Strong Practical Protocols†

Judy H. Moore
Sandia National Laboratories
Albuquerque, NM 87185

Recent progress in the area of cryptography has given rise to strong cryptoalgorithms using complex mathematical systems. These algorithms often require quite sophisticated computing capabilities for their implementation and are designed to withstand attack by equally sophisticated opponents with nearly unlimited resources available to them. However, the mere existence of such algorithms is not enough to solve the problems of message secrecy and authentication. The procedures for handling the data, including the use of a cryptoalgorithm, must insure that the desired level of security is achieved. Such a set of rules or procedures is known as a cryptographic *protocol*.

There have been many examples in the literature ([2],[3],[6],[7],[10], [11]) in the past few years of protocol failures. These examples sometimes involve the cryptanalysis of the particular instance of the cryptosystem used, but often merely demonstrate the failure of the protocol to provide the advertised level of security and/or authentication.

Research in the area of cryptographic protocols has been considering methods to provide "provably secure" protocols ([1],[4],[5],[8],[9]). Progress has been made, although, as might be expected, the initial results either apply to systems which have been idealized in order to simplify the problem or use techniques to gain security which are not practical for application. These techniques are obtained using abstract principles designed to model ideal security systems. However, when cryptosystems are actually fielded, much less abstract principles are applied. Rather than trying to model ideal systems to provide "provably secure" protocols, we propose the development of techniques for use by protocol designers to precisely define the security provided by the protocol.

Modern systems designed to protect information can make use of cryptography, physical barriers and procedural constraints to accomplish their goals. It is important to note that, in general, elements from each of these three areas are required for optimal

†This work performed at Sandia National Laboratories supported by the U. S. Department of Energy under contract no. DE-AC04-76DP00789.

C. Pomerance (Ed.): Advances in Cryptology - CRYPTO '87, LNCS 293, pp. 167-172, 1988.

protection. Perhaps because the physical barriers seem more concrete, the analysis of their effectiveness appears to be easier than the analysis of the cryptographic portions of the protocol. Such an analysis giveS rise to some broad guidelines for the purpose behind any element in a protection system. In particular, it seems that the value of any security measure lies in its capacity to contain the threat within a well-defined and reasonable boundary. The guardian of an object utilizes a security measure, not because it will provide absolute security, but rather because it can limit the threat and clearly define the boundary of the security problem remaining. For example, the purpose of a barbed wire fence around a secure area is to limit the attention of the guards to that well-defined boundary. It also limits the threat to those willing and able to confront the barbed wire. The guard force is still necessary since the fence cannot provide absolute security. Similarly, the use of a secure encryption algorithm to maintain secrecy in communications between two parties, does not eliminate the need for some information to be kept secret. At best, it transfers the need for protection to the key, which should be smaller and more easily controlled than the messages themselves.

From the above discussion, we abstract the following principle for application to the development and analysis of protocols. Each component of a protocol should serve to either limit the number of opponents capable of posing a threat or limit the exposure to these opponents of the protected data. The protocol designer must attempt to demonstrate how the protocol accomplishes this and to what degree it can be expected to provide the desired security. It is important to notice that the key word used here is **limit**, not **eliminate**. This reflects the effect of what is probably an axiom; namely, that **we cannot eliminate the problem; we can only bound the problem**. Absolute security, provable security may not be required, but careful analysis of the how the critical components bound the problem is essential in order to assure that, when the protocol is actually fielded, the declared level of security is not degraded.

To illustrate this kind of analysis, consider the example of a communications network using a public key cryptosystem. We assume a Central Keying Authority (CKA) which is responsible for issuing to each subscriber in the network, a cryptosystem designed to provide a secrecy channel for communications with that subscriber. The CKA also publishes the public parameters of each subscriber's cryptosystem. We will use the notation $E_i(M)$ to denote the encryption of a message M using the published parameters of the ith subscriber. The decryption of a ciphertext C using the secret parameters will be denoted by $D_i(C)$. In order to send a message M to subscriber i, a transmitter looks up

the public encryption parameters and sends $E_i(M)$ across open communication lines.

The specification of the public key algorithm to be used is of course critical her but much can be said before proceeding further. The critical components to be analyzed this level of specification are:

1. The CKA must choose good parameters to insure that the best possible security achievable with the chosen cryptosystem. Without this requirement, the cryptosystem cannot serve to limit the threat. The CKA must also be trusted not to divulge each subscriber's secret parameters since failure to accomplish this may result in exposure t the threat of the data to be protected.

2. The CKA must publish and maintain the integrity of the public parameters. This is necessary not only to facilitate the communications network, but also to limit the exposure to the threat of the data, since the insertion of fraudulent information in the directory would enable an unauthorized user to pose as an authorized subscriber.

3. The subscribers must take appropriate measures to maintain the secrecy of their decryption functions in order to limit the threat by requiring some effort to decrypt th messages.

Since the purpose of the described protocol is to allow for messages which are not understandable by an outsider to be sent to a given subscriber, the remaining analysis required involves the description of the extent to which this is possible. For example, if $E_i(M)$ is sent to subscriber i, the meaning of M might be exposed if:

1. E_i is not a strong cryptofunction, so that a decryption of $E_i(M)$ is possible by a simple cryptanalysis of the algorithm. For example, if RSA is used, the threat is potentially limited to those capable of factoring the modulus. If the parameters are properly chosen, this threat can be made quite small. However, the protocol designer and/or analyser must be careful to avoid the pitfall of claiming that **finding the meanin of M means breaking the cryptosystem E_i**, as the remaining points will address.

