Cryptography

Digital Signatures

Prof. Dr. Heiko Knospe

TH Köln – University of Applied Sciences

April 23, 2025

Signatures and their Objectives

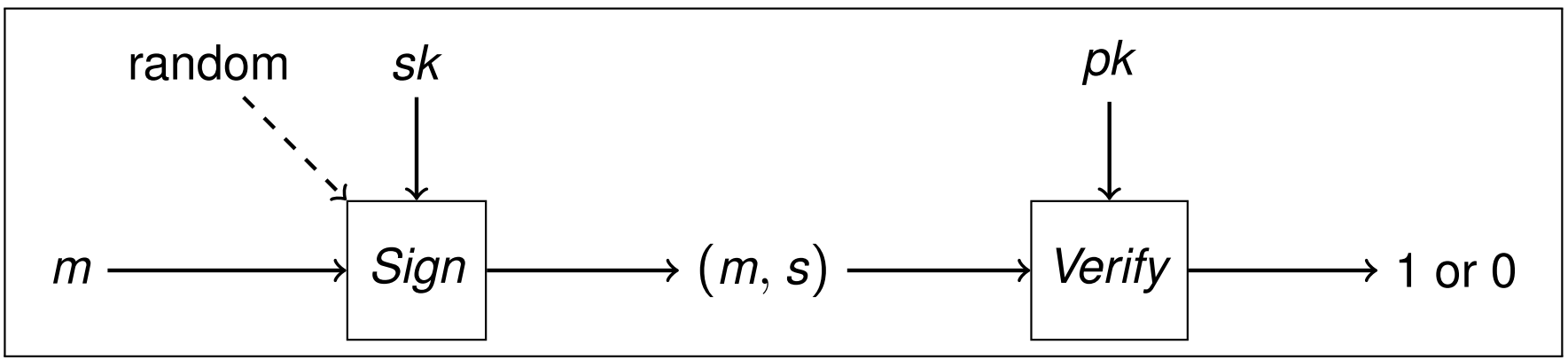
Digital *signatures* are asymmetric cryptographic schemes aiming at data *integrity and authenticity*. Hence the objective is similar to *message authentication codes*. However, digital signatures are verified using a *public key*. The successful verification of a signature shows that the data is authentic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

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

Signatures have applications beyond data integrity protection, for example in entity *authentication protocols*, where a correct signature serves as a *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 forge a valid signature.



