האוניברסיטה העברית בירושלים
The Hebrew University of Jerusalem

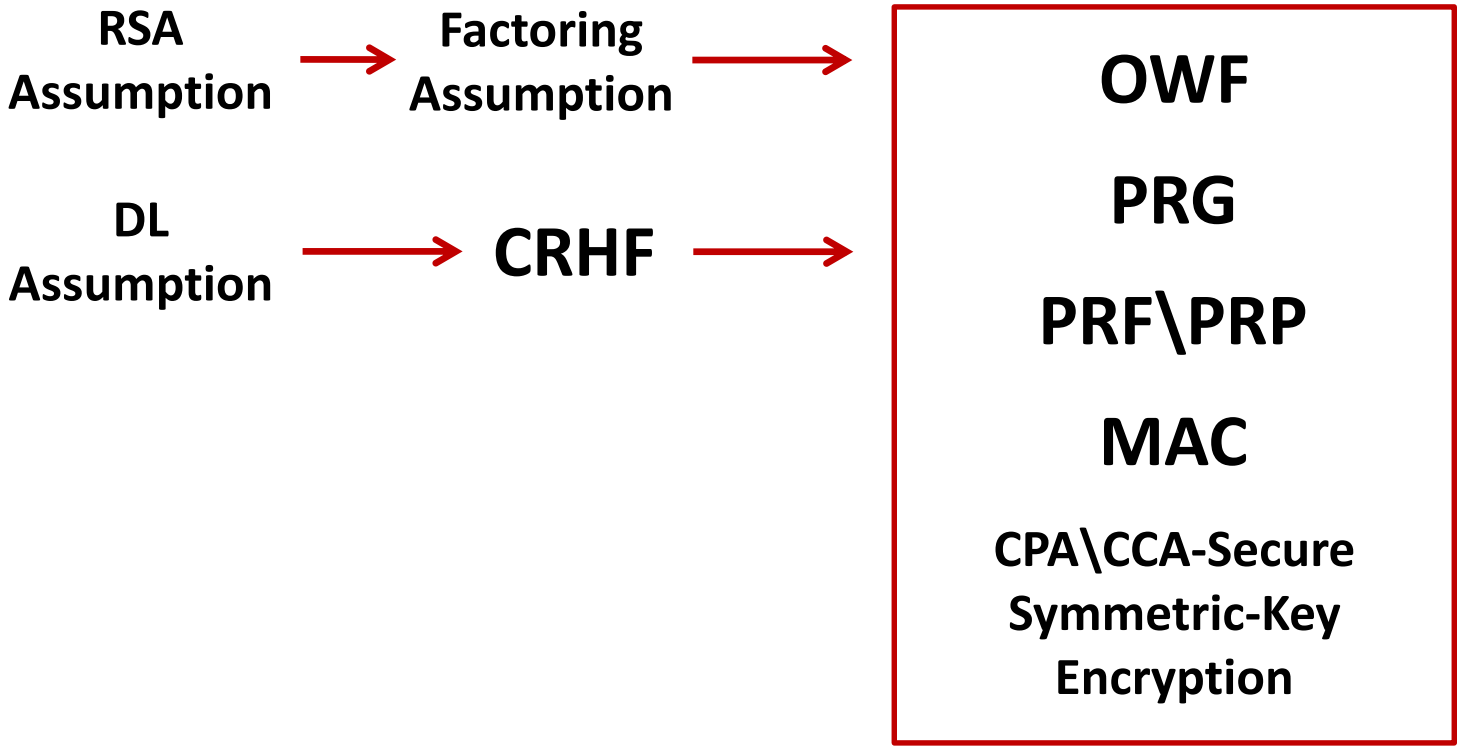
Cryptography

Lecture 8:

Public-Key Cryptography

Gil Segev

The World of Crypto Primitives



Outline

- **Limitations of private-key cryptography**
- **Public-key cryptography**
- **Key-agreement protocols**
  - The Diffie-Hellman protocol

The Limitations of Private-Key Crypto

- Given **shared secret keys** it is possible to securely communicate over an insecure channel
- How do we obtain **shared secret keys**? Cannot be sent over an insecure channel…

