How to repair ESIGN

Louis Granboulan

École Normale Supérieure

Louis.Granboulan@ens.fr

Abstract

Motivation: a flaw in the security proof of ESIGN was found in *Flaws in Applying Proof Methodologies to Signature Schemes*, Stern, Pointcheval, Malone-Lee and Smart, presented at Crypto'02

Two provable variants of ESIGN are proposed:
ESIGN-D and ESIGN-R

Some other modifications are proposed.

Part of this work has been supported by the Commission of the European Communities through the IST Programme under Contract IST-1999-12324 (NESSIE). `http://www.cryptonessie.org/`

1. Description of ESIGN

   From NESSIE submission by NTT Corp. (Fujisaki and al.)

2. The proof of security and its flaw

   Flaw found by Stern and al.

3. ESIGN-D and ESIGN-R

   Our proposals, with a proof of security

4. Other improvements of ESIGN

   Security in the multi key setting & dual use of the public key

Description of ESIGN

Public key: $n = p^2q$ Parameters: $e \geq 8$ and hash H

Intractability assumption: approximate e-th root modulo n

Best known attack (for $e \geq 4$): factorize n

length conditions for security parameter k:
$\log_2 p = \log_2 q = \ell$ and $\log_2 n = 3\ell$ and H outputs $\ell - 1$ bits
where ℓ is such that factoring n needs 2^k operations.

Verification algorithm

The signed message $m \| s$ is valid if $\mathsf{H}(m) \stackrel{?}{=} \lfloor \frac{s^e \bmod n}{2^{2l}} \rfloor$.

Signature algorithm of ESIGN

Basic signature algorithm

Generate uniform random $w < 2^{2l-1}$ and compute

$$s = (\mathsf{H}(m) \cdot 2^{2l} + w)^{1/e} \bmod n$$

Fast signature algorithm

Generate uniform random $r < pq$ and compute

$u = \mathsf{H}(m) \cdot 2^{2l} - r^e \bmod n$, $v = \lceil \frac{u}{pq} \rceil$ and $w = v \cdot pq - u$ until $w < 2^{2l-1}$

Compute $t = \frac{v}{e \cdot r^{e-1}} \bmod p$ and $s = r + t \cdot pq$

Proof of security for ESIGN

Simulation in the random oracle model

Proof similar to Full Domain Hash for RSA

Queries are answered with values corresponding to chosen valid appendices, with the exception of one hash query which is answered with the AER challenge.

Valid appendices are random s such that $s^e = h \cdot 2^{2l} + w$ with $h < 2^{l-1}$ and $w < 2^{2l-1}$

Signature requests are answered with s, Hash requests are answered with h

The AER challenge is solved with probability $\frac{1}{q_S + q_H}$

Flaw in the proof of security

Simulation is not perfect

Multiple signature queries for the same message are given the same answer, this does not mimic ESIGN signature algorithm.

This is not CMA (existentially unforgeable against chosen message attacks) but SO-CMA (existentially unforgeable against single occurence chosen message attacks).

ESIGN-D: making ESIGN deterministic

Additional components

A one-way function ϕ with uniform output in $[0, pq)$
An additional k bit value Δ in the secret key

The modification

Instead of generating uniform random $r < pq$, the signature generation uses the values $\phi(\mathsf{H}(m)\|\Delta\|i)$ for $i = 0, 1, 2, ...$

ESIGN-R : making the randomised signature simulable

Additional component

An additional random seed ρ with at least $2\log_2 q_S$ bits

The modification

$\mathsf{H}(m\|\rho)$ is used instead of $\mathsf{H}(m)$, and ρ is included in the signature appendix.

The simulator in the signature proof generates random ρ to answer signature queries, and collision with hash queries is unlikely.

Comparison of ESIGN-D and ESIGN-R

Both have a loose reduction in their security proof (factor of $\frac{1}{q_S+q_H}$)

Both can do message recovery of $\ell - 1 - 2k$ bits

ESIGN-D costs the computation of ϕ

ESIGN-R needs a source of randomness and increases slightly the signature length and the input of H.

We prefer ESIGN-D, which is also supported by NTT

The multi-key setting

Proofs of security usually only consider the case where a unique public key is used.

The multi-key setting allows the attacker to have access to signature oracles for different keys.

The best security is obtained when the signature oracles give independent answers.

This is also related to security against parameter manipulation.

Security in the multi-key setting

The components of ESIGN are the hash function H and the function $s \mapsto \lfloor \frac{s^e \bmod n}{2^{2l}} \rfloor$.

ESIGN is secure in the multi-key setting if the following two conditions hold:

- H is different for each public key
  This is the case if the public key is added at the beginning of the hash input
- A parellel attack of many instances of the AER problem cannot be faster than independent attacks of these instances.
  This hypothesis is likely to hold if AER is hard

Dual mode of use of the public key

A RSA public key is indistinguishable from an ESIGN public key.

If both use $n = p^2 q$ and the same e, then the simultaneous use of ESIGN and RSA-PSS is secure.

The same public key can be used for fast signing (ESIGN) or high security (RSA-PSS).

Conclusion

ESIGN can easily be repaired

ESIGN-D should replace ESIGN-TSH

One should include the public key in the input of hashes

To simplify the PKI, one can use the same $n = p^2q$ for both RSA and ESIGN