THE USE OF BELIEF LOGICS IN THE PRESENCE OF CAUSAL CONSISTENCY ATTACKS*

Jim Alves-Foss
Laboratory for Applied Logic
Department of Computer Science
University of Idaho
Moscow, ID 83844-1010 USA
jimaf@cs.uidaho.edu

Abstract

This paper discusses a class of attacks against cryptographic protocols that have not been previously representable using BAN-style logics. This problem has resulted in the generation of *proofs* of these protocols that validate final beliefs of the protocol participants even when successful attacks against these protocols have been demonstrated. The failings of the BAN-style proofs of these protocols does not arise from failings of the logics, as alluded to by others, but rather in the use of the logics. This paper looks at the Needham Schroeder public key protocol which has recently been demonstrated flawed, the analysis of the protocol, where the use of the logic failed in the proofs of this protocol and suggests a specific approach for using the logics that avoids these problems.

Keywords: Belief Logics, Authentication Protocols, BAN

Introduction

This paper discusses a class of attacks against cryptographic protocols that have not been previously representable using BAN-style logics [4, 5]. This problem has resulted in the generation of *proofs* of these protocols that validate final beliefs of the protocol participants even when successful attacks against these protocols have been demonstrated. This has brought forth questions about the validity of BAN-style proofs.

There have been several papers describing this class of attacks, including [3, 6, 13, 15]. The common theme in these attacks is the existence of an intruder inserted in the middle of the protocol exchange who uses information from entities on both sides to either obtain shared secret information or to masquerade as one of the entities. This paper looks at the Needham Schroeder (NS) public key protocol [11] which has recently been demonstrated flawed [9, 10], the analysis of the protocol, where the use of the logic failed in the proofs of this protocol and suggests a specific approach for using the logics that avoids these problems.

The realization that proofs using BAN-style logics have failed to successfully guard against this class of attack has been discussed in the literature including in [10, 13, 14, 15]. Syverson [15] calls such attacks, *causal consistency attacks* since they arise when the participant's beliefs in the causality of events is not consistent with the reality of the events. Approaches to overcoming the problems at the implementation level have been discussed by Gong and Syverson in [8] while the use of formal logics with respect to this

*Project sponsored by the National Security Agency under Grant Number MDA904-96-1-0108. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation hereon.

Message 1	$A \rightarrow S:$	A, B
Message 2	$S \rightarrow A:$	$\{K_b, B\}_{K_s^{-1}}$
Message 3	$A \rightarrow B:$	$\{N_a, A\}_{K_b}$
Message 4	$B \rightarrow S:$	B, A
Message 5	$S \rightarrow B:$	$\{K_a, A\}_{K_s^{-1}}$
Message 6	$B \rightarrow A:$	$\{N_a, N_b\}_{K_a}$
Message 7	$A \rightarrow B:$	$\{N_b\}_{K_b}$

Figure 1: Needham Schroeder Public Key Protocol.

class of attack have been discussed in [10, 13, 15]. Unlike the approach presented in this paper, these formal approaches focus on semantic analysis of the system or on the use of model checkers.

In this paper we demonstrate that the failings of the BAN-style proofs of these protocols does not arise from failings of the logics, but rather in the use of the logics. The validity of this demonstration is strengthened by the argument presented by Syverson [15]. Syverson discusses a cryptographic protocol that is subject to a causal consistency attack yet can be proven correct using a BAN-style logic, AT. The AT logic has been proven sound with respect to a semantic model [2], a model which Syverson successfully argues is rich enough to reason about such attacks. If the logic is sound with respect to such a semantic model, then the faulty proofs must be the result of either invalid initial assumptions or invalid application of the logic and not the result of inherent failings of the logics. This paper discusses where the use of the logic failed in the proofs of these protocols and suggests a specific approach for using the logics that avoids these problems. Note, we only discuss the syntactic analysis of the protocol, as is common in these logics, and defer the discussion of semantic analysis to another paper.

The remainder of this paper assumes a working knowledge of cryptography and cryptographic protocols in terms of shared-key and public-key systems, session keys, nonces, and eavesdropping. The paper is outlined as follows. First we provide a brief discussion of the *proven* NS protocol and a causal consistency attack against it. We then provide a brief discussion of BAN-style logics, followed by a discussion of the failed formal analysis of the NS protocol using the BAN logic and why it failed. Next, we present a specific approach for the use of the BAN logic that avoids these problems and demonstrate its use on the NS protocol. Finally we present some conclusions and discuss limitations of our approach and potential future investigations.

