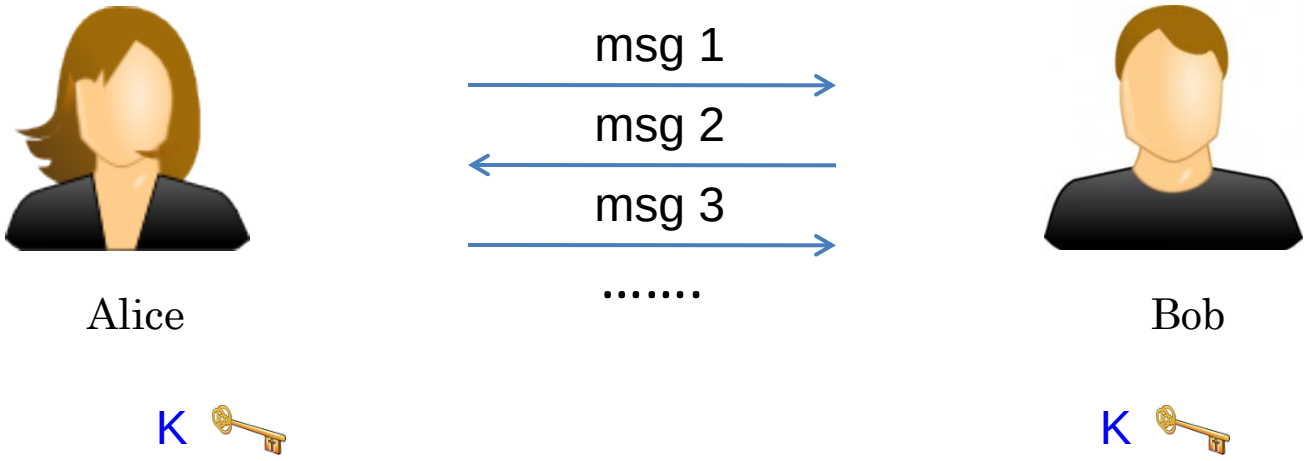
An Introduction to Authenticated Key Exchange Protocols

Guomin Yang
Centre for Computer and Information Security Research
University of Wollongong

Outline

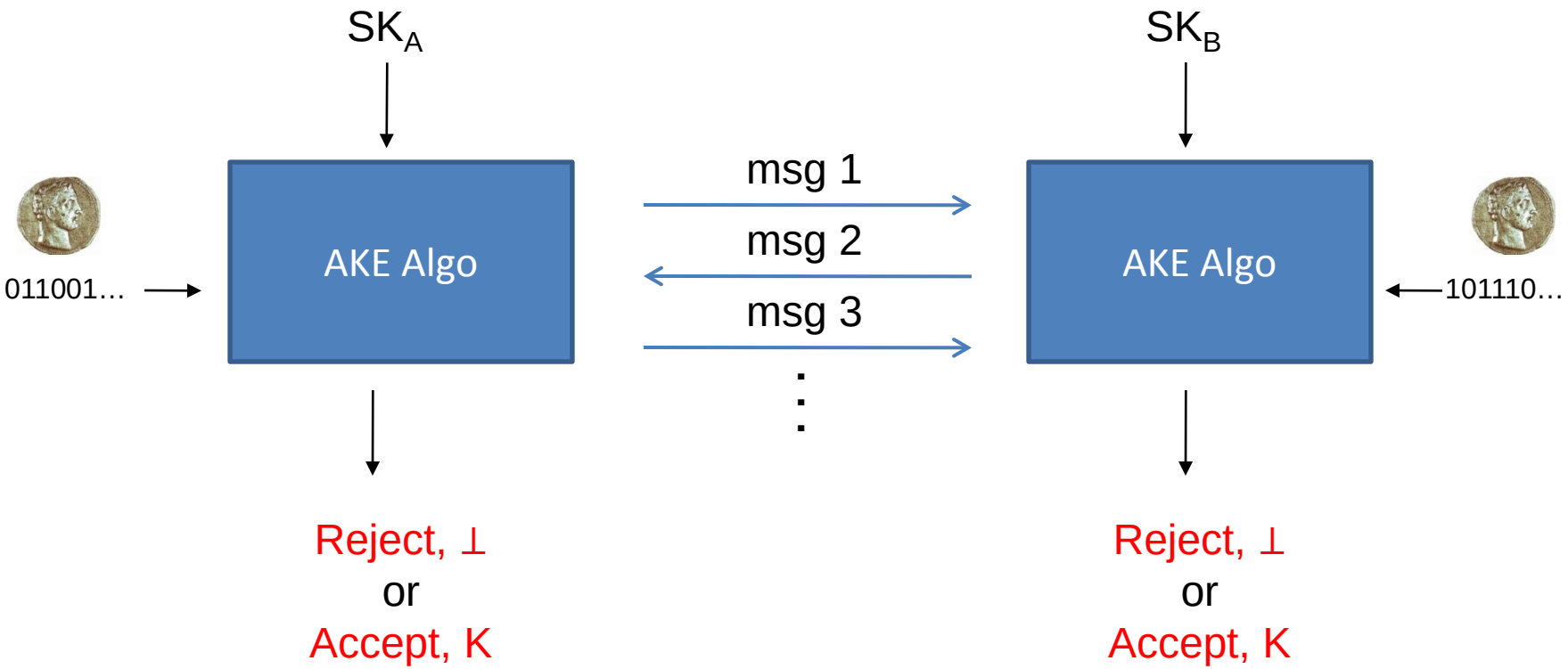
- Introduction
- Attacks against AKE
- Security model
- AKE examples with security analysis
- Conclusions

