CPSC 418/MATH 318 Introduction to Cryptography

Yet More Number Theory, Goldwasser-Micali PKC, More on Provable Security, RSA-OAEP, Digital Signatures

Renate Scheidler

Department of Mathematics & Statistics
Department of Computer Science
University of Calgary

Week 10

Question:	How can you tell the difference between a good cryptography joke and a random string of words?
Answer:	You can't. They're indistinguishable.

Outline

1. Quadratic Residuosity
   - Legendre Symbol
   - Jacobi Symbol
2. Goldwasser-Micali PKC
3. Provable Security Against Active Attacks
4. RSA-OAEP
5. Where are we at?
6. Digital Signatures
   - Signatures via Public Key Cryptosystems

Quadratic Residuosity

Definition 1 (Quadratic residues and non-residues)

Let $m \in \mathbb{N}$ and $a \in \mathbb{Z}_m^*$. Then a is said to be a *quadratic residue* modulo m if there exists some $x \in \mathbb{Z}$ such that $x^2 \equiv a \pmod{m}$. a is a *quadratic non-residue* modulo m otherwise.

So the quadratic residues modulo m are exactly the squares modulo m.

Notation:

- QR_m: set of quadratic residues modulo m.
- QN_m: set of quadratic non-residues modulo m.

Note 1

$\mathbb{Z}_m^* = QR_m \cup QN_m$.

Prime and Composite Moduli

Suppose $m = p$, an odd prime. For any primitive root g of p

- QR_p is the set of even powers of g: g^{2i}, $0 \leq i \leq (p-3)/2$
- QN_p is the set of odd powers of g: g^{2i+1}, $0 \leq i \leq (p-3)/2$

So $|QR_p| = |QN_p| = (p-1)/2$. (Not true for composite moduli!)

Example 2

Find the quadratic residues and the quadratic non-residue modulo $p = 7$

$$1^2 \equiv 1 \pmod{7},\ 2^2 \equiv 4 \pmod{7},\ 3^2 \equiv 2 \pmod{7},$$
$$4^2 \equiv 2 \pmod{7},\ 5^2 \equiv 4 \pmod{7},\ 6^2 \equiv 1 \pmod{7}.$$

So $QR_7 = \{1, 2, 4\}$ and by elimination $QN_7 = \{3, 5, 6\}$.

Theorem 1

$a \in QR_n$ if and only if $a \in QR_p$ for all primes p dividing n.

Euler's Criterion

Recall Fermat's Theorem: $a^{p-1} \equiv 1 \pmod p$ for p prime and $a \in \mathbb{Z}_p^*$.

$$\begin{aligned}
\text{For } p \text{ odd:} \quad & a^{p-1} \equiv 1 \pmod p \\
\iff \quad & p \text{ divides } a^{p-1} - 1 = (a^{\frac{p-1}{2}} + 1)(a^{\frac{p-1}{2}} - 1) \\
\iff \quad & p \text{ divides } a^{\frac{p-1}{2}} + 1 \text{ or } p \text{ divides } a^{\frac{p-1}{2}} - 1 \\
\iff \quad & a^{\frac{p-1}{2}} \equiv \pm 1 \pmod p .
\end{aligned}$$

This is is almost like "taking square roots" of the Fermat congruence !

Theorem 2 (Euler's Criterion)

$a \in QR_p$ if and only if $a^{\frac{p-1}{2}} \equiv 1 \pmod p$.

Then $a \in QN_p$ if and only if $a^{\frac{p-1}{2}} \equiv -1 \pmod p$.

The Legendre Symbol

Legendre symbols are "quadratic residue indicators" modulo primes:

Definition 3 (Legendre symbol)

Let p be an odd prime. The *Legendre symbol* $\left(\frac{a}{p}\right)$ is defined as:

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } p \mid a \\ 1 & \text{if } a \in QR_p \\ -1 & \text{if } a \in QN_p \end{cases}$$

We can compute Legendre symbols — and by Euler's criterion test whether or not $a \in QR_p$ — in polynomial time using binary exponentiation.

