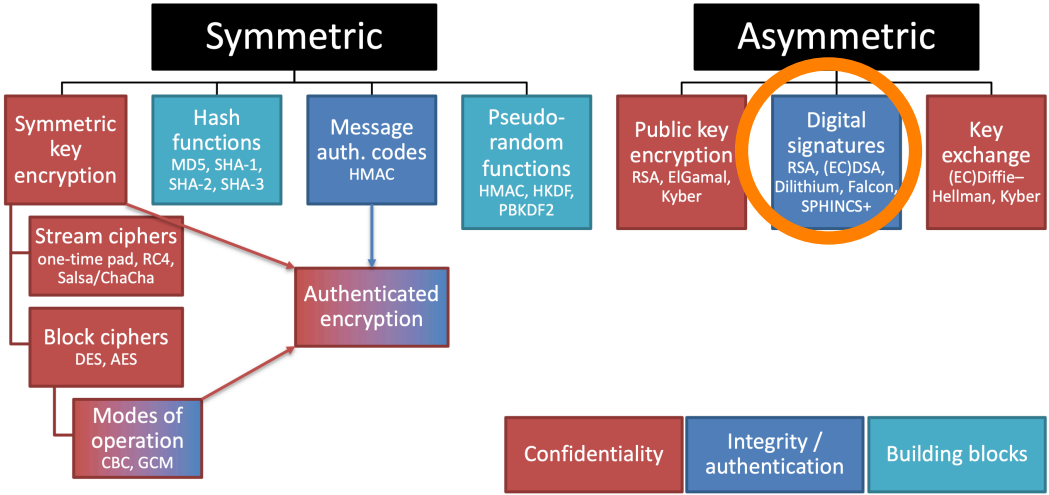
Topic 3.8
Public key cryptography – Digital signatures

Douglas Stebila

CO 487/687: Applied Cryptography
Fall 2024

Map of cryptographic primitives



Motivation

Recall fundamental goals of cryptography:

- Confidentiality: Keeping data secret from all but those authorized to see it.
- Data integrity: Ensuring data has not been altered by unauthorized means.
- Data origin authentication: Corroborating the source of data.
- Non-repudiation: Preventing an entity from denying previous commitments or actions.

Want a public key primitive that achieves data integrity, data origin authentication, and non-repudiation.

Outline

RSA signatures

Defining signature schemes

RSA signatures, continued

Diffie–Hellman-based signatures

Post-quantum digital signatures

RSA Signature Scheme

Ron Rivest, Adi Shamir, and Leonard Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM **21** (2): pp. 120–126, 1978.

Key generation: Same as in RSA encryption. $pk = (n, e)$, $sk = (n, d)$

Signature generation: To sign a message m:

1. Compute $s = m^d \bmod n$.
2. The signature on m is s.

Signature verification: To verify a signature s on a message m:

1. Obtain an authentic copy of the public key (n, e).
2. Compute $s^e \bmod n$
3. Accept (m, s) if and only if $s^e \bmod n = m$.

A weird (and misleading) way to think about digital signatures

- In the definition of a public-key cryptosystem, decryption must be a left inverse of encryption:
$$\mathcal{D}(k_{\text{privkey}}, \mathcal{E}(k_{\text{pubkey}}, m)) = m.$$
- There is no corresponding requirement that decryption be a **right** inverse of encryption:
$$\mathcal{E}(k_{\text{pubkey}}, \mathcal{D}(k_{\text{privkey}}, c)) \stackrel{?}{=} c.$$
- In some cases (e.g. ElGamal), decryption is not a right inverse of encryption.
- In some cases (e.g. plain RSA without padding), decryption is in fact a right inverse of encryption.
  - RSA encryption: $m^e \bmod n$; RSA decryption $c^d \bmod n$
  - RSA signing: $m^d \bmod n$; RSA verification $s^e \bmod n$

Outline

Definition of digital signatures

Definition (Digital signature scheme)

A *digital signature scheme* consists of:

- M – the plaintext space,
- S – the signature space,
- K_{pubkey} – the space of public keys,
- K_{privkey} – the space of private keys,
- A *randomized* key generation algorithm $\mathcal{G} \to K_{\text{pubkey}} \times K_{\text{privkey}}$,
- A (usually probabilistic) signing algorithm $\mathcal{S} : K_{\text{privkey}} \times M \to S$,
- A verification algorithm $\mathcal{V} : K_{\text{pubkey}} \times M \times S \to \{\textbf{true}, \textbf{false}\}$.

Definition of digital signatures

A valid signature is one which verifies. An invalid signature is one which does not verify.

Correctness requirement: For a given key pair $(k_{\text{pubkey}}, k_{\text{privkey}})$ produced by $\mathcal{G}$,

$$\mathcal{V}(k_{\text{pubkey}}, m, \mathcal{S}(k_{\text{privkey}}, m)) = \textbf{true}$$

for all $m \in M$.

Basic security requirements

Goals of a digital signature scheme, from the designer's perspective:

- *Authenticate* the origin of a message.
- Guarantee the *integrity* of a message.
- Basic security requirements:
  - It should be infeasible to deduce the private key from the public key.
  - It should be infeasible to generate valid signatures without the private key.