Problem 1: Key distribution

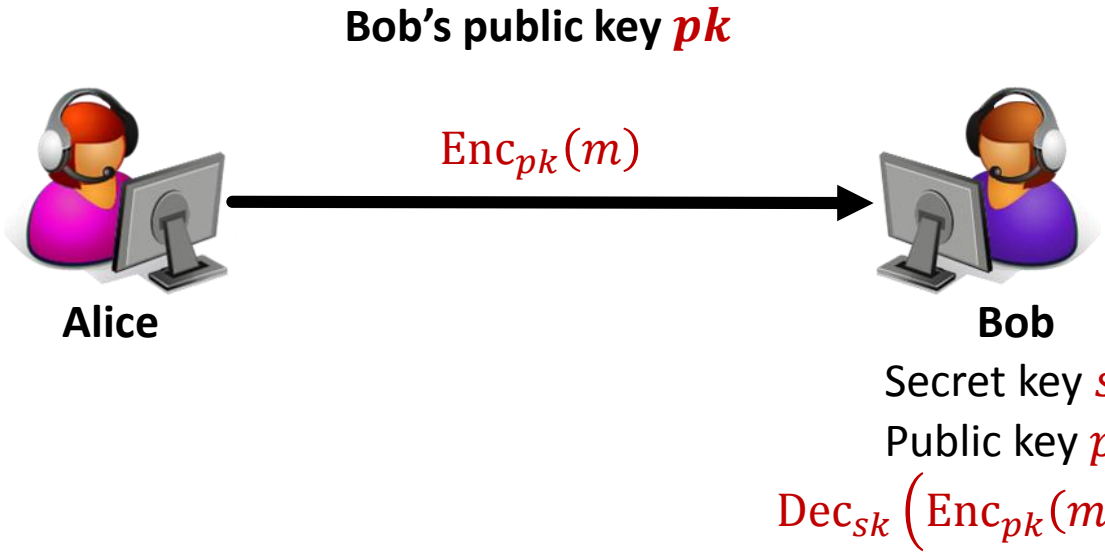
- Physical meeting, trusted messengers, key-distribution centers, etc.
- How can I share a key with Amazon?

Problem 2: Key storage

- n users require $\binom{n}{2} \approx n^2$ keys
- Each user needs to store $n - 1$ keys
- Space is expensive as keys must be stored in a secure fashion

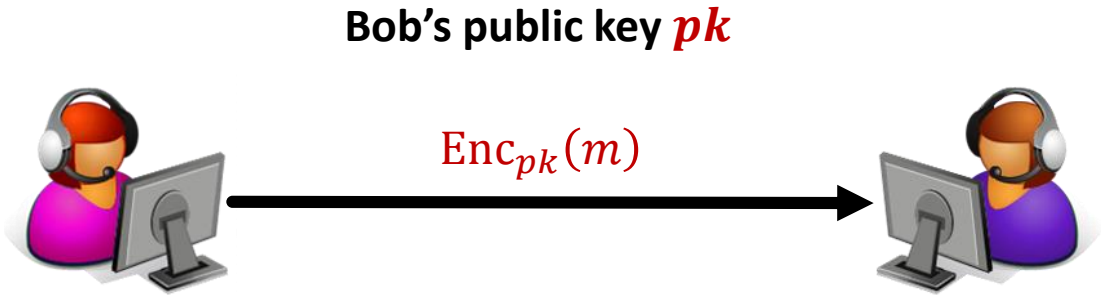
“New Directions in Cryptography”
[Diffie & Hellman ‘76]

- A radical change, introducing the idea of **public-key cryptography**
- One of the first steps towards moving cryptography out of the private domain (e.g., intelligence and military organizations) and into the public one



“New Directions in Cryptography”
[Diffie & Hellman ‘76]

- **Key distribution:** Bob can publically post his public key
- **Key storage:** Bob only needs to store his own secret key



“New Directions in Cryptography”
[Diffie & Hellman ‘76]

Diffie & Hellman envisioned three public-key primitives:

- Public-key encryption
- Digital signatures
- Key-agreement protocols

They invented the first key-agreement protocol

- Known as the Diffie-Hellman key-agreement protocol
- The first public-key encryption and digital signature schemes were invented a year later by Rivest, Shamir and Adleman

