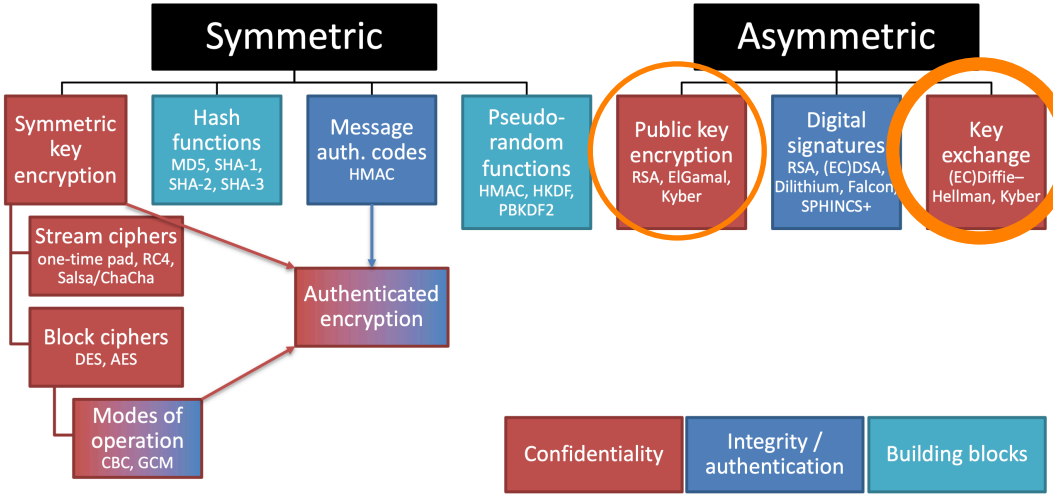
Topic 3.3
Public key cryptography – Diffie–Hellman key exchange

Douglas Stebila

CO 487/687: Applied Cryptography
Fall 2024

Map of cryptographic primitives



Key establishment problem

- Recall that in symmetric-key cryptography, two parties need to establish a shared secret key.
- Establishing shared secret keys is a hard problem.
- Possible solutions:
  1. Use *public-key cryptography*, which does not require shared secret keys.
  2. Use a *key-exchange protocol*, which is a protocol specifically designed to establish shared secret keys from scratch.

Outline

Diffie–Hellman key exchange

Elgamal encryption: Public key encryption built from Diffie–Hellman

Diffie–Hellman key exchange

W. Diffie and M. Hellman, “New directions in cryptography,” IEEE Transactions on Information Theory **22** (6), pp. 644–654, 1976.

This article contains several new ideas, any one of which would have been a major breakthrough:

- Formulates the definition of public-key cryptography
- Formulates the definition of digital signatures
- Describes a practical key-exchange protocol

Mathematical notation

- $\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$
- $\mathbb{Z}_n^* = \{x \in \mathbb{Z}_n : \gcd(x, n) = 1\}$
- Equivalently, $\mathbb{Z}_n^* = \{x \in \mathbb{Z}_n : x^{-1} \text{ exists}\}$
- Examples:
  - $\mathbb{Z}_2^* = \{1\}$
  - $\mathbb{Z}_3^* = \{1, 2\}$
  - $\mathbb{Z}_4^* = \{1, 3\}$
  - $\mathbb{Z}_5^* = \{1, 2, 3, 4\}$
  - $\mathbb{Z}_6^* = \{1, 5\}$
- When $n = p$ is prime, $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$

Note that $\mathbb{Z}_n^*$ is **closed under multiplication**:
If $x \in \mathbb{Z}_n^*$ and $y \in \mathbb{Z}_n^*$, then $xy \in \mathbb{Z}_n^*$

Order and generators

Definition

The order of an element $x \in \mathbb{Z}_n^*$ is defined to be the smallest positive integer t such that $x^t = 1$ in $\mathbb{Z}_n^*$.

Definition

An element g of $\mathbb{Z}_n^*$ is defined to be a generator of $\mathbb{Z}_n^*$ if every y in $\mathbb{Z}_n^*$ can be written as $y = g^x$ for some integer x.

Generators

Definition

An element g of $\mathbb{Z}_n^*$ is defined to be a generator of $\mathbb{Z}_n^*$ if every y in $\mathbb{Z}_n^*$ can be written as $y = g^x$ for some integer x.

