

On Belief Evolution in Authentication Protocols

Rajashekar Kailar and Virgil D. Gligor

Department of Electrical Engineering
University of Maryland, College Park, MD 20742.

Abstract

Authentication protocols can be viewed from the perspective of the evolution of beliefs within a protocol run. Inference rules which ensue from this perspective are presented. These rules can be used to analyze the protocols which the BAN logic [1] can analyze. Additional protocols that can be analyzed include (1) inter-domain authentication where principals must trust all authentication servers of the domains traversed according to a specific policy, and (2) where trust in the secrecy of the encryption key and belief ordering need to be established despite the lack of jurisdiction [5].

1 Introduction

An important, but not the only, goal of authentication between two parties A and B is that of establishing the following types of beliefs:

A believes $A \xleftrightarrow{K_{ab}} B$ ¹
 B believes $A \xleftrightarrow{K_{ab}} B$ (called the *first-level* beliefs henceforth) and

A believes B believes $A \xleftrightarrow{K_{ab}} B$
 B believes A believes $A \xleftrightarrow{K_{ab}} B$ (called the *second-level* beliefs henceforth).

That is, both the communicating parties (A and B) should obtain the belief that the shared encryption key (K_{ab}) has the required quality (i.e., goodness) and is secret (i.e., it will be known only to them and the parties that they trust); this is the first-level belief in the encryption key quality and secrecy of the two parties. Each of the two parties should also believe that the other party has obtained its first-level belief; this is their second-level belief.

In this paper we present a logic of authentication that (1) can establish key secrecy in inter-domain authentication as required by specific security policies, and (2) can establish first-level beliefs in key privacy despite

lack of key jurisdiction². The motivation for these features of this logic is provided by the discussion of privacy and jurisdiction trust of reference [5]; for the sake of brevity, this motivation is not repeated in this paper. Instead, in this paper we propose a new logic of authentication and illustrate its features “by example.” This logic is developed at the same level of abstraction as that of the BAN logic [1,2]. This means that it shares all the scope limitations outlined in [1,2]. In addition, it requires that all authentication protocols preserve the sequential order of message exchanges during a protocol run. This property facilitates the use of this logic with existing, practical tools for formal specification and verification in the same way as that described in [4].

In the logic presented herein, beliefs evolve within a protocol run in a manner similar to the states (output) of a state machine. That is,

$$\text{Belief} + \text{action} \Rightarrow \text{New Belief}.$$

Each new pertinent action is compared with the most recent related belief, and the old belief is updated. The term “knowledge set” is introduced to refer to the principals in the particular session which share a message content. For the most part, the logic deals with the interpretation of message exchanges in terms of *knowledge sets*.

The rest of the paper is organized as follows. In the next section, we discuss the basic notation and assumptions of the proposed logic. In section 3, we present the logical postulates and inference rules to define this logic. Section 4 extends the logic to analyze protocols where it is necessary to detect (lack of) *jurisdiction*. In section 5, we apply the inference rules to analyze some authentication protocols and compare the analysis with that done using the BAN logic [1] (more examples of the application of the proposed logic are included in the appendix). We discuss some features of the logic in section 6. Some of our conclusions about this logic are discussed in section 7.

¹Familiarity with the logic and notation of [1] is assumed in this paper.

²A principal having designated authority to generate encryption keys is said to have jurisdiction over the key.

2 Basic Notation and Assumptions

The symbolic notation is similar to that used in the BAN logic [1]. We briefly explain the constructs of [1] used in the analysis presented herein:

- $A \equiv F$: A believes F , or is entitled to believe F . A may act as though F is true.
- $A \Rightarrow F$: A has *jurisdiction* over F . A has delegated authority over statement F and should be trusted on this matter.
- $\sharp(F)$: F is *fresh*, in the sense that F has never been sent in a message before the current run of the protocol.
- $A \stackrel{K}{\leftrightarrow} B$: A and B share a key K for communication. This key will not be discovered by any party except A or B , or a party which is trusted by either A or B .
- $\stackrel{K}{\rightarrow} A$: A has K as its *public key*. Only A (or a principal trusted by A) can have the matching private key K^{-1} .

Some additional terminology is introduced for ease of presentation. We associate a *message instance* or *message round* with each message in a protocol. Most parameters in the analysis are also associated with the message round; i.e., parameter P will be represented as a tuple (P, M_i) , meaning that the statement involving parameter P is made at message instance i . In the idealized representation of protocols, each message is represented as

$$\{Message\ round, Sender, Receiver, Content(s)\}$$

The *sender* and *receiver* fields in a message do not necessarily correspond to the sender and recipient fields in the actual message (if any). Instead, they represent the fact that the message has been either (1) signed with the secret key of the principal identified in the *sender* field and encrypted with the public key of the principal identified in the *receiver* field, or (2) encrypted with the conventional (shared secret) key between the sender and the recipient.

We assume that for a given pair of principals P and Q , the message sent by P to Q is either (1) signed with the secret key of P and encrypted with the public key of Q (public key encryption as in [11]), or (2) encrypted with the shared private key between P and Q (conventional encryption).

$$Y \triangleright \{M_k, Y, X, C\}$$

denotes that Y sends message M with content C in round k of this session, to principal X .

$$X \triangleleft \{M_k, Y, X, C\}$$

denotes that X sees message M in round k and knows

that it is sent by Y and also reads the message content(s) C . The i^{th} message is often referred to as M_i in the following discussion. The notion of trust is made explicit by denoting $Trust_R(P, Q)$ to mean that principal P trusts principal Q in the context R . The universal set of all members of the session is denoted by S . The authentication server is denoted by AS .

If a message contains parts of a message previously sent by another principal which the present sender is forwarding (without decryption and re-encryption), the recipient treats these portions of the message separately; i.e., as though they were sent by the original source of that message. For example, in the Kerberos protocol [9], the second and third message exchanges are :

$$\begin{aligned} Message\ 2: & AS \rightarrow A : \{T_s, N_a, L, K_{ab}, B, \{T_s, L, K_{ab}, A\}^{K_{bs}}\}^{K_{as}} \\ Message\ 3: & A \rightarrow B : \{T_s, L, K_{ab}, A\}^{K_{bs}}, \{A, T_a\}^{K_{ab}} \end{aligned}$$

This is idealized (written in our notation) as:

$$\begin{aligned} Message\ 2: & AS \triangleright \{M_2, AS, A, (K_{ab}, T_s, N_a, \{M_2, S, B, (T_s, K_{ab})\})\} \\ Message\ 3: & A \triangleright \{M_2, S, B, (T_s, L, K_{ab}, A)\}, \{M_3, A, B, (T_a, K_{ab})\} \end{aligned}$$

Here, T_s is a timestamp affixed by the authentication server, and L is the lifetime of the key K_{ab} (since L does not contribute to the analysis, it is omitted in the idealized version). N_a is a nonce (usually a random number). K_{ab} is the session key distributed to A and B by AS . K_{as} and K_{bs} are keys that A and B share with AS respectively.

Principals belonging to a session are assumed to be able to distinguish between messages belonging to the session and messages belonging to other sessions. This is essential since without this feature, a principal may have to deal with two or more independent and possibly unrelated message histories, and this may lead to inconsistencies in beliefs. For this reason, we have treated each session separately. The set of beliefs of two different (possibly concurrent) sessions are disjoint.

Messages in a session are assumed not to be re-ordered. This is because authentication protocols are typically deterministic; i.e., given the past message history, the next message from the principal in the session who has received the current message is determined by the protocol specification. The problem posed by lost messages is solved by placing more credibility on second-level beliefs. The second-level beliefs based on responses to challenge messages eliminate the possibility of the messages being lost, hence corroborating the possibly “eager” first-level beliefs based on these challenge messages.

3 Inference Rules

In this section, we present some of the inference rules and logical postulates that define the proposed logic of belief evolution. The symbolic presentation of rules is similar to that in [1]; i.e.,

$$\frac{P ; Q}{R}$$

means that if statements P and Q hold simultaneously, then statement R holds. Since we make message instances explicit, P , Q and R need not all be facts related to the same message instance (in fact, in most cases they are not). Hence the use of ‘;’ denotes *conjunction* and not temporal ordering.

3.1 Belief in the Uniqueness of the Message Recipient

This belief stems from the trust that communicating parties have in the encryption scheme. A similar trust is the basis for message meaning recognition in [1].