Revised Quadratic Residue Theorems

Example 4

$\left(\frac{2}{7}\right) = 1$ and $\left(\frac{3}{7}\right) = -1$.

Recall Theorem 2 from last week: $a \in QR_n$ iff $a \in QR_p$ for all primes $p \mid n$.

Remark 2 (Reformulation of Theorem 2)

$a \in QR_n$ if and only if $\left(\frac{a}{p}\right) = 1$ for all primes p dividing n.

Note 3 (Euler's Criterion revisited)

$a^{\frac{p-1}{2}} \equiv \left(\frac{a}{p}\right) \pmod p$ for all $a \in \mathbb{Z}$.

Example: Textbook El Gamal is not Semantically Secure

An attacker can chose $M_1 \in QR_p$ and $M_2 \in QN_p$ and distinguish between their encryptions in polynomial time.

- uses properties of quadratic residues and the Legendre symbol
- see Assignment 3 for the full attack

Solution: replace g by $h \equiv g^2 \pmod p$ everywhere

- every quantity occurring in El Gamal is a quadratic residue modulo p.
- can prove that this variation of El Gamal *is* semantically secure, assuming the *decisional Diffie-Hellman problem* is intractable.

  Decisional DHP: given $g, g^a, g^b, g^c \pmod p$, determine whether $g^c \equiv g^{ab} \pmod p)$.

The Jacobi Symbol

Definition 5 (Jacobi symbol)

Let $Q \in \mathbb{N}$ be odd with prime factorization $Q = \prod_{i=1}^{r} q_i^{e_i}$, and let $P \in \mathbb{Z}$.

The *Jacobi symbol* $\left(\frac{P}{Q}\right)$ is defined as

$$\left(\frac{P}{Q}\right) = \prod_{i=1}^{r} \left(\frac{P}{q_i}\right)^{e_i}$$

where $\left(\frac{P}{q_i}\right)$ is the Legendre symbol.

Note 4

If Q is prime, then the Jacobi symbol $\left(\frac{P}{Q}\right)$ and the Legendre symbol $\left(\frac{P}{Q}\right)$ are the same.

Properties of the Jacobi Symbol

$$\left(\frac{P}{Q}\right) = \left(\frac{P \bmod Q}{Q}\right) \tag{1}$$

$$\left(\frac{P_1P_2}{Q}\right) = \left(\frac{P_1}{Q}\right)\left(\frac{P_2}{Q}\right) \tag{2}$$

$$\left(\frac{P}{Q_1Q_2}\right) = \left(\frac{P}{Q_1}\right)\left(\frac{P}{Q_2}\right) \tag{3}$$

$$\left(\frac{2}{Q}\right) = (-1)^{\frac{Q^2-1}{8}}, \quad \left(\frac{-1}{Q}\right) = (-1)^{\frac{Q-1}{2}}, \quad \left(\frac{1}{Q}\right) = 1 \tag{4}$$

If P is odd:

$$\left(\frac{P}{Q}\right) = \left(\frac{Q}{P}\right)(-1)^{\frac{P-1}{2}\frac{Q-1}{2}} \quad \text{(law of quadratic reciprocity)} \tag{5}$$

Computation of Jacobi Symbols

Given the prime factorization of Q, the Jacobi symbol $\left(\frac{P}{Q}\right)$ can be computed in polynomial time:

- Each Legendre symbol $\left(\frac{P}{q_i}\right)$ can be computed in polynomial time via binary exponentiation (due to Euler's criterion).

However, properties (1), (2), (4) and (5) on the previous slide make it possible to compute $\left(\frac{P}{Q}\right)$ in polynomial time *without* factoring Q.

