A},Game3} \leq \mathsf{Adv}^{CPA}_{\mathcal{A},\text{Game 4}} + \mathsf{Adv}^{LPNDP}_{\mathcal{D}}(\lambda) = \mathsf{Adv}^{LPNDP_\theta(3n,l)}_{\mathcal{D}}(\lambda)$$

This concludes that the overall advantage is

$$\mathsf{Adv}^{CPA}_{\mathcal{A}} \leq \mathsf{Adv}^{LPNDP_\theta(3n,l)}_{\mathcal{D}}(\lambda) + 2 \times \mathsf{Adv}^{ind}_{\mathcal{B}_R,G_R}(\lambda) \qquad \square$$

The definition of the soundness property of DRE can be found in Definition 34.

Theorem 6 The encryption scheme (Gen, Enc, Dec) satisfies DRE-soundness.

Proof If the sender and the receiver are able to extract the same randomness s, then they will extract the same message m from the ciphertext due to the determinism of the decryption.

Now, consider the case $\mathsf{Dec}(\mathsf{pk}_S, \mathsf{sk}_R, c) = \perp$ and $\mathsf{Dec}(\mathsf{pk}_R, \mathsf{sk}_S, c) = m$. We will prove by contradiction that this case never happens. Parse c as (c_R, c_S, c')

and $\mathsf{pk}_R = (G_R, C_R)$ and $\mathsf{pk}_S = (G_S, C_S)$, where the first two parts of the ciphertext have the following form due to being textbook McEliece ciphertexts.

$$c_R = s' \cdot G_R \oplus e_R$$
$$c_S = s \cdot G_S \oplus e_S$$

From $\mathrm{Dec}(\mathsf{pk}_S, \mathsf{sk}_R, c) = \perp$ it follows that the verification step has failed. This means that after recovering the randomness s' from c_R by $recover(\mathsf{sk}_R, c_R) = s'$, where $recover$ is the textbook McEliece decryption, the hamming distance of $s' \cdot G_S$ has to be greater or equal than t to c_S. Considering that $s' \cdot G_S \oplus c_S = s'G_S \oplus sG_S \oplus e_S$ we get

$$wgt(s'G_S \oplus sG_S \oplus e_S) \geq t$$

From this it follows that $s' \neq s$ due to e_S being guaranteed to have the hamming weight $wgt(e_S) < t$ by the syndrome decoding algorithm within the textbook McEliece decryption.
However, from $\mathrm{Dec}(\mathsf{pk}_R, \mathsf{sk}_S, c) = m$ it follows that

$$wgt(sG_R \oplus s'G_R \oplus e_R) < t$$

Now, sG_R and $s'G_R$ are codewords for $s \neq s'$ and therefore are guaranteed to have hamming distance $d(sG_R, s'G_R) \geq 2t + 1$. This contradicts with $wgt(sG_R \oplus s'G_R \oplus e_R) < t$ as $wgt(e_R) < t$. Therefore, this case is not possible. Similar considerations will yield that the case $\mathrm{Dec}(\mathsf{pk}_S, \mathsf{sk}_R, c) = m_R$ and $\mathrm{Dec}(\mathsf{pk}_R, \mathsf{sk}_S, c) = m_S$ with $m_R \neq m_S$ is impossible.
Conclusively, $\Pr[\mathrm{Exp}^{\mathrm{sound}}_{\mathcal{A},\Pi} = 1] = 0$. □

Discussion. Considering that one of the third round finalists of the post-quantum cryptography (PQC) standardization by the NIST[6] is a McEliece variant based on Goppa codes we expect this mechanism to have significantly better parameters than cryptosystems that are based solely on the (low noise) LPN assumption. We argue, however, that our construction may as well be realized with the sole (low noise) LPN assumption or the Niederreiter cryptosystem [230]. Also, a similar augmentation of the randomness recovering variant of the dual Regev [160] cryptosystem may yield a very similar construction of DRE based on LWE. Currently, the Niederreiter cryptosystem

[6] National Institute of Standards and Technology

seems the most promising as it was already shown in [152] that the trapdoor function is one-way under k-correlated input. The tightness loss is expected to be a factor of 3 regarding the number of LPNDP samples and a factor of 2 regarding the indistinguishability assumption. Therefore, we expect our construction of DRE to have roughly the same parameters as their single receiver IND-CPA counterparts without the soundness. An algebraic comparison of the public keys and the ciphertext from our work and the current state of the art in [185] and [311] can be found in the table 5.1.

Construction	Public Key	Ciphertext
Kiltz et al. [185]	$(A, B_0, B_1, C) \in (\mathbb{Z}_2^{m \times n'})^3 \times \mathbb{Z}_2^{l' \times n'}$	$(c, c_0, c_1, c_2) \in (\mathbb{Z}_2^m)^3 \times \mathbb{Z}_2^{l'}$
Yu et al. [311]	$(A, B_0, B_1, C) \in \mathbb{Z}_2^{\overline{n} \times \overline{n}} \times (\mathbb{Z}_2^{q \times \overline{n}})^2 \times \mathbb{Z}_2^{\bar{l} \times \overline{n}}$	$(c, c_0, c_1, c_2) \in \mathbb{Z}_2^{\overline{n}} \times (\mathbb{Z}_2^q)^2 \times \mathbb{Z}_2^{\bar{l}}$
This Work	$(G, C) \in \mathbb{Z}_2^{l \times n} \times \mathbb{Z}_2^{l \times n}$	$(c_R, c_S, c') \in (\mathbb{Z}_2^n)^3$

Table 5.1.: Comparison of public keys and ciphertext between [185, 311] and this work.

At this point some remarks are necessary to understand the comparisons more thoroughly. For the sake of simplicity we will give rough estimations of the respective public key sizes. Kiltz et al. [185] require for their dimensions that $m \geq 2n'$ and $l' \geq m$, where n' is the dimension of the low-noise LPN secret. Current estimations suggest that cryptosystems based on low-noise LPN to have rather large dimensions, e.g., [120] suggest for 80 bits of security $n' = 9000$ when the noise is $\mu = 0.0044$. Therefore, setting $n' = 9000$ leads to the smallest possible $m = 18000$ and $l' = 18000$ and results in a public key size of roughly 77 megabyte.

Yu et al. [311] improved the construction of [185] in such a way that it may be based on constant noise LPN assuming sub-exponential hardness. Current estimations of concrete constant noise LPN hardness suggest much smaller dimensions than in the low-noise variant, e.g., [55] suggest for 80 bits of security $\overline{n} = 1280$ and noise level of $\mu = 0.05$, which meets the restriction from [311] that $\mu \leq 0.1$. The crucial parameter is, however, the choice of an $\alpha > 0$ as this parameter controls the dimension $q = O(\overline{n}^{6 \cdot \alpha + 1})$, which means that minimizing α will minimize the size of the public key. In order to estimate α as small as possible we take the formula $\beta = \frac{1}{2} - \frac{1}{\overline{n}^{3 \cdot \alpha}}$, which controls the number $\beta \cdot q$ of bit flipping errors that a suitable error correcting code will correct. For the sake of simplicity we set $\alpha = 0.04$, which is almost the minimal possible α for an $\overline{n} = 1280$, and get approximately $q = 7127$. Finally,

fixing the remaining dimension $\bar{l} = \bar{n}$ we get a public key size of roughly 2.5 megabyte, which is a substantial improvement compared to [185].

For classic McEliece constructions Bernstein et al. [39] suggests for 80 bits of security to utilize [1632, 1269] Goppa codes. Setting $n = 1632$ and $l = 1269$ in this work leads to a public key size of roughly 505 kilobytes, which is roughly factor 5 smaller than previous works.

We would like to point out that constructions from [185] and [311] are not directly comparable to our construction because we rely on the additional indistinguishability assumption of Goppa codes from random linear codes. However, all three constructions are code-based and implement a secure channel such that (rough) estimations of concrete sizes regarding the same security level may help to understand the differences.

5.2.6. IND-CPA DRE via LWE-Based Binding Encryption

Let $PKE_{\mathrm{LWE},2}$ be a binding encryption scheme from [229] with the restriction of having two receivers, where one is the sender. The authors prove that this DRE scheme satisfies IND-CPA security and the notion of *strong decryption consistency*. However, if we have only two public keys, sender and receiver, the experiment for strong decryption consistency becomes identical to the soundness experiment from [109]. Therefore, $PKE_{\mathrm{LWE},2}$ is also a dual-receiver encryption scheme and as such it can be used to realize sender-binding chosen plaintext attack (SB-CPA) secure encryption. In fact, from the perspective of size-efficiency of the ciphertext, $PKE_{\mathrm{LWE},2}$ is so far the most efficient LWE-based CPA secure DRE construction. The other works either directly construct a less efficient CCA2 secure DRE [198, 312] or concentrate on IND-ID-CPA-secure IBE-DRE constructions [198, 199, 312]. Moreover, none of these works prove the soundness property of DRE introduced by [109].

Note, that $PKE_{\mathrm{LWE},2}$ does not directly surpass prior standard model CCA2 lattice-based constructions in terms of efficiency. Recently, Boyen et al. [62] presented an efficient lattice-based CCA2 secure KEM construction in the standard model, which the authors compare to other efficient constructions from [212] and conclude that their construction surpasses these in efficiency, mainly by not requiring signatures or MACs.

While the LWE-based CPA secure DRE $PKE_{\mathrm{LWE},2}$ may be inferior in terms of efficiency to [62, 212], we would like to point out that Boyen et al. [62] do

not base their KEM on plain LWE but rather on SISnLWE, which they show to be reducible to LWE but do not provide a discussion about the tightness of the reduction.

5.3. Sender-binding Key Encapsulation

This section is based on joint works with Dr. Rebecca Schwerdt, Astrid Ottenhues, Roland Gröll, and Prof. Dr. Jörn Müller-Quade. In these works we devised a security and encryption definition that, in conjunction with authenticated channels, constructs secure channels without the need for a RO and is weaker than all previously known public-key encryption security definitions, with CCA2 being currently the most prevalent one. The results of this work were published in the proceedings of PKC 2022 [46] and the full version [43]. Subsequently, Sarai Eilebrecht and Laurin Benz joined the team, and the security notion was adapted to fit into the hybrid encryption paradigm. This adaptation was published in the proceedings of PKC 2023 [33] and the full version [266]. Dr. Rebecca Schwerdt dedicated her dissertation to establishing the theoretical foundation for exploring the definition hierarchy of public-key encryption security. Interested readers are encouraged to refer to her PhD thesis [265] for a comprehensive overview of the foundations, while this work provides brief security proof sketches, constructions, and essential theorems and lemmas.

Key Encapsulation Mechanisms (KEMs) are fundamental cryptographic primitives widely used to establish secure communication channels in conjunction with symmetric cryptography. Recent post-quantum cryptographic standards, such as ML-KEM [226] and Classic McEliece [277], are examples of KEMs designed for encryption. Similarly, the transition of key exchange protocols to post-quantum security involves moving from classic discrete logarithm-based Diffie-Hellman key exchange to key exchanges based on post-quantum KEMs [264].

In light of these developments, it is a natural progression to extend the previously introduced notion of Sender-Binding CPA from Beskorovajnov et al. [43] for use within the KEM-DEM framework [115]. A dedicated adaptation is necessary because the security definition for KEMs differs from that of public key encryption (PKE). Specifically, while an IND-CPA-secure PKE implies an IND-CPA-secure KEM, the reverse is not true. This

distinction underscores the importance of defining sender-binding specifically for KEMs.

In this section, we introduce the security notion of IND-SB-CPA$_{\mathrm{KEM}}$. To establish a foundation, we first define what it means for a KEM to be sender-binding and explain how this concept enhances traditional KEM functionality.

Sender-binding extends the conventional KEM structure by incorporating a party identifier (ID) as input during encapsulation and decryption. This modification ensures that the ciphertext is cryptographically bound to the sender's identity.

This approach is conceptually related to tag-based KEMs (tag-key encapsulation mechanisms (tag-KEMs)), where an additional "tag" is provided as part of the input to the encapsulation and decapsulation algorithms. In tag-based KEMs, the tag is often used to differentiate keys or provide context-specific key encapsulations, such as ensuring distinct keys for different sessions or parties. Similarly, in a sender-binding KEM, the party identifier (ID) serves as a tag that uniquely associates the ciphertext with the sender. The key distinction lies in the semantics of the tag: while a generic tag-KEM may use an arbitrary context-specific value as a tag, the sender-binding KEM explicitly uses the sender's identity as the tag. This ensures not only context-specific key encapsulation but also cryptographic authentication of the sender, providing sufficient guarantees for constructing a secure channel.

In this section, we develop the security notion of IND-SB-CPA$_{\mathrm{KEM}}$ and provide transformations to demonstrate its relationship with other KEM security notions. Before doing so, we first formally define the sender-binding property for KEMs.

Definition 44 (SB-KEM) A *sender-binding key encapsulation mechanism (SB-KEM)* is given by a set of three PPT algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with

$$\mathsf{Gen} : 1^\lambda \mapsto (\mathsf{sk}, \mathsf{pk}), \quad \mathsf{Enc} : (\mathsf{pk}, S) \mapsto (\mathsf{k}, C), \quad \mathsf{Dec} : (\mathsf{sk}, S, C) \mapsto \mathsf{k}$$

such that the correctness property holds, i.e. $\mathsf{k} = \mathsf{Dec}(\mathsf{sk}, S, C)$ whenever $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and $(\mathsf{k}, C) \leftarrow \mathsf{Enc}(\mathsf{pk}, S)$.

Note that so far, this is only the traditional KEM interface enhanced by a party ID as input for encapsulation and decryption. Although the denomination suggests this, the "sender" and "binding" part only become meaningful

with the respective security notion. Any classic KEM instantly satisfies this definition when its input is adjusted to incorporate a party ID, regardless of whether this ID specifies some sender, receiver or just a random party, regardless of whether there is any binding property or the ID can be easily exchanged, even regardless of whether this ID is used at all in the protocol. The intended use, however, is that the sending or encapsulating party inserts its *own* ID upon encapsulation, this ID is then non-malleably bound to an otherwise malleable ciphertext and decryption is only successful if the *same* ID is used. These properties are expressed in the following IND-SB-CPA$_{\text{KEM}}$ notion, which is adapted from the corresponding sender-binding encryption (SBE) notion introduced in [44].

Definition 45 (IND-SB-CPA$_{\text{SB-KEM}}$) An SB-KEM (Gen, Enc, Dec) satisfies *indistinguishability under sender-binding chosen plaintext attack (IND-SB-CPA)* security, iff for any PPT adversary $\mathcal{A}_{\text{SB-CPA}}$ the probability to win the IND-SB-CPA game shown in Figure 5.3 is negligible in λ.

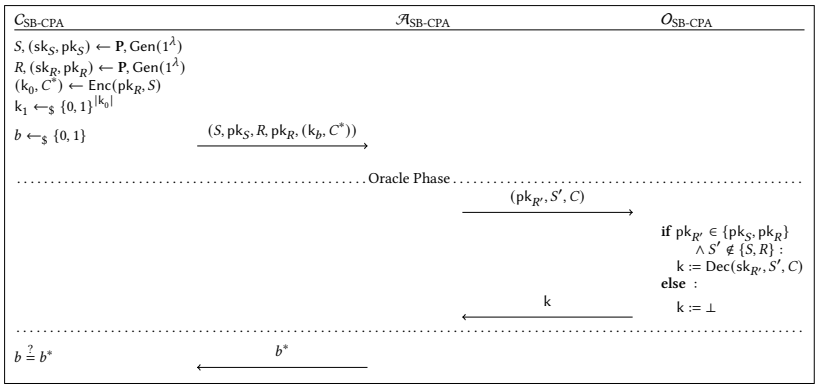

Figure 5.3.: The IND-SB-CPA$_{\text{SB-KEM}}$ Game for SB-CPA$_{\text{SB-KEM}}$

We would like to remark several things about this definition.

- Firstly, IND-SB-CPA$_{\text{KEM}}$ looks very different from other KEM notions at first glance because it has only one oracle phase instead of two. This is not due to less oracle access but because this way is simpler but equivalent: For IND-SB-CPA, the first and second oracle phase permit exactly the same oracle queries (in constrast to CCA2 for instance).

Furthermore in the KEM setting the adversary does not generate any outputs between oracle phases I and II. Hence with IND-SB-CPA$_{\text{KEM}}$ the adversary can save all oracle queries it would make in the first oracle phase and ask them in the second oracle phase instead. We therefore decided to simplify the definition by only including the second oracle phase.

- Secondly, note that although the IND-SB-CPA$_{\text{KEM}}$ security notion contains a key pair $(\mathsf{sk}_S, \mathsf{pk}_S)$ for party S, no such keys need to exist in any protocol. Especially in the session communication setting—but also if communication is one-directional in the single-message setting—only one party needs to have a key pair for the SB-KEM to set up a symmetrically encrypted session. The reason behind the existence of these keys in our security notion is that it makes the notion strictly weaker than if $(\mathsf{sk}_S, \mathsf{pk}_S)$ were not picked by the challenger. Intuitively, an IND-SB-CPA$_{\text{KEM}}$ secure KEM does not need to guarantee anything if S's keys may be adversarially chosen rather than honestly (and secretly) generated. This can clearly be seen when considering the generic DRE construction of an SB-KEM

For this construction each encapsulated key is decryptable by both the receiver *and* sender. Hence the adversary choosing or knowing sk_S would completely break the encapsulation.

Notice that IND-SB-CPA$_{\text{SBE}}$ obviously implies IND-SB-CPA$_{\text{SB-KEM}}$ by the standard PKE to KEM construction of randomly drawing and then encrypting a symmetric key. For KEM security notions, classifying IND-SB-CPA$_{\text{SB-KEM}}$ with respect to classic KEM security is not possible due to important differences in the syntax. While a classic KEM takes no input and requires secrecy and various forms of integrity about the internally determined key, IND-SB-CPA$_{\text{SB-KEM}}$ asserts only secrecy (no integrity) of the key but additionally provides integrity (without secrecy) of some user input–the identity S. Since those two settings are even more incompatible than comparing SBE to classic PKE notions, we will only consider IND-SB-CPA$_{\text{SB-KEM}}$ in relation to the similar setting of tag-KEMs. We refer to the dissertation of Dr. Rebecca Schwerdt [265] for more details about the relationship of IND-SB-CPA$_{\text{SB-KEM}}$ to the various security definitions of tag-KEMs.

5.3.1. Secure Communication from SB-KEM

In this section, we recapitulate the universal composability modeling results from [33]. For the full proof, we refer to the dissertation of Dr. Rebecca Schwerdt [265]. Overall, the results encompass two key functionalities:

- The secure message transfer functionality $\mathcal{F}_{\text{MSMT}}$, which is UC-realized by constructing a hybrid SBE scheme (see theorem 7 and corollary 8).
- The secure channel functionality $\mathcal{F}_{\text{MSC}}$, which is UC-realized by the protocol π_{MSC} (see theorem 9).

The key difference is that, in the real protocol for $\mathcal{F}_{\text{MSMT}}$, we use only ephemeral session keys, whereas in π_{MSC}, a single session key is reused multiple times throughout the session. Both of these protocols can be constructed using the SB-KEMs presented in Section 5.3.3.

5.3.1.1. Construction of a Hybrid SBE Scheme

For $\mathcal{F}_{\text{MSMT}}$, the author restates here that a IND-SB-CPA$_{\text{SB-KEM}}$, an indistinguishability under one-time attack (IND-OT) secure DEM, and authenticated channels are sufficient to realize the secure message transfer functionality $\mathcal{F}_{\text{MSMT}}$. Building on the work of Beskorovajnov et al. [44], who established the feasibility of this approach for IND-SB-CPA secure SBE with authenticated channels, the remaining gap was adressed: combining IND-SB-CPA$_{\text{SB-KEM}}$ with IND-OT$_{\text{DEM}}$ using the KEM-DEM framework results in an IND-SB-CPA secure SBE scheme.

Construction of the Hybrid SBE Scheme For the construction we need two ingredients. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-SB-CPA$_{\text{SB-KEM}}$ secure SB-KEM, and let $(\mathsf{DEM.Enc}, \mathsf{DEM.Dec})$ be a compatible IND-OT secure DEM. Using the KEM-DEM principle, we construct an SBE scheme $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ as follows:

$$\mathtt{Gen} \equiv \mathsf{Gen}$$

The encryption and decryption algorithms are defined as:

$\mathtt{Enc}(\mathsf{pk}_R, S, m)$:
- $(\mathsf{k}, C) \leftarrow \mathsf{Enc}(\mathsf{pk}_R, S)$
- $c \leftarrow \mathrm{DEM.Enc}(\mathsf{k}, m)$

↪ Return (C, c)

$\mathtt{Dec}(\mathsf{sk}_R, S, (C, c))$:
- $\mathsf{k} := \mathsf{Dec}(\mathsf{sk}_R, S, C)$
- $m := \mathrm{DEM.Dec}(\mathsf{k}, c)$

↪ Return m

Theorem 7 (Theorem 4 [265]) The hybrid SBE scheme $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ as defined above is IND-SB-CPA secure.

