ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 1, No. 3, 2006, *pp*. 131-138

A Limitation of BAN Logic Analysis on a Man-in-the-middle Attack

Shiping Yang [+], Xiang Li

Computer Software and Theory Institute, Guizhou University, Guiyang, Guizhou, 550025, China

(*Received February 12, 2006, Accepted April 5, 2006*)

Abstract. In recent years a lot of attention has been paid to the use of special logics to analyse cryptographic protocols, foremost among these being the BAN logic. These logics have been successful in finding weaknesses in various cryptographic protocols. With BAN logic analysis on a Station-to-Station (STS) protocol, the paper presents a limitation of BAN logic analysis on a Man-in-the-middle attack, which shows that it is easy for the BAN logic to approve protocols that are in practice unsound and the some enhancements of the BAN logic should be made or in some cases, the informal method will be required in some security protocol analysis like STS. An improved STS protocol against a man-in-the-middle attack is given in the paper.

Keywords: BAN logic, Key Agreement Protocol, Man-in-the-middle Attack, Diffie-Hellman

1. Introduction

When we look into published security protocols, we find that many of these protocols do not succeed in their stated or implied goals. Many existing protocols are susceptible to various kinds of attacks, which are independent of the weaknesses of the cryptosystem employed. In recent years there has been great interest in the design and analysis of secure protocols. Various new techniques have been developed and used to find a great variety of different attacks on such protocols. One of the most important of these techniques is the Logic of Authentication of Burrows, Abadi and Needham [1], (the 'BAN logic') which was the first of several logics (including e.g. AT [2] and GNY [3]) designed to facilitate more rigorous analysis of cryptographic protocols than is possible by informal methods. It allows reasoning about beliefs held by the principals involved in the protocols. BAN logic analysis proceeds by a four-stage process[4]. First the protocol in question is "idealized" — the actual or concrete protocol is expressed as a sequence of formal steps. Second, the set of assumptions under which the protocol operates are identified and formally expressed. Third, the goals of the protocol are identified and formally expressed. Finally, a proof is constructed, using the inference rules of the logic, showing that given the formal assumptions, and upon carrying out one or more protocol steps, the goals are attained. The BAN logic has been used to find new weaknesses in various cryptographic protocols. A number of variations and enhancements of the basic BAN logic have been developed. Gaarder and Snekkenes[5] define two extensions. Firstly, the BAN logic is extended with axioms and rules for Public Key Cryptographic Systems (PKCS). With these extensions, derivations can be made directly. Secondly, the notion of "time" is extended in the logic. Certificates only have a limited life span, which has to be expressed in the analysis.

It has been recognized by the authors of the BAN logic, as well as others, that there are limitations to its power [6]. These limitations can be attributed to its inability to express certain events. That means the lack of precision in moving from a protocol description to its expression in the logic itself – the process called idealization. In this paper we present a STS protocol as an example to show that the BAN analysis can be dangerous in that it allows protocols to be reasoned as secure that are in fact insecure. We do this by showing that certain variations of protocols cannot easily be distinguished in their BAN logic representations, but that this example also serves to re-emphasize the difficulty in designing protocols correctly and the extreme sensitivity of protocols to subtle modifications.

In the third section a Station-to-Station key agreement protocol [7,8] is analyzed using the BAN logic. In the fourth section an attack seems to exhibit a dilemma in practical use of the idealization step of BAN logic

[+] *Email address*: ysp@gzu.edu.cn

analysis, which shows a limitation of BAN logic analysis on man-in-the-middle attacks [9]. We find it possible to idealize a flawed protocol into a good one and the idealization idea is vaguely specified and extremely difficult to apply correctly [10]. The some enhancements of the BAN logic are necessary or in some cases, the informal method will be required in some security protocol analysis like STS. An improved STS protocol against a man-in-the-middle attack is given in the paper.

2. BAN Logic Representations

The BAN logic is a model logic based on belief and can be used in the analysis and design of a cryptographic protocol. The use of a formal language in the analysis and design process can exclude faults and improve the security of the protocol

2.1. Basic Notations

The symbols A, B, P and Q are principals involved in this sort of key agreement protocol; K_{AB} represents a good session key for communication between A and B.

$P \mid\equiv X$: Principal P believes X. P believes as if X is true.