2. $E_i(M)$ may be jibberish to the outsider, but the meaning of M may still be discernable. This, of course, is possible if the number of messages that can be meaningful to the subscribers is small so that precalculation of $E_i(N)$ for all possibl N could make the meaning clear by a simple table lookup [11]. Therefore, the message space must be large enough to limit this threat to the desired level.

3. $E_i(M)$ might be used together with a collection $\{E_j(M)\}$ of encryptions of the same message to other subscribers. In the case of an RSA system, this is possible when a common modulus for each subscriber or if a small common encryption exponent is used [3],[6],[10].

4. $E_i(M)$ might be used together with $\{E_j(M_j)\}$, where M_j is some known variant of M, to reconstruct M. For example, if M_j is M together with a time stamp and the encryption function is RSA with a small exponent [6].

Each of these represents a known problem for some or all public key cryptosystems which are also logically obvious areas of concern. The protocol designer should therefore take care to consider each of these in the specification of the cryptoalgorithm to be used and the parameters to be chosen.

It is important for the protocol designer to be precise in the statement of the security provided by each component of the protocol. In practical application, broad assumptions need to be replaced by specific measures of the level of degradion of security if the the given condition is not met. For a simple illustrative example, the statement that the CKA must be trusted to maintain the secrecy of the subscribers' decryption functions, should really be replaced by the statement that no messages for a given subscriber can be secret if the CKA has divulged the decryption parameters of the subscriber. While this is of course obvious, if the protocol designer adopts this style of describing the protocol, the security issues become clearly demarked.

When protocols are designed, some assumptions about the setting or the cryptoalgorithm used are inevitably made. There is no problem for a practical protocol application with such assumptions, as long as these are clearly defined. In fact, the failure to identify some assumptions is quite often the root of protocol failures, so that a clear definition of all assumptions is necessary for the development of strong protocols. However, one further step would be very useful to both the designer and the applier of a protocol. This step would entail the analysis and possible quantification of the effect on the protocol if some identified assumption is not valid. The benefits of this approach are twofold. First, when the protocol is applied, the advertised level of security will not be degraded by the failure to meet some unspecified requirements for the system. Secondly, the protocol designer will have a more clearly defined protocol and may find that unnecessary assumptions could be discarded, making the protocol more widely applicable.

By way of summary, what we actually propose is a change in perspective in two areas for the design and analysis of protocols. First, for each element of the protocol, the following questions should be answered:

1. How does this element limit the threat or the exposure to the threat?
2. To what extent does this element actually meet that goal?

Secondly, when assumptions are made, they must be clearly defined and the level of security lost when the setting for an application of the protocol does not satisfy the assumption should also be clearly defined.

References

1. R. Berger, S. Kannan and R. Peralta, 'A Framework for the Study of Cryptographic Protocols', *Advances in Cryptology - Proceedings of Crypto 85*, pp.87-103.

2. G.I. Davida, 'Chosen Signature Cryptanalysis of the RSA (MIT) Public Key Cryptosystem', Tech. Rep. TR-82-2, Dept. of Electrical Engineering and Computer Science, Univ. Of Wisconsin, Milwaukee, WI, Oct. 1982.

3. J.M. DeLaurentis, 'A Further Weakness in the Common Modulus Protocol for the RSA Cryptoalgorithm' *Cryptologia*, Vol. 8, No. 3, July 1984, pp. 253-259.

4. S. Even, O. Goldreich and A. Shamir, 'On the Security of Ping-Pong Protocols Using the RSA', *Advances in Cryptology - Proceedings of Crypto 85*, pp.58-72.

5. S. Goldwasser, S. Micali and A. Yao, 'Strong Signature Schemes', *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, 1983, pp.431-439.

6. J. Hastad, 'On Using RSA with Low Exponent in a Public Key Network', *Advances in Cryptology - Proceedings of Crypto 85*, pp.403-408.

7. R.R. Jueneman, S.M. Matyas and C.H. Meyer, 'Message Authentication with Manipulatior Detection Codes', *Proceedings of the 1983 Symposium on Security and Privacy*, pp.33-54.

8. M. Merritt, *Cryptographic Protocols*, Ph. D. Thesis, Georgia Institute of Technology, GIT-ICS-83/06,1983.

9. J.K. Millen, S.C. Clark and S.B. Freedman, 'The Interrogator: Protocol Security Analysis', *IEEE Transactions on Software Engineering*, Vol. SE-13, No. 2, February 1987, pp.274-288.

10. G.J. Simmons, 'A "Weak" Privacy Protocol Using the RSA Cryptoalgorithm', *Cryptologia*, Vol. 7, pp. 180-182.

11. G.J. Simmons and D.B. Holdridge, 'Forward Search as a Cryptanalytic Tool Against a Public Key Privacy Channel', *Proceedings of the IEEE Computer Society 1982 Symposium on Security and Privacy*, 1982, pp.117-128.