Corollary 8 Combining the KEM-DEM framework from [268] with the encrypt-then-authenticate protocol from [44], an IND-SB-CPA$_{\text{SB-KEM}}$ secure KEM and IND-OT secure DEM suffice to UC-realize the secure message transfer functionality $\mathcal{F}_{\text{MSMT}}$ in the $\mathcal{F}_{\text{AUTH}}$-hybrid model.

Proof The corollary follows directly from Theorem 7 and [44, Theorem 3].□

5.3.1.2. Construction of a Secure Channel Protocol π_{MSC}

The π_{MSC} protocol facilitates secure communication between two parties in a multi-session setup. Its key feature is the efficient exchange of symmetric keys using a KEM, which enables both parties to communicate securely via the associated DEM. Notably, the KEM only requires credentials from one party, allowing the second party to participate in secure communication without direct involvement in the initial key exchange. The protocol ensures that communication remains confidential even if the authenticated channel is used solely for key exchange, without transmitting the actual messages.

For the purposes of this thesis, the $\mathcal{F}_{\text{MSMT}}$ functionality is fully sufficient. The author, therefore, refers the interested reader to Dr. Rebecca Schwerdt's dissertation [265] or to our original publication [33] for further details. Here, only two key theorems are recapitulated for the sake of completeness.

Theorem 9 (Theorem 2 [33]) Under static corruption the protocol π_{MSC} with IND-SB-CPA secure SB-KEM and IND-CCA2$_{\text{DEM}}$ secure DEM realizes $\mathcal{F}_{\text{MSC}}$ in the $\mathcal{F}_{\text{AUTH}}$-hybrid model. I.e.

$$\pi_{\text{MSC}}^{\mathcal{F}_{\text{AUTH}}} \geq_{\text{UC}} \mathcal{F}_{\text{MSC}}.$$

Theorem 10 (Theorem 3 [33]) Under static corruption the protocol π_{MSC} with IND-SB-CPA secure SB-KEM and IND-RCCA secure DEM with super-polynomial message size realizes $\mathcal{F}_{\text{MSC}}$ in the $\mathcal{F}_{\text{AUTH}}$-hybrid model as well.

Dr. Rebecca schwerdt was able to reduce the necessary requirements on the DEM to merely IND-CPA in [265, Theorem 5], which automatically subsumes Theorems 9 and 10. In the next sections we will describe SB-KEMs that can be used in π_{MSC} as well as in π_{MSMT}.

5.3.2. SB-KEM Constructions from Dual-Receiver KEMs

The SB-KEM construction technique inside this section stems from the master's thesis of Konstantin Gegier [157] that was supervised by Wasilij Beskorovajnov. The original idea was due to Prof. Dr. Jörn Müller-Quade and Wasilij Beskorovajnov. Konstantin Gegier formalized and proved the concept and also came up with quintessential details like the instantiation with Encrypt-with-Hardcode constructions.

In order to obtain SB-KEMs from DRE, we first recall a generic deterministic dual-receiver KEM construction, denoted detDKEM, which is based on a deterministic PKE. Conceptually, detDKEM is the natural dual-receiver extension of the KEM construction of Bellare et al. [31]. In the concrete instantiation constructed by Konstantin Gegier, the underlying deterministic PKE is the *Encrypt-with-Hardcore* scheme of Fuller et al. [154], which is built from trapdoor functions with sufficiently many hardcore bits and analyzed in the min-entropy framework for deterministic encryption.

Parameters:

- Let $\Pi = (GEN^{PKE}, ENC^{PKE}, DEC^{PKE})$ be a deterministic mIND-DE secure PKE.
- Let $\nu, s : \mathbb{N} \to \mathbb{N}$ be length functions.

Dual-receiver KEM detDKEM**:** We define (GEN, ENC, DEC) as follows.

- The key generation algorithm $GEN(1^k)$ works as follows:
  - Sample a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} GEN^{PKE}(1^k)$.
  - $\hookrightarrow$ Return $(\mathsf{pk}, \mathsf{sk})$.
- The encapsulation algorithm $ENC(\mathsf{pk}_1, \mathsf{pk}_2)$ works as follows:

  - Sample

$$R \leftarrow_{\$} \{0,1\}^{\nu(k)} \qquad \text{and} \qquad K \leftarrow_{\$} \{0,1\}^{s(k)}.$$

  - Compute

$$c_1 := ENC^{PKE}(\mathsf{pk}_1, R\|K) \qquad \text{and} \qquad c_2 := ENC^{PKE}(\mathsf{pk}_2, R\|K).$$

  - Set $C := (c_1, c_2)$.

  ↪ Return (C, K).

- The decapsulation algorithm $DEC(\mathsf{sk}_i, \mathsf{pk}_{3-i}, C)$ for receiver $i \in \{1,2\}$ works as follows:

  - Parse C as (c_1, c_2) and let c_i be the component for receiver i and c_{3-i} the other component.

  - Compute a candidate

$$R\|K := DEC^{PKE}(\mathsf{sk}_i, c_i).$$

  - (Soundness check) Compute

$$c'_{3-i} := ENC^{PKE}(\mathsf{pk}_{3-i}, R\|K)$$

    and check whether $c'_{3-i} = c_{3-i}$.

  ↪ If $c'_{3-i} = c_{3-i}$, return K; otherwise return $\perp$.

The soundness lemma from [157, Lemma 1] shows that this check is sufficient. The author restates it only informally here for the sake of brevity. For any adversarially generated encapsulation C, either both receivers output the same key K or both reject. If C is malformed or one component does not decrypt, at least one decryption algorithm immediately returns $\perp$. If both components decrypt but to different plaintexts $M_1 \neq M_2$, then correctness and determinism of the underlying PKE imply that re-encrypting M_1 under pk_2 cannot reproduce c_2 and similarly re-encrypting M_2 under pk_1 cannot reproduce c_1. Hence the soundness check fails for both receivers and both decapsulations return $\perp$.

Security of detDKEM is captured by IND-DKEM, a dual-receiver analogue of indistinguishability for KEMs. The IND-DKEM notion is the straightforward

$A_c^\star(1^k)$	$A_m^\star(1^k, K)$	$A_g^\star(1^k, pk, K, c)$
$K \leftarrow \{0,1\}^{s(k)}$	$R \leftarrow \{0,1\}^{\nu(k)}$	$c_1 \leftarrow c[1]$
return K	$K' \leftarrow \{0,1\}^{s(k)}$	$c_2 \leftarrow c[2]$
	$m_0 \leftarrow (R \,\|\| K,\ R \,\|\| K)^\top$	$b_0 \leftarrow B(pk[1], pk[2], (c_1, c_2), K)$
	$m_1 \leftarrow (R \,\|\| K',\ R \,\|\| K')^\top$	**return** b_0
	return (m_0, m_1)	

Figure 5.4.: The mIND-DE adversary $A^\star = (A_c^\star, A_m^\star, A_g^\star)$.

adaptation of the usual IND-CPA security definition for DRE and the standard IND-CPA security definition for KEMs. For the precise experiment and a detailed discussion we refer to the master's thesis of Konstantin Gegier [157].

Theorem 11 (IND-DKEM security of detDKEM [157]) Let $\nu, s : \mathbb{N} \to \mathbb{N}$. Let $\Pi = (GEN^{PKE}, ENC^{PKE}, DEC^{PKE})$ be an mIND-DE$_2$-secure deterministic PKE. Let detDKEM $= (GEN, ENC, DEC)$ be a DKEM using Π as the underlying PKE. Let B be an IND-DKEM adversary. Then there exists an mIND-DE-adversary

$$A = (A_c, A_m, A_g; A_{\mathsf{nm}}(\cdot), A_\nu(\cdot)) \in \mathcal{A}_{\mathsf{HE}} \cap \mathcal{A}_\lambda \cap \mathcal{A}_2,$$

outputting a single message for each receiver, such that for all $k \in \mathbb{N}$,

$$\mathcal{A}^{\text{IND-DKEM}}_{B,\text{detDKEM}}(k) \ \leq\ \mathcal{A}^{\text{mIND-DE}}_{A,\Pi}(k)\,.$$

The running time of A is that of B.

Proof We construct an mIND-DE-adversary $A^\star$ using B as depicted in Figure 5.4.

$A^\star$ has min-entropy $\nu(k)$ because of the selection of R. Finally, let A be the mIND-DE adversary that works just as $A^\star$, except that A_c returns the empty string and K is replaced by a fixed "best" value. It is straightforward to verify that the following equation holds, since we construct B's input exactly as in the IND-DKEM experiment.

$$\Pr\left[\mathsf{Exp}^{\text{mIND-DE}}_{A^\star,\Pi}(k) = 1\right] = \Pr\left[\mathsf{Exp}^{\text{IND-DKEM}}_{B,\text{detDKEM}}(k) = 1\right] \qquad \square$$

It only remains to show that there are constructions of PKEs that suffice the mIND-DE security. To this end Konstantin Gegier proposes to use the *encrypt-with-hardcore* construction from Fuller et al. [154], which he shows in [157, Theorem 3].

In Section 5.2.4 we showed that an IND-CPA-secure DRE can realize an SB-CPA-secure SBE that in turn yields secure channels in our framework. A straightforward adaptation of our argument applies to the dual-receiver KEM setting.[7] In particular, the IND-DKEM security of detDKEM and the soundness proof implies that detDKEM realizes an SB-KEM according to our definition.

5.3.3. Direct SB-KEM Construction from LWE

To construct protocols that realize the functionalities $\mathcal{F}_{\text{MSMT}}$ (or $\mathcal{F}_{\text{MSC}}$), Theorems 9 and 10 require an IND-SB-CPA secure SB-KEM.

In the following, we construct an LWE-based SB-KEM, which is a simplified version of, to the best of our knowledge, the most efficient IND-CCA2$_{\text{PKE}}$ secure scheme without random oracles to date [64], and demonstrate that it satisfies our IND-SB-CPA$_{\text{SB-KEM}}$ security notion. Our construction is a modified version of the KEM component from [64, 211], where we replace the use of a hash with sender IDs and remove the employed MAC entirely.

This scheme and its proof were developed in close collaboration with the co-author Laurin Benz.

The essential building blocks for this construction include:

- The trapdoor function and gadget matrix G from [211], along with the corresponding invert function.
- A full-rank difference encoding function FRD from [10], which translates sender IDs into suitable matrices.
- A key derivation function (KDF) *kdf*.

[7] Yet again, we refer to Dr. Rebecca Schwerdt's dissertations' [265, Lemma 6] for a formal proof of this claim.

- Gaussian distributions $\mathcal{D}$.

Using these components, we define an SB-KEM $\Sigma := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

$\mathsf{Gen}(1^\lambda)$:

- $A \leftarrow_\$ \mathbb{Z}_q^{n \times m}$
- $R \leftarrow \mathcal{D}^{m \times o}_{\omega(\sqrt{\log(n)})}$
- $A_1 := A \cdot R$

$\hookrightarrow$ Return $(\mathsf{sk}, \mathsf{pk}) := \big(R, (A, A_1)\big)$.

$\mathsf{Enc}(\mathsf{pk}, S) = \mathsf{Enc}\big((A, A_1), S\big)$:

- $e \leftarrow \mathcal{D}^n_{\alpha \cdot q}$; $e_0 \leftarrow \mathcal{D}^m_{\alpha \cdot q}$; $e_1 \leftarrow \mathcal{D}^o_\sigma$, where $\sigma^2 = \big(\|e_0\|^2 + m(\alpha q)^2\big) \cdot \omega\big(\sqrt{\log(n)}\big)^2$.
- $k \leftarrow_\$ \{0,1\}^n$
- $s = k \cdot \lfloor \frac{q}{2} \rfloor + e$
- $c_0 = s^\top A + e_0$
- $c_1 = s^\top (A_1 + \mathsf{FRD}(S)G) + e_1$

$\hookrightarrow$ Return $(\mathsf{k}, C) := \big(kdf(k), (c_0, c_1)\big)$.

$\mathsf{Dec}(\mathsf{sk}, S, C) = \mathsf{Dec}\big(R, S, (c_0, c_1)\big)$:

- $(s, e_0, e_1) \leftarrow \mathsf{invert}\big(R, [A|A_1 + \mathsf{FRD}(S)G], [c_0^\top, c_1^\top]\big)$
- Check $\|e_0\| \leq \alpha q \sqrt{m}$ and $\|e_1\| \leq \alpha q \sqrt{2mo} \cdot \omega\big(\sqrt{\log(n)}\big)$.[a]
- For $i \in \{0, \ldots, n-1\}$: $k[i] := \begin{cases} 0, \text{ if } s[i] \text{ closer to } 0 \\ 1, \text{ if } s[i] \text{ closer to } \frac{q}{2} \end{cases}$.
- Check $\|s - k\| \leq \alpha q \sqrt{n}$.[b]

$\hookrightarrow$ Return $\mathsf{k} = kdf(k)$.

[a] If any check fails, abort with output $\perp$.
[b] abort

The correctness of the scheme directly carries over from the similar scheme in [64] which is why we concentrate on its security properties in this work. The security of Σ is based on the hardness of the NLWE problem. (Cf. Definition 12).

Theorem 12 The SB-KEM $\Sigma = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-SB-CPA secure, given that the LWE assumption holds. In particular, let $\mathcal{A}$ be an IND-SB-CPA$_{\text{SB-KEM}}$ adversary against the SB-KEM, then there are distinguishers $\mathcal{A}_{\text{LWE}}$ for NLWE and $\mathcal{A}_{\text{KDF}}$ for KDF kdf, such that for all $\lambda \in \mathbb{N}$

$$\mathsf{Adv}^{\text{SB-CPA}}_{\mathcal{A},\Sigma}(\lambda) \leq \mathsf{Adv}^{\text{LWE}}_{\mathcal{A}_{\text{LWE}}}(\lambda) + \mathsf{Adv}^{\text{KDF}}_{\mathcal{A}_{\text{KDF}}}(\lambda) + \varepsilon,$$

where ε is negligible in λ.

Proof We roughly follow the proof idea of [64], constructing a series of games which slowly transform the original IND-SB-CPA$_{\text{SB-KEM}}$ game into a one which is obviously unwinnable. At each definition of a new game we show how the adversary's view changes from the last one.

Game 0: This is the IND-SB-CPA$_{\text{SB-KEM}}$ game.

Game 1: At this point $A_1 = AR$ is swapped for $\big(AR - \mathsf{FRD}(S)G\big)$ in the generation of $\mathsf{pk}_R = (A, A_1)$. Since the distributions of AR and $\big(AR - \mathsf{FRD}(S)G\big)$ are both statistically close to uniform randomness over $\mathbb{Z}_q^{n\times o}$ they are by transitivity statistically close to each other. Since FRD is a full-rank difference encoding $\mathsf{FRD}(S') - \mathsf{FRD}(S)$ is invertible if and only if $S' \neq S$. I.e. with the new definition of pk_R decryption of ciphertexts is still possible for any sender ID other than S. As oracle queries with $S' = S$ are not permitted for IND-SB-CPA$_{\text{SB-KEM}}$ anyway, this does not change the oracle at all. Hence the adversary's view in Game 1 is statistically close to the view in Game 0.

Game 2: This game is identical to Game 1, other than the definition of the challenge (c_0^*, c_1^*). Instead of using r we draw a new vector $\overline{c} \leftarrow_{\$} \mathbb{Z}_q^m$ uniformly at random and set $c_0^* := \big(\overline{c} + (k^* \cdot \lfloor \frac{q}{2} \rfloor)^\top A\big)$. For the construction of c_1^* a new random error $\overline{e} \leftarrow \mathcal{D}^{\omega}_{\overline{\sigma}}$ with $\overline{\sigma}^2 = m(\alpha q)^2 \cdot \omega\big(\sqrt{\log(n)}\big)^2$ is drawn and c_1^* set to $c_1^* := \big((c_0^*)^\top R + \overline{e}\big)$. We reduce this change to the hardness of NLWE by showing that from an adversary $\mathcal{A}_{1|2}$ distinguishing Game 1 and Game 2 with non-negligible success probability we can construct an adversary $\mathcal{A}_{\text{LWE}}$ with the same success probability in breaking the NLWE assumption: After getting input

(B, b) from the challenger C_{LWE}, $\mathcal{A}_{\mathrm{LWE}}$ follows Game 1 apart from two definitions. In R's public key $\mathsf{pk}_R = (A, A_1)$ the first value is taken to be $A := B$ which also results in $A_1 = BR$. The value $\bar{c}$ is not drawn randomly but set to b. The rest–including oracle queries–is handled as in Game 1 (which is the same as in Game 2). When $\mathcal{A}_{1|2}$ outputs bit b, which indicates that $\mathcal{A}_{1|2}$ thinks it interacts with Game $(b+1)$, $\mathcal{A}_{\mathrm{LWE}}$ outputs the same b to C_{LWE}.
For the analysis of the reduction firstly note that the distribution of the public key A has not changed at all. In case b is of the form $b = x^\top B + y$, we have

$$c_0^* = \left(b + (k^* \cdot \lfloor \frac{q}{2} \rfloor)^\top A\right) = (k^* \cdot \lfloor \frac{q}{2} \rfloor + x)^\top A + y \sim s^\top A + e_0 \tag{5.2}$$

$$c_1^* = (c_0^*)^\top R + \bar{e} \overset{(5.2)}{\sim} (s^\top A + e_0)^\top R + \bar{e} \overset{(*)}{\sim} s^\top \left(A_1 + \mathsf{FRD}(S)G\right) + e_1,$$

where the second statistic closeness $(*)$ is gained by adapting Theorem 3.1 of [234] and Corollary 3.10 of [246]. This means the view of $\mathcal{A}_{1|2}$ is statistically close to Game 1 if b is an NLWE sample. If, on the other hand, $b \leftarrow_{\$} \mathbb{Z}_q^m$ is random, $\bar{c}$ and hence (c_0^*, c_1^*) is obviously distributed the same as in Game 2.

Game 3: Instead of the construction via $\bar{c}$ from Game 2, c_0^* is drawn uniformly at random from $\mathbb{Z}_q^m$. This means the challenge ciphertext C^* is now completely independent of the key k_0. As the value $\bar{c}$ acted as a one-time-pad on $\left((k^* \cdot \lfloor \frac{q}{2} \rfloor)^\top A\right)$ to define c_0^* in Game 2, the statistical view of the adversary does not change by this modification.

Game 4: As the last step, the key k_0 is drawn uniformly at random rather than generated via the KDF as $kdf(k)$. It is obvious that with this change, an adversary distinguishing Game 3 and Game 4 can be used to directly construct a KDF distinguisher with the same success probability.

In Game 4 we see that the adversary is tasked to decide which of two randomly drawn keys k_0 and k_1 it was sent while the rest of its view is completely independent of these keys. This gives the adversary an even one half chance to win Game 4 and overall provides us with the inequality claimed in Theorem 12.□

5.3.4. Direct SB-KEM Construction from Ring-LWE

The following SB-KEM is an adaptation of the one proposed by Benz et al. [33] to the ring setting by Robert Brede. The results presented in this section stem from his master's thesis [68] that was supervised by Wasilij Beskorovajnov and Laurin Benz.

There are two variants of the construction: the *computational variant* and the *statistical variant*. Both variants are secure against computational attackers, but the public key is either computationally or statistically indistinguishable from uniform, depending on the variant. This yields different bounds on the advantage of attackers on the scheme.

For the description of the scheme we need the following building blocks.

- Let $K = \mathbb{Z}[X]/f(X)$ be a cyclotomic number field with dimension n. We denote its ring of integers under the canonical embedding as $H_K \subset H$ and the corresponding lattice as Λ.
- Let B_Λ be the basis of the lattice Λ, which is also the change of basis matrix from the coefficient embedding to the canonical embedding.
- Let q be a modulus, $k = \lceil \log q \rceil$ and $H_q = H_K/qH_K$. Let H_q^* be the set of invertible elements of H_q. For a coefficient vector $x \in \mathbb{Z}_q^n$, we denote with $\lfloor x \rceil_{q/2}$ rounding each coefficient to 0 or $q/2$, which ever is closest mod q.
- Let g be the gadget vector with its *Invert* function and Basis B_g as well as its Gram-Schmidt orthogonalization $\tilde{B}_g$
- A full-rank difference encoding function FRD from [10], which translates sender IDs into suitable matrices H_q.
- A key derivation function kdf.

Let $t \in \mathbb{R}^+$, such that for any diagonal matrix Σ,

$$\delta := Pr\left[\|x\|_2 > \|\sqrt{\Sigma}\|_2 \cdot \frac{1}{t} \cdot \sqrt{n} \mid x \leftarrow \mathcal{D}_{\Lambda,\sqrt{\Sigma}}\right] \leq \epsilon(n) \tag{5.3}$$

where $\epsilon(n)$ is negligible in n. For $t \in (0,1)$, δ is negligible in n for any cyclotomic number field as stated in [68][Lemma 14].

