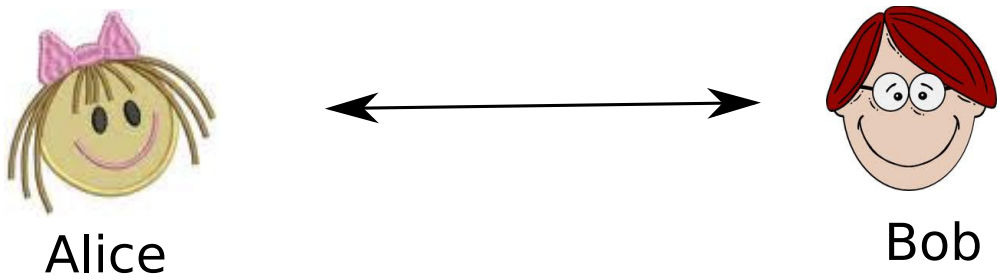
Key Management

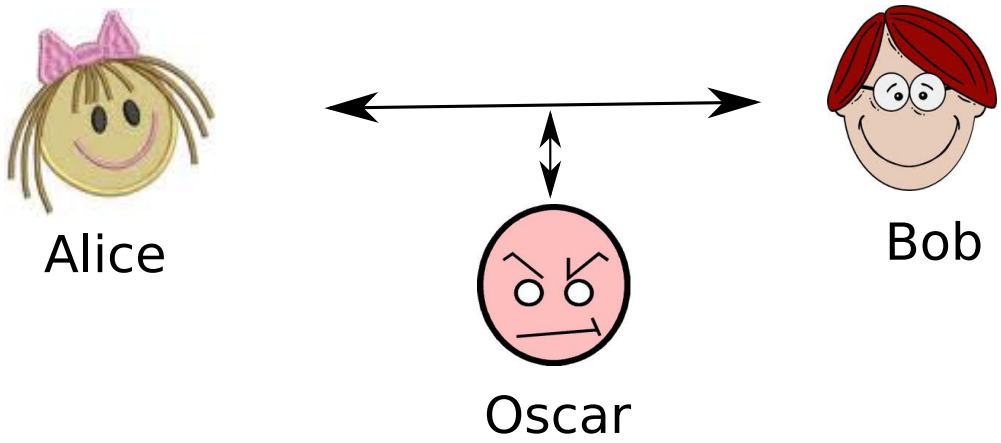
Sushmita Ruj

R C Bose Center for Cryptology and Security
Indian Statistical Institute, Kolkata
E-mail : sush@isical.ac.in
Homepage: http://www.isical.ac.in/∼sush

