

# Security of Key Exchange

---

Vitaly Shmatikov

# Universal Composability

[Canetti]

- 
- ◆ Very general framework dealing with protocols in “real-life” settings
    - Asynchronous, unreliable, unauthenticated communication
    - Variable (even unbounded) number of participants
    - Variable identities
  - ◆ Models cryptographic primitives
    - Encryption, digital signatures, etc.
  - ◆ Deals with adaptive break-ins to honest parties
  - ◆ Deals with concurrent composition

# Why Composability?

---

- ◆ Protocols don't run in a vacuum
  - Cryptographic protocols are typically used as building blocks in a larger secure system
  - For example, a key exchange protocol can be used to implement secure sessions
    - If two parties share a secret symmetric key, they can talk to each other as if connected by a secure “pipe”
- ◆ A protocol can be secure when used in standalone mode, but completely broken when used as a building block in a larger system
  - Therefore, security properties must hold in any environment in which the protocol is used

# Crypto Review: DDH Assumption

---

- ◆  $G$  is a group of large prime order  $q$

For  $g_1, g_2, u_1, u_2 \in G$  define

$$\text{DHP}(g_1, g_2, u_1, u_2) = \begin{cases} 1 & \text{if } \exists x \in \mathbb{Z}_q \text{ s.t. } u_1 = g_1^x, u_2 = g_2^x \\ 0 & \text{otherwise} \end{cases}$$

- ◆ **Decisional Diffie-Hellman** (DDH) Assumption says that there exists no efficient algorithm for computing DHP correctly with negligible error probability on all inputs

# More DDH

---

- ◆ DDH Assumption implies that distributions

$g, \{g^{x_i}\}, \{g^{y_j}\}, \{g^{x_i y_j}\}$  and

$g, \{g^{x_i}\}, \{g^{y_j}\}, \{g^{z_{ij}}\}$  where  $1 \leq i \leq n, 1 \leq j \leq m,$

$g, x_i, y_j, z_{ij}$  are random

are computationally indistinguishable

- ◆ DDH and Leftover Hash Lemma imply that the following are computationally indistinguishable:

$g, g^x, g^y, i, H_i(g^{xy})$  and

$g, g^x, g^y, i, K$

where  $K$  is random bit string,

$H$  is a "universal one-way hash function"

# Security of Digital Signatures

---

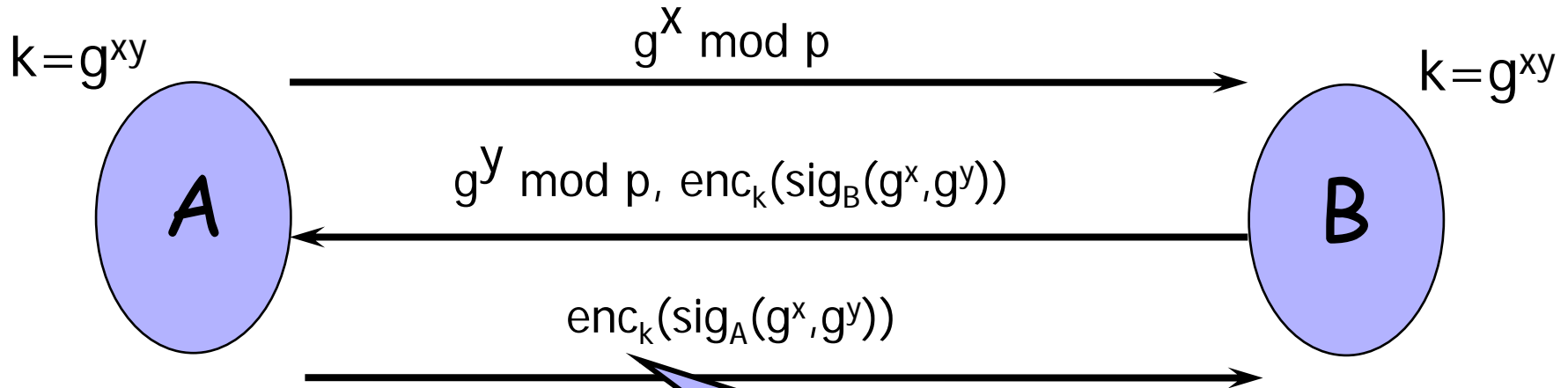
Digital signature scheme is secure if it is infeasible for any efficient adversary to win following game:

1. Signing key is generated, and the corresponding public key is given to the adversary.
2. Adversary requests signatures on any messages of his choice. Messages may depend on received signatures.
3. Adversary wins the game if he outputs a message other than those on which he previously requested signatures along with a valid signature on that message.

