An Alternate Explanation of two BAN-logic "failures"

Paul C. van Oorschot

Bell-Northern Research, P.O. Box 3511 Station C, Ottawa K1Y 4H7 Canada
paulv@bnr.ca

Abstract. Boyd and Mao ("On a Limitation of BAN Logic", in these proceedings) suggest that it is easy to use the authentication logic of Burrows, Abadi and Needham to approve protocols that are in practice unsound, and present two examples. We illustrate that the problem in the first example can be traced to a violation of pre-conditions in the BAN analysis (involving ill-founded trust in a trusted server), while in the second the idealization is simply incorrect. For the latter, a general guideline is proposed to avoid similar problems in the future.

1 Introduction

The BAN logic [3] was the first of several logics (including e.g. AT [1] and GNY [6]) designed to facilitate more rigorous analysis of cryptographic protocols than is possible by informal, ad hoc methods. It allows reasoning about beliefs held by the parties (principals) involved in the protocols. BAN analysis proceeds by a four-stage process. First the protocol in question is "idealized" — the actual or concrete protocol is expressed as a sequence S^* of formal steps ($A \rightarrow B$: X), where A and B are principals and X is a statement in the syntax of the logic. Second, the set of assumptions Q under which the protocol operates are identified and formally expressed. These typically include formalizations of assumptions such as "each principal will not disclose its private keys to other entities". In order to attain the goals established by the formal proofs, these formal assumptions must hold. Third, the goals G of the protocol are identified and formally expressed. A typical goal is the establishment of a cryptographic key shared exclusively with another specifically identified principal. Finally, a proof of the form $Q.S^*.G$ is constructed, using the inference rules of the logic, showing that given the formal assumptions Q, and upon carrying out one or more protocol steps S, the goals G are attained.

It has been suggested that the BAN logic is unable to distinguish secure and flawed versions of some protocols; two illustrative examples were given [2]. We show that in the first example, it is the failure to verify the formal assumptions one would obtain in a detailed BAN analysis that leads to this conclusion, rather than a failure of the BAN technique itself; the true source of the problem is an inappropriate assumption about the trusted server. The problem in the second example shown to be due to incorrect idealization, and a general guideline to follow during BAN idealization is offered to avoid similar problems in the future. In Section 2 we review the Otway-Rees protocol of the first example, and examine the BAN analysis of it. Section 3 discusses the second example — a simplified version of the first protocol. Section 4 concludes this note.

2 BAN analysis of the Otway-Rees protocol

The Otway-Rees protocol examined in [2] is repeated here for reference. S is a trusted server, which generates a symmetric secret key K_{AB} intended for use by A and B. K_{AS} and K_{BS} are symmetric secret keys shared a priori between S and principals A and B, respectively. N_A and N_B are nonces chosen by A and B, intended to allow detection of replayed messages. The field M is not of concern in the present discussion. The messages to be exchanged in a proper run of the protocol are given below. The cleartext identifiers A and B in messages 1 and 2 are used by S to retrieve keys K_{AS} and K_{BS}:

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

A suggested attack on this protocol is as follows. An opponent C impersonates B by intercepting message 2 and substituting his own message in its place, replacing cleartext identifier "B" by "C" (but leaving both enciphered versions of both identifiers "A" and "B" as before), replacing the nonce N_B by his own nonce N, and using key K_{CS} in place of K_{BS}. The opponent is then able to recover the secret key K_{AB} upon intercepting message 3, as it will be encrypted under the key K_{CS} which C shares a priori with S. Whether this attack is successful or not depends on the actions taken by the server S:

Case 1.[1] S simply checks that the values obtained by decrypting the identifier fields (A, B) under the two different keys (K_{AS}, K_{BS}) in message 2 are equal. In this case the attack will succeed.

Case 2. S checks that the values in the cleartext identifier fields (A, B) are equal to the values obtained by decrypting the corresponding identifier fields under each of the keys (K_{AS}, K_{BS}). In this case the attack will not succeed. Clearly this is the desirable version of the protocol.

The analysis given by the BAN logic appears the same in both cases, which, according to [2], suggests a problem in the BAN idealization process. We argue that the BAN logic is indeed capable of distinguishing between these cases. Indeed, consider the following details.