For the statistical variant choose an integer $m \geq 2$, $\beta_T > 2n \cdot q^{1/m+2/(mn)}$ and α with $\frac{1}{\alpha} \geq 2\beta_T \cdot ||\tilde{B}_g^T||_{k,\infty} \cdot ||B_\Lambda^{-1}||_2 \cdot \sqrt{n(2n+1)m}$ such that $q \geq \frac{1}{\sqrt{2\pi} \cdot \alpha t \beta_T \sqrt{m}} \cdot \sqrt{n} \cdot \omega(\sqrt{\log n})$. For the computational variant instead choose $m = 2$, $\beta_T \in \mathbb{R}$, the bounds on α and q as in the statistical variant, such that decision-NRLWE$_{q,\chi}$ with $\chi = \mathcal{D}_{\Lambda, t\beta_T}$ is hard.

Recall, that for $e \in H$, Σ_e is the matrix with the squared norms on the diagonal. We define the following SB-KEM $\Gamma := (gen, enc, dec)$:

gen(1^κ):

- $a_0 \xleftarrow{\$} H_q^*$, $a' \xleftarrow{\$} H_q^{m-1}$, $T \leftarrow \mathcal{D}_{\Lambda, t\beta_T}^{k \times m}$
- $a = (a_0, a') \in H_q^m$
- $a_1 = Ta \in H_q^k$
- Return $(sk, pk) = (T, (a, a_1))$

enc($pk = (a, a_1)$, S):

- $\tilde{e} \leftarrow \mathcal{D}_{\Lambda, tq\alpha}$, $e \leftarrow \mathcal{D}_{\Lambda, tq\alpha}^m$, $seed \xleftarrow{\$} \{0,1\}^n$
- $\Sigma_f = t^2\beta_T^2(\sum_{i=1}^m \Sigma_{e_i}) + t^2\beta_T^2 m(q\alpha)^2 I_n$
- $\bar{f} \leftarrow \mathcal{D}_{\Lambda, \sqrt{\Sigma_f}}^k$
- $s = B_\Lambda(q/2 \cdot seed) + \tilde{e} \in H_q$
- $c_0 = as + e \in H_q^m$
- $c_1 = (a_1 + FRD(S)g)s + \bar{f} \in H_q^k$
- $K_0 = KDF(seed)$
- Return $((c_0, c_1), K_0)$

dec($c = (c_0, c_1)$, S, $sk = T$):

- $s = Invert((c_0, c_1), -T, FRD(S))$
- $e = c_0 - as$, $\bar{f} = c_1 - (a_1 + FRD(S)g)s$
- check $||e_i||_2 \leq q\alpha\sqrt{n}$ $(1 \leq i \leq m)$, else return $\perp$
- check $||\bar{f}_i||_2 \leq q\alpha\beta_T n\sqrt{(n+1)m}$ $(1 \leq i \leq k)$, else return $\perp$

- $seed = \frac{2}{q} \cdot \lfloor B_\Lambda^{-1} s \rceil_{q/2}$
- check $||s - \frac{q}{2} B_\Lambda \cdot seed||_2 \leq q\alpha\sqrt{n}$, else return $\perp$
- $K_0 = KDF(seed)$
- Return K_0

To uniformly draw invertible elements in the Gen algorithm, there are two approaches:

- Rejection Sampling: This method involves drawing an element from H_q, checking whether it is invertible, and repeating the process if it is not. Robert Brede provides a detailed explanation of this process in the fundamentals of cyclotomic fields section of his thesis [68].
- Polynomial-Based Sampling: This approach requires knowledge of polynomials f_i that are irreducible in $\mathbb{Z}_q$, such that $f = \prod_i f_i \mod q$ and $f_i \neq f_j$ for $i \neq j$. Non-zero polynomials are sampled from $\mathbb{Z}_q[X]/f_i(X)$ for each i, and the Chinese Remainder Theorem (CRT) is then used to map back from $\mathbb{Z}_q[X]/f_1 \times \cdots \times \mathbb{Z}_q[X]/f_r$ to R_q. The resulting polynomial is invertible by construction. Finally, the polynomials are embedded into H_q via the map σ.

The following theorems summarize the correctness of the SB-KEM and the security.

Theorem 13 The SB-KEM $\Gamma = (\text{Gen}, \text{Enc}, \text{Dec})$ is correct with overwhelming probability over the choices of $e, \tilde{e}, \bar{f}$.

Theorem 14 The statistical variant of the SB-KEM $\Gamma = (\text{Gen}, \text{Enc}, \text{Dec})$ with parameters $(n, q, k, \alpha, \beta_T)$ is IND-SB-CPA$_{SB-KEM}$ secure given the decision-NRLWE$_{q,\chi}$ assumption with $\chi = \mathcal{D}_{\Lambda, tq\alpha}$. Concretely, if there is an attacker $\mathcal{A}$ on the security of the SB-KEM, there exists an attacker $\mathcal{A}_{KDF}$ on the KDF and $\mathcal{A}_{NRLWE}$ on the decision-NRLWE$_{q,\chi}$ assumption with

$$Adv^{IND-SB-CPA_{SB-KEM}}_{\mathcal{A},\Gamma}(\kappa) \leq Adv^{KDF}_{\mathcal{A}_{KDF},KDF}(\kappa) + poly(n, \log q) \cdot Adv^{d-NRLWE_{q,\chi}}_{\mathcal{A}_{NRLWE}}(\kappa) + \epsilon$$

where ϵ is negligible in κ.

The computational variant is IND-SB-CPA$_{SB-KEM}$ secure given the decision-NRLWE$_{q,t\beta_T}$ assumption in addition to those of the statistical variant. Concretely, for every attacker $\mathcal{A}$ on the security of the SB-KEM, there exists an attacker $\mathcal{A}_{KDF}$ on the KDF, $\mathcal{A}_{NRLWE}$ on the decision-NRLWE$_{q,\chi}$ assumption and $\mathcal{A}'_{NRLWE}$ on the decision-NRLWE$_{q,t\beta_T}$ with

$$Adv^{IND-SB-CPA_{SB-KEM}}_{\mathcal{A},\Gamma}(\kappa) \leq Adv^{KDF}_{\mathcal{A}_{KDF},KDF}(\kappa)$$
$$+poly(n, \log q) \cdot Adv^{d-NRLWE_{q,\chi}}_{\mathcal{A}_{NRLWE}}(\kappa) + k \cdot Adv^{d-NRLWE_{q,t\beta_T}}_{\mathcal{A}'_{NRLWE}}(\kappa) + \epsilon$$

where ϵ is negligible in κ.

For the proofs we refer to the master's thesis of Robert Brede [68].[8]

5.3.4.1. Concrete Parameter Selection

Robert Brede, in his thesis [68], outlines how to choose concrete parameters for the SB-KEM scheme Γ described above. The parameters are set as follows:

- Cyclotomic Number Field Choose the cyclotomic number field K as the $(2n)$-th cyclotomic number field, where n is a power of two. The error distributions are translated into the coefficient embedding, as detailed in [68, lemma 15].
- Dimension m
- Modulus q
- Scaling Factor t
- Parameter β_T
- Parameter α

The scaling factor t influences the probability of incorrect decryption. Specifically, it depends on the probability that a ring element, drawn from a discrete Gaussian distribution with index S, has a norm greater than $1/(t \cdot \sqrt{n} \cdot |S|_2)$, formally:

[8] And to the dissertation of Laurin Benz at some point in the future. I wish him good luck!

$$\delta := Pr\left[\|x\|_2 > \frac{1}{t} \cdot \|S\|_2 \cdot \sqrt{n} \mid x \leftarrow \mathcal{D}_{\Lambda,S}\right]$$

where $\Lambda = \sigma(R)$ is the lattice of the canonically embedded ring of integers of K. By [68, lemma 15] we have

$$\delta < (\frac{1}{t} e^{-\frac{1}{2t^2}} \sqrt{e})^n.$$

For concrete parameters, δ is a number that is required to be small enough. We choose experimentally

$$\delta \leq 2^{-64}.$$

This yields the values for t depending on the dimension n summarized in table 5.2.

n	512	1024	2048
t	0.7216	0.7996	0.8566

Table 5.2.: The values for t depending on the dimension n to achieve $\delta \leq 2^{-64}$

Modulus q The modulo q is a critical factor, alongside n, in determining suitable parameter sets. Its choice involves several trade-offs:

- **Size of** q**:** q should be as small as possible for efficiency. However, if q is too small, α becomes too small, making the underlying NRLWE problem easier to solve.
- **FRD Requirement:** The Full-Rank Difference (FRD) requires that $q = 2r + 1 \mod 4r$ for some power of two r. The parameter r impacts the number of available sender IDs, given by $q^{n/r} - 1$. Additionally, since $\phi_{2n} = \prod_{i=1}^{r}(X^{n/r} - s_i)$ for some $s_i \in \mathbb{Z}_q^*$ (as per [68][Theorem 2]), computations in R_q can be performed in $\mathbb{Z}_q[X]/(X^{n/r} - s_i)$, improving performance for larger r [40]. For this scheme, we choose $r = 16$.

The value of q is determined experimentally as well. It is initially set and adjusted depending on whether suitable parameter sets exist.

Parameter α: The parameter α is chosen as small as possible, with its reciprocal given by:

$$\frac{1}{\alpha} = 2\beta_T \cdot \|\tilde{B}_G^T\|_{k,\infty} \cdot \|B_\Lambda^{-1}\|_2 \cdot \sqrt{n(2n+1)m}.$$

Here, $\|\tilde{B}_G^T\|_{k,\infty}$ was determined numerically and consistently yielded $\|\tilde{B}_G^T\|_{k,\infty} = 4$ for all tested q values when orthogonalization was performed in forward order.

Statistical and Computational Variants:

- **Statistical Variant:** β_T is set to the smallest possible value, $\beta_T = 2n \cdot q^{(n+2)/(nm)}$. The parameter m is balanced with q such that decision-NRLWE$_{q,tq\alpha}$ remains hard. Increasing m increases $q\alpha$, making the problem harder but worsening performance by increasing key and cipher sizes.
- **Computational Variant:** β_T can be chosen more freely but must be large enough to ensure that decision-NRLWE$_{q,t\beta_T}$ remains hard. Here, m is fixed to 2, as increasing m only worsens performance.

For both variants the condition

$$q > \frac{1}{\sqrt{2\pi} \cdot \alpha t \beta_T \sqrt{m}} \cdot \sqrt{n} \cdot \sqrt{\log n}$$

needs to be checked after setting the parameters.

The primary determining factor for the parameters is the hardness of the underlying NRLWE assumptions. For the statistical variant, only d-NRLWE$_{q,t\alpha q}$ needs to be hard, while the computational variant also requires d-NRLWE$_{q,t\beta_T}$ to be hard. To estimate the hardness of NRLWE, we use the lattice estimator[9] [14], specifically commit "bfbd74e" with the "MATZOV" cost model [170]. Although the estimator is designed for LWE, we assume no attacks exploiting the additional structure of rings.

[9] `https://github.com/malb/lattice-estimator`

The lattice estimator requires a distribution χ in the coefficient embedding, whereas NRLWE is defined in the canonical embedding. For power-of-two cyclotomics, the discrete Gaussian can be scaled into the coefficient embedding by a factor of $1/\sqrt{n}$. Consequently, we define the adjusted parameters:

$$\alpha_k := \alpha/\sqrt{n}, \quad \beta_{Tk} := \beta_T/\sqrt{n}.$$

Parameter sets for LWE analyzed by Crockett and Peikert [118] are unsuitable for this construction due to their small modulus q and narrow Gaussian error distributions. Conversely, parameter sets proposed by Bossuat et al. [61] for homomorphic encryption based on Ring LWE involve larger modulo values and are better suited for our purposes. Peikert [235] demonstrated that instances are secure against multiple attacks when the standard deviation r in the dual ring $R^\vee$ is greater than two. For power-of-two cyclotomic integers with dimension n, we have $R = nR^\vee$, corresponding to a standard deviation of $2n$. However, in practice, narrower error distributions are commonly used. For instance, homomorphic encryption guidelines [61] recommend a standard deviation of 3.19, regardless of n. For our construction, we target standard deviations around 3.19 for both optimistic and guideline-compliant parameter sets. Additionally, for completeness, we include two parameter sets with the theoretical standard deviation of $2n$.

5.3.4.2. Parameter Sets

Table 5.3 shows the different parameter sets.

As can be seen, the computational variant has much better key sizes than the statistical variant as q and m can be chosen smaller for the same n and $q\alpha$ because β_T is much smaller. The fourth and fifth parameters sets have the more theoretical width of $2n$. The fourth is computational but significantly worse than the other computational variants. Considering the statistical variant, the key sizes are similar, though the estimated security of the fifth is worse due to the larger q.

5.3.4.3. Comparison to existing KEMs

This section compares the key and cipher sizes of the new SB-KEM with those of existing KEMs. As discussed earlier in Section 5.3, the SB-KEM security

n	q ($\log_2 q$)	m	$t\beta_{Tk}$	$tq\alpha_k$	C/S	SecLvl	size(pk) = size(c)	size(sk)
512	0xF3d21 (20)	2	1.67	1.679	C	87 L	28.2 KB	51.2 KB
1024	0xF69A1 (20)	2	1.1	1.1	C	164 L	56.3 KB	102 KB
1024	0x31F7E1 (22)	2	1.996	2.16	C	165 L	67.6 KB	124 KB
1024	0x32C118A19E1 (42)	2	2048	2076	C	146 L	237 KB	452 KB
1024	0x1E36050A42A1 (45)	6	9222	2535	S	130 L	294 KB	1.56 MB
2048	0x1F24D205A1 (37)	5	13056	3.206	S	192 P	397 KB	1.75 MB

Table 5.3.: Parameter sets and their properties. Columns $t\beta_{Tk}$ and $tq\alpha_k$ represent the widths of discrete Gaussians for T and errors, drawn in the coefficient embedding. C/S indicates computational (C) or statistical (S) variant. Seclvl denotes security level in bits, with "L" determined by the lattice estimator (using ring operations) and "P" for parameter sets from Bossuat et al. [61], confirmed by the estimator. Security levels account for polynomial advantage in the scheme's proof, affecting success probability but not runtime. The last two columns combine public/private key and cipher sizes, assuming $\log q$ bits per $\mathbb{Z}_q$ element.

notion is somewhat incomparable to the standard KEM security notion and, at best, can be compared to tag-based KEMs. However, this work focuses on secure channels outside the Random Oracle Model (ROM). Standard notion KEMs achieving, for example, CCA2 security without relying on the ROM can also be used for this purpose. Although Kyber and ML-KEM [226, 279] are proven secure in the ROM, they are included in the comparison due to their performance being the ultimate benchmark for non-ROM schemes intended for real-world secure channels.

To the best of our knowledge, the most efficient non-ROM KEMs currently available are the LPN-based scheme proposed by Xu and Li [306] and the RLWE-based scheme proposed by Yang, Ma, and Zhang [307].

table 5.4 shows the parameters of the different KEMs. As shown, our SB-KEM

Scheme	SecLvl	pk	sk	c
Kyber512 [279]	128	0.8 KB	1.632 KB	0.768 KB
Kyber768 [279]	192	1.184 KB	2.4 KB	1.088 KB
LPN-based [306]	128	50.78 MB	62.50 MB	4.54 KB
RLWE-based [307]	80	1.923 MB	0.96 MB	1.280 MB
Ours	87	28.2 KB	51.5 KB	28.2 KB
Ours	164	56.3 KB	102 KB	56.3 KB

Table 5.4.: Comparison of the key sizes for our construction with existing KEMs.

does not perform as well as Kyber. However, Kyber is highly optimized,

whereas our SB-KEM is not and we believe there is still much improvement possible. On the other hand, when compared to KEMs that do not rely on the ROM, our new SB-KEM performs significantly better. While the LPN-based scheme achieves smaller ciphertext sizes, its key sizes are considerably larger. Our SB-KEM outperforms the RLWE-based scheme in all categories, though the differences are less pronounced.

Overall, the new SB-KEM demonstrates significant improvement over KEMs proven secure without the ROM. Notably, neither the keys nor the ciphertext exceed one megabyte in size.

5.3.5. Implementation of the Ring-LWE based SB-KEM

A proof of concept for the SB-KEM described in section 5.3.4 was implemented in python[10] version 3.12.

The implementation is specifically for the parameter sets. That means it only supports a power of two cyclotomic as number field. The reason for this is that all calculations are performed in the coefficient embedding. More precisely, the used translation of Gaussian distributions to the coefficient embedding requires a power of two cyclotomic. Also Robert Brede has provided an extensive discussion on implementing the various parametrization choices.

5.4. Instantiation of a Secure Message Transfer Protocol from Sender-Binding Encryption without Random Oracles

Beside the construction of an Ring-LWE based SB-KEM Robert Brede has also incorporated the cryptosystem under the supervision of Wasilij Beskorovajnov and Laurin Benz into the protocol description from Section 5.2.2. This section describes the instantiated protocol.

[10] www.python.org

Functionality $\mathcal{F}_{MSMT}$

Provides:
Multi-receiver multi-message multi-sender secure message transfer with constant message size and polynomially many Parties $P \in \mathbf{P}$.
State:
Function $\rho_{Msg} : SID \times MID \rightarrow M \times \mathbf{P}^2$ of pending messages.
Behavior:

- Upon receiving (**send**, sid, R, m) from some party S, draw fresh mid, send (**send**, sid, mid, S, R) to the adversary $\mathcal{A}$ and append $(sid, mid) \mapsto (m, S, R)$ to ρ_{Msg}.
- Upon receiving (**send ok**, sid, mid) from the adversary $\mathcal{A}$, look up $(m, S, R) := \rho_{Msg}(sid, mid)$. If it exists, output (**sent**, sid, S, m) to R.

Figure 5.5.: The Ideal Functionality $\mathcal{F}_{MSMT}$.

In this section, the goal is to UC-realize $\mathcal{F}_{MSMT}$ depicted in fig. 5.5 without Random Oracles. This ideal functionality describes multi-receiver multi-message multi-sender secure message transfer of polynomially many parties. This means that multiple parties can send multiple messages to different receivers and the adversary can neither see the sent messages in plaintext nor change them.

We also denote that the functionality $\mathcal{F}_{\text{MSC}}$ can be realized by a similar protocol, where however, the construction has to follow the requirements from Theorem 9 or Theorem 10.

In Section 5.3.1 it was shown that an SB-KEM can be combined with a DEM to realize $\mathcal{F}_{MSMT}$ using $\mathcal{F}_{AUTH}$ by following the protocol from Section 5.2.3 as is defined in Corollary 8. $\mathcal{F}_{AUTH}$ on the other hand can be canonically realized by an EUF-CMA secure signature scheme combined with $\mathcal{F}_{CA}$, see e.g. [81]. The protocol $\pi_{MSMT}^{\mathcal{F}_{CA}}$ combines the adjusted protocol with the protocols $\pi_{CERT}^{\mathcal{F}_{CA}}$ and $\pi_{AUTH}^{\mathcal{F}_{CERT}}$ to realize $\mathcal{F}_{MSMT}$ using only the ideal functionality $\mathcal{F}_{CA}$. The parameters and the states of each party are described in fig. 5.6, while fig. 5.7 depicts the behavior of each party. The protocol is secure under static corruption as the following lemma summarizes.

Protocol $\pi_{MSMT}^{\mathcal{F}_{CA}}$

Realizes:
Multi-receiver multi-message multi-sender secure message transfer with constant message size.
Parameters:

- An EUF-CMA secure Signature Scheme $\Sigma = (\Sigma.gen, \Sigma.enc, \Sigma.\text{vfy})$
- An IND-SB-CPA$_{SB-KEM}$ Secure SB-KEM $\Gamma = (\Gamma.gen, \Gamma.enc, \Gamma.dec)$
- An IND-OT DEM = (DEM.enc, DEM.dec)

State of party P:

- Function $f_{CRED} : SID \rightarrow (\Gamma.PK, \Gamma.SK)$ of own credentials for encryption.
- Keypair $(sk_\Sigma, vk) \in (\Sigma.VK, \Sigma.SK)$ of own credentials for signing.
- Function $f_{PK} : SID \times P \rightarrow \Gamma.PK$ of known public keys.
- Function $f_{VK} : SID \times P \rightarrow \Sigma.VK$ of known verification keys.
- Function $f_{send} : SID \times P \rightarrow M^*$ of pending messages.

Figure 5.6.: The Setup for the Protocol $\pi_{MSMT}^{\mathcal{F}_{CA}}$, which Realizes $\mathcal{F}_{MSMT}$ using $\mathcal{F}_{CA}$

Lemma 15 Under static corruption, the protocol $\pi_{MSMT}^{\mathcal{F}_{CA}}$ UC-realizes $\mathcal{F}_{MSMT}$ in the $\mathcal{F}_{CA}$-hybrid model.

