On a Limitation of BAN Logic*

Colin Boyd and Wenbo Mao

Communications Research Group
Electrical Engineering Laboratories
University of Manchester
Manchester M13 9PL
UK

Abstract. In the past few years a lot of attention has been paid to the use of special logics to analyse cryptographic protocols, foremost among these being the logic of Burrows, Abadi and Needham (the BAN logic). These logics have been successful in finding weaknesses in various examples. In this paper a limitation of the BAN logic is illustrated with two examples. These show that it is easy for the BAN logic to approve protocols that are in practice unsound.

1 Introduction

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 [2], (the 'BAN logic') which transforms a protocol into a special form and then uses logical rules to analyse it. 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 [3, 4].

It has been recognised by the authors of the BAN logic, as well as others, that there are limitations to its power [7, 8]. These limitations can be attributed to its inability to express certain events. In this paper we investigate a different kind of difficulty that does not appear to have been discussed before. This is the lack of precision in movin from a protocol description to its expression in the logic itself - the process called *idealisation.*

Cryptographic protocols have not traditionally been expressed in a completely formal manner and so it is inevitable that there must be some conversion of an informal description to a formal description if formal analysis is to take place. However, the BAN logic does not correspond very well to usual formal descriptions of communications protocols such as process algebras [1]. Instead the protocol is reduced to a number of primitives which include the beleifs of principals. According to the BAN authors [2]:

* This work is funded by the UK Science and Engineering Research Council under research grant GR/G19787.

> Only knowledge of the entire protocol can determine the essential logical contents of (each) message.

This seems to us regrettable and to be the cause of difficulties. In our view, messages consist of items of information. Beliefs of principles are updated as a result of messages (information) received, but do not form parts of messages themselves. Thus formalisation should be possible one message at a time. Formal rules to allow this would lead the way to machine assisted analysis of protocols with all its benefits.

In this paper we present examples 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 these variations are critical in deciding whether the protocol is secure or not. These examples also serve to re-emphasise the difficulty in designing protocols correctly and the extreme sensitivity of protocols to subtle modifications.

We would like to emphasise that we are not suggesting that there are inconsistencies in the logical rules defining the BAN logic. What we are pointing out is a difficulty in the practical use of the logic which is equally a problem with other related logics. We cannot suggest practical measures to overcome these difficulties and instead would advocate a different approach altogether where it is desired to obtain a protocols of verified high security.

In the next section two detailed examples are given of failures in the analysis of protocols using the BAN logic, as analysed in the original paper [2]. The two attacks seem to exhibit a dilemma in practical use of the idealisation step of BAN logic analysis. If we regard the BAN logical idealisation technique as easy to apply (as [2] indicated) then straightforwardly following the guidance suggested by [2] for idealising protocols we find it possible to idealise a flawed protocol into a good one; otherwise we would have to say that the idealisation idea is vaguely specified and extremely difficult to apply correctly.

In the final section we discuss possible alternative approaches which may overcome the problems highlighted.

2 The Problem

The two examples we will present both come from the same base protocol, that of Otway and Rees [9]. The BAN logic was used to analyse this protocol [2] and the conclusion reached was that the protocol is basically sound but that there are a number of redundancies.

2.1 Description of the Otway-Rees Protocol

This protocol involves two users A and B and a server S whose role is to pass a new session key, K_{AB}, to A and B. Initially S shares keys K_{AS} and K_{BS} with A and B respectively. The steps in a successful run of the protocol are as follows. Here, and throughout, the notation $\{X\}_K$ indicates the string X encrypted using the key K. In this protocol there is an implicit assumption that a symmetric encryption algorithm is used that provides both authentication and confidentiality of encrypted information.

1. A sends to B: $M, A, B, \{N_A, M, A, B\}_{K_{AS}}$
2. B sends to S: $M, A, B, \{N_A, M, A, B\}_{K_{AS}}, \{N_B, M, A, B\}_{K_{BS}}$
3. S sends to B: $M, \{N_A, K_{AB}\}_{K_{AS}}, \{N_B, K_{AB}\}_{K_{BS}}$
4. B sends to A: $M, \{N_A, K_{AB}\}_{K_{AS}}$

The values N_A and N_B are random *nonce* values chosen by A and B to ensure that their replies from S are new messages and not old ones replayed. The value M is another random nonce chosen by A. It can be seen that A relies on B to relay the messages between her and S. At the end of the protocol it is intended that A and B are both in possession of the shared key K_{AB} and believe it is good for communication with the other. The idealised version of the protocol, as presented by the BAN authors [2], is as follows.