If a message is effectively encrypted, then its contents can be accessed only by its intended recipient(s). More formally,

$$\frac{X \triangleright \{M_i, X, Y, C\}; X \models \{Z \triangleleft \{M_i, X, Y, C\}\}}{X \models \{Z = Y\}}$$

That is, if (1) X sends message M_i to Y , and (2) X believes that Z reads the message, Then X believes that $Z = Y$ (i.e., in Y 's uniqueness).

If X intends more than one principal to receive the message, then X obtains the belief that Z is a subset of Y .

Justification for Recipient Uniqueness

The only underlying assumption is that of an effective encryption scheme. If a message is signed by the sender and encrypted with the public key of the recipient, only the recipient can decrypt this message with his private key, even though the ciphertext is accessible to others. Note that the recipient uniqueness holds even for conventional key encryption schemes. If a message is encrypted with the shared key between P and Q , and if P sends the message, then P is entitled to believe that only Q will read the message. The rule does not assume guaranteed communication. The decryption by the intended recipient(s) is conditional on successful communication of the message.

3.2 Knowledge Set (KS) Belief:

Definition: The knowledge set (KS) of a message content a at message instant i is defined as $X \in KS(a, M_i)$

if a is recognizable by X at and after the instance when M_i is received.

That is, principal X belongs to *knowledge set* of message content a at message instance M_i if X can recognize the message content a at and after the instant when message M_i is received. A principal which is a member of a KS will remain a member throughout the protocol run. In other words, the cardinality of a KS increases monotonically during a session. Each principal maintains certain beliefs about the KS of pertinent message contents (e.g., nonces that it generated or believes to be fresh, or session key), based on the messages it has sent or received at or before the current message round. The principal does (not) update its belief about the *knowledge set* of a certain message content based on the contents of the received message.

General form of the KS belief:

All principals which obtain membership to the knowledge set of a message content ‘a’ between instance $i > 0$ and $j > i$ must have received at least one message containing ‘a’ as its content from some member of the session, at some instance k between i and j . This is denoted symbolically as

$$\frac{X \models KS(a, M_i) = \{S_1\}; X \models KS(a, M_j) = \{S_2\} \mid j \geq i > 0}{X \models \forall Y \in \{S_2\} - \{S_1\}; \exists P, k \mid P \in S; i < k \leq j; Y \triangleleft \{M_k, P, Y, a\}}$$

That is, if (1) X believes that $KS(a, M_i) = \{S_1\}$ and (2) X believes that $KS(a, M_j) = \{S_2\}$, then X is entitled to believe that all principals Y in *knowledge set* $\{S_2\} - \{S_1\}$ must have received a message with content a from some principal in S (i.e., the set S of all principals in the session).

Justification:

At message instant M_i , there are two disjoint sets: (1) S' consisting of all principals in the $KS(a, M_i)$ and (2) $S - S'$, consisting of all principals which are not in the $KS(a, M_i)$ (here, S denotes the universal set of all principals belonging to a session). In symbolic notation, $\forall X \in S$, either $X \in S'$ or $X \in \{S - S'\}$. If $X \in \{S - S'\}$ then it can become a member of S' at some instance $k \mid i < k < j$ if and only if X receives a message with content a . This can happen only when it receives a message from some $P \in S'$.

In the general form of *knowledge set belief*, the set S_1 which corresponds to X 's belief about $KS(a, M_i)$, is possibly an improper subset of the actual $KS(a, M_i)$ which is S' . There are some special cases that follow from the general case of *knowledge set belief*. We present them in the following sections.

Set Inclusion Belief (1)

$$\frac{X \notin KS(a, M_i); X \triangleleft \{M_{i+1}, Y, X, a\}}{X \models KS(a, M_{i+1}) = \{X, Y\}}$$

That is, when principal X receives the message content for the first time, it believes that the message content is now shared between the sender and itself. In other words, it believes that the *knowledge set* of the message content after the message is received, consists of the sender and itself.

Set Inclusion Belief (2)

$$\frac{X \models \{Y \notin KS(a, M_i)\}; X \triangleright \{M_{i+1}, X, Y, a\}}{X \models KS(a, M_{i+1}) = KS(a, M_i) \cup \{Y\}}$$

That is, if (1) principal X believes that Y does not belong to $KS(a, M_i)$, and (2) X sends a message M_{i+1} to Y containing a , then X believes that the elements X and Y belong to $KS(a, M_{i+1})$. The belief obtained by principal X when it sends a message to principal Y involves some risk. X assumes that M_{i+1} will eventually be read by only Y (by *message recipient uniqueness*). X believes that the message will not be lost or suppressed (accidentally or otherwise).

A similar type of risk is illustrated in [7], where the authors provide an example of an “eager protocol” for obtaining common knowledge, and argue that common knowledge cannot be achieved in a practical distributed system (where messages can be delayed or lost) without some risk. If ϵ is the expected message transmission time, then the system is in a knowledge inconsistent state during an ϵ time interval, but this state “soon becomes consistent” [7].

If principal X attaches more credibility to second-level beliefs than to “eager” first-level beliefs, the qualm about messages being lost or suppressed does not arise. The second-level beliefs based on the response to a challenge will eliminate the possibility of the challenge message being lost or suppressed.

3.3 Knowledge Set Belief about Nonces

Nonce Freshness Belief

If a principal **generates** a nonce for a certain session, it believes that the *knowledge set* of that nonce consists of only that principal before the message is sent.

$$\frac{X \models \sharp(N_s, M_i); X \models KS(N_s, M_j) = \{ \} \forall j < i}{X \models KS(N_s, M_{i-1}) = \{X\}}$$

This belief is the definition of a nonce in terms of the KS notation.

Nonce Sharing

If a principal X generates a nonce and sends a message containing the nonce to another principal Y in message M_{i+1} , it gains the belief that the *knowledge set* of the nonce after M_{i+1} is $\{X, Y\}$.

$$\frac{X \models \sharp(N_s, M_i); X \models KS(N_s, M_j) = \{ \} \forall j < i; X \triangleright \{M_{i+1}, X, Y, N_s\}}{X \models \{KS(N_s, M_{i+1}) = \{X, Y\}\}}$$

This is a direct consequence of the *nonce freshness belief* and the *set inclusion belief*(2).

Belief in the Freshness of Message Contents

The belief in the freshness of a nonce generated by a principal will influence its belief about the freshness of other message contents that it receives in the future; i.e., if the received message contains some item that the principal believes to be fresh, then the principal infers that the other parts of the message are also fresh.

$$\frac{X \models \sharp(N_s, M_k); X \triangleleft \{M_k, Y, X, (N_s, C)\}}{X \models \sharp(C, M_k)}$$

That is, principal X sees a message M_k from Y which has a nonce N_s which X had generated earlier during the protocol run, and hence it believes message M_k to be fresh. If principal X believes that at message M_{k-1} , the *knowledge set* of N_s did not have Y , then it infers that Y read the nonce during this protocol run, and gains a belief that the message contents of this message from Y are fresh.

3.4 Belief about Another Principal's Knowledge Set Beliefs

From the *set inclusion belief* (2) it follows that whenever a principal (sender) conveys a known item to another principal (receiver) in the session, it gains the belief that the *knowledge set* of that message content now includes the sender and the recipient. The converse of this is that whenever a principal receives a message sent by another principal, if the message is **signed or encrypted by the sender** and if the recipient of the message believes in the freshness of the message contents, then it will obtain a belief that the sender now believes that the *knowledge set* of the message content contains the sender and the recipient.

$$\frac{X \triangleleft \{M_k, Y, X, C\}; X \models \sharp(C, M_k)}{X \models \{Y \models \{KS(C, M_k) = \{X, Y\}\}\}}$$

That is, if (1) X receives a message from Y with content C , and (2) X believes in the freshness of the content C , then X gains the belief that Y believes that $KS(C, M_k) = \{X, Y\}$.

3.5 Interpretation of First- and Second-Level Beliefs

In the BAN logic [1], the criteria for establishing authenticity between two communicating principals include the first-level and the second-level beliefs about the session keys that they share.