Authenticated Key Exchange (AKE)



- Security Goals
  - Mutual Authentication
  - Secure Key Establishment
- Examples: IPSec (IKE), TLS/SSL, SSH, GSM/3GPP

A Closer Look



Common attacks

- Eavesdropping attack
  - The attacker captures the information sent in the protocol.
- Modification attack
  - The attacker alters the information sent in the protocol.
- Replay attack
  - The adversary records information seen in the protocol, and then sends it to the same, or a different, entity, possibly during a later protocol run.
- Known-key attack
  - The adversary obtains the key of one communication session, and uses it to attack another session
  - The adversary obtains a long-term key, and uses it to attack the old sessions
- ……

Assumptions (Mathuria-Boyd)

- **Assumption 1**
  The adversary is able to eavesdrop, modify, re-route, insert messages during the execution of a cryptographic protocol.
- **Assumption 2**
  The adversary is able to obtain the value of any old session key
- **Assumption 3**
  The adversary may start any number of parallel protocol runs between any parties including different runs involving the same parties.
- **Assumption 4 (for group AKE)**
  The adversary may be a legitimate protocol participant (an insider), or an external party (an outsider), or a combination of both.

Diffie-Hellman Key Exchange

A, $X = g^x$

B, $Y = g^y$

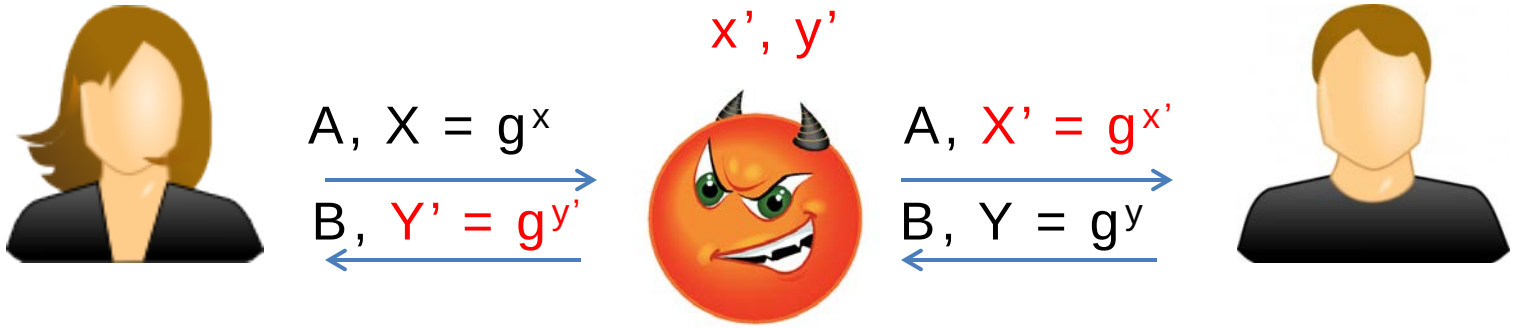
$K_A = Y^x = g^{xy}$

$K_B = X^y = g^{xy}$

- Diffie-Hellman Assumption:

