The Validation of Cryptographic Algorithms

Jacques Stern
Jacques.Stern@ens.fr

Ecole Normale Supérieure
Laboratoire d'informatique
45, rue d'Ulm
75230 Paris Cedex 05

Abstract. Since the appearance of public-key cryptography in the seminal Diffie-Hellman paper, many new schemes have been proposed and many have been broken. Thus, the simple fact that a cryptographic algorithm has withstood cryptanalytic attacks for several years is, by itself, a kind of validation procedure. A completely different paradigm is provided by the concept of provable security. Stated in a more accurate way, this approach proposes computational reductions to well established problems such as factoring or the discrete logarithm problem. Recently, the scope of this method has been considerably widened by using a model where concrete cryptographic tools are replaced by ideal objects: in this model, DES is viewed as a random permutation and SHA as a random function with the appropriate range. Basically, this is another technique for spotting error designs and validating cryptographic algorithms. When cryptanalysis and security proofs combine with each other so that there is virtually no gap between them, the resulting picture becomes quite convincing. The present paper gives several examples of such a situation taken from various areas of cryptography such as signature schemes, public-key identification or even symmetric-key techniques.

1 Introduction

Since the appearance of public-key cryptography in the seminal Diffie-Hellman paper [4], many new schemes have been proposed and many have been broken. Thus, the simple fact that a cryptographic algorithm has withstood cryptanalytic attacks for several years is, by itself, a kind of validation procedure. In this approach, cryptanalysis is viewed as a heuristic measure of the strength of a new proposal. A completely different paradigm is provided by the concept of provable security. This significant line of research has tried to provide proofs in the asymptotic framework of complexity theory. Stated in a more accurate way, this approach proposes computational reductions to well established problems such as factoring or the discrete logarithm problem. Of course, these are not absolute proofs since cryptography ultimately relies on the existence of one-way functions and the $\mathcal{P}$ vs. $\mathcal{NP}$ question. Recently, the scope of this method has been considerably widened by using a model where concrete cryptographic tools are replaced by ideal objects: in this model, DES is viewed as a random permutation and SHA as a random function with the appropriate range. The method was put in systematic form in [1] using the name "Random Oracle Model" and has been quite successful as another technique for spotting error designs and validating cryptographic algorithms. When cryptanalysis and security proofs combine with each other so that there is virtually no gap between them, the resulting picture becomes quite convincing and, accordingly, conveys a reasonably high degree of practical assurance. The aim of the present paper is to give several examples of such a situation taken from various areas of cryptography such as signature schemes, public-key identification or even symmetric-key techniques. The examples include

1. A precise security analysis of the El Gamal signature scheme and its variants
2. A discussion of the size of the hash functions used in zero knowledge identification protocols
3. An account of the work of of Bellare, Kilian and Rogaway [2] on the security of cipher block chaining and a comparison of the hypotheses they use with cryptanalytic results concerning MACs

The first two items are related with previous work of the author. This is merely a matter of practicality: much more work of a similar vein due to many different authors can be found in the bibliography.

2 El Gamal Signatures

At EUROCRYPT 96, by some sort of unexpected coincidence, two papers devoted to the security of the El Gamal signature scheme appeared, one by Bleichenbacher and the other by Pointcheval and the author (see [3, 13]). The first was in the Cryptanalysis section and reported a potential weakness of the scheme whereas the second, included in the signature section, was able to formally prove the security of a variant of the same scheme. A closer look at both papers was even more puzzling since it was explained that the Bleichenbacher attack was applicable to the variant we discussed in the other paper. It was only through a deeper examination that the apparent contradiction could vanish since the security proof was correct for "almost all" choices of the parameters whereas the attack was tracking very specific values. In this section, we will briefly review the ElGamal scheme and its variant as well as the content of the two EUROCRYPT papers. Then we will investigate their "touching point" and derive practical consequences.

