Short Signatures in the Random Oracle Model

Louis Granboulan*

École Normale Supérieure
Louis.Granboulan@ens.fr

Abstract. We study how digital signature schemes can generate signatures as short as possible, in particular in the case where partial message recovery is allowed. We give a concrete proposition named OPSSR that achieves the lower bound for message expansion, and give an exact security proof of the scheme in the ideal cipher model. We extend it to the multi-key setting. We also show that this padding can be used for an asymmetric encryption scheme with minimal message expansion.
Keywords: digital signature, padding, random oracle and ideal cipher models, proven security.

1 Introduction

1.1 Overview of the results

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. We show how to obtain message expansion as small as possible, with a concrete scheme having proven security in the ideal cipher model. The OPSSR technique is a padding for schemes based on trapdoor one-way bijections. Its performance cost is small, and its security is similar to the other schemes in the hash-then-invert paradigm.

The paper is organized as follows. Section 2 describes a formalism for digital signature schemes and describes the properties of the RSA trapdoor one-way bijection. Section 3 shows what are the lower bounds for message expansion. Section 4 describes OPSSR, which has minimal message expansion. Section 5 raises and solves a theoretical problem that arises when having an idealized security

* Part of this work has been supported by the Commission of the European Communities through the IST Programme under Contract IST-1999-12324 (NESSIE). This paper is NESSIE document **NES/DOC/ENS/WP5/021/2**.

model for a multi-key setting. Section 6 discusses open problems. Appendix A compares OPSSR with other paddings. Appendix B explains why OPSSR can also be used for encryption.

1.2 Related work

Many schemes have been proposed with short signatures [4, 11, 12, 17, 18], but their exact security is not proven to be equivalent to the underlying problem with the same parameters, because their security proofs are not tight. Therefore if the parameters are chosen to give short signatures, the security of those schemes is not proven.

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 message expansion in the PSS scheme [2], the Pintsov-Vanstone scheme [5] or the DSA-like schemes [17]. But those schemes do not achieve minimal signature length.

Coron [9] has shown 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 [10] have shown that the PSS padding, which was designed for signature, can also be used for encryption.

1.3 Our contribution

We introduce the definition of message expansion which generalizes the notion of short signature for schemes with message recovery. We show what is the minimal possible message expansion for a given proven security requirement. We describe a padding that achieves this lower bound and that can be used with RSA. This padding can be viewed as a generalization of PSSR and many other paddings. We also show that most current schemes proven secure in a idealized model should go under a small modification that increases their security in the multi-key setting.

2 Definitions

2.1 Digital signature schemes

Notations. If the variable x represents a value taken from a finite set $\mathcal{X}$ of n elements, the we say that the size of x is the value $\#x = \#\mathcal{X} = \log_2 n$, which may not be an integer.
If the elements of $\mathcal{X}$ can be represented by bit strings, then $\sharp x = \sharp\mathcal{X}$ is the length of the bit strings. Of course, $\#x \leq \sharp x$.
For variables $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, the corresponding element of $\mathcal{X} \times \mathcal{Y}$ is written as $x\|y$. This notations comes from the fact that if x and y are bit strings, then $x\|y$ is the concatenation of these strings.

Definitions. A signature scheme is described by the following four algorithms :

- a parameter generation $\mathsf{Generate} : \rho \mapsto \mathsf{param}$,
- a key generation $\mathsf{KeyGen}_{\mathsf{param}} : \rho' \mapsto (\mathsf{pk}, \mathsf{sk})$,
- a signature generation $\mathsf{Sign}_{\mathsf{param},\mathsf{pk},\mathsf{sk}} : (m, r) \mapsto \sigma$
- and a signature verification $\mathsf{Ver}_{\mathsf{param},\mathsf{pk}} : (\sigma, r') \mapsto m$ 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 $\sigma = m\|s$. Signature schemes with message recovery usually have the property that $\sigma = \hat{m}\|s$ and that the whole message is $m = \hat{m}\|\bar{m}$, 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 Security model and proofs

A (t, ε, q_S)-forger is able to make q_S queries for signatures and tries to produce a new valid signature. It succeeds in time t with probability ε. A signature scheme with no (t, ε, q_S)-forger is said to be (t, ε, q_S)-secure against adaptive chosen message attack. This security also means non-repudiation, because it proves that only the signer is able to make valid signed messages.