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

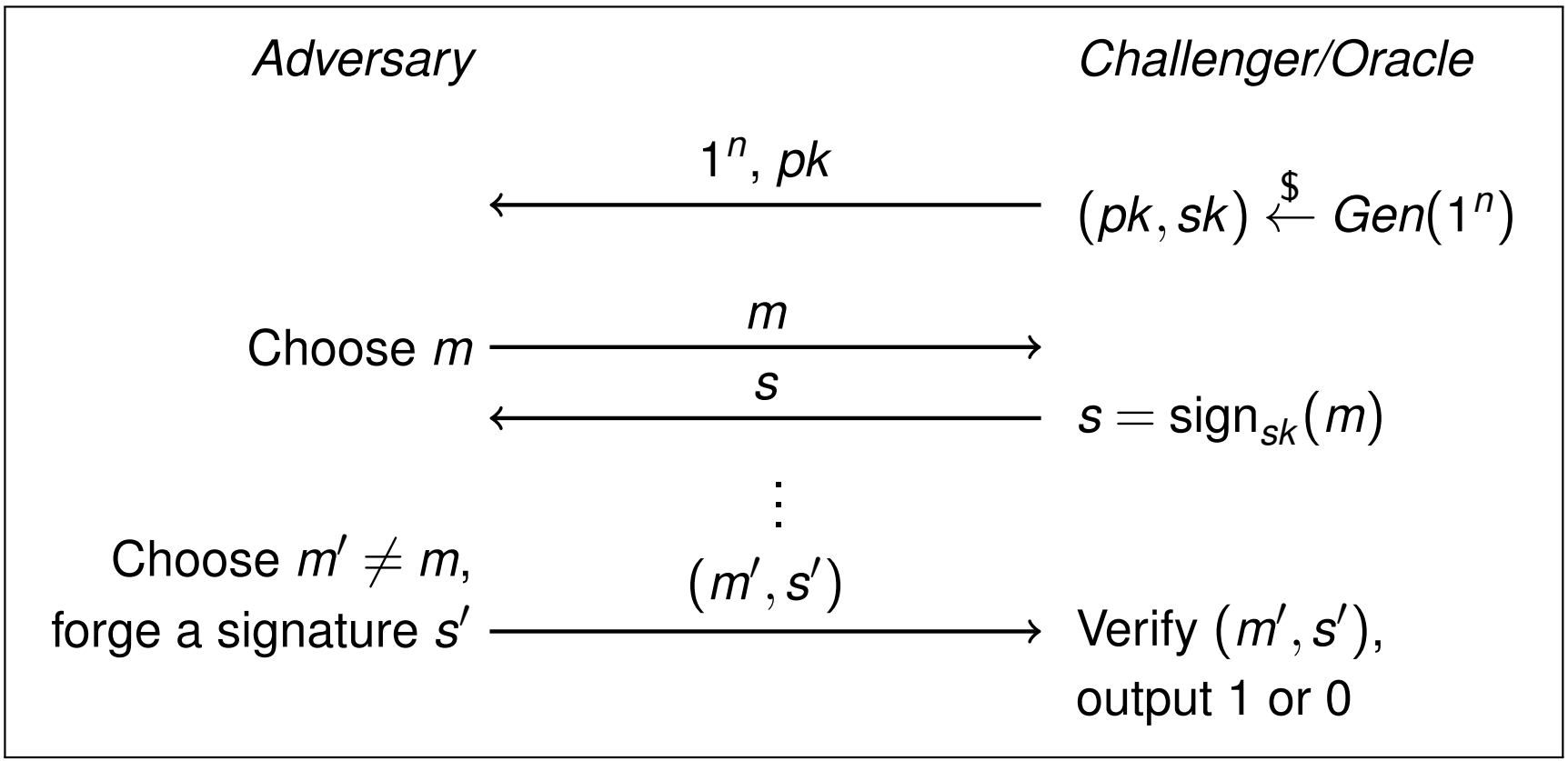
Signature Schemes

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



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

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

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, one wants to sign messages of *arbitrary length*.

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

Also, the plain RSA signature is *multiplicative* which can be exploited.

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*:

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

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

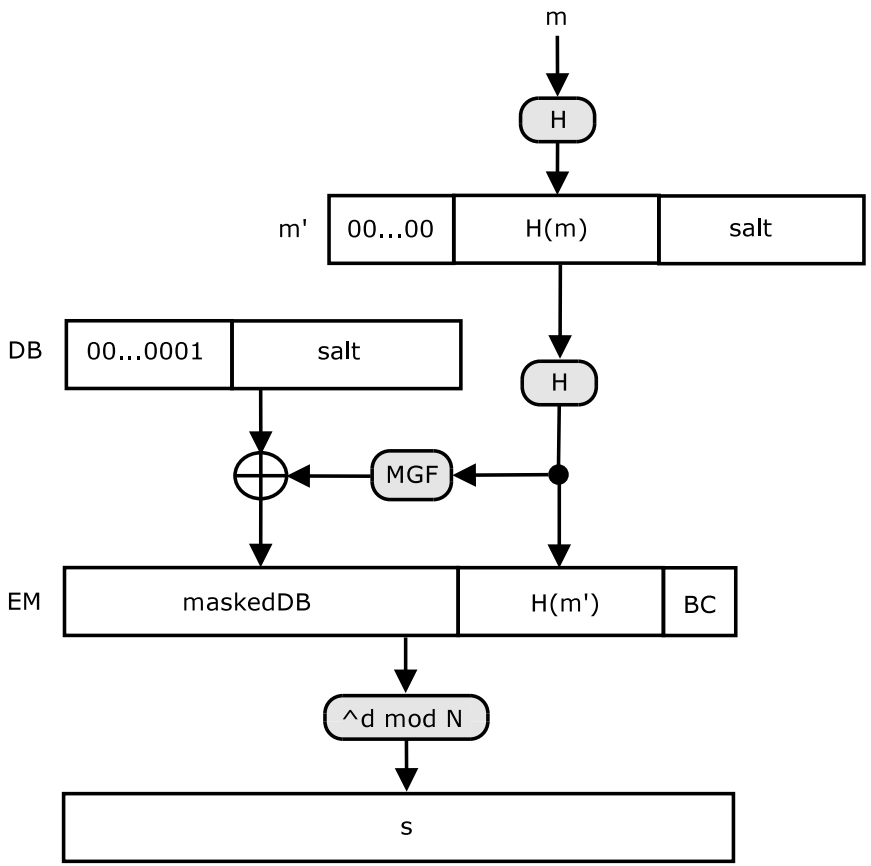
Obviously, collision-resistance of H is crucial, since collisions would produce unintended additional signatures. Furthermore, the one-wayness of H is important to prevent the forgery of signatures.