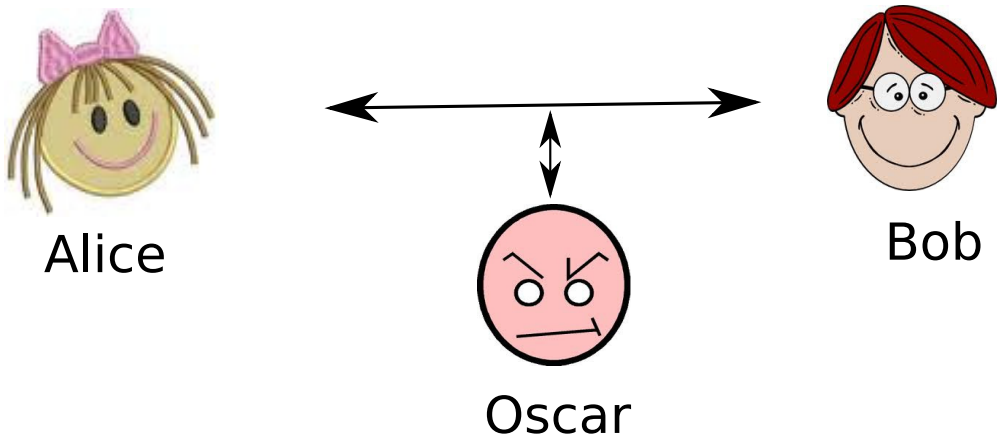
Communication



Need for Secure Communication

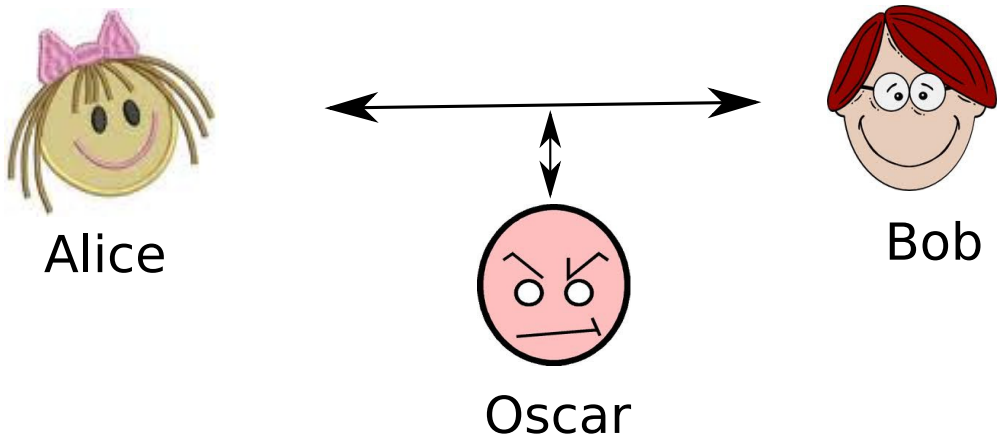


Need for Secure Communication



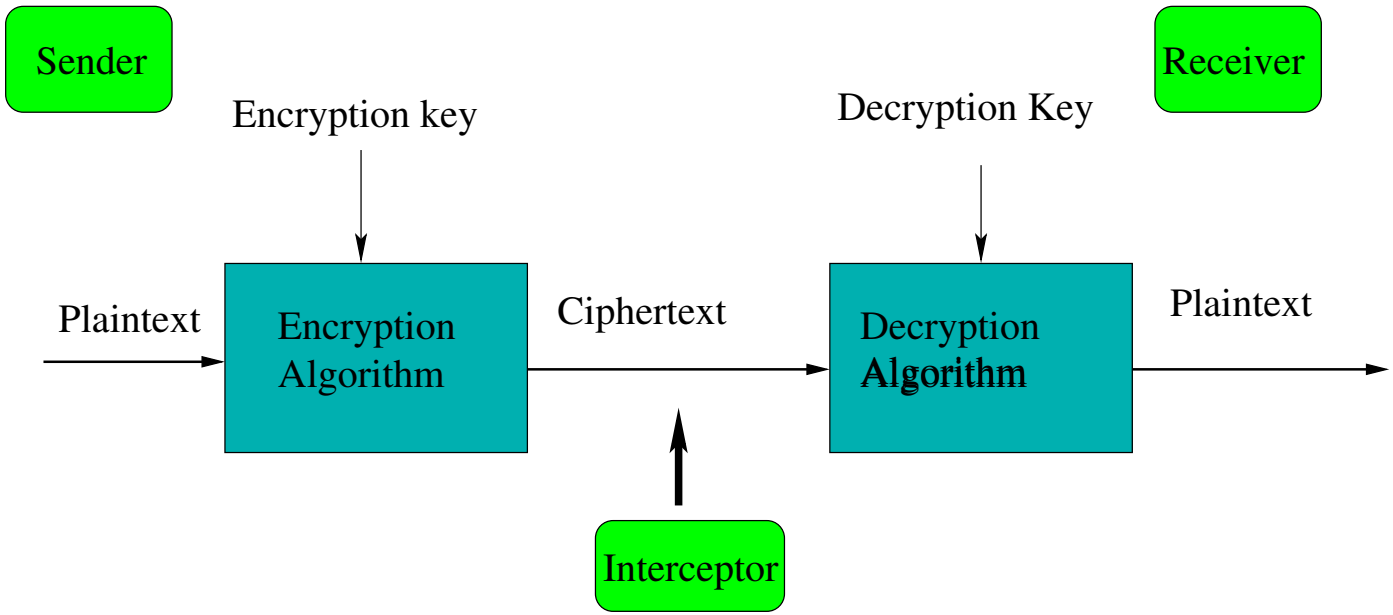
- Oscar can either be either an active attacker or a passive attacker