Weak security means that the forgery should be a valid signed message for a message that was not the input of a query. Strong security means that the forgery should be a valid signed message that was not the answer of a query. These notions are equivalent if the scheme has unique signature.

The security level of a scheme is k bits if there exists no (t, ε, q_S)-forger with $\log_2(t/\varepsilon) < k$. This value k depends of the time unit used for t.

Please note that any (t, ε, q_S)-forger for a signature scheme is also a (t, ε, q_S)-forger for all equivalent signature schemes.

A mathematical problem is (t', ε')-secure if there exist no algorithm that solves an arbitrary instance of the problem in time t' with probability better than ε'. The difficulty is k' bits if there exist no (t', ε')-solver with $\log_2(t'/\varepsilon') < k'$.

[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 proof of security is the description of how to construct a (t', ε')-solver (called *reduction algorithm*) when given access to a (t, ε, q_S)-forger. The reduction simulates an equivalent signature scheme and answers signature queries from the forger. The forgery is used to solve the problem. The reduction does not always succeed, partly because its simulation of an equivalent signature scheme may not be perfect, and partly because the forgery may be useless.
A tight proof of security has $t'/\varepsilon' \simeq t/\varepsilon$.

2.3 Idealized models

An idealized oracle model replaces 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 oracle model replaces hash functions by calls giving random output. The generic group model replaces the operations in some group by random answers that respect the group laws. 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.

A reduction algorithm in a idealized model always gives 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.

The random oracle model is widely used in the literature, the ideal cipher model and the generic group model have been used for proving the security of some specific schemes. Proofs in these models cannot generically be translated into the real world [6, 13], but it is widely believed that a proof in an idealized model give some confidence in the design of a cryptographic primitive. The random oracle model and ideal cipher model are very similar and we believe that they give similar confidence in cryptographic designs: a random oracle can be contructed from ideal ciphers, and it might be possible to build an ideal cipher from random oracles.

2.4 The RSA trapdoor one-way bijection

Bijection. A bijection with length l is a one-to-one and onto mapping F from a set $\mathcal{S}$ with 2^l elements to a set $\mathcal{L}$ with 2^l elements. It is a permutation if $\mathcal{S} = \mathcal{L}$. Let l' be equal to $\sharp\mathcal{S}$.

One-way. A bijection with length l is one-way with security k' bits if F is easy to compute but finding the preimage for a random $y \in \mathcal{L}$ (i.e. the unique $x \in \mathcal{S}$ such that $y = \mathsf{F}(x)$) is a problem with a difficulty of k' bits. Exhaustive search in $\mathcal{S}$ shows that $k' \leq l'$.

Trapdoor. It is a trapdoor one-way bijection if knowing some secret information (the trapdoor) makes easy the computation of F^{-1}.

Random-self-reducibility. The permutation is random-self-reducible if it has the following additional property. There exists a probabilistic algorithm R that takes an input $y \in \mathcal{L}$ and generates a uniformly distributed value $\tilde{y} \in \mathcal{L}$ such that knowing the value of $\mathsf{F}^{-1}(\tilde{y})$ makes it easy to compute $\mathsf{F}^{-1}(y)$.
If F is random-self-reducible, then it is always possible to compute $\mathsf{F}^{-1}(y)$ in time $2^{l/2}$, using the birthday paradox. A table of $2^{l/2}$ random $(x, \mathsf{F}(x))$ pairs is computed. A table of $2^{l/2}$ random $\tilde{y}$ values is generated with $\mathsf{R}(y)$. A collision $\mathsf{F}(x) = \tilde{y}$ gives the value for $\mathsf{F}^{-1}(\tilde{y})$, from which we deduce the value of $\mathsf{F}^{-1}(y)$. For a random-self-reducible trapdoor one-way permutation, we always have $k' \leq l/2$.