2.1 Brief Review of the Signature Scheme

The original El Gamal signature scheme [5] was proposed in 1985 but its security was never proved equivalent to the discrete logarithm problem nor to the Diffie-Hellman problem.

Description of the Original Scheme Let us begin with a description of the original scheme [5]:

- the key generation algorithm: it chooses a random large prime p of size n and a generator g of $(\mathbb{Z}/p\mathbb{Z})^\star$, both public. Then, for a random secret key $x \in \mathbb{Z}/(p-1)\mathbb{Z}$, it computes the public key $y = g^x \bmod p$.
- the signature algorithm: in order to sign a signature of a message m, one generates a pair (r, s) such that $g^m = y^r r^s \bmod p$. To achieve this aim, one has to choose a random $K \in (\mathbb{Z}/(p-1)\mathbb{Z})^\star$, compute the exponentiation $r = g^K \bmod p$ and solve the linear equation $m = xr + Ks \bmod (p-1)$. The algorithm finally outputs (r, s).
- the verification algorithm checks the equation $g^m = y^r r^s \bmod p$.

As already seen in the original paper, one cannot show that the scheme is fully secure because it is subject to existential forgery. Following a design that appears in the work of Schnorr [16], we proposed to modify the scheme by using a hash function f.

Description of the modified El Gamal scheme In this variant, we replace m by the hash value of the part of the computation bound not to change, namely $f(m, r)$.

- the key generation algorithm: unchanged.
- the signature algorithm: in order to sign a message m, one generates a pair (r, s) such that $g^{f(m,r)} = y^r r^s \bmod p$. In order to achieve this aim, one generates K and r the same way as before and solves the linear equation $f(m, r) = xr + Ks \bmod (p-1)$. The algorithm outputs $(r, f(m, r), s)$.
- the verification algorithm checks the signature equation with the obvious changes due to the hash function.

2.2 The Security Result

Of course, the hash functions that we had in mind were practical proposals such as e.g. MD5 [14] or SHS [11]. Still, in order to prove a security result we used the "random oracle model". On other terms, we treated the hash function as an oracle which produces a random value for each new query. Of course, if the same query is asked twice, identical answers are obtained. Proofs in this model ensure security of the overall design of a signature scheme provided the hash function has no weakness. For the modified scheme, we were able to prove a security result in the so-called adaptively chosen message scenario where the attacker can dynamically ask the legitimate user to sign any message, using him as a kind of oracle before he attempts to issue a fake signature. Our result applied to a large variety of moduli p, those for which $p-1$ has a single large prime factor Q. Those prime moduli are precisely used for cryptographic applications of the discrete logarithm problem. In order to give a more precise mathematical definition,we let $|p|$ denote the length of an integer p.

Definition 1. Let α be a fixed real. An α-hard prime number p is such that the factorization of $p-1$ yields $p-1 = QR$ with Q prime and $R \leq |p|^\alpha$.

Theorem 2. *Consider an adaptively chosen message attack in the random oracle model against schemes using α-hard prime moduli. Probabilities are taken over random tapes, random oracles and public keys. If an existential forgery of this scheme has non-negligible probability of success, then the discrete logarithm problem with α-hard prime moduli can be solved in polynomial time.*

2.3 Spotting the Weakness

We will not give the proof of the theorem just stated in the previous section, for which we refer to [13]. We will only mention that it deals with probabilistic polynomial time Turing machines and that we turn any attack into a machine $\mathcal{M}$ which, on input (g, y), outputs, with non-negligible probability, $x \in \mathbb{Z}/(p-1)\mathbb{Z}$ such that $y = g^x \bmod p$ (case 1) or $b \in \mathbb{Z}/R\mathbb{Z}$ and $t \in \mathbb{Z}/(p-1)\mathbb{Z}$ such that t is prime to Q and $bQ = g^t \bmod p$ (case 2). Probabilities are taken over g, y, and the random tapes of $\mathcal{M}$. Case 1, the "good" case immediately yields the discrete logarithm of y. As for the "bad" case 2, it only discloses the discrete logarithm of some small multiple bQ of Q. We could overcome the resulting difficulty by using randomization over g so as to solve discrete logarithms in general.

