Karlsruhe Institute of Technology

KASTEL

Survey on Randomness Homomorphic Commitment Schemes

Bachelor's Thesis of

Simon Wülker

At the KIT Department of Informatics
KASTEL – Institute of Information Security and Dependability

First examiner:	Prof. Dr. Jörn Müller-Quade
Second examiner:	Prof. Dr. Thorsten Strufe
First advisor:	M.Sc. Astrid Ottenhues
Second advisor:	M.Sc. Eva Hetzel

02. April 2024 – 01. August 2024

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself. I have not used any other than the aids that I have mentioned. I have marked all parts of the thesis that I have included from referenced literature, either in their original wording or paraphrasing their contents. I have followed the by-laws to implement scientific integrity at KIT.

Karlsruhe, 01.08.2024

. . . . . .

(Simon Wülker)

Abstract

Commitment schemes are an important cryptographic primitive that allow a committer to decide on a hidden value and possibly reveal it later. They are commonly used for zero knowledge proofs and other protocols like secure coin flipping. Randomness subversion attacks involve an attacker subverting the generation of random numbers to compromise security. To defend against these attacks, rerandomizable commitment schemes are of special interest since they allow modifying the randomness inside a commitment without revealing it. Homomorphic commitment schemes allows for easy rerandomization by combining a potentially subverted commitment with a "fresh" one, without modifying the message itself. Despite this, little research has been done to survey the available options for implementors.

In this work, we present a list of five practical randomness-homomorphic commitment schemes from publications in the field along with proofs of their security. We study their constructions in detail and compare them in terms of both the security they provide and their performance. The presented schemes depend on the hardness of problems from different domains, including the discrete logarithm problem and well known lattice problems like the Short Integer Solution and Learning with Errors problems. Finally, we give recommendations about the applications of the presented schemes.

Zusammenfassung

Commitment-Schemata sind ein wichtiger Teil der Kryptographie, der es einem Committer ermöglicht, sich für einen verborgenen Wert zu entscheiden und ihn möglicherweise später offenzulegen. Sie werden häufig für Zero-Knowledge-Beweise und andere Protokolle wie sicheres Münzwerfen verwendet. Bei Randomness Subversion Attacks untergräbt ein Angreifer die Generierung von Zufallszahlen, um die Sicherheit zu gefährden. Zur Abwehr dieser Angriffe sind rerandomisierbare Commitment-Schemata von besonderem Interesse, da sie es ermöglichen, die Zufälligkeit innerhalb eines Commitments zu ändern, ohne es offenzulegen. Homomorphe Commitment-Schemata ermöglichen eine einfache Rerandomisierung, indem ein möglicherweise untergrabenes Commitment mit einem „frischen“ kombiniert wird, ohne die Nachricht selbst zu ändern. Trotzdem wurde bisher wenig Forschung betrieben, um die verfügbaren Optionen für Entwickler zu untersuchen.

In dieser Arbeit präsentieren wir eine Liste von fünf praktischen Randomness-homomorphen Commitment-Schemata aus Veröffentlichungen auf diesem Gebiet zusammen mit Beweisen ihrer Sicherheit. Wir untersuchen ihre Konstruktionen im Detail und vergleichen sie sowohl hinsichtlich der von ihnen gebotenen Sicherheit als auch ihrer Leistung. Die vorgestellten Schemata hängen von der Schwierigkeit von Problemen aus verschiedenen Bereichen ab, darunter das diskrete Logarithmusproblem und bekannte Gitterprobleme wie die Short Integer Solution und Learning with Errors. Abschließend geben wir Empfehlungen zu den Anwendungen der vorgestellten Schemata.

Contents

List of Figures

List of Tables

1 Introduction

Commitment schemes are two-party protocols between a committer and a receiver. They allow the committer to decide on a chosen value without others being able to identify the value, while being able to later reveal it.
A commitment scheme can be thought of as a two-stage process:

1. Commitment: The committer generates a commitment value, which is a cryptographic representation of the value or statement they want to commit to. The commitment value should not reveal any information about the committed value or statement, while not allowing the committer to change the value without changing the commitment.

2. Unveiling: The committer reveals the committed value or statement to the receiver. This can be done by providing the receiver with a decommitment information that allows them to open or recompute the commitment value.

In some cases the number of possible messages is very small. For example, bit commitments only allow the committer to commit to either 0 or 1. To prevent adversaries from being able to exhaustively try all openings for a commitment, an element of randomness is usually included in addition to the message.

Randomness homomorphic commitments are homomorphic with respect to the randomness inside them. This thesis deals with the question of what randomness homomorphic commitment schemes are available and how they compare to each other. Specifically, we compare the strengths of their security, the problems their security relies on and their performance.

1.1 Applications of Rerandomizable Commitment Schemes

Since the revelations of Snowden in 2013 we know that state-level attackers try to subvert the generation of randomness on computers to weaken the security of cryptographic protocols. There are numerous known practical randomness subversion attacks against popular cryptosystems [46]. In the case of commitment schemes, one possible approach to defending against these attacks is to introduce a rerandomizing reverse firewall [14] [4]. Randomness homomorphic commitment schemes are especially easy to rerandomize and are therefore of special interest.

Rerandomizable commitment schemes are also used in Single Secret Leader Elections (SSLE). SSLE allow multiple parties to randomly choose a leader among them, with the restriction that the identity of the leader will only be known to the leader themselves. Later, the leader can reveal their identity and prove it to the rest of the group. This is useful for example in the context of proof of stake blockchains [5]. This problem was first introduced by Boneh et al. [9, 10].

There are cryptographic operations that cannot be achieved without a common reference string (CRS). Their security depends on the trust granted to the authority that generates the CRS. In practice, there exists no authority that is trusted absolutely. Ananth et al [3] use the existence of rerandomizable commitment schemes to construct a protocol that allows the authority to be held accountable if it were to misuse its powers. To implement this protocol in practice, it is required to have a list of suitable commitment schemes to choose from. This thesis aims to draw a comparison between popular options.

1.2 Related Work

Previous research has been done to survey the construction of commitment schemes. While none focus on randomness homomorphic commitment schemes specifically, there is prior work for other sorts of commitments and homomorphic encryption.

Vector commitments allow to commit on a sequence of values and later reveal one or more of them at a time. Similar to our work, Anca Nitulescu [37] compares a number of different vector commitment schemes on their properties such as runtime complexity of generating and verifying the commitment value and their updateability, ie. the ability to move values in the committed sequence around.

Multivariate polynomial commitment schemes are a group of commitment schemes where the value being committed to is a polynomial involving more than one variable. Ihyun Nam [36] surveys multiple of these commitment schemes, studies their construction and compares them based on runtime complexity and the assumptions their security is based on.

Homomorphic commitment schemes may be constructed from homomorphic encryption schemes. One example of this is the (homomorphic) ElGamal encryption [23], which can be used to construct a homomorphic commitment scheme [1]. This scheme is also demonstrated in Section 3.1.2. Marcolla et al. [35] evaluate several different homomorphic encryption algorithms and compares their performance and evolution over time, as well as the availability of implementations.

1.3 Contributions

We demonstrate five commitment schemes relying on the hardness of different mathematical problems. The selected schemes cover both problems from classic assumptions like the discrete logarithm and problems from the lattice domain, like Short Integer Solution or Learning with Errors. In contrast to the research listed in Section 1.2, we explicitly only survey randomness-homomorphic commitment schemes. We provide example implementations of the schemes and compare their performance to each other.

1.4 Outline

Chapter 2 defines the core properties of commitments and other terms used throughout the work, as well as the mathematical problems that the hardness of the presented commitments depends on. Chapter 3 present the five chosen commitment schemes in detail and provides proofs of their security. It also lists a few more randomness homomorphic commitment schemes that were not included in earlier listing as well as the reasoning behind this decision. Chapter 4 draws comparisons between the schemes presented earlier and gives recommendations about selecting a scheme for a particular application. Lastly, Chapter 5 summarizes the findings and provides an outlook for future research opportunities.

2 Preliminaries

To discuss the advantages and disadvantages of various commitment schemes, we must first give definitions for the schemes themselves and their properties.

We describe a commitment in terms of three algorithms **Setup**, **Commit** and **Verify**. The purpose of these algorithms is detailed in Figure 2.1.

Setup:	The parties agree on any shared parameters required by the scheme.
Commit:	The committer computes a commitment value.
Verify:	The verifier validates the commitment after receiving the opening information by the committer.

Figure 2.1: Description of the three steps in a commitment

We notate the commitment value as $Com(m, r)$, where Com is the **Commit** algorithm of the scheme in question, m is the committed value and r is a random value.

To fulfill the criteria of a commitment scheme, there must be no feasible way to compute m from only $Com(m, r)$ with a probability that significantly exceeds guessing. This is known as the *Hiding* property [18].