RSA permutation. The public parameter is a number n and an odd exponent e, the corresponding secret is the factorization $pq = n$ or the inverse $e^{-1} \bmod \phi(n)$. The function $\mathsf{F}(x) = x^e \bmod n$ is a permutation of the set $\mathbb{Z}_n^*$ of invertible integers modulo n. The trapdoor owner can compute $\mathsf{F}^{-1}(x) = x^{e^{-1}} \bmod n$.
This function F is a random-self-reducible trapdoor one-way permutation. Its random-self-reducibility comes from the algorithm R that generates a random $\tilde{x} \in \mathbb{Z}_n^*$ and returns $\tilde{y} = y \cdot \tilde{x}^e$. Then $\mathsf{F}^{-1}(y) = \mathsf{F}^{-1}(\tilde{y})/\tilde{x}$.
The best known technique to compute F^{-1} is to compute the factorization of n. Here is a table that gives estimates for minimal bit length of n to have some given security levels. The problem of the estimation of the difficulty of factoring large numbers is the object of some controversies and this table should only be understood as a proposal for basing our numbers on realistic estimates. It is not an attempt to solve this controversy. It is based on the hypothesis than the recent factorizations of 512 bits numbers needed a workfactor of 2^{56} and that the asymptotic complexity of the number field sieve is around $L_n[\frac{1}{3}, 1.9]$.
The formula for the following table is $k' = 12 + \log(L_{2^l}[\frac{1}{3}, 1.9])$.

Modulus length	l	512	768	1536	4096	8192
Bit security	k'	56	64	80	128	160

RSA bijection. For the RSA permutation the permuted set $\mathcal{L}$ is $\mathbb{Z}_n^*$ therefore the length l is not an integer. If an integer value is preferred, the RSA bijection is defined as follows.
The set $\mathcal{L}$ contains all integers in $\mathbb{Z}_n$ smaller than 2^l, and $\mathcal{S}$ is its preimage and $l' = \lceil l \rceil$. The computation of $\mathsf{F}(x)$ for $x \in \mathbb{Z}_n$ begins with $y = x^e \bmod n$. If $y \in \mathcal{L}$, it is the answer, else x is rejected because it is not an element of $\mathcal{S}$.

3 Minimal message expansion

3.1 The lower bound

A simple counting argument shows that for any signature scheme with random salt of length $\#r$ and message expansion λ, a signed message is valid with probability at least $1/2^{\lambda-\#r}$. Therefore the security level of the scheme is at most $\lambda - \#r$.

Theorem 1. *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: they don't allow one to go under $2k$ bits of message expansion. Our OPSSR scheme achieves this lower bound.

3.2 Signature schemes with appendix

Coron [9] proved that a signature scheme with unique signature cannot have a tight security proof, and that the lower bound for the relation between the security k of the scheme and the security k' of the underlying problem is $k' \simeq k + \log_2 q_S$.
A signature scheme with appendix based on a problem with security k' has an appendix of length at least k'. Therefore the message expansion for a deterministic signature scheme with appendix is at least $k + \log_2 q_S$.
Randomized signature schemes can enhance the tightness of the proof, but at the cost of a random seed that appears in the signed message. Each bit of gained tightness costs one bit of random seed.

Theorem 2. *The lower bound for a signature scheme with appendix having k bits of security against a forger allowed to make q_S signature queries is a message expansion of $k + \log_2 q_S$ bits.*

None of the previously published techniques achieves this lower bound, and the problem is still open whether it is possible to achieve it or not.

4 The OPSSR padding

4.1 Some previous work: PFDH and PSSR

Quick introduction. Full Domain Hash was formally described and proved by Bellare and Rogaway in [2]. Their proof shows that in the random oracle model with at most q_H hash queries the security k of FDH is related to the security k' of the underlying trapdoor one-way bijection by $k' \simeq k + \log_2(q_H + q_S)$. Coron has shown in [8] that random-self-reducibility helps to improve the proof and obtains $k' \simeq k + \log_2 q_S$. Coron also introduced in [9] a probabilistic variant of Full Domain Hash that we describe below.