What if g is not randomly chosen? Then our argument collapses and we have thus spotted a weakness. More precisely, we have the following

Theorem 3. *From the knowledge of b and t such that t is prime to Q and $bQ = g^t \bmod p$, it is possible to generate signatures without the secret key for a significant proportion of the possible messages.*

Proof. Set $r = g^t \bmod p$. The equation to solve in order to produce the required signature reads $f(m,r) = xr + ts \bmod (p-1)$. Reducing modulo Q, we get $f(m,r) = ts \bmod Q$, which we can solve for s. As for the R-part, which reads $f(m,r) = xr + ts \bmod R$, it can be found by exhaustive search, regardless of the information on x but provided that the solution exists: if t is prime to R, this is always the case. Otherwise, the (unknown) quantity $f(m,r) - xr$ has to be a multiple of the gcd of t and R, which happens with significant probability.

2.4 The Bleichenbacher Attack

First note that theorem 3 does not actually use the full strength of the hypotheses that were needed for the security result: Q need not be prime and it is enough that $R = \frac{p-1}{Q}$ is smooth in order to make the required exhaustive search possible. Bleichenbacher's attack stems from the following:

Theorem 4. *Whenever g is smooth, divides $p-1$ and is not a quadratic residue modulo p, it is possible to generate signatures without the secret key for a significant proportion of the possible messages.*

Note that the above applies to the El Gamal scheme as well as to the variant discussed above.

Proof. Set $Q = \frac{p-1}{g}$ and $t = \frac{p-3}{2}$. Since g is not a quadratic residue, we have $g^{\frac{p-1}{2}} = -1 = p - 1 \bmod p$, hence $g^t = \frac{p-1}{g} = Q \bmod p$. The hypotheses of theorem 3 are met and thus a significant proportion of the messages can be signed. There is a minor problem due to the fact that t is not necessarily prime to Q. Actually, since $p-3$ is a multiple of t, the gcd of t and $p-1$ is at worse 2. A closer examination of the proof of theorem 3, with this observation in mind, shows that the conclusion remains.

2.5 The Final Picture

The apparent contradiction between Bleichenbacher's attack and our security result has thus vanished. Moreover, the overall picture is now very clear: the modified El Gamal signature scheme is secure provided the generator g of $(\mathbb{Z}/p\mathbb{Z})^\star$ is chosen at random. If it is not, then, as reported in [3], there is some danger that a trapdoor has been added. Thus, a reasonable requirement would be that the authority issues some sort of proof that g has been fairly manufactured, as was suggested for the modulus p of the digital signature standard (see [11]).

3 Hash Functions in Identification Protocols

At CRYPTO 93, the author introduced a zero-knowledge identification scheme based on the syndrome decoding problem from the theory of error-correcting codes ([18]). This work followed a line of research trying to find appropriate alternative techniques to number theory. Previous research along the same line had been earlier performed by Shamir who had designed another scheme based on the Permuted Kernel problem (see [17]). Both schemes used hash function at the so-called commitment stage. In the security analysis that we gave in our CRYPTO paper (see also [20]), we noticed that any attack could be turned into a machine which could either output some substitute to the secret key or else find collisions for the hash function, with overwhelming probability. Still, we felt that it might as well be the case that one-wayness was enough. Thus, results of further investigations that we undertook with Marc Girault (see [10]) came as a surprise: collision-freeness is really needed. Again, the correct picture came from the joint effort of security proofs and cryptanalysis.