1. A sends to B : $\{N_A, N_C\}_{K_{AS}}$
2. B sends to S : $\{N_A, N_C\}_{K_{AS}}, \{N_B, N_C\}_{K_{BS}}$
3. S sends to B : $\{N_A, (A \overset{K_{AB}}{\leftrightarrow} B), (B \mid\!\sim N_C)\}_{K_{AS}},$
   $\{N_B, (A \overset{K_{AB}}{\leftrightarrow} B), (A \mid\!\sim N_C)\}_{K_{BS}}$
4. B sends to A: $\{N_B, (A \overset{K_{AB}}{\leftrightarrow} B), (A \mid\!\sim N_C)\}_{K_{BS}}$

In this description $P \mid\!\sim X$ and $P \overset{K}{\leftrightarrow} Q$ are parts of the BAN logic notation, the former meaning that the principal P sent the message X at some time in the past, and the latter meaning that K is a key good for communication between principals P and Q. For brevity, the three elements M, A, B have been amalgamated into the nonce N_C. It will be noted that there are various differences between the idealised version and the concrete protocol. Firstly, all cleartext information has disappeared. Secondly, message 3 from S has additional elements in the idealised version which cannot be mapped to any part of the concrete message. These have been added by understanding what the messages are meant to convey as well as the actual message elements. For example, A will associate the nonce N_A with user B and hence concludes that the key is meant for use with B. The analysis in the logic of the idealised protocol reveals a subtle difference between the final beliefs of A and B but concludes that both A and B will believe that the key is good for use with the other.

2.2 A Faulty Implementation of the Otway-Rees Protocol

Consider what is the point of sending the cleartext information M, A, B to S in message 2 of the concrete protocol. As to M, it is referred to in the original paper of Otway and Rees [9] as a *conversation identifier* and so can be used by A and B to identify that the messages they receive from S refer to this instance of the protocol. The reason for sending the names A and B is so that S is able to know what keys to use to decrypt the encrypted parts of the message. S then recovers the contents of these encrypted messages. It is an essential part of the protocol that S now checks the correctness of these messages. In [2] it is stated that S should check "whether the components M,A, and B match in the encrypted messages" and a similar (although more ambiguous) statement is contained in [9].

Thus we choose to make the following interpretation for the precise actions of S.

(a) S uses the cleartext identifiers to choose keys to decrypt the two messages received.
(b) S checks that the fields containing M, A, B are the same in both messages.
(c) S encrypts the new key and respective random numbers using the same keys used in step (a).

Somewhat surprisingly it turns out that if S does only this checking then the protocol is completely insecure. Any user C can masquerade as any other user B by choosing his own nonce N_C and sending the following message as the second one in the protocol.

2. C sends to S: $M, A, C, \{N_A, M, A, B\}_{K_{AS}}, \{N_C, M, A, B\}_{K_{CS}}$

By following the steps performed by S it can easily be seen that the message will be found correct by S and furthermore that C will get the session key K_{AB} encrypted with the key he shares with S, K_{CS}. Once this problem has been seen it is obvious that S must also check that the values M, A, B in the decrypted messages match with the cleartext versions, as well as with each other. It is rather curious to discover that the security of this protocol is critically dependant on cleartext information which can be considered as incidental to the protocol. Indeed, the BAN authors have said [2]:

> We have omitted cleartext information (in idealised protocols) simply because it can be forged, and so its contribution to an authentication protocol is mostly one of providing hints as to what might be placed in encrypted messages.

Our understanding of this guidance on how to perform idealisation leads us to the following dilemma for the user of BAN logic:

1. an unsound protocol, such as Otway-Rees protocol where the actions of S are interpreted by us as above, can be idealised into a "sound" one; the above example is deliberately made to demonstrate this point, or else
2. deleting cleartext should not be conducted as a simple, straightforward and universal treatment for all protocols (as suggested in [2]). Instead, for a flawed protocol like the one given above, the user should have to single it out for an *ad hoc* analysis (as suggested in [10])

We regard the above situation as a dilemma because the user will certainly not be happy with the first dangerous possibility, nor can s/he find it easy to become aware of the unsoundness of a protocol at the time of idealisation (which means to find a bad protocol without using BAN logical manipulation).

