Efficient Failure Discovery with Limited Authentication

Malte Borcherding University of Karlsruhe, Germany Institute of Computer Design and Fault Tolerance D-76128 Karlsruhe email: borcher@ira.uka.de

Abstract

Solutions for agreement problems in distributed systems can generally be divided into two classes: authenticated protocols and non-authenticated protocols. Authenticated protocols make use of authenticated messages, i.e., the messages can be signed in a way that a signed message can be assigned unambiguously to the signer. Few has been said about how to achieve this kind of authentication; in some settings this is impossible without a trusted dealer or other mechanisms outside the system.

In this paper, we introduce and investigate a weaker kind of authentication, local authentication. It can be achieved within a distributed system with an arbitrary number of arbitrary faults. We then show that Failure Discovery, a problem introduced by Hadzilacos and Halpern, can be solved with authenticated protocols even if only local authentication is available. Since authenticated protocols for this problem have linear message complexity, as opposed to quadratic complexity in the non-authenticated case, the effort of establishing local authentication once results in a substantial reduction of messages in subsequent failure-discovery protocols.

1 Introduction

A fundamental problem in distributed systems is to reach agreement despite the presence of failing nodes. This problem was introduced as *Byzantine Agreement* by Lamport, Shostak, and Pease ([4]) and has since been discussed to a considerable extent. Byzantine Agreement requires all correct nodes in the system to agree on the same value which must be the value of a distinguished *sender* if the sender is correct. One of the many variations of this problem is the *Failure Discovery* problem, introduced by Hadzilacos and Halpern [2]. Failure Discovery requires the nodes to reach Byzantine Agreement provided no node discovers that a failure occurred. We will focus on this problem in our paper.

Solutions for agreement problems can be divided into two classes: authenticated protocols and nonauthenticated protocols. In authenticated protocols, it is assumed that each receiver of a signed message can unambiguously identify the signer. This assumption generally allows better solutions, e.g. with regard to the maximum number of tolerated faulty nodes or the amount of data exchanged. Few has been said in the past about how authentication can be established in a fault-prone system.

To reach the required common knowledge about how to identify the signatures of the respective nodes (known as *key distribution*), one can either use nonauthenticated agreement protocols (which may not work because of too many faulty nodes) or assume some reliable key server or key server group (which contradicts the underlying model of computation).

In this paper we introduce an incomplete authentication technique, called *local authentication* which can be established in a system with an arbitrary number of arbitrarily faulty nodes. Roughly speaking, each node distributes the verification information for its signature by itself; it is not necessary that all nodes reach agreement on this information.

We then show that solutions for the problem of Failure Discovery which use authentication still work when merely local authentication is available. That means that one can use authenticated protocols in an non-authenticated environment after establishing local authentication, thus reducing the overall message complexity substantially.

2 Model of computation

In this section we describe the model of computation which has become the standard model for agreement protocols. Our world consists of a fully interconnected network with n nodes (processors) and $n \cdot (n-1)/2$ bidirectional communication links. The network has the following properties:

- (N1) Messages are transmitted reliably in bounded time.
- (N2) A receiver of a message can identify its immediate sender.

The nodes communicate in successive rounds. In each round a node may send messages to other nodes and receives all messages sent to it in the current round. A sequence of rounds in a protocol is called a run. A view of a node in round i of run r is the sequence of sets of messages it has received in each round of the run r up to round i. The actions a node takes in the next round depend solely on its current view. A run is called failure-free if no node deviates from the given protocol. If a node's view of a run differs from its views of all failure-free runs, it discovers a failure.

In Proceedings of the 15th International Conference on Distributed Computing Systems, pp. 78-82, IEEE 1995.

We make no assumptions about the *type of failures* that occur. If a node is faulty, it may behave in an arbitrary manner. This type of behaviour is usually referred to as *byzantine fault*.

Furthermore, we assume the existence of a signature scheme with the following properties:

- (S1) A node can produce a signed message $\{m\}_S$ if and only if it knows the secret key S and the message m.
- (S2) For each secret key S_i there exists a (public) test predicate T_i with the property $T_i(\{m\}_S) =$ true $\Leftrightarrow S = S_i$.
- (S3) The secret key S_i cannot be extracted from a signed message $\{m\}_{S_i}$ and/or the test predicate T_i .

Examples for signature schemes which fulfill these properties with a sufficiently high probability are DSA and RSA [5, 6].

Note that we do *not* make any assumptions about the authenticity of the distribution of the test predicates amongst the nodes. This point is different from the usual assumptions when signature schemes are involved.