This is known as security against existential forgery

# Station-to-Station Protocol

[Diffie et al. '92]



## Interleaving attack:

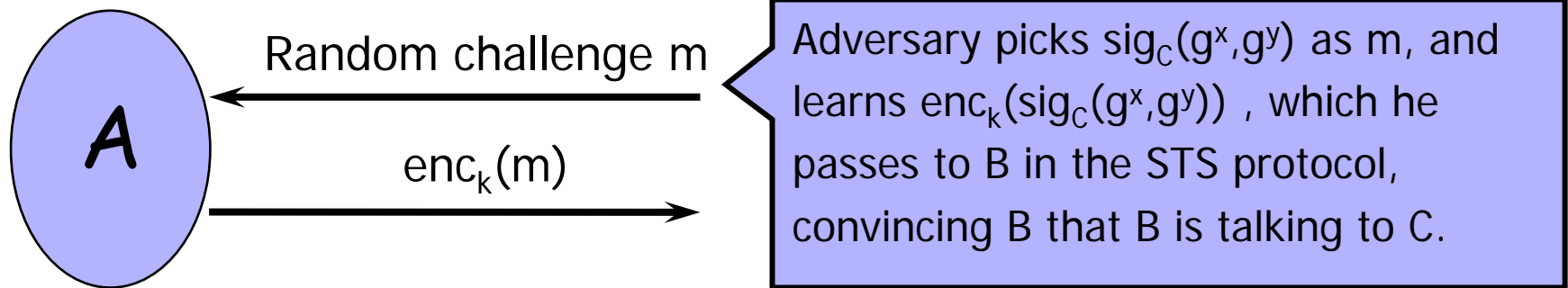
Adversary replays B's own encryption back to B.  
Result: B thinks he is talking to himself, A thinks he is talking to B.

This encryption is critical.

Without it, adversary can send  $\text{sig}_C(g^x, g^y)$ .  
Result: B thinks he is talking to C, while sharing a key with A, who thinks he is talking to B.

# Protocol Interference Attack

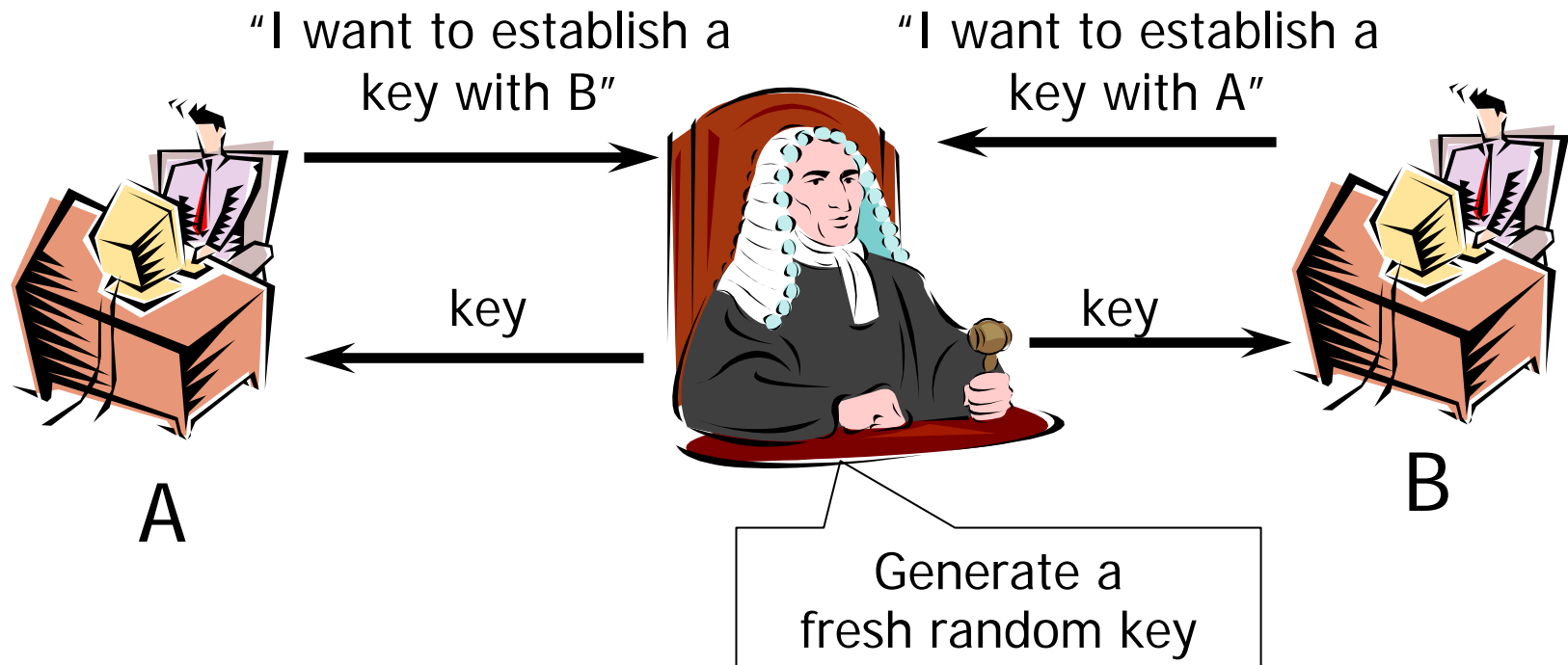
- ◆ What if, in addition to STS, A executes some protocol where this interaction takes place:



- ◆ Problem: challenge-response protocols may be used as encryption oracles by the adversary
- ◆ Problem: "hijacking" of honest user's public key
  - Fool CA into binding A's public key to a different identity

# Ideal Key Exchange Functionality

[Shoup]



This is a pure abstraction of the service that a key exchange protocol is expected to provide to higher-level protocols

# Ideal World: User Instances

---

[Shoup]

## ◆ InitializeUser( $i, ID_i$ )

- Assign unique identity  $ID_i$  to the  $i^{\text{th}}$  user
- User in the ideal world is simply a **placeholder**; he does not actually do anything
  - In ideal world, keys are created and distributed magically

## ◆ InitializeUserInstance( $i, j, \text{role}_{ij}, PID_{ij}$ )

- Same user may participate in multiple instances of the same protocol
- $I_{ij}$  is the identity of the new instance, which is communicating with some counterparty  $PID_{ij}$
- $\text{role}_{ij}$  is either 0, or 1

# Ideal World: Session Key Generation

---

◆ StartSession( $i, j, \text{adversaryKey}$ ) [Shoup]

- Create: IF generates  $K_{ij}$  as random bit string OR
- Connect( $i', j'$ ): IF sets  $K_{ij}$  equal to  $K_{i'j'}$  OR
- Compromise: IF sets  $K_{ij}$  equal to adversaryKey

- “Create” models creation of a brand-new session key to be used between the  $i^{\text{th}}$  and  $j^{\text{th}}$  user
- “Connect” models establishment of this session (the key magically becomes known to both user instances)
- “Compromise” models adversary’s corruption of a user

# Using the Key

---

[Shoup]

## ◆ Application( $f$ )

- IF gives to the adversary  $f(K_{ij})$  where  $\{K_{ij}\}$  is the set of all session keys. This is recorded in the transcript.
  - $f$  is a function or a program, possibly randomized, may have side effects
  - $f$  models how key is used after it is established
- Intuitively, function  $f$  defines what adversary may be able to learn after symmetric key has been established
  - For example, he may be able to learn ciphertexts computed by user using some randomized symmetric cipher and the new key. We encode this cipher as function  $f$ .