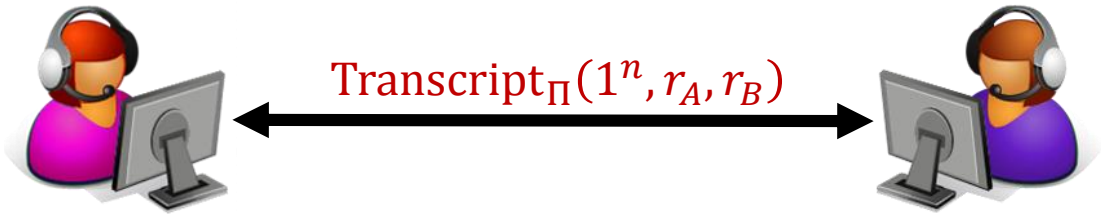
Key-Agreement Protocols

- Alice and Bob run a protocol Π for generating a random key
- r_A and r_B are the random string of Alice and Bob
- $\text{Transcript}_{\Pi}(1^n, r_A, r_B)$ is the transcript of the protocol

Definition (Correctness):

Π is a **key-agreement protocol** if there exists a negligible function $\nu(n)$ such that for all $n \in \mathbb{N}$:

$$\Pr_{r_A, r_B}[K_A(1^n, r_A, r_B) \neq K_B(1^n, r_A, r_B)] \leq \nu(n)$$



Alice

Input: 1^n, r_A

Output: $K_A(1^n, r_A, r_B) \in \mathcal{K}_n$

Bob

Input: 1^n, r_B

Output: $K_B(1^n, r_A, r_B) \in \mathcal{K}_n$

Key-Agreement Protocols

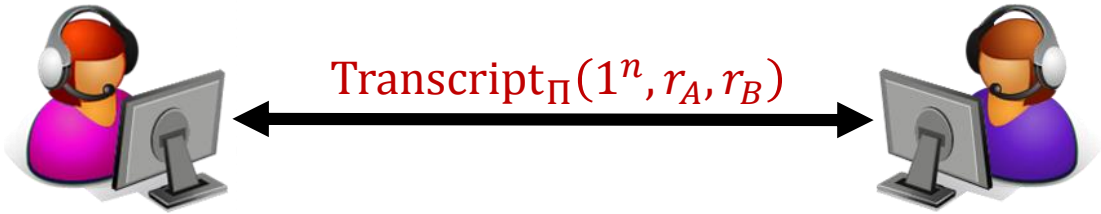
- Eve is eavesdropping the communication channel, and should not learn any information on the resulting key
- Specifically: From Eve's point of view, the key should be "as good as" an **independently chosen** key

Definition (Security):

A key-agreement protocol Π is **secure** if

$$\left(\text{Transcript}_{\Pi}(1^n, r_A, r_B), K_A(1^n, r_A, r_B)\right) \approx^c \left(\text{Transcript}_{\Pi}(1^n, r_A, r_B), K\right)$$

where $r_A, r_B \leftarrow \{0,1\}^*$ and $K \leftarrow \mathcal{K}_n$ are sampled independently and uniformly



Alice

Input: 1^n, r_A

Output: $K_A(1^n, r_A, r_B) \in \mathcal{K}_n$

Bob

Input: 1^n, r_B

Output: $K_B(1^n, r_A, r_B) \in \mathcal{K}_n$

Recall: Computational Indistinguishability

- Two probability distributions are computationally indistinguishable if no efficient algorithm can tell them apart