Secondly, for any commitment $Com(m, r)$, there must be no feasible way to find a message m' and a random vector r' such that $Com(m, r) = Com(m', r')$, as this would allow the committer to reveal a message other than the one it committed to earlier. This is the *Binding* property [18].

Subversion Attacks Subversion attacks are attacks where the adversary undermines parts of the system, without the target noticing. These attacks have received widespread attention after the Snowden leaks. They are nontrivial to detect and sabotage the security of the system as a whole [46]. One example of how such a backdoor can impact security of such a backdoor is demonstrated by Zhichao Yang et al. [46], where the authors demonstrate a practical randomness subversion attack against the LWE encryption scheme.

2.1 Properties of Commitment Schemes

Definition 2.1.1 (Correctness)**.** A commitment scheme is said to be *correct* if the **Verify** algorithm from Figure 2.1 succeeds for every $Com(m, r)$ using opening information for m, r [43]. Informally, a commitment scheme is correct if correct openings for valid commitments are also valid.

Definition 2.1.2 (Rerandomizability)**.** A commitment is *rerandomizable* [10] if it is possible to transform a commitment $Com(m, r)$ on a value m into a new commitment $Com(m, r')$ where $r \neq r'$, without knowledge of m or r.

Definition 2.1.3 (Homomorphy)**.** A commitment scheme is *homomorphic* if the commitment value c, the message m and the random value r belong to abelian groups G_c, G_m and G_r and the combination of two commitments yields a commitment containing the combination of the two messages [30].

$$Com(m, r) \oplus Com(m', r') = Com(m \otimes m', r \otimes r') \tag{2.1}$$

Any homomorphic commitment is rerandomizable, since there exists an identity element $e_m \in G_m$. This allows for rerandomization using $Com(e_m, r')$:

$$\begin{aligned} Com(m, r) \oplus Com(e_m, r') &= Com(m \otimes e_m, r \otimes r') \\ &= Com(m, r \otimes r') \end{aligned}$$

We refer to such commitments as randomness-homomorphic.

Definition 2.1.4 (Malleability)**.** *Malleable* commitments [15] are a generalization of homomorphic commitments. A commitment is said to be malleable if, given a commitment $Com(m, r)$ on m, it is possible to derive a commitment $Com(m', r')$ on m', such that m and m' are related, but different.

Definition 2.1.5 (Equivocability)**.** *Equivocable* commitment schemes [7, 21] allow the committer to reveal different committed messages for the same commitment, which are indistinguishable to the verifier. This means that for a (valid) commitment $Com(m, r)$, the committer can later choose to unveil the commitment as if they had committed on a different value m'.

This is also referred to as the *trapdoor* property. A commitment is said to be *perfectly trapdoor* if it can be opened up to *any* message [30].

Definition 2.1.6 (Extractability)**.** *Extractable* commitments [20, 1] are commitments for which there exists an algorithm indistinguishable from **Setup** that, in addition to the usual setup, sets up a trapdoor. Knowledge of the trapdoor allows extraction of m from any commitment $Com(m, r)$. This is only possible for computationally hiding commitment schemes. For example, in the case of a commitment based on an encryption scheme the decryption key would be the trapdoor.

Extractability and Equivocability might seem like anti-features at first, given that they explicitly violate the hiding and binding properties. However, we reiterate that these violations are only possible given the secrets generated during **Setup**. Without them, all other assumptions still hold. Equivocable commitments are used for constructing non-malleable commitments, zero knowledge proofs and secure signature schemes [25]. Both extractable and equivocable commitments are also used to construct universally-composable commitment schemes [16].

The strength of a commitment schemes security properties is generally divided into three distinct categories.

A commitment is

- *computationally hiding* if the advantage of a probabilistic polynomial-time (PPT) attacker compared to guessing m is negligible.
- *statistically hiding* if the advantage of an unbounded attacker over randomly guessing m is negligible.
- *perfectly hiding* if an unbounded attacker has no advantage over guessing m.

These categories are defined in a similar way for the binding property. A commitment is

- *computationally binding* if the advantage of a polynomial-time committer compared to guessing m', r' is negligible.
- *statistically binding* if the advantage of an unbounded committer compared to guessing m', r' is negligible.
- *perfectly binding* if an unbounded committer has no advantage compared to guessing m', r'.

2.2 Computational Problems

The security of a commitment scheme depends on the hardness of some mathematical problem. This section introduces the mathematical problems that the presented commitment schemes depend on.

2.2.1 Discrete Logarithm

The discrete logarithm problem was first introduced along with the similar Diffie-Hellman problem [22]. For a group G, a generator g of G and some $h \in G$ it asks to solve $g^x = h$ for x. This value can easily be computed for a few specific groups, like multiplicative powers of ten. Whether the discrete logarithm in general can be computed in polynomial time on a classical computer is an open problem in computer science. A quantum computer theoretically able to efficiently compute discrete logarithms using Shors Algorithm [44] and

schemes based on the hardness of doing so will therefore usually not offer post-quantum resistance.

2.2.2 Computational Diffie-Hellman

The *Computational Diffie-Hellman* (CDH) problem [22] states that for a group G with generator g and random $a, b \in G$ given only (g, g^a, g^b) it is hard to compute g^{ab}. The security of the CDH problem relies on the hardness of the discrete logarithm problem.

2.2.3 Short Integer-Solution

For a random matrix $A \in \mathbb{Z}_q^{n \times m}$ and some positive real number β, the Short Integer Solution (SIS) problem requires finding a $z \in \mathbb{Z}^m$ with $|z| < \beta$ such that $Az = 0$. This is similar to solving regular linear systems, but with the additional constraint that the norm of z must be sufficiently small. β must be smaller than q, otherwise $z = (q, 0 \ldots 0) \in \mathbb{Z}^m$ would always be a trivial solution. This problem was first introduced in the seminal work of Ajtai [2]. A more in-depth analysis can be found in [39].

2.2.4 Learning with Errors

Learning with Errors (LWE) is parameterized by a public matrix $A \in \mathbb{Z}^{n \times m}$, some secret $s \in \mathbb{Z}_q^n$ and a secret vector of uniformly random, small error values e. It uses the result of computing $y = As + e$, effectively multiplying s with each row in A and then adding a small error [39]. There are multiple variations of the problem:

The *Decisional LWE Problem* states that it is computationally hard to distinguish y from uniformly random noise [40].

The *Search-LWE Problem* aims to find the secret s given only y [40].

3 Analyzed Commitment Schemes

Throughout this chapter, various different commitment schemes are presented and analyzed. Each scheme is analyzed individually, a comparison between them is later drawn in Chapter 4.

3.1 Discrete Logarithm Schemes

The following commitment schemes are all based on the discrete logarithm problem explained in Section 2.2, or variations thereof. While these are usually simple and performant, they are assumed to be efficiently solvable on a quantum computer.

3.1.1 Pedersen

Pedersen Commitments [38] were one of the first commitment schemes developed and are widely used in literature due to their simplicity and efficiency [28, 29]. Their high popularity and extensibility is the primary reason for their inclusion in this thesis. A definition of the commitment algorithms can be found in Figure 3.1. While the message space in Pedersen commitments themselves is limited to scalar values from $\mathbb{Z}_q$, they have been extended in a variety of ways to support more complex messages. Peiheng Zhang et al. [47] extend Pedersen commitments to polynomials. Fischlin and Fischlin [26] developed an interactive version that is no longer malleable.

What follows are proofs for the fundamental security properties of Pedersen commitments.

Correctness Notice that the computation of the commitment in Figure 3.1 is fully deterministic. Therefore, when the verifier recomputes the commitment in the **Verify** phase, it will always end up computing the same value and accept the commitment if the provided parameters m, r are correct.

Randomness Homomorphy Combining two commitments $Com(m_1, r_1)$ and $Com(m_2, r_2)$ yields

$$g^{m_1} h^{r_1} g^{m_2} h^{r_2} = g^{m_1+m_2} h^{r_1+r_2} \tag{3.1}$$

which is equivalent to $Com(m_1 + m_2, r_1 + r_2)$.

Setup:	The parties agree on a cyclic group G of order q where the discrete logarithm problem is hard as well as two generators g, h of G.
Commit:	The committer, given a message $m \in G$, chooses a uniformly random $r \in G$. Then they compute $Com(m, r) = g^m h^r$.
Verify:	The verifier receives m, r from the committer and recomputes the commitment with the same public parameters. The commitment is accepted if the recomputed commitment matches the given one.

