Key Exchange
and the Public Key Revolution

Slides by Prof. Jonathan Katz.
Lightly edited by me.
Private-key cryptography

• Private-key cryptography allows two users who *share a secret key* to establish a “secure channel”

• The need to share a secret key has several drawbacks...
The key-distribution problem

• *How do users share a key in the first place?*
  • Need to share the key using a secure channel...

• This problem can be solved in some settings
  • E.g., physical proximity, trusted courier, ...
  • Note: this does not make private-key cryptography useless!

• Can be difficult, expensive, or impossible to solve in other settings
The key-management problem

• Imagine an organization with N employees, where each pair of employees might need to communicate securely

• Solution using private-key cryptography:
  • Each user shares a key with all other users
    ⇒ Each user must store/manage N-1 secret keys!
    ⇒ O(N²) keys overall!
Key Distribution Centers

Drawbacks:

• Single point of failure.
  • For liveness. Could duplicate, but...
  • For security! Internal and external.

• Cannot support “open systems”.
  • What if Alice and Bob do not work for the same entity, or trust the same person?
  • E.g. sending credit card information to a merchant.
“Classical” cryptography offers no solution to these problems!
New directions...

• Main ideas:
  • Some problems exhibit *asymmetry* – easy to compute, but hard to invert (factoring, RSA, group exponentiation, …)
  • Use this asymmetry to enable two parties to agree on a shared secret key via public discussion(!)
    • *Key exchange*
Key exchange

Secure against an eavesdropper who sees everything!
More formally...

$k \in \{0,1\}^n$

Security goal: even after observing the transcript, the shared key $k$ should be indistinguishable from a uniform key.
Notes

• Being unable to *compute* the key given the transcript is not a strong enough guarantee

• Indistinguishability of the shared key from uniform is a much stronger guarantee...
  • ...and is necessary if the shared key will subsequently be used for private-key crypto!
Diffie-Hellman key exchange

\[ k_1 = (h_2)^x \]

\[ k_2 = (h_1)^y \]

\[ (G, q, g) \leftarrow G(1^n) \]

\[ x \leftarrow \mathbb{Z}_q \]

\[ h_1 = g^x \]

\[ h_2 = g^y \]
In practice...

\[ k_1 = (h_2)^x \]

\[ h_1 = g^x \]

\[ k_2 = (h_1)^y \]

\[ h_2 = g^y \]

G, q, g
Recall...

• *Decisional Diffie-Hellman (DDH) assumption:*
  • Given $g^x$, $g^y$, cannot distinguish $g^{xy}$ from a uniform group element
Security?

• Eavesdropper sees $G$, $q$, $g$, $g^x$, $g^y$
• Shared key $k$ is $g^{xy}$

• Computing $k$ from the transcript is exactly the *computational* Diffie-Hellman problem

• Distinguishing $k$ from a uniform group element is exactly the *decisional* Diffie-Hellman problem
  \[ \Rightarrow \text{If the DDH problem is hard relative to } G, \text{ this is a secure key-exchange protocol!} \]
A subtlety

- We want our key-exchange protocol to give us a uniform(-looking) key \( k \in \{0,1\}^n \)
- Instead we have a uniform(-looking) group element \( k \in G \)
  - Not clear how to use this as, e.g., an AES key

- Solution: *key derivation*
  - Set \( k' = H(k) \) for suitable hash function \( H \)
    - Secure if \( H \) is modeled as a random oracle
Modern key-exchange protocols

• Security against passive eavesdroppers is insufficient
• Generally want \textit{authenticated} key exchange
  • This requires some form of setup in advance

• Modern key-exchange protocols provide this
  • We will return to this later