On Key Distribution Protocols for Repeated Authentication

Paul Syverson
Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, DC 20375

(syverson@itd.nrl.navy.mil)

Abstract

In [KSL92], Kehne et al. present a protocol (KSL) for key distribution. Their protocol allows for repeated authentication by means of a ticket. They also give a proof in BAN logic [BAN89] that the protocol provides the principals with a reasonable degree of trust in the authentication and key distribution. They present an optimality result that their protocol contains a minimal number of messages. Nonetheless, in [NS93] Neuman and Stubblebine present a protocol (NS) as an explicit alternative to KSL that requires one less message in the initial authentication and key distribution. One goal of this paper is to examine some of the reasons for this discrepancy. Another goal is to demonstrate possible attacks on NS. Like any attacks on cryptographic protocols, these depend on assumptions about implementation details. But, when possible they are serious: a penetrator can initiate the protocol, masquerade as another principal, obtain the session key, and even generate the session key herself.[1] We will set out implementation assumptions required for the attacks to take place and implementation assumptions that preclude such an attack. We will also look at other protocols, including one that is not subject to this form of attack and has the same number of messages as NS. Finally, we will briefly discuss the logical analysis of these repeat authentication protocols.

Introduction

In [KSL92] Kehne, Schönwälder, and Langendörfer present a protocol for multiple authentications that is meant to serve as a nonce based alternative to Kerberos. (Henceforth, this will be called the 'KSL protocol'.) Kerberos makes use of timestamps, which some have argued is a drawback since it requires at least a loose synchronization between the clocks of the various principals. The only timestamp in the KSL protocol is just for the use of the principal who generates it. So, synchronization is not required. The protocol produces a ticket that can be used for subsequent authentication with only three messages. The initial authentication and key distribution requires five messages. Kehne et al. present an argument that this is the minimum number of messages in order to insure adequate trust by the principals in the goodness of the key under the assumptions they set out.

Despite this argument, and as an alternative to the KSL protocol, Neuman and Stubblebine in [NS93] present a nonce-based protocol for key distribution that contains only four messages in the initial exchange. (Henceforth, this will be called the 'NS protocol'. *N.B.*, this is not a reference to the Needham-Schroeder protocol, for which cf. [NS78] and [BAN89].) This protocol will serve as a focus for our initial discussion. We will set out the protocol and describe how to attack it. After that we will analyze the protocol and the assumptions underlying the attacks. In particular, we will look both at what must be assumed for an implementation of the protocol to be secure against such attacks and at other protocols that are not subject to such attacks.

We will also look briefly at the logical analysis of repeat authentication protocols including why the analyses in [KSL92] and [NS93] yield different results.

1 The NS Protocol

The NS Protocol has two parts. The first part consists of an exchange of four messages that results in

---

[1] I have presented these attacks at the 1993 Cambridge Workshop on Cryptographic Protocols and at talks to the Naval Research Laboratory Center for High Assurance Computer Systems, Indiana University Applied Logic Group, and the Norwegian Defense Research Establishment. I thank the attendees of these talks for many useful comments. The attacks have also been independently discovered by Ulf Carlsen [Car93].

the distribution of a session key for secure communication between two parties via a key distribution server. It also results in the distribution to the protocol initiator of a ticket which she can use in a subsequent authentication. The advantages of the ticket are at least twofold. First, it allows for authorization of the session key and authentication of the principals in a subsequent communication involving only three messages. Second, this subsequent communication requires only the present actions of the two principals: no server is needed. Here is a representation of the initial exchange.[2] An explication follows immediately.

