32
The Descent of BAN

Lawrence C. Paulson
(with apologies to Charles Darwin)

The famous BAN paper [3] determined the research agenda of security-protocol verification for nearly a decade. Many others had worked on verifying security protocols, and the problem appeared to be intractable. The real-world systems were too complicated; too many different things could go wrong; the formal treatments were unusable. The BAN logic was abstract, formalizing intuitive notions directly. For example, if you receive a message containing a secret password and you know that the password is known only to you and Joe, then the message must have come from Joe. BAN proofs were short and simple, and each reasoning step could easily be rendered into plain English.

BAN certainly had some deficiencies. The paper incorrectly claimed that the Otway-Rees protocol could be simplified in a certain way. In fact, an intruder could attack this protocol, masquerading as Bob to Alice, when Bob was not even present [7]. More generally, BAN ignored all non-encrypted information, so it could "verify" any protocol that broadcast the session key in clear. Some criticisms arose from a misunderstanding of the logic's objectives. BAN assumed that the protocol would not give secrets away—a defensible assumption, since cryptanalysts already knew how to investigate such questions. BAN's strength was that it provided a precise notation and deductive mechanism for reasoning about freshness and authenticity.

Researchers introduced a great variety of other authentication logics. These were generally more complicated than BAN. Dietrich [4] published a proof of the Secure Sockets Layer (SSL) protocol using the belief logic NCP (Non-monotonic Cryptographic Protocols). This logic allowed formulae to be retracted as well as asserted, and the author accordingly had to write lengthy lists of facts holding at each step. NCP must have been more precise than BAN, but it was obviously difficult to use. Some people attempted to build automatic provers for the BAN logic, which was pointless: BAN logic proofs were easy to write, and if you wrote them yourself, you were unlikely to reach an absurd conclusion. For the more complicated authentication logics, automation became essential; Brackin [2] was a leading exponent of this approach. As do-it-yourself logics

proliferated, their benefits (especially when applied using an automatic prover) were not always clear. Roger may have been right to call BAN "the original and best."

In hindsight, it is clear that all such logics must share certain limitations. Many attacks on security protocols are not clear-cut, and involve disagreements about the working assumptions. The famous attack on the Needham-Schroeder public-key protocol by Lowe [6] is a classic example. Alice opens a session with Charlie, who proceeds to attack Bob. This scenario involves a misbehaving insider, when the traditional threat model assumes that all criminals are outsiders. Only recently have researchers recognized the danger posed by corrupt insiders.

The failure possibilities of modern protocols are rather complicated. The Zhou and Gollmann non-repudiation protocol [9] is designed to be fair. Its principals are Alice and Bob, who are arranging some sort of contract, and a trusted third party, Clarence. A successful run should give both Alice and Bob sufficient evidence to prove the other's participation. It is also acceptable that neither of the pair should obtain this evidence; however, it is unfair if one of them obtains evidence and the other does not. Gürgens and Rudolph [5] recently demonstrated an attack on this protocol. Alice reuses a session identifier, retaining information from the first protocol run in order to attack a second run. She leaves enough time between the runs to ensure that Clarence will have erased all record of the first run. Alice will be left with evidence confirming Bob's participation. When Bob seeks the corresponding evidence from Clarence, it will not be available.

Formal models typically make ideal assumptions, and in this case would probably endow Clarence with unlimited storage. Alice's attack would then fail. In a more detailed model, Clarence would not be able to store all past session identifiers online, and the attack would succeed. In the real world, Clarence would probably maintain a full audit trail, though most of it would be offline. Whether this attack can succeed or not therefore depends on a detailed description of the dispute resolution mechanism. For this protocol, Gürgens and Rudolph have proposed a neat solution: let Bob contribute to the session identifier. However, we can imagine situations in which algorithms (such as the one for dispute resolution) must be formalized as part of the protocol description. In such situations, authentication logics are unlikely to be helpful, and formal models of any sort are likely to yield misleading results unless the practitioner is aware of the critical issues.

My involvement in protocol verification originated in a research project, funded by the EPSRC, which I held jointly with Roger. The project's original objective was to develop a new authentication logic based upon advanced theory. Through informal discussions (involving Kim Wagner) in Roger's office, I became familiar with the concepts of authentication protocols. I noticed that informal justifications of protocols used inductive reasoning: if X went wrong in step 4, then Y must have happened in step 3, but then Z must have happened in step 2, which is impossible by the nature of step 1. Identifying the first step at which something goes wrong is inductive reasoning, and this underlies the inductive approach to protocol verification [7].

