Schoolbook RSA

Tanja Lange

Eindhoven University of Technology

2WF80: Introduction to Cryptology

Public-key cryptology – formal treatment

Public-key encryption requires 3 algorithms:

1. Key generation, generating a public-key private-key pair.
2. Encryption, taking a public key and a message, producing ciphertext.
3. Decryption, taking a private key and a ciphertext, producing plaintext.

Signatures also require 3 algorithms:

1. Key generation, generating a public-key private-key pair.
2. Signing, taking a private key and a message, producing a signature.
3. Verification, taking a public key and a signed message, producing valid or not.

Reminder: signatures and MACs both ensure authenticity and integrity.

But a signature can be verified by *anybody* using a public key
while MACs require *the same shared secret key.*

Signatures belong to public-key cryptography;
MACs belong to symmetric-key cryptography.

(Re-)Watch video on Public-key and symmetric-key cryptology.

Schoolbook RSA encryption

1977 Rivest, Shamir, Adleman. Do not use Schoolbook RSA in practice!

Schoolbook RSA encryption

1977 Rivest, Shamir, Adleman. Do not use Schoolbook RSA in practice!

KeyGen:

1. Pick primes $p, q; p \neq q$.
2. Compute $n = p \cdot q$, $\varphi(n) = (p-1)(q-1)$.
3. Pick $1 < e < n$ with $\gcd(e, \varphi(n)) = 1$.
4. Compute $d \equiv e^{-1} \bmod \varphi(n)$. (Re-)Watch video on XGCD.
5. Output public key (n, e), private key (n, d).

Schoolbook RSA encryption

1977 Rivest, Shamir, Adleman. Do not use Schoolbook RSA in practice!

KeyGen:

1. Pick primes $p, q; p \neq q$.
2. Compute $n = p \cdot q$, $\varphi(n) = (p-1)(q-1)$.
3. Pick $1 < e < n$ with $\gcd(e, \varphi(n)) = 1$.
4. Compute $d \equiv e^{-1} \bmod \varphi(n)$. (Re-)Watch video on XGCD.
5. Output public key (n, e), private key (n, d).

Enc message $0 \leq m < n$:

1. Compute $c \equiv m^e \bmod n$. Watch video on Exponentiation.
2. Output c.

Dec ciphertext $0 \leq c < n$:

1. Compute $m' \equiv c^d \bmod n$.
2. Output m'.

Schoolbook RSA encryption

1977 Rivest, Shamir, Adleman. Do not use Schoolbook RSA in practice!

KeyGen:

1. Pick primes $p, q; p \neq q$.
2. Compute $n = p \cdot q$, $\varphi(n) = (p-1)(q-1)$.
3. Pick $1 < e < n$ with $\gcd(e, \varphi(n)) = 1$.
4. Compute $d \equiv e^{-1} \bmod \varphi(n)$. (Re-)Watch video on XGCD.
5. Output public key (n, e), private key (n, d).

Enc message $0 \leq m < n$:

1. Compute $c \equiv m^e \bmod n$. Watch video on Exponentiation.
2. Output c.

Dec ciphertext $0 \leq c < n$:

1. Compute $m' \equiv c^d \bmod n$.
2. Output m'.

Some k exists with $ed = 1 + k\varphi(n)$

Use Fermat's little theorem.

This works:

$$m' \equiv c^d \equiv (m^e)^d \equiv m^{ed} = m^{1+k\varphi(n)} \equiv m \cdot (m^{\varphi(n)})^k \equiv m \cdot 1 \equiv m \bmod n.$$

Math background

Fermat's little theorem: For $a \in \mathbb{Z}$ with $\gcd(a, n) = 1$:

$$a^{\varphi(n)} \equiv 1 \bmod n,$$

where $\varphi(n)$ is Euler's totient function.

Math background

Fermat's little theorem: For $a \in \mathbb{Z}$ with $\gcd(a, n) = 1$:

$$a^{\varphi(n)} \equiv 1 \bmod n,$$

where $\varphi(n)$ is Euler's totient function.

For $n = \prod p_i^{e_i}$ with p_i prime, $p_i \neq p_j$ and integers $e_i > 1$ we have

$$\varphi(n) = \prod (p_i^{e_i} - p_i^{e_i - 1}) = n \prod (1 - 1/p_i).$$

Math background

Fermat's little theorem: For $a \in \mathbb{Z}$ with $\gcd(a, n) = 1$:

$$a^{\varphi(n)} \equiv 1 \bmod n,$$

where $\varphi(n)$ is Euler's totient function.

For $n = \prod p_i^{e_i}$ with p_i prime, $p_i \neq p_j$ and integers $e_i > 1$ we have

$$\varphi(n) = \prod (p_i^{e_i} - p_i^{e_i - 1}) = n \prod (1 - 1/p_i).$$

Thus $\varphi(p \cdot q) = (p - 1)(q - 1)$.

Math background

Fermat's little theorem: For $a \in \mathbb{Z}$ with $\gcd(a, n) = 1$:

$$a^{\varphi(n)} \equiv 1 \bmod n,$$

where $\varphi(n)$ is Euler's totient function.

