Short Signatures with Message Recovery in the Random Oracle Model

Akihiro Mihara Keisuke Tanaka *

Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology,
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan.
{mihara0, keisuke}@is.titech.ac.jp

January 23, 2004

Abstract

Granboulan [4] proposed the signature scheme in the ideal cipher model named OPSSR that achieves the lower bound for message expansion. In this paper, we propose a scheme which can give the security equivalent to that of OPSSR in the random permutation model that is weaker than the ideal cipher model. We also show exact security proof. We extend our scheme to the multi key setting. By the results of this paper, we partially solve the open problems posed by Granboulan.

keywords

Signature scheme, Message recovery, Ideal chiper model, Random permutation model

1 Introduction

1.1 Signature

A digital signature scheme allows a signer to transform an arbitrary message into a signed message, such that anyone can check the validity of the signed message using the signer's public key, but only the signer is able to generate signed messages. A signed message contains the information about the message, plus some information to prove its validity. For example in the case of a scheme without message recovery, the signed message is the concatenation of the message and of a signature.

The message expansion of a signature scheme is the difference between the length of the signed message and the original message. It is the length of the signature, if there is no message recovery.

The title of the paper [4] by Granboulan is "Short signatures in the random oracle model" [4]. Although the title includes "the random oracle model", the security was proved in the ideal cipher model, which is considered to be stronger than the random permutation model. We improve this situation and prove the same security in the random permutation model.

1.2 Related works

Many schemes proposed with short signatures [6, 7, 8], but their exact security is not proven to be equivalent to the underling problem with the same parameters, because their security proofs are not tight.

Partial message recovery can allow one to reduce message expansion when having security parameters corresponding to the tightness of the security proof. Message recovery has been used to reduce the

*Supported in part by NTT Information Sharing Platform Laboratories and Grant-in-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science, and Technology, 14780190.

message expansion in the PSS scheme [1], the Pintsov-Vanstone scheme [2] or the DSA-like schemes [7]. But those schemes do not achieve minimal signature length.

Coron [3] showed how to reduce the length of the random salt in PSS, to improve the amount of message recovered, and reduce the message expansion. But the result of this improvement is still not optimal. Coron, Joye, Naccache, and Paillier [5] showed that the PSS padding, which was designed for signature, can also be used for encryption.

In [4], Granboulan introduced the definition of message expansion which generalizes the notion of short signature for schemes with message recovery. He showed what is the minimal possible message expansion for a given proven security requirement. He also described a padding that achieves this lower bound and that can be used with RSA.

2 Definitions

2.1 Digital signature schemes

A signature scheme is described by the following four algorithms:

- a parameter generation
  $Generate$: $\rho \mapsto param$
- a key generation
  $KeyGen_{param}$: $\rho' \mapsto (pk, sk)$
- a signature generation
  $Sign_{param,pk,sk}$: $(m, r) \mapsto \sigma$
- a signature verification
  $Ver_{param,pk}$: $(\sigma, r') \mapsto m$ or reject

All these algorithms are deterministic, and the inputs ρ, ρ', r, and r' (if non empty) contain the randomization for the algorithms. They may have some specific format.

Signature schemes with appendix have the property that $\sigma = m \parallel s$. Signature schemes with message recovery usually have the property that $\sigma = \hat{m} \parallel s$ and that the whole message is $m = \hat{m} \parallel \bar{m}$, where $\bar{m}$ is the recovered part of the message. They typically have a lower bound for the size of the whole message, which is also the amount of message recovered.[1] Signature schemes with unique signature have the property that Ver is injective (with the exception of reject), which also implies that $Sign$ does not use any random r (deterministic signature scheme).

Two signature schemes are equivalent when the following conditions are satisfied:

- The possible values of $param$ are the same.
- The distributions of the pk generated are indistinguishable.
- Both verification algorithms are the same.
- The output of the respective $Sign$ operations for fixed m and random r are indistinguishable.

2.2 Ideal cipher model and random permutation model

The ideal cipher and random permutation models replace some components of the verification algorithm with calls to an oracle which is simulated by the reduction. The number of calls to these oracles is bounded e.g. by q_O. Because the actual computation of the idealized components takes time, a scheme with k bits of security with the appropriate time unit always has $q_O \leq 2^k$.

The random permutation model replaces a fixed permutation by a random one constructed in answer to the oracle calls.

The ideal cipher model replaces a keyed permutation by a random one constructed in answer to the oracle calls.