2.3 An Attack on a Simplified Version

As a result of their examination of the Otway-Rees protocol the BAN authors noticed that there would be no change in their logical analysis if a number of simplifications were made. In particular they say [2]:

> ...we may notice that there are various forms of redundancy in the protocol. Two nonces are generated by A; however the verification using N_A could just as well have been done using N_C. Therefore, N_A can be eliminated, so reducing the amount of encryption in the protocol. Moreover, it is clear from the analysis that N_B need not be encrypted in the second message.

We now demonstrate that this is definitely not the case. Such a simplification again results in an attack that completely defeats the protocol. According to BAN authors, the protocol attacked is the same as the Otway-Rees protocol except that messages 1 and 2 are now as follows.

1. A sends to B: $M, A, B, \{M, A, B\}_{K_{AS}}$,
2. B sends to S: $M, A, B, \{M, A, B\}_{K_{AS}}, N_B, \{M, A, B\}_{K_{BS}}$

In this attack, the attacker C masquerades as A in the protocol and is also assumed to be in control of communications between B and the server S. The essence of the attack is that C can change the names presented to S while using the nonce that B associates with A. In the version presented here it is assumed that C has possession of a message fragment $\{M, C, B\}_{K_{BS}}$ which was formed by B during a previous legitimate run of the protocol between C and B. (We are making the reasonable assumption that M is a random number selected by A for each run of the protocol. If M were a timestamp a similar real-time attack would be possible.) The attack proceeds as follows, with B and S acting exactly as in a normal run. Messages 2 and 3, which B and S intend for each other, respectively, are captured by C.

1. C sends to B: $M', A, B, \{M, C, B\}_{K_{CS}}$
2. B sends to C: $M', A, B, \{M, C, B\}_{K_{CS}}, N_B, \{M', A, B\}_{K_{BS}}$

2'. C sends to S: $M, C, B, \{M, C, B\}_{K_{CS}}, N_B, \{M, C, B\}_{K_{BS}}$

3. S sends to C: $M, \{M, K_{CB}\}_{K_{CS}}, \{N_B, K_{CB}\}_{K_{BS}}$

3'. C sends to B: $M', \{M, K_{CB}\}_{K_{CS}}, \{N_B, K_{CB}\}_{K_{BS}}$

4. B sends to C: $M', \{M, K_{CB}\}_{K_{CS}}$

At the end of this attack, B believes he shares the key K_{CB} with A whereas he in fact shares it with C. It may be noted that S needs to ignore the replay of

M if the attack is to succeed. Since M is not intended as a nonce for S, who is not supposed to record all of the clients nonces, this is the expected situation.

Looking at the idealised protocol we will see what is wrong this time. According to [2], the idealised protocol should be (note that N_C stands for M, A, B)

1. A sends to B : $\{N_C\}_{K_{AS}}$
2. B sends to S : $\{N_C\}_{K_{AS}}, \{N_C\}_{K_{BS}}$
3. S sends to B : $\{N_C, (A \overset{K_{AB}}{\leftrightarrow} B), (B \mid\!\sim N_C)\}_{K_{AS}}$,
   $\{N_B, (A \overset{K_{AB}}{\leftrightarrow} B), (A \mid\!\sim N_C)\}_{K_{BS}}$
4. B sends to A: $\{N_B, (A \overset{K_{AB}}{\leftrightarrow} B), (A \mid\!\sim N_C)\}_{K_{BS}}$

However, our attacking run shows that the server S has not told B anything like $A \mid\!\sim N_C$. If S really makes any statement to B for the attacking run, it should be $C \mid\!\sim N_C$ and $C \overset{K_{AB}}{\leftrightarrow} B$. As the matter of fact, S can never make such statement because he actually does not read the syntactic specification of a protocol. What the server can read is messages in a *run*, i.e. an *instance*, of the protocol.

