Cryptography

Lecture 21

Arkady Yerukhimovich

November 11, 2024

Announcements

- Homework 6 is out – Due Before class on Monday, Nov. 18
- Research project videos are due on Friday, Nov. 22.
- Final exam – 12:40-2:40 on Monday, Dec. 16.

Outline

1. Lecture 20 Review
2. Public-Key Encryption (Chapters 11.1, 11.2, 11.4)

Lecture 20 Review

- Private-key crypto from number-theoretic assumptions
- Public-key revolution
- Diffie-Hellman Key Exchange

Going Beyond Key Exchange

	Private-Key	Public-Key
Secrecy	Private-key encryption	Public-key encryption
Integrity	MACs	Digital signatures

Going Beyond Key Exchange

	Private-Key	Public-Key
Secrecy	Private-key encryption	Public-key encryption
Integrity	MACs	Digital signatures

Public-Key Encryption

- User A has keys (pk_A, sk_A)
- Public key pk_A is used to encrypt messages to A
- Secret key sk_A is used by A to decrypt
- A publishes pk_A while keeping sk_A secret
- Anybody can encrypt, only A can decrypt

Going Beyond Key Exchange

	Private-Key	Public-Key
Secrecy	Private-key encryption	Public-key encryption
Integrity	MACs	Digital signatures

Public-Key Encryption

- User A has keys (pk_A, sk_A)
- Public key pk_A is used to encrypt messages to A
- Secret key sk_A is used by A to decrypt
- A publishes pk_A while keeping sk_A secret
- Anybody can encrypt, only A can decrypt

Digital signatures

- A has keys (pk_A, sk_A)
- Secret key sk_A is used by A to sign messages
- Public key pk_A is used to verify A's signatures
- A publishes pk_A while keeping sk_A secret
- Only A can sign, anybody can verify

Outline

1. Lecture 20 Review

2. Public-Key Encryption (Chapters 11.1, 11.2, 11.4)

Public-Key Encryption

Public-key (asymmetric-key) encryption scheme:

- $\mathsf{Gen} : (pk, sk) \leftarrow \mathsf{Gen}(1^n)$ – generates a public key and a secret key
- $\mathsf{Enc}_{pk}(m) : c \leftarrow \mathsf{Enc}_{pk}(m)$ for message m
- $\mathsf{Dec}_{sk}(c) : m = \mathsf{Dec}_{sk}(c)$

Public-Key Encryption

Public-key (asymmetric-key) encryption scheme:

- Gen : $(pk, sk) \leftarrow \text{Gen}(1^n)$ – generates a public key and a secret key
- $\text{Enc}_{pk}(m)$: $c \leftarrow \text{Enc}_{pk}(m)$ for message m
- $\text{Dec}_{sk}(c)$: $m = \text{Dec}_{sk}(c)$

Correctness

For all n, for all pk, sk output by $\text{Gen}(1^n)$ and all messages m, $\text{Dec}_{sk}(\text{Enc}_{pk}(m)) = m$.

Public-Key Encryption

Public-key (asymmetric-key) encryption scheme:

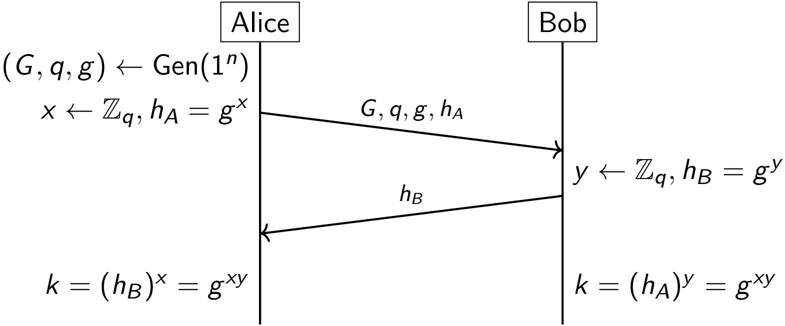
- Gen : $(pk, sk) \leftarrow \text{Gen}(1^n)$ – generates a public key and a secret key
- $\text{Enc}_{pk}(m)$: $c \leftarrow \text{Enc}_{pk}(m)$ for message m
- $\text{Dec}_{sk}(c)$: $m = \text{Dec}_{sk}(c)$

Correctness

For all n, for all pk, sk output by $\text{Gen}(1^n)$ and all messages m, $\text{Dec}_{sk}(\text{Enc}_{pk}(m)) = m$.