[1] This lower bound can be overcome by storing the length of the actual recovered part in $\bar{m}$. E.g. by padding $\bar{m}$ with a 1 followed by a string of 0. With this padding, one bit of message expansion is added.

A reduction algorithm in the ideal cipher and random permutation models always give random answers taken from the set of values that are consistent with previous answers. It has a total freedom for its answer to the first oracle query, and the other answers should not allow the forger to detect that the reduction algorithm took control of the oracle. Consistency for a random oracle means that the same input always give the same output. For a random permutation, two different inputs have different outputs, and queries for the inverse permutation should also be consistent.

To be able to maintain consistency, the reduction algorithm needs to keep tables of the subset of input/output pairs that has been developed to answer the queries. In other words, the reduction algorithm constructs the oracle tables.

3 OPSSR

Granboulan [4] showed what is the minimal possible message expansion for a given proven security requirement, and proposed OPSSR scheme that achieved the lower bound for message expansion.

3.1 The lower bound

Theorem 1 (Granboulan [4]). Minimal message expansion for k bits of security is k bits of message expansion and can only be obtained for a signature scheme with unique signature.

None of the previously published techniques achieve this lower bound. Most of them do not allow one to go under $2k$ bits of message expansion. The OPSSR scheme achieves this lower bound [4].

3.2 Basic OPSSR

We begin with a simplified version of the OPSSR scheme proposed by Granboulan.

This signature scheme can only sign messages of length $l - k$. It has two parameters: a trapdoor one-way bijection F with length l, and an arbitrary permutation E of blocks of size l. The random permutation model for E is used. In practice E can be based on a large block cipher with fixed key 0 and F can be the RSA bijection.

Let κ be a fixed value of k bits, e.g. 0^k. Valid signatures are generated by $m \mapsto F^{-1}(E^{-1}(m \parallel \kappa))$. The verification computes $m \parallel v = E(F(\sigma))$ and checks if $v \stackrel{?}{=} \kappa$.

Security proof

It is shown how it is possible to compute $F^{-1}(\tilde{y})$ for an arbitrary $\tilde{y}$ without knowing the trapdoor, but with access to a forger of OPSSR in the random permutation model.

The number of signature queries is bounded by q_S and the number of oracle queries (to E and E^{-1}) is bounded by q_O. For all answers to the $q_O + q_S \leq 2^k$ queries made by the forger, we will need to generate a value y uniformly. In this proof, one query has $y = \tilde{y}$ and all other queries have $y = F(x)$ for a random x. A table of (y, x) is stored, enabling the lookup of $F^{-1}(y)$.

First we send to the forger the description of F. Then we will answer to four types of queries and the oracle table is updated according to these answers.

- In response to a signature query for m, the reduction generates a value y and updates the oracle table with $y \stackrel{E}{\mapsto} m \parallel \kappa$. The answer is $x = F^{-1}(y)$.

  The signature query aborts if $E(y)$ was already defined. Since y is uniformly, and at most 2^k values were defined, this happens with probability at most $1/2^{l-k}$.

  The signature query also aborts if $y = \tilde{y}$. This has probability $1/2^k$.

- In response to a query for $E^{-1}(m \parallel \kappa)$, that is not in the table, a signature query for m is simulated. The answer is y.

  The oracle query aborts if $E(y)$ was already defined. This has probability at most $1/2^{l-k}$.

  The oracle query does not abort if $y = \tilde{y}$. If the forger later makes a query of a signature for m, then the signature query will abort.

- In response to a query for $E^{-1}(m \parallel v)$ with $v \neq \kappa$, a random value y is generated and the oracle table is updated with $y \overset{E}{\mapsto} m \parallel v$.

  The oracle query aborts if $E(y)$ was already defined. This has probability at most $1/2^{l-k}$.

- In response to a query for $E(y)$, random m and v are chosen, and the oracle table is updated with $y \overset{E}{\mapsto} m \parallel v$. The oracle query aborts if $v = \kappa$. This has probability $1/2^k$.

If $l \geq 2k+1$, then no query make the reduction abort with probability more than 2^{-k}. The total probability of non abortion is $(1-1/2^k)^{2^k} \leq 1/e$. The forger returns a forgery σ which is the signature of a message m with probability better than $1/2^k$. If this message was not in a query for $E^{-1}(m \parallel \kappa)$, then the signature is valid with probability $1/2^k$. Therefore this message was in a query for $E^{-1}(m \parallel \kappa)$ and a value y was generated. The reduction can compute $F^{-1}(\tilde{y})$ if this forgery corresponds to $y = \tilde{y}$, which happens with probability 2^{-k}. Therefore the success probability of the reduction is the one of the forger divided by at most $e \cdot 2^k$.