The initial exchange

(1) $A \rightarrow B$: A, N_a
(2) $B \rightarrow S$: $B, \{A, N_a, T_b\}_{K_{bs}}, N_b$
(3) $S \rightarrow A$: $\{B, N_a, K_{ab}, T_b\}_{K_{as}}, \{A, K_{ab}, T_b\}_{K_{bs}}, N_b$
(4) $A \rightarrow B$: $\{A, K_{ab}, T_b\}_{K_{bs}}, \{N_b\}_{K_{ab}}$

Following standard practice, we refer to the initiator, A, of the protocol and the other principal, B, as 'Alice' and 'Bob' respectively. S is called the 'server'. The protocol runs as follows. Alice sends to Bob her own name (so he knows who is attempting to communicate with him) and a nonce, a random number that she will use to verify the freshness of later messages that contain it. Second, Bob sends to the server his own name and a nonce of his own. In the same message he sends the following, all encrypted together with a key, K_{bs}, good only for communication between Bob and the server: A, N_a, and a suggested expiration time for the authentication and the session key, T_b.[3] In the third message, the server sends to Alice: $\{B, N_a, K_{ab}, T_b\}_{K_{as}}, \{A, K_{ab}, T_b\}_{K_{bs}}, N_b$. The first encrypted chunk tells Alice that the server has been talking to Bob, that the message is fresh (via N_a) and gives her the session key and the expiration time for the key. The second encrypted chunk gives her a ticket that she can use for current and subsequent authentication with Bob. In the fourth message, Alice sends the ticket to Bob. This tells him that the server has talked to Alice and gives him the session key. Because Alice has used the session key to encrypt Bob's nonce, the second encrypted chunk lets him know that she has seen the session key recently. Note that the expiration time ordinarily plays no immediate role in the fourth message. (An exception might be if it took a long time to arrive.) But, it is included for subsequent authentication when it will be used to determine if the key is still valid.

Subsequent authentication

(1′) $A \rightarrow B$: $N'_a, \{A, K_{ab}, T_b\}_{K_{bs}}$
(2′) $B \rightarrow A$: $N'_b, \{N'_a\}_{K_{ab}}$
(3′) $A \rightarrow B$: $\{N'_b\}_{K_{ab}}$

In the first message, Alice generates a new nonce and sends this to Bob, along with the ticket from the initial exchange. Bob then checks the expiration time of the ticket. If the key is still good he generates his own new nonce, which he sends to Alice. He also sends her back the nonce she generated encrypted with the session key. Since this key is used only by Alice and Bob and since she knows the nonce is fresh, upon her receiving this, Bob will be authenticated to Alice. Finally, Alice encrypts Bob's nonce with the session key and sends it back to him, thus authenticating Alice to Bob.

2 Attacking the Protocol

In this section we will look at how to attack the protocol set out above. We will leave any analysis of the attacks and discussion of the assumptions necessary for the attacks to succeed until the next section. We will follow the not yet standard terminology of referring to the penetrator as 'Eve'. When Eve is masquerading as another principal or intercepts a message, we will indicate this via subscripts. For example, the symbol for Eve when sending a message as though from Alice or when intercepting a message intended for Alice is 'E_a'.

Attacking the initial exchange

(1) $E_a \rightarrow B$: A, N_a
(2) $B \rightarrow E_s$: $B, \{A, N_a, T_b\}_{K_{bs}}, N_b$
(3) Omitted
(4) $E_a \rightarrow B$: $\{A, N_a(= K_{ab}), T_b\}_{K_{bs}}, \{N_b\}_{N_a}$

The attack proceeds as follows: the first two steps of the protocol are as usual except that Eve masquerades as Alice to initiate and also intercepts the message from Bob to the server. The third message is omitted. In place of the usual first part of the fourth message Eve (as Alice) sends the encrypted portion of the second message. This has the same format as what should occur except that it substitutes a nonce where the session key should be. Since Eve has this nonce/session key (indeed, she generated it) she is able to use it to encrypt Bob's nonce. Thus, she can produce the second portion of the fourth message. At this point she has succeeded both in establishing a session with Bob while masquerading as Alice and in having him accept a session key that she holds.

---

[2] The initial exchange of the NS protocol is based on a protocol due to Yahalom, which does not appear to be subject to the attacks set out below. (cf. [BAN89] or [Yah])

[3] This is not the standard usage of such notation, which is ordinarily reserved for timestamps.