The First-Level Key Belief

If each of the two principals X and Y believe that the session key K_{xy} is shared with the other principal, and *trusts* all other principals which may have information about the key to (1) have knowledge of the session key K_{xy} and all future messages encrypted with K_{xy} and (2) maintain privacy of the key, then the first-level belief on the privacy of the session key is established. Letting M_l be the last message of the protocol run, we have

$$\begin{aligned} X & \models \{X, Y\} \in KS(K_{xy}, M_l) \\ & Trust_{(K_{xy})}(X, P) \forall P \in KS(K_{xy}, M_l) \\ Y & \models \{X, Y\} \in KS(K_{xy}, M_l) \\ & Trust_{(K_{xy})}(Y, P) \forall P \in KS(K_{xy}, M_l). \end{aligned}$$

The Second-Level Key Belief

If each of the two principals X and Y believe that the other principal has obtained the first-level key belief, then the second-level key belief is said to be established. In our notation,

$$\begin{aligned} X & \models \{Y \models \{X, Y\} \in KS(K_{xy}, M_l) \\ X & \models Trust_{(K_{xy})}(Y, P) \forall P \in KS(K_{xy}, M_l) \\ Y & \models \{X \models \{X, Y\} \in KS(K_{xy}, M_l) \\ Y & \models Trust_{(K_{xy})}(X, P) \forall P \in KS(K_{xy}, M_l). \end{aligned}$$

4 Extension to Accommodate Accumulation of Identities

In the previous section, we have proposed a logic which establishes beliefs in the privacy of the session key without imposing the constraint of *jurisdiction*. We claim that the belief in the privacy of the session key is based primarily on the trust which the members of the *knowledge set* place on each other.

In protocols which aim at establishing the goodness of the session key, we would like to detect (lack of) *jurisdiction*. To achieve this, we introduce the notion of an ordered *KS* (denoted \mathcal{KS} and represented as $\langle \text{ordered list of elements} \rangle$). A $\mathcal{KS}(a, M_i)$ is a set where principal X precedes principal Y if X knows the content a before Y . Each principal in the session keeps an account of the order in which it believes that the principals belonging to the \mathcal{KS} of the session key have obtained knowledge of the session key.

If each principal X in the session obtains the belief that the first element of the \mathcal{KS} ($First(\mathcal{KS})$) of the session key at the end of the protocol run has jurisdiction over the key and that all the principals which precede X in the \mathcal{KS} are trusted to share the key, then X can be assured that there is jurisdiction in the protocol. We modify the *set inclusion beliefs* to include the ordering in the elements of the *knowledge set*.

4.1 Set Inclusion Belief (1')

$$\frac{X \notin \mathcal{KS}(a, M_i); X \triangleleft \{M_{i+1}, Y, X, a\}}{X \models \mathcal{KS}(a, M_{i+1}) = \langle Y, X \rangle}$$

That is, when principal X receives the message content a for the first time from principal Y , it believes that since Y was already a member of the \mathcal{KS} , Y precedes X in the resulting \mathcal{KS} .

4.2 Set Inclusion Belief (2')

$$\frac{X \models \{Y \notin \mathcal{KS}(a, M_i)\}; X \triangleright \{M_{i+1}, X, Y, a\}}{X \models \mathcal{KS}(a, M_{i+1}) = \langle \mathcal{KS}(a, M_i), Y \rangle}$$

That is, if (1) principal X believes that Y does not belong to $\mathcal{KS}(a, M_i)$, and (2) X sends a message M_{i+1} to Y containing the information a , X now adds the new member Y to the last position on its $\mathcal{KS}(a, M_{i+1})$.

4.3 Belief about Another Principal's Belief about Ordered Knowledge Set

If principal Y tells X that it believes that the *ordered knowledge set* of a message content is $\mathcal{KS}(CK, M_{i-1})$, then X believes in what Y says only if X trusts Y to give the valid information about the \mathcal{KS} . If not, X obtains the belief that Y believes that X is now appended to $\mathcal{KS}(CK, M_{i-1})$ at the last position to obtain $\mathcal{KS}(CK, M_i)$.

$$\frac{X \triangleleft \{M_i, Y, X, (\mathcal{KS}(CK, M_{i-1}))\}; X \models \#(M_i)}{X \models Y \models \mathcal{KS}(CK, M_i) = \langle \mathcal{KS}(CK, M_{i-1}), X \rangle}$$

That is, if (1) Y sends M_i to X claiming that the *ordered knowledge set* of the session key CK is $\mathcal{KS}(CK, M_{i-1})$. (2) X believes that this message is fresh, then X believes that Y believes that the $\mathcal{KS}(CK, M_i)$ is $\mathcal{KS}(CK, M_{i-1}) + \{X\}$. If in addition,

$$X \models Y \models \mathcal{KS}(CK, M_{i-1}), \text{ then,}$$

$$X \models \mathcal{KS}(CK, M_i) = \langle \mathcal{KS}(CK, M_{i-1}), X \rangle$$

That is, if X believes in Y 's jurisdiction over $\mathcal{KS}(CK, M_{i-1})$, then X believes in what Y believes about $\mathcal{KS}(CK, M_{i-1})$.

4.4 Interpretation of First- and Second-Level Beliefs

The first- and second-level beliefs are identical to those mentioned in Section 3.5. Though we use *ordered knowledge sets*, the first- and second-level beliefs are obtained irrespective of the order in which the two principals X and Y appear in the $\mathcal{KS}(K_{xy}, M_i)$, M_i being the last message in the protocol run. In addition to the beliefs mentioned in Section 3.5, X and Y need to believe that the first member of the \mathcal{KS} of the session key ($First(\mathcal{KS})$) has jurisdiction over the key. In other words, the principals should obtain the belief that the key was generated by a principal who has authority to generate the key; i.e.,

$$\begin{aligned} X &\models \{First(\mathcal{KS}(K_{xy}, M_i)) \mid \Rightarrow K_{xy}\} \\ Y &\models \{First(\mathcal{KS}(K_{xy}, M_i)) \mid \Rightarrow K_{xy}\} \end{aligned}$$

Accumulation of Beliefs in Identities

In the event that the key generated at P_1 has been conveyed to X through the principals $P_2, P_3, \dots, P_{k-1}, P_k$; i.e.,

$$X \models \mathcal{KS}(K_{xy}, M_i) = \langle P_1, P_2, \dots, P_{k-1}, P_k, X \rangle$$

then X needs to believe that P_1 has jurisdiction over the session key and that $X \models P_k \models P_{k-1} \models \dots \models P_2 \models \mathcal{KS}(K_{xy}, M_f) = \langle P_1 \rangle$, M_f being the first message having key K_{xy} as its content. Also, each of the principals P_2, P_3, \dots, P_k are trusted by X and Y not only to convey the information about the session key but also to have knowledge of the key.

5 Examples of Protocol Analysis

The analysis is similar to that followed in [1]. Each message is first *idealized* (written in terms of the notation introduced). Next, all beliefs which follow from existing beliefs and the contents of a new message are listed. Using the inference rules, new beliefs are derived, and the key beliefs required to establish authenticity are obtained. For brevity, we denote the i level belief of principal J by $B_i(J)$.

In section 5.1 the inter-domain authentication protocol [3] is analyzed using BAN logic. The same protocol is analyzed in section 5.2 using the proposed logic, and the importance of accumulation of beliefs in identities of principals in their order of participation in the session is discussed. In section 5.2, the use of PROXY tickets in 7 is analyzed using BAN logic. Its analysis using *ordered knowledge sets* is presented in section 5.3. In section 5.4, we analyze the Multiparty Session protocol using BAN logic, and compare it with its analysis using the proposed logic in section 5.5.

5.1 Analysis of an Inter-domain Authentication Protocol using BAN logic

It is often necessary to be able to track the route of cascaded requests as part of providing access control [12]. In this protocol [3], secure channels are built in a cascaded fashion; i.e., using a secure channel from P_i to P_j and another secure channel from P_j to P_k , a secure channel is established between P_i and P_k . There is no global trust [3]. For the sake of illustration, we use principals P_1, P_2, P_3 and P_4 . We first analyze this protocol using the BAN logic.

