

Review

- Symmetric Encryption Confidentiality
 - Definition: correctness and security
 - Block Ciphers: fixed plaintext/ciphertext size, use CBC for long plaintexts
 - AES: 3 layers Confusion and Diffusion concepts
 - Limitations: how to share a key?
- Hashing and MAC Integrity and Authentication
 - Definition: collision resistance, preimage resistant
 - HMAC: symmetric key authentication, can be built from hash functions
 - Applications: TOTP, password storing

Plan

- Asymmetric cryptography for Confidentiality and Authenticity (This Lecture)
 - Diffie-Helmann key exchange
 - RSA Encryption
 - El Gamal Encryption
 - Digital Signatures
 - Hand-in 1 assignment!
- Secure Channels Confidentiality and Authenticity in Practice (Lecture 5)
 - Authenticated Key Exchange
 - Public Key Infrastructures
 - The TLS Protocol
 - Vulnerabilities in TLS
- Consensus protocols (e.g. blockchain) Availability
 - Join my course next semester ©

Limitations of symmetric cryptography

• Sender and Receiver should meet in person and choose k

 Need a key for each pair of agents who want to communicate securely

 How to share a secret key securely between two agents over an insecure network?

Abelian Groups

- An abelian *group* is a pair (G, ∘) where G is a set and a binary operation ∘ defined on G such that:
 - (Closure) For all g, h∈G, g∘h is in G
 - There is an identity e∈G such that e∘g=g for g∈G
 - Every $g \in G$ has an inverse $h \in G$ such that $h \circ g = e$
 - (Associativity) For all f, g, $h \in G$, $f \circ (g \circ h) = (f \circ g) \circ h$
 - Commutativity For all g, $h \in G$, $g \circ h = h \circ g$
- The order of a finite group G is the number of elements in G

Group Operation

- The group operation can be written *multiplicatively*
 - I.e., instead of g∘h, write g · h=gh
 - Does not mean that the group operation corresponds to (integer) addition or multiplication
- Identity denoted by 1 or g⁰
- Inverse of group element g denoted by g-1
- Group exponentiation: a^m, applying operation m times to element a

Cyclic Groups

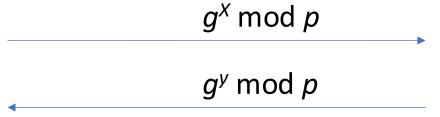
- Let (G, \cdot) be a finite group of order q (written multiplicatively).
- Let g be some element of G.
- Consider the set $< g > = \{g^0, g^1, ...\}.$
- We know $g^q = 1 = g^0$, (Fermat's little theorem) so the set has $\leq q$ elements.
- If the set $\langle g \rangle$ has q elements (all elements in the group), then we say g is a generator of (G, \cdot) .
- A generator g "generates" all elements in the group when the group operation is applied to itself multiple times.
- If a group has a generator, then we say this is a cyclic group.

Diffie-Hellman Key Exchange

Using the group $(\mathbb{Z}_{p_i}^* \cdot)$ with generator g_i , where p is prime

Alice

- 1. choose a random $x \in \mathbb{Z}_p^*$
- 2. compute $g^x \mod p$



3. compute $(g^y)^x \mod p$

Bob

- 1. choose a random $y \in \mathbb{Z}_p^*$
- 2. compute $g^y \mod p$

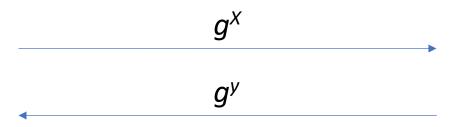
3. compute $(g^x)^y \mod p$

Diffie-Hellman Key Exchange

Using a generic group (G, \cdot) of order q with generator g

Alice

- 1. choose a random $x \in \mathbb{Z}_q^*$
- 2. compute g^x



3. compute $(g^y)^x$

Bob

- 1. choose a random $y \in \mathbb{Z}_q^*$
- 2. compute g^y

- 3. compute $(g^x)^y$
- Diffie-Hellman key exchange is secure under the Computational Diffie-Hellman assumption
- Given g^x it is hard to find x (DL problem)
- Given g^x and g^y it is hard to find g^{xy} (Computational DH)
- Given g^x , g^y , g^{xy} , and a random z it is hard to distinguish g^{xy} from z (Decisional DH)

Diffie-Hellman Key Exchange Question

Select the correct alternative about Diffie-Hellman key exchange:

- a) Diffie-Hellman key exchange makes it infeasible for a Dolev-Yao adversary to impersonate a user.
- b) Diffie-Hellman key exchange makes it infeasible for an eavesdropper to learn the key exchanged between two users.
- c) Diffie-Hellman key exchange is secure under the computational assumption that factoring large integers is hard.
- d) Diffie-Hellman key exchanged is an encryption protocol for sending arbitrary confidential messages through an insecure network.

Asymmetric Cryptography confidentiality and authentication



Source: pinterest

Asymmetric Encryption

Concept

- Every *secret* key sk has an **inverse** *public* key denoted as *pk*
- It is hard to compute sk from public information (including pk)
- The public key can be used to create a ciphertext by encrypting a plaintext.
 The secret key can be used to decrypt the ciphertext back to plaintext.