Need for Secure Communication



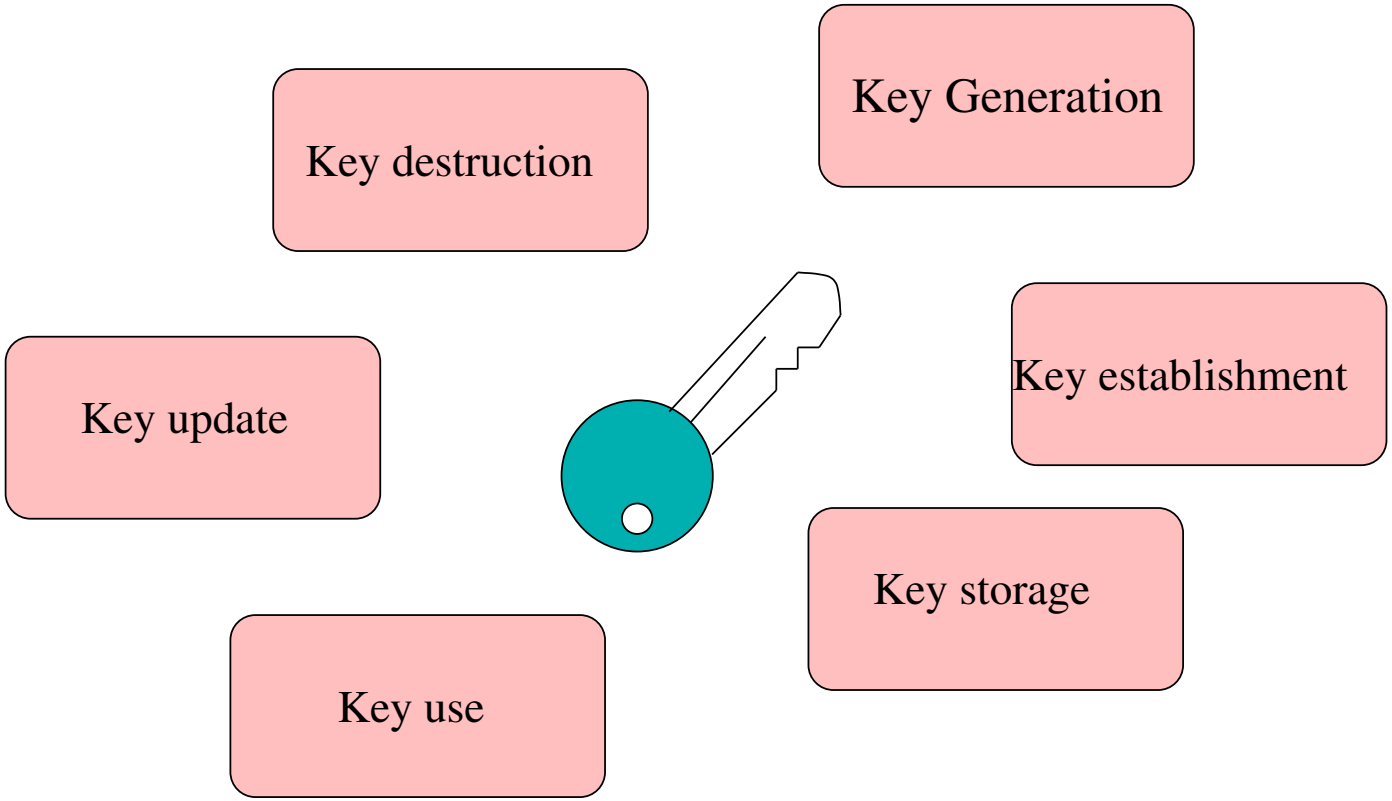
- Oscar can either be either an active attacker or a passive attacker
- Alice and Bob need to communicate securely

Basic Model of a Cryptosystem



Key Management

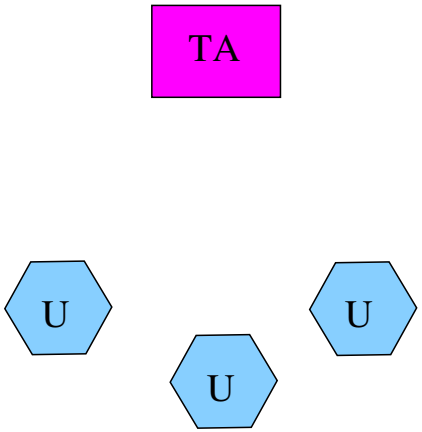
Key management is a generic term that is associated with the entire lifecycle of cryptographic keys:



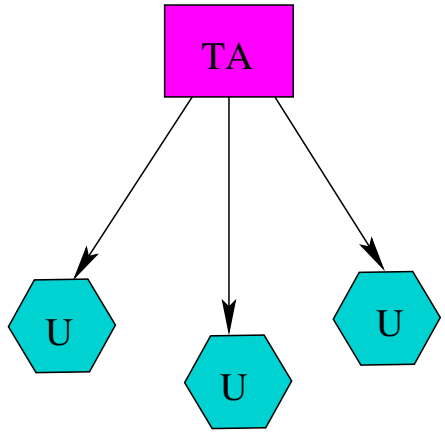
Key Distribution Schemes

- Broadly of three types and might use a Trusted Authority (TA)
- Key predistribution schemes: Users have no communication channels to support key establishment and thus must be able to do so on their own.
  TA Distributes Keys ahead of time.
- Session Key distribution : Session keys are used to encrypt information for a short period of time. TA distributes session keys via an interactive protocol.
- Key agreement: Users use an interactive protocol to construct session keys. Can be both public/private key based scheme.

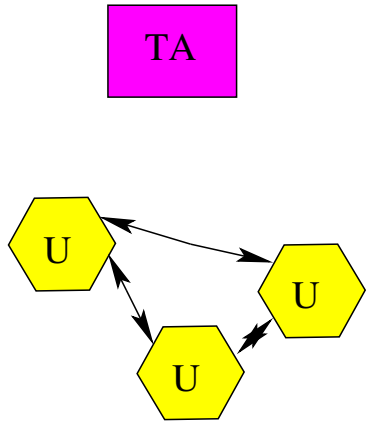
Classification of key management schemes



Key Predistribution



Session Key Distribution



Key Agreement

Types of Keys

- Long-lived Keys (LL Keys): Precomputed and stored securely.
- Short-lived keys: Session keys valid for a short period of time and regularly refreshed.