Idealized Protocol Description

Message 1: $P_1 \rightarrow P_2$: Please forward $\{\overset{K_{P1}}{\rightarrow} P_1\}^{K_{P2}}$ to P_3

Message 2: $P_2 \rightarrow P_1$: $\{\overset{K_{P1}}{\rightarrow} P_1, P_2\}^{K_{P3}}, \{\overset{K_{P1}}{\rightarrow} P_3\}^{K_{P1}}$

Message 3: $P_1 \rightarrow P_3$: Please Forward $\{\overset{K_{P1}}{\rightarrow} P_1, P_2\}^{K_{P3}}$ to P_4

Message 4: $P_3 \rightarrow P_1$: $\{\overset{K_{P1}}{\rightarrow} P_1, P_2, P_3\}^{K_{P4}}, \{\overset{K_{P1}}{\rightarrow} P_4\}^{K_{P1}}$

Message 5: $P_1 \rightarrow P_4$: $\{\overset{K_{P1}}{\rightarrow} P_1, P_2, P_3\}^{K_{P4}}$

Assumptions

The assumptions about shared keys are as follows:

$$P_1 \models (\overset{K_{P2}}{\rightarrow} P_2)$$

$$P_2 \models (\overset{K_{P1}}{\rightarrow} P_1) \text{ and } (\overset{K_{P3}}{\rightarrow} P_3)$$

$$P_3 \models (\overset{K_{P2}}{\rightarrow} P_2) \text{ and } (\overset{K_{P3}}{\rightarrow} P_4)$$

$$P_4 \models (\overset{K_{P3}}{\rightarrow} P_3)$$

Analysis

Message 1: On seeing the request message from P_1 , P_2 obtains the belief $P_2 \models P_1 \models (\overset{K_{P1}}{\rightarrow} P_1)$ (assuming belief in freshness).

Message 2: P_2 conveys the authenticator key of P_3 to party P_1 . P_1 obtains the belief $P_1 \models P_2 \models (\overset{K_{P3}}{\rightarrow} P_3)$

Message 3: When P_3 sees the message encrypted by P_2 , $P_3 \models P_2 \models (\overset{K_{P1}}{\rightarrow} P_1)$

Message 4: P_1 sees M_3 sent by P_3 . From its previous belief about K_{P1} , P_1 obtains

$$P_1 \models P_2 \models P_3 \models (\overset{K_{P1}}{\rightarrow} P_4)$$

Message 5: P_4 sees the message encrypted with the key it shares with P_3 . Hence, it obtains the belief

$$P_4 \models P_3 \models (\overset{K_{P1}}{\rightarrow} P_1)$$

Comments

It can be observed here that the path information is not preserved in the beliefs obtained by principals P_3 and P_4 . While the causal belief is

$$P_4 \mid\equiv P_3 \mid\equiv P_2 \mid\equiv (P_1 \mid\equiv \xrightarrow{K_{P_1}} P_1)$$

P_4 obtains the belief $P_4 \mid\equiv P_3 \mid\equiv (\xrightarrow{K_{P_1}} P_1)$. After message exchange M_4 , principals P_2 and P_3 can also have beliefs about the keys K_{p_1} . If P_2, P_3 is together referred to as *path* from P_1 to P_4 , then the party P_4 should obtain a belief which is conditional on the trust that P_4 places on the *path*; i.e.,

$$P_4 \mid\equiv P_3 \mid\equiv P_2 \mid\equiv (P_1 \mid\equiv \xrightarrow{K_{P_1}} P_1)$$

To be able to have accountability, we need to preserve the (ordered) accumulation of beliefs. In the above example, we see that the analysis of the protocol does not preserve this ordered accumulation of beliefs in the identities of the principals which form a trust path between P_1 and P_4 .

The belief obtained by P_4 will have adverse consequences if the realm P_2 becomes untrustworthy after the protocol run. In this event, P_4 will still continue to believe in the identity of P_1 , as long as P_4 believes that P_3 remains trustworthy. This would result in a scenario where P_2 (untrusted realm) who has knowledge of the key K_{p_1} can access P_4 (service) with key (secret) and get serviced. This attack by P_2 will go undetected by P_4 even if P_4 is aware of the fact that party P_2 has been compromised. This is due to the belief of principal P_4 that P_1 's belief in the key is conditional only on the trustworthiness of realm P_3 .

This situation can be averted if beliefs are accumulated in an ordered manner. If P_4 has information about the identities of the intermediate principals, and trusts P_1 on condition that the *path* is trusted, then P_4 realizes that its key to service P_1 is no longer valid when it learns that the *path* is not to be trusted any more, and hence, it will not accept any future requests with that key.

Using the proposed logic, we show in the next section, that accumulation of information on the identities and on the beliefs of the intermediate principals on the trusted path can be achieved using *ordered knowledge sets*.

5.2 Analysis of the Inter-domain Authentication Protocol Using the Proposed Logic

The protocol description is given in the previous section and will not be repeated here. This protocol is now analyzed using the *ordered knowledge sets*.

Idealization

Message 1: $P_1 \triangleright P_2 : \{M_1, P_1, P_2, ((\mathcal{KS}(K_{p_1}, M_1) = \langle P_1 \rangle), P_3)\}$

Message 2: $P_2 \triangleright P_1 : \{M_2, P_2, P_3, (\mathcal{KS}(K_{p_1}, M_2) = \langle P_1, P_2 \rangle)\}$,

Message 3: $P_1 \triangleright P_3 : \{M_2, P_2, P_1, (\xrightarrow{K_{P_3}} P_3)\}$
 $\{M_2, P_2, P_3, ((\mathcal{KS}(K_{p_1}, M_2) = \langle P_1, P_2 \rangle), P_4)\}$

Message 4: $P_3 \triangleright P_1 : \{M_4, P_3, P_4, (\mathcal{KS}(K_{p_1}, M_4) = \langle P_1, P_2, P_3 \rangle)\}, \{M_4, P_3, P_1, (\xrightarrow{K_{P_4}} P_4)\}$

Message 5: $P_1 \triangleright P_4 : \{M_4, P_3, P_4, (\mathcal{KS}(K_{p_1}, M_4) = \langle P_1, P_2, P_3 \rangle)\}$

Analysis:

Message 2: P_1 receives K_{p_3} from P_2 in message M_2 , and believes that $P_1 \mid\equiv P_2 \mid\equiv (\xrightarrow{K_{P_3}} P_3)$

In the \mathcal{KS} notation,

$$P_1 \mid\equiv \mathcal{KS}(K_{p_3}, M_2) = \langle P_3, P_2, P_1 \rangle$$

Message 4: On receiving M_4 which is encrypted with K_{p_3} , P_1 obtains the belief

$$P_1 \mid\equiv P_2 \mid\equiv P_3 \mid\equiv (\xrightarrow{K_{P_4}} P_4)$$

In the \mathcal{KS} notation,

$$P_1 \mid\equiv \mathcal{KS}(K_{p_4}, M_4) = \langle P_4, P_3, P_2, P_1 \rangle$$

Message 5: On receiving M_5 , P_4 uses *belief about ordered knowledge set of another principal* (assuming that P_4 believes that M_5 is fresh), and obtains the belief,

$$P_4 \mid\equiv P_3 \mid\equiv \mathcal{KS}(K_{p_1}, M_5) = \langle P_1, P_2, P_3 \rangle$$

Writing this in the \mathcal{KS} notation, we have

$$P_4 \mid\equiv \mathcal{KS}(K_{p_1}, M_5) = \langle P_1, P_2, P_3, P_4 \rangle$$

Note that P_4 does not believe $\mathcal{KS}(K_{p_1}, M_5)$ fully unless it believes that P_3 has jurisdiction over $\mathcal{KS}(K_{p_1}, M_5)$.

Comments

The principal P_1 considers the messages from P_4 as valid on condition that the path through P_2, P_3 is trustworthy. Similarly, P_4 considers the messages from P_1 valid only if it trusts the principals in the path from P_1 to P_4 .

It can be observed here that the analysis has preserved belief ordering. Without the accumulation of beliefs in an ordered sequence, the analysis would not yield the required information on the trusted path.

5.3 Analysis of PROXY ticket forwarding using BAN logic

This is an example of a case where a client initiates the session and obtains the ticket for a specific service and/or object and forwards the ticket to another principal. The ticket forwarded is not forwardable. We briefly analyze this protocol [8] using BAN logic, and show the potential limitations.