Step 3 of the protocol might be idealized in BAN as

3. $S \rightarrow B$: $\quad \{N_A, A \overset{K_{AB}}{\leftrightarrow} B, \ldots\}_{K_{AS}}, \{N_B, A \overset{K_{AB}}{\leftrightarrow} B, \ldots\}_{K_{BS}}$

Here the symbol $A \overset{K_{AB}}{\leftrightarrow} B$ asserts that K is a good cryptographic key for use by A and B. The portion of the idealization which is not of concern has been deleted ("…"). If a detailed analysis, such as that carried out for the X.509 authentication protocol by Gaarder and Snekkenes [4], is carried out here, then one is forced to record the formal assumption

Q1. $A \models S \Rightarrow A \overset{K_{AB}}{\leftrightarrow} B$

[1] This is discussed in the section "A Faulty Implementation of the Otway-Rees Protocol" in [2].

which states that A *believes* that S *has jurisdiction over* (i.e. can be trusted regarding, or has control over) statements concerning shared keys between principals A and B. This is required because the formal proof of the final goal $A \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B$ requires use of the BAN jurisdiction rule, with $A \overset{K_{AB}}{\leftrightarrow} B$ in the place of X:

$$\text{Jurisdiction rule:} \qquad \frac{A \mid\equiv S \Rightarrow X, \quad A \mid\equiv S \mid\equiv X}{A \mid\equiv X}$$

This inference rule states that if A trusts S on a statement X, and if A believes that S believes X, then A should believe X. Note that assumption Q1 means that A delegates to S responsibility regarding statements about shared keys with B, and trusts S on such matters. Exploring this further with $A \overset{K_{AB}}{\leftrightarrow} B$ in place of X, this means that A trusts S to properly authenticate B; that is, A delegates authentication of B to the server. It should be clear now that in Case 1 (i.e. the flawed version) of the protocol, this trust is ill-founded, and in fact S should *not* be trusted on statements regarding a shared key with B; however in Case 2 (i.e. the secure version), S is trustworthy on this matter.

In summary, the Otway-Rees protocol requires trust in a server S, and the formal BAN analysis properly captures this requirement through assumption Q1. In the flawed version, Q1 is violated, and thus the intended goal is not reachable, i.e. the proof that the goal is reachable is invalidated. Clearly, the BAN approach does not claim to guarantee that formal assumptions always hold, but rather simply asserts that *if* the formal assumptions hold, *then* proofs regarding goals, which use such assumptions, are valid. Proof of the validity of assumptions is beyond the BAN approach itself.

While it would be helpful if the BAN logic provided automatic verification of all identified assumptions, this is an unrealistic expectation of any analysis tool. We submit that assumptions delegating trust to third parties need be carefully examined for validity in any system, and such verification does not appear easily amenable to automation. We note, however, that a more detailed BAN-like analysis of this protocol would replace assumption Q1 with a proof that step 2 in the protocol allows S to properly authenticate B. This then raises the issue that the actions carried out by S should be more clearly specified in the protocol description. This would focus attention on Case 1 vs. Case 2, and again the BAN approach would indeed distinguish the two cases.

3 BAN analysis of a simplified version of the Otway-Rees protocol

Boyd and Mao [2] also consider a simplified version of the Otway-Rees protocol, in which the nonce N_B in message 2 is no longer part of the message encrypted under K_{BS}, but rather simply sent as cleartext information. Under suitable assumptions, they then describe a possible attack, and conclude that the idealization stage of BAN logic has a fundamental difficulty.

While an attack is indeed possible, this conclusion seems unjustified. The attack outlined is through no fault of BAN-like logics or analysis; the protocol as interpreted in [2] is simply flawed, notwithstanding the fact that the BAN authors themselves apparently suggest the modification that leads to it (see [3], p.17). In all fairness, they do so in a brief concluding note, and there is some ambiguity as to exactly what protocol modifications are suggested, what type(s) of nonce(s) and nonce verification are used, and which parties are responsible for verifying nonces. Nonetheless, the issues raised are significant, and we now examine the protocol more carefully. We assume that the server does indeed carry out the check as outlined in Case 2 above. The assumption of trust in the server is then well-founded, and we must search elsewhere for the problem.

With N_B no longer encrypted in message 2, upon reception of message 3 B is still able to conclude that (i) the key K_{AB} is fresh (since it is bound with the nonce N_B); that (ii) the key is known to S; and that (iii) S intends to make the key known to one other party besides B. However, B no longer has any indication who this other party is. The problem is that this simplified protocol does not allow S to convey to B the identity of the other party the key K_{AB} has been made available to. In the original protocol, this is done implicitly, through the cryptographic binding of the nonce N_B to K_{AB} in message 3, and to the pair of identifiers (A, B) in message 2. This allows both B and S to indirectly associate K_{AB} with principal identifiers (A, B). B trusts S to make the key K_{AB} available to only those parties identified in the last two positions in the encrypted segment $\{N_B, M, A, B\}_{K_{BS}}$. However, in the simplified version, no common understanding between S and B is possible regarding the parties associated with nonce N_B (and thus with key K_{AB}), as N_B is not cryptographically bound to any identifiers in message 2 or 3. This prevents S and B from determining a common instantiation of the identifiers "A" and "B" as principals in the symbol $A \overset{K_{AB}}{\leftrightarrow} B$ in idealized message 3 above.