PFDH. The two components are a random-self-reducible trapdoor one-way bijection F and a cryptographic hash function H. The verification of a signed message splits $\sigma = m\|r\|s$ and says the signature is valid if $s \in \mathcal{S}$ and $\mathsf{H}(m\|r) = \mathsf{F}(s)$. It outputs the message m.
The trapdoor owner signs the message m by first generating a random salt r, then computing $s = \mathsf{F}^{-1} \circ \mathsf{H}(m\|r)$, and returns $\sigma = m\|r\|s$.
The proof shows that if $\#r \geq \log_2 q_S$ then $k' \simeq k$ and if $\#r \leq \log_2 q_S$ then $k' \simeq k + \log_2 q_S - \#r$. We can notice that the output length of the hash function is equal to the length l of the bijection and that the message expansion is $l' + \#r$. Because of random-self-reducibility, $l' \geq 2k'$. PFDH does not allow better message expansion than $2k$.

PSSR. This scheme was introduced in [2] and its optimal proof of security is in [9]. It is a modification of PFDH by adding recovery of the salt and of part of the message.
The hash function H has output length $2k$ and an additional cryptographic hash function G with input length $2k$ and output length $l-2k$ is needed and is modeled as a random oracle.
The verification splits $\sigma = \hat{m}\|s$, checks that $s \in \mathcal{S}$, computes $a\|h = \mathsf{F}(s)$ and $\bar{m} = a \oplus \mathsf{G}(h)$, and checks if $\mathsf{H}(\hat{m}\|\bar{m}) \stackrel{?}{=} h$. It computes $m\|r = \hat{m}\|\bar{m}$ and outputs the message m.
The trapdoor owner signs the message m by first generating a random salt r, then computing $\hat{m}\|\bar{m} = m\|r$ where $\bar{m}$ is $l - 2k$ bits long. Then $h = \mathsf{H}(m\|r)$ and $a = \bar{m} \oplus \mathsf{G}(h)$ are computed. The signed message is $\hat{m}\|\mathsf{F}^{-1}(a\|h)$. PSS is the special case where $\#r = l - 2k$.
The security proof is very similar to the proof for PFDH and shows that PSSR has the same security as PFDH. The addition of G does not weaken the scheme because the probability of a collision in the input of G is low. This is due to the fact that the input size of G is twice the security level of the scheme. The message expansion with PSSR is $2k + \#r$. PSSR does not allow better message expansion than $2k$.

Replacing a XOR with a block cipher. The idea of improving a padding by replacing a XOR with a block cipher was introduced by Jonsson [14] for an improvement of OAEP+ named OAEP^{++}. The same can be done with PSS. It only changes the security properties of the padding when used for asymmetric encryption.

4.2 Basic OPSSR

OPSSR means Optimal Padding for Signature Schemes with message Recovery. We begin with a simplified version of our OPSSR scheme.
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 security k' 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 $m \mapsto \mathsf{F}^{-1}(\mathsf{E}^{-1}(m\|\kappa))$. The verification computes $m\|v = \mathsf{E}(\mathsf{F}(\sigma))$ and checks if $v \stackrel{?}{=} \kappa$.

Security proof. We show how it is possible to compute $\mathsf{F}^{-1}(y)$ for an arbitrary 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 distributed in $\mathcal{L}$. In this proof, one query has $y' = y$ and all other queries have $y' = \mathsf{F}(x')$ for a random x'. A table of (y', x') is stored, enabling the lookup of $\mathsf{F}^{-1}(y')$.

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 $y' \stackrel{\mathsf{E}}{\mapsto} m\|\kappa$. The answer is $x' = \mathsf{F}^{-1}(y')$.
  The signature query aborts if $\mathsf{E}(y')$ was already defined. Since y' is uniformly distributed in $\mathcal{L}$, and at most 2^k values were defined, this happens with probability at most $1/2^{l-k}$.
  The signature query also aborts if $y' = y$. This has probability $1/2^k$.
- In response to a query for $\mathsf{E}^{-1}(m\|\kappa)$, that is not in the table, a signature query for m is simulated. The answer is y'.
  The oracle query aborts if $\mathsf{E}(y')$ was already defined. This has probability at most $1/2^{l-k}$.
  The oracle query does not abort if $y' = y$. If the forger later makes a query of a signature for m, then the signature query will abort.
- In response to a query for $\mathsf{E}^{-1}(m\|v)$ with $v \neq \kappa$, a random value y'' is generated and the oracle table is updated with $y'' \stackrel{\mathsf{E}}{\mapsto} m\|v$.
  The oracle query aborts if $\mathsf{E}(y'')$ was already defined. This has probability at most $1/2^{l-k}$.
- In response to a query for $\mathsf{E}(y'')$, random m and v are chosen, and the oracle table is updated with $y'' \stackrel{\mathsf{E}}{\mapsto} m\|v$.
  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 $(1 - 1/2^k)^{2^k} \geq 1/e$.
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 $\mathsf{E}^{-1}(m\|\kappa)$, then the signature is valid with probability $1/2^k$. Therefore this message was in a query for $\mathsf{E}^{-1}(m\|\kappa)$ and a value y' was generated. The reduction can compute $\mathsf{F}^{-1}(y)$ if this forgery corresponds to $y = y'$, which happens with probability