Protocol Description

Message 1: $X \rightarrow TGS: Request\ proxy\ ticket\ for\ principal\ Y\ to$

access service S

Message 2: $TGS \rightarrow X$:
 $\{Y_{address}, P, X, S\}^{K_S}, \{K_{X/Y-S}\}^{K_{X-TGS}}$

Message 3: $X \rightarrow Y$: $\{Y_{address}, P, X, S\}^{K_S}, \{K_{X/Y-S}\}^{K_{X-Y}}$

Message 4: $Y \rightarrow S$: $\{Y_{address}, P, X, S\}^{K_S},$
 $\{Authenticator_X, X\}^{K_{X/Y-S}}$

Assumptions

The assumptions about shared keys are as follows:

$$\begin{aligned} S & \models \overset{K_S}{\rightarrow} TGS; S \models TGS \mid \Rightarrow X \overset{K_{X-S}}{\longleftrightarrow} S; \\ S & \models TGS \mid \Rightarrow TGS \overset{Proxy}{\longleftrightarrow} S; \\ S & \models \#(Authenticator_X, Proxy) \end{aligned}$$

Idealization

Message 2: $TGS \rightarrow X$: $\{Y, TGS \overset{Proxy}{\longleftrightarrow} S, X\}^{K_S},$
 $\{X \overset{K_{X/Y-S}}{\longleftrightarrow} S\}^{K_{X-TGS}}$

Message 3: $X \rightarrow Y$: $\{Y, TGS \overset{Proxy}{\longleftrightarrow} S, X\}^{K_S},$
 $\{X \overset{K_{X/Y-S}}{\longleftrightarrow} S\}^{K_{X-Y}}$

Message 4: $Y \rightarrow S$: $\{Y, TGS \overset{Proxy}{\longleftrightarrow} S, X\}^{K_S}$
 $\{X \overset{K_{X/Y-S}}{\longleftrightarrow} S\}^{K_{X/Y-S}}$

Analysis

X receives message M_2 from TGS and obtains its first-level belief using the *nonce verification* and *jurisdiction* rules (assuming that X believes in the freshness of M_2); i.e., $X \models \{X \overset{K_{X/Y-S}}{\longleftrightarrow} S\}$

Message 3: X sends the key to Y and gains no new beliefs in the process. Y sees M_3 and decrypts the key $K_{X/Y-S}$, and Y obtains the belief $Y \models X \models \{X \overset{K_{X/Y-S}}{\longleftrightarrow} S\}$ (assuming that Y believes in the freshness of this message).

Message 4: S sees the message M_4 and obtains the beliefs.

$$S \models X \models \{X \overset{K_{X/Y-S}}{\longleftrightarrow} S\}$$

Though S decrypts the proxy ticket given by the TGS and finds Y 's address, the idealization of BAN logic does not permit the derivation of the belief

$S \models Y \models X \models \{X \overset{K_{X/Y-S}}{\longleftrightarrow} S\}$, which preserves causality while informs the principal S about the beliefs that Y may have about the key. In the next section, we show that the ordered accumulation of identities of principals is captured by the evolution of beliefs.

5.4 Analysis Using the Proposed Logic

The protocol description is the same as that in the previous section and will not be repeated here.

Idealization

Message 1: $X \triangleright \{M_1, X, TGS, (Request\ proxy, Y, S)\}$

Message 2: $TGS \triangleright$
 $\{M_2, TGS, S, (Y, Proxy, X, S)\}, \{M_2, TGS, X, (K_{X/Y-S})\}$

Message 3: $X \triangleright$
 $\{M_2, TGS, S, (Y, Proxy, X, S)\}, \{M_3, X, Y, (K_{X/Y-TGS})\}$

Message 4: $Y \triangleright \{M_4, Y, S, (Authenticator_X, X, K_{X/Y-KDC})\},$
 $\{M_2, TGS, S, (Y, Proxy, X, S)\}$

Assumptions

$$\begin{aligned} S & \models \#(Proxy\ ticket); \\ S & \models \#(Authenticator_X); \\ S & \models TGS \mid \Rightarrow Proxy\ ticket \end{aligned}$$

Analysis

Message 3: On sending M_3 , $X \models \mathcal{KS}(K_{X/Y-S}, M_3) = \{TGS, X, Y\}$ using *set inclusion belief (2)*. On receiving M_3 , Y uses *set inclusion belief (1)* and obtains

$$Y \models \mathcal{KS}(K_{X/Y-S}, M_3) = \{X, Y\}$$

Message 4: Y sends M_4 and obtains its first-level belief using *set inclusion belief (2)*.

$$Y \models \mathcal{KS}(K_{X/Y-S}, M_4) = \{X, Y, S\} \dots \dots [B_1(Y)]$$

When S receives M_4 , it interprets the message encrypted by the TGS which has the identity of principal Y , and assumes that the key $K_{X/Y-S}$ is now shared with X and Y . Hence, S obtains its first-level belief using *set inclusion belief (1)*; i.e., $S \models \mathcal{KS}(K_{X/Y-S}, M_4) = \{X, Y, S\} \dots \dots [B_1(S)]$

S believes in the goodness of the *proxy ticket* which is generated by TGS , X also believes in the freshness of the *Authenticator_X* (by assumption). S now knows that the key $K_{X/Y-S}$ is not a secret that it shares with only X . Hence, it interprets M_4 as a message from either X or Y . Using the *belief about the freshness of message contents*,

$$S \models (X/Y) \models \mathcal{KS}(K_{X/Y-S}, M_4) = \{X, Y, S\} \dots \dots [B_2(S)]$$

In addition, S sees that the *proxy ticket* has been generated by the TGS and forwarded without encryption. Hence, $S \models \mathcal{KS}(Proxy\ ticket, M_4) = \{TGS, S\}$. Since S believes that TGS has *jurisdiction* over this ticket, it obtains first-level belief in the goodness of the proxy ticket. S may obtain beliefs about the privacy of the $K_{X/Y-S}$ only if it trusts X and Y to share this key and also to maintain its privacy.

5.5 Analysis of Multiparty Session Protocol using BAN logic

We now consider in detail, the multiparty session protocol. This is an example of a protocol which lacks jurisdiction trust on key goodness, but nevertheless, is able to establish key privacy. The formal top level spe-

cification for this appears in [4].

Protocol Description

Message 1: $X \rightarrow Y \{ch, ss, X, Y, A, B, N_s, nil, nil, 1\}^{PK_Y}$
 Message 2: $Y \rightarrow AS \{ch, ss, X, Y, A, B, N_s, N_d, nil, 2\}^{PK_{AS}}$
 Message 3: $AS \rightarrow X \{ch, ss, X, Y, A, B, N_s, N_d, CK, 3\}^{PK_X}$
 Message 4: $X \rightarrow Y \{ch, ss, X, Y, A, B, N_s, N_d, CK, 4\}^{PK_Y}$
 Message 5: $Y \rightarrow X \{ch, ss, X, Y, A, B, N_s, N_d, CK, 5\}^{PK_X}$

Assumptions

$X \models \#(N_s); Y \models \#(N_d); X \models \xrightarrow{PK_Y} Y; Y \models \xrightarrow{PK_Y} X$
 $Y \models \xrightarrow{PK_X} X; X \models \xrightarrow{PK_X} X; X \models \xrightarrow{PK_{AS}} AS$
 $Y \models \xrightarrow{PK_{AS}} AS$
 $X \models AS \Rightarrow X \xleftrightarrow{CK_{xy}} Y; Y \models AS \Rightarrow X \xleftrightarrow{CK_{xy}} Y$

Idealization

Message 1: $X \rightarrow Y \{X, Y, N_s\}^{PK_Y}$
 Message 2: $Y \rightarrow AS \{X, Y, N_s, N_d\}^{PK_Y}$
 Message 3: $AS \rightarrow X \{X, Y, N_s, N_d, X \xleftrightarrow{CK_{xy}} Y\}^{PK_X}$
 Message 4: $X \rightarrow Y \{X, Y, N_s, N_d, X \xleftrightarrow{CK_{xy}} Y\}^{PK_Y}$
 Message 5: $Y \rightarrow X \{X, Y, N_s, N_d, X \xleftrightarrow{CK_{xy}} Y\}^{PK_X}$

Analysis

Message 1 and 2 do not contribute to the logic.