$P \lhd X$: P sees X. A principal has sent P a message containing X.

$P \mid\sim X$: Principal P once said X. P at some time believed X and sent it as part of a message.

$P \Rightarrow X$: Principal P has jurisdiction over X. Principal P has authority over X and is trusted on this matter.

$\#(X)$: The formula X is fresh. That is, X has not been sent in a message at any time before the current run of the protocol. A message that is created for the purpose of being fresh is called a nonce.

$P \overset{K}{\longleftrightarrow} Q$: P and Q may use a shared key K to communicate. The key is good and will always be known only to P and Q and to any other principal trusted by either of them.

$\overset{K}{\mapsto} P$: P has public key K. The corresponding private key is denoted by K^{-1} and assumed to be known only by P.

$\{X\}_K$: X is encrypted using key K.

2.2. Inference Rules

1. Message Meaning Rules

For shared keys:

$$\frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X} \quad (1)$$

If principal P believes that key K is shared only with principal Q, and sees a message X encrypted under a key K it believes only with principal Q. P may conclude that it was originally created by Q who once said its contents.

$$\frac{P \mid\equiv \overset{K}{\mapsto} Q, P \lhd \{X\}_{K^{-1}}}{P \mid\equiv Q \mid\sim X} \quad (2)$$

Similarly, for public keys:

That is, if principal P believes that the key K is Q's public key and it receives a message $\{X\}_{K^{-1}}$ encrypted under Q's corresponding private key K^{-1}, then P may conclude that principal Q once said the contents of the message.

2. Jurisdiction rule

DOI https://doi.org/2024-JICS-22837 | Generated on 2025-04-05 22:12:23

$$\frac{P \mid\equiv Q \Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X} \quad (3)$$

If P believes that P believes that Q believes X, and also believes that Q has jurisdiction over X, then P should believes X too.

3. Nonce Verification Rule

$$\frac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X} \quad (4)$$

If P believes that X is fresh and that Q once said X, then P believes that Q has said X during the current run of protocol, and hence that Q believes X at present. In order to apply this rule, X should not contain any encrypted text. The nonce verification rule is the only way of 'promoting' once said assertion to actual belief.

4. Freshness Conjuncatenation

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)} \quad (5)$$

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(\alpha^X)} \quad (6)$$

If X is fresh, then any message containing X is fresh in virtue of having X in it. But, (X, Y) being fresh tell us nothing about the freshness either of X by itself or of Y by itself (because the whole may be fresh in virtue of the other part).

5. Belief Conjuncatenation

$$\frac{P \mid\equiv X, P \mid\equiv Y}{P \mid\equiv (X, Y)} \quad (7)$$

$$\frac{P \mid\equiv (X, Y)}{P \mid\equiv X} \quad (8)$$

P believes a set of statements if and only believes each individual statement separately.

6. Session key rule

$$\frac{A \mid\equiv \#(K), A \mid\equiv B \mid\equiv X}{A \mid\equiv A \overset{K}{\longleftrightarrow} B} \quad (9)$$

In which with X the necessary elements for a key is meant.

3. BAN Analysis of the STS Protocol

We present a verification of correctness of a Station to Station protocol to show that the BAN logic can be dangerous in that it allows protocols to be reasoned as secure that are in fact insecure.

3.1. STS Protocol Description

The Station-to-Station protocol is a variation on the Diffie-Hellman protocol for key exchange followed by mutual authentication and as follows:

1. $A \rightarrow B : A$

2. $B \rightarrow A : \alpha^{N_B} \bmod p$

3. $A \rightarrow B : A, \alpha^{N_A} \bmod p,\ Sig_A(\alpha^{N_A} \bmod p, \alpha^{N_B} \bmod p)$

4. $B \rightarrow A : B, \alpha^{N_B} \bmod p,\ Sig_B(\alpha^{N_B} \bmod p, \alpha^{N_A} \bmod p)$

The protocol has two parameters p and α. They are both public and may be used by all the users in a

system. Let p is a prime number, α a generator $\alpha \in z_p$. First, A generates a random private value N_A ($0 \le N_A \le p-2$) and B generates a random private value N_B ($0 \le N_B \le p-2$). Then they derive their public values using parameters p and α and their private values. A's public value is $\alpha^{N_A} \bmod p$ and B's public value is $\alpha^{N_B} \bmod p$. They then exchange their public values. Finally, A computes $\overline{K} = ((\alpha^{N_B} \bmod p)^{N_A}) \bmod p = \alpha^{N_B N_A} \bmod p$, and B computes $K = ((\alpha^{N_A} \bmod p)^{N_B}) \bmod p = \alpha^{N_A N_B} \bmod p$. Since $K = \overline{K}$, A and B now have a shared secret key K_{AB} .