Goals of the Adversary

1. Total break: Recover the private key.
2. Selective forgery: Given a message or a subset of messages, forge a signature for those messages.
3. Existential forgery: Forge a signature for some message (possibly out of your control).

Attack model

Types of interactions allowed:

1. Key-only attack: The public key is known.
2. Known-message attack: Some messages and their valid signatures are known.
3. Chosen-message attack: The adversary may choose some messages and obtain their signatures.

Security Definition

Definition

A signature scheme is said to be secure if it is existentially unforgeable by a computationally bounded adversary who launches a chosen-message attack (EUF-CMA).

In this definition:

- The adversary is given a public key.
- The adversary has access to a signing oracle, which given a message produces a valid signature for that message under the above public key.
- The goal is to compute a valid signature for some message m which has not been provided as input to the signing oracle.

Outline

RSA signatures

Defining signature schemes

RSA signatures, continued

Diffie–Hellman-based signatures

Post-quantum digital signatures

Security of the Basic RSA Signature Scheme

Recall the statement of the RSA problem: Given an RSA public key (n, e) and an element $c \in \mathbb{Z}_n$ such that $\gcd(c, n) = 1$, find an element $m \in \mathbb{Z}_n$ such that

$$c = m^e \bmod n.$$

Theorem

A necessary condition for RSA signatures to be secure is that the RSA problem must be intractable.

Proof.

If the RSA problem is easy, one can forge signatures as follows:

1. Let m be any message.
2. Find s such that $s^e \equiv m \pmod{n}$.
3. Then s is a valid signature for m. □

Security of the Basic RSA Signature Scheme

Theorem

A necessary condition for RSA signatures to be secure is that the RSA problem must be intractable.

Is this a sufficient condition? If the RSA problem is hard, is the RSA signature scheme secure?

Insecurity of Basic RSA Signature Scheme

Even if the RSA problem is intractable, the basic RSA scheme is still insecure.

Here is an existential forgery under a key-only attack:

1. Select $s \in \mathbb{Z}_n$ with $\gcd(s, n) = 1$.
2. Compute $s^e \bmod n$.
3. Set $m = s^e \bmod n$.
4. Then s is a valid signature for m.

Insecurity of Basic RSA Signature Scheme

Here is a selective forgery under a chosen message attack: Given $m \in \mathbb{Z}_n$ with $\gcd(m, n) = 1$:

1. Compute $m' = 2^e \cdot m \bmod n$
2. Request the signature s' of m'
3. Compute $s = s'/2 \bmod n$.
4. Then s is a valid signature for m.

This takes advantage of the malleability property of the basic RSA function:
given $c = m^e \bmod n$ for an unknown m, for any $x \in \mathbb{Z}_n^*$, we can construct c' encrypting mx by computing

$$c' = (x^e \cdot c) \bmod n = (xm)^e \bmod n$$

Full Domain Hash RSA (RSA-FDH)

Let $H\colon \{0,1\}^* \to \mathbb{Z}_n$ be a hash function.

Key generation: Same as in RSA. $pk = (n, e), sk = (n, d)$

Signature generation: To sign a message $m \in \{0,1\}^*$:

1. Compute $s = H(m)^d \bmod n$.
2. The signature on m is s.

Signature verification: To verify a signature s on a message m:

1. Obtain an authentic copy of the public key (n, e).
2. Compute $s^e \bmod n$
3. Accept (m, s) if and only if $s^e \bmod n = H(m)$.

Security of RSA-FDH

Theorem (Bellare & Rogaway, 1996): If the RSA problem is intractable and H is a random function, then RSA-FDH is a secure signature scheme.

Note:

- This theorem does NOT always hold if H is not a random function!
- Example of insecurity: PKCS #1 v1.5

Security Properties of the Hash Function in RSA-FDH

Necessary security properties of the hash function H:

Preimage resistance: If H is not preimage resistant, and the range of H is $[0, n-1]$, E can forge signatures as follows:

1. Select $s \in [0, n-1]$.
2. Compute $s^e \bmod n$.
3. Find m such that $H(m) = s^e \bmod n$.
4. Then s is A's signature on m.

Security Properties of the Hash Function in RSA-FDH

Necessary security properties of the hash function H:

2nd preimage resistance: If H is not 2nd preimage resistant, E could forge signatures as follows:

1. Suppose that (m, s) is a valid signed message.
2. Find an m', $m \neq m'$, such that $H(m) = H(m')$.
3. Then (m', s) is a valid signed message.

Security Properties of the Hash Function in RSA-FDH

Necessary security properties of the hash function H:

Collision resistance: If H is not collision resistant, E could forge signatures as follows:

1. Select m_1, m_2 such that $H(m_1) = H(m_2)$, where m_1 is an "innocent" message, and m_2 is a "harmful" message.
2. Induce A to sign m_1: $s = H(m_1)^d \bmod n$.
3. Then s is also A's signature on m_2.

