Logic for verifying public-key cryptographic protocols

T. Coffey
P. Saidha

Indexing terms: Cryptographic protocol verification, Logic, Public-key cryptography, Security protocols, Deductive reasoning, Authentication

Abstract: A number of techniques based on logic theories have recently been developed to provide formal verification of security protocols. Many of these are based on logics of belief, which are considered useful in evaluating the trust which may be placed in a security protocol. Other techniques are based on logics of knowledge, which are suitable for proving protocol security. A new logic is proposed in the paper for formally analysing public-key protocols. The logic, which combines the logics of knowledge and belief, enables the analysis of the security and trustworthiness of a wide range of security protocols. Axioms are provided which express the low level properties of public-key protocols. These axioms can be combined, using inference rules, in attempting to deduce the desired goals for specific protocols. The paper presents the language syntax for the logic, and a description of the axioms and inference rules. An example of the use of the new logic, in analysing a well known peer-entity authentication protocol, is also described.

1 Introduction

Traditionally, cryptographic protocols have been designed and verified using informal and intuitive techniques. However, an absence of formal verification can lead to flaws and security errors remaining undetected in a protocol. An example of such a security flaw can be found in the CCITT X.509 three-way strong authentication protocol [1], which has been shown to be defective in [2].

Formal verification aims at providing a rigid and thorough means of testing the correctness of a cryptographic protocol so that even subtle defects can be uncovered. A number of formal techniques have recently been developed for this purpose and can be broadly divided into two categories: (i) testing techniques and (ii) logical techniques. Testing techniques, generally comprising exhaustive testing or scenario analysis, involve examining the protocol in search of security breaches. Logical techniques involve a process of deductive reasoning, whereby the desired protocol goals are deduced by applying a set of axioms and inference rules to the assumptions and message exchanges of the protocol.

Many of the current logical methods for formal protocol verification are based on logics of belief [2–5], and logics of knowledge [6, 7]. Logics of belief are useful in evaluating the trustworthiness of a protocol and logics of knowledge are suitable for proving protocol security [8]. Many of the current logic techniques offer limited scope of application. The techniques using logics of belief are generally limited to analysing authentication protocols. Some logics of knowledge tend to offer high level theorems and lack flexibility in analysing a wide range of security properties.

In this paper, a new logical technique is developed which can be used to reason about public-key cryptographic protocols. The technique combines the modal logics of knowledge and belief. Axioms are used to model the low level properties of cryptographic communication systems such as: (i) data encryption and decryption, (ii) the transmission and reception of network data and (iii) the inability of a principal to decrypt a message without knowing the appropriate private key. These properties can be combined using inference rules to allow analysis of a wide range of public-key cryptographic protocols. The constructs of the logic are general purpose, enabling the user to deduce his own theorems, thus allowing for increased flexibility.

2 Protocol analysis using a logic theory

Protocol validation using a formal logic theory can be accomplished by deductive reasoning based on the 'application of valid rules of inference to sets of valid axioms' [9]. Once the logical syntax, rules of inference and axioms have been specified, the deduction process proceeds as follows:

- Formally express protocol steps in the language of the logic
- Formally express the desired protocol goals
- Starting with the initial protocol assumptions, build up logical statements after each protocol step using the logical axioms and inference rules

IEE Proceedings online no. 19960838

Paper first received 9th April 1996 and in revised form 10th September 1996

T. Coffey is with the Department of Electronic & Computer Engineering, University of Limerick, Ireland

P. Saidha is with Motorola Ltd, Mahon Industrial Estate, Blackrock, Co. Cork, Ireland

• Compare these logical statements with the desired protocol goals to see if the goals are achieved.

It is important that the protocol goals be correctly formulated. If the desired goals have not been achieved, then this generally points to a missing hypothesis in the protocol assumptions or the presence of some protocol flaw. If a protocol goal is inaccurate or unaccounted for, then the verification cannot succeed. In this way, a formal analysis not only assesses the reliability of a protocol, but also compels a designer to formally and unambiguously state the protocol assumptions and desired goals.

3 The proposed logic

3.1 Introductory comments

The proposed logic provides a means of verifying public-key cryptographic protocols. The logic can analyse the evolution of both knowledge and belief during a protocol execution and is therefore useful in addressing issues of both security and trust. The logic provides a belief operator and two knowledge operators. One knowledge operator is propositional and deals with the knowledge of statements or facts. The other knowledge operator is a predicate and deals with the knowledge of objects (e.g. cryptographic keys, ciphertext data, etc.).