The protocol depends on the discrete logarithm problem for its security. It assumes that it is computationally infeasible to calculate the shared secret key $\alpha^{N_A N_B} \bmod p$ given the two public values $\alpha^{N_A} \bmod p$ and $\alpha^{N_B} \bmod p$ when the prime p is sufficiently large. $Sig_X(M)$ is the signature of station X on message M.

3.2. Formal Analysis of the STS Protocol

1. Initial Assumptions

Some messages are signed with the private key of the sender and need to be verified with the public key of the sender in run of the protocol.

$$A \mid\equiv \overset{K_A}{\mapsto} A \tag{10}$$

$$A \mid\equiv \overset{K_B}{\mapsto} B \tag{11}$$

$$B \mid\equiv \overset{K_A}{\mapsto} A \tag{12}$$

$$B \mid\equiv \overset{K_B}{\mapsto} B \tag{13}$$

$$A \mid\equiv B \Rightarrow \alpha^{N_B} \tag{14}$$

$$B \mid\equiv A \Rightarrow \alpha^{N_A} \tag{15}$$

2. Idealized Protocol

The first message is omitted, since it does not affect the logical analysis.

$$B \to A : \alpha^{N_B} \tag{16}$$

$$A \to B : \alpha^{N_A}, \{\alpha^{N_A}, \alpha^{N_B}\}_{K_A^{-1}} \tag{17}$$

$$B \to A : \alpha^{N_B}, \{\alpha^{N_B}, \alpha^{N_A}\}_{K_B^{-1}} \tag{18}$$

3. Protocol Goals

$$A \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B \tag{19}$$

$$B \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B \tag{20}$$

$$A \mid\equiv B \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B \tag{21}$$

$$B \mid\equiv A \mid\equiv A \overset{K_{AB}}{\longleftrightarrow} B \tag{22}$$

These goals can be divided in two groups. First (goals (19) and (20)), both parties believe themselves that the key K_{AB} is a good key for communication between A and B. Secondly (goals (21) and (22)), both entities also believe that other entity believes in the key.

4. Protocol Verification

In message (16) $B \to A : \alpha^{N_B}$, B chooses random N_B and calculates α^{N_B}, then sends α^{N_B} to A:

$$B \mid\equiv N_B \tag{23}$$

$$B \mid\equiv \#(N_B) \tag{24}$$

$$A \triangleleft \alpha^{N_B} \tag{25}$$

In message (17) $A \to B : \alpha^{N_A}, \{\alpha^{N_A}, \alpha^{N_B}\}_{K_A^{-1}}$, A chooses random N_A and calculates α^{N_A}:

$$A \mid\equiv N_A \tag{26}$$

$$A \mid\equiv \#(N_A) \tag{27}$$

$$B \triangleleft \alpha^{N_A}, \{\alpha^{N_A}, \alpha^{N_B}\}_{K_A^{-1}} \tag{28}$$

From (28) and initial assumptions (12), via message meaning rules (2), we obtain:

$$B \mid\equiv A \mid\sim (\alpha^{N_A}, \alpha^{N_B}) \tag{29}$$

From (24), via freshness conjuncatenation (5) and (6), we obtain:

$$B \mid\equiv \#(\alpha^{N_A}, \alpha^{N_B}) \tag{30}$$

From (29) and (30), via nonce verification rule (4), we obtain:

$$B \mid\equiv A \mid\equiv (\alpha^{N_A}, \alpha^{N_B}) \tag{31}$$

From (31) decomposition, we obtain:

$$B \mid\equiv A \mid\equiv \alpha^{N_A} \tag{32}$$

$$B \mid\equiv A \mid\equiv N_A \tag{33}$$

From (32) and initial assumptions (15), via Jurisdiction rule (3), we obtain:

$$B \mid\equiv \alpha^{N_A} \tag{34}$$

From (31) decomposition, we obtain:

$$B \mid\equiv A \mid\equiv \alpha^{N_B} \tag{35}$$

From message (18) $B \to A : \alpha^{N_B}, \{\alpha^{N_B}, \alpha^{N_A}\}_{K_B^{-1}}$, we derive:

$$A \triangleleft \alpha^{N_B}, \{\alpha^{N_B}, \alpha^{N_A}\}_{K_B^{-1}} \tag{36}$$

