REASONING ABOUT CRYPTOGRAPHIC PROTOCOLS

R M NEEDHAM

Rapporteurs: Jason N Bain and Rogerio de Lemos

Reasoning about Cryptographic Protocols

R.M. Needham

Introduction

In this talk I want to reflect on the process of reasoning about cryptographic protocols, to consider what existing formalisms capture and what they don't, and to see what can be fixed up and consider how far it it worth while to go. I shall use as an example the style of reasoning first presented by Burrows, Abadi, and myself several years ago, though it isn't intended to make any exclusive claims for it.

In a paper published in 1978, Michael Schroeder and I (NS) presented various protocols for the use of encryption for authentication an networks of computers. We concluded "Finally, protocols such as those developed here are prone to extremely subtle errors that are unlikely to be discovered in normal operation. The need for techniques to verify the correctness of such protocols is great, and we encourage those interested in such problems to consider this area".

This challenge was not taken up at that time by the theoretical community. It was prudent, however, of us to make the remark. A few years later a subtle error was found in the Needham & Schroeder protocol which consisted of five messages and contained essentially one bug. A decade later the CCITT published a draft standard X.509 for authentication. Its protocol for a similar purpose had three messages and three bugs.

It is not entirely obvious why these things are so hard to get right. Most programs of 3 - 5 lines can be got right if one stares at them long enough. At that length they may even be amenable to actual proof. What is it that is special about the present ones? One purpose of this paper is to indicate possible answers.

The BAN logic

It was more or less by accident that in 1987 Burrows, Abadi, and I (BAN) began to take up the challenge that had been lying on the table for ten years. Abadi was (and is) a proper mathematical logician, and we needed him to keep us sound. We did not usually emphasise the formal semantics of the BAN logic - as it became known - but it was very handy to have them available to refute people who thought the work was of dubious mathematical integrity.

The BAN logic is about belief. The pompous term is that it is a doxastic logic. Knowledge is of course stronger than belief, but the things that matter in cryptographic dialogues can't be known - for example that Joe and I share a secret S. As with the existence of God, I can believe this but not know it. The logic also has to be concerned with the bases on which beliefs grow up - the evidence for addition to the stock of belief. It is possible that a calculus could be constructed which took evidence rather than belief as its basis; to do so might be advantageous because of making it possible to use less anthropomorphic language.

In fact the BAN logic builds on some basic premises that are easily - and informally - stated.

1. If I believe that Joe and I share a secret S, and I see a message which indicates unequivocally that the sender knows S, and I didn't send it myself, then I had better believe that the sender was Joe.

2. If I trust Joe about some class of statements C, and I believe Joe believes σ, and σ belongs to C, I had better believe σ too.

3. If the components of a message are bundled up together is such a way that the whole message must have been assembled at once, then if one component is fresh then they all are.

The notion of "freshness" proved to be a very useful one, though it has led to some minor problems of interpretation. It isn't the content of the message that's fresh, it's the utterance of it. Consider a message which reads, on decryption, "God save the Queen! #1A53FB24" where the latter is a correct nonce. The content of the message is extremely stale; the utterance is fresh.

The BAN logic is all about the detailed explanation and exploitation of these principles, with some other useful abstractions for covering up details. Here's an example of one. Generating a suitable session key for use when communicating between A and B is not a trivial task. It had better be new, it had better be secret, and it had better not be 0^{64}, 1^{64}, or checkerboard, or other obvious patterns (though it's been rumoured that using keys like that isn't as uncommon as it should be). We encapsulate all these features in

$$A<K>B$$

read as "K is suitable for conveying secure and authentic information between A and B". Then for A and B to be happy communicating using K, they had better believe

A<K>B.

But how would they get that belief? Because they are caused to believe that some agency they trust to make statements like

A<K>B

has freshly said just that. It is, of course, quite impossible within the formalism, to reason about whether the trusted agency justifies the trust reposed in it. All that can be done is to list informally a set of desirable properties, as outlined above.

The BAN framework of reasoning sketched here is extremely primitive. It has no negation and almost no quantification. I can almost be summed up in a sentence thus

"Trust and Freshness lead to Current Belief".

It nevertheless made it possible to explain the problems of the Needham and Schroeder protocol, to expose redundancies in the Kerberos protocol, and to expose faults in X.509 - among other things. It is a formal approach which has real scalps at its belt, so to speak.

