How to repair ESIGN

Louis Granboulan*

École Normale Supérieure
Louis.Granboulan@ens.fr

Abstract. The ESIGN signature scheme was provided with an inadequate proof of security. We propose two techniques to repair the scheme, which we name ESIGN-D and ESIGN-R.
Another improvement of ESIGN is encouraged, where the public key is hashed together with the message. This allows to have a security proof in the multi key setting.
Additionally, the lower security of ESIGN compared to RSA-PSS leads to suggest that a common public key is used for ESIGN and RSA-PSS, leaving to the signer the choice between fast signature or better security.

1 Description of ESIGN and its security proof

1.1 Introduction

ESIGN is a digital signature scheme whose complete description can be found in the submission to NESSIE [4]. The public key is a composite number $n = p^2q$ and the best known attack against ESIGN has to find this factorization. However, the published proof of its security relies on the intractability of finding an approximation of the e-th root modulo n, where $e \geq 8$.

In this document we will use some numeric estimations for the security induced by the intractability of factorization. We base our estimation on a workfactor of 2^{56} for the factorization of a 512 bit number with NFS and for the factorization with ECM of a number having a 190 bits factor. Then a computing power of 2^{64} should be able to factorize a p^2q of 800 bits, 2^{80} should factorize 1536 bits, 2^{128} should factorize 6000 bits and 2^{160} should factorize 10000 bits.

* 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/WP5/019/3** and is published in the proceedings of SCN'02

1.2 Description of the scheme

The security is measured by the size l of the factors of n. The key generation computes two random prime numbers p and q of size l such that $n = p^2 q$ is of size $3l$. The public key contains n, the private key contains p and pq. The value $e \geq 8$ is a parameter of the scheme and can be considered to be part of the public key. However, it is recommended that all ESIGN signatures in a given application use the same e. A collision-free hash function H with output size $l-1$ is also a parameter of the scheme. Let $\mathsf{f}(x) = \lfloor \frac{x^e \bmod n}{2^{2l}} \rfloor$, the approximate e-th power function.
To sign a message m, we make four steps.

1. Compute $h \leftarrow \mathsf{H}(m)$.
2. For a random $r < pq$ compute $u \leftarrow h \cdot 2^{2l} - r^e \bmod n$ and $v \leftarrow \lceil \frac{u}{pq} \rceil$. Compute $w \leftarrow v \cdot pq - u$ and get another r until $w < 2^{2l-1}$.
3. Compute $t \leftarrow v/(e \cdot r^{e-1}) \bmod p$ and $s \leftarrow r + t \cdot pq$.
4. The signed message is $\sigma = m \| s$ where s has length $3l$.

The verification of a signed message $\sigma = m \| s$ checks if $\mathsf{H}(m) \stackrel{?}{=} \mathsf{f}(s)$.
In some variants of ESIGN the value r is required to be prime with n. It is necessary to make the signature algorithm always work, but the failure probability is extremely low.
ESIGN is a valid signature scheme because for a generated signature the value of $s^e \bmod n$ is $0 \| h \| 0 \| w = h \cdot 2^{2l} + w$. We may notice that the output distribution of the signature algorithm gives fixed h and uniform random w.

1.3 The proof of security

It is similar to the classic proof [1] of Full Domain Hash signature schemes in the random oracle model, based on the intractability to find a preimage for f. The $(t, \varepsilon, q_S, q_H)$-forger is allowed to make q_S signature queries, q_H hash queries and outputs a valid forgery with probability ε after a running time t. The reduction is an algorithm that interacts with the forger and is able to solve any instance of this intractable problem. It is the reduction that answers signature and hash queries, and this simulation is perfect if it has the same statistical distribution as a real signer. Coherent parameters are $q_S \leq q_H$ and also $t \simeq q_H$.

Here is a short description of the security proof for ESIGN.
The reduction receives a challenge: a $n = p^2 q$ number with unknown factors and a $l-1$ bits number η. The goal is to find a value s such that

