Paradigm Shifts in Protocol Analysis

Susan Pancho
University of Cambridge,
Computer Laboratory,
Cambridge CB2 3QG, U. K.,
sp261@cam.ac.uk

Abstract

Authentication protocols are widely believed to be error prone because most analyses conclude with claims of discovering new attacks on the protocols. While proofs of security for authentication protocols are rightly viewed with circumspection, claims of attacks are rarely challenged. We propose a closer examination of how protocol attacks are defined in the light of different conclusions of four different analyses of the Needham-Schroeder protocols. We argue that subtle paradigm shifts often occur during protocol analysis which affect the definition of a protocol attack. By becoming aware of these paradigm shifts, we can be more aware of what a specific attack actually accomplishes.

1 Introduction

Authentication protocols are believed to be error prone because most analyses of these protocols conclude with the discovery of an attack on the protocols. These errors are supposedly rooted in the protocol design and are characteristic of cryptographic protocols in general. There have been guidelines for better engineering of such protocols, e.g., [1] coupled with improved methods for protocol analysis. However, there is no standard way of detecting an attack on a protocol. In fact, there exists several approaches in analysing a protocol. These include the use of formal logics (e.g. BAN [3], GNY [7], SVO [20]), process algebra (e.g. CSP [19]) and specialised state engines such as the NRL Protocol Analyser [16]. These various methods can yield different conclusions about a single protocol because each tool may detect a different type of attack although there may be flaws commonly found by most methods. What is disconcerting is to discover conflicting conclusions such as a pronouncement that a protocol is correct vis-à-vis a discovery of a flaw in the same protocol. Given such conflicting results, we generally believe that one tool has *missed* detecting the flaw found by the second tool.

Specifically, we consider the case of the Needham-Schroeder authentication protocols [18]. Published in the late 1970s, the first known analysis was that conducted by Denning and Sacco [4]. However, these protocols have been continuously analysed and these further analyses gave different conclusions about the security of the protocols. The BAN logic analysis of the protocols [3] is said to have failed to detect the attacks later claimed by Lowe [9] and Meadows [15]. Moreover, Lowe claims to have found a more subtle and more recent attack than that discovered by Denning and Sacco [9, 10] while Meadows claims to have reproduced Lowe's attack in addition to discovering new flaws [15].

Clearly, the use of one method of analysis gives no guarantee that another method will not discover a new flaw with the same protocol. Moreover, proofs

1999 New Security Paradigm Workshop 9/99 Ontario, Canada

of security are rightly viewed with circumspection but claims of attacks are often classed together and rarely challenged. If the use of one method fails to detect a flaw discovered by using a second approach, we generally attribute this conflict to the superiority of the second analysis. However, in examining different analyses of the Needham-Schroeder protocols [3, 4, 9, 10, 15], we observe that most of the attacks were products of how the protocol's security context was interpreted rather than of the use of different analysis tools.

2 Interpreting a protocol's security context

Authentication protocols are used for a variety of applications. This variety is evident in the different flavours of authentication since many applications define the term differently. The variation in how a protocol defines and achieves authentication suggests that in trying to understand a particular protocol, one has to look closely at how seemingly familiar terms are defined. The security context of an authentication protocol is defined by the sum of its components: *goals* it seeks to attain, *assumptions* that define the application environment the protocol was designed for, *messages* describing how participants interact, and the *checks* that participants perform within the course of the protocol. Any analysis of a protocol relies on an interpretation of its security context.

2.1 Interpreting protocol messages and checks

Protocols are often cited and described in terms of the messages exchanged during a protocol run. This description includes the protocol participants, their role(s) in the message exchange, the messages sent, and the checks each participant performs when a particular message is received. The message exchange is the most obvious representation of a protocol, probably because messages are enumerated and are more explicitly stated than the other protocol components. Protocol checks may not be defined with the same level of detail, but during the course of analysing message exchanges, one cannot help but consider what protocol checks are required for each step of the protocol. For example, a protocol check could include distinguishing between the data types of the different components of messages. Some protocols assume that principals can distinguish between a nonce and a name. Without this check, a protocol may be vulnerable to type confusion attacks such as those described in [15]. The exhaustiveness of these protocol checks vary. In some cases, these checks are at best only implied in the protocol design. Some analysis methods such as the BAN logic [3] seek to bring out the protocol checks required by examining what a participant can reasonably believe after receiving a particular message.