3.3 OPSSR

Basic OPSSR only allows one to sign messages of length $l-k$. To sign a message m of arbitrary length greater than $l-k$, the message is split $\hat{m} \parallel \bar{m} = m$ where $\bar{m}$ has length $l-k$ bits. $\hat{m}$ will be transmitted in the clear and $\bar{m}$ will be recovered with the Basic OPSSR scheme.

The security proof still holds if all answers to oracle queries are independent for different values of $\hat{m}$. Therefore the functions E and E^{-1} need to take $\hat{m}$ in their input. For better efficiency, a hash of $\hat{m}$ is used.

In practice, OPSSR will use a collision free hash function H with $2k$ bits of output and a keyed permutation E_k of blocks of size l with a key of size $2k$. The function E_k is modeled as an ideal cipher.

Signature generation. The massage is split $m = \hat{m} \parallel \bar{m}$ with $l-k$ bits in $\bar{m}$. Then $h = H(\hat{m})$ and $x = E_h^{-1}(\bar{m} \parallel \kappa)$ and $s = F^{-1}(x)$ are computed. The signed message is $\sigma = \hat{n} \parallel s$.

Signature verification. The signed message is split $\sigma = \hat{m} \parallel s$ with $s \in S$. Then $x = F(s)$ and $h = H(\hat{m})$ and $\bar{m} \parallel v = E_h(x)$ are computed. The signature is valid if $v = \kappa$.

4 Our padding scheme for signature schemes with message recovery in the random permutation model

We propose OPSSR in the random permutation model. Granboulan's OPSSR [4] used a hash function H, but our scheme use H as a random oracle which size is l bits. Like Granboulan's OPSSR, F is a trapdoor one-way bijection, and E is a random permutation. Furthermore our scheme uses a new random permutation G which size is l bits.

Signature generation. The massage is split $m = \hat{m} \parallel \bar{m}$ with $l-k$ bits in $\bar{m}$. Then $h = H(\hat{m})$ and $x = E^{-1}(\bar{m} \parallel \kappa) \oplus h$ and $z = G^{-1}(x)$ and $s = F^{-1}(z)$ are computed. The signed message is $\sigma = \hat{m} \parallel s$.

Signature verification. The signed message is split $\sigma = \hat{m} \parallel s$ with $s \in S$. Then $z = F(s)$ and $x = G(z)$ and $h = H(\hat{m})$ and $\bar{m} \parallel v = E(x \oplus h)$ are computed. The signature is valid if $v = \kappa$.

Security proof

We show how it is possible to compute $F^{-1}(\tilde{z})$ for an arbitrary $\tilde{z}$ without knowing the trapdoor, but with access to a forger of our padding scheme.

The number of signature queries is bounded by q_S and the number of oracle queries (to E, E^{-1}, G, and G^{-1}) is bounded by q_O. For all answers to the $q_O + q_S \leq 2^k$ queries made by the forger, we will need to generate a parameter uniformly. In this proof, one query has $z = \tilde{z}$ and all other queries have $z = F(s)$ for a random s.

First we send to the forger the description of F. Then we will answer to six types of queries and the oracle table is updated according to these answers.

- In response to a signature query for $m = \hat{m} \parallel \bar{m}$, if there is $y \rightarrow \bar{m} \parallel \kappa$ in E table, and there is $\hat{m} \rightarrow h$ in H table, and there is $x = y(:= E^{-1}(\bar{m} \parallel \kappa)) \oplus h$ in (x, s) table, the answer is $\hat{m} \parallel s$.

Else, the reduction generates s, y, and h at random, and updates E table with $y \to \bar{m} \parallel \kappa$, H table with $\hat{m} \to h$. Then the reduction computes $z = F(s)$ and $x = y \oplus h$, and updates (x, s) table with (x, s). The answer is $\hat{m} \parallel s$.

- In response to a query for $H(\hat{m})$, if there is $\hat{m} \to h$ in H table, the answer is h.

  Else, the reduction generates h at random, and updates H table with $\hat{m} \to h$, and return h.

- In response to a query for $G(z)$, if there is $z \to x$ in G table, the answer is x.

  Else, the reduction generates x at random, and updates G table with $z \to x$, and return x.

- In response to a query for $G^{-1}(x)$, if there is x in (x, s) table, the reduction computes $z = F(s)$, and return z.

  Else, the reduction generates s at random, and computes $z = F(s)$, and updates (x, s) table with (x, s), G table with $z \to x$. The answer is z.