An inductive model has much in common with the models investigated by the Oxford group of Lowe, Roscoe, and Schneider. *Principals* and *messages* are the primitive notions. Messages are recursively constructed from principal names, keys, and nonces by concatenation and encryption. The semantics of a protocol is given by the set of possible traces of *events,* such as the sending and receiving of messages. Such models are far removed from the real world, but more low-level than the BAN models. Roger encouraged this new approach, though it differed radically from his own. He offered advice of the sort that I imagine he offered his research students. He suggested, for example, that I focus attention on a specific message of the Needham-Schroeder shared-key protocol.

Roger's influence, and that of the BAN paper, ensured that my models included the necessary elements. BAN is mainly about freshness: we have received a session key, but how do we know that it is fresh? An old key may have become compromised. One of the BAN paper's most interesting analyses is that of the Yahalom protocol. Here Bob receives in separate packages a session key K (bearing no evidence of freshness) and his nonce NB, encrypted using K. Ordinarily, encryption using a potentially compromised key would yield no firm evidence. However, the Yahalom protocol keeps NB secret; an intruder in possession of K would still be unable to perform the encryption $\{NB\}_K$. Therefore, this message firmly associates NB with K, proving the latter's freshness. BAN formalizes this argument quite easily; in my inductive model of Yahalom, it was much more difficult [8].

Freshness is no less important these days, and protocol designers are careful to include the nonce challenges necessary to achieve it. Recent attacks seldom involve freshness, and many recent formal models do not represent freshness. I have been lucky to work in a research environment that is strong in both theory and computer security. (Roger can be given the credit for creating this environment.) That is how I have been able to avoid some of the mistakes made by researchers who do not work with a security group. If some authors do not understand what a nonce is for, or know that a timestamp should carry a valid time, or appreciate that a certain type of field will always have the same length in bytes, then they should spend time at the Computer Laboratory.

The BAN logic, like many other approaches to analysing security protocols, assumes perfect encryption. This assumption means, in particular, that no information can be deduced from a ciphertext without the corresponding key. Encryption is obviously not perfect, but many protocols are flawed even under this assumption.

The problem of security-protocol verification under perfect encryption is essentially solved. Numerous researchers have worked on it, and even the most complicated protocols have undergone formal scrutiny. Many of today's hard problems concern how to formalize the vulnerabilities of specific encryption methods such as Diffie-Hellman or RSA. Even exclusive-OR is difficult to model, particularly in typed formalisms, because the exclusive-OR of two bit strings can yield data of any type. Probabilistic mechanisms are also difficult to

verify, although recent progress gives ground for optimism. Another difficult area concerns the composition of protocols from separately verified components.

I have heard Roger say that the BAN logic is obsolete. How many researchers would say that about one of their most important achievements? However, even if the BAN logic is obsolete, the BAN paper is certainly not. It remains an excellent tutorial on cryptographic protocols. It describes and analyzes a variety of different protocols. With Roger's other papers, such as Abadi and Needham [1], it remains essential reading for anybody wishing to do research in this area.

References

1. ABADI, M., AND NEEDHAM, R., 'Prudent engineering practice for cryptographic protocols,' *IEEE Transactions on Software Engineering*, vol. 22, no. 1, 1996, pp. 6–15.
2. BRACKIN, S.H., 'A HOL extension of GNY for automatically analysing cryptographic protocols,' in *9th Computer Security Foundations Workshop*. IEEE Computer Society Press, 1996, pp. 62–75.
3. BURROWS, M., ABADI, M., AND NEEDHAM, R.M., 'A logic of authentication,' *Proceedings of the Royal Society of London*, vol. 426, 1989, pp. 233–271.
4. DIETRICH, S., *A formal analysis of the secure sockets layer protocol*. Ph.D. thesis, 1997, Adelphi University, Garden City, New York.
5. GÜRGENS, S., AND RUDOLPH, C., 'Security analysis of (un-)fair non-repudiation protocols.' In A. Abdallah, P. Ryan And S. Schneider, S. (eds.), *Formal aspects of security 2002*. Royal Holloway College, University of London.
6. LOWE, G., 'Breaking and fixing the Needham-Schroeder public-key protocol using CSP and FDR.' In T. Margaria and B. Steffen, (eds.), '*Tools and Algorithms for the Construction and Analysis of Systems*,' *Second International Workshop, TACAS '96*, Lecture Notes in Computer Science 1055, pp. 147–166., Springer, 1996.
7. PAULSON, L.C., 'The inductive approach to verifying cryptographic protocols,' *J. Computer Security*, vol. 6, 1998, pp. 85–128.
8. PAULSON, L.C., 'Relations between secrets: two formal analyses of the Yahalom protocol,' *J. Computer Security,* vol. 9, no. 3, 2001, pp. 197–216.
9. ZHOU, J., AND GOLLMANN, D., 'A fair non-repudiation protocol,' in *Proc. of the 15th IEEE Symposium on Security and Privacy*, 1996, pp 55–61., IEEE Computer Society Press.