Figure 3.1: Description of the Pedersen Commitment

Malleability Pedersen Commitments are malleable. Multiplying a commitment $Com(m, r)$ by g gives $Com(m + 1, r)$:

$$\begin{aligned} Com(m, r) \cdot g &= g^m h^r g \\ &= g^{m+1} h^r \\ &= Com(m + 1, r). \end{aligned} \tag{3.2}$$

Hiding Pedersen Commitments are perfectly hiding. For any $Com(m, r)$ and any potential committed message m there exists a value for r that creates the same commitment. Therefore, all m are equally likely. Since g and h are generators, g^m and h^r are also generators. Since $g^{m'}$ is a generator for G for any m', there exists a value x such that $g^{m'} x = Com(m, r)$ and because h is a generator of G, there also exists a randomness r' such that $h^{r'} = x$.

Binding If a Committer were able to find m', r' such that $Com(m, r) = Com(m', r')$ then they would be able to solve the discrete logarithm problem $log_g(h)$ as follows:

$$\begin{aligned} Com(m, r) &= Com(m', r') \\ g^m h^r &= g^{m'} h^{r'} \\ g^{m-m'} &= h^{r'-r} \\ m - m' &= log_g(h^{r'-r}) \\ m - m' &= log_g(h)(r' - r) \\ \frac{m - m'}{r' - r} &= log_g(h) \end{aligned} \tag{3.3}$$

Therefore finding such m', r' is at least as hard as the discrete logarithm problem in G and therefore Pedersen Commitments are computationally binding.

Equivocability The committer can reveal a commitment to be any message m if a value x is known such that $g^x = h$. This follows directly from the observation that

$$\begin{aligned} Com(m,r) &= g^m h^r \\ &= g^m (g^x)^r \\ &= g^{m+xr}. \end{aligned} \tag{3.4}$$

Any Commitment $Com(m,r)$ can then be opened up to reveal an arbitrary message m' using $r' = m + xr - m'$ as the random vector. This is true because

$$\begin{aligned} Com(m',r') &= g^{m'+xr'} \\ &= g^{m'+x(m+xr-m')} \\ &= g^{m'+m+xr-m'} \\ &= g^{m+xr} \\ &= Com(m,r). \end{aligned} \tag{3.5}$$

This makes Pedersen commitments equivocable.

Extractability Since the committer, given the right information, is able to open any commitment to any value (Equation 3.5), it is impossible for an attacker to extract any specific committed value. This makes unextractable under our assumptions.

3.1.2 ElGamal

ElGamal Commitments are based on the popular ElGamal encryption scheme first described by Taher Elgamal in 1985 [24]. Committing to a value is equivalent to encrypting it under the ElGamal encryption. The value can later be extracted using a trapdoor, which is the encryption key used. The scheme is explained in Figure 3.2.

The scheme is included in this work to demonstrate the possibility of using homomorphic encryption to construct homomorphic commitment schemes.

Setup:	The parties agree on a cyclic group G of order q where the discrete logarithm problem is hard as well as two generators g, h of G.
Commit:	The committer, given a message $m \in G$, chooses a uniformly random $r \in \mathbb{Z}_q$. Then they compute $Com(m, r) = (g^r, mh^r)$.
Verify:	The verifier receives m, r from the committer and recomputes the commitment with the same public parameters. The commitment is accepted if the recomputed commitment matches the given one.

Figure 3.2: Description of the ElGamal Commitment

Correctness As the computation of the commitment is fully deterministic, the correctness of ElGamal commitments follows analogous that of Pedersen commitments described in Figure 3.1.1.

Randomness Homomorphy Let $(g^{r_1}, m_1 h^{r_1})$, $(g^{r_2}, m_2 h^{r_2})$ be two messages, using the same g, h but otherwise independent of each other.

$$\begin{aligned} Com(m_1, r_1) \cdot Com(m_2, r_2) &= (g^{r_1}, m_1 h^{r_1}) \cdot (g^{r_2}, m_2 h^{r_2}) \\ &= (g^{r_1} \cdot g^{r_2}, (m_1 h^{r_1}) \cdot (m_2 h^{r_2})) \\ &= (g^{(r_1+r_2)}, (m_1 \cdot m_2) h^{(r_1+r_2)}) \end{aligned} \tag{3.6}$$

The last step in Equation 3.6 is equivalent to a commitment on a message $m_1 \cdot m_2$ using the randomness $r_1 + r_2$. Therefore, the ElGamal commitment is homomorphic.

Binding The first half of the commitment, g^r, is an injective function on $r \in \mathbb{Z}_q$. Therefore, a second opening would have to use the same randomness, otherwise the first term would be different. But the second half, mh^r uniquely determines m if h^r is fixed. Therefore, no second opening can exist which makes the commitment perfectly binding.

Hiding If an adversary $\mathcal{A}$ was able to efficiently compute m given only $Com(m, r)$ then this implies the existence of an attacker $\mathcal{B}$ who can efficiently solve the computational Diffie-Hellman problem defined in Section 2.2.2. When given an instance of the CDH problem (g, g^a, g^b) $\mathcal{B}$ chooses a random $z \in \mathbb{Z}_q$ and gives $Com(m, r) = (g^a, z)$ to $\mathcal{A}$ with the public parameters $g = g$, $h = g^b$. When $\mathcal{A}$ returns an $m \in \mathbb{Z}_q$, $\mathcal{B}$ simply computes $z/m = h^a = g^{ab}$

Therefore, the hiding property of the ElGamal commitment is at least as hard as the CDH problem, making it computationally hiding.

Malleability The ElGamal commitment scheme is easily malleable, as multiplying the second element of the commitment with any value directly influences the committed value: For a commitment (g^y, mh^y), multiplying with any m' yields $(g^y, mh^y m')$, a valid commitment for the message $m \cdot m'$.

Equivocability The commitment is perfectly binding and therefore cannot be equivocable under our assumptions.

Extractability Because g, h are generators of G, they have the relation $h = g^x$ for some $x \in \mathbb{Z}_q$. This makes x a trapdoor. Knowing x allows opening up any commitment. To open a commitment (g^y, mh^y), divide the second value by the first which gives mg^x. Since x and g are known, this can trivially be solved for m. This is equivalent to knowledge of the private key in the ElGamal encryption scheme.

3.1.3 Groth

Groth demonstrates a homomorphic commitment scheme based on bilinear maps common in pairing cryptography. Like the ElGamal scheme it allows committing to group elements instead of exponents [30]. The scheme itself is explained in Figure 3.3.

Definition 3.1.1 (Bilinear Groups)**.** Bilinear Groups are two groups G, G_t with a bilinear map $e : G \times G \longrightarrow G_T$. The operation e satisfies a number of conditions [42].

- It is linear in both arguments
- It is non-degenerate: $e(x, y) = 0 \Leftrightarrow x = 0 \vee y = 0$
- It is efficiently computable

The security of their scheme depends on the hardness of the *simultaneous triple pairing assumption*, which states that given two random tuples (g_r, g_s, g_t) and (h_r, h_s, h_t) and a bilinear map e it is infeasible to find r, s, t such that

$$e(g_r, r)e(g_s, s)e(g_t, t) = 1 \text{ and } e(h_r, r)e(h_s, s)e(h_t, t) = 1$$

Groth shows that this assumption is implied by the decisional linear assumption [8].

Setup:	The parties agree on two groups G, G_T of order q and a bilinear map e from G to G_T, a number of messages n and $2n + 4$ random elements $(g_s, g_r, g_1 \dots g_n, h_s, h_r, h_1 \dots h_n) \in G^{2n+4}$.
Commit:	The committer, given n messages $m_1 \dots m_n$ chooses two random values $r, s \in G$. The commitment is given by (c, d) with $c = e(g_s, s)e(g_r, r) \prod_{i=1}^{n} e(g_i, m_i)$ and $d = e(h_s, s)e(h_r, r) \prod_{i=1}^{n} e(h_i, m_i)$.
Verify:	The verifier receives $m_1 \dots m_n$, (r, s) from the committer and recomputes the commitment with the same public parameters. The commitment is accepted if the recomputed commitment matches the given one.

Figure 3.3: Description of the Groth Commitment

Correctness Commitments are verified by recomputing and comparing them. Since the commit algorithm is deterministic, the proof of correctness is analogous to that of Pedersen commitments in Figure 3.1.1.

Hiding As shown in the section on equivocability, any commitment can be opened to any message. This implies that the commitment scheme is perfectly hiding, as an attacker cannot possibly rule out any possibilities.

Randomness Homomorphy For any two groups G, G_T both halves of the commitment (c/d) are homomorphic. The following proves this for c, the proof for d follows in the same way. Let c, c' be the first halves of two commitments $Com(m_1 \dots m_n, r)$ and $Com(m'_1 \dots m'_2, r')$.

$$\begin{aligned}&(e(g_s, s)e(g_r, r)\prod_{i=1}^{n} e(g_i, m_i))(e(g_s, s')e(g_r, r')\prod_{i=1}^{n} e(g_i, m'_i))\\ &= e(g_s, ss')e(g_r, rr')\prod_{i=1}^{n} e(g_i, m_i m'_i)\end{aligned} \tag{3.7}$$

This is equivalent to the first half of a commitment $Com(mm', (ss', rr'))$. Therefore, the Groth commitment is randomness-homomorphic.