Susceptible Cryptographic Protocols

In this section we present the NS protocol and a causal consistency attack against it. In this discussion we use K to denote keys, N to denote nonces, A, B and S to denote valid participants in the protocol (where S is a trusted server) and E to denote the intruder with E_x indicating E masquerading as x. To follow tradition we will use the names Alice, Bob and Eve to refer to A, B, and E, respectively. We use the notation $A \rightarrow B : X$ to indicate that A is sending the specified message to B; $\{M\}_K$ to indicate the encryption of message M with key K (we do not specify what type of encryption is employed); K_{ab} to indicate a key shared by A and B; and K_a to indicate a public key for A with K_a^{-1} as the corresponding private key.

Needham-Schroeder Public-Key Protocol (NS)

In their original paper, Needham and Schroeder proposed a public-key protocol that allows the participants to exchange two independent, secret numbers [11] (this protocol should not be confused with their more famous shared key protocol). The protocol progresses as shown in Figure 1. Alice and Bob communicate with a trusted server S to obtain each other's public keys. Alice sends a nonce, N_a to Bob, Bob sends a nonce N_b to Alice along with N_a, and Alice responds with N_b. This is suppose to permit both parties to be assured that they are currently communicating with the other party with no tampering. The nonces can subsequently be used to generate a shared key.

Message 1	$A \rightarrow S:$	A, E	
Message 2	$S \rightarrow A:$	$\{K_e, E\}_{K_s^{-1}}$	
Message 3	$A \rightarrow E:$	$\{N_a, A\}_{K_e}$	
	Message 1'	$E \rightarrow S:$	E, B
	Message 2'	$S \rightarrow E:$	$\{K_b, B\}_{K_s^{-1}}$
	Message 3'	$E_a \rightarrow B:$	$\{N_a, A\}_{K_b}$
	Message 4'	$B \rightarrow S:$	B, A
	Message 5'	$S \rightarrow B:$	$\{K_a, A\}_{K_s^{-1}}$
	Message 6'	$B \rightarrow E_a:$	$\{N_a, N_b\}_{K_a}$
Message 4		*Ignored by E*	
Message 5		*Ignored by E*	
Message 6	$E \rightarrow A:$	$\{N_a, N_b\}_{K_a}$	
Message 7	$A \rightarrow E:$	$\{N_b\}_{K_e}$	
	Message 7'	$E_a \rightarrow B:$	$\{N_b\}_{K_b}$

Figure 2: An Attack on the NS protocol

Lowe [9] found a flaw in the protocol using a CSP-based tool, FDR. The attack compromises NS using a man-in-the-middle attack. Meadows [10] independently verified this attack using the NRL tool. We follow Meadows' presentation of the attack in Figure 2. In this attack, Eve waits until Alice attempts to establish valid communication with Eve. Eve then uses information from this session (Alice's secret number) to begin a session with Bob, claiming to be Alice. Eve forwards Bob's response (including his secret number) to Alice as Eve's own response. Alice returns the secret number to Eve as a final verification check. Unfortunately, this response is Bob's secret number which Eve can now use to fool Bob into believing that Eve is actually Alice. In other words, Eve tricked Alice into decrypting Bob's secret and giving it back to Eve. Lowe suggests changing message 6 from $\{N_a, N_b\}_{K_a}$ to $\{N_a, N_b, B\}_{K_a}$ so that the originator of the message is not ambiguous.

Causal Consistency Attacks

In general, the problem with protocols susceptible to causal consistency attacks is that they allow the existence of ambiguity in the encrypted messages. An intruder is able to utilize these ambiguities to replay messages from the same, or different, runs of the protocol, thus giving the impression that they created a message for which they do not know the appropriate secrets or keys.

Approaches to avoiding this type of attack have been discussed in the literature [8, 15, 16]. They include the use of direction bits (which indicate the direction of the message to prevent reflection of a message back to the sender), encrypted participant fields (as used above), strict type checking (to avoid the use of a nonce or other value in place of a key), protocol step encoding (inclusion of the protocol name and step number in each encrypted message), and unique protocol sequence numbers. Note that the encryption of participant fields and the use of unique sequence numbers are the only techniques that work to prevent the above attack on the NS protocol, and unique protocol sequence numbers require a more complex implementation. In the following sections we discuss the use of formal logics in the analysis of this protocol with and without the encrypted participant fields.

Belief Logics

In the late 1980s, Burrows, Abadi and Needham [4, 5] introduced a methodology for the formal analysis of cryptographic protocols, now known as the BAN logic. The intent of the logic is to provide a mechanism for protocol developers to determine the beliefs held by participants throughout a protocol run. At the end of an analysis, we know not only what beliefs are held by the participants, but on what initial assumptions

these beliefs are based. This BAN-style reasoning has given rise to other belief logics based on the same premises. We discuss BAN and these other logics in the remainder of this section.

