On the Perfect Encryption Assumption

O. Pereira[*] and J.-J. Quisquater
UCL Crypto Group
{pereira, quisquater}@dice.ucl.ac.be

Nearly all models proposed within the scope of the study of security protocols make perfect encryption assumptions. These hypotheses can be summarised as follow:

- The decryption key must be known in order to extract the plaintext corresponding to a given ciphertext.
- There is enough redundancy in the cryptosystem that a ciphertext can only be generated using encryption with the appropriate key and message.

This assumption is obviously not true in practice. A first example is the one of the cryptosystems proceeding by cipher-block-chaining (CBC). In such systems, the encryption of message block sequence $P_1P_2\ldots P_n$ is $C_0C_1C_2\ldots C_n$ where C_0=I (Initialisation bloc) and $C_i=\{C_{i-1}\oplus P_i\}_K$.

It can be noticed that they present the following interesting particularity:

If $\quad C_0C_1C_2\ldots C_iC_{i+1}\ldots C_n = \{P_1P_2\ldots P_iP_{i+1}\ldots P_n\}_K$
Then $\quad C_0C_1C_2\ldots C_i = \{P_1P_2\ldots P_i\}_K$

This property can be exploited (see [Boy90] or [SG92] for other instances) to flaw the Needham-Schroeder symmetric key authentication protocol [NS78]. This protocol intends to permit Alice to establish a shared secret key K_{ab} with Bob and to obtain mutual conviction of the possession of the key by each other. The key is provided by a trusted server S who shares the secret keys K_{as} and K_{bs} with A and B respectively. This protocol can be described as follow:

$$\begin{aligned}
A \rightarrow S &: A.B.N_a \\
S \rightarrow A &: \{N_a.B.K_{ab}.\{K_{ab}.A\}K_{bs}\}_{Kas} \\
A \rightarrow B &: \{K_{ab}.A\}K_{bs} \\
B \rightarrow A &: \{N_b\}_{Kab} \\
A \rightarrow B &: \{N_b\text{-}1\}_{Kab}
\end{aligned}$$

Beyond other existing attacks (the most famous having been proposed in [DS81]); this protocol can be flawed as follow. Suppose that the message $\{N_a.B.K_{ab}.\{K_{ab}.A\}K_{bs}\}_{Kas}$ has ciphertext $C_0C_1C_2\ldots C_n$ and that all components have length one block. Then $\{N_a.B\}_{Kas}$ has ciphertext $C_0C_1C_2$. But this message is of the form A might expect to receive as third message of a later session of the protocol where B is considered to play initiator's role. Thus, A can be fooled into accepting the publicly known N_a as a secret key shared with B.

The cryptosystems using block ciphers are furthermore particularly sensitive to known-pairs or chosen-pairs attacks. This imposes to design protocols in order to avoid disclosure of such pairs, but it remains a tough problem, even for a team of experts. Recent works have been performed by S. Stubblebine and C. Meadows [SM00] in order to avoid such attacks by automated analysis.

A second example is the one of the RSA cryptosystem. Indeed, this system presents a multiplicative structure ($(m_1m_2)^e \equiv (m_1)^e(m_2)^e \equiv c_1c_2 \pmod n$), what is in conflict with the second part of the perfect encryption assumptions. This feature can be exploited in order to break the Needham

[*] This author would like to thank the Belgian National Fund for Scientific Research (F.N.R.S) for its financial support.

Schroeder public key authentication protocol in its version fixed by Lowe [Low95] (who proved that it was correct in a context of perfect encryption). This protocol can be described as follow:

$$\begin{aligned} A &\rightarrow B : \{N_a.A\}_{Kb} \\ B &\rightarrow A : \{N_a.N_b.B\}_{Ka} \\ A &\rightarrow B : \{N_b\}_{Kb} \end{aligned}$$

Alice sends a first message to Bob in which she asks him to decrypt a nonce N_a and to send it back. Bob answers in the second message and encloses with N_a the nonce N_b and his identifier. Alice replies then with the third message in which she proves that she has been able to decrypt the nonce N_b.

