Fundamentals of Cryptography

David Jao

Topics in Quantum-Safe Cryptography

CryptoWorks21

UNIVERSITY OF WATERLOO

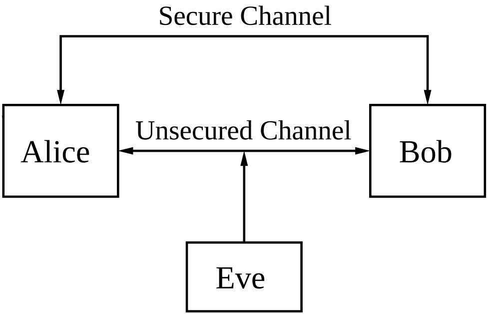
June 30, 2016

Part VIII

Public-key cryptography

Drawbacks with symmetric-key cryptography

Symmetric-key cryptography: Communicating parties a priori share some secret information.



Diffie-Hellman key exchange (1976)

Given a group G and an element $g \in G$, two parties can establish a shared secret over a public channel by:

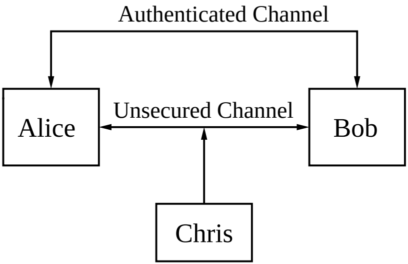
- choosing (respectively) secret integers α and β
- sending (respectively) g^α and g^β
- computing (respectively) $g^{\alpha\beta} = (g^\alpha)^\beta$ and $(g^\beta)^\alpha$

The security of Diffie-Hellman is based on the computational infeasibility of discrete logarithms:

- Given g and g^α, find α (modulo the order of g)

Public-key cryptography

- Public-key cryptography: Communicating parties a priori share some authenticated (but non-secret) information.



- Invented by Ralph Merkle, Whitfield Diffie, and Martin Hellman in 1976.
  (And in 1970 by researchers at GCHQ.....)

Public-key vs. symmetric-key

Advantages of public-key cryptography:

- No requirement for a secret channel.
- Each user has only 1 key pair, which simplifies key management.
- Facilitates the provision of non-repudiation services (with digital signatures).

Disadvantages of public-key cryptography:

- Public keys are typically larger than symmetric keys.
- Public-key schemes are slower than their symmetric-key counterparts.

Definition of public-key cryptography

Definition: A *public-key cryptosystem* consists of:

- M – the plaintext space,
- C – the ciphertext space,
- K_{pubkey} – the space of public keys,
- K_{privkey} – the space of private keys,
- A randomized algorithm $\mathcal{G}\colon \{\mathbb{1}^\ell : \ell \in \mathbb{N}\} \to K_{\text{pubkey}} \times K_{\text{privkey}}$, called a *key-generation function*,
- An *encryption* algorithm $\mathcal{E}\colon K_{\text{pubkey}} \times M \to C$,
- A *decryption* algorithm $\mathcal{D}\colon K_{\text{privkey}} \times C \to M$.

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

$$\mathcal{D}(k_{\text{privkey}}, \mathcal{E}(k_{\text{pubkey}}, m)) = m$$

for all $m \in M$.

The RSA encryption 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.
- Also invented by Clifford Cocks in 1973 (GCHQ).
- Key generation:
  - Choose random primes p and q with $\log_2 p \approx \log_2 q \approx 2^{\ell/2}$.
  - Compute $n = pq$ and $\phi(n) = (p-1)(q-1)$.
  - Choose an integer e with $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$.
  - Compute $d = e^{-1} \bmod \phi(n)$. The public key is (n, e) and the private key is (n, d).
- Message space:
  $M = C = \mathbb{Z}_n^* = \{m \in \mathbb{Z} : 0 \leq m < n \text{ and } \gcd(m, n) = 1\}$.
- Encryption: $\mathcal{E}((n, e), m) = m^e \bmod n$.
- Decryption: $\mathcal{D}((n, d), c) = c^d \bmod n$.

A framework for security definitions

Recall that for a symmetric-key encryption scheme, security depends on three questions:

1. How does the adversary interact with the communicating parties?
2. What are the computational powers of the adversary?
3. What is the adversary's goal?