Example

$3 \in \mathbb{Z}_{17}^*$ is a generator of $\mathbb{Z}_{17}^*$:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3^i	3	9	10	13	5	15	11	16	14	8	7	4	12	2	6	1

$2 \in \mathbb{Z}_{17}^*$ is not a generator of $\mathbb{Z}_{17}^*$:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2^i	2	4	8	16	15	13	9	1	2	4	8	16	15	13	9	1

Facts about order and generators

$\mathbb{Z}_n^*$ for general n

- Every element has order that divides $\varphi(n)$
- If n is composite, then $\mathbb{Z}_n^*$ is not cyclic and does not have a generator

$\mathbb{Z}_p^*$ for prime p

- $\mathbb{Z}_p^*$ is a cyclic group
- Every generator of $\mathbb{Z}_p^*$ has order $\varphi(p) = p - 1$

Groups

A group is a set G with an operation $*$ such that:

- $a * (b * c) = (a * b) * c$ for all $a, b, c \in G$ (associative law),
- There exists $\mathcal{O} \in G$ such that $a * \mathcal{O} = \mathcal{O} * a = a$ for all $a \in G$,
- For all $a \in G$ there exists $a^{-1} \in G$ with $a * a^{-1} = a^{-1} * a = \mathcal{O}$.

A commutative or abelian group satisfies the additional property:

- $a * b = b * a$ for all $a, b \in G$ (commutative law).

G is cyclic group if it can be generated by a generator g, in which case we write $G = \langle g \rangle$.

Groups

Examples of (abelian) groups:

- The set of integers: $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$ under the addition operation ($\mathcal{O} = 0$).
- The set of integers mod p: $\mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$ under the addition operation ($\mathcal{O} = 0$).
- The set of nonzero integers mod a prime p: $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$ under the multiplication operation ($\mathcal{O} = 1$).

Diffie–Hellman notation

Notation

- p is a prime
- $\mathbb{Z}_p^*$ is the set $\{1, 2, \ldots, p-1\}$
- $g \in \mathbb{Z}_p^*$ is an element of $\mathbb{Z}_p^*$ of large prime order q
  - g is not a generator of $\mathbb{Z}_p^*$, but does generate a subgroup $\mathbb{Z}_p^*$ of prime order q

Diffie–Hellman key exchange

Diffie–Hellman key exchange allows two people to establish a shared secret, without *transmitting* any secret information.

Pick a prime p and
an element $g \in \mathbb{Z}_p^*$
of large prime order q

Pick $a \in \mathbb{Z}_q$ at random.
Send $g^a \bmod p$ to Bob. $\longrightarrow$ $g^a \bmod p$

Pick $b \in \mathbb{Z}_q$ at random.
$g^b \bmod p$ $\longleftarrow$ Send $g^b \bmod p$ to Alice.

Both Alice and Bob can compute $g^{ab} \equiv (g^a)^b \equiv (g^b)^a \bmod p$.

Square and multiply algorithm

To compute $g^a \bmod p$:

1. Write a in binary,
2. Use repeated squaring to compute g^{2^i}, for $i \geq 0$,
3. Multiply together the g^{2^i} values corresponding to the 2^i values appearing in the binary representation of a.

Square and multiply algorithm

Algorithm 1 Sequential algorithm for computing $g^a \bmod p$.

Given: $g, a, p \in \mathbb{N}$

1: $k \leftarrow \lceil \log_2 a \rceil$
2: $y_0 \leftarrow g$
3: **for** $i = 1, \ldots, k$: $y_i \leftarrow y_{i-1}^2 \bmod p$
4: Write $a = a_k a_{k-1} \ldots a_1 a_0$ in binary
5: $z \leftarrow 1$
6: **for** $i = 0, \ldots, k$: **if** $a_i = 1$ **then** $z \leftarrow z \cdot y_i \bmod p$
7: **return** z

Square and multiply algorithm

Algorithm 2 Recursive algorithm for computing $g^a \bmod p$.

Given: $g, a, p \in \mathbb{N}$

1: **if** $a = 0$ **then** output 1
2: **else if** a is even **then** output $(g^{\frac{a}{2}} \bmod p)^2 \bmod p$
3: **else if** a is odd **then** output $(g^{a-1} \bmod p) \cdot (g \bmod p)$

Algorithm 3 Iterative algorithm for computing $g^a \bmod p$.

Given: $g, a, p \in \mathbb{N}$

1: $y \leftarrow 1$, $b \leftarrow a$, $x \leftarrow g$
2: **while** $b > 1$ **do**
3: **if** b is odd **then** $y \leftarrow x \cdot y$, $b \leftarrow b - 1$
4: $x \leftarrow x^2$, $b \leftarrow b/2$
5: **return** $x \cdot y$

Diffie–Hellman assumption

(All quantities except a and b are assumed to be taken mod p.)