Notation

In addition to the cryptographic notation mentioned previously, belief logics introduce other symbols and notations. We will follow the notation of [5] in this discussion (this notation is very similar to the notation used by most authors). The following is a discussion of a subset of the notation used in BAN that is sufficient for the discussions in this paper.

- $A \overset{K_{ab}}{\leftrightarrow} B$ is used to indicate that the key K_{ab} is a shared key between participants A and B; as before, $\{M\}_{K_{ab}}$ is used to denote encryption of the message M with the key K_{ab}. We assume that this is a valid key that can not be determined by any other participant.
- $\overset{K_a}{\mapsto} A$ is used to indicate that the key K_a is a public key for A. We assume that this is a good public key and that the matching private key K_a^{-1} can not be determined by any other participant.
- $A \overset{S}{\rightleftharpoons} B$ is used to indicate that the value S is a shared secret between participants A and B; $\langle M \rangle_S$ is used to denote composition of the message M with the secret S. This composition typically involves pairing of M and S within an encrypted message.
- A **believes** F is used to indicate that participant A believes in the validity of F.
- A **said** F is used to indicate that participant A sent the message F. The term **says** is used to indicate that the message was sent in the current protocol run (i.e., it is fresh).
- A **sees** F is used to indicate that participant A received the message F and can read and repeat it (possibly after decryption).
- A **controls** F is used to indicate that participant A is a trusted authority with respect to the message F; such that if we can be sure that A recently said F, then we can believe F to be valid. For example, S **controls** $\overset{K}{\mapsto} B$ indicates that S is trusted to distributed public keys for B.
- **fresh**(F) is used to indicate that the message F was recently (within the current protocol run) generated.

The BAN Logic

In the BAN logic, there exist several axioms and rules of inference that allow us to combine the assumptions of the protocol into statements about participants beliefs. Although there are several inference rules, we outline only those most relevant to our discussion.

1. *Message-meaning rules.* These rules are used to help in the interpretation of messages

$$\frac{P \textbf{ believes } Q \overset{K}{\leftrightarrow} P,\ P \textbf{ sees } \{X\}_K}{P \textbf{ believes } Q \textbf{ said } X} \qquad \frac{P \textbf{ believes } \overset{K}{\mapsto} Q,\ P \textbf{ sees } \{X\}_{K^{-1}}}{P \textbf{ believes } Q \textbf{ said } X}$$

$$\frac{P \textbf{ believes } Q \overset{S}{\rightleftharpoons} P,\ P \textbf{ sees } \langle X \rangle_S}{P \textbf{ believes } Q \textbf{ said } X}$$

2. *Nonce-verification rule.* This is used to determine senders current beliefs. (Note we have to assume that a participant will only send what it believes to be true).

$$\frac{P \textbf{ believes fresh}(X),\ P \textbf{ believes } Q \textbf{ said } X}{P \textbf{ believes } Q \textbf{ believes } X}$$

3. *Jurisdiction Rules.* This is used to transfer belief between participants.

$$\frac{P \textbf{ believes } Q \textbf{ controls}(X),\ P \textbf{ believes } Q \textbf{ believes } X}{P \textbf{ believes } X}$$

Other BAN-style Logics

Since the publishing of the BAN logics there have been proposed extensions and modifications of the logics. The most notable of these are the logics we denote as GNY, AT, and SVO. In the analysis in the following section we stick to the BAN logic, since it is the logic used in the original analysis. A mapping of the proof to another logic such as SVO or GNY would result in the same beliefs and invalid proof (assuming an extension of SVO that allows shared secrets). For completeness, we provide a brief discussion of each of these other logics.

GNY The GNY logic, as presented by Gong, Needham and Yahalom [7], modifies the BAN logic to separate what is possessed from what is believed and defines a notion of recognizability. This enables participants to forward messages that they possess, but not necessarily believe; enabling the analysis of a wider class of protocols.

AT The AT logic, as presented by Abadi and Tuttle [2], provides a subset of the BAN logic (for example, it does not provide a mechanism for reasoning about public keys) that encorporates the separation of belief and possession as developed in GNY. In addition, the AT logic includes a formal semantics for the logic, against which the inference rules are proved sound.

SVO The SVO logic, as presented by Syverson and van Oorschot [18], follows an approach very similar to the AT logic but also enables reasoning about protocols that encorporate public keys and requires a discussion of message comprehension. However, it does not provide rules for shared secrets, although this is a straight-forward extension of the logic. SVO also encorporates a formal semantics of the logic.