Simple ways to distribute LL Keys

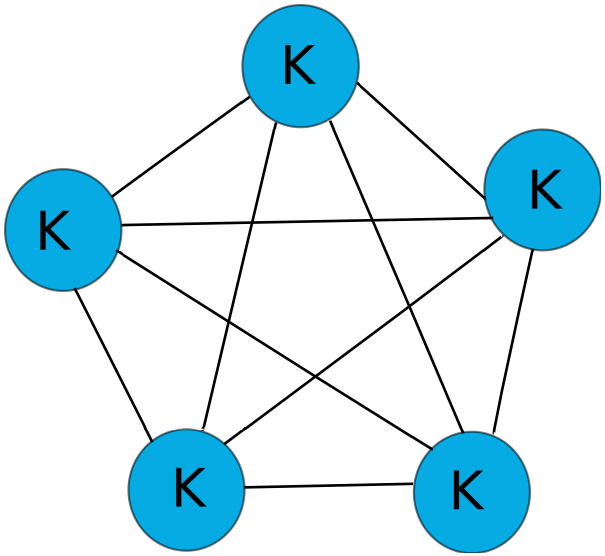


Figure : Master key distribution

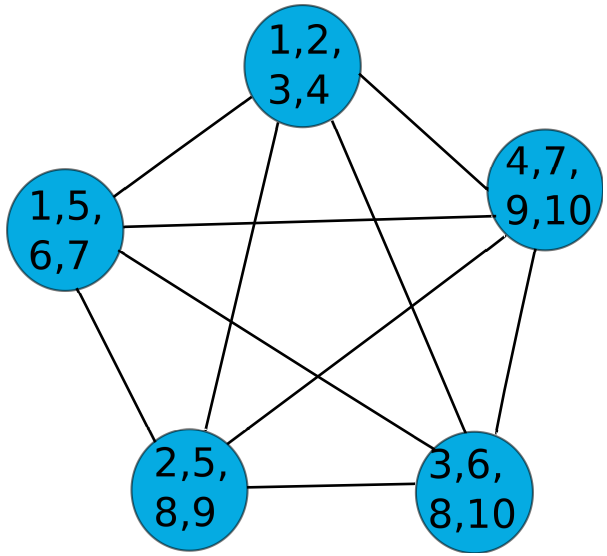


Figure : Pairwise key distribution

- Fully connected
- Minimal storage

- Fully connected
- High storage

Simple ways to distribute LL Keys

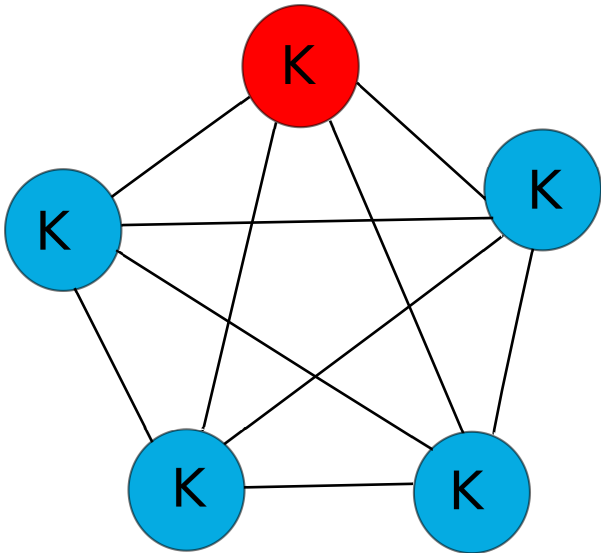


Figure : Master key distribution

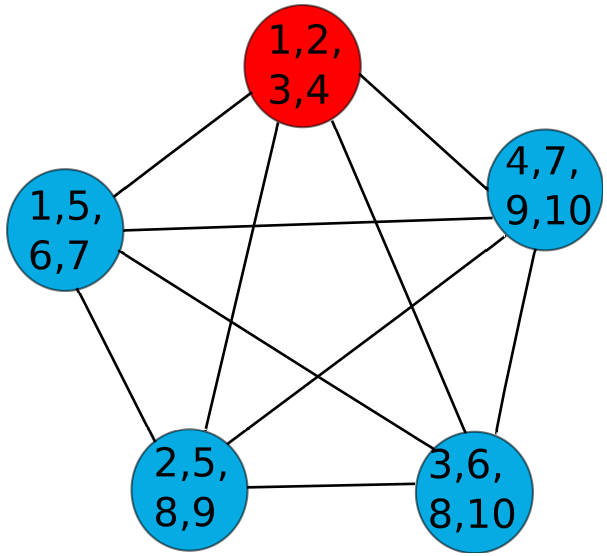


Figure : Pairwise key distribution

- Fully connected
- Minimal storage

- Fully connected
- High storage

Simple ways to distribute LL Keys

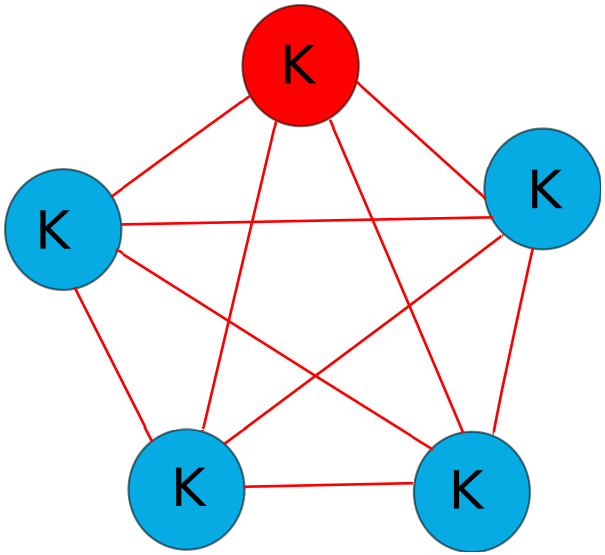


Figure : Master key distribution

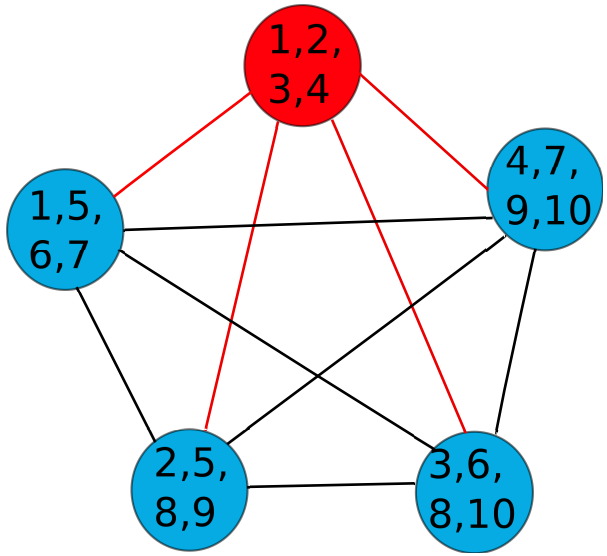


Figure : Pairwise key distribution

- Fully connected
- Minimal storage
- Low resilience

- Fully connected
- High storage
- High resilience

Types of attacks

- Passive Attacks: Eavesdroping on the channel
- Active Attacks:
  - Alter messages being transmitted over the channel
  - Save messages for future reuse (replay attacks)
  - Masquerade as various users in the network

Types of key establishment: based on users

- Pairwise: Two parties establish keys for communication
- Groupwise: Multiple parties establish common key for communication
- Pairwise keys are much simpler to construct
- So, we focus on pairwise key management for most of the talk

Polynomial based pairwise scheme

- Generalized version of Blom's Scheme
- Let $\mathbb{Z}_p$ be a finite field, p prime