Equivocability To set up a trapdoor, one chooses $x_1, y_1 \dots x_n, y_n, x_s, y_s, x_r, y_r$ at random such that $x_s y_r \neq x_r y_s$ [30]. Then, the key is defined as

$$g_1 = g_1^x, h_1 = g_1^y \dots g_n = g^{x_n}, h_n = g^{y_n}, g_s = g_s^x, h_s = g_s^y, g_r = g_r^x, h_r = g_r^y.$$

Choosing $x_s y_r - x_r y_s$ enables computing

$$\begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix} = \begin{pmatrix}x_r & x_s\\ y_r & y_s\end{pmatrix}^{-1}. \tag{3.8}$$

Then, the trapdoor key is $(x_1, y_1 \dots x_n, y_n, x_s, y_s, x_r, y_r, \alpha, \beta, \gamma, \delta)$ [30]. Notice that computing the trapdoor key given only the public key is computationally hard due to the discrete logarithm problem.

Then, to create an equivocable commitment, the committer chooses a random $(s, r) \in G^2$ and computes a commitment $(c, d) \in G_T^2$ as follows [30]:

$$c = e(g_r, r)e(g_s, s) \text{ and } d = e(h_r, r)e(h_s, s).$$

Such a commitment can then be opened to reveal any message $(m_1, \dots m_n)$ by computing

$$a = r^{x_r} s^{x_s} \prod_{i=1}^{n} m_i^{-x_i} \text{ and } b = r^{y_r} s^{y_s} \prod_{i=1}^{n} m_i^{-y_i}$$

Then, the opening (r', s') where $r' = a^{\alpha} b^{\beta}$ and $s' = a^{\gamma} b^{\delta}$ is a valid opening for $(m_1, \dots m_n)$[30].

This is the case since

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \times \begin{pmatrix} x_r & x_s \\ y_r & y_s \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{3.9}$$

Due to the properties of a bilinear map defined in Definition 3.1.1 we know that

$$\begin{aligned} e(g_r, r')e(g_s, s') &= e(g^{x_r}, a^{\alpha} b^{\beta}) e(g^{x_s}, a^{\gamma} b^{\delta}) \\ &= e(g^{x_r}, a^{\alpha}) e(g^{x_r}, b^{\beta}) e(g^{x_s}, a^{\gamma}) e(g^{x_s}, b^{\delta}) \\ &= e(g, a)^{\alpha x_r} e(g_b)^{\beta x_r} e(g, a)^{\gamma x_s} e(g, b)^{\delta x_s} \\ &= e(g, a)^{\alpha x_r + \gamma x_s} e(g, b)^{\beta x_r + \delta x_s} \\ &= e(g, a). \end{aligned} \tag{3.10}$$

In the same way, it follows that $e(h_r, r')e(h_s, s') = e(h, b)$.

To verify the trapdoor commitment above, the verifier recomputes the commitment. Substituting in $a = r^{x_r} s^{x_s} \prod_{i=1}^{n} m_i^{-x_i}$ and $b = r^{y_r} s^{y_s} \prod_{i=1}^{n} m_i^{-y_i}$ gives

$$\begin{aligned} e(g_s, s')e(g_r, r') \prod_{i=1}^{n} e(g_i, m_i) &= e(g, a) \prod_{i=1}^{n} e(g_i, m_i) \\ &= e(g, r^{x_r} s^{x_s} \prod_{i=1}^{n} m_i^{-x_i}) \prod_{i=1}^{n} e(g^{x_i}, m_i) \\ &= e(g, r^{x_r} s^{x_s}) \prod_{i=1}^{n} e(g, m_i^{x_i - x_i}) \\ &= e(g, r^{x_r} s^{x_s}) \\ &= e(g^{x_r}, r) e(g^{x_s}, s) \\ &= e(g_r, r) e(g_s, s) \\ &= c. \end{aligned} \tag{3.11}$$

In the same way it can be proven that $e(h_s, s')e(h_r, r') \prod_{i=1}^{n} e(h_i, m_i) = d$, making the commitment valid [30].

Binding Assume there was an adversary $\mathcal{A}$ that, given a commitment key $(g_1, h_1 \dots g_n, h_n)$, is able to compute two different messages $(m_1 \dots m_n)$, $(m'_1 \dots m'_n)$ that compute to the same commitment with a non-negligible probability ϵ. The existence of $\mathcal{A}$ implies the existence of an attacker $\mathcal{B}$ that solves the simultaneous triple pairing problem defined in Section 3.1.3 with non-negligible probability.

Let $(g_r, g_s, g_t, h_r, h_s, h_t)$ be a challenge for the simultaneous triple pairing problem. If $e(g_s, h_r) = e(g_r, h_s)$ then a solution for the challenge is given by $(g_s, g_r^{-1}, 1)$ [30], since

$$e(g_r, g_s)e(g_s, g_r^{-1})e(g_t, 1) = 1 \text{ and } e(h_r, g_s)e(h_s, g_r^{-1})e(h_t, 1) = 1.$$

This solution is non-trivial unless $g_s = g_r^{-1} = 1$. In that case, $(h_s, h_r^{-1}, 1)$ is another solution. If that one is trivial as well, then $g_r = g_s = h_r = h_s$ and $(g, g, 1)$ is a non-trivial solution [30].

If $e(g_s, h_r) \neq e(g_r, h_s)$ then the discrete logarithms, defined the same way as in Section 3.1.3, satisfy $x_s y_r \neq x_r y_s$. This means that (x_s, y_r) and (x_r, y_s) are linearly independent in $\mathbb{Z}_q^2$ [30]. Choose $\rho_1, \sigma_1, \tau_1 \ldots \rho_n, \sigma_n \tau_n \in \mathbb{Z}_q$ and determine $g_1, h_1 \ldots g_n, h_n$ by computing $g_i = g_r^{\rho_i} g_s^{\sigma_i} g_t^{\tau_i}$ and $h_i = h_r^{\rho_i} h_s^{\sigma_i} h_t^{\tau_i}$ [30].

Since (x_s, y_r) and (x_r, y_s) are linearly independent, these g_i, h_i are uniformly distributed among $\mathbb{Z}_q$. This means that commitment keys defined as $(g_1, h_1 \ldots g_n, h_n)$ have the same distribution as commitment keys generated by the setup algorithm [30].

With a non-negligible probability $\mathcal{A}$ is able to compute two different $(m_1 \ldots m_n)$, $(m'_1 \ldots m'_n)$ and randomness values (r, s), (r', s') that create the same commitments. Then, define $\mu_i = m'_i m_i^{-1}$, $r'' = r' r^{-1}$ and $s'' = s' s^{-1}$. Because of the homomorphic property proven in Section 3.1.3, $Com(\mu_1 \ldots \mu_n, (r'', s''))$ is $(1, 1)$ [30]. Then

$$e(g_r, r'')e(g_s, s'') \prod_{i=1}^{n} e(g_i, \mu_i) = e(g_r, r'' \prod_{i=1}^{n} \mu_i^{\rho_i}) e(g_s, s'' \prod_{i=1}^{n} \mu_i^{\sigma_i}) e(g_t, \prod_{i=1}^{n} \mu_i^{\tau_i}) = 1$$
$$e(h_r, r'')e(h_s, s'') \prod_{i=1}^{n} e(h_i, \mu_i) = e(h_r, r'' \prod_{i=1}^{n} \mu_i^{\rho_i}) e(h_s, s'' \prod_{i=1}^{n} \mu_i^{\sigma_i}) e(h_t, \prod_{i=1}^{n} \mu_i^{\tau_i}) = 1$$

This means that $(r'' \prod_{i=1}^{n} g_r^{\rho_i}, s'' \prod_{i=1}^{n} g_s^{\sigma_i}, \prod_{i=1}^{n} g_t^{\tau_i})$ is a solution to the triple pairing challenge. Since $(m_1 \ldots m_n) \neq (m'_1 \ldots m'_n)$ there is at least one $\mu_i \neq 1$. And because τ_i is uniformly random, the probability that $\prod_{i=1}^{n} g_t^{\tau_i} = 1$ is $\frac{1}{q}$. Therefore the probability that the solution above is trivial is at most $\frac{1}{q}$.

To summarize, if $e(g_s, h_r) = e(g_r, h_s)$ then the triple pairing challenge can be solved with probability 1. If $e(g_s, h_r) \neq e(g_r, h_s)$ then $\mathcal{B}$ has at least the (non-negligible) probability $\epsilon - \frac{1}{q}$ to pass the challenge. Therefore, the commitment must be computationally binding [30].