The inference rules provided are the standard inferences required for natural deduction. The axioms of the logic are sufficiently low level to express the fundamental properties of public-key cryptographic protocols, such as the ability of a principal to encrypt/decrypt based on knowledge of a cryptographic key. The axioms reflect the underlying assumptions of the logic which are as follows:

(i) The communication environment is assumed to be hostile, as defined by Bieber [6], and subject to attack. The data communication system itself is assumed to be reliable, so that message loss and transmission errors will not occur without interference from a hostile party.

(ii) The public-key cryptosystem is ideal. That is, the encryption and decryption functions are completely noninvertible without knowledge of the appropriate cryptographic key and are invertible with knowledge of the appropriate cryptographic key, so that the following equations hold true:

$$d(e(x, k_\Sigma), k_\Sigma^{-1}) = x \quad \text{and} \quad e(d(x, k_\Sigma^{-1}), k_\Sigma) = x$$

The cryptosystem is collision-free, so that it is not possible to create the same ciphertext from two different pieces of plaintext. Also, the probability of making a 'dumb-luck' discovery in relation to a key or a piece of ciphertext is negligibly small.

(iii) A public key used by the system is considered valid if it has not exceeded its validity periods and the corresponding private key is known only to its rightful owner.

(iv) If a piece of data is encrypted or decrypted, then the entity which performed the encryption or decryption must know that data. Since only one entity may know a private decryption key, then if x is decrypted using the private key k_Σ^{-1}, then Σ must know x.

3.2 The logic language

The language of a logic is used to formally express the logical postulates and protocol facts. The language for our logic is as follows:

a, b, c, ..., general propositional variables

Φ, an arbitrary statement

Σ and Ψ, arbitrary entities

i and j, range over entities

ENT, the set of all possible entities

k, a cryptographic key. In particular, k_Σ is the public key of entity Σ and k_Σ^{-1} is the corresponding private key of entity Σ

t, t', t'' ... time

$e(x, k_\Sigma)$, encryption function, encryption of x using key k_Σ

$d(x, k_\Sigma^{-1})$, decryption function, decryption of x using key k_Σ^{-1}

K, propositional knowledge operator of Hintikka [10]. $K_{\Sigma,t}\Phi$ means Σ knows statement Φ at time t.

L, knowledge predicate. $L_{\Sigma,t}x$ means Σ knows and can reproduce object x at time t.

B, belief operator. $B_{\Sigma,t}\Phi$ means Σ believes at time t that statement Φ is true.

C, 'contains' operator. $C(x, y)$ means that the object x contains the object y. The object y may be cleartext or ciphertext in x.

S, emission operator. $S(\Sigma, t, x)$ means Σ sends message x at time t.

R, reception operator. $R(\Sigma, t, x)$ means Σ receives message x at time t.

The language includes the classical logical connectives of conjunction ($\wedge$), disjunction ($\vee$), complementation ($\neg$) and material implication ($\rightarrow$). The symbols $\forall$ and $\exists$ denote universal and existential quantification, respectively. $\in$ indicates membership of a set and / denotes set exclusion. The symbol $\vdash$ denotes a logical theorem. The logic does not contain specific temporal operators, but the knowledge, belief and message transfer operators are time-indexed. Because of space limitations, the semantics for these logical components are not included in this paper, but can be found in [11].

3.3 The logic postulates

A logical deduction system makes use of a set of axioms and inference rules to formulate proofs. Rules of inference allow the deduction of new statements from statements that are already known. A rule of inference consists of a number of premiss statements and the conclusions which can be inferred from these. Axioms are statements which model various facts relating to the logic and to the systems under analysis (cryptographic protocols in our case).

3.3.1 Rules of inference: The logic incorporates the following rules of inference [12, 13]:

R1: From $\vdash p$ and $\vdash (p \rightarrow q)$ infer $\vdash q$

R2: (*a*) From $\vdash p$ infer $\vdash K_{\Sigma,t}p$

(*b*) From $\vdash p$ infer $\vdash B_{\Sigma,t}p$

R1 is the *modus ponens* and states that if schema p can be deduced and $(p \rightarrow q)$ can be deduced, then q can also be deduced. R2 consists of the generalisation rules, which state that if p is a theorem, then knowledge and belief in p are also theorems.