2.2 Interpreting protocol goals and assumptions

Specific goals define what a protocol seeks to provide. They give a more concrete picture than the ambiguous meanings attached to *authentication*. Without the explicit statement of these goals, it is possible to misunderstand the purpose of a protocol. When the purpose is unclear, a protocol may be unknowingly used for an application it was not designed for. It may also be analysed as defective because it did not achieve goals it never sought to attain in the first place. For example, an authentication protocol that establishes a shared secret between two principals may have a different purpose from a challenge-response protocol even if both are classified as authentication protocols. Indeed, there are several possible interpretations of an authentication protocol's goals [2, 5, 11].

Assumptions define the application environment for which the protocol was designed for. Assumptions are important because they do not only describe a protocol's environment, they also define its limitations.

Assumptions can be made regarding the behaviour of protocol participants. For example, a protocol that assumes that all principals are honest could fail an analysis that relaxes this assump-

tion. This is because in the original context, the designers would most likely be geared towards securing the protocol from external threats rather than from misbehaved principals. Assumptions about behaviour of participants also define the protocol's concept of an intruder. If all participants are assumed honest, then an intruder would be an outsider to the protocol system.

Assumptions can also be made regarding the properties of the cryptographic mechanisms used in the protocol. For example, some protocols use encryption as a security primitive but it is not often spelled out why using an encryption algorithm is necessary. In fact, authentication protocols may omit the use of encryption functions and instead apply one-way functions [6, 13]. If encryption is used, it is important to consider the intended purpose. For example, a cipher block chaining (CBC) algorithm was unlikely to be suitable as a cipher algorithm for an authentication protocol if the intended purpose is providing integrity instead of confidentiality [13].

Unlike protocol messages, goals and assumptions are not often stated in detail. Some assumptions may not even be explicitly spelled out. In [8], it was shown how under-specification of system behaviour and the assumptions made about the system's environment could produce undesirable emergent behaviours. This lack of explicitness could lead to confusion when one tries to model a protocol's overall security context.

3 Attacks and variation of contexts

There is no standard way of defining the context of a protocol. It is possible that a protocol description could be ambiguous and this could be due to an oversight of the designers or a limitation of the notation used for protocol specification. There are proposals for standardised specifications, e.g., [17] as well as stricter notations [12, 14] to help avoid confusion, but none has yet replaced the current ad-hoc descriptions of protocols.

Different interpretations of a protocol specification can lead to an unintentional change in the context of a protocol during analysis. However, one can also intentionally deviate from the original context, in order to examine the effectiveness of the protocol in a different environment. Thus, intentionally or not, protocol analysis could operate on a different protocol context from the original. One would expect protocol designers to design a secure protocol based on their original conditions and assumptions. However, when an analysis deviates from the protocol's intended environment, it is possible to discover a new *weakness* that would reflect the importance of the original condition that was modified or removed.

3.1 Modifying protocol assumptions

Protocol designers make assumptions about the environment of the protocol and these assumptions in turn contribute to the definition of the protocol itself. When one of these original conditions is removed or modified, one has already changed the environment of the protocol. Discovering a flaw in this revised protocol could be due to the change made in protocol environment.