2^{-k}. Therefore the success probability of the reduction is the one of the forger divided by at most $e2^k$.
The running time t of the (real world) forger includes some actual computations of F, E and E^{-1}. The answer to an oracle query by the reduction algorithm needs some table lookups and at most one computation of F. Under the hypothesis that the time for all these computations are similar, the running time for the reduction is γt for some small constant γ.
A difficulty level of $k' \simeq 2k$ is needed and this scheme has minimal message expansion.

Random-self-reducibility. The same technique as in [8] can be used when F is random-self-reducible. This technique consists in a change of the way the values y' are generated. The full details on how to optimize the parameters can be found in Coron's papers.
The basic idea is to have a proportion α/q_S of the values y' generated with the algorithm R. A signature query will abort if such a y' was generated, which happens with probability α. However, if the reduction does not abort, then its success probability is the success probability of the forger divided by q_S/α.
This idea applies to OPSSR as well and a difficulty level of $k' \simeq k + \log_2 q_S$ is needed and the scheme has minimal message expansion.

Randomization. The same technique as in [2, 9] can be used to enhance the tightness of the reduction, if F is random-self-reducible. The message m is padded with a random salt r before being signed. The signature verification works as before but the salt is discarded.
The reason why this improves the tightness of the reduction is that a much higher proportion of the values y' can be generated with the algorithm R, because a signature query can choose a value for the salt for which $y' = \mathsf{F}(x')$.
This idea applies to OPSSR as well and a difficulty level of $k' \simeq k + \log_2 q_S - \#r$ is sufficient when the salt has length $\#r \leq \log_2 q_S$. However this randomized scheme does not have minimal message expansion, because the salt is recovered and the expansion is $k + \#r$.

4.3 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 $\hat{m}\|\bar{m} = m$ 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 message is split $m = \hat{m}\|\bar{m}$ with $l - k$ bits in $\bar{m}$. Then $h = \mathsf{H}(\hat{m})$ and and $x = \mathsf{E}_h^{-1}(\bar{m}\|\kappa)$ and $s = \mathsf{F}^{-1}(x)$ are computed. The signed message is $\sigma = \hat{m}\|s$.
Signature verification. The signed message is split $\sigma = \hat{m}\|s$ with $s \in \mathcal{S}$. Then $x = \mathsf{F}(s)$ and $h = \mathsf{H}(\hat{m})$ and $\bar{m}\|v = \mathsf{E}_h(x)$ are computed. The signature is valid if $v = \kappa$.

4.4 RSA-OPSSR and comparison with other schemes

RSA-OPSSR. With a goal of 80 bits of security and $\log_2 q_S \simeq 48$, OPSSR can be used for a proven (in the ideal cipher model) deterministic signature algorithm with 80 bits of message expansion with a 4096 bits RSA (or whatever is the modulus size for 128 bits of RSA security), or for a proven probabilistic signature algorithm with 128 bits of message expansion with 48 bits of salt and 1536 bits RSA (80 bits of RSA security).

RSA-PSSR. With a goal of 80 bits of security and $\log_2 q_S \simeq 48$, PSSR can be used for a proven (in the random oracle model) deterministic signature algorithm with 160 bits of message expansion with a 4096 bits RSA (or whatever is the modulus size for 128 bits of RSA security), or for a proven probabilistic signature algorithm with 208 bits of message expansion with 48 bits of salt and 1536 bits RSA (80 bits of RSA security).