The logic also includes the following standard propositional rules of natural deduction [14]:

R3: From $(p \wedge q)$ infer p

R4: From p and q infer $(p \wedge q)$

R5: From p infer $(p \vee q)$

R6: From $\neg(\neg p)$ infer p

R7: From (from p infer q) infer $(p \rightarrow q)$

3.3.2 Axioms: Axioms are statements of truth that apply to a logic theory and are of two types. Logical axioms are general statements which can be made in relation to any system. Nonlogical axioms are formal expressions of the assumptions of a particular system. In the case of the logic theory being developed here, the nonlogical axioms are the hypotheses underlying public-key cryptographic communication protocols.

(*a*) *Logical Axioms*: The logic includes the following standard modal axioms for knowledge and belief:

A1: (*a*) $\exists\, t\, \exists\, p\, \exists\, q(K_{\Sigma,t}p \wedge K_{\Sigma,t}(p \rightarrow q) \rightarrow k_{\Sigma,t}q)$

(*b*) $\exists\, t\, \exists\, p\, \exists\, q(B_{\Sigma,t}p \wedge B_{\Sigma,t}(p \rightarrow q) \rightarrow B_{\Sigma,t}q)$

A2: $\exists\, t\, \exists\, p(K_{\Sigma,t}p \rightarrow p)$

The axioms A1(*a*) and A1(*b*) are applications of the *modus ponens* to the knowledge and belief operators. The axiom A2 is called the knowledge axiom and is said to logically characterise knowledge. If something is known, then it is true. This property distinguishes between knowledge and belief. For further discussion on other axioms which can be used to characterise knowledge see [12].

A3: (*a*) $\exists\, t\, \exists\, x\, \exists\, i, i \in \{\text{ENT}\}\ (L_{i,t}x \rightarrow \forall\, t', t' \geq t\ L_{i,t'}x)$

(*b*) $\exists\, t\, \exists\, x\, \exists\, i, i \in \{\text{ENT}\}\ (K_{i,t}x \rightarrow \forall\, t', t' \geq t\ K_{i,t'}x)$

(*c*) $\exists\, t\, \exists\, x\, \exists\, i, i \in \{\text{ENT}\}\ (B_{i,t}x \rightarrow \forall\, t', t' \geq t\ B_{i,t'}x)$

The axioms A3(*a*), A3(*b*) and A3(*c*) relate to the monotonicity of knowledge and belief, which asserts that knowledge and belief, once gained, cannot be lost.

A4: $\exists\, t\, \exists\, x\, \exists\, y(\exists\, i, i \in \{\text{ENT}\}\ L_{i,t}y \wedge C(y, x) \rightarrow \exists\, j, j \in \{\text{ENT}\}\ L_{j,t}x)$

The final logical axiom is concerned with the C predicate. If a piece of data is constructed from other pieces of data, then each piece of data involved in the construction must be known to some entity.

(*b*) *Nonlogical axioms*: The nonlogical axioms (A5–A10) reflect the underlying assumptions of the logic, which were outlined in Section 3.1. These assumptions relate to the emission and reception of messages and to the use of encryption and decryption in these messages.

A5: $\exists\, t\, \exists\, x(S(\Sigma, t, x) \rightarrow L_{\Sigma,t}x \wedge \exists\, i, i \in \{\text{ENT}/\Sigma\}\ \exists\, t', t' > t\ R(i, t', x))$

The emission axiom (A5) states that: if Σ sends a message x at time t, then Σ knows x at time t and some entity i other than Σ will receive x at time t' subsequent to t.

A6: $\exists\, t\, \exists\, x(R(\Sigma, t, x) \rightarrow L_{\Sigma,t}x \wedge \exists\, i, i \in \{\text{ENT}/\Sigma\}\ \exists\, t', t' < t\ S(i, t', x))$

The reception axiom (A6) states that: if Σ receives a message x at time t, then Σ knows x at time t and some entity i other than Σ has sent x at time t' prior to t.

A7: (*a*) $\exists\, t\, \exists\, x\, \exists\, i, i \in \{\text{ENT}\}\ (L_{i,t}x \wedge L_{i,t}k_\Sigma \rightarrow L_{i,t}(e(x, k_\Sigma)))$

(*b*) $\exists\, t\, \exists\, x\, \exists\, i, i \in \{\text{ENT}\}\ (L_{i,t}x \wedge L_{i,t}k_\Sigma^{-1} \rightarrow L_{i,t}(d(x, k_\Sigma^{-1})))$

Axioms A7(*a*) and A7(*b*) refer to the ability of an entity to encrypt or decrypt a message when it has knowledge of a public or private cryptographic key.