It is now seen that the idealization of message 3 specified above is simply incorrect for the simplified protocol. To avoid such incorrect idealizations in the future, we offer the following remarks as a guideline to be followed during BAN idealization.

In BAN idealization, a key K_{AB} in a concrete protocol is often replaced by the symbol $A \overset{K_{AB}}{\leftrightarrow} B$ in the idealized protocol. It is important to note that this latter symbol implies not only a key, but also the identities of two specific principals. A key denoted K_{AB} sent in a message, e.g. from a server to a principal B, should not be idealized as $A \overset{K_{AB}}{\leftrightarrow} B$ unless it is possible for both the message originator and intended recipient to instantiate the identifiers *(A* and B) either directly (by actual identifiers sent along with and cryptographically bound to K_{AB}) or indirectly (e.g. as through the nonce N_B in the original protocol, or implicitly through use of another shared secret). We suggest that if this is not possible, then the idealization is incorrect and unsound. Note also that the subscript identifier *AB* in a key symbol K_{AB} is typically purely notational, and for the purpose of formal analysis would best be deleted to avoid confusion between the parties *intended* to share this key, and the parties who *actually* end up sharing it.

Finally, note that the *implicit* association of N_B in message 3 to two parties as specified by a binding in message 2, could be made *explicit* if S returned, within the encrypted portions of message 3, the actual identifiers of the parties which S was making the new key available to. This idea was discussed in a preliminary draft of [2].

4 Conclusion

Regarding semantics, the BAN authors state that the assumption $S \models A \overset{K_{AB}}{\leftrightarrow} B$ "indicates that the server initially knows a key which *is to become* [emphasis ours] a shared secret between A and B" ([3], p.16). This differs from the actual definition of $A \overset{K_{AB}}{\leftrightarrow} B$ (A and B may use the shared secret K_{AB} as a good cryptographic key), and illustrates an ambiguity between the intention of sharing a secret key, and the actual state of that key already being shared and/or secret. This a potential cause of problems in many idealizations, including [7], of protocols in which one party is trusted to choose and transport a secret key to another. That this implies that the server is responsible for authenticating these parties is apparently often overlooked in analysis. The guideline offered above attempts to address this in the case of symmetrically generated keys, and a proposal to clarify this confusion in the case of a jointly established key is included in [8].

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 automate or prove correct, and remains the most critical step. However, verification of the validity of formal assumptions is also essential, as the resulting conclusions are conditional upon them. Although appearing straight-forward, analysis by BAN logic does require attention to detail; however we do not believe that it is fatally deficient as suggested in [2]. The cited failures can be linked to the failure to verify the validity of formal assumptions, which to its credit, BAN analysis requires one to record explicitly; and to improper idealization. The latter can be avoided by exercising caution in the use of the symbol $A \overset{K}{\leftrightarrow} B$ in idealization, taking due care to note that this symbol has implications about both the *quality* of the key K as a shared secret, and the *identity* of the parties which supposedly share it. Finally, we note that many of the known weaknesses of BAN-like logics have previously been discussed in the literature [5], along with proposed logic improvements including those aimed at simplifying idealization and providing more detailed handling of cleartext in messages (e.g. see [6]).

5 References

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

[2] C. Boyd, W. Mao, "On a Limitation of BAN Logic", presented at *Eurocrypt'93*, Lofthus, Norway, 1993 May 24-26 (to appear in these proceedings).

[3] M. Burrows, M. Abadi, R. Needham, "A logic of authentication", *ACM Trans. Computer Systems* **8** (Feb. 1990), 18-36. A more detailed version is available in: M. Burrows, M. Abadi and R. Needham, "A logic of authentication", *Digital Systems Research Centre SRC Report #39* (1990 Feb. 22), 62 pages.

[4] K. Gaarder, E. Snekkenes, "Applying a formal analysis technique to CCITT X.509 strong two-way authentication protocol", *J. Cryptology* **3** (Jan. 1991), 81-98.

[5] V. Gligor, R. Kailar, S. Stubblebine, L. Gong. "Logics for cryptographic protocols — virtues and limitations". *Proc. IEEE 1991 Computer Security Foundations Workshop* (Franconia, New Hampshire).

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

[7] D.M.Nessett. "A critique of the Burrows, Abadi and Needham logic". *Operating Systems Review* 24 (1990), 35-38.

[8] P. Van Oorschot. "Extending cryptographic logics of belief to key agreement protocols". *Proc. 1st ACM Conference on Communications and Computer Security* (Fairfax, Virginia, Nov. 3-5 1993).