- In response to a query for $E(y)$, if there is $y \to \bar{m} \parallel v$ in E table, the answer is $\bar{m} \parallel v$.

  Else, the reduction generates $\bar{m}$, v at random, and updates E table with $y \to \bar{m} \parallel v$, and return $\bar{m} \parallel v$.

- In response to a query for $E^{-1}(\bar{m} \parallel v)$, if there is $y \to \bar{m} \parallel v$ in E table, the answer is y.

  Else, the reduction generates y at random, and updates E table with $y \to \bar{m} \parallel v$, and return y.

By making such simulator, we can prove safety like Granboulan's proof. Since no query make the simulator abort with probability more than 2^{-k}, the total probability of non abortion is $(1-1/2^k)^{2^k} \leq 1/e$. If the forger returns a forgery σ which is valid signature with probability better than $1/2^k$, there should be a query for G^{-1}. The reduction can compute $F^{-1}(\tilde{z})$ if this forgery corresponds to $z = \tilde{z}$, which happens with probability 2^{-k}. Therefore the success probability of the reduction is the one of the forger divided by at most $e \cdot 2^k$.

5 Discussion

5.1 Difference in table size between the models

In the ideal cipher model, a keyed random permutations are assumed. When the size of permutation is l bits and the size of key is k bits, the size of a required table will be 2^{l+k}. On the other hand, in random permutation model, only 2^l is required.

5.2 Efficiency

Previous scheme [4] uses a hash function H, a random permutation E, and a trapdoor one-way bijection F. Our scheme uses a random oracle instead of the hash function H, and add a new random permutation G.

Therefore in our scheme a hash function is no longer used and three independent random oracles are used, while one random oracle used in the previous scheme.

6 Multi-key setting

Proofs of security for digital signature schemes only consider the case where the forger is able to ask signature queries for one public key, and has to make a valid signature for that public key.

However, it may be the case that computations done by the forger to attack one public key also help to attack another public key. Taking this into consideration is called the multi-key setting.

This consideration first appeared in a different form in the description of KCDSA for security against parameter manipulation [9]. As in [4], since the performance cost for having proofs of security against attacks in the multi-key setting is small, we also consider that it is better to take this into account.

To make the proof take the multi-key setting in account, one can make sure that all the components completely change if the public key changes.

For OPSSR using RSA, we have to meet the two following requirements:

- the best way to factor a lot of RSA numbers is to factor separately each of them.
- the function E, in the idealized world, depends on the public key.

The first requirement does not depend on the padding and may not be met by the RSA bijection, because it may be possible to factor a lot of RSA numbers faster than factoring them individually.

To meet the second requirement Granboulan proposed a straightforward and simple improvement of the OPSSR scheme. The only change is that $h = H(\hat{m}, pk)$. This can also be employed in our scheme.

References

[1] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures — how to sign with RSA and Rabin. *Lecture Notes in Computer Science*, Vol. 1070, pp. 399–416, 1996.

[2] Daniel R. L. Brown and Don B. Johnson. Formal security proofs for a signature scheme with partial message recovery. *Lecture Notes in Computer Science*, Vol. 2020, , 2001.

[3] J. S. Coron. Optimal security proofs for PSS and other signature schemes. In *Theory and Application of Cryptographic Techniques*, pp. 272–287, 2002.

[4] Louis Granboulan. Short signatures in the oracle model. In *Advances in Cryptology ASIACRYPT 2002*, Vol. 2501 of *Lecture Notes in Computer Science*, pp. 364–378. Springer-Verlag Heidelberg, 1–5 December 2002.

[5] Marc Joye J. S. Coron, David Naccache, and Pascal Paillier. Universal padding schemes for RSA.

[6] L. Goubin N. Courtois and J. Patarin. Quartz, 128-bit long digital signatures. In *Cryptographers' Track Rsa Conference 2001*, Vol. 2020 of *Lecture Notes in Computer Science*. Springer-Verlag Heidelberg, April 2001.

[7] David Naccache and Jacques Stern. Signing on a postcard. *Lecture Notes in Computer Science*, Vol. 1962, pp. 121–135, 2001.

[8] Leon A. Pintsov and Scott A. Vanstone. Postal revenue collection in the digital age. *Lecture Notes in Computer Science*, Vol. 1962, pp. 105–120, 2000.

[9] KCDSA Task Force Team. The korean certificate-based digital signature algorithm. *Lecture Notes in Computer Science*, Vol. 1514, pp. 175–186, 1998.