A8: (*a*) $\exists\, t\, \exists\, x\, \exists\, i, i \in \{\text{ENT}\}\ (\neg L_{i,t}k_\Sigma \wedge \forall\, t', t' < t \neg L_{i,t''}(e(x, k_\Sigma)) \wedge \neg(\exists\, y(R(i, t, y) \wedge C(y, e(x, k_\Sigma)))) \rightarrow \neg L_{i,t}(e(x, k_\Sigma)))$

(*b*) $\exists\, t\, \exists\, x\, \exists\, i, i \in \{\text{ENT}\}\ (\neg L_{i,t}k_\Sigma^{-1} \wedge \forall\, t', t' < t \neg L_{i,t'}(d(x, k_\Sigma^{-1})) \wedge \neg(\exists\, y(R(i, t, y) \wedge C(y, d(x, k_\Sigma^{-1})))) \rightarrow \neg L_{i,t}(d(x, k_\Sigma^{-1})))$

Axioms A8(*a*) and A8(*b*) refer to the impossibility of encrypting or decrypting a message without knowledge of the correct key. Axiom A8(*a*) states that if an entity does not know k_Σ at t and does not know, prior to t, the encryption $e(x, k_\Sigma)$ and also does not receive $e(x, k_\Sigma)$ at t in a message, then the entity cannot know $e(x, k_\Sigma)$ at time t. Axiom A8(*b*) makes a similar statement for the decryption of a message x without knowledge of the decryption key.

A9: $\forall\, t(\forall\, i, i \in \{\text{ENT}\}\ L_{i,t}k^{-1} \wedge \forall\, j, j \in \{\text{ENT}/i\}\ \neg L_{j,t}k_i^{-1})$

The key secrecy axiom (A9) states that the private keys used by the system are known only to their rightful owners.

A10: $\exists\, t\, \exists\, x(\exists\, i, i \in \{\text{ENT}\}\ L_{i,t}d(x, k_\Sigma^{-1}) \rightarrow L_{\Sigma,t}x$

Axiom A10 refers to the fact that a private key owner must know any data which has been decrypted using that private key.

4 An application of the proposed logic

In this Section, as an example, the new logic is demonstrated in the analysis of a well known public-key protocol. The logical analysis of the Needham–Schroeder [15] authentication protocol reveals a known flaw [2] in the protocol. The analysis also detects a protocol assumption which is generally not specified.

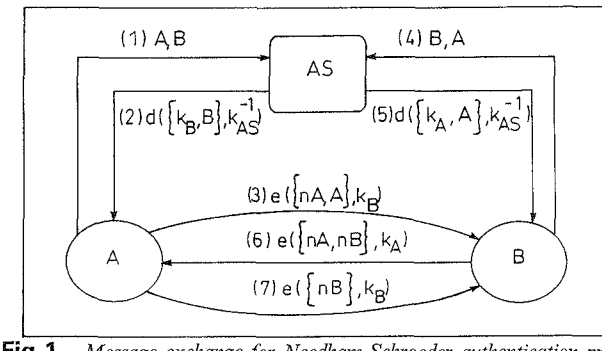

Fig. 1 *Message exchange for Needham-Schroeder authentication protocol*

In the Needham–Schroeder authentication scheme, two principals, A and B, obtain each other's public keys from a trusted authentication server AS, and exchange two secret nonces, *nA* and *nB*, for the purpose of mutual authentication. The message exchanges are illustrated in Fig. 1. It is assumed that both A and B know the public key k_{AS} of the AS, and that they trust the AS to distribute good public keys. That is, if a principal knows that a message has just been sent to it by AS, then it will trust the public key contained in that message. The objectives of messages 1, 2, 4 and 5 are to provide A and B with each other's public keys. Messages 3, 6 and 7 exchange the secret nonce information between A and B. Message 6 should satisfy A that its origin was B, and similarly message 7 should satisfy B that its origin was A.