Attacking subsequent authentication

In this attack the penetrator may either masquerade as Alice for the initial exchange or she may simply eavesdrop on a legitimate initial exchange. In either case she also has the option of allowing the initial exchange to run to completion or she can stop it after the second message. She can then attack subsequent authentication as follows:

(1′) $E_a \rightarrow B$: $N_a', \{A, N_a(= K_{ab}), T_b\}_{K_{bs}}$
(2′) $B \rightarrow E_a$: $N_b', \{N_a'\}_{N_a}$
(3′) $E_a \rightarrow B$: $\{N_b'\}_{N_a}$

The attack involves the same substitution for the ticket as the above attack, i.e., the encrypted portion of the second message from the initial exchange. Once again nonce N_a is substituted for K_{ab}. But, since N_a occurred as plaintext in the initial exchange, it is available to Eve—whether or not she actually generated it.

3 Analysis

In the next subsection we set out some of the assumptions on which the attack relies. In addition, we will propose possible ways to avoid the attack. These are of two kinds. First, one can separate out those implementations where the attack is possible from those in which it is not and then explicitly state that the protocol is to be implemented only in those ways that preclude the attack. Second, one can consider alternatives to the protocol (on the current level of abstract description) that do not have implementations subject to this form of attack. We will consider both.

Implementation Dependencies

The primary assumption on which the attack depends is of course that substitution of a nonce for a key may pass undetected. This is not entirely unreasonable; they are after all both freshly generated random numbers. In a given implementation of the protocol this might be detectable if the implementation provides for adequate recognition of types, i.e., distinguishes nonces from keys. Even if typing were not detectable, the attack might still depend on whether or not nonces and keys were typically of different length. Eve might be able to compensate for this by padding or truncating the nonce in the first message. The success of her attempt to compensate could depend on still other details. Perhaps the implementation detects incorrect lengths of certain fields. Alternatively, if some field were of the incorrect length, the implementation might automatically truncate the field when producing the encrypted portion of the second message, or it might simply fail.

Another implementation assumption is that the same encryption algorithm is used in the second and the fourth message. Even though the same key is used for the relevant part of both messages, if the encryption algorithm is different the attack will not be possible.[4]

The attack also assumes that direction bits are not being used.[5] Though their means of implementation differs, using direction bits is at most as strong as having encrypted *to* and *from* fields. Assuming such fields, in any encrypted text within a given message it is always detectable who sent the message. Since the second message is from Bob and the ticket is from the server (via Alice), Bob should be able to detect the substitution of the encrypted portion of the second message into the fourth. Thus, an alternative to the NS protocol that differs only in adding these fields would not be subject to the above attack. This would, however, add at least slightly to the expense of encrypting and sending of the relevant messages. Therefore, rather than this alternative protocol we might want to simply restrict ourselves to implementations of NS where direction bits are being used, which might be less costly.[6]

We note also that while direction bits would solve the attacks on this particular protocol, they cannot be viewed as a general solution. In [Syv93] we discussed a general class of attacks on protocols that we called "causal consistency attacks" because they are characterized by a mismatch of the local histories of the protocol for each of the principals. Thus, although they each apparently see the right things in executing the protocol, their model of the causal chain of events differs. In other words, the principals fail to have matching histories of the protocol. (Matching histories was raised as a component in the characterization of a secure protocol in [DvOW92].) The above attacks are causal consistency attacks. In [Syv93] we presented a protocol that is subject to causal consistency attack whether or not direction bits are used. Similar attacks have also been looked at in [BGH$^+$92] and [Sne92], and they are also detectable using the methods of [SM93]. They are generally not expressible in BAN, of which more will be said below. For a more general discussion of such attacks and such methods the reader is referred to the papers cited in this paragraph.

Another possible alternative protocol would be one that introduces a more general kind of typing. One

---

[4] I thank Wenbo Mao for pointing this out to me.

[5] This was first pointed out to me by Virgil Gligor and Li Gong. It was also pointed out independently by Stuart Stubblebine.

[6] I thank Doug Maughan and Li Gong for helpful discussions on the general nature and application of direction bits.

could attach to each chunk of encrypted text an indicator that says that, e.g., this is a piece of message n in a run of protocol X.[7] This would preclude the above attack, and I suspect it would preclude any causal consistency attacks. One drawback to requiring such types in all cryptographic protocols would be the unnecessary additional expense in computation and communication when such attacks were not possible for other reasons. (For example, if we knew that using direction bits were sufficient to rule out the attack against a given protocol and we limited ourselves to those implementations employing direction bits.) This additional expense might be outweighed by the general cost of determining where the types are necessary and where they are not. A more immediate practical concern is the redundancy introduced by this measure. Amongst other possibilities, it would allow for exhaustive search attacks on password based encryption as in, for example, Kerberos.