$\mathsf{f}(s) = \eta$. This is the AER (approximate e-th root) problem for η and modulo n.
One integer $j \leq q_S + q_H$ is randomly selected. To answer a query for $\mathsf{H}(m)$, random values s are generated until the most significant and the $2l$-th bit of $s^e \bmod n$ are 0 and $h = \mathsf{f}(s)$ is returned (this value h can be proven to have uniform distribution). To answer a signature query for m, a query for $\mathsf{H}(m)$ is simulated and the signature is the corresponding s. The exception is the j-th hash query which answers η.
The simulation of a signer succeeds if no signature query is made for the j-th hash query. That happens with probability $(1-\frac{1}{q_S+q_H})^{q_S} \simeq \frac{q_H}{q_S+q_H} \geq \frac{1}{2}$. The reduction can find an approximate e-th root of η if the forgery corresponds to the j-th hash query. That happens with probability $\frac{1}{q_S+q_H}$.
Therefore the conclusion is that if finding an approximate e-th root is $(t, \varepsilon/(q_S+q_H))$-hard, then ESIGN is $(t, \varepsilon, q_S, q_H)$-secure. For example if we aim at $k = 80$ bits of security, we need $\log_2(t/\varepsilon) = k$ and $\log_2 q_H = k$, and factoring n should need a workfactor of 2^{160}, which is probably obtained with $l = 3300$.

1.4 The mistake

Stern and al. noticed [5] that the security proof for ESIGN makes the (invalid) hypothesis that it is a deterministic signature scheme. Indeed, the simulator gives the same answer to multiple signature queries for the same message.
Therefore the conclusion of the proof is invalid. The forger can make two signature queries for the same message. If the signer is simulated by the reduction then these will give the same answer. If it is a real signer the probability that they are identical is 2^{-2l}. If the forger makes q_S signature queries for the same message he will detect the simulator with probability $1 - 2^{-2l(q_S-1)}$.

1.5 A proof based on claw-free permutations[1]

It is similar to the improved proof of RSA-FDH by Coron [2] and is based on the claw intractability of f together with another function g that has uniform $l-1$ bits output.
It has the same flaw as the classic proof but does apply to ESIGN-D and ESIGN-R.

[1] This paragraph was not in the preproceedings distributed at the SCN'02 conference ; it uses the results of Dodis and Reyzin [3] presented at the conference.

The security proof works as follows: most queries for $\mathsf{H}(m)$ are answered as in the classic FDH proof, but a proportion of about $\frac{\alpha}{q_S}$ is answered with the value $\mathsf{g}(z)$ for a random z. To answer a signature query for m, a query for $\mathsf{H}(m)$ is simulated and the signature is the corresponding s.
The simulation of a signer succeeds if no signature query corresponds to a hash query that was answered with some $\mathsf{g}(z)$. That happens with probability $(1-\frac{\alpha}{q_S})^{q_S} \simeq 1-\alpha$.
The reduction can find a claw $\mathsf{f}(s) = \mathsf{g}(z)$ if the forgery corresponds to such a hash query. That happens with probability $\frac{\alpha}{q_S}$. Therefore the conclusion is that if finding a claw is $(t, \varepsilon/q_S)$-hard, then ESIGN is about $(t, \varepsilon, q_S, q_H)$-secure.

A natural candidate is $\mathsf{g}(z) = \lfloor \frac{\eta z^e}{2^{2l}} \rfloor$ for a random η. This proof can be based on the intractability of the following *Claw-AER problem.*
An instance of the problem is a $n = p^2 q$ number with unknown factors and a target η. A solution is a pair of values s and z such that $\lfloor \frac{s^e}{2^{2l}} \rfloor = \lfloor \frac{\eta z^e}{2^{2l}} \rfloor$.
However, the Claw-AER problem has never been studied before and it is risky to base the security of a scheme on this assumption.

2 Repairing ESIGN

We propose two simple ways for repairing ESIGN such that it is provable.

2.1 ESIGN-D : making ESIGN deterministic

We build a deterministic variant of ESIGN which can have a proven security of k bits. A one-way function ϕ with output randomly distributed in the numbers modulo pq will be used.
An additional k bit value Δ is included in the secret key. Only step 2 of the signature generation is changed. Instead of generating random values r until an adequate value is found, the signature algorithm uses the values $\phi(\mathsf{H}(m)\|\Delta\|i)$ for $i = 0, 1, 2, \ldots$
The proof of this scheme is exactly the one in paragraph 1.3. It is important that the values $w = (s^e \bmod n) \bmod 2^{2l-1}$ from the simulation are indistinguishable from the values generated by the signature algorithm. It is the case because Δ is secret and sufficiently large to withstand exhaustive search, therefore the output of ϕ is unpredictible for the attacker.
The function ϕ can be built based on a cryptographic hash function. One possibility is to have a $4l$ bits output reduced modulo pq, another possibility is to have a $2l+1$ bits output and increment i until it is smaller than pq. We may want to force an upper bound on i to be able to represent

it in a fixed length bit string, its representation can be of variable length if the hash function accepts variable length input.