Definition:
Two probability ensembles $X = \{X_n\}_{n\in\mathbb{N}}$ and $Y = \{Y_n\}_{n\in\mathbb{N}}$ are **computationally indistinguishable** if for all PPT distinguishers $\mathcal{D}$ there exists a negligible function $\nu(\cdot)$ such that

$$|\Pr[\mathcal{D}(1^n, x) = 1] - \Pr[\mathcal{D}(1^n, y) = 1]| \leq \nu(n)$$

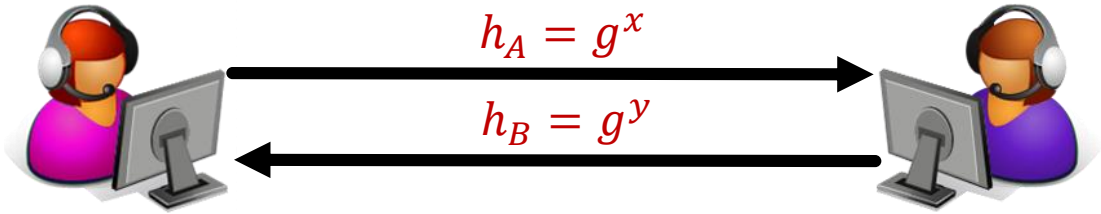
where $x \leftarrow X_n$ and $y \leftarrow Y_n$.

- Denoted $X \approx^c Y$ (or $X \equiv^c Y$)
- Typically consider **efficiently samplable** X and Y
- Pseudorandom = computationally indistinguishable from uniform

Diffie-Hellman Key-Agreement

- Let $\mathcal{G}$ be a PPT algorithm that on input 1^n outputs $(\mathbb{G}, q, g)$, where $\mathbb{G}$ is a cyclic group of order q that is generated by g, and q is an n-bit prime
- Assume that $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ is generated and known to both parties

Otherwise can be generated by Alice and sent to Bob



Sample $x \leftarrow \mathbb{Z}_q$

Output $K_A = (h_B)^x$

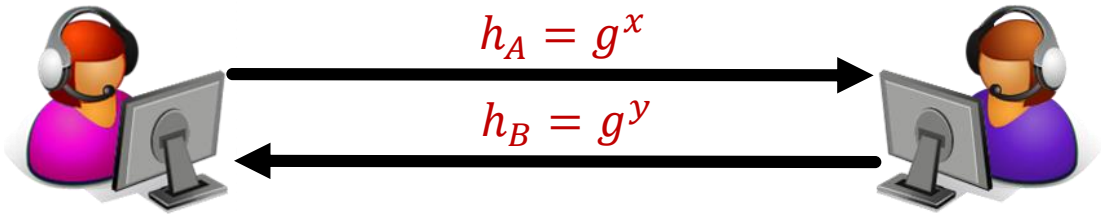
Sample $y \leftarrow \mathbb{Z}_q$

Output $K_B = (h_A)^y$

Diffie-Hellman Key-Agreement

- **Correctness:**

$$K_A = (h_B)^x = (g^y)^x = (g^x)^y = (h_A)^y = K_B$$



Sample $x \leftarrow \mathbb{Z}_q$

Output $K_A = (h_B)^x$

Sample $y \leftarrow \mathbb{Z}_q$

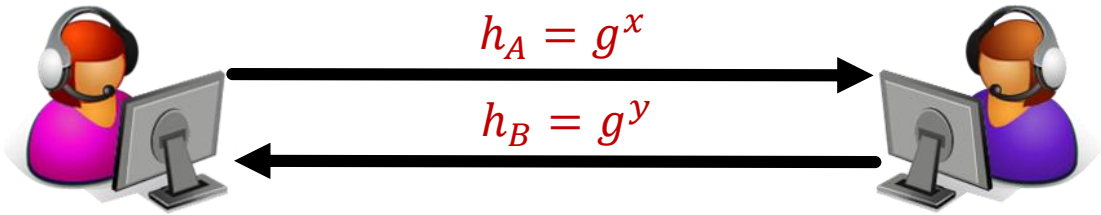
Output $K_B = (h_A)^y$

Diffie-Hellman Key-Agreement

- **Security:**
  Does $\left(\text{Transcript}_{\Pi}(1^n, r_A, r_B), K_A(1^n, r_A, r_B)\right) \approx^c (\text{Transcript}_{\Pi}(1^n, r_A, r_B), K)$?