3 Local authentication

In this section we introduce a new mechanism for optimizing failure discovery: local authentication. This mechanism should fill the gap between nonauthenticated protocols and authenticated protocols.

Authenticated protocols require public keys to be distributed authentically amongst the nodes, i.e., the nodes have consistent information on how to check signed messages. This state can be reached either by using an agreement protocol for each public key or by relying on some kind of trusted dealer (or group of dealers) which never fails. The first method may not be feasible because of an insufficient number of correct nodes, the second introduces problems by the requirement of certain nodes to be of higher reliability.

When using local authentication, each node distributes its public key amongst the other nodes by itself. This leads to a limited kind of authentication: While each node has the same public key for a correct node, the public keys for faulty nodes may differ. However, we give a distribution protocol which guarantees that a node can distribute only public keys for which it has the appropriate private key. That means that no faulty node can claim a public key of a correct node for itself. We will show that protocols for Failure Discovery which are designed for global authentication can safely be used with local authentication. With this approach, we can make use of the advantages of authenticated protocols without the need to assume global authentication.

3.1 The key distribution protocol

Fig. 1 shows the distribution protocol which establishes locally authentic public keys. First, each node P_i selects a pair of keys S_i and T_i . One (the secret key S_i) is used to sign messages, the other one (the public key T_i) is used to verify the signatures. For notational reasons, we suppose that the public key is cast into a *test predicate* which checks whether a message was signed with the corresponding secret key. Then each node sends its test predicate to all the other nodes.

On reception of a test predicate, a node starts a challenge-response protocol to see whether the sender actually has access to the appropriate secret key. It sends a message with a random number r_j together with the names of both nodes to the other node. It accepts the test predicate if and only if the message comes back with a correct signature. The challenged node, on the other hand, signs the challenge if and only if it contained both its own name and that of the challenger.

The message complexity of the protocol is $3 \cdot n \cdot (n-1)$, as each node needs three messages to convince any other node of its test predicate. It takes 3 rounds of communication.

3.2 General properties

Signatures have the purpose to allow the assignment of a message to a unique signer. We state this fact in the following definition:

Definition 3.1 (Assignment): A node assigns a message $\{m\}_S$ to a node P_i , if it has accepted T_i as belonging to P_i and $T_i(\{m\}_S) =$ true.

In the case of local authentication, a node accepts a test predicate during the key distribution protocol (Fig. 1). When global authentication is used, acceptance of a test predicate must be reached by other means (e.g. by communication with a perfectly reliable node).

Authentication has the purpose to let the nodes make the assignments in a correct and consistent manner. We will compare local and global authentication with respect to their assignment properties. These are the properties of global authentication:

- (G1) If a correct node assigns a signed message to a correct node P, then P has signed the message.
- (G2) A message signed by a correct node P is assigned to P by all correct nodes.
- (G3) Each correct node assigns a signed message to the same node.

Property (G3) captures the case where the signer and the "owner" of a signature are faulty. When some faulty node gives its secret key to some other faulty node which uses this key to sign its messages, the signed messages are not assigned to the real signer. But still all correct recipients of the signed message assign it to the same node. Local authentication shares properties (G1) and (G2) as shown in the following theorem:

Theorem 3.2 After the key distribution protocol, (G1) and (G2) hold.

Protocol for each node P_i : Generate a secret key S_i and an appropriate test predicate T_i send T_i to all other nodes for each received T_j : select a random number r_j send $\{P_i, P_j, r_j\}$ to P_j for each received $\{P_j, P_i, r\}$ from P_j : send $\{P_j, P_i, r\}_{S_i}$ to P_j for each received $\{P_i, P_j, r\}_{S_j}$ from P_j : if $T_j(\{P_i, P_j, r\}_{S_j}) = \mathbf{true}$ and $r = r_j$ accept T_j as belonging to P_j

Figure 1: Key Distribution Protocol

Proof:

- 1. (G1): If a message $\{m\}_S$ is assigned to P by a correct node, P has shown that it knows S in the key distribution protocol. Since S is not sent in any step of the distribution protocol (P is correct) and (S1) and (S3) hold, P must have signed the message.
- 2. (G2): The proof is by contradiction. Assume a correct node P_i does not assign a message $\{m\}_{S_j}$ signed by a correct node P_j to the signer. Two things could have happened:
 - a) P_i does not recognize the signature or
 - b) P_i assigns the message to another node P_k .

In case a) P_j did not send its T_j to P_i or it did not correctly answer the challenge from P_i . Hence P_j would be faulty. In case b) P_i must have received a message $\{P_i, P_k, r\}_{S_j}$ from P_k . Hence either P_k must know the secret key S_j or P_j has signed the message instead. In both cases P_j would be faulty.