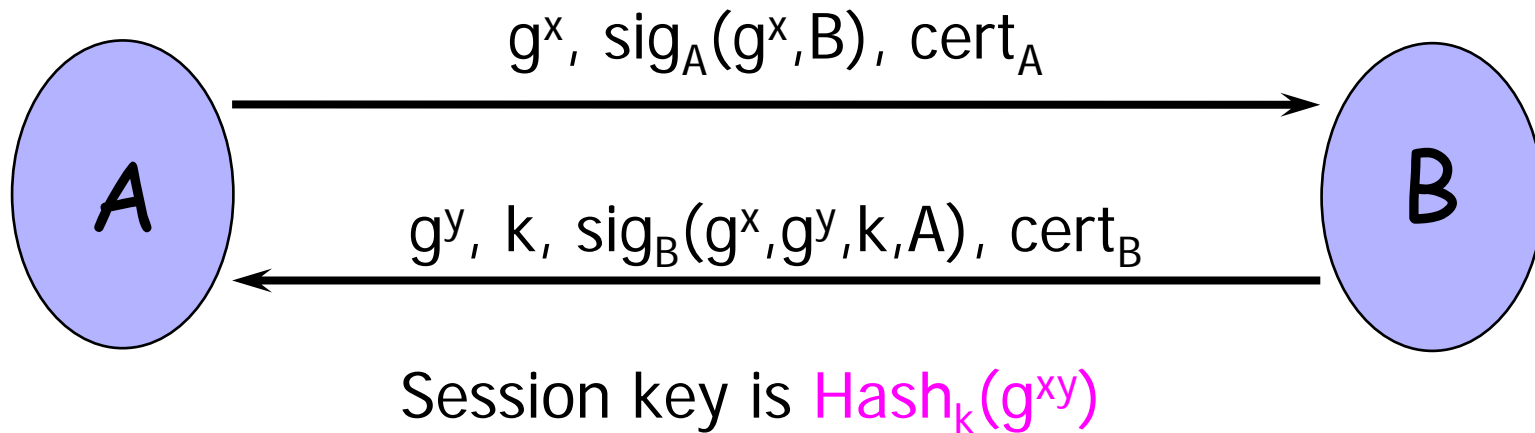
# Discussion

---

- ◆ Compositionality is much more than key secrecy
- ◆ Application operation allows keys to be used in an arbitrary way by higher-level protocols
  - Can encode any higher-level key-based functionality as some function  $f$ , then add  $\text{Application}(f)$  to transcript
- ◆ Real protocol is indistinguishable from the ideal functionality regardless of how keys are used later
  - Keys are indistinguishable from random strings
    - Even if they are completely revealed at the end of the protocol  
(why?)

# DHKE Protocol

---



Assuming the digital signature scheme is secure against existential forgery,

DHKE is a secure key exchange protocol under the DDH assumption

# Proof of Simulatability

---

- ◆ Given real-world adversary  $R$ , construct ideal-world simulator  $S$  who simulates protocol to  $R$ 
  - $R$  should not be able to distinguish transcript of real-world protocol execution from a simulation created by  $S$
- ◆ When a session is established in the real world,  $S$  “connects” corresponding user instances in the ideal world
  - In the ideal world,  $I$  uses randomly generated ideal keys instead of real-world computed keys
    - Must prove that the difference is undetectable

# DHKE: Security for Responder (1)

---

- ◆ Suppose user instance B received 1<sup>st</sup> message and accepted
- ◆ If  $PID_B$  is not assigned to honest user, then “compromise” B in the ideal world
  - $PID_B$  is initiator’s identity (in responder’s view)
    - $PID_B$  not assigned means that the protocol is being executed with the adversary (or adversary-controlled user) as initiator
  - Extract session key from the responder in the real world, and use it as argument to the “compromise” operation in the ideal world

# DHKE: Security for Responder (2)

---

- ◆ If  $\text{PID}_B$  has been assigned to user, then “create” B in the ideal world
  - This means that the protocol is being executed with an honest user as the initiator
  - “Create” models key creation in ideal world. IF creates a random session key for B. In the real world, the key is not random: it is computed as  $\text{Hash}_k(g^{xy})$ .
- ◆ DDH Assumption and Leftover Hash Lemma imply that  $\text{Hash}_k(g^{xy})$  is computationally indistinguishable from a random key even if  $g^x$ ,  $g^y$ , and  $k$  are known

# DHKE: Security for Initiator (1)

---

- ◆ Suppose user instance A received 2<sup>nd</sup> message and accepted
- ◆ If  $PID_A$  is not assigned to honest user, then “compromise” A in the ideal world
  - $PID_A$  is responder’s identity (in initiator’s view)
    - $PID_A$  not assigned means that the protocol is being executed with the adversary (or adversary-controlled user) as responder
  - Extract session key from the responder in the real world, and use it as argument to the “compromise” operation in the ideal world

# DHKE: Security for Initiator (2)

---

- ◆ If  $PID_A$  has been assigned to user B, then “connect” A and B in the ideal world
  - Protocol is being executed with honest responder B
  - “Connect” magically gives B’s random session key to A
- ◆ Security of digital signature scheme guarantees that A’s and B’s values of  $g^x$ ,  $g^y$ , and  $k$  match
  - Therefore, A’s and B’s keys are equal in the real world
- ◆ There is no detectable difference between worlds
  - A’s and B’s keys are equal in both worlds
  - In ideal world, keys are random. In real world, they are computed, but this is not computationally detectable