(g^x, g^y, g^{xy})

(g^x, g^y, g^z)



Alice

Sample $x \leftarrow \mathbb{Z}_q$

Output $K_A = (h_B)^x$

Bob

Sample $y \leftarrow \mathbb{Z}_q$

Output $K_B = (h_A)^y$

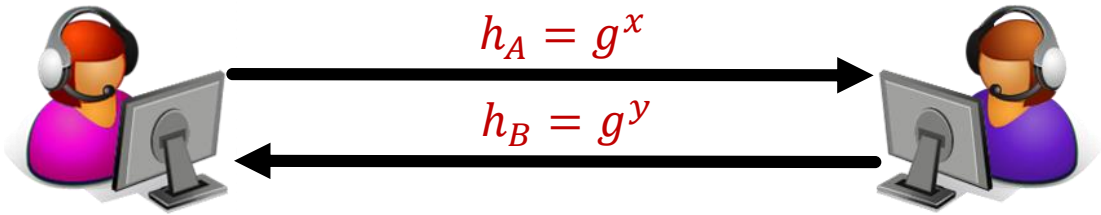
Diffie-Hellman Key-Agreement

- **Security:**
  Does $\left(\text{Transcript}_{\Pi}(1^n, r_A, r_B), K_A(1^n, r_A, r_B)\right) \approx^c (\text{Transcript}_{\Pi}(1^n, r_A, r_B), K)$?

The Decisional Diffie-Hellman (DDH) Assumption:
It holds that

$$(\mathbb{G}, q, g, g^x, g^y, g^{xy}) \approx^c (\mathbb{G}, q, g, g^x, g^y, g^z)$$

where $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$, and $x, y, z \leftarrow \mathbb{Z}_q$.



Sample $x \leftarrow \mathbb{Z}_q$
Output $K_A = (h_B)^x$

Sample $y \leftarrow \mathbb{Z}_q$
Output $K_B = (h_A)^y$

The Diffie-Hellman Assumptions

The Decisional Diffie-Hellman (DDH) Assumption:
For every PPT algorithm $\mathcal{A}$ there exists a negligible function $\nu(\cdot)$ such that

$$|\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1]| \leq \nu(n)$$

where $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$, and $x, y, z \leftarrow \mathbb{Z}_q$.

The Computational Diffie-Hellman (CDH) Assumption:
For every PPT algorithm $\mathcal{A}$ there exists a negligible function $\nu(\cdot)$ such that

$$|\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y) = g^{xy}]| \leq \nu(n)$$

where $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$, and $x, y \leftarrow \mathbb{Z}_q$.

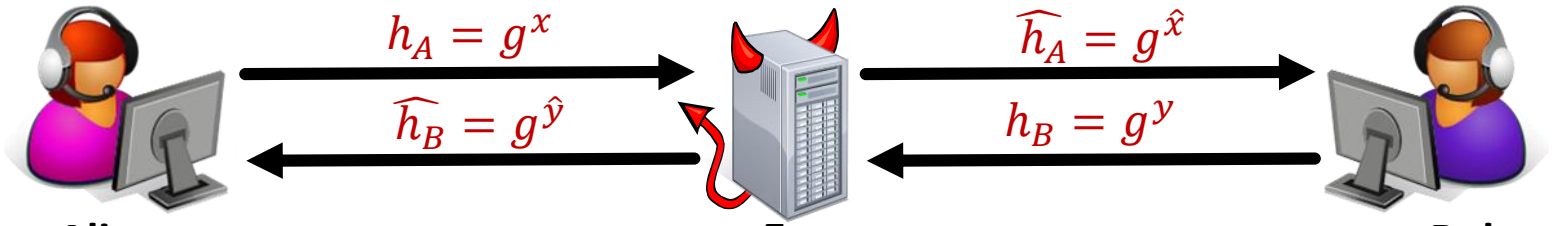
Diffie-Hellman Key-Agreement

Random elements vs. random strings

- Alice and Bob agree on a random group element $g^{xy} \in \mathbb{G}$
- Typically need a random n-bit key $K \in \{0,1\}^n$
- There are generic tools to extract such a key ("randomness extractors")

Insecurity against active adversaries ("man-in-the-middle" attacks)

- Alice and Bob end up with $K_A \neq K_B$, Eve knows both K_A and K_B
- In practice: Authenticated variants of the Diffie-Hellman protocol