- Method is reminiscent of the Euclidean Algorithm.
- Best illustrated with an example:

Example

$$\left(\frac{127}{35}\right) = \left(\frac{127 \bmod 35}{35}\right) = \left(\frac{22}{35}\right) = \left(\frac{2}{35}\right)\left(\frac{11}{35}\right)$$

$$= (-1)^{\frac{35^2-1}{8}}\left(\frac{11}{35}\right) = (-1)^{\text{odd}}\left(\frac{11}{35}\right) = -\left(\frac{11}{35}\right)$$

$$= -(-1)^{\frac{11-1}{2}\frac{35-1}{2}}\left(\frac{35}{11}\right) = -(-1)^{\text{odd}}\left(\frac{35}{11}\right) = \left(\frac{35}{11}\right)$$

$$= \left(\frac{35 \bmod 11}{11}\right) = \left(\frac{2}{11}\right) = (-1)^{\frac{11^2-1}{8}} = (-1)^{\text{odd}} = -1\,.$$

Note: In fact $\left(\frac{127}{5}\right) = -1$ and $\left(\frac{127}{7}\right) = 1$, so $\left(\frac{127}{35}\right) = (-1)\cdot 1 = -1$.

Example: Leakage in Textbook RSA

Another weakness of textbook RSA arising from its multiplicative property is *leakage* of information: $C \equiv M^e \pmod{n}$ implies

$$\left(\frac{C}{n}\right) = \left(\frac{M}{n}\right)^e = \left(\frac{M}{n}\right),$$

since e is odd and $\left(\frac{M}{n}\right) = \pm 1$.

So one bit of information about the message is leaked (namely the value of the Jacobi symbol $\left(\frac{M}{n}\right)$.

- Thus, basic RSA is *not* sematically/polynomially secure.
- This would not happen if the ciphertext in RSA were randomized.

The Quadratic Residuosity Problem

Recall Remark 2: $a \in QR_n$ iff $\left(\frac{a}{p}\right) = 1$ for all primes $p \mid n$.
So when n is composite, we can have $\left(\frac{a}{n}\right) = 1$, even though $a \notin QR_n$.

Example 6

$\left(\frac{2}{15}\right) = \left(\frac{2}{3}\right)\left(\frac{2}{5}\right) = (-1)(-1) = 1$. So $2 \notin QR_{15}$ but $\left(\frac{2}{15}\right) = 1$.

Definition 7 (Quadratic Residuosity Problem (QRP))

Given an odd composite integer n and any $a \in \mathbb{Z}$ with $\left(\frac{a}{n}\right) = 1$, determine whether $a \in QR_n$.

Note 5

By Remark 1, the Integer Factorization Problem (IFP) is at least as hard as the QRP. Equivalence is believed, but unproved.

Pseudosquares

Definition 8 (Pseudosquare)

Let $n = pq$ with distinct odd primes p, q. A *pseudosquare* $\pmod{n}$ is an integer $a \in \mathbb{Z}$ with $\left(\frac{a}{n}\right) = 1$ and a is a quadratic non-residue $\pmod{n}$.

$\left(\frac{a}{n}\right) = 1$ makes a "look like" a quadratic residue $\pmod{n}$, but $a \notin QR_n$.

Example 8 above establishes that 2 is a pseudosquare modulo 15.

Example 9 (QRP for Pseudosquares)

If $n = pq$ (p, q odd primes), and $\left(\frac{a}{n}\right) = 1$, then there are two possibilities:

- *Case 1*: if $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1$, then a is a quadratic residue modulo n.
- *Case 2*: if $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$, then a is a pseudosquare modulo n.

Here, QRP asks to distinguish quadratic residues (squares) from pseudosquares.

The Goldwasser-Micali PKC

Example 9 above is the basis for the Goldwasser-Micali PKC.

Achieves semantic security assuming the intractability of the QRP.

- Private key: (p, q) where p and q are distinct large primes.
- Public key: (n, y) where $n = pq$ and y is a pseudo-square modulo n.

Note 6

How to find y:

- Generate random integers $y \in \mathbb{Z}_n^*$ until a pseudosquare is found.
- Since there are four combinations $(\pm 1, \pm 1)$ for $\left(\left(\frac{y}{p}\right), \left(\frac{y}{q}\right)\right)$, one in four choices of y yields $(-1, -1)$.
- Hence, we expect to find a pseudosquare $\pmod{n}$ after four trials at a value of y.