One more possible way of avoiding the above attacks on the NS protocol would be to simply revert to the KSL protocol that preceded NS. This has the advantage of not being subject to the implementation restrictions of NS, i.e., we need not make the above mentioned assumptions about ability to recognize type, the use of direction bits, etc. in order for an implementation to be secure against attacks like the one above.[8] However, KSL has the disadvantage of requiring one more message than NS. Is it possible to construct a protocol that has no implementations subject to the above type of attack (like KSL) but that is no more expensive than NS? We can block the substitution of a nonce for a key by simply switching the order of timestamp and nonce in the encrypted field of the second message of NS. For convenience, this is called the permuted protocol.

Permuted protocol

(1) $A \rightarrow B$: A, N_a
(2) $B \rightarrow S$: $B, \{A, T_b, N_a\}_{K_{bs}}, N_b$
(3) $S \rightarrow A$: $\{B, N_a, K_{ab}, T_b\}_{K_{as}}, \{A, K_{ab}, T_b\}_{K_{bs}}, N_b$
(4) $A \rightarrow B$: $\{A, K_{ab}, T_b\}_{K_{bs}}, \{N_b\}_{K_{ab}}$

Even if type substitutions are not readily detectable, the above attack is not possible on this protocol because the timestamp in the second message must match the timestamp in the fourth, and these are not in the same location within the encrypted field. Unfortunately, other similar attacks are possible. For example, the penetrator can obtain a fresh timestamp, T_b, via a correct initial exchange. She can then attack the protocol using both the initial exchange and the subsequent authentication as follows:

(1) $E_a \rightarrow B$: $A, N_a (= N, T_b)$
(2) $B \rightarrow E_s$: $B, \{A, T_b', N, T_b\}_{K_{bs}}, N_b$
(3) Omitted
(4) Omitted

Here N_a consists of the timestamp previously obtained, T_b, appended to a nonce (or nonce initial segment) generated by Eve. Using the encrypted field from the second message of the initial exchange as a ticket, Eve then follows the subsequent authentication part of the protocol to authenticate herself as Alice to Bob:

(1′) $E_a \rightarrow B$: $N_a', \{A, K_{ab} (= T_b', N), T_b\}_{K_{bs}}$
(2′) $B \rightarrow E_a$: $N_b', \{N_a'\}_{K_{ab}}$
(3′) $E_a \rightarrow B$: $\{N_b'\}_{K_{ab}}$

This attack works as long as T_b has not expired, which should not occur for a while since it is designed to be good for repeated authentications. (Of course this attack also relies on assumptions about implementation similar to those for the attacks on NS.) Note that while T_b' does not appear as cleartext it is easily predicted by the penetrator, all the more so since she has T_b. Thus, she can obtain K_{ab} through a combination of known and predictable plaintext. (Even if she could not obtain T_b via a correct initial exchange, it should also be fairly predictable.)

It may thus seem unclear how to construct a protocol that is both no more expensive in computation or communication than NS and also free of extra security assumptions at the implementation level. However, we can combine features from both KSL and NS so as to do just that. The feature of KSL that makes such attacks so difficult to find is that the ticket is encrypted using a key exclusively for that purpose (rather than one that is used for communication between a principal and the server). In KSL Bob uses his ticket key to produce the ticket for Alice. If we allow the server to produce the ticket instead, then we can reduce the number of messages in the initial exchange.

Ticket key protocol[9]

(1) $A \rightarrow B$: A, N_a
(2) $B \rightarrow S$: $B, \{A, N_a, T_b\}_{K_{bs}}, N_b$
(3) $S \rightarrow A$: $\{B, N_a, K_{ab}, T_b\}_{K_{as}}, \{A, K_{ab}, T_b\}_{K_{bb}}, N_b$
(4) $A \rightarrow B$: $\{A, K_{ab}, T_b\}_{K_{bb}}, \{N_b\}_{K_{ab}}$

---

[7] Martín Abadi proposed this to me as a potential general way of dealing with causal consistency attacks.