PVSSR or Naccache-Stern. With a goal of 80 bits of security and $\log_2 q_S \simeq 48$, They can be used for a proven (in the generic group model) probabilistic signature algorithm based on 160 bits elliptic curve discrete logarithm and achieving 240 to 208 bits of message expansion.

5 Idealized security models and multi-key setting

5.1 The 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 [15, section 4.2].
Since the performance cost for having proofs of security against attacks in the multi-key setting is small, we believe that signature schemes should take this into account.

5.2 A concrete solution

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

- the best way to factor a bunch 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 bunch of RSA numbers faster than factoring them individually [7]. [2]

To mett the second requirement we propose here a straightforward and simple improvement of the OPSSR scheme. The only change is that $h = \mathsf{H}(\hat{m}, \mathsf{pk})$.
All other signature schemes proven secure in an idealized model can benefit from a similar improvement of their security. For example with RSA-PSS, it is sufficient to include the public key in the input of both hash functions H and G.

6 Discussion and open problems

6.1 Large block cipher

OPSSR with 4096 bits RSA needs a block cipher able to encrypt blocks of 512 bytes. No such block cipher has been widely studied. Using a deterministic mode of operation of a 8 or 16 byte block cipher is not a solution because it is not a valid implementation of the ideal cipher model.
Two research directions can be proposed.

- Is it possible to replace this ideal cipher with random oracles, for example with a sufficient number of Feistel rounds ?
- How many rounds of the generalization of Rijndael that is based on 512 parallel S-boxes and an adequate MDS matrix are needed to have a secure cipher ?

6.2 Optimal trapdoor one-way permutations

Another drawback of using OPSSR with RSA is that even if the message expansion is small, the minimal length for a signed message is equal to the size $l' = \lceil l \rceil$ of the RSA modulus. Optimal trapdoor one-way permutation have minimal input length and would minimize this value.
With an optimal trapdoor one-way (non random-self-reducible) permutation, i.e. that permutes l bits blocks with $k' = l$ bits of security, (deterministic) OPSSR can be applied with $l \simeq 2k$. The minimal length for a signed message is $2k$ and the message expansion is k.

[2] This requirement is not met for schemes with security based on the hardness of the discrete logarithm in some fixed integer multiplicative group. The multi-key setting needs distinct groups for distinct public keys.

With an optimal random-self-reducible trapdoor one-way permutation, i.e. that permutes l bits blocks with $k' = l/2$ bits of security, (deterministic) OPSSR can be applied with $l = 2k' = 2(k + \log_2 q_S)$. The minimal length for a signed message is $2k + 2\log_2 q_S$ and the message expansion is k. Randomized OPSSR can also be applied with $\#r = \log_2 q_S$ and $l = 2k' = 2k$. The minimal length for a signed message is $2k + \log_2 q_S$ and the message expansion is $k + \log_2 q_S$.
But the problem of finding an explicit candidate for being an optimal (random-self-reducible) trapdoor one-way permutation is old and still unsolved.

6.3 Avoiding idealized security models

The other important open problem is how to get rid of the idealized oracle models, which are the core of our proofs of security. Signature schemes based on chameleon hash functions or similar techniques cannot be an answer, because the information needed to commit to some hash has to be in the signed message, and will increase the message expansion.

Acknowledgements

We would like to thank one anonymous referee for many interesting remarks that helped to improve this paper.