Observations:

- pk can be published on public bulletin board, enables anyone to encrypt
- sk must be kept secret, allows only recipient to decrypt.



Observation

At the end of Π, A and B share a key g^{xy} that is indistinguishable from a random group element

A Technical Lemma

Lemma

Let G be a finite group, for any element $x \in G$

$$\forall y \in G, \Pr_{k \leftarrow G}[k \cdot x = y] = \frac{1}{|G|}$$

A Technical Lemma

Lemma

Let G be a finite group, for any element $x \in G$

$$\forall y \in G, \Pr_{k \leftarrow G}[k \cdot x = y] = \frac{1}{|G|}$$

Proof:

$$\Pr_{k \leftarrow G}[k \cdot x = y] = \Pr_{k \leftarrow G}[k = y \cdot x^{-1}] = \frac{1}{|G|}$$

A Technical Lemma

Lemma

Let G be a finite group, for any element $x \in G$

$$\forall y \in G, \Pr_{k \leftarrow G}[k \cdot x = y] = \frac{1}{|G|}$$

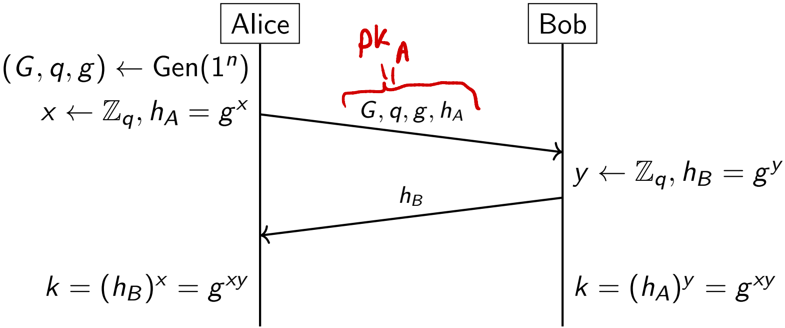
Proof:

$$\Pr_{k \leftarrow G}[k \cdot x = y] = \Pr_{k \leftarrow G}[k = y \cdot x^{-1}] = \frac{1}{|G|}$$

Takeaways

- For a random key k, the value $k \cdot x$ is equally likely to be any group element y
- This functions as a multiplicative OTP.

Diffie-Hellman Key Exchange



Observation

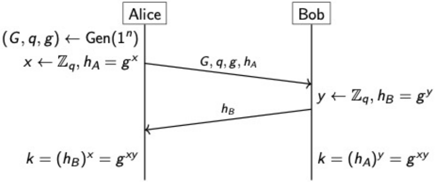
At the end of Π, A and B share a key g^{xy} that is indistinguishable from a random group element

From Key Exchange to Public-Key Encryption

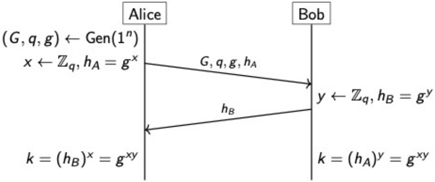
We will now convert DH KE into a public-key encryption scheme:

1. Recall that as a result of DH key exchange, Alice and Bob both output a random-looking group element g^{xy}. Assuming that $m \in G$, how can you use this shared key to "encrypt" m?
2. The DH key exchange protocol is interactive, while we want a public-key encryption scheme to be non-interactive. How can Alice use the first message of DH key exchange to produce a public key?
3. How can Bob use this public-key to encrypt a message to Alice? (Hint: Remember that encryption must be randomized).
4. How can Alice decrypt?