Protocol Analysis

The analysis of a cryptographic protocol, using any of the above logics, requires a sequence of four steps. These steps are:

1. *Idealization of the protocol.* This involves transformation of the protocol from the standard notation (such as that used in Figure 1) to a more formalized notation. In [4], idealization involves removal of all plain text components of a message, transformation of keys into statements of the intended use of the key (e.g., $A \overset{K_{ab}}{\leftrightarrow} B$), explicit statements relating to beliefs (e.g., A **believes** $A \overset{K_{ab}}{\leftrightarrow} B$), and composition with shared secrets (e.g., $\langle A \overset{N_b}{\rightleftharpoons} B \rangle_{N_a}$). A recent paper by Syverson [17] suggests abandoning the idealization step and instead increasing the set of initial assumptions. This is similar to the approach we take in the next section and further discussion is deferred until then.

2. *State initial assumptions.* At the beginning of each protocol run there are a set of initial assumptions that are made by participants, such as beliefs in the validity of a shared key, or the freshness of generated nonces. Each of these assumptions must be explicitly stated.

3. *Annotation of the protocol.* This involves enumerating, for each protocol step, the changes in the state of the system. For example, after $A \rightarrow B : X$, we can infer that B **sees** X.

4. *Analysis of beliefs.* After completion of the above steps we have a set of assumptions (both initial and those following each protocol step). We now use these assumptions and the inference rules from the logic to develop conclusions related to participants beliefs.

The following provides an outline of this style of analysis for the NS protocol as defined in Figure 1. Recall, that although we are using the BAN logic, similar approaches and results can be obtained using other logics. However, this would require some minor modifications to the logics. As presented in the literature, AT can not reason about the use of public keys in NS and SVO can not reason about the use of shared keys in NS.

Original Analysis of NS

Details of this analysis are provided in [4], so we will only outline them here due to space constraints.

1. *Protocol Idealization.* Burrows, et.al. provide the following idealization of the protocol:

   Message 2 $S \rightarrow A : \quad \{\overset{K_b}{\mapsto} B\}_{K_s^{-1}}$
   Message 3 $A \rightarrow B : \quad \{N_A\}_{K_b}$
   Message 5 $S \rightarrow B : \quad \{\overset{K_a}{\mapsto} A\}_{K_s^{-1}}$
   Message 6 $B \rightarrow A : \quad \{\langle A \overset{N_b}{\rightleftharpoons} B\rangle_{N_a}\}_{K_a}$
   Message 7 $A \rightarrow B : \quad \{\langle A \overset{N_a}{\rightleftharpoons} B, B \textbf{ believes } (A \overset{N_b}{\rightleftharpoons} B)\rangle_{N_b}\}_{K_b}$

   This is not the only possible idealization, but it is consistent with techniques demonstrated in the literature. Additional fields could be added to the idealization to enforce conditions on the message transport (see [7] for further explanation) or to indicate that a previous participant said something. Since these additional fields are not necessary for the proof outlined below we do not use them[1]. Note the differences between the idealization and the original protocol description. We have removed all plain text message components, and we have included the notation $A \overset{N_b}{\rightleftharpoons} B$ in message 6 and similar notation in message 7 as well as a field in message 7 about Bob's beliefs. The purpose of these messages is to indicate and transfer beliefs between Alice and Bob. The annotation about Bob's belief will provide Bob with slightly more information which will be demonstrated in the conclusions. Without such notation there is no explicit intent or meaning associated with the protocol messages. Syverson [17] argues that this meaning should be tied to the receiver, not the sender, but still agrees that the notation must be used to indicate meaning. This all arises from the fact that we are performing a pure syntactic analysis, and must syntactically demonstrate meaning.

2. *Initial Assumptions.* The initial assumptions outlined in [4] are:

A **believes** $\overset{K_a}{\mapsto} A$	B **believes** $\overset{K_b}{\mapsto} B$
A **believes** $\overset{K_s}{\mapsto} S$	B **believes** $\overset{K_s}{\mapsto} S$
S **believes** $\overset{K_a}{\mapsto} A$	S **believes** $\overset{K_b}{\mapsto} B$
S **believes** $\overset{K_s}{\mapsto} S$	
A **believes** S **controls** $\overset{K}{\mapsto} B$	B **believes** S **controls** $\overset{K}{\mapsto} A$
A **believes fresh**(N_a)	B **believes fresh**(N_b)
A **believes** $A \overset{N_a}{\rightleftharpoons} B$	B **believes** $A \overset{N_b}{\rightleftharpoons} B$
A **believes fresh**$(\overset{K_b}{\mapsto} B)$	B **believes fresh**$(\overset{K_a}{\mapsto} A)$

[1] Although one could argue that the message extensions of GNY would be useful here, we will defer discussion of these to a later Section of this paper.