Message 3: From the assumption about its own public key, X recognizes that the message is sent for it. It then sees AS 's signature on the message and finds that AS said it.

$$X \models AS \sim \{X, Y, N_s, N_d, X \xleftrightarrow{CK_{xy}} Y\}^{PK_Y}$$

Now, since X believes in the freshness of N_s , it believes that the contents of message 3 i.e., (N_d, CK_{xy}) are fresh. Using the *nonce verification*,

$$X \models AS \models \{X \xleftrightarrow{CK_{xy}} Y\}$$

Using *jurisdiction* (assumption),

$$X \models \{X \xleftrightarrow{CK_{xy}} Y\} \dots \dots \dots [B_1(X)]$$

Message 4: The analysis here is similar to message 3 except that in this case there is no jurisdiction because Y does not believe that X controls the session key. Hence, using the *nonce verification* and the key beliefs (assumption),

$$Y \models X \models \{X \xleftrightarrow{CK_{xy}} Y\} \dots \dots \dots [B_2(Y)]$$

Message 5: Using *nonce verification* and key beliefs, we obtain

$$X \models Y \models \{X \xleftrightarrow{CK_{xy}} Y\} \dots \dots [B_2(X)]$$

Though principal Y has not obtained its first-level belief, X believes that Y has obtained it.

Comments

The principal Y does not obtain its *first – level key belief*. The main stumbling block in the application of this logic is the lack of jurisdiction in the key messages to principal Y . Y does not obtain its first-level belief in both the goodness and the privacy of the session key.

Using the proposed logic, we will show in the following section that the Multiparty Session protocol achieves first-level beliefs on key privacy.

5.6 Analysis of Multiparty Session Protocol using the proposed logic

The protocol description is same as in the previous section and will be omitted here for sake of conciseness.

Idealization

Message 1: $X \triangleright \{M_1, X, Y, N_s\}$
 Message 2: $Y \triangleright \{M_2, Y, AS, (N_d, N_s)\}$
 Message 3: $AS \triangleright \{M_3, AS, X, (N_d, N_s, CK)\}$
 Message 4: $X \triangleright \{M_4, X, Y, (N_s, N_d, CK)\}$
 Message 5: $Y \triangleright \{M_5, Y, X, (N_s, N_d, CK)\}$

Assumptions

$$X \models \#(N_s); \quad Y \models \#(N_d)$$

It is unnecessary to mention the key beliefs here since we already have made the initial assumption that the sender and recipient identity can be established by the recipient who can decrypt messages encrypted with his public key, and can identify the sender by his signature on the message.

Analysis

Message 1: Since X generates the nonce N_s , using *nonce sharing* and *set inclusion belief (2)*,

$$X \models KS(N_s, M_1) = \{X, Y\}$$

Message 2: Y sends N_s, N_d to AS . Since the belief about KS of N_s is not pertinent to Y (since Y does not believe in the freshness of N_s) we use only Y 's belief about KS of N_d . $Y \models KS(N_d, M_2) = \{Y, AS\}$

Message 3: At M_1 , X had believed that $KS(N_s, M_1) = \{X, Y\}$. When it sees M_3 from AS , using *set inclusion belief (1)*,

$$X \models KS(N_s, M_3) = \{X, Y, AS\}$$

X does not know about N_d or CK so far. Now when it receives them from AS , using *set inclusion belief (1)*,

$$X \models KS(N_d, M_3) = \{AS, X\}$$

$$X \models KS(CK, M_3) = \{AS, X\}$$

Principal X knows that its own nonce N_s is fresh. Since the present message has N_s , using the *belief in the freshness of message contents*, it obtains the belief

that the contents of M_3 ; i.e., $X \models \sharp((N_d, CK), M_3)$

Message 4: From its earlier belief about $KS(CK, M_3)$, X updates the KS belief when it sends M_4 to Y .

$$X \models KS(CK, M_4) = \{AS, X, Y\} \dots [B_1(X)]$$

When Y receives CK from X for the first time, it believes that the $KS(CK, M_4)$ consists of only X and Y .

$$Y \models KS(CK, M_4) = \{Y, X\} \dots [B_1(Y)]$$

$$Y \models KS(N_d, M_4) = \{AS, Y, X\}$$

From the previous belief of Y that $KS(N_d, M_2) = \{Y, AS\}$, and using the *belief in the freshness of message contents*, Y obtains the belief that the key CK and nonce N_s are fresh. Since Y receives the key CK which it believes to be fresh from X , using the *belief about another principal's KS beliefs*, we obtain

$$Y \models X \models KS(CK, M_4) = \{X, Y\} \dots [B_2(Y)]$$

Message 5: Since Y has gained the belief about the freshness of CK from the last message exchange, it uses the *belief about KS belief of other principals*, $X \models Y \models KS(CK, M_5) = \{X, Y\} \dots [B_2(X)]$

6 Features of the Logic

The proposed logic models the evolution of beliefs within a protocol run. Each principal keeps an account of the principals that he believes to be sharing a pertinent piece of information in that protocol run. The beliefs evolve based on whether or not the most recent action is related to the existing belief.

The beliefs evolve within a protocol in a manner similar to the states (output) of a state machine. The state machine is memoryless, hence the beliefs of the past are not accumulated in general. In other words, at message instance M_{i+1} , the *knowledge set* of an item at M_i is updated. The old *knowledge set* is no longer of consequence for future inferences. Hence, only the most recent *knowledge set* is retained. Membership of principals in any given KS is permanent for that session. The cardinality of the KS grows monotonically within a session run.

The principals in a session can obtain first-level belief only if they trust all the members of the *knowledge set* of the session key to have knowledge of the session key and access to all future messages encrypted with this key. In addition to this, they need to believe that all principals belonging to the KS of the session key will maintain confidentiality of the session key and also the messages encrypted with the session key in future.

Since the inference rules presented invariably involve message numbering, the messages in a protocol run are

assumed to be ordered; i.e., M_i can occur only after M_{i-1} and before M_{i+1} . Messages are assumed not to be lost or reordered during the session run. Beliefs obtained based on messages which are sent ('eager beliefs') involve some risk, since there is no guarantee that the message which is sent ('challenge') will be received by the recipient. However, the response to the challenge message enables the sender of the challenge to corroborate his 'eager' first-level belief. Hence, obtaining second-level belief based on the response to a challenge will imply that (1) the challenge message was received, (2) first-level belief has been obtained by the other principal.

A positive outcome of this logic is that by eliminating the constraint that all key messages should be encrypted by the authentication server, it essentially lends itself to the analysis of a de-centralized data flow. Using the modified inference rules, we can detect the lack of jurisdiction in protocols where jurisdiction is important. The extended logic is also applicable to the class of protocols in which ordered accumulation of belief in the identities of principals is important.

7 Conclusion

The belief evolution model takes advantage of the order in which message exchanges take place within a session run to obtain concise proofs of authentication protocols. This logic can be applied to the analysis of the class of protocols (1) which require (ordered) accumulation of beliefs in the principals identities, and (2) which do not necessarily make use of key jurisdiction properties. The proof preserves ordering of beliefs and hence maintains consistency of beliefs. The 'eager' first-level beliefs obtained based on messages which are sent are corroborated when the responses to these messages are received. The second-level beliefs obtained based on these responses eliminate the possibility that the messages sent were lost.

Acknowledgements

We would like to thank Li Gong and Martin Abadi for their comments on a preliminary draft of this paper.

References

- [1] M.Burrows, M.Abadi, and R.Needham, "A Logic of Authentication," in Proceedings of the 12th ACM Symposium on Operating Systems Principles, Litchfield Park, Arizona, December, 1989. Published as *ACM Operating Systems Review*,

Vol.23, No.5, December 1989; a fuller version appears in DEC System Research Center report No.39, Palo Alto, California, February, 1989.