Encryption

To encrypt a message M intended for a user with the above public/private key pair, proceed as follows:

1. Represent M as a bit-string $(m_1, m_2, \ldots, m_t)$ $(m_i \in \{0, 1\})$.
2. For $i = 1, \ldots, t$:
   a. Select random $r_i \in \mathbb{Z}_n^*$.
   b. Put $c_i \equiv y^{m_i} r_i^2 \pmod{n}$ with $0 < c_i < n$
   (so $c_i \equiv r_i^2 \pmod{n}$ if $m_i = 0$ and $c_i \equiv y r_i^2 \pmod{n}$ if $m_i = 1$).
3. Send $C = (c_1, c_2, \ldots, c_t)$.

Decryption

To decrypt $C = (c_1, c_2, \ldots, c_t)$, the recipient proceeds as follows:

1. for $i = 1, \ldots, t$:
   a. Compute the Legendre symbol $e_i = \left(\frac{c_i}{p}\right)$.
   b. $m_i = (1 - e_i)/2$ (so $m_i = 0$ if $e_i = 1$ and $m_i = 1$ if $e_i = -1$).
2. $M = (m_1, m_2, \ldots, m_t)$.

Correctness of Decryption

Proof that decryption is correct.

For all $i \in \{1, \ldots, t\}$, we have

$$e_i = \left(\frac{c_i}{p}\right) = \left(\frac{y^{m_i} r_i^2}{p}\right) = \left(\frac{y}{p}\right)^{m_i} \left(\frac{r_i}{p}\right)^2 = \left(\frac{y}{p}\right)^{m_i} (\pm 1)^2 = \left(\frac{y}{p}\right)^{m_i} = (-1)^{m_i}.$$

Thus, if $e_i = 1$ then $m_i = 0$ and if $e_i = -1$ then $m_i = 1$. $\square$

Polynomial Security of Goldwasser-Micali

Proof sketch of polynomial security.

Since r_i is selected at random:

- r_i^2 is a random quadratic residue modulo n
- thus, $y r_i^2$ is a random pseudosquare modulo n.

The cryptanalyst only sees a sequence of r_i^2 or $y r_i^2$ (quadratic residues and pseudosquares), and as the QRP is hard, she cannot distinguish one from the other. $\square$

Major disadvantages:

- Huge message expansion, by a factor of $\log_2(n)$: a t-bit message yields a ciphertext of length $\approx t \log_2(n)$
- Costly decryption algorithm (t Legendre symbols)

IND-CCA1 and IND-CCA2 Security

To address chosen *ciphertext* attacks, we need even stronger security notions than semantic/polynomial security

Definition 10 (IND-CCA1 and IND-CCA2 security)

A PKC is IND-CCA (or IND-CCA1) secure if it satisfies *indistinguishability under chosen ciphertext attacks*; in other words, no (active) adversary with blackbox access to a *decryption oracle* (that decrypts arbitrary ciphertexts) can in expected polynomial time select two plaintext messages M_1 and M_2 and then correctly distinguish between encryptions of M_1 and M_2 with probability significantly greater than 1/2.

A PKC is IND-CCA2 secure if it satisfies *indistinguishability under adaptive chosen ciphertext attacks*, i.e. an attacker may use the decryption oracle adaptively (of course as always, she may not submit the encryption given to her to distinguish M_1 from M_2).

IND-CCA1 and IND-CCA2 Security, cont.

IND-CCA has the same definition as as polynomial security except that access to a decryption oracle is granted. It is the active attack equivalent of semantic security.

In addition, for IND-CCA2, an adaptive CCA strategy is permitted.

Security levels:

- IND-CCA2 — indistinguishability under adaptive chosen ciphertext attacks
- IND-CCA1 — indistinguishability under (non-adaptive) chosen ciphertext attacks
- IND-CPA — indistinguishability under chosen plaintext attacks (same as polynomial security)

Note that IND-CCA2 $\Longrightarrow$ IND-CCA1 $\Longrightarrow$ IND-CPA.