Extractability The Groth commitment scheme cannot be extractable under our assumptions, since it is perfectly hiding.

3.2 Lattice Schemes

The following commitment schemes no longer depend on the hardness of computing the discrete logarithm, instead, they use well known lattice problems like Short Integer Solution or Learning with Errors. These problems are assumed to be hard to solve even on a quantum computer.

3.2.1 Ajtai

The Ajtai Commitment is implicitly given in the seminal work of Ajtai[2] was later explicitly defined by Lyubashevsky et al. [34]. It is a fairly direct translation from the Short Integer Solution problem to the realm of commitments. The scheme allows committing to an arbitrarily sized vector of values at once, with the nice property that the size of the commitment is independent of the number of committed values. A higher number of values does weak the security, since the SIS problem becomes easier when more column vectors are available. Computing the commitment is conceptually very simple and easy to implement, requiring only a few basic matrix-vector operations. Figure 3.4 defines the **Setup**, **Commit** and **Verify** algorithms for the Ajtai Commitment.

The scheme is included in this thesis due to its simplicity and ease of implementation.

Setup:	The parties agree on two matrices $A_1 \in \mathbb{Z}_q^{n \times j}$ and $A_2 \in \mathbb{Z}_q^{n \times k}$. They also choose a real number $\beta < q$.
Commit:	The committer, given a message $m \in \mathbb{Z}_q^j$ with $\|m\| < \beta$, chooses a uniformly random $r \in \mathbb{Z}_q^k$ with $\|r\| < \beta$. Then they compute $Com(m, r) = A_1 m + A_2 r$.
Verify:	The verifier receives m, r from the committer and recomputes the commitment. The commitment is accepted if the recomputed commitment matches the given one and both $\|m\|, \|r\| < \beta$.

Figure 3.4: Description of the Ajtai Commitment

Correctness Commitments are verified by recomputing and comparing them, as defined in the **Verify** algorithm in Figure 3.4. Since the **Commit** algorithm is deterministic, the proof of correctness is analogous to that of Pedersen commitments in Section 3.1.1.

Randomness Homomorphy Adding two commitments $Com(m_1, r_1)$ and $Com(m_2, r_2)$ together creates a valid commitment $Com(m_1 + m_2, r_1 + r_2)$:

$$\begin{aligned} Com(m_1, r_1) + Com(m_2, r_2) &= (A_1 m_1 + A_2 r_1) + (A_1 m_2 + A_2 r_2) \\ &= A_1(m_1 + m_2) + A_2(r_1 + r_2) \\ &= Com(m_1 + m_2, r_1 + r_2). \end{aligned} \tag{3.12}$$

The homomorphy of the Ajtai commitment is limited. It is not guaranteed that $\|m_1 + m_2\|$, $\|r_1 + r_2\| < \beta$, meaning that it is possible to create an invalid commitment by combining two valid ones. This can be avoided by choosing messages and randomness values with norm $< \frac{\beta}{2}$ in advance.

Binding Assume a committer was able to find short values m_1, m_2, r_1, r_2 where $m_1 \neq m_2$ and $Com(m_1, r_1) = Com(m_2, r_2)$.

Then $A_1 m_1 + A_2 r_1 = A_1 m_2 + A_2 r_2$ and therefore $A_1(m_1 - m_2) + A_2(r_1 - r_2) = 0$. This can be rewritten as

$$\begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} (m_1 - m_1) \\ (r_1 - r_2) \end{bmatrix} = 0. \tag{3.13}$$

Since $\|m_1\|$, $\|m_2\|$, $\|r_1\|$ and $\|r_2\|$ must all be smaller than β, $\|m_1 - m_2\|$ and $\|r_1 - r_2\|$ must be smaller than 2β.

The above is equivalent to an instance of SIS with $\beta' = 2\beta$. Since solving this instance is computationally hard, the commitment is computationally binding. This has a few implications, like SIS, the binding property can only become stronger for higher n and weaker for higher j, k.

Hiding $(A_2, A_2 r)$ is indistinguishable from random noise according to the leftover hash lemma [32]. Adding $A_1 m$ where $\|m\|$ is small does not change the uniform distribution [39].

Malleability For any vector v, adding $A_1 v$ to the commitment creates a valid commitment for $m + v$. The only constraint is that $|m + v|$ must be small (otherwise the commitment will be rejected). Since $|m|$ is known to be small, $|v|$ has to be small as well.

Equivocability The seminal work of Ajtai in 1996 [2] showed an efficient construction of a lattice trapdoor. Using this algorithm, it is possible to efficiently compute a matrix A that is statistically close to uniformly random and a corresponding short vector t such that $At = 0$.

To set up a trapdoor for the Ajtai commitment, one needs to compute two of these SIS trapdoors for for $A_1 m_{12} = 0$ and $A_2 r_{12} = 0$. Decomposing $m_{12} = m_1 - m_2$ and $r_{12} = r_1 - r_2$ where m_1, m_2, r_1, r_2 are short gives two openings for the commitment with public parameters A_1, A_2.

Extractablity As the scheme is statistically hiding, it cannot be extractable under our assumptions.

3.2.2 Baum et al.

Baum et al.[6] demonstrate an additively homomorphic vector commitment scheme over polynomial rings. It is based on the ideas demonstrated in [19]. Notably, their scheme can be instantiated with different parameters which determine the security properties. Figure 3.5 gives a definition of the scheme itself.

The scheme is defined over two rings. For some power of two N, $R = \mathbb{Z}[X]/\langle X^N+1\rangle$ is used to compute element norms and $R_q = \mathbb{Z}_q[X]/\langle X^N + 1\rangle$ is used for most other computations. The choice of $X^N + 1$ is further explained in section 2.1 of [6].

The scheme is based on the hardness of two different problems:

Definition 3.2.1 ($SKS^2_{n,k,\beta}$)**.** The $SKS^2_{n,k,\beta}$ problem asks to find a (short) vector $y \in R_q^k$ with $|y| < \beta$ satisfying $\begin{bmatrix} I_n & A' \end{bmatrix} \cdot y = 0^n$ when given a random A'. This is equivalent to solving Module-SIS. The $SKS^2_{n,k,\beta}$ problem becomes easier for larger β [6].

Definition 3.2.2 ($DKS^\infty_{n,k,\beta}$)**.** The $DKS^\infty_{n,k,\beta}$ problem asks to distinguish the distribution $\begin{bmatrix} I_n & A' \end{bmatrix} \cdot y$ for $|y| < \beta$ from a uniformly random distribution when given $A' \in R_q^{n\times(k-n)}$. The $DKS^\infty_{n,k,\beta}$ becomes harder for larger β [6].

The parameter β that these problems are instantiated with determines the security of the hiding and binding properties of the commitment scheme. For large β, the hiding property becomes strong and the binding property is weakened, whereas for small β the hiding property is weaker and the commitment is strongly binding. This is illustrated in Figure 3.6. This configurability is the primary strength of the Baum et al. scheme and the reason it was chosen to be included in the thesis, as none of the other analyzed schemes can be configured.

The verification of the commitment schemes involves a polynomial f in addition to the message m and randomness r. If a committer simply wishes to reveal the contents of a commitment they can always choose $f = 1$. Baum et al. present an algorithm that allows zero-knowledge proofs of the committed message and might result in $f \neq 1$.

The randomness in the **Commit** algorithm from Figure 3.5 is sampled from S^k_β, which is equivalent to R^k except individual elements are bounded by β in the l_∞ norm.

Unlike other schemes presented, the Baum et al. scheme has additional capabilities for zero knowledge proofs. The scheme allows a committer to prove that they know valid opening information for a commitment, without revealing said opening information itself. It is also possible to prove certain relations between the messages of multiple published commitments in a zero-knowledge manner. Specifically, the committer can prove a linear relation between two commitments and that the messages of multiple commitments sum up to the message in another commitment.

Setup: Choose message length l. Then, choose $A_1 \in R_q^{n \times k}$, $A_2 \in R_q^{l \times k}$ as
$A_1 = \begin{bmatrix} I_n & A_1' \end{bmatrix}$ with $A_1' \leftarrow R_q^{n \times (k-n)}$
$A_2 = \begin{bmatrix} 0^{l \times n} & I_l & A_2' \end{bmatrix}$ with $A_2' \leftarrow R_q^{l \times (k-n-l)}$

Commit: To commit to a message $m \in R_q^l$ with randomness $r \in S_\beta^k$, compute
$$Com(m, r) = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} r + \begin{bmatrix} 0^n \\ m \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Verify: An opening for $Com(m, r)$ given m, r and a with polynomial f is valid if $f \cdot \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} r + f \cdot \begin{bmatrix} 0^n \\ m \end{bmatrix}$ and for all i, $||r_i||_2 < 4\sigma\sqrt{N}$.