  given g^x and g^y, it is computationally infeasible to compute g^{xy}

Man-In-The-Middle Attack

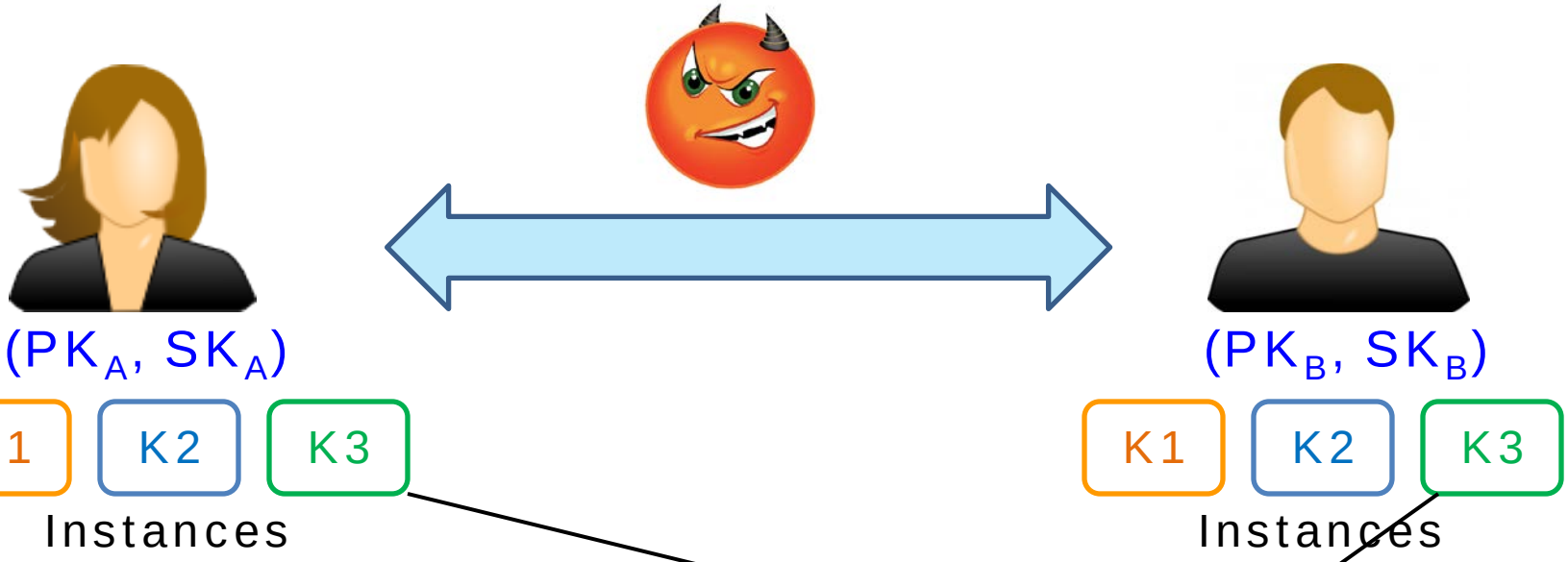


- The adversary is able to derive both K_A and K_B
- Weakness in DH: no authentication

AKE Security Model

(Canetti-Krawczyk Eurocrypt'01)