Security:

- Recall that **no** secret information is transmitted.
- Since g^a and g^b are transmitted, they aren't secret.
- But g^{ab} is supposed to be secret. Hence we assume:

Computational Diffie–Hellman assumption (CDH)

Let $a, b \in_R \mathbb{Z}_q$. Given g, g^a, g^b, it is computationally infeasible to determine g^{ab}.

Decisional Diffie–Hellman assumption (DDH)

Let $a, b \in_R \mathbb{Z}_q$. Given g, g^a, g^b, and either g^{ab} or g^c for a random $c \in_R \mathbb{Z}_q$, it is computationally infeasible to determine whether you were given g^{ab} or g^c.

Discrete logarithm assumption

(All quantities except a and b are assumed to be taken mod p.)

If we could compute a from g and g^a, then we could (trivially) break Diffie–Hellman.

Discrete logarithm assumption (DLOG)

Let $a \in_R \mathbb{Z}_q$. Given g and g^a, it is computationally infeasible to determine a.

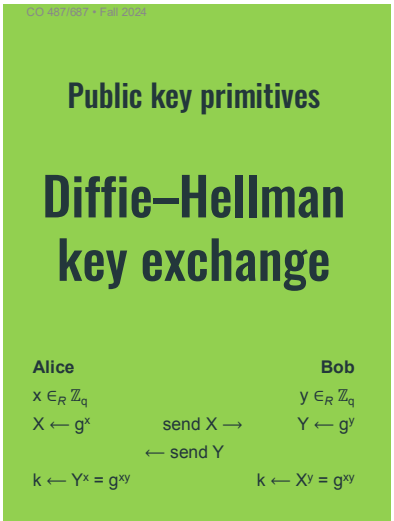
Theorem (U. Maurer & S. Wolf, 1999)

For almost all values of p, the CDH assumption and the DLOG assumption are equivalent.

Diffie–Hellman vs. RSA

- Diffie–Hellman:
  - Published in 1976. (Malcolm Williamson, GCHQ: 1974)
  - Key exchange only: Can exchange shared secret keys, but not arbitrary messages.
  - Interactive: Both parties must be online simultaneously.
  - Forward secrecy: Compromising one key exchange does not compromise future key exchanges.
- RSA:
  - Published in 1978. (Clifford Cocks, GCHQ: 1973)
  - Public-key cryptosystem: Can exchange any message chosen by the sender.
  - Non-interactive: An encrypted message can be decrypted later.
  - No forward secrecy: A compromised private key exposes all previous ciphertexts.

4+8+7 things to remember from CO 487



- **Building block for confidentiality.**
- Only secure against passive attacks, need to add authentication to protect against man-in-the-middle attacks.
- Secure options as of 2024:
  - Not post-quantum:
    - Elliptic curve DH in cryptographically secure groups like nistp256 or curve25519
  - Post-quantum:
    - ML-KEM (Kyber)

Outline

Diffie–Hellman key exchange

Elgamal encryption: Public key encryption built from Diffie–Hellman

Elgamal public key encryption

Taher Elgamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory **31** (4), pp. 469–472, 1985.

- Setup:
  - Choose a single, globally public prime p.
  - Choose a single, globally public element $g \in \mathbb{Z}_p^*$ of large prime order q.
- Key generation:
  - Choose $x \in_R \mathbb{Z}_q$
  - Set $k_{\text{pubkey}} = g^x \bmod p$ and $k_{\text{privkey}} = x$.
- Encryption: Given $m \in \mathbb{Z}_p^*$,
  - Choose $r \in_R \mathbb{Z}_q$
  - Set $E(m) = (g^r, m \cdot (g^x)^r) \bmod p$
- Decryption: Given a ciphertext $(c_1, c_2) \in (\mathbb{Z}_p^*)^2$, compute $D(c_1, c_2) = c_2 \cdot (c_1^{-1})^x \bmod p$.

Elgamal design principles

Basic idea:

- Publish one half of the Diffie–Hellman key exchange as the public key.
- Include the other half of the key exchange as the first half of the ciphertext.
- Encrypt the plaintext under the shared secret key as the second half of the ciphertext.

In other words, Elgamal is the public-key encryption analogue of Diffie–Hellman.

Interestingly, the encryption operation in Elgamal is randomized, whereas in RSA it was deterministic.