From (36) and initial assumptions (11), via message meaning rules (2), we obtain:

$$A \mid\equiv B \mid\sim (\alpha^{N_B}, \alpha^{N_A}) \tag{37}$$

From (27), via freshness conjuncatenation (5) and (6), we obtain:0

$$A \mid\equiv \#(\alpha^{N_B}, \alpha^{N_A}) \tag{38}$$

From (37) and (30), via nonce verification rule (4), we obtain:

$$A \mid\equiv B \mid\equiv (\alpha^{N_B}, \alpha^{N_A}) \tag{39}$$

From (39) decomposition, we obtain:

$$A \mid\equiv B \mid\equiv \alpha^{N_B} \tag{40}$$

$$A \mid\equiv B \mid\equiv N_B \tag{41}$$

From (40) and initial assumptions (14), via Jurisdiction rule (3), we obtain:

$$A \mid\equiv \alpha^{N_B} \tag{42}$$

From (39) decomposition, we obtain:

$$A \mid\equiv B \mid\equiv \alpha^{N_A} \tag{43}$$

A calculates session key $K_{AB} = (\alpha^{N_B})^{N_A}$ according to α^{N_B} received from B:

From (25) and (27), via freshness conjuncatenation (6), we obtain:

$$A \mid\equiv \#(K_{AB}) \tag{44}$$

From (44) and (41), via Session key rule (9), we obtain:

$$A \mid\equiv A \xleftrightarrow{K_{AB}} B \tag{45}$$

Due to the symmetry of the protocol, A believes that B is bound to derive same beliefs:

$$A \mid\equiv B \mid\equiv A \xleftrightarrow{K_{AB}} B \tag{46}$$

B calculates session key $K_{AB} = (\alpha^{N_A})^{N_B}$ according to α^{N_A} received from A:

$$B \mid\equiv \#(K_{AB}) \tag{47}$$

From (47) and (24), via Session key rule (9), we obtain:

$$B \mid\equiv A \xleftrightarrow{K_{AB}} B \tag{48}$$

$$B \mid\equiv A \mid\equiv A \xleftrightarrow{K_{AB}} B \tag{49}$$

5. Protocol Results

Through the deduction above, we derive following beliefs：

$$A \mid\equiv A \xleftrightarrow{K_{AB}} B$$

$$A \mid\equiv B \mid\equiv A \xleftrightarrow{K_{AB}} B$$

$$B \mid\equiv A \xleftrightarrow{K_{AB}} B$$

$$B \mid\equiv A \mid\equiv A \xleftrightarrow{K_{AB}} B$$

It expresses that the goals of the protocol can be reached

4. Security Analysis and Improvement

From above verification we can see that STS protocol is safe in BAN logic analysis but the fact does not like this. It is vulnerable to a man-in-the-middle attack as follows:

1. $A \to I_B : A$
2. $I \to B : I$
3. $B \to I : \alpha^{N_B} \bmod p$
4. $I_B \to A : \alpha^{N_B} \bmod p$
5. $A \to I_B : A, \alpha^{N_A} \bmod p,\ Sig_A(\alpha^{N_A} \bmod p, \alpha^{N_B} \bmod p)$
6. $I \to B : I, \alpha^{N_A} \bmod p,\ \ Sig_A(\alpha^{N_A} \bmod p, \alpha^{N_B} \bmod p)$

DOI https://doi.org/2024-JICS-22837 | Generated on 2025-04-05 22:12:23

7. $B \to I : B, \alpha^{N_B} \bmod p,\ Sig_B(\alpha^{N_B} \bmod p, \alpha^{N_A} \bmod p)$

8. $I_B \to A : B, \alpha^{N_B} \bmod p,\ Sig_B(\alpha^{N_B} \bmod p, \alpha^{N_A} \bmod p)$

A tries to start a run of the protocol with B, sending own identity A to B, but I intercepts the message. The attacker I masquerades as A in the protocol and then starts a second run, sending his own identity I to B. This is a man-in-the-middle attack because A thought he was running the protocol with B, while B thought he was running the protocol with I, and may never have heard of A. The consequences are not serious, since the attacker does not learn the value of the key. However, this certainly represents an attack, since it leads to holding incorrect beliefs: A believes that B thought he (B) was talking to A.