3.1.1 The Denning-Sacco analysis

The Needham-Schroeder authentication protocol using shared keys seeks to establish a shared secret key between two principals (A and B) who wish to communicate over an insecure network. An authentication server (AS) trusted by both principals generates this shared secret key, CK that will be subsequently used by A and B for communication. Each principal shares a secret key with the authentication server, denoted as K_{AS} and K_{BS}. A nonce generated by an entity I is denoted as N_I while encryption of a message X with key K is denoted as $\{X\}_K$. The protocol messages are as follows:

1. $A \rightarrow AS : A, B, N_A$
2. $AS \rightarrow A : \{N_A, B, CK, \{CK, A\}_{K_{BS}}\}_{K_{AS}}$
3. $A \rightarrow B : \{CK, A\}_{K_{BS}}$

4. $B \rightarrow A : \{N_B\}_{CK}$

5. $A \rightarrow B : \{N_B - 1\}_{CK}$

Needham and Schroeder assumed that the communication key CK is unpredictable, fresh and is known only to the two communicating principals and the trusted authentication server [18]. In [4], Denning and Sacco removed this major condition and examined the consequences of the compromise of the communication key CK. If an intruder E obtains this key CK, the protocol may be attacked as follows:[1]

1. $E(A) \rightarrow B : \{CK, A\}_{K_{BS}}$

2. $B \rightarrow E(A) : \{N_B\}_{CK}$

3. $E(A) \rightarrow B : \{N_B - 1\}_{CK}$

The intruder E intercepts and replays an old message to initiate a session with B. From the old message $\{CK, A\}_{K_{BS}}$, B assumes that A initiated the session and sends the reply to A. However, the intruder E can intercept this message and decrypt it because E possesses the supposedly secret key CK. Further messages sent between A and B that make use of this communication key are vulnerable to interception and decryption by E. E is also capable of sending faked messages. Denning and Sacco suggested the use of a timestamp, T to prevent replays of old, compromised keys.

1. $A \rightarrow AS : A, B$

2. $AS \rightarrow A : \{B, CK, T, \{CK, A, T\}_{K_{BS}}\}_{K_{AS}}$

3. $A \rightarrow B : \{CK, A, T\}_{K_{BS}}$

The use of timestamps was also recommended for the public-key Needham-Schroeder protocol to guarantee the integrity of public keys.

[1] $E(A)$ denotes E posing as A.

3.2 Lowe and Meadow's analysis

Instead of removing an original assumption, one can modify a protocol's context by modifying an assumption. One way to do this is to change the threat model of a protocol by changing the assumptions about who can be classified as an *intruder*.

The public-key version of the Needham-Schroeder protocols has the following messages:[2]

1. $A \rightarrow AS : A, B$

2. $AS \rightarrow A : \{K_B, B\}_{K_{AS}^{-1}}$

3. $A \rightarrow B : \{N_A, A\}_{K_B}$

4. $B \rightarrow AS : B, A$

5. $AS \rightarrow B : \{K_A, A\}_{K_{AS}^{-1}}$

6. $B \rightarrow A : \{N_A, N_B\}_{K_A}$

7. $A \rightarrow B : \{N_B\}_{K_B}$

This protocol has also been previously analysed using the BAN logic [3] (see Section 3.3).

The primary goal of this protocol is to allow two principals to communicate securely over an insecure network. The threat was external since principals are assumed to be honest. We quote from Needham and Schroeder [18]:

> "Our viewpoint throughout is to provide authentication services to principals that choose to communicate securely."

This assumption was modified in [9]; a principal can also be an intruder and can abuse the trust of other principals. Quoting Lowe [9]:

> "We assume that the intruder I is a user of the computer network, and so is able to set up standard sessions with other agents, and other agents may try to set up sessions with I..."

[2] K_X^{-1} denotes X's private key while K_X denotes X's public key.

The attack is as follows:[3]

1. $A \rightarrow E : \{N_A, A\}_{K_E}$
2. $E(A) \rightarrow B : \{N_A, A\}_{K_B}$
3. $B \rightarrow E(A) : \{N_A, N_B\}_{K_A}$
4. $E \rightarrow A : \{N_A, N_B\}_{K_A}$
5. $A \rightarrow E : \{N_B\}_{K_E}$
6. $E(A) \rightarrow B : \{N_B\}_{K_B}$

Lowe assumed that the intruder E is a principal with whom other principals may initiate a session with (as seen in message 1). Thus, the protocol adopts a context wherein principals can be dishonest. This poses a threat from within the system. The *intruder* can choose to be a legitimate protocol participant and could manipulate this to its advantage. This is a stronger assumption than that of an *external intruder*, the original assumption of Needham and Schroeder. An *external intruder* can be thought of as one who is interposed between the two principals who wish to communicate. It can intercept, copy and modify the messages passed. In the BAN logic analysis of Needham-Schroeder [3], it was explicitly stated that in the original protocols, encryption was used to allow two principals to communicate securely in the presence of such an intruder but no provision was made for principals who misbehave.