We suppose that the encryption is made with RSA using a 512 bits modulus, that the identifiers are 32 bits numbers, and that the nonces are 64 bits numbers. Given that we use the RSA for encryption (rather than for signature), no specific padding will normally be used and we assume that the messages will begin with a suitable number of zeros. We will also assume that a principal who receives a message will not check if the zeros are actually present and will only read the bits that he needs to end the session of the protocol. In this context, a principal C whose identifier is such that $C \equiv 1 \bmod 8$ can attack the four principals having the identifiers A such that $A^2 \bmod 2^{32} = C$ as follow:

α_1 $A \rightarrow C(B) : \{N_a.A\}_{Kb}$

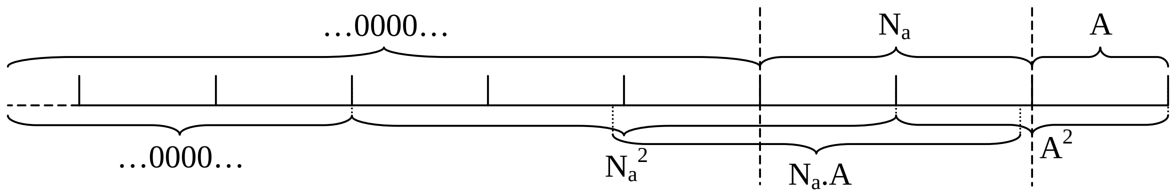
β_1 $C \rightarrow B : \{N_c.C\}_{Kb} = (\{N_a.A\}_{Kb})^2 \bmod n$ (where n is the public modulus of B)

β_2 $B \rightarrow C : \{N_c.N_b.B\}_{Kc}$

α_2 $C(B) \rightarrow A : \{N_a.N_d.B\}_{Ka}$

α_3 $A \rightarrow C(B) : \{N_d\}_{Kb}$

In this scenario, C intercepts the message that A sends to B, raises it to its square, and resends it to B (messages α_1 and β_1). B answers to C, what allows him to compute the value of N_a. This can be done by noticing that[1] $\{(\{N_a.A\}_{Kb})^2 \bmod n\}_{Kb}^{-1} = (N_a.A)^2 \bmod n = (2^{32}{*}N_a+A)^2 \bmod n = (2^{64}{*}Na^2+2^{33}{*}N_a{*}A+A^2) \bmod n$. This last expression can be represented as follow:



It can be easily checked that the identifier read by B is actually $C = A^2 \bmod 2^{32}$, so B will send the message β_2 to the intruder. Moreover, the value of N_a can easily be computed from the one of N_c, which is the sum of the 32 most significant bits of A^2, of the 64 least significant bits of twice $N_a{*}A$, and of 2^{32} times the 32 least significant bits of N_a^2, the hole token modulo 2^{64}. The choice between the different solutions of this equation can easily be done by recomputing the value of $\{N_a.A\}_{Kb}$ for the various possible values of N_a. Having obtained this value, C can easily finish the session α of the protocol.

We can see that a designer who, for the sake of a greater security, would increase the size of the used RSA modulus would in fact make such attacks easier rather than the opposite. A simple way to avoid them is to add redundancies in the messages before encrypting them. The experience of such countermeasures in the context of the protection against signature forgery shows that it is unfortunately very difficult to achieve if we want to avoid every multiplicative attacks. We are thinking for instance at the attack described by Grieu [Gri00] who shows how to produce the signature of a valid message given those of three others, despite the restricting rules imposed by the ISO/IEC 9796-1 standard [ISO98].

The setting of efficient protections would be made much easier if we defined formal models taking into account the imperfections of the encryption schemes. Their checking would allow us to become aware of the way in which those features can be exploited, what would be helpful in order to

[1] We use the symbol « . » to denote the concatenation, and the symbol « ∗ » to denote the product.

avoid such attacks. We can indeed observe that the attack presented above can be avoided by simply inverting the order of the nonce and the identifier in the first message since this prevents the intruder to foresee the value of the identifier after squaring. A variant of this attack consisting in multiplying the first message by a small factor encrypted would nevertheless work in a great amount of circumstances.