Figure 3.5: Description of the Baum et al. Commitment

Correctness Commitments are verified by recomputing and comparing them. Since the **Commit** algorithm is deterministic, the proof of correctness is analogous to that of Pedersen commitments in Section 3.1.1.

Randomness Homomorphy Adding two commitments $Com(m_1, r_1)$ and $Com(m_2, r_2)$ creates a valid commitment $Com(m_1 + m_2, r_1 + r_2)$.

$$\begin{aligned} Com(m, r) + Com(m', r') &= \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} r_1 + \begin{bmatrix} 0 \\ m_1 \end{bmatrix} + \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} r_2 + \begin{bmatrix} 0 \\ m_2 \end{bmatrix} \\ &= \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} (r_1 + r_2) + \begin{bmatrix} 0 \\ m_1 + m_2 \end{bmatrix} \\ &= Com(m_1 + m_2, r_1 + r_2) \end{aligned} \tag{3.14}$$

Therefore, the commitment scheme is randomness-homomorphic.

Hiding If an attacker was able to distinguish commitments to two different messages x_0, $x_1 \in R_q^l$ with a non-negligible advantage ϵ then it would be possible to solve the $DKS_{n+l,k,\beta}^{\infty}$ problem.

For an instance $B = \begin{bmatrix} I_{n+l} & B' \end{bmatrix}, t$ of the $DKS_{n+l,k,\beta}^{\infty}$ problem, $\mathcal{A}'$ chooses a random $R \in R_q^{n \times l}$ and defines the public parameters of the scheme as

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} I_n & R \\ 0_{n+l} & I_l \end{bmatrix} \cdot B. \tag{3.15}$$

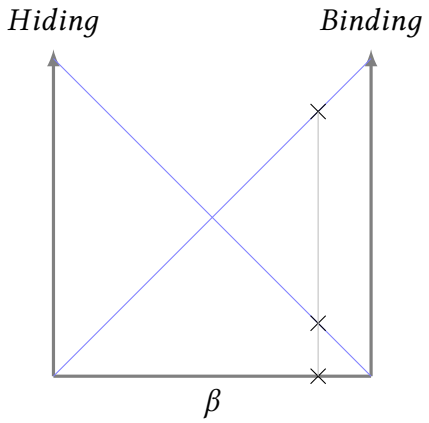

Figure 3.6: Illustration of how β influences securty.
Adapted from Baum et al. [6], Fig. 1

The matrices A_1, A_2 are distributed identically to the actual public parameters of the commitment scheme, since

$$\begin{aligned}\begin{bmatrix}A_1\\A_2\end{bmatrix} &= \begin{bmatrix}I_n & R\\0^{n\times l} & I_l\end{bmatrix} B\\ &= \begin{bmatrix}I_n & R\\0^{n\times l} & I_l\end{bmatrix}\begin{bmatrix}I_n & 0^{n\times l} & B'_1\\0^{l\times n} & I_l & B'_2\end{bmatrix}\\ &= \begin{bmatrix}I_n & R & B'_1 + RB'_2\\0^{l\times n} & I_l & B'_2\end{bmatrix}.\end{aligned} \tag{3.16}$$

Since R, B'_1, B'_2 are uniformly and independently distributed, this is indistinguishable from the actual key generation algorithm.

Then, $\mathcal{A}'$ chooses a random $b \leftarrow \{0,1\}$ and computes the commitment

$$\begin{bmatrix}c_1\\c_2\end{bmatrix} = \begin{bmatrix}A_1\\A_2\end{bmatrix} r + \begin{bmatrix}0^n\\x_b\end{bmatrix} \tag{3.17}$$

This information is then given to $\mathcal{A}$, which will output $b' \in \{0,1\}$. If $b = b'$ then $\mathcal{A}'$ outputs 1, otherwise 0.

There are two possibilities: Either t is truly random, then $\begin{bmatrix}c_1\\c_2\end{bmatrix}$ are independent of x_b and $\mathcal{A}$ has a probability of $\frac{1}{2}$ of being correct. Or $t = Br$ for some r, then $\mathcal{A}$ by definition has at least a probability of $\frac{1}{2} + \epsilon$ of being correct. Thus, $\mathcal{A}'$ has an advantage of at least ϵ.

Binding If an attacker $\mathcal{A}$ were able to find openings (m, r, f) and (m', r', f') with $m \neq m'$ for a commitment $\begin{bmatrix}c_1\\c_2\end{bmatrix}$ with a non-negligible advantage ϵ, then an attacker $\mathcal{A}'$ exists who can solve the $SKS^2_{n,k,\beta}$ with advantage ϵ.

Given an instance $A_1 = \begin{bmatrix} I_n & A_1' \end{bmatrix}$ of $SKS^2_{n,k,\beta}$, the attacker computes A_2 in the same manner as during public key generation. Then, with an advantage ϵ, $\mathcal{A}$ can find (x, r, f) and (x', r', f') with $m \neq m'$ for the public key A_1, A_2. That means that

$$\begin{aligned} f \cdot \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} &= \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} r + f \cdot \begin{bmatrix} 0^n \\ m \end{bmatrix} \\ f' \cdot \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} &= \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} r' + f' \cdot \begin{bmatrix} 0^n \\ m' \end{bmatrix}. \end{aligned} \tag{3.18}$$

Multiplying the first equation by f', the second by f and subtracting the two yields

$$\begin{aligned} A_1(f' \cdot r_1 - f \cdot r_2) &= 0^n \\ A_2(f' \cdot r_1 - f \cdot r_2) + f \cdot f' \cdot (x - x') &= 0^l. \end{aligned} \tag{3.19}$$

Since $x \neq x'$, $f \cdot f' \cdot (x - x')$ is known to be nonzero. Therefore, $(f' \cdot r_1 - f \cdot r_2)$ must also be nonzero. Because $f, f' \in C$, $||f||_2, ||f'||_2 < 2\sqrt{\kappa}$.

Since all the polynomials r_i are bounded by $4\sigma\sqrt{N}$ in the euclidian norm, we know that $||f \cdot r||_2, ||f \cdot r||_2 \leq 8\sigma\sqrt{\kappa N}$. Therefore, their difference is at most $16\sigma\sqrt{\kappa N}$ and $(f' \cdot r_1 - f \cdot r_2)$ is a solution to the $SKS^2_{n,k,16\sigma\sqrt{\kappa N}}$ instance.

3.3 Other Schemes

During literature research, a number of commitments were considered but not included in the final survey. We decided to limit ourselves to schemes that depend on the hardness of the discrete logarithm problem, the Short Integer Solution problem or Learning with Errors. This was done to be able to draw a useful comparison between the complexity of these schemes. More information can be found in Chapter 4. Publications in the field often improve upon existing work, by extending them to add new functionality. Examples of this are given in Section 3.1.1. Such publications are mentioned in this survey, but not explicitly detailed to avoid duplication.

For completeness, these commitments are listed here.

- Abhishek Jain et al. [33] constructed a perfectly binding string commitment. However, it is based on the Learning Parity with Noise (LPN) problem and not homomorphic.
- Peiheng Zhang et al. [47] introduced a polynomial commitment scheme that is both homomorphic and based on the discrete logarithm problem. However, if the degree of the polynomial being committed to is one, their commitment reduces back to Pedersen commitments. Since polynomial commitments themselves are not the focus of this work, their commitment was not included.
- Tore Kasper Frederiksen et al. [27] developed a very efficient, additively homomorphic commitment scheme that is secure in the universal-composability framework of shown by Canetti [13]. An efficient implementation was later developed by Rindal and Trifiletti [41]. Their scheme is not included here since it is based on oblivious transfer functions and error correcting codes, not the discrete logarithm problem or lattice problems.

4 Direct Comparison between the Schemes

While previous chapters introduced individual commitment schemes we also want to give a more concise comparison between them.

4.1 Security Properties

Table 4.1 summarizes the findings from the sections on individual commitments in chapter 3 and 4. The `Performance` column is meant to give a rough comparison of the relative performance to each other, more precise benchmarks are shown in Figure 4.1.

The commitment of Baum et al. is an exception as it cannot clearly be categorized regarding the strength of its hiding and binding properties. Since the scheme is configurable, either one of the two properties can be statistical and the other one computational, or both can be computational.

Name	Problem	Hiding	Binding	Equivocable	Extractable	Performance
Pedersen	dlog	Perfect	Computational	✓	✗	Good
ElGamal	dlog	Computational	Perfect	✗	✓	Good
Groth	dlog	Perfect	Computational	✓	✗	Bad
Baum et al.	SIS/LWE	Configurable		✗	✗	Okay
Ajtai	SIS	Statistical	Computational	✓	✗	Okay

Table 4.1: Security Properties of the different schemes

4.2 Commitment Size

Another aspect when evaluating the suitability of a commitment for a particular purpose is the size of the commitment value. In resource constrained environments, large commitments can be impractical. Both the Pedersen and the ElGamal commitments are very small, consisting of only a single group element. The Groth commitment (c, d) contains two group elements and is therefore twice as large. All the lattice schemes have very large commitments that depend mostly on the dimension of the chosen lattice.