Outline

Digital Signature Algorithm (DSA)

The Digital Signature Algorithm (NIST FIPS 186-3) is a digital signature scheme based on Diffie–Hellman/ElGamal.

- Setup:
  - A prime p, a prime q dividing $p-1$, an element $g \in \mathbb{Z}_p^*$ of order q, a hash function $H\colon \{0,1\}^* \to \mathbb{Z}_q$.
- Key generation:
  - Choose $\alpha \in_R \mathbb{Z}_q^*$ at random. Return $(k_{\text{pubkey}}, k_{\text{privkey}}) = (g^\alpha \bmod p, \alpha)$
- Signing: To sign a message $m \in \{0,1\}^*$,
  - Choose $k \in_R \mathbb{Z}_q^*$ at random
  - Calculate $r = (g^k \bmod p) \bmod q$ and $s = \frac{H(m)+\alpha r}{k} \bmod q$.
  - Repeat if k, r, or s are zero. Otherwise, return signature $\sigma = (r, s)$.
- Verification: Given $k_{\text{pubkey}} = g^\alpha$, m, and $\sigma = (r, s)$,
  - Check $0 < r < q$ and $0 < s < q$ and $(g^{\frac{H(m)}{s}} g^{\frac{\alpha r}{s}} \bmod p) \bmod q = r$.

Elliptic Curve Digital Signature Algorithm (ECDSA)

- Setup:
  - A prime p, a prime q, an elliptic curve E over $\mathbb{Z}_p$ of cardinality $|E| = q$, a generator $P \in E$ of order q, and a hash function $H\colon \{0,1\}^* \to \mathbb{Z}_q$.
- Key generation:
  - Choose $\alpha \in_R \mathbb{Z}_q^*$ at random.
  - $(k_{\text{pubkey}}, k_{\text{privkey}}) = (\alpha P, \alpha)$
- Signing: To sign a message $m \in \{0,1\}^*$,
  - Choose $k \in_R \mathbb{Z}_q^*$ at random
  - Calculate $r = x_{kP} \bmod q$, and $s = \frac{H(m)+\alpha r}{k} \bmod q$.
  - Repeat if k, r, or s are zero.
  - The signature is $\sigma = (r, s)$.
- Verification: Given αP, m, and (r, s),
  - Check $0 < r < q$ and $0 < s < q$.
  - Check that the x-coordinate of $\frac{H(m)}{s} P + \frac{r}{s}(\alpha P)$ is congruent to r modulo q.

Outline

RSA signatures

Defining signature schemes

RSA signatures, continued

Diffie–Hellman-based signatures

Post-quantum digital signatures

Post-quantum digital signatures

Dilithium a.k.a. ML-DSA

- Based on module learning with errors (MLWE) and a related problem (module short integer solutions)
- Dilithium2: 128-bit security, 1312 byte public keys, 2420 byte signatures
- Design pattern: Identification protocol proving knowledge of a short vector, combined with Fiat–Shamir transform (see Topic 4.5)
- Selected for standardization by NIST in 2022, draft standard 2023, standard est. 2024

Falcon

- Based on NTRU lattice problem
- Falcon512: 128-bit security, 897 byte public keys, 666 byte signatures
- Selected for standardization by NIST in 2022, draft standard est. 2024

Post-quantum digital signatures

SPHINCS+ a.k.a. SLH-DSA

- Based on hash functions
- SLH-DSA-SHA2-128s: 128-bit security, 32 byte public keys, 7856 byte signatures
- Selected for standardization by NIST in 2022, draft standard 2023, standard est. 2024

HSS/LMS and XMSS

- Based on hash functions
- Stateful signature schemes: must update secret key after signing, can only sign limited number of signatures
- XMSS-SHA2-20-256: 128-bit security, 2^{20} maximum signatures, 32-byte public keys, 2820-byte signatures
- Standardized by IRTF and approved by NIST

4+8+7 things to remember from CO 487

Public key primitives

Digital signatures

KeyGen() → (pk, sk)
Sign(sk, m) → sig
Verify(pk, m, sig) → T/F

- **Provides integrity.**
- Security goal: given public key, existential unforgeability under chosen message attack.
- Secure options as of 2024:
  - Not post-quantum:
    - RSA-3072 Full Domain Hash but not basic RSA, RSA-PSS
    - DSA, ECDSA, Ed25519
  - Post-quantum:
    - ML-DSA (Dilithium), Falcon, SLH-DSA(SPHINCS+)

Summary of digital signatures

- Public key primitive giving integrity, data origin authentication, and non-repudiation
- Security goal: existential unforgeability under chosen message attacks
- Basic RSA signatures insecure
- RSA-Full Domain Hash secure if factoring and RSA problems hard
- DSA secure if discrete logarithms and Diffie–Hellman problems hard
- ECDSA secure if elliptic curve discrete logarithms and elliptic curve Diffie–Hellman problems hard
- Can use same key sizes as RSA public key encryption, ElGamal public key encryption, and elliptic curve Diffie–Hellman respectively
- RSA, DSA, ECDSA vulnerable to attack by quantum computers
- Post-quantum digital signatures being standardized