- Let $P(x, y)$ be a symmetric polynomial of degree c with coefficients in $\mathbb{Z}_p$
- So, $P(x, y) = P(y, x)$
- User U_i receives the polynomial $f_i(x) = P(x, i)$
- User U_j receives the polynomial $f_j(x) = P(x, j)$
- Common key between i and j is $P(i, j) = P(j, i)$

Example

- $p = 7$, Coefficients in $\mathbb{Z}_7$
- $P(x, y) = x^2 + y^2 + 5xy$. Symmetric polynomial of degree 2.
- Let there be 7 users.

Node n_i	polynomial $f_i(x)$
U_0	x^2
U_1	$x^2 + 5x + 1$
U_2	$x^2 + 3x + 4$
U_3	$x^2 + x + 2$
U_4	$x^2 + 6x + 2$
U_5	$x^2 + 4x + 4$
U_6	$x^2 + 2x + 1$

- For instance, U_2 and U_6 can calculate the common key identifier as $f_2(6) = 6^2 + 3.6 + 4 = 2$ or $f_6(2) = 2^2 + 2.2 + 1 = 2$.
  Their common key is 2.

Security of polynomial based scheme

Lagrange's Interpolation

Suppose q is a prime and $y_0, y_1, \ldots, y_n$ are distinct elements in $\mathbb{Z}_q$. Suppose $f_0(x)$, $f_1(x)$, ..., $f_n(x) \in \mathbb{Z}_q[x]$ are polynomials of degree at most c. Then there exists a polynomial of degree at most c in two variables x and y, such that $P(x, y_i) = f_i(x)$, where $0 \leq i \leq n$. $P(x, y)$ is given by

$$P(x, y) = \sum_{j=0}^{n} f_j(x) \Pi_{0 \leq h \leq n, j \neq h} \frac{y - y_h}{y_j - y_h}.$$

Security of polynomial based scheme

Example

$f_1(x) = x^2 + 5x + 1$, $f_2(x) = x^2 + 3x + 4$ and $f_3(x) = x^2 + x + 2$

$\frac{(y-2)(y-3)}{(1-2)(1-3)} = 4y^2 + y + 3$, $\frac{(y-1)(y-3)}{(2-1)(2-3)} = 6y^2 + 4y + 4$,

$\frac{(y-1)(y-2)}{(3-1)(3-2)} = 4y^2 + 2y + 1$.

