Lecture 12: Key-Agreement and Public-key Encryption

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have $a \bullet b \in G$

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have $a \bullet b \in G$
  - Associativity: For all $a, b, c \in G$, we have $(a \bullet b) \bullet c = a \bullet (b \bullet c)$

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have $a \bullet b \in G$
  - Associativity: For all $a, b, c \in G$, we have $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
  - Identity: There exists an element e such that for all $a \in G$, we have $e \bullet a = a$

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have $a \bullet b \in G$
  - Associativity: For all $a, b, c \in G$, we have $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
  - Identity: There exists an element e such that for all $a \in G$, we have $e \bullet a = a$
  - Inverse: For every $a \in G$, there exists $b \in G$ such that $a \bullet b = e$

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have $a \bullet b \in G$
  - Associativity: For all $a, b, c \in G$, we have $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
  - Identity: There exists an element e such that for all $a \in G$, we have $e \bullet a = a$
  - Inverse: For every $a \in G$, there exists $b \in G$ such that $a \bullet b = e$
- Think: Is $a \bullet b$ always equal to $b \bullet a$?

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have $a \bullet b \in G$
  - Associativity: For all $a, b, c \in G$, we have $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
  - Identity: There exists an element e such that for all $a \in G$, we have $e \bullet a = a$
  - Inverse: For every $a \in G$, there exists $b \in G$ such that $a \bullet b = e$
- Think: Is $a \bullet b$ always equal to $b \bullet a$?
  - Read: Abelian Groups

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have $a \bullet b \in G$
  - Associativity: For all $a, b, c \in G$, we have $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
  - Identity: There exists an element e such that for all $a \in G$, we have $e \bullet a = a$
  - Inverse: For every $a \in G$, there exists $b \in G$ such that $a \bullet b = e$
- Think: Is $a \bullet b$ always equal to $b \bullet a$?
  - Read: Abelian Groups
- Think: Can there be different left and right identity elements?

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have $a \bullet b \in G$
  - Associativity: For all $a, b, c \in G$, we have $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
  - Identity: There exists an element e such that for all $a \in G$, we have $e \bullet a = a$
  - Inverse: For every $a \in G$, there exists $b \in G$ such that $a \bullet b = e$
- Think: Is $a \bullet b$ always equal to $b \bullet a$?
  - Read: Abelian Groups
- Think: Can there be different left and right identity elements?
- Think: Can there be different left and right inverses?

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have $a \bullet b \in G$
  - Associativity: For all $a, b, c \in G$, we have $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
  - Identity: There exists an element e such that for all $a \in G$, we have $e \bullet a = a$
  - Inverse: For every $a \in G$, there exists $b \in G$ such that $a \bullet b = e$
- Think: Is $a \bullet b$ always equal to $b \bullet a$?
  - Read: Abelian Groups
- Think: Can there be different left and right identity elements?
- Think: Can there be different left and right inverses?
- Example: $(\mathbb{Z}, +)$

Groups

- A group G is defined by a set of elements and an operation which maps two elements in the set to a third element
- $(G, \bullet)$ is a group if it satisfies the following conditions:
  - Closure: For all $a, b \in G$, we have $a \bullet b \in G$
  - Associativity: For all $a, b, c \in G$, we have $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
  - Identity: There exists an element e such that for all $a \in G$, we have $e \bullet a = a$
  - Inverse: For every $a \in G$, there exists $b \in G$ such that $a \bullet b = e$
- Think: Is $a \bullet b$ always equal to $b \bullet a$?
  - Read: Abelian Groups
- Think: Can there be different left and right identity elements?
- Think: Can there be different left and right inverses?
- Example: $(\mathbb{Z}, +)$
- Read: (Example) Symmetry Group

Cyclic Groups

- A group $(G, \cdot)$ is a cyclic group if it is generated by a single element

Cyclic Groups

- A group $(G, \cdot)$ is a cyclic group if it is generated by a single element
- That is: $G = \left\{1 = e = g^0, g^1, \dots, g^{n-1}\right\}$, where $|G| = n$