Idea of Malleability

Recall the multiplicative attacks on RSA where an attacker proceeds as follows:

1. Generates $X \in \mathbb{Z}_n^*$ with $X^e \not\equiv 1 \pmod{n}$.
2. Computes $C' \equiv CX^e \pmod{n}$ (this is the chosen ciphertext; note that $C' \neq C$).
3. Obtains the corresponding plaintext
$$M' \equiv (C')^d \equiv C^d(X^e)^d \equiv MX \pmod{n}$$
4. Computes $M \equiv M'X^{-1} \pmod{n}$, where X^{-1} is the inverse of $X \pmod{n}$

The attacker can generate C' from C in such a way that M' is related to M in a known, efficiently computable manner (i.e. C is *malleable*).

Non-Malleability

Definition 11 (Non-malleability)

A PKC is *non-malleable* if, given a ciphertext C corresponding to some message M, it is computationally infeasible to generate a different ciphertext C' whose decryption M' is related to M in some known manner, i.e. $M' = f(M)$ for some arbitrary but known (efficiently invertible) function f.

Non-malleability provides data integrity of ciphertexts *without* any source identification (public-key analogue of "encrypt-then-MAC").

We have

- NM-CPA $\Longrightarrow$ IND-CPA
- NM-CCA1 $\Longrightarrow$ IND-CCA1
- NM-CCA2 $\Longleftrightarrow$ IND-CCA2

It is known that IND-CPA $\not\Longrightarrow$ NM-CPA and IND-CCA1 $\not\Longrightarrow$ NM-CCA1.

Plaintext Awareness

Plaintest awareness is a very strong notion of security.

Definition 12 (Plaintext awareness)

A PKC is *plaintext-aware* if it is computationally infeasible for an adversary to produce a "valid" ciphertext (whose decryption has prescribed redundancy) without knowledge of the corresponding plaintext.

This means it is infeasible to create a valid ciphertext without being aware of the corresponding plaintext.

A plaintext-aware PKC resists adaptive CCAs because any adaptive modification of a target ciphertext will with high probability not be "valid."

- Plaintext awareness $\Longrightarrow$ Indistinguishability.
- Plaintext awareness $\Longrightarrow$ Non-malleability.

Optimal Asymmetric Encryption Padding (OAEP)

Optimal Asymmetric Encryption Padding (OAEP):

- Bellare and Rogaway, Eurocrypt 1994
- An invertible transformation from a PKC plaintext space to the domain of a one-way trapdoor function (e.g. a public key encryption map).

OAEP augments PKCs to provide plaintext awareness by adding redundancy and transforming the plaintext before encryption. It works with most PKCs.

RSA-OAEP

Standardized in RSA's PKCS#1, IEEE P1363, e-commerce protocol SET (Secure Electronic Transaction)

Parameters

- n — length of plaintext messages to encrypt (in bits)
- (N, e) — Alice's RSA public key (N has $k = n + k_0 + k_1$ bits, where 2^{-k_0} and 2^{-k_1} must be sufficiently small). For example, if $k = 3072$, can take $k_0 = k_1 = 128$ and $n = 2816$.
- d — Alice's RSA private key
- $G : \{0,1\}^{k_0} \mapsto \{0,1\}^{k-k_0}$ (random function)
- $H : \{0,1\}^{k-k_0} \mapsto \{0,1\}^{k_0}$ (random function)

Encryption

Encryption (message M):

1. Generate a random k_0-bit number r.
2. Compute $s = (M \| 0^{k_1}) \oplus G(r)$ (append k_1 0 bits to M for data integrity checking and XOR with $G(r)$). Note: s has $n + k_1 = k - k_0$ bits.
3. Compute $t = r \oplus H(s)$. Note: t has k_0 bits, so $(s\|t)$ has k bits (same as N), but could be a bit bigger than N. If $(s\|t) \geq N$, go to 1 (make sure concatenation of s and t as an integer is less than the RSA modulus).
4. RSA-encrypt $(s\|t)$, i.e., compute $C \equiv (s\|t)^e \pmod{N}$.

$$C \equiv \Big((M\|0^{k_1} \oplus G(r)) \parallel (r \oplus H(M\|0^{k_1} \oplus G(r))) \Big)^e \pmod{N}$$

Decryption

$$C \equiv \Big(\big(M \| 0^{k_1} \oplus G(r)\big) \parallel \big(r \oplus H(M\|0^{k_1} \oplus G(r))\big) \Big)^e \pmod{N} \ .$$