We believe that the idealisation scheme of BAN logic has a fundamentally difficulty. This is apparent if we view a protocol specification as analogous to a program specification written in a programming language (e.g. Pascal). The specification only contains *identifiers*, such as principals, nonces, etc. These identifiers will be filled with *real values* during the time of run. Just as inputing a wrong value into a program can result in computational mistakes, running a protocol by filling it with wrong principals or wrong nonces can establish false statements as long as the protocol contains statements about identifiers, just like an idealised protocol under the scheme of BAN logic.

An extension of the BAN logic due to Gong, Needham and Yahalom (GNY) [3] resorted to the same idea of idealisation as BAN. The trivial difference is that in GNY a recipient of a message needs to "convey" a statement of formal parameters from the message.

Here the dilemma for the user of BAN logic still exists, i.e. either to risk the danger of "idealising" a bad protocol into a good one, or find a protocol to be bad without using BAN logical manipulation.

2.4 Implications for BAN logic

Both the above attacks can be easily avoided once they have been spotted. However the idealisation of these protocol is quite reaonsable within the rules defined for the BAN logic. The attacks show that there are instances where the translation depends on subtle factors and may be more difficult than it appears.

If we follow through the logical analysis given in [2] we see that S is able to deduce the origin of the messages sent in message 2. In the attacks, S can correctly deduce that the attacker C sent his part of message 2. However, in the idealised protocol the message that S sends back to the participants says that in fact it was A and B who said the nonce M and that the key sent is good for communication between A and B. We see that this interpretation is quite unjustified in either of the two attack scenarios.

Thus with hindsight it may be argued that the idealisation for the protocols in the two attacks should not in fact be the same as that of the original protocol where the server checks both plaintext and encrypted principal names. But this merely serves to illustrate how difficult the process is to get right since this idealisation was made by the BAN authors themselves. A formal analysis is not very helpful if the protocols it analyses have to be completely understood before analysis can begin.

3 Solutions

We regard the most effective and lasting solution to the problems identified by the above examples to be the development of a new approach to logical analysis of protocols which does not require an idealisation step in the same way as in the BAN approach. Such an approach is detailed in [5].

A second approach is to limit the kinds of protocols under consideration to include only a few message formats whose semantics are well understood. Such an approach is suggested by the lessons learned from formal analyis of computer programs. It is much easier to design a system to be correct in the first place rather than to take some given system and prove that it is correct *post hoc.* Our analysis has revealed that the Otway-Rees protocol is difficult to analyse because the actions of S are unnecessarily complex and the semantics of the messages from S are consequently difficult to pin down. We intend to explore this approach in more detail in future work.

Acknowledgements Anmar Alani carried out the formal specification of the Otway-Rees protocol which led to discovery of the first attack. We are grateful to Professor Roger Needham for helpful comments on an earlier draft of this paper and to Paul van Oorschot for numerous interesting and lively discussions.

References

1. , Colin Boyd, *A Formal Framework for Authentication*, Computer Security - ESORICS 92, pp.273-292, Springer-Verlag, 1992.
2. M.Burrows, M.Abadi, and R.Needham, *A Logic of Authentication*, Proceedings of the Royal Society, Vol A426,pp 233-271,1989.
3. L.Gong, R.Needham & R.Yahalom, *Reasoning about Belief in Cryptographic Protocols*, Proceedings of IEEE Symposium on Research in Security and Privacy, pp.234-248, 1990.
4. R.Kailar & V.D.Gligor, *On Belief Evolution in Authentication Protocols*, Proceedings of IEEE Symposium on Research in Security and Privacy, pp.103-116, 1991.
5. Wenbo Mao & Colin Boyd, *Towards Formal Analysis of Security Protocols*, Proceedings of Sixth IEEE Workshop on Foundations of Computer Security, pp 147-158, June, 1993.

6. Chris Mitchell & Andy Thomas, *Standardising Authentication Protocols based on Public Key Techniques*, circulated within BSI IST/33/-2, 1992.
7. R.M.Needham, *Reasoning about Cryptographic Protocols*, Presented at ESORICS 92.
8. D.M.Nessett, *A Critique of the Burrows, Abadi and Needham Logic*, ACM Operating Systems Review, 24,2,pp.35-38,1990.
9. Dave Otway & Owen Rees, *Efficient and Timely Mutual Authentication* ACM Operating Systems Review, 21,1,pp.8-10, 1987.
10. Paul van Oorschot, An Alternate Explanation of two BAN-logic "failures", Talk delivered at the Rump Session of Eurocrypt 93.