Theorem

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

PSS

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



Signing a message m with RSASSA-PSS.

Security of PSS

The length of cryptographic hashes is usually smaller than the size of the RSA modulus. Now PSS encoding for signatures stretches the hash by randomized padding. When the salt is randomly chosen and sufficiently long, the PSS signature is randomized, and signing a message twice with the same key gives different signatures.

It is hard to forge a valid signature:

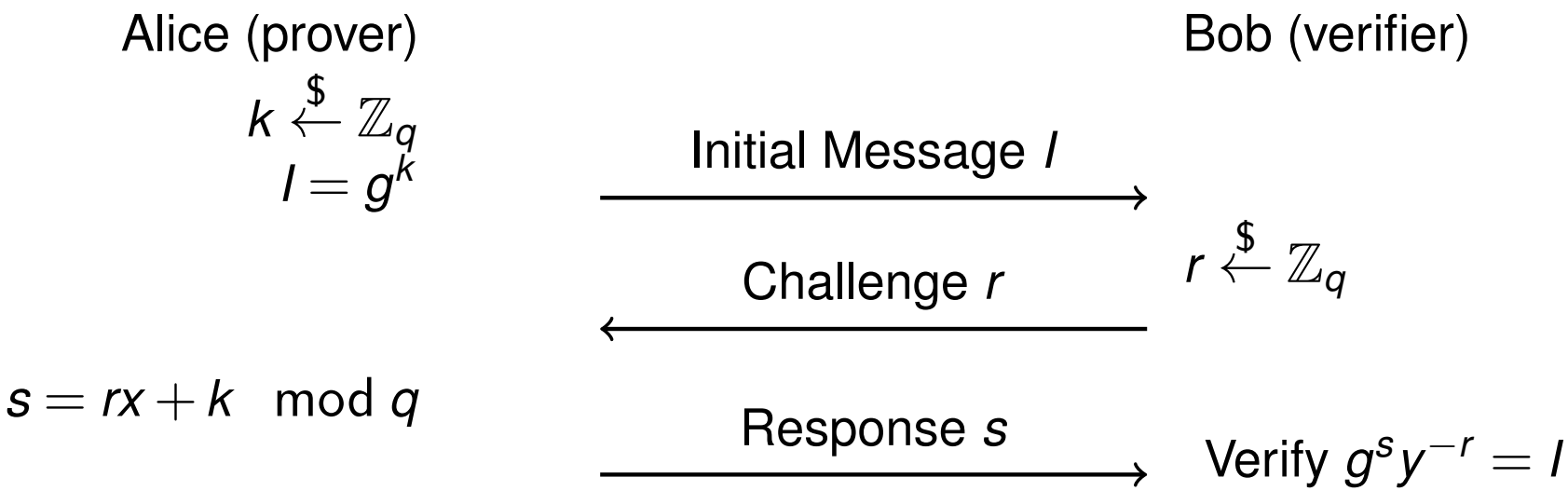
Theorem

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

It is recommended that the RSA modulus should have ≥ 3000 bits, and at least SHA-256 (in particular not SHA-1) should be used as hash function.

Schnorr Identification Scheme

The *Schnorr identification scheme* is based on the discrete logarithm problem. Let G be a cyclic group of order q and let g be a generator of G. The prover (Alice) chooses a uniform private key $x \in \mathbb{Z}_q$, and sets $y = g^x \in G$. The public key is (G, q, g, y). The Schnorr scheme is a *zero knowledge proof* of Alice's private key x and authenticates Alice to Bob.



Security of the Schnorr Scheme

It is easy to show that the Schnorr Identification Scheme is correct:

$$g^s y^{-r} = g^{rx+k} g^{-rx} = g^k = I$$

One can show that the Schnorr scheme is *secure against passive attacks* if the discrete logarithm problem is hard relative to G. In the corresponding experiment, the adversary can eavesdrop on multiple executions of the protocol. He outputs an initial message I and obtains a uniform challenge r. Finally, the adversary (prover) outputs a response s. He wins the experiment if the verifier accepts the response. A scheme is secure if *for all probabilistic polynomial-time adversaries, the probability of winning the experiment is negligible*.

Fiat-Shamir Transform

The *Fiat-Shamir* transform converts an interactive identification scheme into a signature scheme. The *Schnorr signature scheme* is defined as the Fiat-Shamir transform of the Schnorr identification scheme.

The signer acts as prover and generates the initial message I. He or she also generates the challenge r by hashing I and the message m. Then the signer computes the response s. The challenge r and the response s form the signature of m.

The verifier recomputes I and checks that r is correct for the given message m by hashing I and m.