In [15], the threat model adopted was closer to that used by Lowe [9]. In addition, new attacks were discovered because it was further assumed that type confusion between nonces and names can occur. This thus leads to the attack described in [15] (see Figure 1).

From message 5 in Figure 1, A interprets N_E as a nonce but N_B is interpreted as a name. In the next message, A then sends this "name" to the server S to obtain N_B's public key. This "name" is sent in clear which allows the intruder to intercept it.

The second attack from [15] is given in Figure 2.

[3]Steps related to obtaining public keys from the server AS were omitted by Lowe.

The attack in Figure 2 is similar to the first attack since it also takes advantage of type confusion. It could occur when a principal A sends a message to itself although it is not clear under what circumstances this will happen. Furthermore, the attack consists of the intruder E intercepting a message sent by A to A itself and forwarding this "intercepted" message to A (step 4). This intercepted message will be interpreted by A as a successful response to itself because of the presence of its nonce in the message.

Meadows' analysis also shows how a change in a protocol's security context could lead to the discovery of an attack. In this case, by considering the effect of type confusion, several possible attacks were discovered.

3.3 Using the same context

The BAN analysis [3] of the Needham-Schroeder protocols used the original context of the protocol. The BAN logic assumes that principals are honest which is also an assumption of the Needham-Schroeder protocol. The purpose of the analysis was to discover whether the protocol meets its intended goals and what assumptions are necessary for such an achievement. It was not intended to discover the effect if principals are dishonest.

From the analysis [3], the shared-key version of the protocol was found to have made a strong assumption about B's belief regarding the freshness of the shared key CK. The analysis highlights that B needs to believe in the freshness of the shared key CK without good reason to do so. Unlike the message received by A from the server AS, the message B receives has no component which B can trust to be fresh. The logic shows that without this assumption, it is not possible for B to accept the shared key CK as fresh.

The analysis of the public-key Needham-Schroeder protocol also highlighted two strong requirements of the protocol: each principal has to assume that the public key of the other principal is fresh. There is no basis in the protocol which the principals can use to deduce the freshness of public keys and thus the use of timestamps was proposed.

1. $E(A) \rightarrow B : \{N_E, A\}_{K_B}$

2. $B \rightarrow S : A$

3. $S \rightarrow B : \{K_A, A\}_{K_S^{-1}}$

4. $B \rightarrow A : \{N_E, N_B\}_{K_A}$ E intercepts this message.

5.$E(N_B) \rightarrow A : \{N_E, N_B\}_{K_A}$ E sends the intercepted message to A as the initiator of the protocol, with N_B as the name field.

6.$A \rightarrow S : N_B$ A sends the "name" N_B to S in order to get its public key.

7.$E(A) \rightarrow B : \{N_B\}_{K_B}$

Figure 1: Meadow's First Attack

1. $A \rightarrow S : A$

2. $S \rightarrow A : \{K_A, A\}_{K_S^{-1}}$

3. $A \rightarrow A : \{N_A, A\}_{K_A}$ This message is intercepted by E.

4. $E(A) \rightarrow A : \{N_A, A\}_{K_A}$

Figure 2: Meadow's Second Attack

The BAN analysis of Needham-Schroeder gives an example of a protocol analysed under its own terms. Results achieved in this manner reflect on the original protocol design and highlight potential holes even if the protocol was used as it was intended to be used.