In the analysis which follows, *tn* refers to the time at the end of protocol step *n*. Thus, *t*0 is the time at the start of the protocol execution and *t*7 is the time at the protocol conclusion. We first formally state the

protocol goals:

(i) $K_{A,t2}(\exists\ t,\ t0 < t < t2,\ S(\mathrm{AS},\ t,\ d(\{k_B,\ B\},\ k_{AS}{}^{-1})))$

(ii) $K_{B,t5}(\exists\ t,\ t0 < t < t5,\ S(\mathrm{AS},\ t,\ d(\{k_A,\ A\},\ k_{AS}{}^{-1})))$

(iii) $K_{A,t6}(\exists\ t,\ t0 < t < t6,\ S(\mathrm{B},\ t,\ e(\{nA,\ nB\},\ k_A)))$

(iv) $K_{B,t7}(\exists\ t,\ t0 < t < t7,\ S(\mathrm{A},\ t,\ e(\{nB\},\ k_B)))$

Goals (i) and (ii) are concerned with A and B obtaining each other's public keys from AS. Messages 2 and 5 must originate from AS and be timely (i.e. sent in the current protocol execution). Goals (iii) and (iv) deal with the nonce exchanges. Messages 6 and 7 must be timely transmissions from authentic principals B and A, respectively.

The *initial assumptions* relating to the protocol participants are outlined as follows:

(i) $L_{A,t0}(k_{AS})$

(ii) $L_{B,t0}(k_{AS})$

(iii) $L_{AS,t0}(k_A)$

(iv) $L_{AS,t0}(k_B)$

(v) $K_{A,t0}(\forall\ i,\ i \in \{\mathrm{ENT}\},\ \forall\ t,\ t < t0,\ \neg\ L_{i,t}nA)$

(vi) $K_{B,t0}(\forall\ i,\ i \in \{\mathrm{ENT}\},\ \forall\ t,\ t < t0,\ \neg\ L_{i,t}nB)$

Both A and B know the public key of AS, and likewise AS knows the public keys of both A and B. Also, A knows that no other entity knows its challenge nonce *nA* at the start of the protocol execution and B knows that no other entity knows *nB*.

We now examine the message exchanges in order to verify the four desired protocol goals. In the interest of brevity, we do not provide complete details of the analysis here. We wish only to demonstrate the general nature of analysis using our technique.

If A receives the message of step 2, then the following statement holds true:

$$\mathrm{K}_{\mathrm{A},t2}(\mathrm{R}(\mathrm{A}, t2, d(\{k_{\mathrm{B}}, \mathrm{B}\}, k_{\mathrm{AS}}^{-1}))) \tag{1}$$

By application of inference rule R3 to axiom A6, expr. 1 can be used to deduce that

$$\mathrm{K}_{\mathrm{A},t2}(\exists i, i \in \{\mathrm{ENT/A}\}, \exists t, t < t2, \mathrm{S}(i, t, d(\{k_{\mathrm{B}}, \mathrm{B}\}, k_{\mathrm{AS}}^{-1}))) \tag{2}$$

Since, by axiom A9, only AS knows the private key $k_{AS}{}^{-1}$, then by manipulation of axiom A8(*b*), expr. 2 becomes

$$\mathrm{K}_{\mathrm{A},t2}(\exists t, t < t2, \mathrm{S}(\mathrm{AS}, t, d(\{k_{\mathrm{B}}, \mathrm{B}\}, k_{\mathrm{AS}}^{-1}))) \tag{3}$$

By applying a similar process to step 5, the following statement can be deduced:

$$\mathrm{K}_{\mathrm{B},t5}(\exists t, t < t5, \mathrm{S}(\mathrm{AS}, t, d(\{k_{\mathrm{A}}, \mathrm{A}\}, k_{\mathrm{AS}}^{-1}))) \tag{4}$$

It can be seen that exprs. 3 and 4 are comparable with protocol goals (i) and (ii) above, except that the time range is not restricted to being after *t*0. This represents a known flaw in the Needham–Schroeder protocol. A and B cannot be sure of the timeliness of messages 2 and 5, and are therefore unable to distinguish between authentic messages sent by the AS and replays of previous messages. To overcome this problem, it is necessary to establish the timeliness of the message, perhaps through the use of random nonces or time stamps.