[8] Of course we must still make the standard assumptions that encryption is not broken via direct cryptanalysis, honest principals do not broadcast secret keys, etc.

[9] The subsequent exchange has virtually the same form as that of KSL and NS. We do not bother to reproduce it here.

Here, following Kehne et al., we use 'K_{bb}' to represent a key used exclusively to produce a ticket to be checked only by Bob. Not following Kehne et al., the ticket key is assumed to be known to the server as well as to Bob. However, the server is expected to use it only for this purpose. And, Bob is expected to be able to detect the error should he receive either a putative ticket encrypted with K_{bs} or a non-ticket encrypted with K_{bb}. This protocol does require that the server store two keys for each principal. But, except for this storage, the expense is identical to that of NS, and it does not require the additional security assumptions on implementations that NS does.

Logical Analysis

Both KSL and NS were analyzed by their original authors using BAN logic. According to Neuman and Stubblebine, the NS protocol was evaluated using BAN to show that both principals have adequate belief in the goodness of the distributed key to meet functional requirements. And, there are no apparent errors in their analysis. The inability of BAN to represent such flaws was discussed in [Sne92] and [Syv93]. Snekkenes used Bieber's logic CKT5 ([Bie89], [Bie90]) to analyze such flaws while [Syv93] used a temporal extension of Abadi and Tuttle's version of BAN. [AT91] Very recently Carlsen has built on the work of Bieber and Snekkenes and applied CKT5 to the implementation dependent flaws of NS discussed herein. [Car93] We will not here generally analyze the advantages and limitations of BAN, which have been the subject of much previous discussion. (In addition to the papers just mentioned, cf., e.g., [GNY90], [GKSG91], [Nes90], [BAN90], [Sne91], [Syv91], [Syv92].)

We will also not rehash the details of the analyses by Kehne et al. or Neuman and Stubblebine; we will focus primarily on their results. These logical results help shed light on an apparent discrepancy between [KSL92] and [NS93], viz: Neuman and Stubblebine are able to produce a protocol for repeated authentication requiring only four messages despite the analysis in [KSL92] showing that the five message initial exchange of KSL is minimal. Kehne et al. derived the following conclusions concerning their protocol:

1. A believes $A \xleftrightarrow{K_{ab}} B$.
2. B believes A believes $A \xleftrightarrow{K_{ab}} B$.
3. B believes $A \xleftrightarrow{K_{ab}} B$.
4. A believes B believes $A \xleftrightarrow{K_{ab}} B$.

'$A \xleftrightarrow{K_{ab}} B$' means that K_{ab} is a good key for A to speak with B and vice versa. No one other than A or B will ever encrypt messages using K_{ab}. The meaning of the formalisms of BAN is discussed in [BAN89] and they are given a precise model-theoretic semantics in [AT91].

Kehne et al. state the assumptions under which their protocol is minimal, including that "the formalized goals of authentication stated above have to be deducible". Neuman and Stubblebine do not produce a protocol in which the above results are obtained via a four message initial exchange. Rather they claim that the goals in question are not necessary to meet the "functional objectives" of the protocol. It is true that their protocol meets the functional objectives they state with only four messages; however, they simply state that these are also the objectives of KSL. KSL also supports the objective that Bob know's Alice has the session key while NS does not (nor do the permuted or ticket key protocols). This objective corresponds to the fourth logical result above. Since Kehne et al. do not put things specifically in terms of functional objectives it is difficult to evaluate their intentions in those terms. They did not say whether or not they meant to include such an objective. However, without it their optimality analysis is spurious. Therefore, it is uncharitable at best to assume for them that this is not an objective of their protocol.

Neuman and Stubblebine also claim that the logical derivation by Kehne et al. of results regarding the subsequent authentication protocol are incorrect. This is because the derivation depends on Bob's belief in the freshness of T_b.[10] Neuman and Stubblebine's evidence is the account of freshness given in [BAN89]. Therein something is called fresh if it has not been sent in a message at any time before the current run of the protocol. But, in a subsequent authentication run, T_b would have been used previously, at least during the initial exchange. Thus, they conclude that the derivation (with respect to subsequent authentication) that Bob believes K_{ab} is a good key relies on a spurious assumption.