3.1 Brief Description of the Scheme

The scheme is base on a fixed randomly generated binary matrix H of large size, $m \times n$, say 256×512. Each user U receives a secret key s_U, chosen at random by the authority among all n-bit words with a prescribed number p of 1's, say $p = 56$. This prescribed number p is also part of the system. The public identification of the user is computed as

$$i_U = H(s_U)$$

This allows a registered participant to perform the basic interactive protocol that enables any user U (which we call the prover) to identify himself to another entity (which we call the verifier). The protocol includes r rounds, each of these being performed as follows:

1. The prover picks a random n-bit word y together with a random permutation σ of the integers $\{1 \cdots n\}$ and sends commitments c_1, c_2, c_3 respectively as

$$c_1 = \langle \sigma || H(y) \rangle$$
$$c_2 = \langle y.\sigma \rangle$$
$$c_3 = \langle (y \oplus s_U).\sigma \rangle$$

to the verifier. In the above $\langle\rangle$ simply denotes the hash function.
2. The verifier sends a random element b of $\{0, 1, 2\}$.
3. If b is 0, then, the prover returns y and σ. If b is 1 then, the prover reveals $y \oplus s$ and σ. Finally, if b equals 2, then the prover discloses both $y.\sigma$ and $s_U.\sigma$.
4. If b equals 0, the verifier checks that commitments c_1 and c_2, which were made in step 1, have been computed honestly. More accurately, let $\tilde{y}$ and $\tilde{\sigma}$ be the answers received from the prover at step 3, then the equations to check are as follows:

$$c_1 = \langle \tilde{\sigma} || H(\tilde{y}) \rangle$$
$$c_2 = \langle \tilde{y}.\tilde{\sigma} \rangle$$

If b equals 1, the verifier checks that

$$c_1 = \langle \tilde{\sigma} || H(\tilde{y}) \oplus i_U \rangle$$
$$c_3 = \langle \tilde{y}.\tilde{\sigma} \rangle$$

Finally, if b is 2, the verifier checks the weight property and commitments c_2 and c_3, i.e. with obvious notations,

$$c_2 = \langle \tilde{y} \rangle$$
$$c_3 = \langle \tilde{y} \oplus \tilde{s} \rangle$$

The security result whose proof we omit, reads as follows:

Theorem 5. *Assume that some probabilistic polynomial-time adversary $\tilde{P}$ is accepted with probability $\geq (2/3)^r + \epsilon$, $\epsilon > 0$, after playing a constant number r of rounds of the identification protocol. Then there exists a polynomial-time probabilistic machine which outputs an acceptable key s from the public data or else finds collisions for the hash function, with overwhelming probability.*

Here an acceptable key is any word s with the prescribed weight such that $H(s) = i_U$.

3.2 Attacks Based on Collisions

In order to give an abstract treatment of the work appearing in [10], we introduce the following definition:

Definition 6. A *sample* for a hash function is a subset of its possible inputs. Given two samples S_1, S_2 for a hash function, a collision between these samples consists of $x_1 \in S_1$ and $x_2 \in S_2$ such that $\langle x_1 \rangle = \langle x_2 \rangle$.

We always assume implicitly that samples and hash values are produced by polynomial-time machines and that samples have exponential size whereas hash values have small length. Hence collisions do exist. The main result in [10] reads as follows.

Theorem 7. *Any adversary that can produce collisions between samples can be accepted without knowledge of the secret key.*

Proof. As shown in [10], the attacker can choose to attack any of the three commitments. We focus on c_2. The impostor selects a permutation σ and a word y'. He next considers two samples

1. The sample consisting of inputs to the hash function of the form $y_1.\sigma$ such that y_1 is a solution of the equation $H(y_1) = H(y') \oplus i_U$. Note that there are exponentially many such solutions.
2. The sample consisting of inputs to the hash function of the form $y_2.\sigma$ such that $y_2 \oplus y'$ has weight p.