These assumptions highlight participants beliefs in their public keys and freshness of shared secrets and the server's ability to distribute public keys.

3. *Annotations.* Specific annotations are not demonstrated in [4], but consist only of adding the assumptions that recipients can **see** the messages sent to them.

4. *Proof.* Through uses of message meaning, nonce-verification and jurisdiction, the participants end up with a final set of beliefs containing:

   A **believes** $\overset{K_b}{\mapsto} B$ $\qquad$ B **believes** $\overset{K_a}{\mapsto} A$

   A **believes** B **believes** $A \overset{N_b}{\rightleftharpoons} B$ $\qquad$ B **believes** A **believes** $A \overset{N_a}{\rightleftharpoons} B$

   B **believes** A **believes** B **believes** $A \overset{N_b}{\rightleftharpoons} B$

Note that the final belief is the result of message 7, where Bob obtains information about Alices's beliefs regarding Bob's beliefs. We have thus obtained a set of participant's beliefs that are in contradiction with demonstrated attacks on the system. In the following section we discuss how such faulty proofs have come about and what we can do to avoid them.

A New Approach

If we look back at the attack on NS and on the proof of the protocol, it becomes apparent where the problem occurs. The attack on NS took message 6 from one protocol run and used it as message 6 in another run. The real and idealized message 6 from the runs are:

Real Message 6 (NS)	$B \rightarrow A:$	$\{N_a, N_b\}_{K_a}$
Idealized Message 6 (NS)	$B \rightarrow A:$	$\{\langle A \overset{N_b}{\rightleftharpoons} B\rangle_{N_a}\}_{K_a}$

Notice that this idealization involves the injection of a statement about validity of a shared secret into the message. The problem occurs where upon receipt of this message the recipient believes that the sender believes in the validity of the shared secret. However, in the actual protocol, there is nothing that specifically justifies this belief. Thus it is the idealization that is wrong, and not the logic. The logic correctly validated the idealized protocol, but the idealized protocol is not consistent with the original protocol.

In NS, the situation is easy to explain and remedy. The message sent from the intruder, Eve, to Alice is actually message 6 of the protocol sent from Bob to Eve masquerading as Alice. The problem is that in the attack, Alice decrypts message 6, believing it to be sent from Eve and responds with the nonce N_b that Eve can then use as a response to Bob. This occurs because upon receipt of message 6, Alice believes that the message states that Eve believes that N_b is a valid shared secret for use between Alice and Eve; when in fact the message stated that Bob believes that this is a valid shared secret between Alice and Bob. The original message contained no reference to Bob, in violation of Abadi and Needham's principle 3 [1], which states that if the identity of a principal is essential to the meaning of a message, then it should be specifically encoded in the message. The lack of a reference to the source allows Eve to forward the message to Alice and then take Alice's response to convince Bob that Eve is in fact Alice.

In other words, the idealization of the protocol is incorrect; since we can not be sure that the message received by Alice as message 6 is actually Bob's validation of N_a and N_b as shared keys between Alice and Bob. Using this idealization forces us to assume that this can be detected by Alice and allows the verification to proceed. The problems that arise from the analysis of this and other protocols can be summarized as follows (the first two of which are addressed in [1, 8]):

- Sufficient information must be encoded in a message to determine the intended source, recipient and other participants relevant to the protocol step.
- Sufficient information must be encoded in a message to distinguish it from other messages in the protocol run.

- Annotations of the protocol should specifically state the conditions that must be met before a protocol step is taken. This is similar to the use of message extensions of GNY[2]. A similar approach, termed faithfulness requirements, is presented in [15], but is used in terms of the semantics of the logic and not with the annotations or protocol idealization. Syverson also presents another approach in [17] that we discuss later.

What is needed is an approach to using the logic that specifically addresses these problems. The following section presents such an approach and then demonstrates how it can be applied to the NS protocol.

The Approach

In this section we present an approach for analyzing of cryptographic protocols using BAN-style logics which avoids the problems outline above. Specifically, we suggest changes to the annotation and protocol idealization phases of the analysis so that we may more accurately represent the assumptions made by the participants and the encoding of those assumptions in the protocol messages. There are two parts of this approach, the first consists of the idealization of the protocol, the second consists of adding inference rules in place of standard annotations suggested in [4].