Both the Groth and the two lattice commitments are vector commitments, meaning that a committer can commit to multiple values at once. The size of the Groth and Ajtai

commitments are independent of the number of committed values, though committing to more values at once does weaken the security. The commitment size of the Baum et al. scheme linearly depends on the number of commitment values.

4.3 Performance

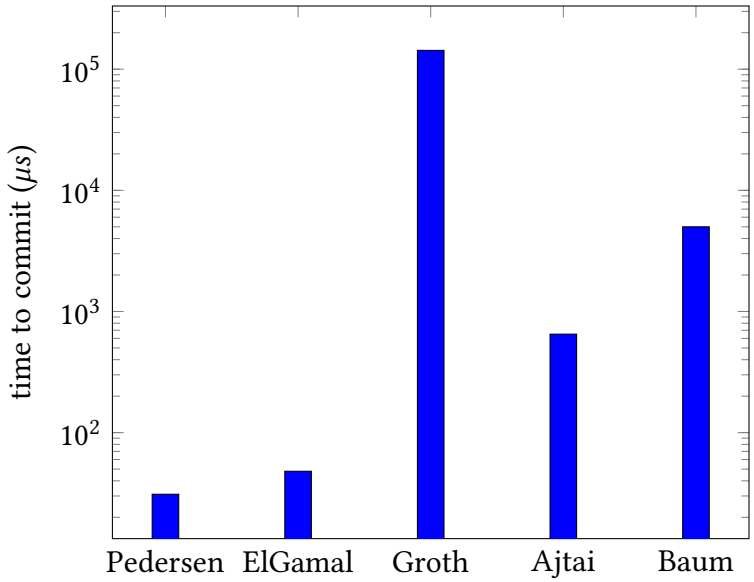

Figure 4.1: Performance comparison of the five commitment schemes

For this work, the first three schemes were implemented in the rust programming language [12] . We used the `curve25519-dalek` [17] library for elliptic curves and `bn` [11] for bilinear maps. The two lattice schemes were implemented in python [45] using `numpy` [31] instead. We do not expect this to distort the performance results too much, as the vast majority of operations are implemented by `numpy` using highly optimized C routines. All implementations are publicly available on GitHub[1]. The benchmarks were executed on a Linux PC using an AMD Ryzen 7 3700X and 32GB of DDR4 SDRAM.

Figure 4.1 shows that the simpler discrete-logarithm-based schemes outperform lattice-based approaches. Among them, the Pedersen commitment outperforms ElGamal commitments slightly. Perhaps surprisingly, the Groth commitment is significantly slower than even the more complex lattice schemes. This is due to the heavy usage of bilinear maps and might be an issue with the `bn` library itself.

[1] `https://github.com/Wuelle/randomness-homomorphic-commitments`

4.4 Finding a Suitable Commitment

We present an algorithm for finding a suitable commitment scheme for a given application, using the key differences between the schemes analyzed in Section 4.1 and Section 4.3. The process is illustrated in Figure 4.2 as a flow graph. One starts at the top and works their way downwards following the arrows by questions. When choosing a commitment scheme for a particular application, the first decision is whether or not the application requires post-quantum security. If not, then the added overhead of lattice-based schemes is likely not worth it and a simpler scheme like Pedersen is preferrable. Within these two groups the main difference between the schemes are the strength of the hiding and binding properties, with the exception of the Groth commitment the performance difference is minor. The Groth and Pedersen commitments show similar properties except for performance and the fact that Groth commitments allow committing to multiple messages at once. Among the lattice-based commitments, the Ajtai scheme the is simpler and slightly more performant one. Besides being conceptually simple, it is also easy to implement, requiring only simple matrix-vector multiplications. The scheme of Baum et al. is significantly more complicated than any other scheme presented in this work, but the configurability of the hiding and binding properties make it suitable for a wider variety of applications.

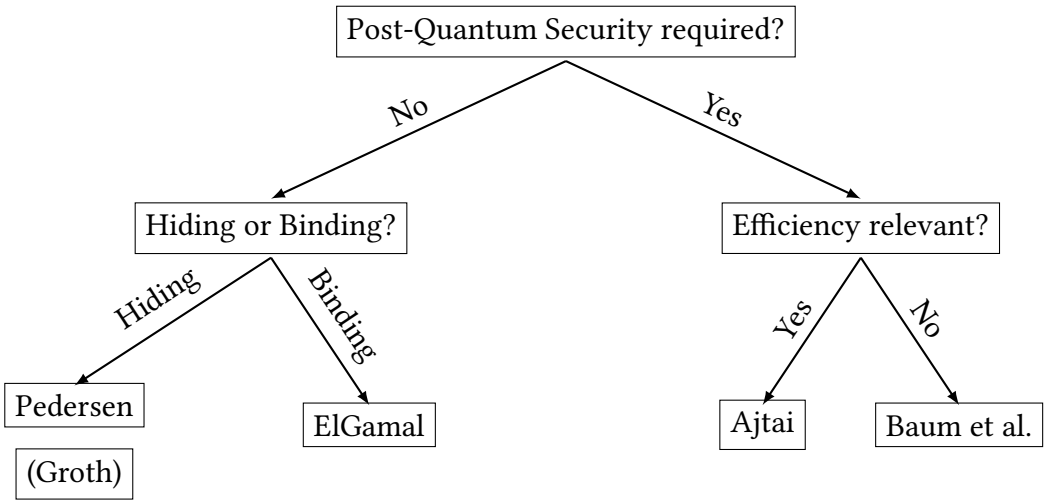

Figure 4.2: Flow graph for deciding on a commitment scheme

For generic applications like the rerandomization firewall explained in Section 1.1, the Ajtai and Pedersen schemes are the most promising, due to both their simplicity and high performance.

5 Conclusion

The goal of this thesis was to give an overview of the available options for randomness homomorphic commitments and to evaluate their individual advantages and disadvantages. We presented five different options for rerandomizable commitments and explained their construction in detail. For each of the commitments, we provided proofs of their extractability and equivocability. We also compared all the schemes with regards to security, performance and ease of implementation. We observed that there are significant differences in both performance and complexity between the commitments that developers need to be aware of.

While this work focused on rerandomizable commitments without further constraints, future work could additionally look into more specific variants of commitments. For example, in the context of zero knowledge proofs it is often necessary to prove relations between committed values without directly revealing them. Of the presented schemes only the one by Baum et al. explicitly allows for this.

Overall, there are significant differences between the known available options for randomness homomorphic commitment schemes. Individual applications might have specific requirements not considered here. However, we conclude that among the schemes considered in this thesis, the Pedersen commitment and the Ajtai commitments are the most promising options for general use cases. Both offer simplicity and ease of implementation, while also being the most performant within their problems domain.

Bibliography

[1] Michel Abdalla, Céline Chevalier, and David Pointcheval. "Smooth Projective Hashing for Conditionally Extractable Commitments". In: *Advances in Cryptology - CRYPTO 2009*. Ed. by Shai Halevi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 671–689. ISBN: 978-3-642-03356-8.

[2] Miklós Ajtai. "Generating hard instances of lattice problems (extended abstract)". In: *Electron. Colloquium Comput. Complex.* TR96 (1996). URL: https://api.semanticscholar.org/CorpusID:6864824.

[3] Prabhanjan Ananth et al. *Towards Accountability in CRS Generation*. Cryptology ePrint Archive, Paper 2021/1090. https://eprint.iacr.org/2021/1090. 2021. URL: https://eprint.iacr.org/2021/1090.

[4] Paula Arnold et al. *Protection Against Subversion Corruptions via Reverse Firewalls in the plain Universal Composability Framework*. Cryptology ePrint Archive, Paper 2023/1951. https://eprint.iacr.org/2023/1951. 2023. URL: https://eprint.iacr.org/2023/1951.

[5] Sarah Azouvi, Patrick McCorry, and Sarah Meiklejohn. *Betting on Blockchain Consensus with Fantomette*. 2018. arXiv: 1805.06786 [cs.CR].

[6] Carsten Baum et al. *More Efficient Commitments from Structured Lattice Assumptions*. Cryptology ePrint Archive, Paper 2016/997. https://eprint.iacr.org/2016/997. 2016. URL: https://eprint.iacr.org/2016/997.

[7] Donald Beaver. "Adaptive zero knowledge and computational equivocation (extended abstract)". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 629–638. ISBN: 0897917855. DOI: 10.1145/237814.238014. URL: https://doi.org/10.1145/237814.238014.