While this is perhaps one reasonable interpretation of BAN in application to KSL, it is not the only one. Thus, the derivation by Kehne et al. need not be considered "incorrect" as Neuman and Stubblebine claimed. For, there is nothing to preclude interpreting 'current run of the protocol' as beginning with the first message of the initial exchange. This leaves the termination of the current run open ended, but in practice that is equally true of a standard protocol without any such reauthentication options. It may take a very long time to proceed from the first

---

[10] According to Kehne et al., in KSL, 'T_b' refers to a timestamp. In NS it is considered to be the expiration time of the ticket. We will see presently that there is an important tension underlying the choice of terminology.

message to the last. Such a protocol generally has a specific final message, but there is nothing in the basic description to rule out a month passing from the time the session key is initially sent out until the final message is sent. Threats that this might engender are generally dealt with by some sort of timeout feature, and in the KSL protocol this is explicitly incorporated into the use of the ticket: in subsequent authentication Bob checks the timestamp to determine whether the current run of the protocol has timed out. These sorts of debates over the meaning of formal expressions are commonplace in BAN analyses. As a rule of thumb, BAN is probably the most easy to use tool around that sometimes uncovers previously unnoticed features and/or flaws in protocols, but one should be excruciatingly cautious in attaching implications for security to any positive results one derives with it.

Nonetheless, reauthentication poses questions that were not in the scope of consideration of the original BAN authors. And, the difference between the above interpretation and that by Neuman and Stubblebine brings this out clearly. Even though under the above interpretation, there is nothing wrong with the BAN analysis of KSL by Kehne et al., there is a tension between this interpretation and the need for reauthentication. For, if freshness is bounded only by the beginning of the initial exchange, then there is no need for reauthentication. As Neuman and Stubblebine put it, there is "an inconsistency between the protocol description and the idealization of the protocol to obtain the following assumption about the freshness of the timestamp: B believes fresh T_b." However, Bob's belief that K_{ab} is good is clearly determined by whether or not T_b has expired. And, this sounds suspiciously like a freshness concern. Neuman and Stubblebine rely on extralogical reasoning to determine that, because T_b has not expired, K_{ab} is still good . Kehne et al. attempt to justify this logically using BAN, but they can only do so at the expense of rendering reauthentication seemingly superfluous. What is needed is the capability to represent relative freshness. In this way one could reason logically that, because of T_b's freshness with respect to the initial exchange, Bob considers K_{ab} to be good. But, because T_b is not considered fresh with respect to the subsequent exchange, more is needed to authenticate Alice to Bob. Both, Bieber's CKT5 [Bie90] and my temporal version of AT [Syv93], seem to be capable of expressing these subtleties, but the details are beyond the scope of this paper.

4 Conclusions

We have looked at a number of protocols for key exchange with repeated authentication in this paper. We have also seen that BAN cannot clearly be applied in general to such protocols because these may require differentiation of relative freshness. (This is no criticism of BAN since it was not designed to apply to such protocols.) Each of the protocols we looked at has certain advantages over the others. KSL attains objectives not attainable by the others. NS attains a reasonable set of objectives with fewer messages than KSL, as do the permuted and ticket key protocols. The permuted protocol is subject to different attacks than NS; hence, it may rely on different assumptions about the security provided at other levels of implementation. The ticket key protocol has the same computational and communication expense as NS but does not have all the insecure implementations that NS does. Nonetheless, unlike NS, it requires that each principal have two keys held by him and the server. Which of these features is most important is ultimately up to the protocol implementor.

Acknowledgements

I have thanked various people at specific points throughout the paper. These people are also generally deserving of thanks for helpful discussions on the topics touched herein. I also thank Jim Gray, Cathy Meadows, Cliff Neuman, and Stuart Stubblebine for comments on a draft of this paper.

References

[AT91] Martín Abadi and Mark R. Tuttle. A Semantics for a Logic of Authentication. In *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*, pages 201–216. ACM Press, August 1991.