Proof The lemma follows from (Canetti [79] Claim 3, Claim 4), Corollary 8). □

The real protocol To send a plaintext from A to B:

1. A uses the SB-KEM to encapsulate a fresh symmetric key K, producing the KEM ciphertext c_1.

Protocol $\pi_{MSMT}^{\mathcal{F}_{CA}}$

Behavior of Party P:

- Before signing for the first time: set $(sk_\Sigma, vk) \leftarrow \Sigma.gen(1^\kappa)$ and send $(\textbf{Register}, P, vk)$ to $\mathcal{F}_{CA}$
- Upon receiving $(sid_{AUTH}, S, (\textbf{init}, sid), \sigma)$ with fresh sid_{AUTH}, if there is no entry $f_{Cred}(sid)$ yet:
  1. If $f_{VK}(S)$ does not exists, send $(\textbf{Retrieve}, S)$ to $\mathcal{F}_{CA}$ to obtain $(\textbf{Retrieve}, S, vk)$. If $vk = \perp$, ignore the original message. Else set $f_{VK}(S) = vk$.
  2. Look up $vk_S = f_{VK}(S)$. If $\Sigma.\text{vfy}(vk_S, (\textbf{init}, sid, P, sid_{AUTH}), \sigma) = 0$, ignore the original message.
  3. $(sk, pk) \leftarrow \Gamma.gen(1^\kappa)$
  4. Append $sid \mapsto (sk, pk)$ to f_{Cred}
  5. For each party $P \neq P'$: Draw a fresh sid'_{AUTH}, set $m = (\textbf{inited}, sid, pk)$, $\sigma \leftarrow \Sigma.sign(sk_\Sigma, (m, P', sid'_{AUTH}))$ and send $(sid'_{AUTH}, P, m, \sigma)$ to P'
- Upon receiving $(sid_{AUTH}, P', (\textbf{inited}, sid, pk_{P'}), \sigma)$ with fresh sid_{AUTH}, if there is no entry $f_{PK}(sid, P')$ yet:
  1. If $f_{VK}(P')$ does not exists, send $(\textbf{Retrieve}, P')$ to $\mathcal{F}_{CA}$ to obtain $(\textbf{Retrieve}, P', vk)$. If $vk = \perp$, ignore the original message. Else set $f_{VK}(P') = vk$.
  2. Look up $vk_{P'} = f_{VK}(P')$. If $\Sigma.\text{vfy}(vk_{P'}, (\textbf{inited}, sid, pk_{P'}, P, sid_{AUTH}), \sigma) = 0$, ignore the original message.
  3. Append $(sid, P') \mapsto pk_{P'}$ to f_{Pk}
  4. For any $m \in f_{Send}(sid, P')$
     a) Remove m from $f_{Send}(sid, P')$
     b) $(K, c_0) \leftarrow \Gamma.enc(pk_{P'}, P)$
     c) $c_1 \leftarrow DEM.enc(K, m)$
     d) Draw fresh sid_{AUTH}, set $m' = (sid, (c_0, c_1))$
     e) Draw $\sigma \leftarrow \Sigma.sign(sk_\Sigma, (m', P', sid_{AUTH}))$ and send $(sid_{AUTH}, P, m', \sigma)$ to P'
- Upon receiving input $(\textbf{send}, sid, R, m)$ with $m \in \{0,1\}^n$ from environment $\mathcal{Z}$:
  - If $R = P$ report output $(\textbf{sent}, sid, P, m)$ to the environment
  - Else if no entry $f_{pk}(sid, R)$ exists yet:
    1. Append m to $f_{Send}(sid, R)$
    2. Draw fresh sid_{AUTH}
    3. For $m' = (\textbf{init}, sid)$ draw $\sigma \leftarrow \Sigma.sign(sk_\Sigma, (m', R, sid_{AUTH}))$ and send $(sid_{AUTH}, P, m', \sigma)$ to R
  - Else:
    1. $pk_R = f_{pk}(sid, R)$
    2. $(K, c_0) \leftarrow \Gamma.enc(pk_R, P)$
    3. $c_1 \leftarrow KEM.enc(K, m)$
    4. Draw fresh sid_{AUTH}. For $m' = (sid, (c_0, c_1))$ draw $\sigma \leftarrow \Sigma.sign(sk_\Sigma, (m', R, sid_{AUTH}))$ and send $(sid_{AUTH}, P, m', \sigma)$ to R
- Upon receiving $(sid_{AUTH}, S, (sid, (c_0, c_1)), \sigma)$ with fresh sid_{AUTH}:
  1. If $f_{VK}(S)$ does not exists, send $(\textbf{Retrieve}, S)$ to $\mathcal{F}_{CA}$ to obtain $(\textbf{Retrieve}, S, vk)$. If $vk = \perp$, ignore the original message. Else set $f_{VK}(S) = vk$.
  2. Look up $vk_S = f_{VK}(S)$. If $\Sigma.\text{vfy}(vk_S, (sid, (c_0, c_1), P, sid_{AUTH}), \sigma) = 0$, ignore the original message.
  3. Look up $sk := f_{CRED}(sid)$ and $pk := f_{pk}(sid, S)$. If one of them does not exist, abort.
  4. $K \leftarrow \Gamma.dec(sk, c_0, S)$
  5. $m \leftarrow KEM.dec(K, c_1)$
  6. Report output $(\textbf{sent}, sid, S, m)$ to the environment $\mathcal{Z}$

Figure 5.7.: The Behavior of Each Party in the Protocol $\pi_{MSMT}^{\mathcal{F}_{CA}}$, which realizes $\mathcal{F}_{MSMT}$ using $\mathcal{F}_{CA}$

2. Using K, A encrypts the plaintext with the DEM, obtaining the ciphertext c_2.
3. A signs the tuple (c_1, c_2) with its digital signature scheme and sends (c_1, c_2, σ) to B.

Upon receipt, B verifies the signature. If it is valid, B decapsulates c_1 to recover K and then decrypts c_2 using K. If B has not yet generated an SB-KEM key pair, A first sends a special setup message prompting B to do so. Once generated, the public keys are broadcast to all parties. A certificate authority (CA) allows parties to map a party ID to a verification key, enabling verification of the claimed sender.

Impersonation by re-signing. An attacker might replace the sender ID and the signature. If the message contains a ciphertext, decryption fails because the SB-KEM ties the ciphertext to the original sender ID. Using a different ID breaks decapsulation. If the message has no ciphertext, its contents are public to all parties, so forging or forwarding it yields no advantage.

Relay/Replay attacks. The intended receiver is included in the signed data. If an attacker forwards a valid message to a different recipient, the new recipient rejects it because the signature does not verify for that receiver. Each message carries a signed freshness identifier sid_{AUTH}. Receivers accept only fresh values of sid_{AUTH}, so a replayed message is ignored. If an attacker alters sid_{AUTH}, the signature no longer verifies.

5.4.1. Concrete Instantiation

This section fixes a concrete version of the protocol $\pi_{MSMT}^{\mathcal{F}_{CA}}$ by specifying the remaining cryptographic primitives and by defining the party and message spaces.

The remaining parts are: an EUF-CMA secure signature scheme and an IND-OT data encapsulation mechanism (DEM).

DEMs Two natural DEM options are:

- a one-time pad (OTP), which yields fixed-length messages of n bits; and
- a construction based on a pseudorandom generator (PRG), which can support longer messages.

For simplicity, we use the OTP, resulting in ciphertexts of multiple lengths depending on the chosen parameter set, i.e. $n \in \{512, 1024, 2048\}$. It is obviously not the most efficient choice. There are other DEMs could provide variable message sizes but they would complicate the already complex overall protocol.

Signatures To the best of our knowledge, only a few post-quantum signature schemes avoid the random oracle model. We select SPHINCS-256 [37], a stateless hash-based scheme with public and secret keys of about 1 KB and signatures of roughly 41 KB. SPHINCS+/SLH-DSA [228] is standardized and closely related to SPHINCS-256, but it is not suitable here because its security proof relies on the random oracle model. Another alternative is LMS [112], a stateful hash-based scheme. Since state management adds operational complexity, we do not adopt LMS here and only state that it's one of the options.

Full-Rank Decoding Function (FRD) Let (n, q, r, m) be the parameters of SB-KEM Γ. The message space is fixed by the OTP as

$$M^* = \{0,1\}^n.$$

Party identifiers must fit the input domain of the FRD associated with Γ, namely $\mathbb{Z}_q^{n/r} \setminus \{0\}$. This domain can encode $\log q \cdot n/r$ bits. Hence the party-ID space is

$$\mathbf{P} \subseteq \{0,1\}^{\log q \cdot n/r} \setminus \{0\}.$$

In practice, one can set the party ID to be the hash of a party's email address using a collision-resistant hash function. The all-zero string must not be a valid ID. This can be enforced by the hash function itself or, for example, by appending a trailing 1 bit whenever the hash would otherwise be zero.

5.4.2. Performance Analysis

This section analyzes the performance of the protocol, specifically the message size and required storage. For concrete numbers, we use the second parameter set of $n = 1024, \log q = 20, r = 16, m = 2$ as it provides more than 128 bits of security. Let s be the number of session identifiers and p be the number of parties.

Key Storage Each party has to store s key pairs of the SB-KEM Γ, one key pair from SPHINCS256 and $s \cdot p$ public keys from Γ and verification keys from SPHINCS256. As the plaintexts are only 128 bytes and only stored short term when waiting for the key generation of the receiving party, we ignore them. If the sessions are not long-living then the factor s can be dropped.

This leads to the following formula of required storage of the sessions are long-living and every session creates their own keypair.

$$s \cdot (32.6KB + 56.3KB) + 1.088KB + 1.056KB + s \cdot p \cdot (56.3KB + 1.056KB)$$

Depending on the number of parties, the most dominant part is either the key pairs of Γ for each session or the public key of Γ for each party.

Message Sizes All messages contain sid_{AUTH}, the sender ID, the payload and a signature. The payload always contains the sid. The largest payload is the message containing the ciphers. Assuming 64 bit sids and 256 bit sender IDs, the size of the messages is up to

$$8B + 32B + 56.3KB + 32B + 41KB \approx 97.3KB.$$

Using the smallest parameter set instead, the message sizes are up to 69.3 KB, but the secure channel only provides 87 bits of security.

6. Conclusion and Future Work

6.1. Future Work

The work in this thesis addresses three challenges that, in the author's view, have so far remained insufficiently addressed in real-world cryptographic deployments: internal attackers, quantum-computer-assisted cryptanalysis, and the random oracle methodology. The results presented here should therefore be understood as a starting point rather than an end point. In the following, the author outlines several directions for future research that naturally emerge from the case studies and constructions developed throughout this thesis.

6.1.1. Internal attackers and Outsourced Computation

The case studies on homomorphic-encryption-based outsourced computation and separated-duty contact tracing show that a careful choice of system architecture and trust assumptions can mitigate a number of internal-attacker risks without resorting either to pervasive employee surveillance or to heavily inefficient generic cryptographic constructions. In theory, MPC can realize (almost) everything[1], but in reality the industry is still lagging behind even on comparatively simple end-to-end encryption measures.

Thus, several aspects remain to be explored. First, the server–server non-collusion assumption employed in the outsourcing and contact-tracing protocols merits a more fine-grained investigation. In practice, it is plausible that collusion might occur only partially (e.g., among a subset of administrators) or only during a limited time window or with limited information exchange. Modeling such relaxed or time-bounded collusion assumptions,

[1] Yes, the author is well aware that there are impossibility results.

and quantifying the resulting leakage, would yield a more nuanced picture of the residual risk and might suggest new cryptographic mechanisms for graceful degradation.

Moreover, the outsourced computation model presented in this thesis can be easily generalized to a multiparty variant of the decryptor, in order, for example, to capture Multi-Key Homomorphic Encryption [201] or MPC mechanisms limited to the decryption operation. Despite MPC being in most cases too inefficient to be considered for real-world use, the author believes that constrained-functionality MPC (such as decryption) can work or be sufficiently optimized.

The model is also trivially realizable with a Trusted Execution Environment (TEE) that handles the decryption. See, for example, TEEFHE [298], where such a module is used to bootstrap efficiently. The author believes that this paradigm is already usable within the current model. In fact, it is one of the reasons why this thesis relies only on somewhat/leveled homomorphic encryption and leaves bootstrapping out of scope.

Second, the integration of cryptographic defenses with organizational measures such as auditing, logging, and access control remains largely unexplored in formal models. As argued in Chapter 4, a naïve strengthening of these measures quickly leads to highly privacy-invasive "Zero Trust" solutions that undermine the very protections that privacy-enhancing cryptography is intended to provide. A promising line of work is to extend universally composable functionalities with explicit auditing and accountability features and to design protocols that simultaneously realize privacy and verifiable non-collusion without relying on permanent surveillance of employees.

Third, the formal outsourcing framework in this thesis focuses on equality-filtering algorithms as a representative workload. While the equality kernel captures a wide range of applications, there is growing interest in more complex outsourced computations, including range queries, joins, ranking and top-k queries, and machine-learning inference. Theoretically, they are all realizable within the model, but the main reason why we chose such a simple algorithm for our initial study is that actors from industry described a real problem with genuine privacy concerns, and we were not able to find related work that already suited the adversary model, despite our adversary model being one of the simplest ones and the outsourced computation protocol also being one of the simplest ones, i.e., a single-server secure function evaluation. We were quite baffled at the time by the sheer amount of related work on

outsourced PSI only (cf. Appendix A.1), without finding a single work that fit our needs. The authors impression is that there is currently a large gap between the academic work being done on privacy-preserving cryptography and the real needs of society and industry. The author's suggestion for future work is to align cryptographic research better[2] with these needs, and his hope is that the outsourced computation framework described in this thesis is a step in this direction.

6.1.2. Contact Tracing, Anonymous Credentials, and Post-Quantum Security

The separated-duty contact-tracing protocol illustrates how internal attacker risks can be reduced by distributing trust among multiple entities and by employing anonymous, rate-limited reporting mechanisms. The post-quantum discussion in Section 4.3.5, however, reveals that key building blocks are not yet satisfactorily available in quantum-safe form. In particular, a post-quantum re-randomizable public-key encryption scheme that matches the simplicity and flexibility of ElGamal would be a big win. While instantiations based on lattice-based cryptosystems such as ML-KEM appear promising candidates, their re-randomization properties and correctness guarantees under realistic parameters and their optimizations require dedicated analysis.

A second open problem is the construction of quantum-safe e-token dispensers or, more generally, anonymous credential systems with periodic rate limits and unlinkable tokens. The periodic n-times anonymous authentication scheme used in the contact-tracing protocol is not quantum-safe, and to the author's knowledge no drop-in replacement with comparable functionality existed until very recently, i.e., [272][3].

A similar situation currently arises in the area of anonymous authentication, where it is often argued that we should not wait until sufficiently efficient post-quantum replacement candidates have been developed[4], but instead deploy the existing schemes that are not quantum-safe. The author's opinion on this is ambivalent. On the one hand, there is a strong need for privacy,

[2] Unfortunately, i didn't find the general wisdom on how to do this.
[3] Which is an `arXiv` work that still requires scrutiny.
[4] Bilinear Pairings replacement candidates are hard to find.

despite there being almost no demand for privacy in practice. As explained in the introduction of Chapter 4, privacy is a quintessential requirement for democracy. Thus, methods that preserve privacy share this urgency, and therefore we need them now. On the other hand, the privacy guarantees of non-quantum-safe schemes may be void once a quantum computer becomes available in the future. However, this argument is not fully correct and requires a balanced consideration. Authentication, by its nature, is less prone to the harvest-now-decrypt-later challenge that many cryptographic protocols face today, but it still requires a careful analysis of whether anonymity can be broken post factum, years later, and whether this can be done at scale. Recent advances in post-quantum anonymous credentials and rate-limited tokens provide a starting point, but further work is needed to reconcile efficiency, revocation, strong anonymity, and quantum resistance in a single scheme suitable for large-scale deployments such as contact tracing.

Beyond these cryptographic aspects, future work should examine how the separated-duty architecture generalizes to other applications where internal attackers and privacy concerns coincide, such as whistleblowing platforms, digital exposure notification in non-health settings, or privacy-preserving security logging. Detailed case studies, carried out in collaboration with practitioners and regulators, would help to translate the abstract ideal functionalities into concrete system and policy recommendations.

6.1.3. Sender-binding Security and the Random Oracle Methodology.

The sender-binding security notions introduced in this thesis, together with their instantiations via dual-receiver encryption and sender-binding KEMs, demonstrate that secure message transfer can be realized without relying on the random oracle methodology and with security assumptions strictly weaker than CCA2. At the same time, the current constructions are noticeably less efficient than their random-oracle-based counterparts, and the broader landscape of sender-binding security notions is only beginning to be charted.

One natural direction for future work is to optimize sender-binding KEMs and their underlying primitives. In particular, lattice-based constructions with smaller public keys and ciphertexts, tighter security reductions, and more

efficient implementations would narrow the gap to standardized schemes such as ML-KEM. Similarly, code-based sender-binding constructions leveraging Classic McEliece and related systems deserve further exploration in order to better understand the trade-offs between security, size, and performance in the standard model.

On the foundational side, the definition hierarchy surrounding sender-binding encryption and sender-binding KEMs has already been thoroughly analyzed by Dr. Rebecca Schwerdt in her thesis [265]. Her results show that sender-binding is not only sufficient, but in a precise sense also necessary, and thus essentially minimal for realizing secure message transfer in general.

The author's humble aspiration was to find constructions that could seriously challenge random-oracle-based schemes at the level of concrete efficiency. This has not (yet) led to the results he was hoping for. A key obstacle, which became clearer over the course of this work, is that most existing foundations that can be repurposed for sender-binding encryption were originally developed with the ultimate goal of achieving CCA2-secure encryption. In retrospect, the intuition that sender-binding should be substantially "easier" than CCA2, because we "only" require a non-malleable integration of a sender identity, turned out to be not fully substantiated.

Conceptually, an SB-CPA-secure scheme still has to satisfy a surprisingly demanding bundle of properties:

- (A) ordinary CPA security
- (B) extractability for honest ciphertexts
- (C) sender-identity non-malleability (preventing the attacker from exchanging the sender identity in the challenge ciphertext)

While achieving (A) and (C) is not particularly difficult—and can be done with relatively simple LWE- or McEliece-style templates—the real bottleneck is (B). Honest-ciphertext extractability amounts to a weak, but still nontrivial, form of CCA-style power. Once a construction offers (A)–(C), it is arguably no longer very far from full CCA2 security, as also reflected in the connections to various (tagged) weak CCA notions in the literature (e.g., the atag-/stag-wCCA landscape discussed by Kiltz et al. [184]).

The dual-receiver-encryption-based sender-binding constructions in this thesis are something of an exception: thanks to soundness and knowledge-of-registration-key properties, they circumvent some of the usual extractability

obstacles. From a broader perspective, however, they still illustrate the same phenomenon. Due to the double-encryption overhead, we are still performing a re-encryption check, which is also prevalent in FO-transformed PKEs. Thus, dual-receiver encryption, despite being a notable exception, currently appears to be a dead end for achieving significantly more efficient sender-binding encryption.

For constructing truly efficient sender-binding encryption schemes, the research community will therefore likely have to take a few steps back and develop direct SB-CPA-secure PKE and KEM primitives, rather than deriving sender-binding as a by-product of designs whose original target was CCA2. Identifying such "native" sender-binding constructions that avoid the full complexity of plaintext awareness and classical CCA2 machinery remains, in the author's view, an important and largely open direction.

6.2. Conclusion

The guiding theme of this thesis has been the tension between the elegance of cryptographic theory and the messy realities of information security practice. The author has argued that this tension manifests itself most clearly in the assumptions that underlie our protocols: assumptions about the behaviour of insiders and system operators, about the long-term hardness of mathematical problems in the presence of quantum computers, and about the legitimacy of modeling hash functions as ideal random oracles. These assumptions often remain implicit in real-world deployments, yet they function as Chekhov's guns and Damocles' swords: they sit silently in the background until a new attack or an unforeseen deployment scenario brings them to the foreground.

The first part of the thesis focused on the internal attacker as an often-overlooked threat in industrial practice. While organizational countermeasures such as audits and monitoring are indispensable, they also pose significant privacy risks for employees and users. The author therefore explored how cryptographic techniques can be used to reduce the reliance on such measures by designing systems that remain secure even in the presence of internal adversaries. The homomorphic-encryption-based outsourcing framework introduced in Chapter 4 formalizes a realistic web-service setting with dynamic, one-shot clients, and decoupled computation and output.

Within this framework, the thesis presented efficient outsourced equality-filtering protocols and showed how a two-server, non-colluding architecture can mitigate insider risks compared to traditional single-server designs. The separated-duty contact-tracing protocol analyzed in the same chapter applied similar ideas in a different context, demonstrating how privacy-preserving contact tracing can be achieved by splitting trust among several parties and carefully combining anonymous authentication, re-randomizable encryption, and separation of duties.

The second line of work addressed quantum-computer-assisted cryptanalysis and the resulting need to migrate real-world systems to quantum-safe assumptions. Rather than proposing yet another candidate primitive, the thesis examined how post-quantum building blocks can be integrated into complex protocols such as secure channels and contact-tracing systems. To this end, the author contributed dual-receiver encryption schemes based on McEliece, LPN, and LWE, as well as an LWE-based sender-binding key encapsulation mechanism, and analyzed how these constructions can be used to instantiate secure channels and other higher-level functionalities in a quantum-safe way. The resulting protocols are not only of theoretical interest,but they also illustrate which parts of current real-world systems can already be migrated to post-quantum assumptions and which parts still depend on future progress in post-quantum anonymous credentials and re-randomizable encryption.