2.2 ESIGN-R : making randomized ESIGN signature simulable

We increase the randomness of signature generation such that the reduction will be able to simulate the probabilistic output of the signature oracle.
The new signature algorithm needs an additional random input ρ which is appended to the message. Its length should be more than $2\log_2 q_S$ bits.

1. Compute $h \leftarrow \mathsf{H}(m\|\rho)$.
2. For a random $r < pq$ compute $u \leftarrow h \cdot 2^{2l} - r^e \bmod n$ and $v \leftarrow \lceil \frac{u}{pq} \rceil$. Compute $w \leftarrow v \cdot pq - u$ and get another r until $w < 2^{2l-1}$.
3. Compute $t \leftarrow v/(e \cdot r^{e-1}) \bmod p$ and $s \leftarrow r + t \cdot pq$.
4. The signed message is $\sigma = m\|\rho\|s$.

The verification of a signed message $\sigma = m\|\rho\|s$ checks if $\mathsf{H}(m\|\rho) \stackrel{?}{=} \lfloor \frac{s^e}{2^{2l}} \rfloor$.
The proof needs to be adapted. To answer signature queries for m, a random value ρ is selected, a query for $\mathsf{H}(m\|\rho)$ is simulated and the signature is $m\|\rho\|s$.
If two signature queries are answered with the same value for ρ, then they also have the same s and the simulations fails as with the original ESIGN scheme. But the length of ρ is more than $2\log_2 q_S$ bits and collisions in ρ are very unprobable, therefore the simulation's answers to signature queries have uniform random values w.

2.3 Comparison of both techniques

From a performance point of view, ESIGN-D adds on average two calls to ϕ. Both cost a few calls to e.g. SHA-1 with small input, plus a modular reduction $\bmod pq$. What dominates this additional performance cost is the (two) modular reductions.
ESIGN-R adds a few bytes in the input of H, both for signature and verification. This is much faster than the modular reductions.
ESIGN-R also increases the length of the signature by the number of bits of ρ, which can be 80 if $\log_2 q_S = 30$. But this overhead is small compared to the $3l = 10000$ bits of s that are needed for proven 80 bits security.
ESIGN-R has a tight security proof under the Claw-AER assumption (this is the direct translation of the security proof of PFDH or PSS).

Both algorithms can be modified to allow (partial) message recovery, by using another hash function G in a way similar to PSS. The hash function H has output of $2k$ bits whereas G has output of $l-1-2k$ bits. The recovered message $\bar{m}$ has length $l-1-2k$ bits, and instead of using $\mathsf{H}(m)$ the signature algorithm uses $\mathsf{H}(m)\|(\bar{m} \oplus \mathsf{G}(\mathsf{H}(m))$. The verification algorithm computes $h\|a \leftarrow \lfloor \frac{s^e}{2^{2l}} \rfloor$ and recovers $\bar{m} \leftarrow a \oplus \mathsf{G}(h)$ before checking if $\mathsf{H}(m) \stackrel{?}{=} h$.

ESIGN-R apparently has a slightly better performance, the same proven security under AER and better security under Claw-AER. However, an external source of randomness is needed and this has a performance cost that can dominate the signing time. Moreover deterministic signatures might be mandatory in some applications. We prefer ESIGN-D.

2.4 A concluding remark

Because of the bad efficiency of the security reduction to AER, both modifications of ESIGN don't increase the security if we use a 1152 or 1536 bits modulus, because for these sizes the cost of factoring can be estimated to be between 2^{64} and 2^{80}. We only proved that the scheme has at least 32 to 40 bits of security, and this is within the computing power of most computers.

However ESIGN-D avoids the (usually expensive) use of external randomness. We strongly suggest the replacement of ESIGN with ESIGN-D in all contexts. If heuristic security is sufficient, a 1536 bits modulus can be used. If provable security is a concern, 10000 bits are needed.

3 The multi key setting

The multi key setting corresponds to the case where many public keys are used for one digital signature scheme, and the attacker wants to forge a new signature for each of these public key.
A scheme is secure in the multi key setting if the best strategy for the forger is to attack each public key independently. None of ESIGN-D or ESIGN-R as described above is proven secure in the multi key setting, because the random oracle H replaces a function that is common to all public keys.
The solution (also found in KCDSA for example) is to have a different hash function for each public key, for example by prepending the public key at the beginning of the input of H.