Unfortunately, property (G3) does not hold for local authentication. Cooperating faulty nodes may well distribute their test predicates in a mixed manner, such that two correct nodes assign a message to different faulty nodes. Another possibility is that a faulty node distributes different test predicates to the correct nodes. This leads to classes of nodes such that the faulty node can select the class of nodes which can assign the message at all. How this problem can be overcome in the context of failure discovery is shown in the next section.

3.3 Failure discovery properties

In this section we show that local authentication can be used to solve the *Failure Discovery* problem, introduced by Hadzilacos and Halpern [2], efficiently. The problem is to devise an algorithm that will ensure the following properties in the presence of up to t faulty nodes:

(F1) Weak Termination: Each correct node eventually either chooses a decision value or discovers a failure.

- (F2) Weak Agreement: If no correct node discovers a failure, then no two correct nodes choose different decision values.
- (F3) Weak Validity: If no correct process discovers a failure and the sender is correct, then no correct node chooses a value different from the sender's initial value.

If no failure is discovered, this is essentially Byzantine Agreement as defined in [4]. Note that it is not necessary that a failure discovering node can *identify* the faulty node, it has merely to *notice the existence* of a failure.

Hadzilacos and Halpern show that a protocol for Failure Discovery can be extended under certain conditions to a protocol for Byzantine Agreement. The interesting point is that the extended protocol requires in its failure-free runs the same number of messages as the underlying Failure Discovery protocol.

The authors point out that Failure Discovery can be solved with $O(n^2)$ messages without authentication and with O(n) messages if global authentication is available. We will show that, in the context of Failure Discovery, local authentication has the same properties as global authentication. So, once local authentication is established, one can run arbitrarily many Failure Discovery protocols with low message complexity. We start by showing that condition (F1) is not violated by the use of local authentication:

Lemma 3.3 If (F1) is fulfilled by a protocol under global authentication, it is fulfilled by the same protocol under local authentication.

Proof: The introduction of local authentication does not change the failure-free runs of the protocol. If a node has the view of a failure-free run, it hence will eventually decide for a value. As soon as its view is different, it discovers a failure. \Box

Conditions (F2) and (F3) are fulfilled trivially if a correct node discovers a failure. So, if we can guarantee that some node discovers a failure as soon as the properties of global authentication are violated, we are done. Of course, the protocol still has to ensure that failures which lead to incorrect agreement and are not related to the use of local signatures are discovered.

Protocol for P_1 : send value $\{v\}_{S_1}$ to P_2 Protocol for P_i $(2 \le i \le t)$: receive $m = \{S_{i-2} : \ldots S_1 : \{v\}_{S_1} \ldots \}_{S_{i-1}}$ from P_{i-1} check the signatures of the message and the submessages if negative then discover failure and stop else accept v and send $\{S_{i-1} : m\}_{S_i}$ to P_{i+1} Protocol for P_{t+1} : receive $m = \{S_{t-1} : \ldots S_1 : \{v\}_{S_1} \ldots \}_{S_t}$ from P_t check the signatures of the message and the submessages if negative then discover failure and stop else accept v and send $\{S_t : m\}_{S_{t+1}}$ to P_{t+2} to P_n Protocol for $P_t + 2$ to P_n : receive $m = \{S_t : \ldots S_1 : \{v\}_{S_1} \ldots \}_{S_{t+1}}$ from P_{t+1} check the signatures of the message and the submessages if negative then discover failure else accept v

Figure 2: Failure Discovery Protocol

Since properties (G1) and (G2) have been shown to be the same for both types of authentication (Theorem 3.2), all that is left to show is that (G3) holds if no failure is discovered. To show that, we first have a closer look at *chain signatures*. Chain signatures are a common mechanism in authenticated protocols. A message with a chain signature is a message which has been signed by a sequence of nodes, each one signing the signed message of its predecessor.

For our purposes we require that a message which has been signed before is always signed together with the name of the node it is assigned to. Hence, a message with chain signature has the following structure:

$$\{P_{n-1}: \{\ldots, P_2: \{P_1: \{m\}_{S_1}\}_{S_2}, \ldots\}_{S_{n-1}}\}_{S_n}$$

If a message $\{P_2 : \{P_1 : \{m\}_{S_1}\}_{S_2}\}_{S_3}$ is assigned to P_3 , it can be interpreted as: P_3 said that P_2 said that P_1 said m. We will call the messages which are contained within a message *submessages*. The submessages in the above example are $\{P_1 : \{m\}_{S_1}\}_{S_2}, \{m\}_{S_1}$ and m.