Limitations of the BAN logic

Like many formalisms the powers of the BAN logic are easy to overstate. When Burrows, Abadi, and I broke a draft X.509 it was *not* done by applying the formalism to the definition, turning the handle, and noting what did not appear. We had spent so much time looking at protocols like that that what we actually did was to write the definition on the board, see the errors, and go to lunch. What the formalism did enable us to do was to express precisely what the problems were, to convince ourselves and (very importantly) others of their reality, and to check out proposed fixes. We heard much later that the standards body thought our fix wasn't quite elegant and replaced it by one that is more symmetrical. This apparently introduced a quite different error - yet the people involved were certainly not inexperienced or unknowledgeable.

The BAN logic has been criticised quite a lot for not doing things it did not try to do (notably ensuring that secrets stay secret); there are other limitations which have mainly been noticed by its authors.

Thus one largely unaddressed point concerns the use of speculative keys. If you (A) see

$\{M\}_k$

where the braces denote encryption, and you believe

A<k>B

then you believe that B once said M. That's straightforward in BAN. BAN however can't handle well the equally plausible "If you receive a message and decrypt it with k and find the expected M, then you should believe A<k>B". In the BAN papers an example of this is discussed in relation to a protocol proposed by R. Yahalom, but the authors sidestep the issue by changing Yahalom's protocol so that the problem doesn't arise. The BAN logic only enables you to make an inference from the result of decrypting a message with a key if you already have good reason to believe that the key is freshly associated with a known principal.

If one is to be able to handle speculative keys, it is necessary to have some idea about what keys may be speculatively tried. It is usually assumed that keys can't be guessed, so that you can only try decrypting a message with key k if you have 'seen' it. This is rather a tricky notion, because keys are often thought of as numbers, and there are well established rules for proceeding from one number to another. It seems a bit odd to say that we are aware of 5 so we can try it as a key, but 6, 7, and 8 are completely strange to us. The BAN authors suggest that you can try anything that somebody said sometime was a key, and then give up.

Gong, Needham, and Yahalom (GNY) attempted to do a better job of capturing this notion of what one has 'seen'. The idea is that you can only decrypt using a trial key if you have 'possess' it. You've 'seen' any bag of bits that comes your way. You 'possess' anything you can get by decrypting or encrypting something you've seen with something else you 'possess'.

Underlying all this is the view that accidents don't happen. If you decrypt a bag of bits and get Lincoln's Gettysburg address then the bag of bits was made by encrypting that most noble text. Probabilities with large enough denominators are treated as zero. This important observation is not always made as prominent as it should be. Allowing for things which have been 'seen' or 'possessed', the GNY logic captures some things that BAN does not, but at the cost of a lot more apparatus per new feature captured. This may of course reflect some lack of talent in the design of the GNY logic, or it may be that the ideas involved really are complicated. We are nevertheless not all the way yet to capturing important features of protocols.

Protocol-Specific Inference

The approaches so far have to do with interpretation of messages with a very sparse view of their context. We have been concerned with inferences that may be made on the basis of the octets of the message and very specific state in the recipient such as beliefs about jurisdiction and freshness, The BAN message meaning rule says

If A believes A<k>B, and A sees $\{x\}_k$, then A believes B once said x.

Now consider a slightly idealised part of the Kerberos protocol. K_{ab} is the session key that is being distributed, and A and B each believe they have shared secrets K_{as} and K_{bs} with S, and trust S to make up K_{ab} properly and to have a good clock.

A->S	A,B	
S->A	$\{K_{ab},B,T,\{K_{ab},A,T\}K_{bs}\}K_{as}$	α
A->B	$\{K_{ab},A,T\}K_{bs}$	β

The BAN logic lets B infer

$A<K_{ab}>B$

but not that A was very recently present. That much can be inferred by B without knowledge of the Kerberos protocol. But if B does know (as we may reasonably assume) that what it gets is the third message of Kerberos, then it knows too that the only way to get things like β which are freshly time-stamped is if A recently decrypted a message such as α, and was thus present in the very recent past. There isn't a lot of point in a further handshake to check that A is around now (unless our freshness criteria are very lax). It is of mild historic interest to note that the BAN authors couldn't see what the point of the double encryption in message 2 was, and said it was redundant. It is in fact the double encryption that justifies the protocol-specific inference. The BAN authors fell into the trap of saying that if their formalism couldn't say something, it wasn't important. They were far from the first, and won't be the last, people to fall into that error when peddling formalisms.

There is no accepted scheme for handling protocol-specific inferences. On the face of it way to proceed is to label messages according to what they are, so that instead of β you have

$\{K_{ab},A,T,K3\}K_{bs}$

where the K3 indicates that this is Kerberos message 3. It then becomes possible to put in a protocol-specific inference rule

if B believes $B<K_{bs}>S$, and B sees $\{K_{ab},A,T,K3\}K_{bs}$

then B believes $B<K_{ab}>A$ and also that A believes $B<K_{ab}>A$.