Decryption (ciphertext C):

1. Compute $(s\|t) \equiv C^d \pmod{N}$.

$$C^d \equiv \big(M\|0^{k_1} \oplus G(r)\big) \parallel \big(r \oplus H(M\|0^{k_1} \oplus G(r))\big) \pmod{N}$$

2. Compute $u = t \oplus H(s)$ (k_0 bit) and $v = s \oplus G(u)$ ($k - k_0$ bits).

$$u = t \oplus H(s) = \big(r \oplus H(M\|0^{k_1} \oplus G(r))\big) \oplus H(M\|0^{k_1} \oplus G(r)) = r$$

$$v = s \oplus G(u) = \big(M\|0^{k_1} \oplus G(r)\big) \oplus G(r) = M\|0^{k_1}$$

3. Output M if $v = (M\|0^{k_1})$ (*i.e.* the decrypted message has the required redundancy), otherwise reject as invalid.

Security of RSA-OAEP

Can be proven to be plaintext-aware assuming that the RSA problem (computing e-th roots modulo n) is intractable:

- Defeats CCAs because only messages with the prescribed redundancy (0^{k_1} appended) are accepted. Probability of a random ciphertext decrypting to an acceptable value is 2^{-k_1}.
- Plaintext is also randomized — prevents small message space attacks (2^{k_0} possible encryptions of each message).

Random Oracle Model

RSA-OAEP's proof of security relies on the assumption that the functions G and H are random, i.e. mathematical functions mapping every possible query (input) to a random response from its output domain (output).

Such functions are referred to as *random oracles*, and security proofs relying on this type of assumption are said to use the *random oracle model* (ROM).

In practice, G and H are realized with a hash function like SHA-3.

- In this case, the encryption scheme cannot be proven to be plaintext-aware.
- Nevertheless provides much greater security assurances than standard RSA

IND-CCA2 Security without Random Oracles

A variation of the El Gamal PKC due to Cramer and Shoup (CRYPTO 1998) is IND-CCA2 secure under the assumption that the decision Diffie-Hellman problem is hard.

- The proof does *not* use the ROM.
- Dent (EUROCRYPT 2006) showed that it is also plaintext-aware, again without assuming random oracles.

Were are we at?

Recall cryptographic services:

- Data confidentiality: discussed
- Data integrity: discussed
- Authentication, next
- Non-repudiation: next
- Access Control: discussed a bit

Recall cryptographic mechanisms:

- Encryption — for confidentiality and limited data integrity: discussed
- Hash functions, Message Authentication Codes (MACs) — for data integrity : discussed
- Digital signatures — for data origin authentication and non-repudiation : next
- Authentication protocols — for entity authentication

Digital Signatures: Definition

Data origin authentication is usually achieved by means of a *signature*, i.e. a means by which the recipient of a message can authenticate the source of the message.

Definition 13 (Digital signature)

A means for data origin authentication that should have two properties:

1. Only the sender can produce their signature.
2. *Anyone* should be easily able to verify the validity of the signature.

Digital Signatures: Observations

Observations:

- Properties 1 and 2 provide *non-repudiation:* if there is a dispute over a signature (a receiver claims that the sender signed the message, whereas the signer claims they didn't), anyone can resolve the dispute. For ordinary written signatures, one might need a hand-writing expert.
- Signatures are different from MACs:
  - both sender and receiver can generate a MAC, whereas only the sender can generate a signature.
  - only sender and receiver can verify a MAC, whereas anyone can verify a signature.
- In order to prevent *replay attacks* (replay a signed message later), it may be necessary to include a time stamp or sequence numbers in the signature.

Signature Capable PKCs

Definition 14 (Signature capability)

A PKC is *signature capable* if $\mathcal{M} = \mathcal{C}$.

As a result, in a signature capable PKC, decryptions are right and left inverses, *i.e.* actual inverses, of encryptions (because $\mathcal{M} = \mathcal{C}$ implies that the encryption injections are actually bijections).