The third part of the thesis revisited the random oracle methodology from the perspective of secure channel construction. While the random oracle has enabled some of the most efficient and widely deployed cryptographic schemes, it remains a heuristic that cannot be instantiated in the real world. Rather than rejecting the methodology outright, the author proposed to separate the concerns of efficient design and formal soundness. The sender-binding encryption (SBE) and sender-binding key encapsulation (SB-KEM) notions developed in Chapter 5 provide one concrete step in this direction. They demonstrate that it is possible to realize secure message transfer in the standard model, under assumptions strictly weaker than CCA2, while retaining a modular KEM–DEM structure. The concrete constructions in this thesis—although not yet competitive with ROM-based standards such as ML-KEM in all metrics—show that secure channels without random oracles are not merely a theoretical curiosity, but a feasible design option whose efficiency may improve as the underlying primitives are further optimized.

Taken together, the three strands of work support a broader conclusion about the design of real-world cryptographic systems. Security proofs and definitions remain indispensable, but they are only as meaningful as the assumptions they rest on and the threat models they address. Even the most expressive threat models, and thus the resulting proofs are only as meaningful as the reality in which they are defined. By making the internal attacker explicit, by embedding post-quantum building blocks into concrete protocols, and by offering an alternative to the random-oracle methodology for secure channels, this thesis aims to contribute to a more honest and nuanced treatment of assumptions in both academic research and industrial practice. In each case, the goal was not to provide definitive answers, but to clarify the landscape of risks and to demonstrate that different, and arguably more realistic, choices of assumptions are possible.

Of course, the cycle of cryptography and cryptanalysis is continuous. New attacks may invalidate some of the assumptions used here, just as previous attacks have reshaped our understanding of isogeny-based cryptography or multivariate signatures. The hope is that the frameworks and constructions developed in this thesis will make such evolutions less surprising and less catastrophic by encouraging a more systematic articulation of the Chekhov's guns and Damocles' swords in our designs. Ultimately, real-world cryptography will remain a moving target, but by aligning our cryptographic abstractions more closely with operational realities, we can at least ensure that the systems we build today are better prepared for the challenges of tomorrow.

Bibliography

[1] Dagstuhl Seminar 25112. *PETs and AI: Privacy Washing and the Need for a PETs Evaluation Framework*. 2025. URL: https://www.dagstuhl.de/seminars/seminar-calendar/seminar-details/25112.

[2] DP-3T Project. *Privacy and Security Risk Evaluation of Digital Proximity Tracing Systems*. Apr. 21, 2020. URL: https://github.com/DP-3T/documents/blob/master/Security%20analysis/Privacy%20and%20Security%20Attacks%20on%20Digital%20Proximity%20Tracing%20Systems.pdf.

[3] DP-3T Project. *Security and privacy analysis of the document 'PEPP-PT: Data Protection and Information Security Architecture'*. Apr. 19, 2020. URL: https://github.com/DP-3T/documents/blob/master/Security%20analysis/PEPP-PT_%20Data%20Protection%20Architecture%20-%20Security%20and%20privacy%20analysis.pdf.

[4] DP-3T Project. *Security and privacy analysis of the document 'ROBERT: ROBust and privacy-presERving proximity Tracing'*. Apr. 22, 2020. URL: https://github.com/DP-3T/documents/blob/master/Security%20analysis/ROBERT%20-%20Security%20and%20privacy%20analysis.pdf.

[5] Aydin Abadi, Sotirios Terzis, and Changyu Dong. "O-PSI: Delegated Private Set Intersection on Outsourced Datasets". In: *ICT Systems Security and Privacy Protection*. Ed. by Hannes Federrath and Dieter Gollmann. Vol. 455. IFIPAICT. Springer, Cham, May 2015, pp. 3–17. DOI: 10.1007/978-3-319-18467-8_1.

[6] Aydin Abadi, Sotirios Terzis, and Changyu Dong. "VD-PSI: Verifiable Delegated Private Set Intersection on Outsourced Private Datasets". In: *FC 2016*. Ed. by Jens Grossklags and Bart Preneel. Vol. 9603. LNCS. Springer, Berlin, Heidelberg, Feb. 2017, pp. 149–168. DOI: 10.1007/978-3-662-54970-4_9.

[7] Aydin Abadi et al. “Efficient Delegated Private Set Intersection on Outsourced Private Datasets”. In: *IEEE Transactions on Dependable and Secure Computing* 16.4 (May 2017), pp. 608–624. DOI: 10.1109/TDSC.2017.2708710.

[8] Aydin Abadi et al. “Multi-party Updatable Delegated Private Set Intersection”. In: *FC 2022*. Ed. by Ittay Eyal and Juan A. Garay. Vol. 13411. LNCS. Springer, Cham, May 2022, pp. 100–119. DOI: 10.1007/978-3-031-18283-9_6.

[9] Dirk Achenbach et al. “Your Money or Your Life—Modeling and Analyzing the Security of Electronic Payment in the UC Framework”. In: *FC 2019*. Ed. by Ian Goldberg and Tyler Moore. LNCS 11598. Springer, 2019, pp. 243–261. DOI: 10.1007/978-3-030-32101-7_16.

[10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. “Efficient Lattice (H)IBE in the Standard Model”. In: *Advances in Cryptology – EUROCRYPT 2010*. Ed. by Henri Gilbert. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 553–572. ISBN: 978-3-642-13190-5.

[11] Adi Akavia et al. “Setup-free secure search on encrypted data: Faster and post-processing free”. In: *Cryptology ePrint Archive* (2018).

[12] K. Moriarty et al. *RFC 8017*. 2016. URL: https://datatracker.ietf.org/doc/html/rfc8017 (visited on 01/14/2025).

[13] Martin Albrecht et al. *Homomorphic Encryption Security Standard*. Tech. rep. HomomorphicEncryption.org, Nov. 2018.

[14] Martin R Albrecht, Rachel Player, and Sam Scott. “On the concrete hardness of learning with errors”. In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203.

[15] Andreea Alexandru et al. “Application-aware approximate homomorphic encryption: Configuring fhe for practical use”. In: *Cryptology ePrint Archive* (2024).

[16] Mohammad Ali et al. “Attribute-based fine-grained access control for outscored private set intersection computation”. In: *J: InS* 536 (Oct. 2020), pp. 222–243. DOI: 10.1016/j.ins.2020.05.041.

[17] Asma Aloufi et al. “Blindfolded evaluation of random forests with multi-key homomorphic encryption”. In: *IEEE Transactions on Dependable and Secure Computing* 18.4 (2019), pp. 1821–1835.

[18] Thamer Altuwaiyan, Mohammad Hadian, and Xiaohui Liang. "EPIC: Efficient Privacy-Preserving Contact Tracing for Infection Detection". In: *ICC 2018*. IEEE, 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422886.

[19] Apple and Google. *Privacy-Preserving Contact Tracing*. 2020. URL: http://www.apple.com/covid19/contacttracing.

[20] Benny Applebaum et al. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Heidelberg, Aug. 2009, pp. 595–618. DOI: 10.1007/978-3-642-03356-8_35.

[21] Gennaro Avitabile et al. *Towards Defeating Mass Surveillance and SARS-CoV-2: The Pronto-C2 Fully Decentralized Automatic Contact Tracing System*. Apr. 27, 2020. iacr: 2020/493.

[22] Christian Badertscher et al. "Revisiting (R)CCA Security and Replay Protection". In: *IACR Cryptol. ePrint Arch.* 2020 (2020), p. 177. URL: https://eprint.iacr.org/2020/177.

[23] M Welleda Baldoni, Ciro Ciliberto, and Giulia Maria Piacentini Cattaneo. *Elementary number theory, cryptography and codes*. Springer, 2009.

[24] Jane Bambauer, Krishnamurty Muralidhar, and Rathindra Sarathy. "Fool's Gold: An Illustrated Critique of Differential Privacy". In: *Vanderbilt Journal of Entertainment & Technology Law* 16.4 (Oct. 2014), pp. 701–755.

[25] Aslı Bay et al. "Practical Multi-Party Private Set Intersection Protocols". In: *TIFS 2021* 17 (Oct. 2021), pp. 1–15. DOI: 10.1109/TIFS.2021.3118879.

[26] James Bell et al. "TraceSecure: Towards Privacy Preserving Contact Tracing". In: *ArXiv e-prints* (Apr. 8, 2020). arXiv: 2004.04059 [cs.CR].

[27] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. "Deterministic and Efficiently Searchable Encryption". In: *Advances in Cryptology – CRYPTO 2007*. Ed. by Alfred Menezes. Vol. 4622. Lecture Notes in Computer Science. Springer, 2007, pp. 535–552. DOI: 10.1007/978-3-540-74143-5_30.

[28] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. "Deterministic and efficiently searchable encryption". In: *Advances in Cryptology-CRYPTO 2007: 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007. Proceedings 27*. Springer. 2007, pp. 535–552.

[29] Mihir Bellare, Rafael Dowsley, and Sriram Keelveedhi. "How Secure Is Deterministic Encryption?" In: *Public-Key Cryptography – PKC 2015*. Ed. by Jonathan Katz. Vol. 9020. Lecture Notes in Computer Science. Springer, 2015, pp. 52–73. DOI: 10.1007/978-3-662-46447-2_3.

[30] Mihir Bellare and Phillip Rogaway. "Random oracles are practical: A paradigm for designing efficient protocols". In: *Proceedings of the 1st ACM Conference on Computer and Communications Security*. 1993, pp. 62–73.

[31] Mihir Bellare et al. "Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles". In: *Advances in Cryptology – CRYPTO 2008*. Ed. by David Wagner. Vol. 5157. Lecture Notes in Computer Science. Springer, 2008, pp. 360–378. DOI: 10.1007/978-3-540-85174-5_20.

[32] Laurin Benz et al. "Chosen-Ciphertext Secure Dual-Receiver Encryption in the Standard Model Based on Post-quantum Assumptions". In: *IACR International Conference on Public-Key Cryptography*. Springer. 2024, pp. 257–288.

[33] Laurin Benz et al. "Sender-binding Key Encapsulation". In: *IACR International Conference on Public-Key Cryptography*. Springer. 2023, pp. 744–773.

[34] Alex Berke et al. "Assessing Disease Exposure Risk with Location Data: A Proposal for Cryptographic Preservation of Privacy". In: *ArXiv e-prints* (Mar. 31, 2020). arXiv: 2003.14412 [cs.CR].

[35] David Bernhard, Olivier Pereira, and Bogdan Warinschi. "How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2012, pp. 626–643.

[36] Daniel J Bernstein et al. "How to Manipulate Curve Standards: A White Paper for the Black Hat http://bada55. cr. yp. to". In: *International Conference on Research in Security Standardisation*. Springer. 2015, pp. 109–139.

[37] Daniel J Bernstein et al. “SPHINCS: Practical Stateless hash-Based Signatures”. In: *Advances in Cryptology – EUROCRYPT 2015*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 368–397. ISBN: 978-3-662-46800-5.

[38] Daniel J. Bernstein and Tanja Lange, eds. *eBACS: ECRYPT Benchmarking of Cryptographic Systems*. May 26, 2021. URL: https://bench.cr.yp.to/results-sign.html.

[39] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. “Attacking and Defending the McEliece Cryptosystem”. In: *PQCrypto 2008*. Ed. by Johannes Buchmann and Jintai Ding. Vol. 5299. Lecture Notes in Computer Science. Springer, 2008, pp. 31–46. DOI: 10.1007/978-3-540-88403-3_3. URL: https://doi.org/10.1007/978-3-540-88403-3%5C_3.

[40] Pauline Bert et al. “Implementation of lattice trapdoors on modules and applications”. In: *International Conference on Post-Quantum Cryptography*. Springer. 2021, pp. 195–214.

[41] Wasilij Beskorovajnov and Jörn Müller-Quade. “How to kickstart Secure Message Transfer with Short Authentication Strings and Out-Of-Band Communication”. In: *Cryptology ePrint Archive* (2025).

[42] Wasilij Beskorovajnov et al. “A Formal Treatment of Homomorphic Encryption Based Outsourced Computation in the Universal Composability Framework”. In: *Cryptology ePrint Archive* (2025).

[43] Wasilij Beskorovajnov et al. “A New Security Notion for PKC in the Standard Model: Weaker, Simpler, and Still Realizing Secure Channels”. In: *Cryptology ePrint Archive* (2021).

[44] Wasilij Beskorovajnov et al. “A New Security Notion for PKC in the Standard Model: Weaker, Simpler, and Still Realizing Secure Channels”. In: *Public-Key Cryptography – PKC 2022*. Ed. by Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe. Cham: Springer International Publishing, 2022, pp. 316–344. ISBN: 978-3-030-97131-1.

[45] Wasilij Beskorovajnov et al. “A New Security Notion for PKC in the Standard Model: Weaker, Simpler, and Still Realizing Secure Channels”. In: *PKC 2022, Part II*. Ed. by Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe. Vol. 13178. LNCS. Springer, Cham, Mar. 2022, pp. 316–344. DOI: 10.1007/978-3-030-97131-1_11.

[46] Wasilij Beskorovajnov et al. “A new security notion for PKC in the standard model: weaker, simpler, and still realizing secure channels”. In: *IACR International Conference on Public-Key Cryptography*. Springer. 2022, pp. 316–344.

[47] Wasilij Beskorovajnov et al. “ConTra Corona: Contact Tracing against the Coronavirus by Bridging the Centralized–Decentralized Divide for Stronger Privacy”. In: *Cryptology ePrint Archive* (2020).

[48] Wasilij Beskorovajnov et al. “Contra corona: Contact tracing against the coronavirus by bridging the centralized–decentralized divide for stronger privacy”. In: *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part II 27*. Springer. 2021, pp. 665–695.

[49] Ward Beullens. “Breaking Rainbow Takes a Weekend on a Laptop”. In: *Advances in Cryptology – CRYPTO 2022*. Vol. 13508. Lecture Notes in Computer Science. Cham: Springer, 2022, pp. 464–479. DOI: 10.1007/978-3-031-15979-4_16.

[50] Song Bian, Masayuki Hiromoto, and Takashi Sato. “SCAM: Secured content addressable memory based on homomorphic encryption”. In: *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*. IEEE. 2017, pp. 984–989.

[51] Song Bian et al. “HE3DB: An efficient and elastic encrypted database via arithmetic-and-logic fully homomorphic encryption”. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2023, pp. 2930–2944.

[52] Alexander Bienstock et al. “A more complete analysis of the signal double ratchet algorithm”. In: *Annual International Cryptology Conference*. Springer. 2022, pp. 784–813.

[53] Marina Blanton and Fattaneh Bayatbabolghani. “Efficient Server-Aided Secure Two-Party Function Evaluation with Applications to Genomic Computation”. In: *Proc. Priv. Enhancing Technol.* 2016.4 (2016), pp. 144–164. DOI: 10.1515/POPETS-2016-0033. URL: https://doi.org/10.1515/popets-2016-0033.

[54] Peter Bogetoft et al. “Secure multiparty computation goes live”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2009, pp. 325–343.

[55] Sonia Bogos, Florian Tramèr, and Serge Vaudenay. "On Solving Lpn using BKW and Variants". In: *IACR Cryptol. ePrint Arch.* (2015), p. 49. URL: http://eprint.iacr.org/2015/049.

[56] Dan Boneh and Jonathan Katz. "Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption". In: 2005, pp. 87–103. DOI: 10.1007/978-3-540-30574-3_8.

[57] Dan Boneh, Amit Sahai, and Brent Waters. "Functional encryption: Definitions and challenges". In: *Theory of Cryptography: 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings 8*. Springer. 2011, pp. 253–273.

[58] Dan Boneh et al. "Chosen-Ciphertext Security from Identity-Based Encryption". In: 36.5 (2007), pp. 1301–1328.

[59] Dan Boneh et al. "Random oracles in a quantum world". In: *Advances in Cryptology–ASIACRYPT 2011: 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings 17*. Springer. 2011, pp. 41–69.

[60] Charlotte Bonte and Ilia Iliashenko. "Homomorphic string search with constant multiplicative depth". In: *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*. 2020, pp. 105–117.

[61] Jean-Philippe Bossuat et al. "Security guidelines for implementing homomorphic encryption". In: *Cryptology ePrint Archive* (2024).

[62] Xavier Boyen, Malika Izabachène, and Qinyi Li. "A Simple and Efficient CCA-Secure Lattice KEM in the Standard Model". In: 2020, pp. 321–337. DOI: 10.1007/978-3-030-57990-6_16.

[63] Xavier Boyen, Malika Izabachène, and Qinyi Li. "Secure Hybrid Encryption in the Standard Model from Hard Learning Problems". In: *Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021*. Ed. by Jung Hee Cheon and Jean-Pierre Tillich. Springer, Heidelberg, July 2021, pp. 399–418. DOI: 10.1007/978-3-030-81293-5_21.

[64] Xavier Boyen, Malika Izabachène, and Qinyi Li. "Secure Hybrid Encryption in the Standard Model from Hard Learning Problems". In: *Post-Quantum Cryptography*. Ed. by Jung Hee Cheon and Jean-Pierre Tillich. Cham: Springer International Publishing, 2021, pp. 399–418. ISBN: 978-3-030-81293-5.

[65] Samuel Brack, Leonie Reichert, and Björn Scheuermann. *CAUDHT: Decentralized Contact Tracing Using a DHT and Blind Signatures*. Ed. by Hwee-Pink Tan, Lyes Khoukhi, and Sharief Oteafy. 2020. DOI: 10.1109/LCN48667.2020.9314850.

[66] Zvika Brakerski. "Fully homomorphic encryption without modulus switching from classical GapSVP". In: *Annual cryptology conference*. Springer. 2012, pp. 868–886.

[67] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping". In: *ACM Transactions on Computation Theory (TOCT)* 6.3 (2014), pp. 1–36.

[68] Robert Brede. "Improving Secure Channels without Random Oracles based on Ring-LWE". Master's Thesis. Karlsruhe Institute of Technology, 2024.

[69] Emmanuel Bresson, Dario Catalano, and David Pointcheval. "A Simple Public-Key Cryptosystem with a Double Trapdoor Decryption Mechanism and Its Applications". In: *ASIACRYPT 2003*. Ed. by Chi-Sung Laih. Vol. 2894. LNCS. Springer, Berlin, Heidelberg, Nov. 2003, pp. 37–54. DOI: 10.1007/978-3-540-40061-5_3.

[70] BSI. *Online Wahlen*. 2020. URL: https://www.bsi.bund.de/DE/Themen/Oeffentliche-Verwaltung/Moderner-Staat/Online-Wahlen/online-wahlen_node.html.

[71] BSI. *Positionspapier Zero-Trust*. 2020. URL: https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Zero-Trust/zero-trust_node.html.

[72] BSI. *Projekt 374 Final Report*. 2020. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Studie-374.html.

[73] BSI. *The State of IT Security in Germany*. 2024. URL: https://www.bsi.bund.de/EN/Service-Navi/Publikationen/Lagebericht/lagebericht_node.html.

[74] Jon Callas et al. "OpenPGP Message Format". In: *RFC* 4880 (2007), pp. 1–90. DOI: 10.17487/RFC4880. URL: https://doi.org/10.17487/RFC4880.

[75] Jan Camenisch et al. “How to win the clonewars: efficient periodic n-times anonymous authentication”. In: *CCS 2006*. Ed. by Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati. ACM, 2006, pp. 201–210. DOI: 10.1145/1180405.1180431.

[76] Jan Camenisch et al. “The wonderful world of global random oracles”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2018, pp. 280–312.

[77] Ran Canetti. “Universally Composable Security”. In: *J. ACM* 67.5 (Sept. 2020), p. 94. DOI: 10.1145/3402457.

[78] Ran Canetti. “Universally Composable Security: A New Paradigm for Cryptographic Protocols”. In: 2001, pp. 136–145. DOI: 10.1109/SFCS.2001.959888.

[79] Ran Canetti. “Universally composable security: a new paradigm for cryptographic protocols”. In: *42nd FOCS*. Oct. 2001, pp. 136–145. DOI: 10.1109/SFCS.2001.959888.

[80] Ran Canetti. “Universally composable signature, certification, and authentication”. In: *Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004.* IEEE. 2004, pp. 219–233.

[81] Ran Canetti. “Universally composable signature, certification, and authentication”. In: *Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004.* IEEE. 2004, pp. 219–233.

[82] Ran Canetti, Oded Goldreich, and Shai Halevi. “The random oracle methodology, revisited”. In: *Journal of the ACM (JACM)* 51.4 (2004), pp. 557–594.

[83] Ran Canetti, Shai Halevi, and Jonathan Katz. “A forward-secure public-key encryption scheme”. In: *Advances in Cryptology—EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22*. Springer. 2003, pp. 255–271.

[84] Ran Canetti, Abhishek Jain, and Alessandra Scafuro. “Practical UC security with a global random oracle”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 597–608.

[85] Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. “Relaxing Chosen-Ciphertext Security”. In: 2003, pp. 565–582. DOI: 10.1007/978-3-540-45146-4_33.

[86] Ran Canetti and Tal Rabin. "Universal Composition with Joint State". In: 2003, pp. 265–281. DOI: 10.1007/978-3-540-45146-4_16.

[87] Ran Canetti, Ari Trachtenberg, and Mayank Varia. "Anonymous Collocation Discovery: Harnessing Privacy to Tame the Coronavirus". In: *ArXiv e-prints* (Mar. 30, 2020). arXiv: 2003.13670 [cs.CY].

