Cryptography

Digital Signatures

Prof. Dr. Heiko Knospe

TH Köln – University of Applied Sciences

June 25, 2022

Signatures and their Objectives

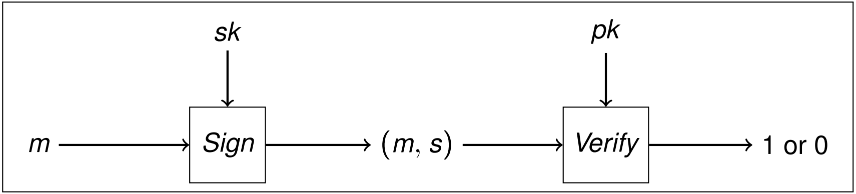
Digital *signatures* are asymmetric cryptographic schemes which aim at data *integrity and authenticity*. This is very similar to message authentication codes. However, digital signatures are verified using a *public key*. The successful verification of a signature shows that the data is authentic and has not been tampered with.

Since the private key is exclusively controlled by the signer, digital signatures can also achieve *non-repudiation*. This means that the signer cannot later deny his or her signature.

Signatures have applications beyond integrity protection, for example in entity authentication protocols, where a correct signature serves as proof of identity.

Signature Generation and Verification

Messages are signed using a *private key*. Verification requires the *public key* of the signer. As with public-key encryption, it is crucial that an adversary is not able to derive the private signature key from the public key, or otherwise forge a valid signature.



Signing uses the private key sk and verification the public key pk.

Signature Schemes

Definition

A *digital signature* scheme is given by:

- A message space $\mathcal{M}$,
- A space of key pairs $\mathcal{K} = \mathcal{K}_{pk} \times \mathcal{K}_{sk}$,
- A randomized key generation algorithm $Gen(1^n)$ that takes a security parameter 1^n as input and outputs a pair of keys (pk, sk),
- A signing algorithm, which may be randomized. It takes a message m and a private key sk as input and outputs a signature $s \leftarrow \text{sign}_{sk}(m)$,
- A deterministic verification algorithm that takes a public key pk, a message m and a signature s as input and outputs 1 if the signature is valid, and 0 otherwise.

Security Definition

Adversary		*Challenger/Oracle*
	$\xleftarrow{1^n,\ pk}$	$(pk, sk) \xleftarrow{\$} Gen(1^n)$
Choose m	$\xrightarrow{m}$	
	$\xleftarrow{s}$	$s = \text{sign}_{sk}(m)$
	$\vdots$	
Choose $m' \neq m$, forge a signature s'	$\xrightarrow{(m', s')}$	Verify (m', s'), output 1 or 0

Signature forgery experiment.

Secure Signature Schemes

Secure signatures should be *unforgeable*:

Definition

A signature scheme is called *existentially unforgeable under an adaptive chosen message attack* (*EUF-CMA secure* or just *secure*), if for all probabilistic polynomial-time adversaries, the probability of successfully forging a signature is negligible in n.

The verification of a digital signature requires the *authentic public key* of the signer. Although public keys can be openly shared, their authenticity is not self-evident. A *man-in-the-middle* might replace the message, the signature and the public key with his own data.

Plain RSA Signature

Definition

The RSA signature scheme uses the same parameters as RSA encryption.

- A key generation algorithm $Gen(1^n)$ generates p, q, $N = pq$, e, d and outputs the public key $pk = (e, N)$ as well as the private key $sk = (d, N)$.
- The message space is $\mathcal{M} = \mathbb{Z}_N^*$.
- The deterministic signature algorithm takes sk and a message $m \in \mathcal{M}$ as input and outputs the signature

$$s = \mathrm{sign}_{sk}(m) = m^d \mod N.$$

Plain RSA Signature

Definition

- The verification algorithm takes pk, a message $m \in \mathbb{Z}_N^*$ and a signature s. It computes

$$s^e \mod N,$$

and outputs 1 (valid) if $m = s^e \mod N$, and 0 otherwise.

Unfortunately, this scheme is both impractical and insecure. Firstly, the message length is limited by the size of the RSA modulus N. However, we want to sign messages of *arbitrary length*.

Secondly, the plain RSA signature scheme is insecure, because signatures can be easily forged: choose s and set $m = s^e \mod N$. Furthermore, the plain RSA signature is *multiplicative*.

RSA-FDH

The RSA-FDH (*Full Domain Hash*) signature is similar to the plain RSA scheme, but leverages a hash function $H : \{0,1\}^* \to \mathbb{Z}_N^*$.
A message m is first *hashed* and then *signed*:

$$s = \mathrm{sign}_{sk}(m) = H(m)^d \mod N$$

In the verification step, $H(m)$ is computed and then compared to $s^e \mod N$. A signature is valid if $H(m) = s^e \mod N$.

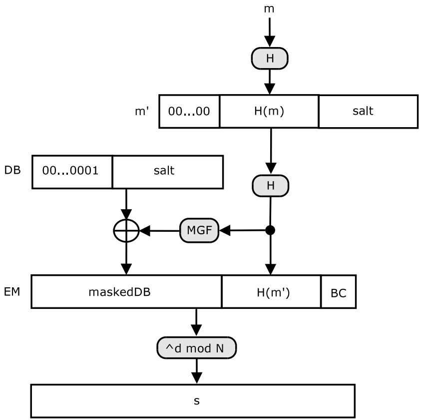
Obviously, collision-resistance of H is crucial since collisions can produce unintended additional signatures.

Theorem

If H has range $\mathbb{Z}_N^*$ and is modeled as a random oracle, then the RSA-FDH scheme is EUF-CMA secure under the RSA assumption.

RSA-PSS

The padded and randomized *Probabilistic Signature Scheme* (RSA-PSS) is standardized in PKCS #1 version 2.2 and in RFC 8017.



Signing a message m using RSA-PSS.

Security of RSA-PSS

The length of cryptographic hashes is usually smaller than the size of the RSA modulus, and RSA-PSS stretches the hash by randomized padding. If the salt is randomly chosen and sufficiently long, then the RSA-PSS signature is randomized, and signing the same message twice using the same key gives different signature values.

The RSA-PSS construction makes it very hard to forge a valid signature:

Theorem

The RSA-PSS signature scheme is EUF-CMA secure in the random oracle model under the RSA assumption.

Other Signature Schemes

An alternative to RSA are signature schemes that are based on the *discrete logarithm problem* in a cyclic group, similar to the Diffie-Hellman key exchange:

- ElGamal signature scheme,
- DSA/DSS (Digital Signature Algorithm),
- ECDSA (Elliptic Curve Digital Signature Algorithm).

Furthermore, there are *hash-based signatures schemes*, e.g.:

- Lamport signature scheme,
- Extended Merkle Signature Scheme (XMSS, RFC 8391),
- SPHINCS+, a candidate for *post-quantum cryptography*.