Let us assume for now that such measures have been taken and that both principals are aware of each other's good public keys. If A receives the message of step 6, then the following statement holds true:

$$\mathrm{K}_{\mathrm{A},t6}(\mathrm{R}(\mathrm{A}, t6, e(\{nA, nB\}, k_{\mathrm{A}}))) \tag{5}$$

By application of inference rule R3 to axiom A6, expr. 5 can be used to deduce that

$$\mathrm{K}_{\mathrm{A},t6}(\exists i, i \in \{\mathrm{ENT/A}\}, \exists t, t < t6, \mathrm{S}(i, t, e(\{nA, nB\}, k_{\mathrm{A}}))) \tag{6}$$

Since, by initial assumption (v), no entity knows *nA* prior to *t*0, then by axiom A4, expr. 6 becomes

$$\mathrm{K}_{\mathrm{A},t6}(\forall i, i \in \{\mathrm{ENT}\}, \forall t, t < t0, \neg \mathrm{L}_{i,t} e(\{nA, nB\}, k_{\mathrm{A}})) \tag{7}$$

By manipulation of axiom A5, expr. 7 yields

$$\mathrm{K}_{\mathrm{A},t6}(\forall i, i \in \{\mathrm{ENT}\}, \forall t, t < t0, \neg \mathrm{S}(i, t, e(\{nA, nB\}, k_{\mathrm{A}}))) \tag{8}$$

Combining the time dependencies of exprs. 6 and 8 gives the following:

$$\mathrm{K}_{\mathrm{A},t6}(\exists i, i \in \{\mathrm{ENT/A}\}, \exists t, t0 < t < t6, \mathrm{S}(i, t, e(\{nA, nB\}, k_{\mathrm{A}}))) \tag{9}$$

By applying a similar process to step 7, the following statement can be deduced:

$$\mathrm{K}_{\mathrm{B},t7}(\exists i, i \in \{\mathrm{ENT/B}\}, \exists t, t0 < t < t7, \mathrm{S}(i, t, e(\{nB\}, k_{\mathrm{B}}))) \tag{10}$$

Exprs. 9 and 10 do not satisfy the requirements of protocol goals (iii) and (iv). While the timeliness of messages 6 and 7 has been established, their origins are not known. All that is known is that message 6 is sent by some entity other than A and message 7 by some entity other than B. To ensure that message 6 has originated at B, it would be necessary to show that

$$\mathrm{K}_{\mathrm{A},t6}(\forall i, i \in \{\mathrm{ENT/(A,B)}\}, \forall t, t < t6, \neg \mathrm{S}(i, t, e(\{nA, nB\}, k_{\mathrm{A}}))) \tag{11}$$

To show that expr. 11 is correct, we need to be able to assert that

$$\mathrm{K}_{\mathrm{A},t6}(\forall i, i \in \{\mathrm{ENT/(A,B)}\} \neg \mathrm{L}_{i,t6} nA) \tag{12}$$

If no entity other than A or B knows *nA*, then no entity other than A or B can know $e(\{nA, nB\}, k_A)$ or send it as a message. However, it is not possible to assert expr. 12 without making some assumptions. The problem is that A has released *nA* to B in message 3. Therefore, if A is to know that only itself and B are aware of *nA*, it must know that B has not divulged *nA* to any other entity. It is not possible for A to know this. However, A may justifiably trust B to keep *nA* secret, in which case, the following additional initial assumption is obtained:

$$\mathrm{B}_{\mathrm{A},t0}(\forall t, t0 < t < t6, \mathrm{S}(\mathrm{B}, t, m) \wedge \mathrm{C}(\mathrm{m}, nA) \rightarrow m = e(x, k_{\mathrm{A}}) \wedge \mathrm{C}(x, nA)$$

... assumption (vii). A situation of trust such as this allows the establishment of belief rather than knowledge. The assertion is that A believes that any time B sends a message containing *nA*, B will encrypt that message with k_A. Using this assertion and the fact that A can know that it does not itself divulge *nA* to any entity other than B during a protocol run, it is possible to deduce a variation of expr. 12:

$$\mathrm{B}_{\mathrm{A},t6}(\forall i, i \in \{\mathrm{ENT/(A,B)}\} \neg \mathrm{L}_{i,t6} nA) \tag{13}$$

Combining exprs. 9 and 13 results in

$$\mathrm{B}_{\mathrm{A},t6}(\exists t, t0 < t < t6, \mathrm{S}(\mathrm{B}, t, e(\{nA, nB\}, k_{\mathrm{A}}))) \tag{14}$$

Similarly, for message 7, the following additional initial assumption can be made:

$$\mathrm{B}_{\mathrm{B},t0}(\forall t, t0 < t < t7, \mathrm{S}(\mathrm{A}, t, \mathrm{m}) \wedge \mathrm{C}(\mathrm{m}, nB) \rightarrow \mathrm{m} = e(x, k_{\mathrm{B}}) \wedge \mathrm{C}(x, nB)$$

... assumption (viii). This leads to the deduction of the

following statement:

$$B_{B,t7}(\exists t, t0 < t < t7, S(A, t, e(\{nB\}, k_B))) \quad (15)$$

Exprs. 14 and 15 show that goals (iii) and (iv) of the protocol must be changed to reflect the assumption of trust in the protocol. We have established belief in the origin and timeliness of messages 6 and 7 rather than knowledge.