For $n = \prod p_i^{e_i}$ with p_i prime, $p_i \neq p_j$ and integers $e_i > 1$ we have

$$\varphi(n) = \prod (p_i^{e_i} - p_i^{e_i - 1}) = n \prod (1 - 1/p_i).$$

Thus $\varphi(p \cdot q) = (p - 1)(q - 1)$.
Note that we didn't exclude m with $\gcd(m, n) > 1$.

Unlikely to hit by accident;

Math background

Fermat's little theorem: For $a \in \mathbb{Z}$ with $\gcd(a, n) = 1$:

$$a^{\varphi(n)} \equiv 1 \bmod n,$$

where $\varphi(n)$ is Euler's totient function.

For $n = \prod p_i^{e_i}$ with p_i prime, $p_i \neq p_j$ and integers $e_i > 1$
we have

$$\varphi(n) = \prod (p_i^{e_i} - p_i^{e_i - 1}) = n \prod (1 - 1/p_i).$$

Thus $\varphi(p \cdot q) = (p - 1)(q - 1)$.
Note that we didn't exclude m with $\gcd(m, n) > 1$.

Unlikely to hit by accident;
but actually $m^{1+k\varphi(n)} \equiv m \bmod n$ holds in any case.

Math background

Fermat's little theorem: For $a \in \mathbb{Z}$ with $\gcd(a, n) = 1$:

$$a^{\varphi(n)} \equiv 1 \bmod n,$$

where $\varphi(n)$ is Euler's totient function.

For $n = \prod p_i^{e_i}$ with p_i prime, $p_i \neq p_j$ and integers $e_i > 1$ we have

$$\varphi(n) = \prod (p_i^{e_i} - p_i^{e_i - 1}) = n \prod (1 - 1/p_i).$$

Thus $\varphi(p \cdot q) = (p - 1)(q - 1)$.
Note that we didn't exclude m with $\gcd(m, n) > 1$.

Unlikely to hit by accident;
but actually $m^{1+k\varphi(n)} \equiv m \bmod n$ holds in any case.

Lagrange's theorem: For G a finite group and $a \in G$:

$$a^{|G|} = 1,$$

where 1 denotes the neutral element of the group.

Schoolbook RSA signatures

1977 Rivest, Shamir, Adleman. Do not use Schoolbook RSA in practice!

Schoolbook RSA signatures

1977 Rivest, Shamir, Adleman. Do not use Schoolbook RSA in practice!

KeyGen:

1. Pick primes $p, q; p \neq q$.
2. Compute $n = p \cdot q$, $\varphi(n) = (p-1)(q-1)$.
3. Pick $1 < e < n$ with $\gcd(e, \varphi(n)) = 1$.
4. Compute $d \equiv e^{-1} \bmod \varphi(n)$.
5. Output public key (n, e), private key (n, d).

Schoolbook RSA signatures

1977 Rivest, Shamir, Adleman. Do not use Schoolbook RSA in practice!

KeyGen:

1. Pick primes $p, q; p \neq q$.
2. Compute $n = p \cdot q$, $\varphi(n) = (p-1)(q-1)$.
3. Pick $1 < e < n$ with $\gcd(e, \varphi(n)) = 1$.
4. Compute $d \equiv e^{-1} \bmod \varphi(n)$.
5. Output public key (n, e), private key (n, d).

Sign message $0 \leq m < n$:

1. Compute $s \equiv (h(m))^d \bmod n$.
2. Output s.

Verify signature $0 \leq c < n$:

1. Compute $h' \equiv s^e \bmod n$.
2. Output true if $h' = h(m)$. Else output false.

Schoolbook RSA signatures

1977 Rivest, Shamir, Adleman. Do not use Schoolbook RSA in practice!

KeyGen:

1. Pick primes $p, q; p \neq q$.
2. Compute $n = p \cdot q$, $\varphi(n) = (p-1)(q-1)$.
3. Pick $1 < e < n$ with $\gcd(e, \varphi(n)) = 1$.
4. Compute $d \equiv e^{-1} \bmod \varphi(n)$.
5. Output public key (n, e), private key (n, d).

Sign message $0 \leq m < n$:

1. Compute $s \equiv (h(m))^d \bmod n$.
2. Output s.

Verify signature $0 \leq c < n$:

1. Compute $h' \equiv s^e \bmod n$.
2. Output true if $h' = h(m)$. Else output false.

This works for the same reason as RSA encryption works.
We use $h(m)$ to sign arbitrarily long messages.

Schoolbook RSA signatures

1977 Rivest, Shamir, Adleman. Do not use Schoolbook RSA in practice!

KeyGen:

1. Pick primes $p, q; p \neq q$.
2. Compute $n = p \cdot q$, $\varphi(n) = (p-1)(q-1)$.
3. Pick $1 < e < n$ with $\gcd(e, \varphi(n)) = 1$.
4. Compute $d \equiv e^{-1} \bmod \varphi(n)$.
5. Output public key (n, e), private key (n, d).

Sign message $0 \leq m < n$:

1. Compute $s \equiv (h(m))^d \bmod n$.
2. Output s.

Verify signature $0 \leq c < n$:

1. Compute $h' \equiv s^e \bmod n$.
2. Output true if $h' = h(m)$. Else output false.

This works for the same reason as RSA encryption works.
We use $h(m)$ to sign arbitrarily long messages.

Note that RSA is unusual in that signing matches decryption.