The intent of this kind of signature is to make everyone agree on who said what to whom. Whereas this aim is reached by global authentication, it is not reached by local authentication. Here, a message signed by a faulty node may be assigned to different faulty nodes or to no node at all, depending on the behaviour of the signer in the key distribution protocol. Fortunately, such a misbehaviour of a faulty node can at least be *discovered* under local authentication as will be shown in theorem 3.4.

We first observe the following: If in the above example a correct node not only assigns the complete message to P_3 but also the submessages to the respective given nodes, it knows that it has made the same assignments as its correct predecessors. Furthermore,

since the immediate sender of a message is known (N1) and all messages are signed, it knows that all other recipients of that message (as submessage or not) will assign it to P_3 or discover a failure. This observation is stated in the following theorem:

Theorem 3.4 After the key distribution protocol (Fig. 1) the following holds: All correct nodes assign a (sub-)message to the same node or at least one of them discovers a failure.

Proof: Assignment to the last signer: Since all messages have to be signed, the last signature must stem from the immediate sender of the message. This sender is recognizable for all nodes (N1). If a node assigns the message to a different node, it discovers a failure.

Assignment to the signers of submessages: If a correct node does not assign a submessage to the node stated before the message, it discovers a failure. Otherwise, it assigns the submessage to the same node as the other nodes which do not discover a failure. \Box

This shows that (G1) to (G3) hold for local authentication if no failure is discovered. Hence, a protocol that fulfills (F1) to (F3) with the assumption of global authentication has the same failure discovery properties when only local authentication is available.

4 Protocols

Fig. 2 shows a simple failure discovery protocol for an arbitrary value range taken from [1]: The sender, P_1 , signs its value and sends it to P_2 . P_2 , in turn, signs the message and gives it to P_3 . This is iterated until the message reaches P_{t+1} (with t denoting the number of tolerated faulty nodes). P_{t+1} then signs the message and disseminates it to the rest of the participants. This protocol works with the minimal number of messages of n-1 (cf. [3]). Since all messages are signed and the correctness of the protocol has been shown for global authentication, it can be applied under local authentication.

If the value range is known a priori and small compared to n, solutions with fewer messages are possible by assigning values to missing messages. Protocols of this type given in [3] fulfill the conditions for the application of local authentication, too.

In the same paper, Hadzilacos and Halpern state that non-authenticated protocols for arbitrary failures need $O(n \cdot t)$ messages, with t denoting the number of tolerated faulty nodes. With a constant portion of the nodes being faulty, this makes $O(n^2)$ messages. Hence, after executing the key distribution protocol once $(O(n^2)$ messages in 3 rounds), we can reduce the number of messages per protocol run from $O(n^2)$ to O(n).

5 Summary

We have introduced and examined a new authentication assumption for agreement protocols. This *local authentication* can be established in a nonauthenticated system without assumptions about the number or behaviour of faulty nodes. The necessary key distribution protocol needs $3 \cdot n \cdot (n-1)$ messages in 3 rounds.

We have shown that Failure Discovery protocols which were designed for completely authenticated environments can also be applied under local authentication. Since the message complexity of authenticated protocols (O(n)) is much better than that of non-authenticated protocols $(O(n^2))$, this approach gives a substantial message complexity gain in nonauthenticated environments.

Further research is necessary to investigate the use of local authentication with other agreement protocols, esp. with Byzantine Agreement. We conjecture that the fundamental impossibility results for nonauthenticated environments do not change with the assumption of a signature scheme, but we hope for improvements in the area of average message complexity and the parameters of weaker types of agreement, e.g. Degradable Agreement ([7]).

References

- Birgit Baum-Waidner. Byzantine Agreement with a minimum number of messages both in the faultless and worst case. In Proceedings of the 23rd International Symposium on Fault-Tolerant Computing (FTCS), pages 554-563. IEEE Computer Society Press, 1993.
- [2] Vassos Hadzilacos and Joseph Y. Halpern. The Failure Discovery problem. Math. Systems Theory, 26:103-129, 1993.
- [3] Vassos Hadzilacos and Joseph Y. Halpern. Message-optimal protocols for Byzantine Agreement. Math. Systems Theory, 26:41-102, 1993.
- [4] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. ACM Transactions on Programming Languages and Systems, 4(3):382-401, 1982.
- [5] National Institute of Standards and Technology. The Digital Signature Standard. Communications of the ACM, 35(7):36-40, July 1992.
- [6] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120-126, February 1978.
- [7] Nitin H. Vaidya and Dhiraj K. Pradhan. Degradable agreement in the presence of byzantine faults. In Proceedings of the 13th International Conference on Distributed Computing Systems, pages 237-244, 1993.