4 Dual mode of use of the public key

4.1 The idea

The core remark is that the public key n for ESIGN is indistiguishible of a RSA public key, and that the signature verification algorithms are very similar. It might be useful in a number of settings to disseminate a unique public key that could be used for both ESIGN and RSA-PSS.
This modification adds one bit to the signature, which decides which verification procedure has to be used. The public key is $n = p^2q$ and the public exponent is odd. $e = 9$ or 65537 are reasonable choices. Recommended parameter values are such that $8|(3l+1)$, for example $l = 373$. Another security parameter $k_r = 32$ is needed for RSA-PSS.

4.2 Dual signature scheme with appendix

A digest $\kappa = \mathsf{H}(n, e)$ is computed with a collision-intractable hash function. Three other hash functions are used and are modeled as distinct random oracles. H_0 and H_1 output $l-1$ bits and H_2 outputs $2l-1$ bits.

Verification procedure for a signed message σ
- σ is split in $m\|s\|b$ with one bit for b and $3l$ bits s.
- $0\|h\|0\|a \leftarrow s^e \bmod n$
- if $b = 0$ then $r\|0...0 \leftarrow a \oplus \mathsf{H}_2(h)$ else $r \leftarrow \epsilon$ (empty string)
- The signature is valid if $h \stackrel{?}{=} \mathsf{H}_b(\kappa\|m\|r)$

RSA-PSS based signature algorithm
- get a random k_r bits value r
- $h \leftarrow \mathsf{H}_0(\kappa\|m\|r)$
- $a \leftarrow (r\|0...0) \oplus \mathsf{H}_2(h)$, $\quad s \leftarrow (0\|h\|0\|a)^{1/e}$, $\quad \sigma \leftarrow m\|s\|1$.

ESIGN-D based signature algorithm
- $h \leftarrow \mathsf{H}_1(\kappa\|m)$
- get the smallest i such that $v \cdot pq - u < 2^{2l-1}$, with $r \leftarrow \phi(h\|\Delta\|i)$, $u \leftarrow h \cdot 2^{2l} - r^e \bmod n$ and $v \leftarrow \lceil \frac{u}{pq} \rceil$.
- $t \leftarrow v/(e \cdot r^{e-1}) \bmod p$, $\quad s \leftarrow r + t \cdot pq \bmod n$, $\quad \sigma \leftarrow m\|s\|0$.

4.3 Security proof

The same key can be used in two contexts. For documents where long term security is more important, only RSA-PSS signature can be taken as valid. For more common documents, ESIGN signature will also be accepted, and the signer will have the possibility to sign faster.

The natural question is whether this dual mode introduces a security flaw.

To prove the security of this dual mode of use of a public key, we need a new proof. In the random oracle, we will assume the existence of a $(t, \varepsilon, q_S, q_H)$-forger, where $q_S = q_{SE} + q_{SR}$ and $q_H = q_{H_0} + q_{H_1} + q_{H_2})$. The forger is allowed to make q_{SE} signature queries to the ESIGN-D signature oracle, q_{SR} signature queries to the RSA-PSS signature oracle, q_{H_0} hash queries for H_0, q_{H_1} hash queries for H_1, q_{H_2} hash queries for H_2 and outputs a valid forgery with probability ε after a running time t.
The reduction algorithm for the dual mode of use has to find a solution for the AER problem with target η or a solution for the RSA problem with target $\eta' = \eta \cdot 2^{2l} + w$. It runs in parallel the reduction algorithm from the security proof for ESIGN-D with target η and using a $(t, \varepsilon, q_{SE}, q_{H_1})$-forger and the reduction algorithm from the security proof for RSA-PSS with target η' and using a $(t, \varepsilon, q_{SR}, q_{H_0}, q_{H_2})$-forger.
If the forgery is a ESIGN-D signature, then the reduction can solve the AER problem with probability $\frac{1}{q_{SE}+q_{H_1}}$. If the forgery is a RSA-PSS signature, then the reduction can solve the RSA problem with probability close to 1.

We can conclude that using the same key for ESIGN-D and RSA-PSS does not weaken the security of these schemes.

References

1. 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 at `http://www-cse.ucsd.edu/users/mihir/crypto-research-papers.html`.
2. 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`.
3. Y. Dodis and L. Reyzin. On the Power of Claw-Free Permutations. *Proc. SCN'02* (this book), 2002.
4. E. Fujisaki, T. Kobayashi, H. Morita, H. Oguro, T. Okamoto, S. Okazaki. E-SIGN: Efficient Digital Signature (Submission to NESSIE) Available at `http://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions/esign.zip`.
5. J. Stern, D. Pointcheval, J. Malone-Lee and N.P. Smart. Flaws in Applying Proof Methodologies to Signature Schemes. *Proc. Crypto'02*, LNCS 2442, Aug. 2002.