Schnorr Signature Scheme

Definition

- The key generation algorithm $Gen(1^n)$ generates G, q and g. Choose a uniform $x \in \mathbb{Z}_q$ and set $y = g^x$. The private key is x and the public key is (G, q, g, y). Furthermore, a hash function $H : \{0,1\}^* \to \mathbb{Z}_q$ is chosen.
- The randomized signature algorithm takes a message m and the private key x as input. Choose a uniform $k \in \mathbb{Z}_q$ and set $I = g^k$. Then compute $r = H(I, m)$ and $s = rx + k \mod q$. Output the signature (r, s).
- The verification algorithm takes the public key (G, q, g, y), a message m and a signature (r, s), computes $I = g^s y^{-r}$ and outputs 1 (valid) if $H(I, m) = r$.

Security of the Schnorr Signature Scheme

First, we consider the security of the identification scheme.

Theorem

If the discrete logarithm is hard relative to the group parameters then the Schnorr Identification Scheme is secure (against passive attacks).

An active attacker might act as a Man-in-the-Middle and successfully authenticate with his own message and response values. However, security against passive attacks is sufficient for a secure signature scheme.

Theorem

If the discrete logarithm is hard relative to the group parameters and the hash function H is modeled as a random oracle, then the Schnorr Signature Scheme is secure.

DSA and ECDSA Signatures

The *Digital Signature Algorithm* (DSA) and the *Elliptic Curve Digital Signature Algorithm* (ECDSA) were standardized (by the American NIST) as part of the Digital Signature Standard (DSS). They are based on the discrete logarithm problem in a multiplicative group of a finite field (DSA) or in a group of points on an elliptic curve over a finite field (ECDSA). DSA and ECDSA are variants of the Schnorr signature scheme.

DSA and ECDSA are often used as alternatives to RSA-PSS. It is now recommended to use ECDSA for new signatures.

Definition of DSA and ECDSA

Definition

- The key generation algorithm $Gen(1^n)$ generates a group G of prime order q and a generator g. Choose a uniform $x \in \mathbb{Z}_q$ and set $y = g^x$. The private key is x and the public key is (G, q, g, y). Fix a function (mod q) which maps G to $\mathbb{Z}_q$. Furthermore, a hash function $H : \{0,1\}^* \to \mathbb{Z}_q$ is chosen.
- The randomized signature algorithm takes a message m and the private key x as input. Chooses a uniform $k \in \mathbb{Z}_q$ and set $r = g^k \mod p \mod q$. Then compute $s = k^{-1}(H(m) + xr) \mod q$. If $r = 0$ or $s = 0$ then start again with a new choice of k. Output the signature (r, s).
- The verification algorithm takes the public key (G, q, g, y), a message m and a signature (r, s), where $r \neq 0 \mod q$ and $s \neq 0 \mod q$. The algorithm outputs 1 (valid) if

$$r = \left(g^{H(m)s^{-1}} y^{rs^{-1}} \mod p\right) \mod q.$$

Correctness of DSA and ECDSA

If the signature (r, s) is correctly computed then $k = s^{-1}(H(m) + xr) \mod q$ and

$$r = g^k = g^{s^{-1}H(m)} g^{s^{-1}xr} = g^{H(m)s^{-1}} y^{rs^{-1}} \mod p \mod q.$$

Hence the verification is correct.

It is important that $k \in \mathbb{Z}_q$ is chosen uniformly at random for each signature. Furthermore, the signer must keep k secret, since otherwise the private key x can be easily computed.

Key Lengths and Efficiency of DSA and ECDSA

DSA uses a subgroup G of the multiplicative group $GF(p)^*$, for which sub-exponential attacks against the discrete logarithm exist. It is therefore recommended to use a prime p with more than 3000 bits. The subgroup G of order q should have at least 250 bits. Note that signature generation works modulo q, which makes DSA more efficient than RSA, at least for signature generation.

ECDSA uses a group G of points on an elliptic curve of order q over $GF(p)$ (see next chapter). The prime numbers p and q should have at least 250 bits. Hence ECDSA has shorter keys than RSA and DSA.

Other Signature Schemes

- Hash-based signatures schemes:
  - Lamport signature scheme,
  - Extended Merkle Signature Scheme (XMSS, RFC 8391),
  - SPHINCS+, a *post-quantum* signature scheme based on XMSS, standardized in FIPS 205.
- Lattice-based signature schemes:
  - CRYSTALS Dilithium, a *post-quantum* scheme based on hard problems in module *lattices*, standardized in FIPS 204,
  - FALCON, a *post-quantum* scheme based on hard problems in NTRU *lattices*.