- *Idealization.* The idealization of the protocol involves the following:
  1. Remove all plain text components of the messages.
  2. Insert $A \overset{K_{a^b}}{\leftrightarrow} B$ in the message in place of the key K_{ab} only if
     (a) For public-key encryption, A and B are explicitly stated in the encrypted portion of the message, and the message explicitly indicates who is sending the message (A, B or S). Note: if B is sending the message to A using A's public key, A's name need not be specifically included in the body of the encryption.
     (b) For shared-key encryption, A and B are explicitly stated in the encrypted portion of the message (with the same exception as above).
     (c) There exists a precise mechanism for determining that the intent of the sender is to use this field of the message as a key. This could be a specific encoding in the message, type enforcement, inclusion of a shared secret, or uniqueness of the message format. If two messages in the protocol are encrypted by the same sender, using the same key, with the same number of fields, there must be a secure mechanism in place for the recipient to differentiate the messages (note that the use of plain text identifiers is not sufficient).
  3. Insert $A \overset{S}{\rightleftharpoons} B$ in the message in place of the field S only if the same conditions for shared key insertion hold.
  4. Insert A **believes** X or other formula of the logic not specifically tied to a field of the message into the message only if the sender believes in the truth of this statement (e.g., in the inference rules discussed below, a precondition to sending this message must be that the sender believes that A **believes** X). Syverson [17] argues that only the recipient should be able to assert these beliefs and that these assertions should only occur in the premises of the protocol, and not in the messages of the protocol. This is due to the fact that it is the recipient that is determining its own beliefs about other's beliefs. Although this is a reasonable point, Syverson does not explicitly state any constraints on the premises similar to those we have stated here.
- *Annotation.* The annotation of the protocol involves the following:
  1. For each step of the protocol determine precisely what needs to happen before the sending of a message. In other words, highlight what the protocol participant needs to **see** and **believe** before sending a response.

[2] Note that this problem also resolves the issues of out-of-order protocol runs presented in [12], as it ensures that protocol steps are not seen as an unordered list of messages but rather with specific dependencies.

For message 3:

$$\frac{A \textbf{ believes } S \textbf{ said } \{\overset{K_b}{\mapsto} B\}_{K_s^{-1}}}{B \textbf{ sees } \{N_a\}_{K_b}, \quad A \textbf{ believes } A \textbf{ said } \{N_a\}_{K_b}}$$

For message 6:

$$\frac{B \textbf{ sees } \{N_a\}_{K_b}, \qquad B \textbf{ believes } S \textbf{ said } \{\overset{K_a}{\mapsto} A\}_{K_s^{-1}}}{A \textbf{ sees } \{\langle A \overset{N_b}{\rightleftharpoons} B\rangle_{N_a}\}_{K_a}, \quad B \textbf{ believes } B \textbf{ said } \{\langle A \overset{N_b}{\rightleftharpoons} B\rangle_{N_a}\}_{K_a}}$$

For message 7:

$$\frac{\begin{array}{ll} A \textbf{ believes } A \textbf{ said } \{N_a\}_{K_b}, & A \textbf{ believes } S \textbf{ said } \{\overset{K_b}{\mapsto} B\}_{K_s^{-1}} \\ A \textbf{ believes } B \textbf{ said } \{\langle A \overset{N_b}{\rightleftharpoons} B\rangle_{N_a}\}_{K_a}, & A \textbf{ believes } B \textbf{ believes } A \overset{N_b}{\rightleftharpoons} B \end{array}}{\begin{array}{c} B \textbf{ sees } \{\langle A \overset{N_a}{\rightleftharpoons} B, B \textbf{ believes } A \overset{N_b}{\rightleftharpoons} B\rangle_{N_b}\}_{K_b} \\ A \textbf{ believes } A \textbf{ said } \{\langle A \overset{N_a}{\rightleftharpoons} B, B \textbf{ believes } A \overset{N_b}{\rightleftharpoons} B\rangle_{N_b}\}_{K_b} \end{array}}$$

Figure 3: Inference rule annotations for the NS protocol.

2. Annotate each step of the protocol with inference rules that indicate the preconditions of sending a message as premises of the inference, and both the recipients seeing the idealized message and the sender believing they said the message as the conclusion of the inference. This corresponds to explicitly defining the decisions being made by the participants before sending a message, and allows us to specify a participant's beliefs in its own actions[3].

- *Additional Notes.* The following must be taken into consideration when developing protocol assumptions:

1. A key or shared secret can not be believed to be a secret unless it is transmitted encrypted under a key K, where K is not the private key of a public-key pair, and K has not been compromised.
2. A key, or shared secret, is shared by all of those participants who share the key under which it is encrypted.
3. A participant Alice can not believe in the freshness of a nonce, or other value unless Alice created the nonce or a component of a composite message.