In this spirit, we developed a model inspired by those used in [MCJ97] or [Low98], but taking into account some consequences of the multiplicative structure of RSA. To that purpose, we introduced a new data type *f* representing small factors. Those factors can be used to multiply messages, considering the property that $f^*(m_1.m_2)=(f^*m_1).(f^*m_2)$ which is often effective in practice (the practical upper-bound for the size of these factors is in fact dependent of the studied protocol and of the proposed flaw). Besides, we supposed that the intruder was in possession of private keys corresponding to identifiers which are multiple of those of honest principals by these small factors.

The main difficulty in the study of our model is in the modelling of the associative and commutative structure of the multiplication and of the distributivity property of multiplication on concatenation. Indeed, those properties prevent us from using "normalised derivations" of messages (such as in [MCJ97]) or "unique readability" axioms (such as in [FHG98]).

In order to solve this problem, we have limited the ability of the intruder into generating messages by imposing him the use of only one small factor (maybe inverted of encrypted) for multiplication. Furthermore, we allowed him to multiply an encrypted message only by a small factor encrypted with the same key as the message. This last requirement was anyway necessary in order to keep plausible the distributivity assumption.

Having so bounded the number of possible interpretations of the messages (and the size of the state space of our system at the same time), we specified our model in Promela, which is the specification language of the model-checker SPIN [SPH] developed at Bell-Labs since around twenty years. This language presents the particularity of supporting the use of the integer type with additive and multiplicative operators. Given that, we defined our small factor as the integer 2 and assigned ranges of integer values to all types of atomic messages seeing to avoid unexpected conflicts of interpretation (i.e. conflicts other than those related to the particular properties of intruder's identifiers).

Those light restrictions on the capabilities of the intruder combined with the particularities of the Promela language allowed us to test our model on the Needham-Schroeder-Lowe protocol. Within a few seconds of computation, SPIN provided us two attacks similar to the one described above.

Bibliography

[Boy90] C. Boyd. *Hidden Assumptions in Cryptographic Protocols.* In Proceedings of the IEE, 137 Pt E(6) : 433-436, Nov 1990.

[DS81] D. Denning and G. Sacco. *Timestamps in Key Distribution Protocols*. Communications of the ACM, 24(8), Aug 1981

[Gri00] F. Grieu. *A Chosen Message Attack on ISO/IEC 9796-1 Signature Scheme.* In Proceedings of Eurocrypt'2000, Springer-Verlag's Lecture Notes in Computer Science, Vol 1807, pages 70-80, May 2000.

[FHG98] F.J. Thayer Fábrega, J.C. Herzog, J.D. Guttman. *Honest Ideals on Strand Spaces.* In Proceedings of the 11th IEEE Computer Security Foundations Workshop. IEEE Computer Society Press, Jun 1998.

[ISO98] ISO/IEC 9796-1 Second edition Final Committee Draft. Information technology – Security techniques – Digital Signature Scheme giving message recovery – Part 1 : Mechanisms using redundancy. Circulated as ISO/IEC JTC1/SC27 N2175 (1998).

[Low95] G. Lowe. *An Attack on the Needham-Schroeder Public-Key Authentication Protocol*. Information Processing Letters, volume 56, number 3, pages 131-133. 1995

[Low98] G. Lowe. *Casper: A Compiler for the Analysis of Security Protocols*. In Journal of Computer Security, Volume 6, pages 53-84, 1998.

[MCJ97] W. Marrero, E. Clarke, S. Jha. *A Model Checker for Authentication Protocols*. In Proceedings of the DIMACS workshop on design and formal verification of security protocols. Sept 1997.

[NS78] R.M. Needham, M.D. Schroeder. *Using Encryption for Authentication in Large networks of Computers*. In Communications of the ACM. Vol 21, Num 12, p. 993-999, Dec 1978.

[SG92] S. Stubblebine, V. Gligor. *On Message Integrity in Cryptographic Protocols*. In IEEE Symposium on research in Security and Privacy, pages 85-104, May 1992.

[SM00] S. Stubblebine, C. Meadows. *Formal Characterization and Automated Analysis of Known-Pair and Chosen-Text Attacks.* In IEEE Journal on Selected Areas in Communications, Vol 18, Num 4, Apr 2000.

[SPH] SPIN Homepage. http://netlib.bell-labs.com/netlib/spin/whatispin.html.