- [2] M. Burrows, M. Abadi, and R. Needham, "A logic of Authentication," *ACM Transactions on Computer Systems*, Vol.8, No.1, February, 1990.
- [3] A.D. Birrel, B. W. Lampson, R. M. Needham and M.D. Schroeder, "A Global Authentication Service without Global Trust," in the Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, 1986.
- [4] P.-C. Cheng and V.D. Gligor, "On the formal Specification and verification of a Multiparty Session Protocol," Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, May 7-9, 1990, Oakland, California.
- [5] V.D. Gligor, R. Kailar, S. Stubblebine and L. Gong, "Logics for Cryptographic Protocols - Virtues and Limitations," these Proceedings.
- [6] L. Gong, R. Needham, R. Yahalom, "Reasoning about Belief in Cryptographic Protocols," Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, May 7-9, 1990, Oakland, California.
- [7] J.Y. Halpern, Y. Moses, "Knowledge and Common knowledge in a distributed environment," Proceedings of the ACM Conference on Distributed Computing, 1984.
- [8] J. Kohl, C. Neuman, J. Steiner, "Kerberos Version 5 RFC, draft 2," MIT Project Athena, November 1989.
- [9] S.P. Miller, C. Neuman, J.I. Schiller, J.H. Saltzer, "Kerberos Authentication and Authorization System," Project Athena Technical Plan, Section E.2.1, MIT, July 1987.
- [10] J. Pato, "DCE Authorization Services, - Privilege Server," Cooperative Computing Division, Hewlett-Packard Co., Chelmsford, MA 01824, Version 4, March 22, 1990.
- [11] R.L. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," *Communication of the ACM*, Vol.21, No.2, February 1978.
- [12] K.R. Sollins, "Cascaded Authentication," Proceedings of IEEE Symposium on Security and Privacy, April 18-21, 1988, Oakland, California.

[13] V.L. Voydock, S.T. Kent, "Security Mechanisms in High-level Network Protocols," *Computing Surveys*, Vol 15. No.2, 1983.

A The Kerberos Protocol

The top level specification of the protocol appears in [9]. A simplified version of the protocol is as follows:

Message 1: $A \rightarrow AS : A, B, T_c$

Message 2: $AS \rightarrow A : \{T_s, T_c, L, K_{ab}, B, \{T_s, L, K_{ab}, A\}^{K_{bs}}\}^{K_{as}}$

Message 3: $A \rightarrow B : \{T_s, L, K_{ab}, A\}^{K_{bs}}, \{A, T_a\}^{K_{ab}}$

Message 4: $B \rightarrow A : \{T_a + 1\}^{K_{ab}}$

Idealization

Message 1: $A \triangleright \{M_1, A, AS, (A, B, T_c)\}$

Message

2:

$AS \triangleright \{M_2, AS, A, (K_{ab}, T_s, T_c, \{M_2, AS, B, (T_s, K_{ab})\})\}$

Message 3: $A \triangleright \{M_2, AS, B, K_{ab}\}, \{M_3, A, B, (T_a, K_{ab})\}$

Message 4: $B \triangleright \{M_4, B, A, (T_a, K_{ab})\}$

Assumptions

$A \models \#(T_a, T_c); A \models \#(T_s); B \models \#(T_a); B \models \#(T_s)$

Since we have made the initial assumption that the message recipient can identify the sender, we don't need the additional assumptions about the shared keys between principals. The assumptions mentioned above still rely on synchronized clocks.

Analysis

We start at message 1 and work towards the end of the protocol, producing the beliefs gathered on the way. In order to be sure that A is not responding to a replayed message M_2 , it needs to verify the freshness of M_2 . To do this, A includes a nonce T_c in its first message to AS and checks the second message to find the same nonce in it.

Message 1: From *nonce sharing*, A believes that after M_1 , the nonce T_c is shared between A and AS ; i.e., $A \models KS(T_c, M_1) = \{A, AS\}$

Message 2: A reads T_c in M_2 , and using *belief in the freshness of message contents*, $A \models \#(M_2)$. Using *set inclusion belief (1)*,

$A \models KS(K_{ab}, M_2) = \{A, AS\}$

Message 3: From *set inclusion belief (2)*,

$A \models KS(K_{ab}, M_3) = \{A, AS, B\} \dots [B_1(A)]$

B first decrypts the key message M_2 from AS and gets the key K_{ab} . Now it reads the second part of the message M_3 . Using the *set inclusion belief (1)*,

$B \models KS(K_{ab}, M_3) = \{A, B, AS\} \dots [B_1(B)]$

Since B believes in the freshness of T_a (assumption), using *belief in the freshness of message contents*, $B \models \#(K_{ab})$. Now, using *belief about another principal's KS beliefs*,

$$B \models A \models \{KS(K_{ab}, M_3) = \{A, B\}\} \dots [B_2(B)]$$

Message 4 : A believes in the freshness of T_a (assumption). Using *belief in the freshness of message contents*,

$$A \models \#(K_{ab})$$

Applying the *belief in another principal's KS belief* as before,

$$A \models B \models \{KS(K_{ab}, M_4) = \{A, B\}\} \dots [B_2(A)]$$

Note that apart from establishing all the key beliefs, the proof maintains the ordering in beliefs. That is, principal $X(Y)$ can obtain the second-level belief about principal $Y(X)$ only after principal $Y(X)$ has obtained its first-level belief. This property of the logic leads to beliefs which are ordered.

B The Andrew Secure RPC Handshake

We provide this here to show that the inference rules developed can detect weaknesses in authentication protocols in a similar way as those of [1]. In this protocol, principal A has key K_{ab} which it shares with the server B . A obtains a new key K'_{ab} from server B using a handshake operation.

Protocol Description

Message 1: $A \rightarrow B : A, \{N_a\}^{K_{ab}}$

Message 2: $B \rightarrow A : \{N_a + 1, N_b\}^{K_{ab}}$

Message 3: $A \rightarrow B : \{N_b + 1\}^{K_{ab}}$

Message 4: $B \rightarrow A : \{K'_{ab}, N'_b\}^{K_{ab}}$

Assumptions

$$A \models \#(N_a); \quad B \models \#(N_b); \quad B \models \#(K'_{ab})$$

Idealization

Message 1: $A \triangleright \{M_1, A, B, N_a\}$

Message 2: $B \triangleright \{M_2, B, A, (N_a, N_b)\}$

Message 3: $A \triangleright \{M_3, A, B, N_b\}$

Message 4: $B \triangleright \{M_4, B, A, (CK, N'_b)\}$

Analysis

Message 1: Principal A sends N_a to principal B in the first round. Hence, from *set inclusion belief (2)*,

$$A \models KS(N_a, M_1) = \{A, B\}$$

From *set inclusion belief (1)*,

$$B \models KS(N_a, M_1) = \{A, B\}$$

Note that since B does not see anything in M_1 that it believes to be fresh, it does not get the second-level

belief about N_a .

Message 2: Principal B generates and sends N_b to A . From *set inclusion belief (2)*,

$$B \models KS(N_b, M_2) = \{A, B\}$$

Since $A \models \#(N_a)$, from *belief in the freshness of message contents*, $A \models \#(N_b)$. Using *belief about another principal's KS belief*,

$$A \models B \models KS(N_b, M_2) = \{A, B\}$$

Message 3: B receives M_3 in which it sees N_b which it believes to be fresh (assumption). From *belief about another principal's KS belief*,

$$B \models A \models KS(N_b, M_3) = \{A, B\}$$

Message 4: Using *set inclusion belief (2)*,

$$B \models KS(CK, M_4) = \{A, B\} \dots [B_1(B)]$$

$$B \models KS(N'_b, M_4) = \{A, B\}$$

Principal A receives M_4 , but does not see any content in it that it believes to be fresh (i.e., $A \not\models \#(N'_b)$) Using *set inclusion belief (1)*,

$$A \models KS(CK, M_4) = \{A, B\} \dots [B_1(A)]$$

Since A does not believe in the freshness of M_4 , it cannot obtain its second-level belief.

Comments

At the end of the protocol run, the two principals A and B do not obtain second-level beliefs. Though principal B obtains its first-level belief, this is not corroborated by the second-level belief which can come only if A replies with a key message which B believes to be fresh. Since there is no belief obtained about the freshness of messages, the protocol is vulnerable to replay attacks. The remedy suggested in [1] is to add the nonce N_a to the last message. The proof of the concrete realization of this protocol follows.