Analysis of NS Using the New Approach

We have to modify the physical protocol to enable us to generate the idealized protocol. To do this, we add a reference to B in message 6. This enables us to justify the claim in the idealization that N_b is a shared secret between Alice and Bob. Figure 3 shows the inference rules generated in this protocol analysis.

Some may argue that this style of inference rule is unwieldy and contains too many preconditions. We have tried to remedy this by removing all initial assumptions from the preconditions (e.g., assumptions about nonce freshness, shared keys, control, etc.). Since we have done this, the inference rules for messages 2 and 5 can be considered as axioms, where Alice and Bob see the message from the server containing the other's public key. These axioms are in the same form as BAN annotations. Writing down the inferences in their full form explicitly reveals the assumptions being made by the sender. Good practice involves writing down all the inference rules first, and then create the initial assumptions from the preconditions that are not derived from inference rules. These assumptions can then be removed from the statements of the preconditions as we have done in Figure 3.

[3] Although beyond the scope of this paper, a formal analysis of the soundness of these inference rules would provide stronger evidence for the correctness of the protocol.

The analysis proceeds as defined in [4], except that now we must ensure that all of the preconditions of sending a message are met before we can add the annotation that the recipient sees the message. These preconditions include previous messages sent by the same participant and those received by the participant, as well as other assumptions. We can still derive all the conclusions in the original analysis; but with the actual protocol conditions explicitly defined. Specifically:

- From message 2, message meaning and jurisdiction rules we get:

$$A \textbf{ believes } \stackrel{K_b}{\mapsto} B$$

- From message 3, since Alice has received message 2 from the server, we can use the above conditions for message 3 and allow Bob to decrypt the message to get:

$$B \textbf{ sees } N_a$$

  Note that we use the initial assumptions that N_a is fresh and believed by Alice to be a shared secret with Bob.

- Message 5 is similar to message 2 giving us:

$$B \textbf{ believes } \stackrel{K_a}{\mapsto} A$$

- Message 6 will be sent if Bob has received an encrypted nonce, N_b, and Bob believes that N_b is fresh and a shared secret with Alice. Given the inclusion of N_a in the message, along with Alice's beliefs about the freshness and shared secret status of N_a, Alice can deduce:

$$A \textbf{ believes } B \textbf{ believes } A \stackrel{N_b}{\rightleftharpoons} B$$

- Message 7 can now be sent back to Bob. That is, only if Alice believes she has sent all the correct previous messages in the protocol, and received appropriate responses from Bob, as outline in the above inference rule for message 7. Bob, given his beliefs in the freshness and shared secret status of N_b and in messages received over the current protocol run, can deduce the remaining two beliefs:

$$B \textbf{ believes } A \textbf{ believes } A \stackrel{N_a}{\rightleftharpoons} B$$

$$B \textbf{ believes } A \textbf{ believes } B \textbf{ believes } A \stackrel{N_b}{\rightleftharpoons} B$$

Applying our approach to the NS protocol, with the understanding that message 6 must be physically changed to include a reference to Bob as sender, we have derived the same set of final beliefs as in [4]. This is a correct protocol, as long as the physical representation of the messages correspond to the idealized form of the protocol.

Conclusion

Causal consistency attacks, in the form of man-in-the-middle attacks, against cryptographic protocols have resulted in the breaking of some published protocols. Some of these protocols have been previously shown to be correct with respect to a BAN-style belief logic. The soundness of some of these belief logics (i.e., AT [2] and SVO [18]) has been proven with respect to a formal semantics which is rich enough to reason about this class of attack [15]. It is therefore only logical to assume that it is the use of the logic, and not the logic itself that is at fault.

We have presented an approach for using BAN-style logics that will not incorrectly prove the correctness of protocols susceptible to this type of attack. In addition, our approach also overcomes a limitation of the use of belief logics first presented by Snekkenes [12]. The use of this approach also provides a cleaner format for mapping protocol comments and conditional steps into the annotations.

Our approach helps overcome some of the published limitations of cryptographic protocols, but does not overcome all of them. The inability of these protocols to reason about secrecy, the possibility of implementation dependent flaws being introduced has not been addressed. There are limits to the expressibility of any of the published logics when it comes to actual implementations. Along with formal specification and analysis we recommend that protocol designers also follow the implementation advice in the literature [1, 8]. In addition, we have not provided a formal analysis of the use of this approach with respect to a formal semantics of a belief logic. Although we feel that such an analysis would be useful and could help demonstrate the similarity between this approach and those proposed by Syverson [15, 17], we leave such analysis for future discussion.