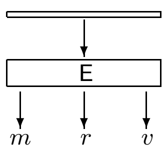
References

1. M. Bellare and P. Rogaway. Optimal asymmetric encryption - how to encrypt with RSA. *Proc. Eurocrypt'94*, LNCS 950, pages 92-111, May 1994. Available from `http://www-cse.ucsd.edu/users/mihir/crypto-research-papers.html`.
2. M. Bellare and P. Rogaway. The exact security of digital signatures: how to sign with RSA and Rabin. *Proc. Eurocrypt'96*, LNCS 1070, pages 399-416, May 1996. Revised version available from `http://www-cse.ucsd.edu/users/mihir/crypto-research-papers.html`.
3. D. Boneh. Simplified OEAP for the RSA and Rabin functions. *Proc. Crypto'01*, LNCS 2139, pages 275-291, Aug. 2001. Available at `http://crypto.stanford.edu/~dabo/papers/saep.ps`.
4. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *Proc. Asiacrypt'01*, LNCS 2248, pages 514-532, Dec. 2001. Available at `http://crypto.stanford.edu/~dabo/papers/weilsig.ps`.
5. D. Brown and D. Johnson. Formal Security Proofs for a Signature Scheme with Partial Message Recovery. 2000. Available at `http://www.cacr.math.uwaterloo.ca/techreports/2000/corr2000-39.pdf`.
6. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *Proc. STOC'98*, ACM, pages 209-218, May 1998. Available at `http://theory.lcs.mit.edu/~oded/rom.html`.
7. D. Coppersmith. Modifications of the Number Field Sieve. *Journal of Cryptology*, vol. 6, n. 3, pages 169-180, 1993.

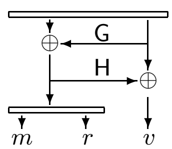
8. J.-S. Coron. On the exact security of Full Domain Hash. *Proc. Crypto'00*, LNCS 1880, pages 229-235, Aug. 2000. Available at `http://www.eleves.ens.fr/home/coron/fdh.ps`.
9. J.-S. Coron. Optimal security proofs for PSS and other signature schemes. *Proc. Eurocrypt'02*, LNCS 2332, pages 272-287, May 2002. Available at `http://eprint.iacr.org/2001/062/`.
10. J.-S. Coron, M. Joye, D. Naccache and P. Paillier. Universal Padding Schemes for RSA. *Proc. Crypto'02*, LNCS, Aug. 2002. Available at `http://eprint.iacr.org/2002/115/`.
11. N. Courtois, M. Finiasz, and N. Sendrier. How to Achieve a McEliece-based Digital Signature Scheme. *Proc. Asiacrypt'01*, LNCS 2248, 157-174, Dec. 2001. Available at `http://www.minrank.org/mceliece/`.
12. N. Courtois, L. Goubin, and J. Patarin. Quartz, 128-bit long digital signatures. *Cryptographers' Track Rsa Conference 2001*, LNCS 2020, Apr. 2001. Available at `http://www.minrank.org/quartz/`.
13. A. Dent. Adapting the weaknesses of the Random Oracle model to the Generic Group model. To appear in *Asiacrypt'02*. Available at `http://eprint.iacr.org/2002/086/`.
14. J. Jonsson. An OAEP variant with a tight security proof. *Manuscript*, Mar. 2002. Available at `http://eprint.iacr.org/2002/034/`.
15. KCDSA Task Force Team. The Korean Certificate-based Digital Signature Algorithm. *Proc. Asiacrypt'98*, LNCS 1514, pages 175-186, Oct. 1998. Also available at `http://grouper.ieee.org/groups/1363/P1363a/PSSigs.html` as an IEEE P1363a submission.
16. K. Kobara and H. Imai. OAEP++ : A very simple way to apply OAEP to deterministic OW-CPA primitives. *Manuscript*, Aug. 2002. Available at `http://eprint.iacr.org/2002/130/`.
17. D. Naccache and J. Stern. Signing on a Postcard. *Proc. FC'00*, LNCS 1962, pages 121-135, Feb. 2000. Available at `http://grouper.ieee.org/groups/1363/Research/contributions/Postcard.ps`.
18. L. Pintsov and S. Vanstone. Postal revenue collection in the digital age. *Proc. FC'00*, LNCS 1962, pages 105-120, Feb. 2000. Available at `http://www.cacr.math.uwaterloo.ca/techreports/2000/corr2000-43.ps`. Analysed in [5].
19. V. Shoup. OAEP Reconsidered. *Proc. Crypto'01*, LNCS 2139, pages 239-259, Aug. 2001. Available at `http://www.shoup.net/papers/oaep.pdf`.

A Comparison of OPSSR with other paddings

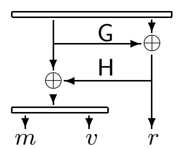
Many other paddings have been proposed. We show below the description of those paddings, when used for private decryption in an asymmetric encryption scheme or for public verification in a digital signature scheme. Their output is the message m, a random seed r and a validation value v. A non zero value for v leads to a rejection.