Let y_1, y_2 be a collision between the samples. At each execution of the basic protocol, the impostor sends

$$c_1 = \langle \sigma || H(y_1) \rangle$$
$$c_2 = \langle y_1.\sigma \rangle = \langle y_2.\sigma \rangle$$
$$c_3 = \langle y'.\sigma \rangle$$

If the verifier asks $b = 0$, the cheater replies with y_1 and σ; if $b = 1$, he returns y', σ; finally, if $b = 2$, he answers $y_2.\sigma$ and $(y_2 \oplus y').\sigma$. In all three cases, the verifier is satisfied with the answer.

3.3 Practical Consequences

If we identify hash functions with random functions, then by the birthday paradox, the running time of finding collisions for samples is $O(\sqrt{2^k})$, where k is the size of the hash values. As a consequence, the practical meaning of the previous results is that 64 bit hash values should be avoided. Our identification scheme really needs long hash values and 128 bits is a minimum.

4 Cipher Block Chaining

At CRYPTO 94, Bellare, Kilian and Rogaway gave a security proof for the classical design known as cipher block chaining (see [2]). More accurately, they considered authentication of a message $x = x_1, \cdots, x_m$ by tagging x with a prefix of

$$f^{(m)}(x) = f(f(\cdots(f(f(x_1) \oplus x_2)) \oplus \cdots \oplus x_{m-1}) \oplus x_m)$$

where f is a block cipher (e.g. DES). Their setting was quite similar to the one discussed above in section 2, in that the attacker was allowed to request the MAC values of adaptively chosen messages. They were able to prove that any attack which distinguishes $f^{(m)}$ from random functions with significant probability, can be turned into a test distinguishing f itself from random functions. They had a more precise quantitative version that appears below and involves the number of queries q made by the attacker. If one compares this version with the collision attacks stemming from the birthday paradox, we see that there is a small gap. The aim of the present section is to understand this gap.

4.1 Brief Review of the Security Result

Consider an attack that distinguishes the CBC function $f^{(m)}$ built from a function f whose inputs are ℓ bit long. Let q be the number of queries asked, t be the time taken and ϵ be the success probability. The result of Bellare, Kilian and Rogaway reads as follows:

Theorem 8. *Assume $qm \leq 2^{(\ell+1)/2}$, then there is another algorithm that distinguishes f itself from a random function, whose success probability is $\epsilon' = \epsilon - 3q^2m^2 2^{-\ell-1}$. This algorithm asks $q' = qm$ queries and takes time $t + O(qml)$.*

We refer to [2] for the proof.

4.2 The Gap with Cryptanalysis

It is know that MAC collisions can be found through the birthday paradox by querying $\sqrt{2}2^{\ell/2}$ MAC values, where ℓ is the number of bits of the inputs to f. It is not surprising therefore that the authors of [2] have a condition that relates q, m and $2^{\ell/2}$. It turns out that this condition cannot be simply $q \leq O(2^{\ell/2})$. This follows from a result by Preneel and van Oorschot [12] who observe that if the messages have s trailing blocks in common, the number of MACs needed for finding a collision w.r.t. $f^{(m)}$ goes down to approximately $\sqrt{2/(s+1)}2^{\ell/2}$. Thus, collision can be found with $O(\frac{2^{\ell/2}}{\sqrt{m}})$ queries by setting $s = m - 1$ and this distinguishes $f^{(m)}$ from random functions. However, the condition from [2] reads $qm \leq 2^{(\ell+1)/2}$, whereas cryptanalysis hints towards a weaker condition of the form $q \leq O(\frac{2^{\ell/2}}{\sqrt{m}})$. It is unclear whether the gap can be narrowed.

5 Conclusion

The content of the present paper is methodological in character. We have shown several examples where security proofs and cryptanalysis almost match up. If they do, the resulting picture is very convincing in terms of practical security. If the match is not tight, it is often an indication that further research is needed.