Related Work

There is some commonality between our approach and others presented in the literature. The inherent difficulty of the problem domain and the lack of formalism in the specification of the protocols has lead to several investigations in this area. As we mentioned previously several authors, most notably Gong and Syverson [8] have discussed approaches to the development of correct protocol implementations. In addition there has been work related to the logics themselves.

Snekkenes [12] pointed out that proofs of protocol correctness are usually based on a final state of a protocol run. In other words, all assumptions about messages said and beliefs held are established as assumptions prior to the proof, and thus casual relationships between messages may be inadvertently removed. This may lead to proofs of invalid protocols by rearranging the order of messages sent. Our approach requires that dependent messages be sent only if the dependencies have already been established, thus maintaining the causal link between messages. Syverson [15] discusses a different solution to this problem, through the use of a semantic model of the protocol.

Syverson [17] also discusses a new approach to the syntactic analysis of authentication protocols using SVO, an approach which discards the concepts of idealization, in contrast to the original SVO paper [18]. In this approach, only briefly outlined, the user adds assumptions such as:

$$A \textbf{ believes } (B \textbf{ says } \{N_a, N_b\}_{K_a} \supset B \textbf{ believes } \langle A \overset{N_b}{\rightleftharpoons} B \rangle_{N_a})$$

In the original protocol, message 6 is $\{N_a, N_b\}_{K_a}$ which was idealized to $\{\langle A \overset{N_b}{\rightleftharpoons} B \rangle_{N_a}\}_{K_a}$. Syverson avoids this style of idealization through the use of the above assumption. If we extend SVO using the axioms of Abadi and Tuttle [2] to allow for reasoning with shared secrets and use Syverson's new approach we will still have problems. In the new approach, upon receipt of the original message, Alice can reason that Bob sent the message since Alice believes that N_a is a shared secret with Bob. From this conclusion we conclude the Alice believes that Bob believes that N_b is a shared secret. This is exactly the belief that the attack successfully defeats. Our approach prohibits this attack by preventing any use of the notation $\langle A \overset{N_b}{\rightleftharpoons} B \rangle_{N_a}$ unless the identities of Alice and Bob are included in the message. However, using Syverson's approach instead of idealization of the protocol does have a certain appeal and deserves further investigation in conjunction with the work presented here.

References

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

[2] M. Abadi and M Tuttle. A semantics for a logic of authentication. In *Proc. Tenth ACM Symposium on Principles of Distributed Computing*, pages 201–216, August 1991.

[3] R. Bird, I. Gopal, A. Herzberg, P. Jason, S. Kutten, R. Molva, and M. Yung. Systematic design of two-party authentication protocols. In *Advances in Cryptology — CRYPTO '91*, 1991.

[4] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. Technical Report 39, DEC Systems Research Center, Palo Alto, CA, February 1989.

[5] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, February 1990.

[6] W. Diffie, P. van Oorschot, and M. Wiener. Authentication and authenticated key exchanges. *Design Codes and Cryptograhpy*, 2:107–125, 1992.

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

[8] L. Gong and P. Syverson. Fail-stop protocols: An approach to designing secure protocols. In *International Conference on Dependable Computing for Critical Applications*, pages 44–55, 1995.

[9] G. Lowe. Breaking and fixing the Needham-Shroeder public-key protocol using FDR. In *Proc. TACAS*, pages 147–166. Springer-Verlag, 1996.

[10] C Meadows. Analyzing the Needham-Schroeder public key protocol: A comparison of two approaches. In *Proc. ESORICS 96*. Springer Verlag, 1996.

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

[12] E. Snekkenes. Exploring the BAN approach to protocol analysis. In *Proc. IEEE Symposium on Research in Security and Privacy*, pages 171–181, 1991.

[13] E. Snekkenes. Roles in cryptographic protocols. In *Proc. IEEE Symposium on Research in Security and Privacy*, pages 105–119, 1992.

[14] P. Syverson. The use of logic in the analysis of cryptographic protocols. In *Proc. IEEE Symposium on Research in Security and Privacy*, pages 156–170, 1991.

[15] P. Syverson. Adding time to a logic of authentication. In *Proc. First ACM Conference on Computer and Communications Security*, pages 97–101. ACM Press, 1993.

[16] P. Syverson. On key distribution protocols for repeated authentication. *Operating Systems Review*, 27(4):24–30, October 1993.

[17] P. Syverson. A new look at an old protocol. *Operating Systems Review*, 30(3):1–4, July 1996.

[18] P. Syverson and P. van Oorschot. On unifying some cryptographic protocol logics. In *Proc. IEEE Symposium on Research in Security and Privacy*, pages 14–28, 1994.