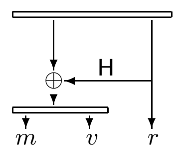
OPSSR



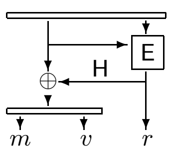
PSS-R [2]



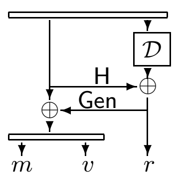
OAEP [1]



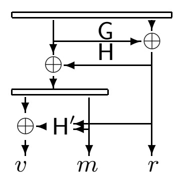
SAEP [3]



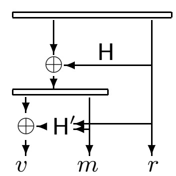
OAEP^{++} [14]



OAEP++ [16]



OAEP+ [19]



SAEP+ [3]

All these paddings have security proofs, where the internal components (the hash functions G, H and H' and the encryption functions E) are modelized as random oracles and ideal ciphers.
They are special implementations of OPSSR where the encryption function has a special form, but the security proof for OPSSR does not apply to this special form.
For example with PSS-R, if v is k bits long, then it is easy to find a collision $\mathsf{H}(m\|r) = \mathsf{H}(m'\|r')$ in time $2^{k/2}$. If E is the corresponding encryption function for OPSSR (an unbalanced 2-rounds Feistel scheme based on G and H), that means that if $\mathsf{E}^{-1}(m\|r\|v) = a\|b$ is known, then the attacker can deduce that $\mathsf{E}^{-1}(m'\|r'\|v) = a'\|b$ where $a' = a \oplus m\|r \oplus m'\|r'$. This is incompatible with the ideal cipher model for E.

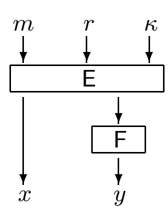
B OPSSR is an optimal universal padding scheme

Basic OPSSR for encryption. This scheme can only encrypt messages of length $l - k$. It is built on a trapdoor one-way permutation F and a permutation E of blocks of size l.
The encryption of the message m is $\mathsf{F} \circ \mathsf{E}(m\|\kappa)$. The decryption of the cipher c is $m\|v = \mathsf{E}^{-1} \circ \mathsf{F}^{-1}(c)$ and is rejected if $v \neq \kappa$.

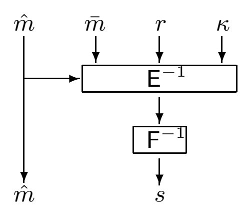
To improve the tightness of the security proof, the scheme needs to be randomized. The encryption of m is $\mathsf{F} \circ \mathsf{E}(m\|r\|\kappa)$ and the decryption $m\|r\|v = \mathsf{E}^{-1} \circ \mathsf{F}^{-1}(c)$ is rejected if $v \neq \kappa$.

OPSSR for encryption. To be able to encrypt arbitrary-length messages, one can use the same technique as Jonsson [14] and notice that the whole $\mathsf{E}(m\|r\|\kappa)$ does not need to be permuted with F. To encrypt m we compute $x\|y = \mathsf{E}(m\|r\|\kappa)$ and the cipher is $c = x\|\mathsf{F}(y)$.

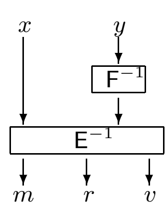
Properties. All properties of PSS described in [10] for a dual encryption+signature usage of the same public key are also valid for OPSSR. Moreover, the security reduction for the encryption scheme is as tight as for OAEP^{++}. This can be proved with the technique from [14].
The main advantage of using OPSSR for encryption rather than these other paddings is that the message expansion is minimal, like it is the case for signature with OPSSR. The main disadvantage is that the encryption of a message of n bits with k bits of security and k bits of expansion needs a random permutation of blocks of $n + k$ bits.



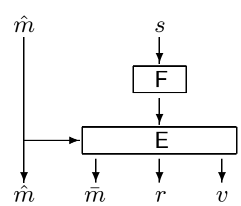
OPSSR for encryption



OPSSR for signature



OPSSR for decryption



OPSSR for verification