We can't simply do that however. Where do labels like K3 come from? We shall need to extend our notions of trust to labelling statements as well as uttering them. Not all the consequences of this have been worked out, and again there are questions about how far it is worth going. We shall return to this point.

I mentioned earlier the problem with a more symmetrical X.509. The fix for that is a labelling one, though the question of trust has as far as I know not been resolved.

Ground-Rules

Underlying all this discussion is an assumption that has not been stated, and may appear obvious. We have assumed (as did BAN and GNY) that A and B are honest citizens trying to establish secure and authentic communication in the face of a wicked world. This is very much not so in some cryptographic applications (for example verification of arms control treaties), but one does not have to go to those rather exotic lengths to find examples where it isn't a sensible assumption.

Password Guessing

People are not very good at remembering keys. They are not bad at remembering passwords, though they are pretty bad at choosing them. Traditionally attempts have been made to persuade/force people to have well-chosen passwords, from which encryption keys can be derived by algorithm. It is possible to propose protocols (LGNS) which make good use of (even ill-chosen) passwords, by making it difficult to mount a guessing attack undetected. The principle is to avoid sending any messages which it is possible to copy and experiment with at leisure off-line to see whether you have guessed a password right. If the only way you can see whether you have guessed a password right involves an interaction with "the system", then the system can notice repeated attempts and raise the alarm. (The *locus classicus* of guessing attacks is the UNIX password file, which has all the wrong characteristics.) It is possible (and the referenced paper shows how) to arrange authentication dialogues which are immune to guessing attacks, although they involve everyone knowing a single public key. That can of course safely be written down, or rather recorded magnetically, because of its basic public character. It is believed that one cannot obtain immunity against guessing without at least use of public key cryptography, but nobody knows how to prove this.

The reason for mentioning this subject here is that presumably, if A and B are concerned to set up a secure and authentic channel between themselves by a process which is immune to guessing attacks on their passwords, they do not want to be open to such an attack by their partners any more than by the world at large. Here is an example, and not a very far-fetched one, of lack of trust between the players in an authentication dialogue. Quite a bit of the complexity of the safe protocol comes from the need to protect against insider attacks - the difficulty being of course that the untrusted insider knows the session key, and is thus in a better position than the random citizen to verify a guess. It is possible to set up systematic methods for checking against vulnerability to guessing attacks; it is not a wholly academic matter since the well-known Kerberos protocol has vulnerabilities of this type.

Arbitration

Implicit in many cryptographic protocols is trust in some kind of intermediary. To what extent is it desirable that the trust placed in the intermediary be verifiable? Supposing that one of the clients of the intermediary repudiates a transaction and says the intermediary was at fault, should he have to be met just by blank denial? Or are there properties of protocols that help with this, and should formalisms capture them? I owe this point to Mark Lomas.

Applications

It has sometimes been suggested that all this reasoning apparatus is not really necessary because there will only be a couple of standard authentication dialogues in the world to reason about, and once they are demonstrably OK we may as well shut up shop. This is an incorrect view, though it is unfortunately promoted by the title used for the BAN paper - "*A Logic of Authentication*". It is no accident that the present title is "*Reasoning about Cryptographic Protocols*". A later paper in this Symposium describes a real-life application from the financial world.

What's worth formalising?

Standing back from the details for a moment, how should we see the role of formalism in the study of cryptographic protocols? It is possible to see various models. One approach is to try to construct a logic which will capture every possible aspect of the subject. The GNY effort was a bit like that, and it entails the erection of a vast conceptual apparatus for rather little advance in coverage. The BAN logic has paid off very well in an area where common sense has proved to be a very unreliable guide to successful design. It may be that the BAN logic gets close enough to complete analysis for common sense to be a reasonable guide to understanding what is left. The discussion of protocol-specific reasoning is an example of this effect. We can add to the logic, but only in such an ad hoc way that no evident insight is gained.

References

(NS) R.M. Needham and M.D. Schroeder, "Using encryption for authentication in large networks of computers" CACM December 1978

(LGNS) T.M.A. Lomas, L.Gong, R.M.Needham,& J.H. Saltzer "Reducing risks from poorly chosen keys", Operating Systems Review 23(5) 1989

(BAN) M. Burrows, M. Abadi, & R.M. Needham "A Logic of Authentication" ACM Transactions on Computer Systems, 1990

(GNY) L.Gong, R.M. Needham, & R. Yahalom "Reasoning about belief in cryptographic protocols", IEEE Symposium on Security and Privacy, 1990.