4 Paradigm shifts and security contexts

Of the four different analyses of the Needham-Schroeder protocols [4, 3, 9, 15] that we have examined, three modified the original protocol environment. In the case of Denning and Sacco's analysis, the modification was intentional and explicitly stated. However, the different interpretations of the protocol's environment in Lowe's and Meadow's analyses reflect a subtle paradigm shift in defining the protocol's threat model.

When Needham and Schroeder designed their protocols, their focus was on allowing two parties to communicate over an insecure network. In their original scenario, the two communicating parties were assumed to be honest but their messages have to pass through a wicked network. Thus, they modelled their attacker to be someone who could tap the network lines, eavesdrop on the messages that get sent back and forth, and even insert and modify the messages that pass through the network. However, they did not account for one of the communicating parties abusing their role as a legitimate protocol participant.

Lowe's and Meadow's analyses assumed a stronger attacker, who is in effect, a protocol participant. Under these new conditions, the protocol was expected to anticipate ill behaviour from a participant and prevent it from adversely affecting other parties. A protocol that is not designed to be robust against insider attacks would naturally fail under these conditions.

This subtle shift from threats from external attackers to misbehaving insiders is reflected in the differences in the security contexts. With a growing number of potential principals that could make use of a security protocol, the current threat model has become significantly different from that considered by Needham and Schroeder. Participants still choose to communicate securely but we now have to contend with the harm that a peer could achieve, in addition to the attacks an outsider could mount. This paradigm shift has occurred subconsciously and has affected how we perceive protocols, even those that have been designed under a different paradigm.

5 Weighing the claims

An analysis that finds a weakness in a protocol under the original context highlights an oversight of the protocol designers. The BAN logic [3] analysis of Needham-Schroeder achieved this when they pointed out several key assumptions that were not explicitly stated in the original paper. These weaknesses correctly refer to the original protocol context. On the other hand, an analysis that modifies the protocol context produces a result that does not strictly refer to the original protocol but rather to the protocol under a different operating environment. It is still a useful result because it qualifies the limitations of the original protocol. However, it is important to be explicit about the change in context.

The Denning-Sacco analysis of the Needham-Schroeder protocols is often quoted as having found a flaw in the protocol. However, their study modified the context of the original protocol by relaxing an original assumption of Needham-Schroeder. The authors were aware that they have changed the protocol's context and have noted this in their paper [4]. We quote:

> "If communication keys and private keys are never compromised (as Needham and Schroeder assume), the protocol is secure (i.e. can be used to establish a secure channel)."

They qualified that their finding is significant if the Needham-Schroeder protocols are applied to an

environment wherein communication keys may be compromised.

If the protocol environment is changed and this is not explicitly stated, there is a danger of unequivocally accepting claims of attacks without realising the context of the analysis. The modified protocol contexts applied by Lowe and Meadows led to the discovery of new attacks. These attacks were a product of a subtle paradigm shift that led to differences in security contexts. Thus, whenever these attacks are cited the results are assumed to reflect on the original protocol. This gives the impression that the protocols were supposed to have been secure against insider attacks although they were not intended to be so. Qualifying the protocol context does not subtract from the benefits of the analysis, rather it helps dispel some of the misconception and confusion about how vulnerable a protocol really is.

Given that there are differing interpretations of a protocol's security context, one may be tempted to seek for a methodology that would eliminate the ambiguities through the use of a highly specific language. Such a language could be used to explicitly state the goals and assumptions of a protocol. This language could even be standardised for security protocols in general. There are in fact, several efforts in this direction, e.g. [17]. The problem with relying only on this solution is that it gives the impression that the whole protocol environment could be easily articulated. However, as in the case of subtle paradigm shifts in protocol environments, one can only articulate well what one is already aware of.

6 Conclusion

When an analysis of an authentication protocol results in validation of a previously known hole or discovery of a new attack, it reinforces the belief that such protocols are error-ridden. However, the ambiguities in defining a protocol affects the analysis process since interpretation of a protocol's security context could vary widely. Moreover, the analysis itself could intentionally deviate from the original context. Thus, we have a situation wherein not only do we vary in defining what a protocol achieves, but also on what constitutes as an attack on that protocol. A *protocol attack* can be a flaw in the original protocol or it can be a qualification of a limitation of the protocol. A *protocol attack* can be one that would immediately break the protocol or it could be a possible error that would occur when the protocol is used in a certain way.

