Cryptanalysis and Protocol Failures

Gustavus J. Simmons
P. O. Box 365, Sandia Park, NM 87047

Abstract

Most information integrity protocols depend crucially on one or more cryptographic or crypto-like operations to deny unauthorized access to, or use of, information whose integrity the protocol is intended to insure. Obviously, if the underlying cryptoalgorithm were to be broken, then the intended function of the protocol could be subverted. What is not obvious, however, and indeed often comes as a shock to a protocol designer or user, is that a protocol can be completely subverted without impeaching, or even eroding, the security of the underlying cryptoalgorithm. A failure of this dramatic type is said to be a *protocol failure*. There are, of course, many protocols in which the security of the cryptographic portion is progressively weakened as a result of the protocol being exercised -- say by reducing the size of the key space that would have to be searched to identify an active cryptographic key -- but these are not considered to be examples of true protocol failures, in spite of the fact that they are clearly examples of potential sources of failure in the intended security function(s) of a protocol.

In this lecture examples will be given of key distribution protocols that distribute keys to unintended recipients, secrecy protocols that publicly reveal the contents of (supposedly) secret communications, digital signature protocols that make forgery easy -- all based on cryptoalgorithms that are sound so far as is known. In at least one case the cryptographic algorithm that is employed is Vernam encryption/decryption with a properly chosen one time key which is well known to be unconditionally secure; in spite of which the protocol fails totally.

From the standpoint of applications there is scarcely any topic of greater importance than the cryptanalysis of protocols, since protocols are -- in the vernacular of advertising -- "where the rubber meets the road", i.e. where the principles of cryptography get applied to the practice of insuring the integrity of information. The design and/or analysis of cryptographic algorithms is the domain of the mathematician and the cryptographer and can be carried out in large part without regard to applications. The design and analysis of protocols, however, is inextricably linked to the system in which the protocol is to be used, and originates with an application: the function of the protocol being to realize the integrity properties required by the application. Cryptographic algorithms are simply component elements in the design of protocols -- and as we've indicated, the security of the one does not necessarily imply the security of the other. When expressed in this way, protocol failures do not seem so improbable or surprising as they do when described as defined above. In real life though, almost every example of a true protocol failure is also an example of what can aptly be characterized as "Well I'll be damned" discoveries, since this describes the reaction of most people when they first have such a failure pointed out to them.

The cryptanalysis of protocols is essentially formal paranoia, since it depends on suspecting everything, especially those things that are accepted, but not stated. For example, many protocols call for one of the participants to choose a random number at some point in the execution of the protocol. Explicit in this is the belief that the number will be chosen from some known or specified set or range according to a known probability distribution. This may be a verifiable or even an enforceable hypothesis, depended on the protocol itself. Implicit though is the belief that the probability that the random number chosen by the participant will be known to some other participant is simply the probability that if he were to randomly and independently choose a number from the same set or range using the same probability distribution, that he would get the same value. However, there is no way to prevent anyone from sharing anything they know with anyone they trust. Hence this latter assumption about what "random" means is neither verifiable nor enforce-

1st Conf.- Computer & Comm. Security '93-11/93 -VA,USA
© 1993 ACM 0-89791-629-8/93/0011...$1.50

able. Consequently, in any protocol that calls for the generation of a random number, it is essential to the cryptanalysis of the protocol to determine whether there are deceptions that could either be carried out or furthered if the random value were to be shared with one or more of the other participants -- but in secret from some them. One of the neatest of the protocol failures that will be described depends on precisely this for its success.

Similiarly, if a protocol calls for one of the participants -- who may be a "trusted" key generation bureau for example -- to start by constructing a composite number as the product of two primes, chosen so as to make the factorization of their product be computationally infeasible, the suspicion must be that the product is not of this form. It is easy to verify in probability that a number is not a prime, and computationally feasible for numbers of a few hundred decimal digits in size to do so deterministically. It is generally believed by computational number theorists, however, that it just as difficult to test whether a composite number is the product of more than two factors as it is to factor it. Consequently, if a protocol calls for such a composite number to be generated by one of the participants, it is essential in the cryptanalysis to examine whether there are any exploitable consequences of it being the product of more than two prime numbers. For example, it is easy to conceal a covert channel in a signature protocol that calls for the use of a modulus which is the product of two primes, if the modulus is the product of three primes instead.

There is a long list -- too long for a single paper and much too long for an abstract -- of examples of protocol failures that derive from a quantity not being what it is supposed to be, or what it is advertised to be. The two examples above should give the reader a feeling for what is involved in protocol analysis.

The cryptanalysis of protocols consists of three steps:

1. Carefully enumerate all of the properties of all of the quantities involved; both those explicitly stated in the protocol and those implicitly assumed in the setting.

2. Take nothing for granted. In other words go through the list of properties assuming that none of them are as they are claimed or tacitly assumed to be unless a proof technique exists to verify their nature. For each such violation of property, critically examine the protocol to see if this makes any difference in the outcome of the execution of the protocol. Combinations of parameters as well as single parameters must be considered.

3 Finally, if the outcome can be influenced as a result of a violation of one or more of the assumed properties , it is essential to then determine whether this can be exploited to advance some meaningful deception. There are several well known protocols in which it is possible to influence the outcome by violating the assumed properties of one or more of the parameters involved, but in which no known meaningful deception can be worked or furthered as a result. Protocol failures occur whenever the function of the protocol can be subverted as a consequence of the violations.

This lecture will illustrate the application of these rules for the cryptanalysis of protocols with several examples of pure protocol failures discovered using them.