- Basic assumption (Kerckhoffs's principle, Shannon's maxim): The adversary knows everything about the algorithm, except the secret key k. (Avoid security by obscurity!!)

The same principles also apply to public-key cryptography.

Chosen ciphertext security

Definition

A public-key cryptosystem is said to be secure if it is semantically secure against an adaptive chosen-ciphertext attack by a computationally bounded adversary.

- Adaptive chosen-ciphertext attack: The adversary can choose which ciphertexts to query, based on the results of previous queries.
- RSA with proper random padding (e.g. RSA-OAEP) is secure.

Thought exercise: Why is semantic security against a chosen-plaintext attack a good enough definition for symmetric-key encryption schemes, but not for public-key cryptosystems?

Part IX

Digital signatures

Definition of digital signatures

Definition: A *digital signature scheme* consists of:

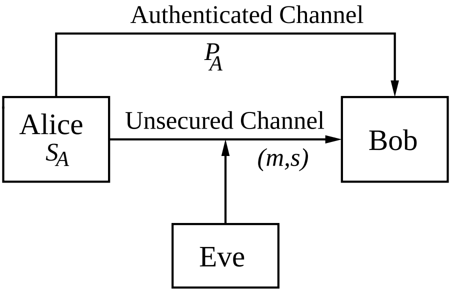
- M – the plaintext space,
- S – the signature space,
- K_{pubkey} – the space of public keys,
- K_{privkey} – the space of private keys,
- A randomized algorithm $\mathcal{G}\colon \{1\!\!1^{\ell} : \ell \in \mathbb{N}\} \to K_{\text{pubkey}} \times K_{\text{privkey}}$, called a *key-generation function*,
- A *signing* algorithm $\mathcal{S}\colon K_{\text{privkey}} \times M \to S$,
- A *verification* algorithm $\mathcal{V}\colon K_{\text{pubkey}} \times M \times S \to \{\textbf{true}, \textbf{false}\}$.

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 $m \in M$.

Digital signatures



- To sign a message m, Alice does:
  1. Compute $s = \text{Sign}(S_A, m)$.
  2. Send m and s to Bob.
- To verify Alice's signature s on m, Bob does:
  1. Obtain an authentic copy of Alice's public key P_A.
  2. Accept if $\text{Verify}(P_A, m, s) = \text{Accept}$.

Basic security requirements

Goals of a digital signature scheme:

- *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.

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: Each entity A does the following:

1. Randomly select 2 large distinct primes p and q of the same bitlength.
2. Compute $n = pq$ and $\phi(n) = (p-1)(q-1)$.
3. Select arbitrary e, $1 < e < \phi(n)$, such that $\gcd(e, \phi(n)) = 1$.
4. Compute d, $1 < d < \phi(n)$, such that $ed \equiv 1 \pmod{\phi(n)}$.
5. A's public key is (n, e); A's private key is d.

Signature Generation and Verification

Signature generation: To sign a message $m \in M$, A does the following:

1. Compute $H(m)$, where $H\colon M \to \mathbb{Z}_n^*$ is a hash function.
2. Compute $s = H(m)^d \bmod n$.
3. A's signature on m is s.

Signature verification: To verify A's signature s on m, B does the following:

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

Goals of the Adversary

1. Total break: E recovers A's private key, or a method for systematically forging A's signatures (i.e., E can compute A's signature for arbitrary messages).
2. Selective forgery: E forges A's signature for a selected subset of messages.
3. Existential forgery: E forges A's signature for a single message; E may not have any control over the content or structure of this message.

Attack Model

Types of attacks E can launch:

1. Key-only attack: The only information E has is A's public key.
2. Known-message attack: E knows some message/signature pairs.
3. Chosen-message attack: E has access to a signing oracle which it can use to obtain A's signatures on some messages of its choosing.

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.

Note: The adversary has access to a signing oracle. Its goal is to compute a single valid message/signature pair for any message that was not previously given to the signing oracle.

Further topics

Cryptographic primitives:

- Elliptic curve cryptography
- Post-quantum cryptography: lattices, codes, isogenies

Protocols:

- Key exchange
- Homomorphic encryption
- Functional encryption