Cyclic Groups

- A group $(G, \cdot)$ is a cyclic group if it is generated by a single element
- That is: $G = \{1 = e = g^0, g^1, \ldots, g^{n-1}\}$, where $|G| = n$
- Written as: $G = \langle g \rangle$

Cyclic Groups

- A group $(G, \cdot)$ is a cyclic group if it is generated by a single element
- That is: $G = \{1 = e = g^0, g^1, \ldots, g^{n-1}\}$, where $|G| = n$
- Written as: $G = \langle g \rangle$
- Order of G: n

Discrete Logarithm Problem

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g

Discrete Logarithm Problem

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Given $(g, b = g^a)$, where $a \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$, it is hard to predict a

- Let G be a cyclic group $(G, \cdot)$ or order 2^n with generator g

Computational Diffie-Hellman Assumption

- Let G be a cyclic group $(G, \cdot)$ or order 2^n with generator g
- Give (g, g^a, g^b) to the adversary

Computational Diffie-Hellman Assumption

- Let G be a cyclic group $(G, \cdot)$ or order 2^n with generator g
- Give (g, g^a, g^b) to the adversary
- Hard to find g^{ab}

Decisional Diffie-Hellman Assumption

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g

Decisional Diffie-Hellman Assumption

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Pick $b \xleftarrow{\$} \{0, 1\}$

Decisional Diffie-Hellman Assumption

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Pick $b \xleftarrow{\$} \{0, 1\}$
- If $b = 0$, send (g, g^a, g^b, g^{ab}), where $a, b \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$

Decisional Diffie-Hellman Assumption

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Pick $b \xleftarrow{\$} \{0, 1\}$
- If $b = 0$, send (g, g^a, g^b, g^{ab}), where $a, b \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$
- If $b = 1$, send (g, g^a, g^b, g^r), where $a, b, r \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$

Decisional Diffie-Hellman Assumption

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Pick $b \xleftarrow{\$} \{0, 1\}$
- If $b = 0$, send (g, g^a, g^b, g^{ab}), where $a, b \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$
- If $b = 1$, send (g, g^a, g^b, g^r), where $a, b, r \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$
- Adversary has to guess b

Decisional Diffie-Hellman Assumption

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Pick $b \xleftarrow{\$} \{0, 1\}$
- If $b = 0$, send (g, g^a, g^b, g^{ab}), where $a, b \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$
- If $b = 1$, send (g, g^a, g^b, g^r), where $a, b, r \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$
- Adversary has to guess b
- Effectively: $(g, g^a, g^b, g^{ab}) \approx (g, g^a, g^b, g^r)$, for $a, b, r \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$ and any g

Relationship

Relationship

$$\mathsf{DDH} \implies \mathsf{CDH} \implies \mathsf{DL}$$

Key Agreement: Definition

- Alice picks a local randomness r_A

Key Agreement: Definition

- Alice picks a local randomness r_A
- Bob picks a local randomness r_B

Key Agreement: Definition

- Alice picks a local randomness r_A
- Bob picks a local randomness r_B
- Alice and Bob engage in a protocol and generate the transcript τ

Key Agreement: Definition

- Alice picks a local randomness r_A
- Bob picks a local randomness r_B
- Alice and Bob engage in a protocol and generate the transcript τ
- Alice's view $V_A = (r_A, \tau)$ and Bob's view $V_B = (r_B, \tau)$

Key Agreement: Definition

- Alice picks a local randomness r_A
- Bob picks a local randomness r_B
- Alice and Bob engage in a protocol and generate the transcript τ
- Alice's view $V_A = (r_A, \tau)$ and Bob's view $V_B = (r_B, \tau)$
- Eavesdropper's view $V_E = \tau$

Key Agreement: Definition

- Alice picks a local randomness r_A
- Bob picks a local randomness r_B
- Alice and Bob engage in a protocol and generate the transcript τ
- Alice's view $V_A = (r_A, \tau)$ and Bob's view $V_B = (r_B, \tau)$
- Eavesdropper's view $V_E = \tau$
- Alice outputs k_A as a function of V_A and Bob outputs k_B as a function of V_B