[8] Dan Boneh, Xavier Boyen, and Hovav Shacham. *Short Group Signatures*. Cryptology ePrint Archive, Paper 2004/174. https://eprint.iacr.org/2004/174. 2004. URL: https://eprint.iacr.org/2004/174.

[9] Dan Boneh, Aditi Partap, and Lior Rotem. *Post-Quantum Single Secret Leader Election (SSLE) From Publicly Re-randomizable Commitments*. Cryptology ePrint Archive, Paper 2023/1241. https://eprint.iacr.org/2023/1241. 2023. URL: https://eprint.iacr.org/2023/1241.

[10] Dan Boneh et al. "Single Secret Leader Election". In: *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies* (2020). URL: https://eprint.iacr.org/2020/025.pdf.

[11] Sean Bowe. *bn*. https://github.com/zcash-hackworks/bn. Accessed: 31-07-2024.

[12] William Bugden and Ayman Alahmar. *Rust: The Programming Language for Safety and Performance*. 2022. arXiv: 2206.05503 [cs.PL]. URL: https://arxiv.org/abs/2206.05503.

[13] R. Canetti. "Universally composable security: a new paradigm for cryptographic protocols". In: *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. 2001, pp. 136–145. DOI: 10.1109/SFCS.2001.959888.

[14] Suvradip Chakraborty et al. *Universally Composable Subversion-Resilient Cryptography*. Cryptology ePrint Archive, Paper 2022/244. https://eprint.iacr.org/2022/244. 2022. URL: https://eprint.iacr.org/2022/244.

[15] Mingjie Chen et al. *Malleable Commitments from Group Actions and Zero-Knowledge Proofs for Circuits based on Isogenies*. Cryptology ePrint Archive, Paper 2023/1710. https://eprint.iacr.org/2023/1710. 2023. URL: https://eprint.iacr.org/2023/1710.

[16] Giovanni Di Crescenzo. "Equivocable and Extractable Commitment Schemes". In: *Security in Communication Networks*. Ed. by Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 74–87. ISBN: 978-3-540-36413-9.

[17] *Curve25519 Dalek*. https://github.com/dalek-cryptography/curve25519-dalek. Accessed: 31-07-2024.

[18] Ivan Damgård. "Commitment Schemes and Zero-Knowledge Protocols". In: *Lectures on Data Security: Modern Cryptology in Theory and Practice*. Ed. by Ivan Bjerre Damgård. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 63–86. ISBN: 978-3-540-48969-6. DOI: 10.1007/3-540-48969-X_3. URL: https://doi.org/10.1007/3-540-48969-X_3.

[19] Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. "On the Existence of Statistically Hiding Bit Commitment Schemes and Fail-Stop Signatures". In: *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*. Vol. 773. Lecture Notes in Computer Science. Springer, 1993, pp. 250–265. DOI: 10.1007/3-540-48329-2_22.

[20] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. "Necessary and Sufficient Assumptions for Non-interactive Zero-Knowledge Proofs of Knowledge for All NP Relations". In: *Automata, Languages and Programming*. Ed. by Ugo Montanari, José D. P. Rolim, and Emo Welzl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 451–462. ISBN: 978-3-540-45022-1.

[21] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. "Non-interactive and non-malleable commitment". In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC '98. Dallas, Texas, USA: Association for Computing Machinery, 1998, pp. 141–150. ISBN: 0897919629. DOI: 10.1145/276698.276722. URL: https://doi.org/10.1145/276698.276722.

[22] W. Diffie and M. Hellman. “New directions in cryptography”. In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[23] T. Elgamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: *IEEE Transactions on Information Theory* 31.4 (1985), pp. 469–472. DOI: 10.1109/TIT.1985.1057074.

[24] T. Elgamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: *IEEE Transactions on Information Theory* 31.4 (1985), pp. 469–472. DOI: 10.1109/TIT.1985.1057074.

[25] Marc Fischlin. “Trapdoor commitment schemes and their applications”. doctoralthesis. Universitätsbibliothek Johann Christian Senckenberg, 2001, p. 159.

[26] Marc Fischlin and Roger Fischlin. “Efficient Non-Malleable Commitment Schemes”. In: *J. Cryptol.* 24.1 (Jan. 2011), pp. 203–244. ISSN: 0933-2790. DOI: 10.1007/s00145-009-9043-4. URL: https://doi.org/10.1007/s00145-009-9043-4.

[27] Tore Kasper Frederiksen et al. “On the Complexity of Additively Homomorphic UC Commitments”. In: *Theory of Cryptography*. Ed. by Eyal Kushilevitz and Tal Malkin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 542–565. ISBN: 978-3-662-49096-9.

[28] Jens Groth. *A Verifiable Secret Shuffle of Homomorphic Encryptions*. Cryptology ePrint Archive, Paper 2005/246. https://eprint.iacr.org/2005/246. 2005. URL: https://eprint.iacr.org/2005/246.

[29] Jens Groth. “Efficient Zero-Knowledge Arguments from Two-Tiered Homomorphic Commitments”. In: *ASIACRYPT*. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, pp. 431–448. DOI: 10.1007/978-3-642-25385-0_23. URL: https://www.iacr.org/archive/asiacrypt2011/70730421/70730421.pdf.

[30] Jens Groth. “Homomorphic Trapdoor Commitments to Group Elements”. In: *IACR Cryptol. ePrint Arch.* 2009 (2009), p. 7. URL: https://api.semanticscholar.org/CorpusID:13274797.

[31] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[32] Johan HÅstad et al. “A Pseudorandom Generator from any One-way Function”. In: *SIAM Journal on Computing* 28.4 (1999), pp. 1364–1396. DOI: 10.1137/S0097539793244708. eprint: https://doi.org/10.1137/S0097539793244708. URL: https://doi.org/10.1137/S0097539793244708.

[33] Abhishek Jain et al. “Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise”. In: *Advances in Cryptology – ASIACRYPT 2012*. Ed. by Xiaoyun Wang and Kazue Sako. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 663–680. ISBN: 978-3-642-34961-4.

[34] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. "Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General". In: *Advances in Cryptology – CRYPTO 2022*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Cham: Springer Nature Switzerland, 2022, pp. 71–101. ISBN: 978-3-031-15979-4.

[35] Chiara Marcolla et al. "Survey on fully homomorphic encryption, theory, and applications". In: *Proceedings of the IEEE* 110.10 (2022), pp. 1572–1609.

[36] Ihyun Nam. *A Survey of Multivariate Polynomial Commitment Schemes*. 2023. arXiv: 2306.11383 [cs.CR].

[37] Anca Nitulescu. "Sok: Vector commitments". In: *URL: https://www. di. ens. fr/~ nitulesc/files/vc-sok. pdf* (2023).

[38] Torben P. Pedersen. "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing". In: *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*. Vol. 576. Lecture Notes in Computer Science. Springer, 1991, pp. 129–140. DOI: 10.1007/3-540-46766-1_9.

[39] Chris Peikert. *A Decade of Lattice Cryptography*. Cryptology ePrint Archive, Paper 2015/939. https://eprint.iacr.org/2015/939. 2015. URL: https://eprint.iacr.org/2015/939.

[40] Oded Regev. *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*. 2024. arXiv: 2401.03703 [cs.CR].

[41] Peter Rindal and Roberto Trifiletti. *SplitCommit: Implementing and Analyzing Homomorphic UC Commitments*. Cryptology ePrint Archive, Paper 2017/407. https://eprint.iacr.org/2017/407. 2017. URL: https://eprint.iacr.org/2017/407.

[42] Lior Rotem. *Topics in Cryptography (CS 355)*. https://crypto.stanford.edu/cs355/23sp/CS_335__L07.pdf. 2023.

[43] Lior Rotem. *Topics in Cryptography (CS 355)*. https://crypto.stanford.edu/cs355/23sp/lec3.pdf. Accessed: 31-07-2024.

[44] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172. eprint: https://doi.org/10.1137/S0097539795293172. URL: https://doi.org/10.1137/S0097539795293172.

[45] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[46] Zhichao Yang et al. "On the Security of LWE Cryptosystem against Subversion Attacks". In: *The Computer Journal* 63.4 (Sept. 2019), pp. 495–507. ISSN: 0010-4620. DOI: 10.1093/comjnl/bxz084. eprint: https://academic.oup.com/comjnl/article-pdf/63/4/495/33153492/bxz084.pdf. URL: https://doi.org/10.1093/comjnl/bxz084.

[47] Peiheng Zhang et al. “Efficient Noninteractive Polynomial Commitment Scheme in the Discrete Logarithm Setting”. In: *IEEE Internet of Things Journal* 11.5 (2024), pp. 8078–8089. DOI: 10.1109/JIOT.2023.3319338.