This attack is easily prevented, by including both A and B's identity in the signed part of message 3 and 4. The improvement of the protocol is as follows:

1. $A \to B : A$

2. $B \to A : \alpha^{N_B} \bmod p$

3. $A \to B : A, \alpha^{N_A} \bmod p,\ \ Sig_A(A, B, \alpha^{N_A} \bmod p, \alpha^{N_B} \bmod p)$

4. $B \to A : B, \alpha^{N_B} \bmod p,\ Sig_B(B, A, \alpha^{N_B} \bmod p, \alpha^{N_A} \bmod p)$

The improved protocol above does not affect logic analysis process in BAN. The goals of improved protocol are as follows:

$$A \mid\equiv A \xleftrightarrow{K_{AB}} B$$

$$B \mid\equiv A \xleftrightarrow{K_{AB}} B$$

$$A \mid\equiv B \mid\equiv A \xleftrightarrow{K_{AB}} B$$

$$B \mid\equiv A \mid\equiv A \xleftrightarrow{K_{AB}} B$$

They all can be deduced in completely same way. It should be kept in mind that the BAN logic is meant for reasoning over cryptographic protocols. A "verification" with BAN logic does not necessarily imply that no attacks on the protocol are possible. A proof with the BAN logic is a good proof of correctness, based on the assumptions. However, questions may arise over the semantics of the logic and the logic does exclude possible attacks.

5. Conclusions

In this paper, the Station-to-Station protocol is analyzed with BAN logic. It shows what can be done with the BAN logic, but it also shows the imperfections of the BAN logic in analyzing the man-in-the-middle attack, which presents that the BAN logic cannot handle the man-in-the-middle attack and explicit arithmetic in protocols. This attack still needs to be analyzed by means of the informal methods. The nature of formal analysis using BAN depends heavily on the details of the formalization of initial assumptions, and on protocol idealization. The latter appears difficult to prove correct, remains the most critical step. However, verification of the validity of formal assumptions is also essential, as the resulting conclusions are conditional upon them. So, adding describing ability about the intruder in BAN logic, simplifying idealization and providing more detailed handling of cleartext in messages are essential in further research and improvements for BAN.

6. References

[1] M. Burrows, M. Abadi, R. Needham. A logic of authentica-tion. *ACM Transaction on Computer Systems*. 1990, **8**: 18-36.

[2] M. Abadi, M. Tuttle. A semantics for a logic of authentication. *Proc. 1991 ACM Symp. on Principles of Distributed Computing*, 201-216.

[3] L. Gong, R. Needham, R. Yahalom. reasoning about belief in cryptographic protocols. *Proc. 1990 IEEE Symp. on Security and Privacy* (Oakland, CA), 234-248.

[4] P. C. van Oorschot. An alternate explanation of two BAN-logic "failures". *Eurocrypt'93, Springer LNCS*, 1994,

765: 443- 447, 1994.

[5] K. Gaarder and E. Snekkenes. Applying a formal analysis technique to the ccitt x.509 strong two-way authentication protocol. *Journal of Cryptology*, 1991, **3**: 81-98.

[6] R. M. Needham, Reasoning about Cryptography Protocols, *ESORICS 92*.

[7] Jan Wessels. Applications of Ban-Logic. http://www.win.tue.nl/ipa/archive/springdays2001/banwessels.pdf. April 19, 2001.

[8] M. Abadi. Two new attacks on authentication protocols. http://citeseer.ist.psu.edu/abadi97explicit.html. March, 1997.

[9] A. J. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC-Press, Boca Raton, Florida, 1997.

[10] C. Boyd, W. Mao. On a limitation of BAN Logic. http://sky.fit.qut.edu.au/~boydc/papers/euro93.ps