DISCUSSION

Rapporteurs: Jason N Bain and Rogerio de Lemos

Lecture One

During the lecture, Professor Dijkstra asked the speaker what was the meaning of the acronym FDDI. Professor Needham answered that it stands for Fibre Distributed Data Interface which is a 100 Mbits per second token ring. He added that he would not go into any detail because he considered that FDDI was an abortion, even for the purposes that were initially intended.

After the lecture, Professor Tanenbaum made a comment on the issue of FDDI being an abortion. He agreed that FDDI was an abortion, but for totally different reasons. He said that the main problem with FDDI was the size of the driver, typically 20000 lines of C, the same size that UNIX used to be. Professor Tanenbaum made a second comment related to the relative packet size and ATM cells. His argument was: if he has a 1Mbyte file to be transferred from a file server to a workstation, he would like to transfer it as one 1Mbyte packet rather than 20000 packets of size 53 bytes because of the overhead in transmitting packets. He concluded his second comment by stating that the choice of packet size was very much dependent upon the application. On this second comment, Professor Needham replied that it was absolutely hopeless to handle a cell as a packet. If someone wishes to have high performance, a cell has to be below the packet level. If you wish to have high performance packet interfaces for cell networks, then you must perform packet segmentation and re-assembly in the interface itself.

On the subject of standards, Professor Rabin queried the flexibility of technological decisions, such as the one made with regards to the size of ATM cells. For example, if in ten years time these decisions are found to be inappropriate, they would be very difficult to change. Professor Needham answered that the freezing of cell size had probably already happened to some extent. The argument for a shorter packet was defended by the Europeans because they are concerned with the transmission of voice. However, if one starts to use Gbit or multiple Gbit transmission rates, the cell size has to get bigger because in the time it takes to transmit the data, it may be too fast for the intelligence in the system to make decisions about the contents of the cells. He continued by saying that there was a real conflict with whatever logic employed. On the other hand, the size of the cell has to vary while it goes around the world. Concluding his reply, Professor Needham said that he did not know what would happen in the end. However, in his opinion, cells should have more then one size based upon any rate that is chosen so that the gear change from one size to a different size, and back again, would not be too uncomfortable.

Professor McCarthy made the observation that when the ARPANET was designed in the 1970's, it was based on two ideas: the first was a method of communication between computers, the second was packet switching. He added that this has created the mess that still exists today. The symptom of this mess can be characterised by the fact that Fax has completely outrun electronic mail, even though electronic mail is technologically simpler and has many performance advantages. To use a Fax, all one has to do is buy a fax machine, thus allowing one to communicate with any other Fax machine in the world. However, if one wishes to use electronic mail, there are constraints such as compatibility between interconnecting nodes. Professor Needham replied that it was impossible to disagree with what Professor McCarthy had said. He added that for some considerable amount of time he had believed that the packet network exemplified by the ARPANET had had a fine response for the demands of it's time, but did not think it ought to be necessary to have separate networks for that problem. He found the convergence of the computing and communications industry encouraging, as this should eventually lead to the demise of a dedicated computer network.

Lecture Two

During the lecture, Professor Hoare made the remark that the use of formalisms can only show the absence of bugs. Professor Needham said that in the referred case, utilisation of formalisms did show the presence of bugs, but not their location. He added that formalisms were really good for convincing the pro-prizes of the standard that it was wrong. They are effective in explaining the gory detail of what the problem was in a manner that was very difficult to deny. This is also very useful, as one may suspect that something is incorrect about a standard, but one has to be able to communicate this to someone who is willing to share your beliefs.

Professor Rabin asked if within the properties of the "K" (A<K> - A and B are to communicate using the key K) apart from the ones already discussed (to be new, secret, and not obvious), would one need an additional assumption that "K" has to be chosen from a very large range. Professor Needham agreed and said that he should have mentioned this during his lecture.

Professor Hoare asked if Professor Needham regarded formalisms with regard to protocols as being good, but not to programming. Professor Needham replied that this was not the impression he was trying to convey. He thought that over a number of years part of formalisation had gone too far - it had given apparatus that does not do anything useful. Professor Hoare suggested that if the formalisation cannot explain something, then the explicand must be incorrect. Professor Needham continued by saying that he thought backward proofs had given a substantial improvement to computing. He added if you can use formal reasoning, you should use it. It can be used to show the limits of an activity. Professor Needham concluded that formal reasoning should only be used in skilled hands.

Professor Rogers gave an example in using passwords. He said that an insurance company 1.5 years ago noted that of their 7000 users, 2000 had the same password - 'Saddam'. Professor Needham said that passwords were not the way to proceed. He thought that smart cards with PINs should be increasingly used. However, he did not think this would happen until the costs of such cards decreased, as they are much more expensive than normal magnetic stripe cards.