Goals

- Guarantee message confidentiality
- Provide provable security under well studied mathematical assumptions, i.e. confidentiality is guaranteed as long as mathematical problem is "hard"

Asymmetric Encryption

Syntax

- Key Generation: outputs pair of secret and public keys (sk,pk)
- Encryption $\mathcal{E}(m,pk)$: Outputs ciphertext c given a message m and public key pk
- Decryption $\mathcal{D}(\mathcal{E}(m,pk),sk)$: Outputs plaintext message m given a ciphertext c and secret key sk

Properties

- Correctness: $\mathcal{D}(\mathcal{E}(m,pk),sk) = m$
- Security: if sk' is not the secret key corresponding to pk, $\mathcal{D}(\mathcal{E}(m,pk),sk')$ reveals no information about m
- Examples:
 - RSA, ElGamal...
 - Typical key lenght: 256 (El Gamal based on Elliptic Curves) to 4096 (RSA) bits
 - Slower than symmetric crypto

Asymmetric Encryption Question

Select the correct alternative about Asymmetric Encryption:

- a) The same key k is used to both encrypt and decrypt messages.
- b) Asymmetric Encryption guarantees that an encrypted message was sent by a given party, i.e. protects message authenticity.
- c) Asymmetric Encryption guarantees that an encrypted message was not modified, i.e. protects message integrity.
- d) Given a key pair (sk,pk) and a ciphertext $\mathcal{E}(m,pk)$ =c, an adversary who does not know sk learns no information about m.

El Gamal

Alice

- 1. choose a random $sk \in \mathbb{Z}^*$
- 2. compute $pk=g^{sk}$

$$pk=g^{sk}$$

$$C_1=g^r, c_2=g^{sk\cdot r}\cdot m$$

3. compute
$$\frac{c_2}{c_1^{sk}} = \frac{c_2}{g^{rsk}} = \frac{g^{sk \cdot r} \cdot m}{g^{sk \cdot r}} = m$$

Using a generic group (G, \cdot) of order q with generator g

Bob

- 1. choose a random $r \in \mathbb{Z}_q^*$
- 2. compute $c_1 = g^r$
- 3. compute $(g^{sk})^r$
- 4. compute $c_2 = g^{sk \cdot r} \cdot m$

El Gamal

Key Generation

- 1. choose a random $sk \in \mathbb{Z}_q^*$
- 2. compute $pk=g^{sk}$

Encryption

- 1. choose a random $r \in \mathbb{Z}_q^*$
- 2. compute $c_1 = g^r$
- 3. compute $c_2 = m \cdot pk^r = m \cdot (g^{sk})^r$
- 4. Output $c=(c_1,c_2)$

Decryption

- 1. Compute $\frac{c}{g^{rsk}} = \frac{g^{sk \cdot r} \cdot m}{g^{sk \cdot r}} = m$
- 2. Output m

Using a generic group (G, \cdot) of order q with generator g

- Given g^x, g^y, g^{xy}, and a random z
 it is hard to distinguish g^{xy} from z
 (Decisional DH)
- El Gamal encryption is secure under the Decisional DH assumption.

El Gamal Question

Select the correct option about El Gamal Encryption

- a) If Diffie-Hellman key exchange is secure, then El Gamal encryption is also guaranteed to be secure.
- b) Encrypting message with El Gamal encryption requires no randomness.
- c) El Gamal encryption ensures confidentiality when constructed over any group where the DDH (i.e. Decisional DH) assumption holds.
- d) The secret key sk can be used to verify which user sent a ciphertext under the corresponding public key $pk=g^{sk}$

Digital Signature

- Goal: integrity + authenticity
 - No message secrecy
- Based on asymmetric crypto
 - Secret/signing key to create the signature
 - Public/verification key to verify the signature
- An algorithm $\sigma = \text{sign}(m,sk)$
- An algorithm $d = ver(\sigma, m, vk)$
- (sk,vk) = KeyGen()
- Correctness: ver(sign(m,sk),m,vk) = true

Digital Signature

Why don't just use MAC?

El Gamal Signatures

Key Generation

- 1. choose a random $sk \in \mathbb{Z}_p^*$
- 2. compute $pk=g^{sk} \mod p$

Signature

- 1. choose a random $k \in \mathbb{Z}_p^*$ such that k is relatively prime to p-1
- 2. compute $r=g^k \mod p$
- 3. compute $s = (H(m)-sk \cdot r)k^{-1} \mod (p-1)$, if s=0, go to step 1
- 4. Output (r,s)

Verification

- 1. Check that 0<r<p and 0<s<p-1
- 2. Check that $g^{H(m)} = pk^r r^s$
- 3. Output 1 if and only if all checks pass, output 0 otherwise.

Using the group ($\mathbb{Z}_{p,}^*$) with generator g, where p is prime and a cryptographic hash function H()

Schnorr Signatures

Key Generation

- 1. choose a random $sk \in \mathbb{Z}_q^*$
- 2. compute $pk=g^{sk}$

Signature

- 1. choose a random $k \in \mathbb{Z}_q^*$ and compute $r=g^k$
- 2. compute $e = H(r \mid | m)$
- 3. compute $s = k+sk \cdot e$
- 4. Output (s,e)

Verification

- 1. Compute r'=gspke
- 2. Compute e'=H(r' || m)
- 3. Output 1 if and only if e'=e, output 0 otherwise.

Using a generic group (G, \cdot) of order q with generator g and a cryptographic hash function H()

Digital Signature Question

Select the correct alternative about Digital Signatures

- a) A digital signature scheme protects message confidentiality
- b) A user who only knows sk cannot sign a message.
- c) Digital signature schemes protect message integrity and authenticity.
- d) Digital signatures have the same function as Message Authentication Codes (MAC).

Summary

- Diffie-Helmann Key Exchange
 - Protocol: computational hardness
 - Issues: man-in-the-middle attacks
- Asymmetric Cryptography
 - Definition: correctness and security
 - El-gamal: based on DH problem
 - Digital Signature: goals