Sample $x \leftarrow \mathbb{Z}_q$

Output $K_A = \left(\widehat{h_B}\right)^x$

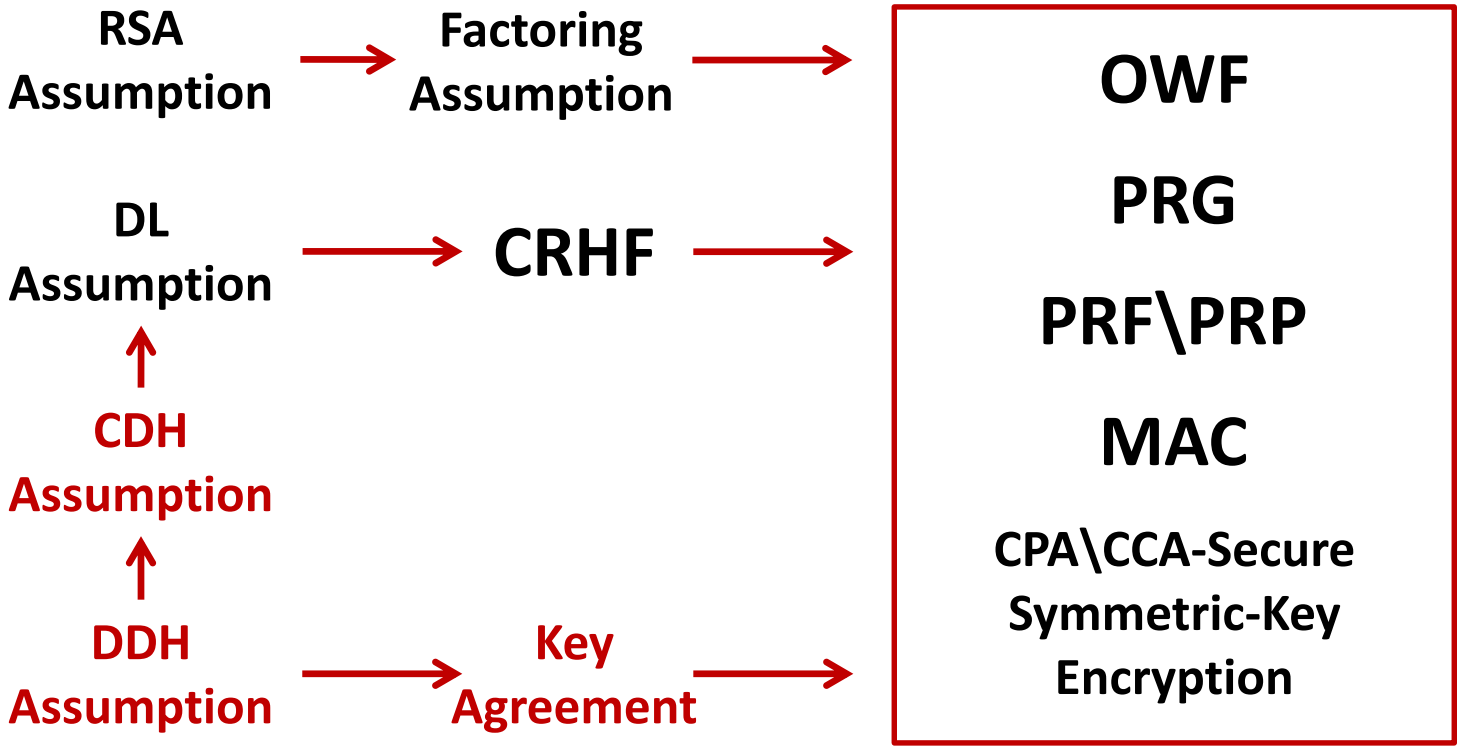
Sample $\hat{x}, \hat{y} \leftarrow \mathbb{Z}_q$

$K_A = (h_A)^{\hat{y}}$ and $K_B = (h_B)^{\hat{x}}$

Sample $y \leftarrow \mathbb{Z}_q$

Output $K_B = \left(\widehat{h_A}\right)^y$

The World of Crypto Primitives



Recommended Reading

- J. Katz and Y. Lindell. **Introduction to Modern Cryptography**.
  Chapter 10 (Key Management and the Public-Key Revolution)