As a result of this analysis, we have found that the protocol contains some security flaws and requires some additional measures in order to fulfil its objectives. Protocol steps 2 and 5 are vulnerable to replay attacks which could result in the distribution of invalid public keys. This problem can be overcome by the use of nonces or time stamps to establish the timeliness of the messages. The defect is actually well known and has been described previously in [2].

By our method, we have also been able to identify some hypotheses missing from the initial protocol assumptions. These relate to A and B believing that they can trust each other not to reveal the random nonce information to any other entity. Of course, it can be argued that it would not be in the best interest of either entity to divulge a nonce, but this fact only justifies the additional assumptions and does not exclude the necessity to state them explicitly.

5 Conclusions

In this paper we have presented a new logical technique for verifying public-key cryptographic protocols. We have detailed the language syntax, axioms and inference rules of our logic. By way of an example we have applied our logic to the Needham–Schroeder public-key authentication protocol and we have been able to rediscover a known security defect in this protocol. We have also identified protocol assumptions which are not generally specified.

Our logic allows the representation of both knowledge and belief, and is therefore capable of assessing the security and trustworthiness of a cryptographic protocol. It provides the building blocks with which theorems can be derived as required, thereby enabling our logic to analyse a wide variety of cryptographic protocols.

6 References

1 CCITT: 'The directory — authentication framework'. CCITT Rec. X.509, 1988

2 BURROWS, M., ABADI, M., and NEEDHAM, R.M.: 'A logic of authentication'. DEC System Research Centre Report No. 39, February 1989

3 GONG, L., NEEDHAM, R., and YAHALOM, R.: 'Reasoning about belief in cryptographic protocols'. Proceedings of the 1990 IEEE symposium on *Research in security and privacy*, 1990, pp. 234–248

4 GAARDER, K., and SNEKKENES, E.: 'On the formal analysis of PKCS authentication protocols'. Lect. Notes Comput. Sci., Adv. Cryptol., AUSCRYPT'90, 1990, pp. 106–121

5 KAILAR, R., and GLIGOR, V.D.: 'On belief evolution in authentication protocols'. Proceedings of the Computer Security Foundations workshop IV, 1991, pp. 103–116

6 BIEBER, P.: 'A logic of communication in a hostile environment'. Proceedings of the Computer Security Foundations workshop III, 1990, pp. 14–22

7 SYVERSON, P.: 'A logic for the analysis of cryptographic protocols'. Naval research Laboratory Formal Report 9305, December 1990

8 SYVERSON, P.: 'The use of logic in the analysis of cryptographic protocols'. Proceedings of the 1991 IEEE symposium on *Research in security and privacy*, 1991, pp. 156–170

9 HOARE, C.A.R.: 'An axiometric basis for computer programing', *Commun. ACM*, 1969, **12**, (10), pp. 576–580, 583

10 HINTIKKA, J.: 'Knowledge and belief: an introduction to the logic of two notions' (Cornell University Press, Ithaca, 1962)

11 COFFEY, T., and SAIDHA, P.: 'Semantic definitions for a logic of knowledge and belief'. Report 67/95, Department of Electronic and Computer Engineering, University of Limerick, Ireland, 1995

12 HALPERN, J.Y., and MOSES, Y.: 'A guide to the modal logics of knowledge and belief: preliminary draft'. Proceedings of the 9th international joint conference on *Artificial intelligence*, 1985, pp. 480–490

13 ABADI, M., and TUTTLE, M.R.: 'A semantics for a logic of authentication'. Proceedings of the 10th annual ACM symposium on *Principles of distributed computing*, August 1991, pp. 201–216

14 GALTON, A.: 'Logic for information technology' (Wiley, 1990)

15 NEEDHAM, R.M., and SCHROEDER, M.D.: 'Using encryption for authentication in large networks of computers', *Commun. ACM*, 1978, **21**, (12), pp. 993–999