In the case of the Needham-Schroeder protocols, several known flaws often attributed to weaknesses in the original design are more likely caused by a modified security context in the analysed protocol. In Denning and Sacco's analysis, the modification was intentional. However, in the analyses done by Lowe [9] and by Meadows [15], a primary assumption in the original protocol's threat model was modified, thus changing the protocol environment. This change was subtle and implicit and was instrumental to the discovery of the "insider attack" on the protocol. It was precipitated by a general paradigm shift in security protocols, where protection from insider attacks had to be considered in addition to external threats. This indicates that conflicting results from different analyses could be a product not just of differences in analysis methods, but in differences in protocol modelling.

Although these initial observations are restricted because only one protocol was studied, both the protocol and the analyses that were considered are characteristic of others in the field. Examining the analyses of other security protocols, specially those that have different or conflicting conclusions could provide further insights to protocol design and analysis.

Acknowledgments

This paper has benefited greatly from discussions with Prof. Dieter Gollmann. The author would also like to thank the workshop participants and the anonymous referees for their constructive comments as well as suggestions for further investigations.

References

[1] M. Abadi and R. Needham. Prudent engineering practice for cryptographic protocols. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, pages 122–136, 1994.

[2] C. Boyd. Towards extensional goals in authentication protocols. In *Proceedings of the DIMACS Workshop on Cryptographic Protocol Design and Verification*, 1997.

[3] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. Technical Report 39, DEC Systems Research Center, February 1990.

[4] D. E. Denning and G. M. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8):533–536, August 1981.

[5] D. Gollmann. What do we mean by entity authentication? In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 46–54, May 1996.

[6] L. Gong. Using one-way functions for authentication. *Computer Communication Review*, 19(5):8–11, 1989.

[7] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 234–248, 1990.

[8] H. M. Hinton. Under-specification, composition and emergent properties. In *Proceedings of the 1997 ACM New Security Paradigms Workshop*, pages 83–93, 1997.

[9] G. Lowe. An attack on the Needham-Schroder public-key authentication protocol. *Information Processing Letters*, 56(3):131–133, 1995.

[10] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science 1055, pages 147–166. Springer-Verlag, 1996.

[11] G. Lowe. A hierarchy of authentication specifications. In *Proceedings of the 10th IEEE Computer Security Foundations Workshop*, pages 31–43, 1997.

[12] W. Mao and C. Boyd. Classification of cryptographic techniques in authentication protocols. In *Selected Areas in Cryptography*, pages 95–106, May 1994.

[13] W. Mao and C. Boyd. Development of authentication protocols: Some misconceptions and a new approach. In *Proceedings of the IEEE Computer Security Foundations Workshop*, pages 178–186, June 1994.

[14] W. Mao and C. Boyd. Methodical use of cryptographic transformations in authentication protocols. *IEE Proceedings on Computers and Digital Techniques*, 142(4), July 1995.

[15] C. Meadows. Analyzing the Needham-Schroeder public key protocol: A comparison of two approaches. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS) 1996*, Lecture Notes in Computer Science 1146, pages 351–364. Springer Verlag, 1996.

[16] C. Meadows. Language generation and verification in the NRL protocol analyzer. In *Proceedings of the 1996 Computer Security Foundations Workshop*, pages 48–61, June 1996.

[17] J. Millen. CAPSL: Common authentication protocol specification language. http://www.csl.sri.com/~ millen/capsl. 2 September 1999.

[18] R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, December 1978.

[19] S. Schneider. Verifying authentication protocols with CSP. In *Proceedings of the 10th IEEE Computer Security Foundations Workshop*, pages 3–17, 1997.

[20] P. F. Syverson and P. C. van Oorschot. On unifying some cryptographic protocol logics. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, May 1994.