[88] Ran Canetti et al. *Privacy-Preserving Automated Exposure Notification*. July 9, 2020. iacr: 2020/863.

[89] Ganyuan Cao, Sylvain Chatel, and Christian Knabenhans. "HE-based On-the-Fly MPC, Revisited: Universal Composability, Approximate and Imperfect Computation, Circuit Privacy". In: *Cryptology ePrint Archive* (2025).

[90] Henry Carter, Charles Lever, and Patrick Traynor. "Whitewash: Outsourcing garbled circuit generation for mobile devices". In: *Proceedings of the 30th Annual Computer Security Applications Conference*. 2014, pp. 266–275.

[91] Henry Carter and Patrick Traynor. "Outsourcing computation for private function evaluation". In: *International Journal of Information and Computer Security* 11.6 (2019), pp. 525–561.

[92] Henry Carter et al. "Outsourcing Secure Two-Party Computation as a Black Box". In: *Cryptology and Network Security - 14th International Conference, CANS 2015, Marrakesh, Morocco, December 10-12, 2015, Proceedings*. Ed. by Michael K. Reiter and David Naccache. Vol. 9476. Lecture Notes in Computer Science. Springer, 2015, pp. 214–222. DOI: 10.1007/978-3-319-26823-1_15. URL: https://doi.org/10.1007/978-3-319-26823-1%5C_15.

[93] Henry Carter et al. "Secure Outsourced Garbled Circuit Evaluation for Mobile Devices". In: *22nd USENIX Security Symposium (USENIX Security 13)*. 2013, pp. 289–304.

[94] J. W. S. Cassels and A. Fröhlich, eds. *Algebraic Number Theory*. London: Academic Press, 1967.

[95] Claude Castelluccia et al. *DESIRE: A Third Way for a European Exposure Notification System*. May 9, 2020. URL: https://github.com/3rd-ways-for-EU-exposure-notification/project-DESIRE.

[96] Wouter Castryck and Thomas Decru. "An efficient key recovery attack on SIDH". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 423–447.

[97] Justin Chan et al. “PACT: Privacy Sensitive Protocols and Mechanisms for Mobile Contact Tracing”. In: *ArXiv e-prints* (Apr. 7, 2020). arXiv: 2004.03544 [cs.CR].

[98] Hao Chen, Kim Laine, and Peter Rindal. “Fast Private Set Intersection from Homomorphic Encryption”. In: *CCS 2017*. CCS ’17’. Association for Computing Machinery, Oct. 2017, pp. 1243–1255. DOI: 10.1145/3133956.3134061.

[99] Lichao Chen et al. “Two Anti-quantum Attack Protocols for Secure Multiparty Computation”. In: *CTCIS 2018*. Ed. by Huanguo Zhang, Bo Zhao, and Fei Yan. Vol. 960. CCIS. Springer, Singapore, Jan. 2019, pp. 338–359. DOI: 10.1007/978-981-13-5913-2_21.

[100] Haitao Cheng et al. “Simpler CCA Secure PKE from LPN Problem Without Double-Trapdoor”. In: 2018, pp. 756–766. DOI: 10.1007/978-3-030-01950-1_46.

[101] Jung Hee Cheon, Miran Kim, and Myungsun Kim. “Optimized search-and-compute circuits and their application to query evaluation on encrypted data”. In: *IEEE Transactions on Information Forensics and Security* 11.1 (2015), pp. 188–199.

[102] Jung Hee Cheon, Miran Kim, and Myungsun Kim. “Search-and-compute on encrypted data”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2015, pp. 142–159.

[103] Jung Hee Cheon et al. “Attacks against the IND-CPAD security of exact FHE schemes”. In: *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 2024, pp. 2505–2519.

[104] Jung Hee Cheon et al. “Homomorphic Encryption for Arithmetic of Approximate Numbers”. In: *ASIACRYPT 2017, Part I*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10624. LNCS. Springer, Cham, Dec. 2017, pp. 409–437. DOI: 10.1007/978-3-319-70694-8_15.

[105] Jung Hee Cheon et al. “Reusable dynamic multi-party homomorphic encryption”. In: *Cryptology ePrint Archive* (2025).

[106] Ilaria Chillotti et al. “Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds”. In: *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*. Springer. 2016, pp. 3–33.

[107] Hyunghoon Cho, Daphne Ippolito, and Yun William Yu. "Contact Tracing Mobile Apps for COVID-19: Privacy Considerations and Related Trade-offs". In: *ArXiv e-prints* (Mar. 25, 2020). arXiv: 2003.11511 [cs.CR].

[108] Benny Chor et al. "Private information retrieval". In: *Journal of the ACM (JACM)* 45.6 (Nov. 1998), pp. 965–981. DOI: 10.1145/293347.293350.

[109] Sherman S. M. Chow, Matthew K. Franklin, and Haibin Zhang. "Practical Dual-Receiver Encryption - Soundness, Complete Non-malleability, and Applications". In: 2014, pp. 85–105. DOI: 10.1007/978-3-319-04852-9_5.

[110] CISA. *Insider Threat Mitigation Guide*. 2020. URL: https://www.cisa.gov/resources-tools/resources/insider-threat-mitigation-guide.

[111] Julie E. Cohen. *Configuring the Networked Self: Law, Code, and the Play of Everyday Practice*. New Haven, CT: Yale University Press, 2012.

[112] David A Cooper et al. "Recommendation for Stateful Hash-Based Signature Schemes". In: *NIST Special Publication* 800.208 (2020), pp. 800–208. DOI: 10.6028/NIST.SP.800-208.

[113] David A. Cooper et al. *Recommendation for Stateful Hash-Based Signature Schemes*. NIST Special Publication 800-208. Gaithersburg, MD: National Institute of Standards and Technology, Oct. 2020. DOI: 10.6028/NIST.SP.800-208. URL: https://doi.org/10.6028/NIST.SP.800-208.

[114] Alain Couvreur et al. "Distinguisher-based attacks on public-key cryptosystems using Reed–Solomon codes". In: *Designs, Codes and Cryptography* 73 (2014), pp. 641–666.

[115] Ronald Cramer and Victor Shoup. "Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack". In: *SIAM Journal on Computing* 33 (Jan. 2002). DOI: 10.1137/S0097539702403773.

[116] Ronald Cramer and Victor Shoup. "Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack". In: 33.1 (2003), pp. 167–226.

[117] Cas Cremers, Niklas Medinger, and Aurora Naska. “Impossibility Results for Post-Compromise Security in Real-World Communication Systems”. In: *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2025, pp. 4391–4405.

[118] Eric Crockett and Chris Peikert. “Challenges for ring-LWE”. In: *Cryptology ePrint Archive* (2016).

[119] Ivan Damgard, Kasper Dupont, and Michael Ostergaard Pedersen. “Unclonable Group Identification”. In: *EUROCRYPT 2006*. Ed. by Serge Vaudenay. Vol. 4004. LNCS. Springer, 2006, pp. 555–572. DOI: 10.1007/11761679_33.

[120] Ivan Damgård and Sunoo Park. “Is Public-Key Encryption Based on LPN Practical?” In: *IACR Cryptol. ePrint Arch.* (2012), p. 699. URL: http://eprint.iacr.org/2012/699.

[121] Ivan Damgård et al. “Confidential benchmarking based on multiparty computation”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2016, pp. 169–187.

[122] Ivan Damgård et al. “Multiparty Computation from Somewhat Homomorphic Encryption”. In: *CRYPTO 2012*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. LNCS. Springer, Berlin, Heidelberg, Aug. 2012, pp. 643–662. DOI: 10.1007/978-3-642-32009-5_38.

[123] Ivan Damgård et al. “Practical covertly secure MPC for dishonest majority–or: breaking the SPDZ limits”. In: *European Symposium on Research in Computer Security*. Springer. 2013, pp. 1–18.

[124] Noel Danz et al. *Provable Security and Privacy of Decentralized Cryptographic Contact Tracing*. Oct. 20, 2020. iacr: 2020/1309.

[125] Sumit Kumar Debnath et al. “Secure Outsourced Private Set Intersection with Linear Complexity”. In: *DSC 2021*. IEEE, Feb. 2021, pp. 1–8. DOI: 10.1109/DSC49826.2021.9346230.

[126] Deloitte. *Cybersecurity Threat Trends Report 2024*. 2024. URL: https://www2.deloitte.com/us/en/pages/risk/articles/cybersecurity-threat-trends-report-2024.html.

[127] Theodore Diament et al. “The Dual Receiver Cryptosystem and Its Applications”. In: 2004, pp. 330–343. DOI: 10.1145/1030083.1030128.

[128] Whitfield Diffie and Martin E. Hellman. “New Directions in Cryptography”. In: 22.6 (1976), pp. 644–654.

[129] Jintai Ding and Dieter Schmidt. "Rainbow, a New Multivariable Polynomial Signature Scheme". In: *Applied Cryptography and Network Security (ACNS 2005)*. Vol. 3531. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 164–175. DOI: 10.1007/11496137_12.

[130] Yevgeniy Dodis and Adam Smith. "Entropic Security and the Encryption of High Entropy Messages". In: *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005*. Ed. by Joe Kilian. Vol. 3378. Lecture Notes in Computer Science. Springer, 2005, pp. 556–577. DOI: 10.1007/978-3-540-30576-7_30.

[131] Nico Döttling et al. *A CCA2 Secure Variant of the McEliece Cryptosystem*. Cryptology ePrint Archive, Report 2008/468. https://eprint.iacr.org/2008/468. 2008.

[132] Rafael Dowsley, Jörn Müller-Quade, and Anderson CA Nascimento. "A CCA2 secure public key encryption scheme based on the McEliece assumptions in the standard model". In: *Cryptographers' Track at the RSA Conference*. Springer. 2009, pp. 240–251.

[133] DP3T Project. *FAQ: Decentralized Proximity Tracing*. 2020. URL: https://github.com/DP-3T/documents/blob/master/FAQ.md.

[134] Thai Duong, Duong Hieu Phan, and Ni Trieu. "Catalic: Delegated PSI Cardinality with Applications to Contact Tracing". In: *ASIACRYPT 2020*. LNCS 12493. Springer, 2020, pp. 870–899. DOI: 10.1007/978-3-030-64840-4_29.

[135] Cynthia Dwork. "Differential privacy". In: *Automata, Languages, and Programming*. Ed. by Michele Bugliesi et al. Vol. 4052. LNCS. Springer, Berlin, Heidelberg, July 2006, pp. 1–12. DOI: 10.1007/11787006_1.

[136] Edward Eaton and Fang Song. "A note on the instantiability of the quantum random oracle". In: *Post-Quantum Cryptography: 11th International Conference, PQCrypto 2020, Paris, France, April 15–17, 2020, Proceedings 11*. Springer. 2020, pp. 503–523.

[137] EDPB. *Guidelines 01/2025 on Pseudonymization*. 2025. URL: https://www.edpb.europa.eu/our-work-tools/documents/public-consultations/2025/guidelines-012025-pseudonymisation_en.

[138] ENISA. *Cyber Threat Landscape 2024*. 2024. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024.

[139] ENISA. *Cyber Threats 2024*. 2024. URL: https://www.enisa.europa.eu/topics/cyber-threats/threats-and-trends.

[140] ENISA. *Cyber Threats 2025*. 2025. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2025 (visited on 05/10/2025).

[141] ENISA. *Data Pseudonymisation: Advanced Techniques and Use Cases*. 2021. URL: https://www.enisa.europa.eu/publications/data-pseudonymisation-advanced-techniques-and-use-cases.

[142] ENISA. *Deploying Pseudonymisation Techniques*. 2022. URL: https://www.enisa.europa.eu/publications/deploying-pseudonymisation-techniques.

[143] ENISA. *Pseudonymisation Techniques and Best Practices*. 2019. URL: https://www.enisa.europa.eu/publications/pseudonymisation-techniques-and-best-practices.

[144] EP. *Art. 5 GDPR Principles relating to processing of personal data*. 2016. URL: https://gdpr-info.eu/art-5-gdpr/.

[145] EP. *Recommendations on shaping technology according to GDPR provisions*. 2018. URL: https://op.europa.eu/en/publication-detail/-/publication/0e1ca64f-29c7-11e9-8d04-01aa75ed71a1/language-en.

[146] ESA. *Final report on draft RTS on ICT Risk Management Framework and on simplified ICT Risk Management Framework*. 2024. URL: https://www.eba.europa.eu/sites/default/files/2024-01/bf5a2976-1a48-44f3-b5a7-56acd23ba55c/JC%202023%2086%20-%20Final%20report%20on%20draft%20RTS%20on%20ICT%20Risk%20Management%20Framework%20and%20on%20simplified%20ICT%20Risk%20Management%20Framework.pdf.

[147] Junfeng Fan and Frederik Vercauteren. "Somewhat practical fully homomorphic encryption". In: *Cryptology ePrint Archive* (2012).

[148] Dennis M Feehan and Ayesha S Mahmud. "Quantifying population contact patterns in the United States during the COVID-19 pandemic". In: *Nature communications* 12.1 (2021), pp. 1–9. DOI: 10.1038/s41467-021-20990-2.

[149] Jack K Fitzsimons et al. "A note on blind contact tracing at scale with applications to the COVID-19 pandemic". In: *ARES 2020*. Ed. by Melanie Volkamer and Christian Wressnegger. ACM, 2020, 92:1–92:6. DOI: 10.1145/3407023.3409204.

[150] Fraunhofer AISEC. *Pandemic Contact Tracing Apps: DP-3T, PEPP-PT NTK, and ROBERT from a Privacy Perspective*. Apr. 27, 2020. iacr: 2020/489.

[151] Tore Kasper Frederiksen et al. "Attribute-based single sign-on: Secure, private, and efficient". In: *Cryptology ePrint Archive* (2023).

[152] David Mandell Freeman et al. "More Constructions of Lossy and Correlation-Secure Trapdoor Functions". In: 2010, pp. 279–295. DOI: 10.1007/978-3-642-13013-7_17.

[153] Eiichiro Fujisaki and Tatsuaki Okamoto. "Secure Integration of Asymmetric and Symmetric Encryption Schemes". In: 1999, pp. 537–554. DOI: 10.1007/3-540-48405-1_34.

[154] Benjamin Fuller, Adam O'Neill, and Leonid Reyzin. "A Unified Approach to Deterministic Encryption: New Constructions and a Connection to Computational Entropy". In: *Journal of Cryptology* 28.3 (2015), pp. 671–717. DOI: 10.1007/s00145-013-9174-5.

[155] Giuseppe Garofalo et al. *PIVOT: PrIVate and effective cOntact Tracing*. Jan. 22, 2021. iacr: 2020/559.

[156] Tingjian Ge and Stan Zdonik. "Answering aggregation queries in a secure system model". In: *Proceedings of the 33rd international conference on Very large data bases*. 2007, pp. 519–530.

[157] Konstantin Gegier. "On Novel Constructions of Dual Receiver Key Encapsulation Mechanisms Based on Deterministic Encryption". MA thesis. Karlsruhe Institute of Technology (KIT), 2020.

[158] Rosario Gennaro, Shai Halevi, and Tal Rabin. "Secure hash-and-sign signatures without the random oracle". In: *Advances in Cryptology—EUROCRYPT'99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings 18*. Springer. 1999, pp. 123–139.

[159] Craig Gentry. "Fully homomorphic encryption using ideal lattices". In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 2009, pp. 169–178.

[160] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: 2008, pp. 197–206. DOI: 10.1145/1374376.1374407.

[161] Shafi Goldwasser, Yael Tauman Kalai, and Guy N Rothblum. "Delegating computation: interactive proofs for muggles". In: *Journal of the ACM (JACM)* 62.4 (2015), pp. 1–64.

[162] Shafi Goldwasser and Silvio Micali. "Probabilistic Encryption". In: 28.2 (1984), pp. 270–299.

[163] Shafi Goldwasser and Silvio Micali. "Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information". In: *14th ACM STOC*. ACM Press, May 1982, pp. 365–377. DOI: 10.1145/800070.802212.

[164] Shafi Goldwasser and Yael Tauman Kalai. "Cryptographic assumptions: A position paper". In: *Theory of Cryptography Conference*. Springer. 2015, pp. 505–522.

[165] S Dov Gordon et al. "Multi-party computation of polynomials and branching programs without simultaneous interaction". In: *Advances in Cryptology–EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings 32*. Springer. 2013, pp. 575–591.

[166] Shai Halevi, Yehuda Lindell, and Benny Pinkas. "Secure computation on the web: Computing without simultaneous interaction". In: *Advances in Cryptology–CRYPTO 2011: 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings 31*. Springer. 2011, pp. 132–150.

[167] Xi He, Ashwin Machanavajjhala, and Bolin Ding. "Blowfish privacy: Tuning privacy-utility trade-offs using policies". In: *SIGMOD 2014*. SIGMOD '14. Association for Computing Machinery, June 2014, pp. 1447–1458. DOI: 10.1145/2588555.2588581.

[168] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. "A modular analysis of the Fujisaki-Okamoto transformation". In: *Theory of Cryptography Conference*. Springer. 2017, pp. 341–371.

[169] Andreas Huelsing et al. *XMSS: eXtended Merkle Signature Scheme*. RFC 8391. May 2018. DOI: 10.17487/RFC8391. URL: https://www.rfc-editor.org/info/rfc8391.

[170] MATZOV IDF. *Report on the Security of LWE: Improved Dual Lattice Attack*. 2022. URL: https://zenodo.org/records/6412487.

[171] Ilia Iliashenko and Vincent Zucca. "Faster homomorphic comparison operations for BGV and BFV". In: *Proceedings on Privacy Enhancing Technologies* 2021.3 (2021), pp. 246–264.

[172] Ilia Iliashenko et al. "Homomorphically counting elements with the same property". In: *Proceedings on Privacy Enhancing Technologies* (2022).

[173] ISO. *ISO 25237:2017 Health informatics — Pseudonymization*. 2017. URL: https://www.iso.org/standard/63553.html.

[174] ISO/IEC. *ISO/IEC 20889:2018 Privacy enhancing data de-identification terminology and classification of techniques*. 2018. URL: https://www.iso.org/standard/69373.html.

[175] Arvind K. Jain. "Corruption: A Review". In: *Journal of Economic Surveys* 15.1 (Feb. 2001), pp. 71–121. DOI: 10.1111/1467-6419.00133.

[176] Thomas P. Jakobsen, Jesper Buus Nielsen, and Claudio Orlandi. "A Framework for Outsourcing of Secure Computation". In: *CCSW 2014*. CCSW '14. Association for Computing Machinery, Nov. 2014, pp. 81–92. DOI: 10.1145/2664168.2664170.

[177] Amirhossein Adavoudi Jolfaei, Hamid Mala, and Maryam Zarezadeh. "EO-PSI-CA: Efficient outsourced private set intersection cardinality". In: *J. JISA* 65.102996 (Mar. 2022), p. 11. DOI: https://doi.org/10.1016/j.jisa.2021.102996.

[178] Seny Kamara, Payman Mohassel, and Mariana Raykova. *Outsourcing Multi-Party Computation*. Cryptology ePrint Archive, Report 2011/272. 2011. URL: https://eprint.iacr.org/2011/272.

[179] Seny Kamara, Payman Mohassel, and Ben Riva. "Salus: a system for server-aided secure function evaluation". In: *Proceedings of the 2012 ACM conference on Computer and communications security*. 2012, pp. 797–808.

[180] Seny Kamara et al. "Scaling Private Set Intersection to Billion-Element Sets". In: *FC 2014*. Ed. by Nicolas Christin and Reihaneh Safavi-Naini. Vol. 8437. LNCS. Springer, Berlin, Heidelberg, Mar. 2014, pp. 195–215. DOI: 10.1007/978-3-662-45472-5_13.

[181] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2007.

[182] Dmitry Khovratovich, Ron D Rothblum, and Lev Soukhanov. "How to prove false statements: Practical attacks on fiat-shamir". In: *Annual International Cryptology Conference*. Springer. 2025, pp. 3–26.

[183] Daniel Kifer and Ashwin Machanavajjhala. "Pufferfish: A framework for mathematical privacy definitions". In: *TODS* 39.1 (Jan. 2014), pp. 1–36. DOI: 10.1145/2514689.

[184] Eike Kiltz. "Chosen-Ciphertext Security from Tag-Based Encryption". In: 2006, pp. 581–600. DOI: 10.1007/11681878_30.

[185] Eike Kiltz, Daniel Masny, and Krzysztof Pietrzak. "Simple Chosen-Ciphertext Security from Low-Noise LPN". In: 2014, pp. 1–18. DOI: 10.1007/978-3-642-54631-0_1.

[186] Eike Kiltz, Payman Mohassel, and Adam O'Neill. "Adaptive Trapdoor Functions and Chosen-Ciphertext Security". In: 2010, pp. 673–692. DOI: 10.1007/978-3-642-13190-5_34.

[187] Miran Kim and Kristin Lauter. "Private genome analysis through homomorphic encryption". In: *BMC medical informatics and decision making* 15.Suppl 5 (2015), S3.

[188] Myungsun Kim et al. "On the efficiency of FHE-based private queries". In: *IEEE Transactions on Dependable and Secure Computing* 15.2 (2016), pp. 357–363.