Key Agreement: Definition

- Alice picks a local randomness r_A
- Bob picks a local randomness r_B
- Alice and Bob engage in a protocol and generate the transcript τ
- Alice's view $V_A = (r_A, \tau)$ and Bob's view $V_B = (r_B, \tau)$
- Eavesdropper's view $V_E = \tau$
- Alice outputs k_A as a function of V_A and Bob outputs k_B as a function of V_B
- Correctness: $\Pr_{r_A, r_B}[k_A = k_B] \approx 1$

Key Agreement: Definition

- Alice picks a local randomness r_A
- Bob picks a local randomness r_B
- Alice and Bob engage in a protocol and generate the transcript τ
- Alice's view $V_A = (r_A, \tau)$ and Bob's view $V_B = (r_B, \tau)$
- Eavesdropper's view $V_E = \tau$
- Alice outputs k_A as a function of V_A and Bob outputs k_B as a function of V_B
- Correctness: $\Pr_{r_A, r_B}[k_A = k_B] \approx 1$
- Security: $(k_A, V_E) \equiv (k_B, \tau) \approx (r, \tau)$

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Alice picks $a \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$ and sends g^a to Bob

Key Agreement: Construction (Diffie-Hellman)

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Alice picks $a \xleftarrow{\$} \{0, \dots, 2^n - 1\}$ and sends g^a to Bob
- Bob picks $b \xleftarrow{\$} \{0, \dots, 2^n - 1\}$ and sends g^b to Alice

Key Agreement: Construction (Diffie-Hellman)

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Alice picks $a \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$ and sends g^a to Bob
- Bob picks $b \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$ and sends g^b to Alice
- Alice outputs $(g^b)^a$ and Bob outputs $(g^a)^b$

Key Agreement: Construction (Diffie-Hellman)

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Alice picks $a \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$ and sends g^a to Bob
- Bob picks $b \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$ and sends g^b to Alice
- Alice outputs $(g^b)^a$ and Bob outputs $(g^a)^b$
- Adversary sees: (g^a, g^b)

Key Agreement: Construction (Diffie-Hellman)

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Alice picks $a \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$ and sends g^a to Bob
- Bob picks $b \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$ and sends g^b to Alice
- Alice outputs $(g^b)^a$ and Bob outputs $(g^a)^b$
- Adversary sees: (g^a, g^b)
- Correctness?

Key Agreement: Construction (Diffie-Hellman)

- Let $(G, \cdot)$ be a cyclic group of order 2^n with generator g
- Alice picks $a \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$ and sends g^a to Bob
- Bob picks $b \xleftarrow{\$} \{0, \ldots, 2^n - 1\}$ and sends g^b to Alice
- Alice outputs $(g^b)^a$ and Bob outputs $(g^a)^b$
- Adversary sees: (g^a, g^b)
- Correctness?
- Security? Use DDH to say that g^{ab} is perfectly hidden from it

Public-key Encryption

- Key Generation: Alice generates $(sk, pk) \xleftarrow{\$} \mathsf{Gen}(1^n)$

Public-key Encryption

- Key Generation: Alice generates $(sk, pk) \xleftarrow{\$} \mathsf{Gen}(1^n)$
- Alice announces pk

Public-key Encryption

- Key Generation: Alice generates $(sk, pk) \xleftarrow{\$} \mathsf{Gen}(1^n)$
- Alice announces pk
- Encryption: Bob computes $c \xleftarrow{\$} \mathsf{Enc}(m, pk)$

Public-key Encryption

- Key Generation: Alice generates $(sk, pk) \xleftarrow{\$} \mathrm{Gen}(1^n)$
- Alice announces pk
- Encryption: Bob computes $c \xleftarrow{\$} \mathrm{Enc}(m, pk)$
- Correctness: Alice computes $m = \mathrm{Dec}(c, sk)$

Public-key Encryption

- Key Generation: Alice generates $(sk, pk) \xleftarrow{\$} \mathsf{Gen}(1^n)$
- Alice announces pk
- Encryption: Bob computes $c \xleftarrow{\$} \mathsf{Enc}(m, pk)$
- Correctness: Alice computes $m = \mathsf{Dec}(c, sk)$
- Security: Given (pk, c) the message seems uniformly random

- Use the key as a one-time pad

- Use the key as a one-time pad
- Formalize this intuition