In particular $E_{K_1}(D_{K_2}(M)) = M$ for all $M \in \mathcal{M}$.

Example 15

RSA has signature capability. El Gamal and Goldwasser-Micali do not.

Note that $\mathcal{M} \neq \mathcal{C}$ for El Gamal and Goldwasser-Micali.

Signatures Without Secrecy Using PKC

Alice wishes to send a non-secret message M to Bob along with a signature S that authenticates her to Bob.

She sends (A, M, S) where

- A is her identity,
- M is the message,
- $S = D_A(M)$ is the "decryption" of M under her private key.

To verify S, Bob

- checks A and looks up Alice's public key,
- computes the "encryption" $E_A(S)$ of S under Alice's public key,
- accepts the signature if and only if $M = E_A(S)$

Note that $E_A(S) = E_A(D_A(M)) = M$ if everything was done correctly.

RSA Digital Signatures

Alice wishes to send a non-secret message M to Bob along with a signature S that authenticates her to Bob.

She sends (A, M, S) where

- A is her identity,
- M is the message,
- $S = M^{d_A} \pmod{n_A}$, where d_A is her RSA private key.

To verify S, Bob

- checks A and looks up Alice's RSA public key (e_A, n_A),
- computes the "encryption" $S^{e_A} \equiv M' \pmod{n_A}$,
- accepts the signature if and only if $M = M'$

Properties

Anyone can verify a signature since anyone can encrypt under Alice's public key.

In order to forge a signature of a particular message M, Eve would have to be able to do decryption under Alice's private key.

Signatures With Secrecy Using PKC

Alice wishes to send an authenticated secret message M to Bob.

She sends $(A, E_B(S, M))$ where A and S are as before and E_B denotes encryption under Bob's public key.

To verify S, Bob decrypts $E_B(S, M)$ and then verifies S as before.

Security of Signatures

Definition 16 (Existential forgery)

A signature scheme is susceptible to *existential forgery* if an adversary can forge a valid signature of another entity for at least one message.

Goals of the attacker:

- total break — recover the private key
- universal forgery — can generate a signature for any message
- selective forgery — can generate a signature for some message of choice
- existential forgery — can generate a signature for at least one message

Existential Forgery on PKC-Generated Signatures

Consider generating a signature S to a message M using a signature-capable PKC as described above.

Eve can create a forged signature from Alice as follows:

1. Selects random $S \in \mathcal{M}$,
2. Computes $M = E_A(S)$,
3. Sends (A, M, S) to Bob.

Bob computes $E_A(S)$ which is M and thus accepts the "signature" S to "message" M.

Usually foiled by language redundancy, but may be a problem if M is random (eg. a cryptographic key).

Preventing This Existential Forgery Attack

Solution:

- Alice sends $(A, M, S = D_A(H(M)))$ where H is a public pre-image resistant hash function on $\mathcal{M}$.
- Bob computes $E_A(S)$ and $H(M)$, and accepts the signature if and only if they match.

Foils the attack:

- If Eve generates random S, then she would have to find X such that $H(X) = M = E_A(S)$ (*i.e.* a pre-image under H), and send (A, X, S) to Bob.
- Bob then computes $D_A(H(X))$ and compares with S.
- Not computationally feasible if H is pre-image resistant.

Existential Forgery if H is not Collision Resistant

Suppose Alice uses a pre-image resistant hash function as described above to sign her messages.

If H is not collision resistant, Eve can forge a signature as follows:

1. Find $M, M' \in \mathcal{M}$ with $M \neq M'$ and $H(M) = H(M')$ (a collision)
2. If S is the signature to M, then S is also the signature to M', as $E_A(S) = H(M) = H(M')$

Note that if Eve intercepts (A, M, S), then she could also find a weak collision M' with $H(M) = H(M')$.

Summary on Signatures via PKC

1. Use a secure signature capable PKC and a cryptographic (i.e. preimage resistant and collision resistant) hash function H (security depends on both).

2. Signing $H(M)$ instead of M also results in faster signature generation if M is long.

3. H should be a fixed part of the signature protocol, so Eve cannot just substitute H with a cryptographically weak hash function.