Adversarial game: n Parties and 1 Adversary

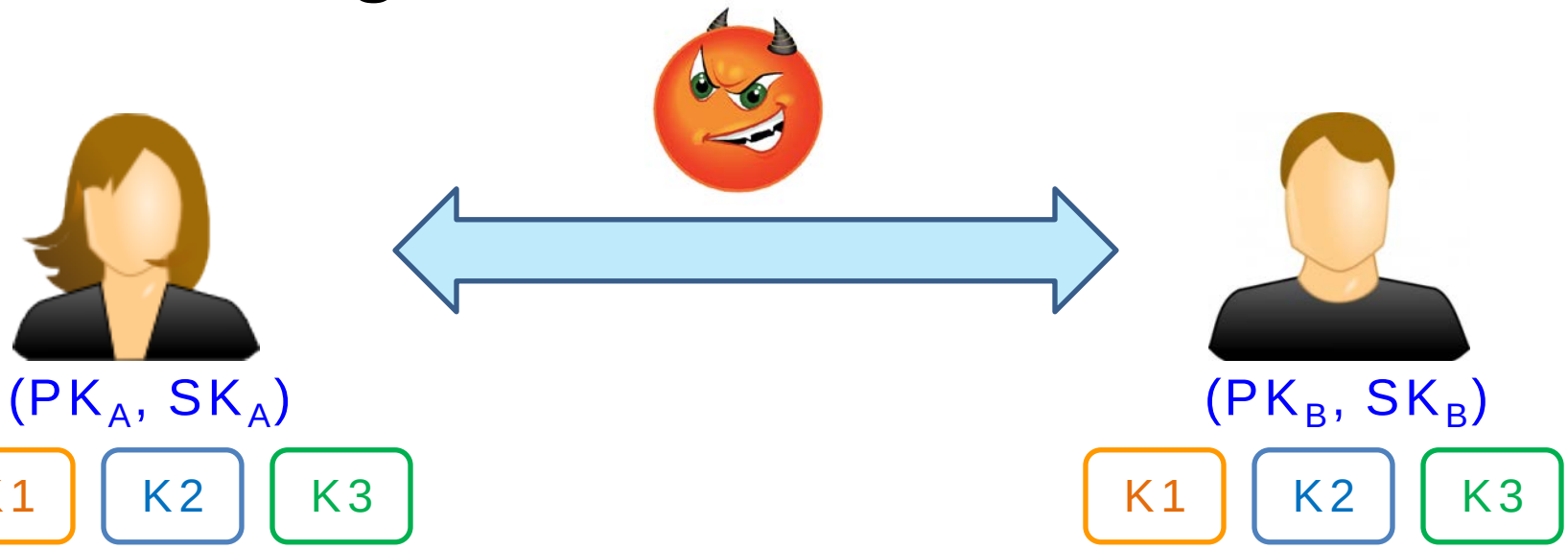


- Queries:
  - Send
  - Session key reveal
  - Session state reveal
  - Corruption

Partners: two instances having the same session id
(sid: communication transcript or part of it)

AKE Security Model

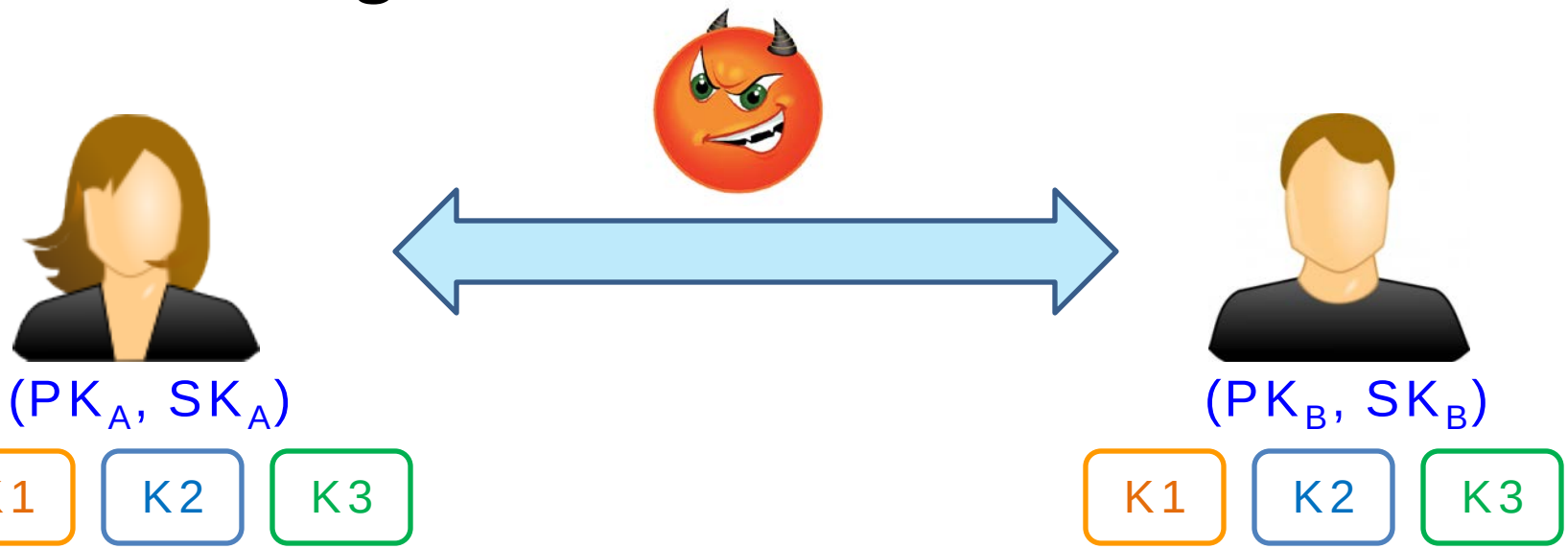
Adversarial game:



- Queries (cont):
  - Test: instance i at user P
    1. Instance i has successfully completed the session (with knowledge of peer party Q)
    2. No session key reveal to i
    3. No session state reveal to i
    4. No corruption to P before the completion of i
    5. If i has a partner instance j at Q, then 2,3,4 also apply to j
    6. If i has no partner instance at Q, then Q cannot be corrupted

AKE Security Model

Adversarial game:



- Toss a random coin b
  - If b = 0, return Ki to adversary
  - If b = 1, return a random value to adversary
- The adversary can continue the game after Test
- Adversary outputs b'
- If b' = b, the Exp. returns 1; otherwise, the Exp. Returns 0
- Secure AKE:

$$\Pr[\text{Exp. outputs } 1] = 1/2 + \text{negl}$$

SIG-DH V1

(SK_A, PK_A)

$K_A = Y^x = g^{xy}$

$A, X = g^x , Sig(SK_A,X)$

$B, Y = g^y, Sig(SK_B,X,Y)$

(SK_B, PK_B)

$K_B = X^y = g^{xy}$

- Is this protocol secure?

SIG-DH V2

(SK_A, PK_A)

$K_A = Y^x = g^{xy}$

$A, X = g^x$

$B, Y = g^y, Sig(SK_B, X, Y)$

$Sig(SK_A, Y, X)$

(SK_B, PK_B)

$K_B = X^y = g^{xy}$

- Is this protocol secure?

An unknown key share attack

Adversary first corrupts a user E.

The adversary activates A to start a new session with B

1: $A \rightarrow Adv: A, Y_A$

1': $Adv \rightarrow B: E, Y_A$

2': $B \rightarrow Adv: B, Y_B, Sig_B(Y_B, Y_A)$

2: $Adv \rightarrow A: B, Y_B, Sig_B(Y_B, Y_A)$

3: $A \rightarrow Adv: Sig_A(Y_A, Y_B)$

3': $Adv \rightarrow B: Sig_E(Y_A, Y_B)$

The session in blue colour is fresh!
Session key reveal allows the adversary to win the game.

SIG-DH V3

(SK_A, PK_A)

$K_A = Y^x = g^{xy}$

$A, X = g^x$ →

← $B, Y = g^y, Sig(SK_B, X, Y, A)$

$Sig(SK_A, Y, X, B)$ →

(SK_B, PK_B)

$K_B = X^y = g^{xy}$

- Is this protocol secure?
- Yes (Canetti-Krawczyk'01)
- None of the three elements in the signature can be omitted

Security proof sketch

- Exp 0: original CK game
- Exp 1: denote by FORGE the following event
  - Adversary makes a send query with valid signature S of P
  - P is not corrupted at the time the send query is made
  - S does not appear in the answer of any send query

If a FORGE event happens, then Exp1 returns a random bit

Security proof sketch

$\Pr[exp0 \rightarrow 1] - \Pr[exp1 \rightarrow 1] \leq \Pr[FORGE]$

Lemma: If $Pr[A \mid \neg C] = Pr[B \mid \neg C]$, then

$$|Pr[A] - Pr[B]| \leq Pr[C]$$

- Exp 2: Replace the session key of the test session by a random value

$\Pr[exp1 \rightarrow 1] - \Pr[exp2 \rightarrow 1] \leq AdvDDH$

- $\Pr[exp2 \rightarrow 1] = 1/2$

A Generic Approach

- A passive secure KE protocol P
- An authenticator A
- An active secure AKE protocol P’
  - Secure every message of P using A

Authenticator Examples

Signature based

$$P_i \rightarrow P_j : m$$
$$P_i \leftarrow P_j : m,\ N_j$$
$$P_i \rightarrow P_j : m,\ SIG_{P_i}(m, N_j, P_j)$$

Encryption based

$$P_i \rightarrow P_j : m$$
$$P_i \leftarrow P_j : m, ENC_{P_i}(N_j)$$
$$P_i \rightarrow P_j : m,\ MAC_{N_j}(m, P_j)$$

HMQV

$PK_A = g^a$

$A, X = g^x$

$B, Y = g^y$

$PK_B = g^b$

$d = G(X, B), e = G(Y, A)$

$S_A = (Y \cdot PK_B^e)^{x+da} = g^{(x+da)(y+eb)}$
$K_A = H(S_A)$

$S_B = (X \cdot PK_A^d)^{y+eb} = g^{(x+da)(y+eb)}$
$K_B = H(S_B)$

- Only implicit authentication
- Easy to achieve explicit authentication (by adding key confirmation using MAC)
- Security proof – refer to the presentation by Yangguang Tian

Research topics on AKE

- Leakage-resilient AKE
  - Alwen et al. *Crypto'09*
  - Dodis et al. *Asiacrypt'10*
  - The model can be further strengthened
- AKE under bad randomness
  - Yang et al. *FC'11*
  - Efficiency can be improved
  - $HMQV^{+}$
- Post-quantum AKE