[BAN89] Michael Burrows, Martín Abadi, and Roger Needham. A Logic of Authentication. Research Report 39, Digital Systems Research Center, February 1989. Parts and versions of this material have been presented in many places including *Transactions on Computer Systems*, 8(1): 18–36, Feb. 1990, and *Proceedings of the Royal Society of London A*, 426: 233–271, 1989. All references herein are to the SRC Research Report 39 as revised Feb. 22, 1990.

[BAN90] Michael Burrows, Martín Abadi, and Roger Needham. Rejoinder to Nessett. *Operating Systems Review*, 24(2):39–40, April 1990.

[BGH+92] Ray Bird, Inder Gopal, Amir Herzberg, Phil Janson, Shay Kutten, Refik Molva, and Moti Yung. Systematic Design of Two-Party Authentication Protocols. In Joan Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, 1992.

[Bie89] Pierre Bieber. *Aspects Epistémiques des Protocoles Cryptographiques*. PhD thesis, Université Paul-Sabatier de Toulouse, October 1989.

[Bie90] Pierre Bieber. A Logic of Communication in a Hostile Environment. In *Proceedings of the Computer Security Foundations Workshop III*, pages 14–22. IEEE Computer Society Press, Los Alamitos, California, 1990.

[Car93] Ulf Carlsen. Using Logics to Detect Implementation-Dependent Flaws. In *Proceedings of the Annual Computer Security Applications Conference*, 1993. Forthcoming.

[DvOW92] Whitfield Diffie, Paul C. van Oorschot, and Michael J. Wiener. Authentication and Authenticated Key Exchanges. *Designs, Codes, and Cryptography*, 2:107–125, 1992.

[GKSG91] V.D. Gligor, R. Kailar, S. Stubblebine, and L. Gong. Logics for Cryptographic Protocols — Virtues and Limitations. In *Proceedings of the Computer Security Foundations Workshop IV*, pages 219–226. IEEE Computer Society Press, Los Alamitos, California, 1991.

[GNY90] Li Gong, Roger Needham, and Raphael Yahalom. Reasoning about Belief in Cryptographic Protocols. In *Proceedings of the 1990 IEEE Symposium on Security and Privacy*, pages 234–248. IEEE Computer Society Press, Los Alamitos, California, 1990.

[KSL92] Kehne, Schönwälder, and Langendörfer. A Nonce-Based Protocol for Multiple Authentications. *Operating Systems Review*, 26(4):84–89, October 1992.

[Nes90] Dan M. Nessett. A Critique of the Burrows, Abadi, and Needham Logic. *Operating Systems Review*, 24(2):35–38, April 1990.

[NS78] Roger M. Needham and Michael D. Schroeder. Using Encryption for Authentication in Large Networks of Computers. *Communications of the ACM*, 21(12):993–999, December 1978.

[NS93] B. Clifford Neuman and Stuart G. Stubblebine. A Note on the Use of Timestamps as Nonces. *Operating Systems Review*, 27(2):10–14, April 1993.

[SM93] Paul Syverson and Catherine Meadows. A Logical Language for Specifying Cryptographic Protocol Requirements. In *Proceedings of the 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 165–177. IEEE Computer Society Press, Los Alamitos, California, 1993.

[Sne91] Einar Snekkenes. Exploring the BAN Approach to Protocol Analysis. In *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 171–181. IEEE Computer Society Press, Los Alamitos, California, 1991.

[Sne92] Einar Snekkenes. Roles in Cryptographic Protocols. In *Proceedings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 105–119. IEEE Computer Society Press, Los Alamitos, California, 1992.

[Syv91] Paul F. Syverson. The Use of Logic in the Analysis of Cryptographic Protocols. In *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 156–170. IEEE Computer Society Press, Los Alamitos, California, 1991.

[Syv92] Paul F. Syverson. Knowledge, Belief, and Semantics in the Analysis of Cryptographic Protocols. *Journal of Computer Security*, 1(3):317–334, 1992.

[Syv93] Paul F. Syverson. Adding Time to a Logic of Authentication. In *Proceedings of the First ACM Conference on Computer and Communications Security*, 1993.

[Yah] Raphael Yahalom. Optimality of Asynchronous 2-Party Secure Data-Exchange Protocols. *Journal of Computer Security*. Forthcoming.