C Concrete Realization of The Andrew Secure RPC Handshake Protocol

Protocol Description

Message 1: $A \rightarrow B : A, N_a$

Message 2: $B \rightarrow A : \{N_a, K'_{ab}\}^{K_{ab}}$

Message 3: $A \rightarrow B : \{N_a\}^{K'_{ab}}$

Message 4: $B \rightarrow A : \{N'_b\}$

Assumptions

$$A \models \#(N_a); \quad B \models \#(N_b); \quad B \models \#(K'_{ab})$$

Idealization

Message 1: $A \triangleright \{M_1, A, B, N_a\}$

Message 2: $B \triangleright \{M_2, B, A, (N_a, K'_{ab})\}$
 Message 3: $A \triangleright \{M_3, B, A, (N_a, K'_{ab})\}$
 Message 4: $B \triangleright \{M_4, B, A, N'_b\}$

Analysis

Message 1: A sends message to B . Using *set inclusion belief (2)*,

$$A \models KS(N_a, M_1) = \{A, B\}$$

Using *set inclusion belief (1)*, B obtains belief:

$$B \models KS(N_a, M_1) = \{A, B\}$$

Message 2: B sends message containing K'_{ab} to A . Using the *set inclusion belief (2)*,

$$B \models KS(K'_{ab}, M_2) = \{A, B\} \dots [B_1(B)]$$

A receives the message containing K'_{ab} for the first time from B . Hence,

$$A \models KS(K'_{ab}, M_2) = \{A, B\} \dots [B_1(A)]$$

A believes in the freshness of nonce N_a since it has generated it in the present run of the protocol. Using *belief about freshness of message contents*,

$$A \models \#(K'_{ab})$$

Since A believes that K'_{ab} is fresh and that B has sent it, using *belief about another principal's KS belief*,

$$A \models B \models KS(K'_{ab}, M_2) = \{A, B\} \dots [B_2(A)]$$

Message 3: B sees a message encrypted with the new key that it has generated. Since it believes in the freshness of this new key, and since it knows that only A had received it (*recipient uniqueness belief*), it now believes that this message from A is fresh. Using *belief about another principal's KS belief*,

$$B \models A \models KS(K'_{ab}, M_3) = \{A, B\} \dots [B_2(B)]$$

Hence, this realization of the Andrew Secure RPC Handshake protocol achieves all key beliefs.

D Privileged Ticket distribution Protocol

In this protocol [10], a logically independent server, the *privilege server* is used to create tickets that contain sealed information, *unique identifiers (UUIDs)* for the principal and the groups to which the principal belongs.

The session involves three parties: (1) The *client* who needs the session key to communicate with the *ticket granting server*, (2) the *privilege server* which acts as an intermediary between the *ticket granting server* and the *client*, and (3) The *ticket granting server*.

Notation:

The *client*, *privilege server* and *ticket granting server* are denoted by C , PS and TGS respectively.

The shared private key between parties X and Y is denoted by K_{xy} . The plaintext ticket between parties X and Y is denoted by T_{xy} . The secret key of party

X is denoted by K_x . The privileges associated with principal C are denoted by $C.Priv$.

The key between the *privilege server* and the *ticket granting server* that is given to the *client* at the end of the session is denoted by $K_{ps(c)-tgs}$.

The protocol description:

Message 1: $C \rightarrow TGS : C, TGS$

Message 2: $TGS \rightarrow C : \{T_{c-tgs}\}^{K_{tgs}}, \{K_{c-tgs}\}^{K_c}$

Message 3: $C \rightarrow TGS : \{T_{c-tgs}\}^{K_{tgs}}, \{N_c\}^{K_{c-tgs}}$

Message 4: $TGS \rightarrow C : \{T_{c-ps}\}^{K_{ps}}, \{K_{c-ps}\}^{K_{c-tgs}}$

Message 5: $C \rightarrow PS : \{\{T_{c-ps}\}^{K_{ps}}, \{N_c\}^{K_{c-ps}}, TGS\}$

Message 6: $PS \rightarrow TGS : \{\{T_{ps-tgs}\}^{K_{tgs}}, \{N_{ps}\}^{K_{ps-tgs}}\} \{C.Priv\}^{K_{ps-tgs}}$

Message 7: $TGS \rightarrow PS : \{T_{ps-tgs}, C.Priv\}^{K_{tgs}},$

$$\{K_{ps(c)-tgs}\}^{K_{ps-tgs}}$$

Message 8: $PS \rightarrow C : \{T_{ps-tgs}, C.Priv\}^{K_{tgs}},$

$$\{K_{ps(c)-tgs}\}^{K_{c-ps}}$$

Message 9: $C \rightarrow TGS :$

$$\{\{T_{ps-tgs}, C.Priv\}^{K_{tgs}}, \{N_c\}^{K_{ps(c)-tgs}}, S\}$$

Message 10:

$$TGS \rightarrow C : \{\{T_{ps}, C.Priv\}^{K_s}, \{K_{cs}\}^{K_{ps(c)-tgs}}\}$$

Idealization

Message 1: $C \triangleright \{M_1, C, TGS, (C, TGS)\}$

Message 2: $TGS \triangleright$

$$\{M_2, TGS, C, (\{M_2, TGS, TGS, T_{c-tgs}\}, K_{c-tgs})\}$$

Message 3: $C \triangleright \{M_3, C, TGS, (\{M_2, TGS, TGS, T_{c-tgs}\}, N_c)\}$

Message 4: $TGS \triangleright$

$$\{M_4, TGS, C, (\{M_4, TGS, PS, T_{c-ps}\}, K_{c-tgs})\}$$

Message 5: $C \triangleright \{M_5, C, PS, (T_{c-ps}, N_c, TGS)\}$

Message 6: $PS \triangleright \{M_6, PS, TGS, (T_{ps-tgs}, N_{ps}, C.Priv)\}$

Message 7: $TGS \triangleright$

$$\{M_7, TGS, PS, (\{M_7, TGS, TGS, (T_{ps-tgs}, C.Priv)\}, K_{ps(c)-tgs})\}$$

Message 8: $PS \triangleright$

$$\{M_8, PS, C, (\{M_7, TGS, TGS, (T_{ps-tgs}, C.Priv)\}, K_{ps(c)-tgs}, K_{c-ps})\}$$

Message 9: $C \triangleright$

$$\{M_9, C, TGS, (\{M_7, TGS, TGS, (T_{ps-tgs}, C.Priv)\}, N_c, S)\}$$

Message 10: $TGS \triangleright$

$$\{M_{10}, TGS, C, (\{M_6, TGS, S, (T_{ps}, C.Priv, K_{ps})\}, K_{ps(c)s})\}$$

Assumptions

$$TGS \models \#(K_{ps(c)-tgs}); C \models \#(N_c); C \models \#(K_{c-ps})$$

Analysis

Messages 1 through 6 do not contribute to the logic. We now analyze message 7.

Message 7: TGS sends the message containing the key $K_{ps(c)-tgs}$ for the first time to *privilege server* PS . From *set inclusion belief (2)*,

$$TGS \models KS(K_{ps(c)-tgs}, M_7) = \{TGS, PS\}$$

Message 8: C receives M_8 containing $K_{ps(c)-tgs}$ from PS . From *set inclusion belief (1)*,

$$C \models KS(K_{ps(c)-tgs}, M_8) = \{PS, C\}$$

From the assumption that C believes in the freshness of K_{c-ps} , it believes in the freshness of message M_8 . Hence, it follows that $C \models \#(K_{ps(c)-tgs})$

Message 9: C sends M_9 containing $K_{ps(c)-tgs}$ to D . Using *set inclusion belief (2)*,

$$C \models KS(K_{ps(c)-tgs}, M_9) = \{C, PS, TGS\} \dots [B_1(C)]$$

On receiving M_9 containing $K_{ps(c)-tgs}$ from C , using the *set inclusion belief (1)* principal TGS obtains the new belief,

$$TGS \models KS(K_{ps(c)-tgs}, M_9) = \{TGS, PS, C\} \dots [B_1(TGS)]$$

Since TGS believes that the key $K_{ps(c)-tgs}$ which it has generated in M_7 is fresh, using the *belief about message freshness*, TGS believes that the message from C is fresh. Further, using the *belief about another principal's KS belief*,

$$TGS \models C \models KS(K_{ps(c)-tgs}, M_9) = \{C, TGS\} \dots [B_2(TGS)]$$

Message 10: C receives M_{10} , in which it sees $K_{ps(c)-tgs}$ which it believes to be fresh. Hence, using *belief about another principal's KS belief*,

$$C \models TGS \models KS(K_{ps(c)-tgs}, M_{10}) = \{C, TGS\} \dots [B_2(C)]$$