[189] Myungsun Kim et al. "Private compound wildcard queries using fully homomorphic encryption". In: *IEEE Transactions on Dependable and Secure Computing* 16.5 (2017), pp. 743–756.

[190] Jan-Martin Knorr. "Abstreitbare Nachrichtenauthentifikation mit Post-Quanten-Kryptographie (in German)". Master's Thesis. Karlsruhe Institute of Technology, 2016.

[191] Brian Knott et al. "Crypten: Secure multi-party computation meets machine learning". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 4961–4973.

[192] Neal Koblitz and Alfred J Menezes. "The random oracle model: a twenty-year retrospective". In: *Designs, Codes and Cryptography* 77 (2015), pp. 587–610.

[193] Christiane Kuhn, Martin Beck, and Thorsten Strufe. "Covid Notions: Towards Formal Definitions – and Documented Understanding – of Privacy Goals and Claimed Protection in Proximity-Tracing Services". In: *Online Soc. Networks Media* 22 (2021). DOI: 10.1016/j.osnem.2021.100125.

[194] Hyesun Kwak et al. "A Unified Framework of Homomorphic Encryption for Multiple Parties with Non-Interactive Setup." In: *IACR Cryptol. ePrint Arch.* 2021 (2021), p. 1412.

[195] Russell WF Lai, Henry KF Cheung, and Sherman SM Chow. "Trapdoors for ideal lattices with applications". In: *International Conference on Information Security and Cryptology*. Springer. 2014, pp. 239–256.

[196] Gaëtan Leurent and Phong Q Nguyen. "How risky is the random-oracle model?" In: *Annual International Cryptology Conference*. Springer. 2009, pp. 445–464.

[197] Baiyu Li and Daniele Micciancio. "On the security of homomorphic encryption on approximate numbers". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2021, pp. 648–677.

[198] Yanyan Liu et al. "(Identity-based) dual receiver encryption from lattice-based programmable hash functions with high min-entropy". In: *Cybersecur.* 2.1 (2019), p. 18. DOI: 10.1186/s42400-019-0034-y. URL: https://doi.org/10.1186/s42400-019-0034-y.

[199] Yuan Liu et al. "New Constructions of Identity-Based Dual Receiver Encryption from Lattices". In: *Entropy* 22.6 (2020), p. 599. DOI: 10.3390/e22060599. URL: https://doi.org/10.3390/e22060599.

[200] Adriana Lopez-Alt, Eran Tromer, and Vinod Vaikuntanathan. *Cloud-Assisted Multiparty Computation from Fully Homomorphic Encryption*. Cryptology ePrint Archive, Report 2011/663. 2011. URL: https://eprint.iacr.org/2011/663.

[201] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. "Multikey Fully Homomorphic Encryption and Applications". In: *SIAM Journal on Computing* 46.6 (2017), pp. 1827–1892. DOI: 10.1137/14100124X.

[202] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption". In: *STOC 2012*. STOC '12. Association for Computing Machinery, May 2012, pp. 1219–1234. DOI: 10.1145/2213977.2214086.

[203] Vadim Lyubashevsky. "Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2009, pp. 598–616.

[204] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "A toolkit for ring-LWE cryptography". In: *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2013, pp. 35–54.

[205] Philip D. MacKenzie, Michael K. Reiter, and Ke Yang. "Alternatives to Non-malleability: Definitions, Constructions, and Applications (Extended Abstract)". In: 2004, pp. 171–190. DOI: 10.1007/978-3-540-24638-1_10.

[206] Daniel A. Marcus. *Number Fields*. Universitext. New York: Springer, 1977. DOI: 10.1007/978-1-4684-9356-6.

[207] Ueli Maurer, Renato Renner, and Clemens Holenstein. "Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology". In: *Theory of Cryptography: First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004. Proceedings 1*. Springer. 2004, pp. 21–39.

[208] Robert J McEliece. "A public-key cryptosystem based on algebraic". In: *Coding Thv* 4244 (1978), pp. 114–116.

[209] David McGrew, Michael Curcio, and Scott Fluhrer. *Leighton-Micali Hash-Based Signatures*. RFC 8554. Apr. 2019. DOI: 10.17487/RFC8554. URL: https://www.rfc-editor.org/info/rfc8554.

[210] Frank McSherry and Kunal Talwar. "Mechanism Design Via Differential Privacy". In: *48th FOCS*. IEEE, Oct. 2007, pp. 94–103. DOI: 10.1109/FOCS.2007.66.

[211] Daniele Micciancio and Chris Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 700–718. ISBN: 978-3-642-29011-4.

[212] Daniele Micciancio and Chris Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: 2012, pp. 700–718. DOI: 10.1007/978-3-642-29011-4_41.

[213] Daniele Micciancio and Oded Regev. "Lattice-based Cryptography". In: *Post-Quantum Cryptography*. Ed. by Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Vol. 5299. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2009, pp. 147–191. DOI: 10.1007/978-3-540-88702-7_5.

[214] Arno Mittelbach et al. "The Random Oracle Controversy". In: *The Theory of Hash Functions and Random Oracles: An Approach to Modern Cryptography* (2021), pp. 461–475.

[215] Atsuko Miyaji, Kazuhisa Nakasho, and Shohei Nishida. "Privacy-Preserving Integration of Medical Data". In: *Journal of Medical Systems* 41.37 (Jan. 2017), p. 10. DOI: 10.1007/s10916-016-0657-4.

[216] Payman Mohassel, Ostap Orobets, and Ben Riva. "Efficient server-aided 2pc for mobile phones". In: *Proceedings on Privacy Enhancing Technologies* (2016).

[217] Benjamin Mood et al. "Reuse It Or Lose It: More Efficient Secure Computation Through Reuse of Encrypted Values". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*. Ed. by Gail-Joon Ahn, Moti Yung, and Ninghui Li. ACM, 2014, pp. 582–596. DOI: 10.1145/2660267.2660285. URL: https://doi.org/10.1145/2660267.2660285.

[218] Koki Morimura, Daisuke Maeda, and Takashi Nishide. "Accelerating polynomial evaluation for integer-wise homomorphic comparison and division". In: *Journal of Information Processing* 31 (2023), pp. 288–298.

[219] Christian Mouchet, Elliott Bertrand, and Jean-Pierre Hubaux. "An Efficient Threshold Access-Structure for RLWE-Based Multiparty Homomorphic Encryption". In: *Journal of Cryptology* 36.2 (Mar. 2023), p. 10. DOI: 10.1007/s00145-023-09452-8.

[220] Christian Mouchet et al. "Multiparty Homomorphic Encryption from Ring-Learning-with-Errors". In: *PoPETs* 2021.4 (June 2021), pp. 291–311. DOI: 10.2478/popets-2021-0071.

[221] Pratyay Mukherjee and Daniel Wichs. "Two round multiparty computation via multi-key FHE". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2016, pp. 735–763.

[222] Johannes Müller and Volker Hösel. “Contact tracing & super-spreaders in the branching-process model”. In: *Journal of Mathematical Biology* 86.2 (2023), p. 24.

[223] FHE Multikey. “Towards Round-Optimal Secure Multiparty Computations: Multikey FHE Without a CRS”. In: *Information Security and Privacy*. Springer, p. 101.

[224] Helen Nissenbaum. *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford, CA: Stanford Law Books, 2010.

[225] NIST. *Module-Lattice-Based Digital Signature Standard*. 2024. URL: https://csrc.nist.gov/pubs/fips/204/final.

[226] NIST. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. 2024. URL: https://csrc.nist.gov/pubs/fips/203/final.

[227] NIST. *NIST Privacy-Enhancing Cryptography Project*. 2024. URL: https://csrc.nist.gov/projects/pec.

[228] NIST. *Stateless Hash-Based Digital Signature Standard*. 2024. URL: https://csrc.nist.gov/pubs/fips/205/final.

[229] Geontae Noh et al. “A Strong Binding Encryption Scheme from Lattices for Secret Broadcast”. In: *IEEE Commun. Lett.* 16.6 (2012), pp. 781–784. DOI: 10.1109/LCOMM.2012.041112.112495. URL: https://doi.org/10.1109/LCOMM.2012.041112.112495.

[230] Ryo Nojima et al. “Semantic security for the McEliece cryptosystem without random oracles”. In: *Des. Codes Cryptogr.* 49.1-3 (2008), pp. 289–305. DOI: 10.1007/s10623-008-9175-9. URL: https://doi.org/10.1007/s10623-008-9175-9.

[231] Ivan Oleynikov, Elena Pagnin, and Andrei Sabelfeld. “Outsourcing mpc precomputation for location privacy”. In: *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE. 2022, pp. 504–513.

[232] Mahdi Mahdavi Oliaiy et al. “A Verifiable Delegated Set Intersection Without Pairing”. In: *IranianCEE 2017*. IEEE, May 2017, pp. 2047–2051. DOI: 10.1109/IranianCEE.2017.7985395.

[233] Felix Peter Paul. “Goppa Codes”. In: *Codebasierte Post-Quanten-Kryptografie : Goppa Codes und das McEliece Kryptosystem*. Wiesbaden: Springer Fachmedien Wiesbaden, 2025, pp. 17–37. ISBN: 978-3-658-46743-2. DOI: 10.1007/978-3-658-46743-2_3. URL: https://doi.org/10.1007/978-3-658-46743-2_3.

[234] Chris Peikert. "An Efficient and Parallel Gaussian Sampler for Lattices". In: *Advances in Cryptology – CRYPTO 2010*. Ed. by Tal Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 80–97. ISBN: 978-3-642-14623-7.

[235] Chris Peikert. "How (not) to instantiate ring-LWE". In: *International Conference on Security and Cryptography for Networks*. Springer. 2016, pp. 411–430.

[236] Chris Peikert and Brent Waters. "Lossy trapdoor functions and their applications". In: 2008, pp. 187–196. DOI: 10.1145/1374376.1374406.

[237] Jonathon W. Penney. "Chilling Effects: Online Surveillance and Wikipedia Use". In: *Berkeley Technology Law Journal* 31.1 (2016), pp. 117–182.

[238] PePP-PT e.V. *Pan-European Privacy-Preserving Proximity Tracing*. 2020. URL: https://www.pepp-pt.org/content.

[239] PePP-PT e.V. *PEPP-PT NTK High-Level Overview*. 2020. URL: https://github.com/pepp-pt/pepp-pt-documentation/blob/master/PEPP-PT-high-level-overview.pdf.

[240] PePP-PT e.V. *ROBust and privacy-presERving proximity Tracing protocol*. 2020. URL: https://github.com/ROBERT-proximity-tracing/documents.

[241] Andreas Peter, Erik Tews, and Stefan Katzenbeisser. "Efficiently Outsourcing Multiparty Computation Under Multiple Keys". In: *IEEE Transactions on Information Forensics and Security* 8.12 (Dec. 2013), pp. 2046–2058. DOI: 10.1109/TIFS.2013.2288131.

[242] Giacomo Pope and Luca De Feo. *Is SIKE broken yet?* 2024. URL: https://issikebrokenyet.github.io/index.html.

[243] Manoj Prabhakaran and Mike Rosulek. "Rerandomizable RCCA Encryption". In: *CRYPTO 2007*. Ed. by Alfred Menezes. Vol. 4622. LNCS. Springer, 2007, pp. 517–534. DOI: 10.1007/978-3-540-74143-5_29.

[244] Shuo Qiu et al. "PPSI: Practical private set intersection over large-scale datasets". In: *SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2019*. IEEE, Aug. 2019, pp. 1249–1254. DOI: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00232.

[245] Charles Rackoff and Daniel R. Simon. "Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack". In: *Advances in Cryptology — CRYPTO '91*. Ed. by Joan Feigenbaum. Lecture Notes in Computer Science. Springer, 1992, pp. 433–444. ISBN: 978-3-540-46766-3. DOI: 10.1007/3-540-46766-1_35.

[246] Oded Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography". In: *J. ACM* 56.6 (Sept. 2009). ISSN: 0004-5411. DOI: 10.1145/1568318.1568324.

[247] Oded Regev. "The Learning with Errors Problem". In: *Proceedings of the 25th Annual IEEE Conference on Computational Complexity (CCC 2010)*. 2010, pp. 191–204. DOI: 10.1109/CCC.2010.26.

[248] Xuanle Ren et al. "HEDA: multi-attribute unbounded aggregation over homomorphically encrypted database". In: *Proceedings of the VLDB Endowment* 16.4 (2022), pp. 601–614.

[249] Jochen Rill. "Towards Applying Cryptographic Security Models to Real-World Systems". PhD thesis. Karlsruhe Institute of Technology, Germany, 2020. URL: https://nbn-resolving.org/urn:nbn:de:101:1-202004290457141991835 1.

[250] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. "On data banks and privacy homomorphisms". In: *Foundations of secure computation* 4.11 (1978), pp. 169–180.

[251] Ronald L. Rivest et al. *A Global Coalition for Privacy-First Digital Contact Tracing Protocols to Fight COVID-19*. URL: https://tcn-coalition.org/.

[252] Ronald L. Rivest et al. *The PACT protocol specification*. Apr. 8, 2020. URL: https://pact.mit.edu/wp-content/uploads/2020/04/The-PACT-protocol-specification-ver-0.1.pdf.

[253] Damien Robert. "Breaking SIDH in polynomial time". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 472–503.

[254] Alon Rosen and Gil Segev. "Chosen-Ciphertext Security via Correlated Products". In: 2009, pp. 419–436. DOI: 10.1007/978-3-642-00457-5_25.

[255] Ou Ruan, Xiongbo Huang, and Hao Mao. "An efficient private set intersection protocol for the cloud computing environments". In: *BigDataSecurity, HPSC and IDS 2020*. IEEE, May 2020, pp. 254–259. DOI: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00053.

[256] Valentin Rupp and Max von Grafenstein. "Clarifying "personal data" and the role of anonymisation in data protection law including and excluding data from the scope of the GDPR (more clearly) through refining the concept of data protection". In: *J. CLSR* 52.105932 (Apr. 2024), p. 25. DOI: 10.1016/j.clsr.2023.105932.

[257] Tushar Kanti Saha and Takeshi Koshiba. "Efficient private conjunctive query protocol over encrypted data". In: *Cryptography* 5.1 (2021), p. 2.

[258] Tushar Kanti Saha and Takeshi Koshiba. "Private conjunctive query over encrypted data". In: *International Conference on Cryptology in Africa*. Springer. 2017, pp. 149–164.

[259] Tushar Kanti Saha and Takeshi Koshiba. "Private equality test using ring-LWE somewhat homomorphic encryption". In: *2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*. IEEE. 2016, pp. 1–9.

[260] Tushar Kanti Saha, Mayank Rathee, and Takeshi Koshiba. "Efficient private database queries using ring-LWE somewhat homomorphic encryption". In: *Journal of Information Security and Applications* 49 (2019), p. 102406.

[261] Alessandra Scafuro and Tanner Verber. "A New Paradigm for Server-Aided MPC". In: *IACR Communications in Cryptology* 1.4 (2025).

[262] Jim Schaad, Blake Ramsdell, and Sean Turner. "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification". In: *RFC* 8551 (2019), pp. 1–63. DOI: 10.17487/RFC8551. URL: https://doi.org/10.17487/RFC8551.

[263] Berry Schoenmakers, Meilof Veeningen, and Niels de Vreede. "Trinocchio: Privacy-Preserving Outsourcing by Distributed Verifiable Computation". In: *ACNS 2016*. Ed. by Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider. Vol. 9696. LNCS. Springer, Cham, June 2016, pp. 346–366. DOI: 10.1007/978-3-319-39555-5_19.

[264] Peter Schwabe, Douglas Stebila, and Thom Wiggers. "Post-quantum TLS without handshake signatures". In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 1461–1480.

[265] Rebecca Schwerdt. "Sender-binding Encryption : Towards Finding the Weakest Encryption Security to Realise Secure Communication". 46.23.01; LK 01. PhD thesis. Karlsruher Institut für Technologie (KIT), 2025. 118 pp. DOI: 10.5445/IR/1000181209.

[266] Rebecca Schwerdt et al. "Sender-binding key encapsulation". In: *Cryptology ePrint Archive* (2023).

[267] Peter W Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.

[268] Victor Shoup. *A Proposal for an ISO Standard for Public Key Encryption*. Cryptology ePrint Archive, Paper 2001/112. 2001. URL: https://eprint.iacr.org/2001/112.

[269] Kris Shrishak. *Privacy Washing through PETs: the Case of Worldcoin*. 2025. URL: https://cacm.acm.org/blogcacm/privacy-washing-through-pets-the-case-of-worldcoin/.

[270] Signal Messenger LLC. *Is it private? Can I trust it?* https://support.signal.org/hc/en-us/articles/360007320391-Is-it-private-Can-I-trust-it. Accessed: 12 Nov. 2025. 2025.

[271] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. "Practical techniques for searches on encrypted data". In: *Proceeding 2000 IEEE symposium on security and privacy. S&P 2000*. IEEE. 2000, pp. 44–55.

[272] Junjie Song et al. "Lattice-Based Dynamic k-times Anonymous Authentication". In: *arXiv preprint arXiv:2509.21786* (2025).

[273] Emil Stefanov and Elaine Shi. "Multi-cloud oblivious storage". In: *CCS 2013*. CCS '13. Association for Computing Machinery, Nov. 2013, pp. 247–258. DOI: 10.1145/2508859.2516673.

[274] Arisa Tajima, Hiroki Sato, and Hayato Yamana. "Outsourced Private Set Intersection Cardinality with Fully Homomorphic Encryption". In: *ICMCS 2018*. IEEE, May 2018, pp. 1–8. DOI: 10.1109/ICMCS.2018.8525881.

[275] Riivo Talviste et al. "Applying secure multi-party computation in practice". In: *Ph. D. dissertation* (2016).

[276] Benjamin Hong Meng Tan et al. “Efficient private comparison queries over encrypted databases using fully homomorphic encryption with finite fields”. In: *IEEE Transactions on Dependable and Secure Computing* 18.6 (2020), pp. 2861–2874.

[277] Classic McEliece team. *Classic McEliece Homepage*. 2017. URL: https://classic.mceliece.org/nist.html (visited on 01/14/2025).

[278] FrodoKEM team. *FrodoKEM Homepage*. 2017. URL: https://frodokem.org/ (visited on 01/14/2025).

[279] Kyber Team. *Homepage Kyber*. 2016. URL: https://pq-crystals.org/kyber/.

[280] Shintaro Terada and Kazuki Yoneyama. “Improved Verifiable Delegated Private Set Intersection”. In: *ISITA 2018*. IEEE, Oct. 2018, pp. 520–524. DOI: 10.23919/ISITA.2018.8664310.

[281] Thales. *Data Threat Report*. 2024. URL: https://cpl.thalesgroup.com/data-threat-report.

[282] V. Thangam and K. Chandrasekaran. “Elliptic Curve Based Secure Outsourced Computation in Multi-party Cloud Environment”. In: *Security in Computing and Communications*. Ed. by Peter Mueller et al. Vol. 625. CCIS. Springer, Singapore, Sept. 2016, pp. 199–212. DOI: 10.1007/978-981-10-2738-3_17.

[283] *The Emissions Issue*. Volgswagen Annual Report 2015, Group Managenebt Report. Accessed: 2024-09-04. 2015. URL: https://annualreport2015.volkswagenag.com/group-management-report/the-emissions-issue.html.

[284] The Tor Project, Inc. *TOR Project*. URL: https://www.torproject.org/.

[285] Ni Trieu et al. “Epione: Lightweight Contact Tracing with Strong Privacy”. In: *IEEE Data Eng. Bull.* 43.2 (2020), pp. 95–107. URL: http://sites.computer.org/debull/A20june/p95.pdf.

[286] Carmela Troncoso et al. *Decentralized Privacy-Preserving Proximity Tracing*. Apr. 12, 2020. URL: https://github.com/DP-3T/documents/raw/master/DP3T%5C%20White%5C%20Paper.pdf.

[287] Carmela Troncoso et al. “Decentralized Privacy-Preserving Proximity Tracing”. In: *IEEE Data Eng. Bull.* 43.2 (2020). First published 3 April 2020 on https://github.com/DP-3T/documents, pp. 36–66. URL: http://sites.computer.org/debull/A20june/p36.pdf.

[288] Truesec. *Truesec Threat Intelligence Report 2024*. 2024. URL: https://insights.truesec.com/hub/report/truesec-threat-intelligence-report-2024.

[289] Nik Unger and Ian Goldberg. "Improved Strongly Deniable Authenticated Key Exchanges for Secure Messaging." In: *PoPETs* 2018.1 (2018), pp. 21–66.

[290] United Nations General Assembly. *The Right to Privacy in the Digital Age*. Resolution A/RES/68/167. Adopted 18 December 2013. 2013.

[291] Marten Van Dijk et al. "Fully homomorphic encryption over the integers". In: *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*. Springer. 2010, pp. 24–43.

[292] Serge Vaudenay. *Centralized or Decentralized? The Contact Tracing Dilemma*. May 6, 2020. iacr: 2020/531.

[293] Serge Vaudenay. "Centralized or decentralized? The contact tracing dilemma". In: (2020).