References

1. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In Proceedings of the 1st ACM Conference on Computer and Communications Security (1993) pp. 62–73.
2. Bellare, M., Kilian, J., Rogaway, P.: The security of cipher block chaining. In Advances in Cryptology – Proceedings of CRYPTO '94 (1994) vol. Lecture Notes in Computer Science 839 Springer-Verlag pp. 341–358.
3. Bleichenbacher, D.: Generating ElGamal signatures wothout knowing the secret key. In Advances in Cryptology – Proceedings of EUROCRYPT '96 (1996) vol. Lecture Notes in Computer Science 1070 Springer-Verlag pp. 10–18.
4. Diffie, W., Hellman, M.: New directions in cryptography. In IEEE Transactions on Information Theory (november 1976) vol. IT–22, no. 6 pp. 644–654.
5. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In IEEE Transactions on Information Theory (july 1985) vol. IT–31, no. 4 pp. 469–472.
6. Fiat, A., Shamir, A.: How to prove yourself: practical solutions of identification and signature problems. In Advances in Cryptology – Proceedings of CRYPTO '86 (1986) vol. Lecture Notes in Computer Science 263 Springer-Verlag pp. 186–194.
7. Goldwasser, S., Micali, S., Rackoff, C.: Knowledge complexity of interactive proof systems. In Proceedings of the 17th ACM Symposium on the Theory of Computing STOC (1985) pp. 291–304.
8. Goldwasser, S., Micali, S., Rivest, R.: A digital signature scheme secure against adaptative chosen-message attacks. SIAM journal of computing **17** (1988) pp. 281–308.
9. Guillou, L., Quisquater, J.: A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In Advances in Cryptology – Proceedings of EUROCRYPT '88 (1989) vol. Lecture Notes in Computer Science 330 Springer-Verlag pp. 123–128.
10. Girault, M., Stern, J.: On the length of the cryptographic hash values used in identification schemes. In Advances in Cryptology – proceedings of CRYPTO '94 (1994) vol. Lecture Notes in Computer Science 839 Springer-Verlag pp. 202–215.
11. NIST: Secure Hash Standard (SHS). Federal Information Processing Standards PUBlication 180–1 April 1995.
12. Preneel, B., van Oorschot P.C., MDx-MAC and building fast MAC's from hash functions. In Advances in Cryptology – proceedings of CRYPTO '95 (1995) vol. Lecture Notes in Computer Science 963, Springer-Verlag pp. 1–14.
13. Pointcheval, D., Stern, J.: Security proofs for signature schemes. In Advances in Cryptology – Proceedings of EUROCRYPT '96 (1996) vol. Lecture Notes in Computer Science 1070 Springer-Verlag pp. 387–398.
14. Rivest, R.: The MD5 message-digest algorithm. RFC 1321 april 1992.

15. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM **21** (1978) pp. 120–126.
16. Schnorr, C.: Efficient identification and signatures for smart cards. In Advances in Cryptology – Proceedings of CRYPTO '89 (1990) vol. Lecture Notes in Computer Science 435 Springer-Verlag pp. 235–251.
17. Shamir, A.: An efficient identification scheme based on permuted kernels. In Advances in Cryptology – Proceedings of CRYPTO '89 (1990) vol. Lecture Notes in Computer Science 435 Springer-Verlag pp. 606–609.
18. Stern, J.: A new identification scheme based on syndrome decoding. In Advances in Cryptology – proceedings of CRYPTO '93 (1994) vol. Lecture Notes in Computer Science 773 Springer-Verlag pp. 13–21.
19. Stern, J.: Designing identification schemes with keys of short size. In Advances in Cryptology – proceedings of CRYPTO '94 (1994) vol. Lecture Notes in Computer Science 839 Springer-Verlag pp. 164–173.
20. Stern, J.: A new paradigm for public key identification. In IEEE Transactions on Information Theory (1996), to appear.

This article was processed using the LaTeX macro package with LLNCS style