From Key Exchange to Public-Key Encryption

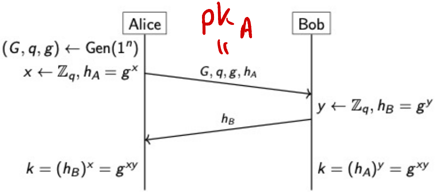


From Key Exchange to Public-Key Encryption



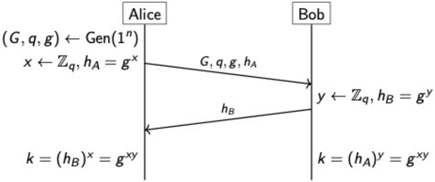
1. To encrypt $m \in G$ using g^{xy}, compute $m \cdot g^{xy}$.
   This is essentially a multiplicative OTP

From Key Exchange to Public-Key Encryption



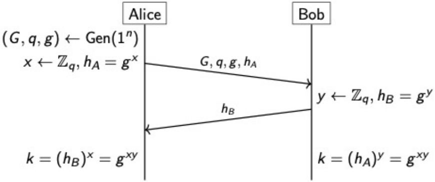
1. To encrypt $m \in G$ using g^{xy}, compute $m \cdot g^{xy}$.
   This is essentially a multiplicative OTP
2. To make encryption non-interactive, A sets pk to be her first message $pk_A = (G, q, g, h_A)$

From Key Exchange to Public-Key Encryption



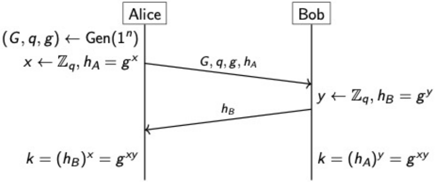
1. To encrypt $m \in G$ using g^{xy}, compute $m \cdot g^{xy}$.
   This is essentially a multiplicative OTP
2. To make encryption non-interactive, A sets pk to be her first message $pk_A = (G, q, g, h_A)$
3. To encrypt a message m, B has to complete the KE and use the resulting key to encrypt

From Key Exchange to Public-Key Encryption



1. To encrypt $m \in G$ using g^{xy}, compute $m \cdot g^{xy}$.
   This is essentially a multiplicative OTP
2. To make encryption non-interactive, A sets pk to be her first message $pk_A = (G, q, g, h_A)$
3. To encrypt a message m, B has to complete the KE and use the resulting key to encrypt
   - Choose $y \leftarrow \mathbb{Z}_q$
   - Compute $g^{xy} \cdot m$

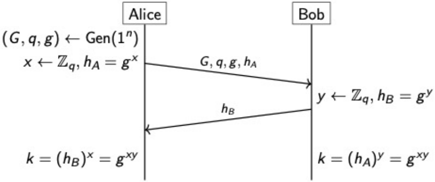
From Key Exchange to Public-Key Encryption



1. To encrypt $m \in G$ using g^{xy}, compute $m \cdot g^{xy}$.
   This is essentially a multiplicative OTP
2. To make encryption non-interactive, A sets pk to be her first message $pk_A = (G, q, g, h_A)$
3. To encrypt a message m, B has to complete the KE and use the resulting key to encrypt
   - Choose $y \leftarrow \mathbb{Z}_q$
   - Compute $g^{xy} \cdot m$
   - To enable A to decrypt, include B's message $h_B = g^y$ in c

$c = (g^y, g^{xy} \cdot m)$

From Key Exchange to Public-Key Encryption



1. To encrypt $m \in G$ using g^{xy}, compute $m \cdot g^{xy}$.
   This is essentially a multiplicative OTP
2. To make encryption non-interactive, A sets pk to be her first message $pk_A = (G, q, g, h_A)$
3. To encrypt a message m, B has to complete the KE and use the resulting key to encrypt
   - Choose $y \leftarrow \mathbb{Z}_q$
   - Compute $g^{xy} \cdot m$
   - To enable A to decrypt, include B's message $h_B = g^y$ in c
4. To decrypt, A computes $(h_B)^x = g^{xy}$ and uses this to unmask m