[294] Thijs Veugen et al. "A Framework for Secure Computations With Two Non-Colluding Servers and Multiple Clients, Applied to Recommendations". In: *IEEE Transactions on Information Forensics and Security* 10.3 (Nov. 2015), pp. 445–457. DOI: 10.1109/TIFS.2014.2370255.

[295] Ari Ezra Waldman. "Privacy law's false promise". In: *Wash. UL Rev.* 97 (2019), p. 773.

[296] Boyang Wang et al. "A tale of two clouds: Computing on data encrypted under multiple keys". In: *2014 IEEE Conference on Communications and Network Security*. IEEE. 2014, pp. 337–345.

[297] Boyang Wang et al. "Computing encrypted cloud data efficiently under multiple keys". In: *CNS 2013*. IEEE, Oct. 2013, pp. 504–513. DOI: 10.1109/CNS.2013.6682768.

[298] Wenhao Wang et al. "Toward Scalable Fully Homomorphic Encryption Through Light Trusted Computing Assistance". In: *CoRR* abs/1905.07766 (2019). arXiv: 1905.07766. URL: https://arxiv.org/abs/1905.07766.

[299] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. "Global-scale secure multiparty computation". In: *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*. 2017, pp. 39–56.

[300] Xu An Wang et al. “A privacy-preserving fuzzy interest matching protocol for friends finding in social networks”. In: *Soft Computing* 22 (Feb. 2018), pp. 2517–2526. DOI: 10.1007/s00500-017-2506-x.

[301] Rui Wen et al. “Leaf: A faster secure search algorithm via localization, extraction, and reconstruction”. In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 1219–1232.

[302] WhatsApp LLC. *About end-to-end encryption*. https://faq.whatsapp.com/820124435853543. Accessed: 12 Nov. 2025. 2025.

[303] WHO. *WHO Director-General’s opening remarks at the media briefing – 5 May 2023*. 2023. URL: https://www.who.int/news-room/speeches/item/who-director-general-s-opening-remarks-at-the-media-briefing---5-may-2023.

[304] Yulin Wu et al. “Generic server-aided secure multi-party computation in cloud computing”. In: *Comput. Stand. Interfaces* 79 (2022), p. 103552. DOI: 10.1016/J.CSI.2021.103552. URL: https://doi.org/10.1016/j.csi.2021.103552.

[305] Lida Xu et al. “Cloud-Assisted Privacy-Preserving Spectral Clustering Algorithm Within a Multi-User Setting”. In: *IEEE Access* 12 (2024), pp. 75965–75982.

[306] Shengfeng Xu and Xiangxue Li. “Chosen-Ciphertext Secure Key Encapsulation Mechanism in the Standard Model”. In: *IEEE Access* 9 (2021), pp. 13683–13690.

[307] Xiaopeng Yang, Wenping Ma, and Chengli Zhang. “Efficient chosen ciphertext secure key encapsulation mechanism in standard model over ideal lattices”. In: *International Journal of Computer Mathematics* 94.5 (2017), pp. 866–883.

[308] Andrew C Yao. “Protocols for secure computations”. In: *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE. 1982, pp. 160–164.

[309] Masaya Yasuda. “Secure Hamming distance computation for biometrics using ideal-lattice and ring-LWE homomorphic encryption”. In: *Information Security Journal: A Global Perspective* 26.2 (2017), pp. 85–103.

[310] Masaya Yasuda et al. “Secure pattern matching using somewhat homomorphic encryption”. In: *Proceedings of the 2013 ACM workshop on Cloud computing security workshop*. 2013, pp. 65–76.

[311] Yu Yu and Jiang Zhang. “Cryptography with Auxiliary Input and Trapdoor from Constant-Noise LPN”. In: *CRYPTO 2016*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9814. Lecture Notes in Computer Science. Springer, 2016, pp. 214–243. DOI: 10.1007/978-3-662-53018-4_9. URL: https://doi.org/10.1007/978-3-662-53018-4%5C_9.

[312] Daode Zhang et al. “Lattice-Based Dual Receiver Encryption and More”. In: 2018, pp. 520–538. DOI: 10.1007/978-3-319-93638-3_30.

[313] Zhou Zhang et al. “ArcEDB: an arbitrary-precision encrypted database via (amortized) modular homomorphic encryption”. In: *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 2024, pp. 4613–4627.

[314] Qingji Zheng and Shouhuai Xu. “Verifiable Delegated Set Intersection Operations on Outsourced Encrypted Data”. In: *IC2E 2015*. IEEE, Mar. 2015, pp. 175–184. DOI: 10.1109/IC2E.2015.38.

[315] Guy Zyskind et al. “High-Throughput Universally Composable Threshold FHE Decryption”. In: *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*. 2025, pp. 2339–2353.

A. Appendix

A.1. Related Work on Outsourced PSI

To the best of our knowledge most of the literature focuses on protocols where only one server is involved and therefore collusion with the only-existing server is explicitly not assumed. Also many of those protocols are highly interactive. Interactiveness means that the input clients need to be online more than once and sometimes at the same time in order to communicate with each other. This may be useful in other applications, such as when input clients must explicitly approve their inputs being used together with inputs from specific other clients. However, in our case, this is not an issue. On the other hand, there are plenty of applications and use cases with systems that are always online, where the non-interactiveness may not be a necessary functional requirement. In such cases, the following works may be more suitable than our work. Such outsourced PSI works are Miyaji et al. [215] and Wang et al. [300]

A special category of such interactive protocols are multi-key homomorphic encryption protocols or threshold homomorphic encryption schemes, which inherently require clients to be online during the decryption stage and perform a collaborative protocol in addition to the communication in the outsourcing phase, such as in [25, 99, 220].

The second line of related outsourced PSI constructions assume the non-collusion between the single server and any client, which we deem dangerous in the setting of our case study. This is unsurprising, as PSI is an MPC protocol originally designed for two or more *peer* parties. Having a centralized server with no inputs of its own and no legitimate reason to learn any information about the output of the PSI can lead to the server learning this information, due to the collusion with an input client or an initiating client. One could argue that this, in some way, is detrimental to the original MPC security definition. As a result, scenarios where adversaries can corrupt both a server

and a client simultaneously are typically excluded, and protection measures for such cases are usually not described. This leads to many of these protocols relying on the assumption of non-collusion between server and client, which, as discussed, introduces a single point of failure. However, in real-world settings where the threshold for client participation should be as low as possible, administrators of the server infrastructure–who can create accounts themselves and thus access a dummy client's state–could carry out this attack, violating the passive server-client non-collusion assumption. The following works deliberately do not consider this attack, such as those by [5, 6, 7, 8, 125, 177, 180, 255, 274, 280].

The remaining works fall into the category of miscellaneous protocols, which are not suitable for our application or incomparable for various reasons. Many of these works lack a (simulation-based) proof, despite having theorems described, making it difficult to compare them in terms of their adversary model. One such work is by Zheng et al. [314], where they require a dedicated trusted third party, which is outside the scope of our use case. This argument may seem peculiar because we require a key registration with knowledge in Π_{OutComp}, a specialized variant of a public key infrastructure, which is itself a trusted third party. However, we argue that such variants of trusted third parties are more standard and already widely available in the real-world. While Ali et al. [16] provide an extensive proof, their threat/adversary model is highly specific. In their model, the authors distinguish between unauthorized and authorized clients, which is not applicable to our scenario. Furthermore, they assume a trusted third party, referred to as a certificate authority (CA), but this differs significantly from the standard definition. In their case, the CA generates key pairs for the clients, which is not the case in our key registration with knowledge, as the key generation is realized within the real protocol by the parties themselves. Given that our security definition is non-trivial, we exclude works that do not provide a proper proof, as this hinders a meaningful comparison with our own work, i. e., [232, 244]

Overall, we have identified one heuristic that seems to be either having a partially interactive outsourced PSI protocol that is secure against server-client collusions, or excluding server-client collusions and, in return, being fully non-interactive after the outsourcing phase. This is unsurprising, as the server-client non-collusion assumption allows secret information to be hidden within the client's state, which can be used to protect all clients from an honest-but-curious server. Our protocol, on the other hand, achieves full non-interactiveness while also being secure in the case of a server-client

collusion, which we consider more dangerous in our use case than server-server collusions. Also, to the best of our knowledge, there does not exist one OPSI protocol that was proven within the UC model but rather in the weaker real-ideal simulation paradigm (a. k. a. standalone security or sequential composability model).

A.2. IND-CPA DRE via 2-repetition McEliece

Let $PKE_{\text{McE},2} = (\text{Gen}_{\text{McE},2}, \text{Enc}_{\text{McE},2}, \text{Dec}_{\text{McE},2})$ be a verifiable 2-repetition encryption scheme, which is a variant ($k = 2$) from [131] based on the McEliece cryptosystem, and $M = \{0,1\}^l$, where $l = l_1 + l_2$ (as in [230]). We define the cryptosystem as follows.

- The key generation algorithm $\text{Gen}_{\text{McE},2}(1^n)$ works as follows:
  - Sample a generator matrix $G' \in \{0,1\}^{l \times n}$ of an irreducible binary Goppa code, which can correct up to t errors with a code dimension l.
  - Sample a random non-singular matrix $S \in \{0,1\}^{l \times l}$.
  - Sample a random permutation matrix $P \in \{0,1\}^{n \times n}$.
  - Set $G := SG'P$.

  ↪ Return $\text{pk} = (G, t)$ and $\text{sk} = (S, G', P)$
- The encryption algorithm $\text{Enc}_{\text{McE},2}(\text{pk}_R, \text{pk}_S, m)$ works as follows:
  - Parse pk_R as (G_R, t) and pk_S as (G_S, t)
  - Sample $s \leftarrow_{\$} \{0,1\}^{l_1}$, where $l_1 \in \Omega(n)$.
  - $e_R, e_S \leftarrow \mathcal{B}_\theta$, where $\mathcal{B}_\theta$ is the Bernoulli distribution with $\theta = \frac{t}{n} - \varepsilon$ for some $\varepsilon > 0$.
  - $c_R = [s|m] \cdot G_R \oplus e_R$
  - $c_S = [s|m] \cdot G_S \oplus e_S$

  ↪ Return $c = (c_R, c_S)$.
- The decryption algorithm $\text{Dec}_{\text{McE},2}(\text{sk}_R, \text{pk}_S, c)$ works as follows:

- Parse c as (c_R, c_S) and sk_R as (S_R, G'_R, P_R)
- Compute $\hat{y}_R = c_R \cdot P_R^{-1} = ([s|m]S_R) \cdot G'_R \oplus e_R \cdot P_R^{-1}$
- Compute $[s|m] \cdot S_R = Correct(\hat{y}_R)$
- Compute $[s|m] = ([s|m]S_R)S_R^{-1}$
- Compute $c'_S = [s|m] \cdot G_S$
- Set the verification bit b as follows
  - Set $b = 1$ if the hamming weight of $c'_S \oplus c_S$ is smaller than t.
  - Set $b = 0$ otherwise.

$\hookrightarrow$ If $b = 1$ return m, else $\perp$

Theorem 16 The encryption scheme $PKE_{\text{McE},2}$ is IND-CPA secure, given that both the McEliece assumption and the LPNDP hold. In particular, let $\mathcal{A}$ be an IND-CPA adversary against $PKE_{\text{McE},2}$. Then there is a distinguisher $\mathcal{B}$ for Goppa codes and a distinguisher $\mathcal{D}$ for the LPNDP, such that for all $\lambda \in \mathbb{N}$

$$\mathsf{Adv}^{\text{CPA}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\text{LPNDP}_\theta(2n,l)}_{\mathcal{D}}(\lambda) + 2 \times \mathsf{Adv}^{\text{ind}}_{\mathcal{B}_R, G_R}(\lambda).$$

The original proof can be found in [131]. However, the authors did not explicitly state the advantage of the adversary.

Proof **Game 1** This is the DRE IND-CPA game.

Game 2 Same as Game 1, except that the generator matrix G_R within the public key is replaced by uniformly random matrix $U_R \in \{0,1\}^{l \times n}$. Therefore, the receiver public key in Game 2 is $pk_R := (U_R, t)$.

Any distinguisher $\mathcal{A}_R$ distinguishing between Game 1 and Game 2 yields a distinguisher $\mathcal{B}_R$ for a random irreducible Goppa code from a random linear code. Therefore,

$$\mathsf{Adv}^{CPA}_{\mathcal{A}} \leq \mathsf{Adv}^{CPA}_{\mathcal{A},\text{Game 2}} + \mathsf{Adv}^{ind}_{\mathcal{B}_R, G_R}(\lambda)$$

Game 3 Same as Game 1, except that the generator matrix G_S within the public key is replaced by uniformly random matrix $U_S \in \{0,1\}^{l \times n}$. Therefore, the sender public key in Game 3 is $pk_S := (U_S, t)$.

Any distinguisher $\mathcal{A}_S$ distinguishing between Game 2 and Game 3 yields a distinguisher $\mathcal{B}_S$ for a random irreducible Goppa code from a random linear code. Therefore, w.l.o.g

$$\mathsf{Adv}^{CPA}_{\mathcal{A}} \leq \mathsf{Adv}^{CPA}_{\mathcal{A},\text{Game 3}} + 2 \times \mathsf{Adv}^{ind}_{\mathcal{B}_R,\mathcal{G}_R}(\lambda) \tag{A.1}$$

Game 4 Instead of computing the challenge ciphertext as

$$c^* = ([s|m_b]U_S \oplus e_S, [s|m_b]U_R \oplus e_R)$$

the challenger chooses $c^* = (c_1, c_2)$ with $c_1, c_2 \leftarrow_\$ \mathcal{U}_n$ instead.

The indistinguishability of Game 4 from Game 3 is shown as follows.

- Observe $\forall i \in \{R, S\}$ that $U_i^T = \left(U_{i,1}^T | U_{i,2}^T\right)$ with $U_{i,1}^T \in \{0,1\}^{l_1 \times n}$ and $U_{i,2}^T \in \{0,1\}^{l_2 \times n}$ s.t. $l_1 + l_2 = l$.
- Then $\forall i \in \{R, S\}$ the ciphertext can be transformed as $[s|m_b]U_i \oplus e_i = (s \cdot U_{i,1} \oplus e_i) \oplus m \cdot U_{i,2}$

Thus, the ciphertext can be transformed into:

$$\begin{aligned} c^* &= ([s|m_b]U_S \oplus e_S, [s|m_b]U_R \oplus e_R) \\ &= \left((s \cdot U_{S,1} \oplus e_S) \oplus m_b \cdot U_{S,2}, (s \cdot U_{R,1} \oplus e_R) \oplus m_b \cdot U_{R,2}\right) \end{aligned}$$

Firstly, set the matrices $U_1 = \left(U_{S,1}|U_{R,1}\right) \in \{0,1\}^{l_1 \times 2n}$ and $U_2 = \left(U_{S,2}|U_{R,2}\right) \in \{0,1\}^{l_2 \times 2n}$. Secondly, summarize the error vectors into one, i.e. $e = (e_s|e_R) \in \{0,1\}^{2n}$ The ciphertext is now:

$$c^* = ((s \cdot U_1 \oplus e) \oplus m_b \cdot U_2)$$

Finally, we can interpret $(s \cdot U_1 \oplus e)$ as an instance of the LPNDP and replace by a random value $u \leftarrow_\$ \mathcal{U}_{2n}$. The random vector u acts as a OTP s.t. the ciphertext is transformed into a uniformly distributed random vector:

$$c^* = (u \oplus m_b \cdot U_2)$$

This is the challenge ciphertext used in Game 4. The advantage of the original DRE IND-CPA adversary $\mathcal{A}$ is now 0, as the succeeding probability is $\frac{1}{2}$. The indistinguishability follows from the hardness of

the LPNDP.
If the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has non-negligibly different succeeding probabilities in Games 3, 4 then we can use this adversary to solve any LPNDP. To this end we can use the following distinguisher D for a given LPNDP oracle $\mathcal{O}$, which is either $\mathcal{Q}_{s,\theta}$ with $s \in \{0,1\}^l$ or $\mathcal{U}_{l+1}$.

1. Generate the public keys U_R, U_S for $\mathcal{A}$ in 2 steps. Remember, that $U_R = (U_{R,1}|U_{R,2})$ (resp. U_S).
   - Call the LPNDP oracle for enough samples $(a_1, b_1), \cdots, (a_{2n}, b_{2n}) \leftarrow \mathcal{O}$.
   - Set $b_R = (b_1|\cdots|b_n)$ and $b_S = (b_{n+1}|\cdots|b_{2n})$.
   - Set $U_{R,1} = (a_1|\cdots|a_n)$ and $U_{S,1} = (a_{n+1}|\cdots|a_{2n})$.
   - Sample the remaining part of the public key uniformly random, i.e. $U_{R,2} \leftarrow_\$ \{0,1\}^{l_2 \times n}$ (resp. $U_{S,2}$).

   Finally, the public keys are

$$\mathsf{pk}_R = U_R = (U_{R,1}|U_{R,2}) \in \{0,1\}^{l \times n}$$
$$\mathsf{pk}_S = U_S = (U_{S,1}|U_{S,2}) \in \{0,1\}^{l \times n}$$

2. $(m_0, m_1) \leftarrow \mathcal{A}_1(U_R, U_S)$
3. $b \leftarrow_\$ \{0,1\}$
4. Set the challenge ciphertext to

$$c^* = (c_1, c_2) = \big((b_R \oplus m_b \cdot U_{R,2}), (b_S \oplus m_b \cdot U_{S,2})\big)$$

5. $b' \leftarrow \mathcal{A}_2(U_R, U_S, c^*)$
6. If $b' = b$ then return 1, else return 0.

If $\mathcal{O} = \mathcal{Q}_{s,\theta}$, then we have the same situation as in Game 3, else $\mathcal{O} = \mathcal{U}_{l+1}$ and we have the same situation as in Game 4. Therefore:

$$\mathsf{Adv}^{CPA}_{\mathcal{A},Game3} \leq \mathsf{Adv}^{CPA}_{\mathcal{A},\text{Game 4}} + \mathsf{Adv}^{LPNDP_\theta(2n,l)}_{\mathcal{D}}(\lambda) = \mathsf{Adv}^{LPNDP_\theta(2n,l)}_{\mathcal{D}}(\lambda)$$

This concludes that the overall advantage is

$$\mathsf{Adv}^{CPA}_{\mathcal{A}} \leq \mathsf{Adv}^{LPNDP_\theta(2n,l)}_{\mathcal{D}}(\lambda) + 2 \times \mathsf{Adv}^{ind}_{\mathcal{B}_R,G_R}(\lambda) \qquad \square$$

The following Theorem 17 was already implicitly shown in [190] by reducing it to the property of verifiability of a verifiable k-repetition PKE from [131].

Theorem 17 The encryption scheme $(\mathsf{Gen}_{\mathrm{McE},2}, \mathsf{Enc}_{\mathrm{McE},2}, \mathsf{Dec}_{\mathrm{McE},2})$ satisfies DRE-soundness.

The definition of the soundness property of DRE can be found in Definition 34.

Proof Consider the case $\mathsf{Dec}(\mathsf{pk}_S, \mathsf{sk}_R, c) = \perp$ and $\mathsf{Dec}(\mathsf{pk}_R, \mathsf{sk}_S, C) = m$. We will prove by contradiction that this case never happens. Parse C as (c_R, c_S) and $\mathsf{pk}_R = G_R$ and $\mathsf{pk}_S = G_S$, which ultimately have the following form due to being textbook McEliece ciphertexts.

$$c_R = m'G_R \oplus e_R$$
$$c_S = mG_S \oplus e_S$$

From $\mathsf{Dec}(\mathsf{pk}_S, \mathsf{sk}_R, c) = \perp$ it follows that the verification step has failed. This means that after recovering m' from c_R by $\mathsf{Dec}_{McE}(\mathsf{sk}_R, c_R) = m'$ the hamming distance of $m'G_S$ has to be greater or equal than t to c_S. Considering that $m'G_S \oplus c_S = m'G_S \oplus mG_S \oplus e_S$ we get

$$wgt(m'G_S \oplus mG_S \oplus e_S) \geq t$$

From this it follows that $m' \neq m$ due to e_S being guaranteed to have the hamming weight $wgt(e_S) < t$ by the syndrome decoding algorithm within the textbook McEliece decryption.
However, from $\mathsf{Dec}(\mathsf{pk}_R, \mathsf{sk}_S, c) = m$ it follows that

$$wgt(mG_R \oplus m'G_R \oplus e_R) < t$$

Now mG_R and $m'G_R$ are codewords for $m \neq m'$ and therefore are guaranteed to have hamming distance $d(mG_R, m'G_R) \geq 2t + 1$. This contradicts with $wgt(mG_R \oplus m'G_R \oplus e_R) < t$ as $wgt(e_R) < t$. Therefore, this case is not possible.
The same considerations will yield that the case $\mathsf{Dec}(\mathsf{pk}_S, \mathsf{sk}_R, c) = m_R$ and $\mathsf{Dec}(\mathsf{pk}_R, \mathsf{sk}_S, c) = m_S$ with $m_R \neq m_S$ is impossible.
Conclusively, $\Pr[\mathsf{Exp}^{\mathrm{sound}}_{\mathcal{A},\Pi} = 1] = 0$. □