Hence,

$$P(x, y) = (x^2 + 5x + 1)(4y^2 + y + 3) + (x^2 + 3x + 4)(6y^2 + 4y + 4) + (x^2 + x + 2)(4y^2 + 2y + 1)$$
$$= x^2 + y^2 + 5xy.$$

Security of polynomial based scheme

Example

$f_1(x) = x^2 + 5x + 1$, $f_2(x) = x^2 + 3x + 4$ and $f_3(x) = x^2 + x + 2$

$\frac{(y-2)(y-3)}{(1-2)(1-3)} = 4y^2 + y + 3$, $\frac{(y-1)(y-3)}{(2-1)(2-3)} = 6y^2 + 4y + 4$,

$\frac{(y-1)(y-2)}{(3-1)(3-2)} = 4y^2 + 2y + 1$.

Hence,

$$P(x, y) = (x^2 + 5x + 1)(4y^2 + y + 3) + (x^2 + 3x + 4)(6y^2 + 4y + 4) + (x^2 + x + 2)(4y^2 + 2y + 1)$$
$$= x^2 + y^2 + 5xy.$$

- This is Lagrange's interpolation.
- If the values of more than c polynomials are known, then $P(x, y)$ can be found by interpolation
- If more than c nodes are compromised, then the scheme is "broken"
- $c - secure$ scheme

Combinatorial Key Predistribution

- Proposed by Mitchell and Piper (1988)
- A set system or design is a pair $(X, \mathcal{A})$, where $\mathcal{A}$ is a set of subsets of X, called blocks.

Combinatorial Key Predistribution

- Proposed by Mitchell and Piper (1988)
- A set system or design is a pair $(X, \mathcal{A})$, where $\mathcal{A}$ is a set of subsets of X, called blocks.
- $X = \{0, 1, 2, 3, 4, 5\}$
- $\mathcal{A} = \{\{0, 1, 2\}, \{0, 1, 3\}, \{0, 2, 4\}, \{0, 3, 5\}, \{0, 4, 5\}, \{1, 2, 5\}, \{1, 3, 4\}, \{1, 4, 5\}, \{2, 3, 4\}, \{2, 3, 5\}\}$

Combinatorial Key Predistribution

- Proposed by Mitchell and Piper (1988)
- A set system or design is a pair $(X, \mathcal{A})$, where $\mathcal{A}$ is a set of subsets of X, called blocks.
- $X = \{0, 1, 2, 3, 4, 5\}$
- $\mathcal{A} = \{\{0, 1, 2\}, \{0, 1, 3\}, \{0, 2, 4\}, \{0, 3, 5\}, \{0, 4, 5\}, \{1, 2, 5\}, \{1, 3, 4\}, \{1, 4, 5\}, \{2, 3, 4\}, \{2, 3, 5\}\}$
- A $BIBD(v, b, r, k, \lambda)$, is a design which satisfies the following conditions:
  1. $|X| = v$, $|\mathcal{A}| = b$,
  2. Each subset in $\mathcal{A}$ contains exactly k elements,
  3. Each element in X occurs in r many blocks,
  4. Each pair of elements in X is contained in exactly λ blocks in $\mathcal{A}$.
- The above design is a $(6, 10, 5, 3, 2)$-design

Combinatorial basis of the problems

- Take the Dual of the design, interchange between elements and blocks.

Combinatorial basis of the problems

- Take the Dual of the design, interchange between elements and blocks.
- $X' = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $\mathcal{A}' = \{\{0, 1, 2, 3, 4\}, \{0, 1, 5, 6, 7\}, \{0, 2, 5, 8, 9\}, \{1, 3, 6, 8, 9\}, \{2, 4, 6, 7, 8\}, \{3, 4, 5, 6, 9\}$

Combinatorial basis of the problems

- Take the Dual of the design, interchange between elements and blocks.
- $X' = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $\mathcal{A}' = \{\{0, 1, 2, 3, 4\}, \{0, 1, 5, 6, 7\}, \{0, 2, 5, 8, 9\}, \{1, 3, 6, 8, 9\}, \{2, 4, 6, 7, 8\}, \{3, 4, 5, 6, 9\}\}$
- A $Dual - BIBD(v, b, r, k, \lambda)$, is a design which satisfies the following conditions:
  1. $|X'| = b$, $|\mathcal{A}'| = v$,
  2. Each subset in $\mathcal{A}'$ contains exactly r elements,
  3. Each element in X' occurs in k many blocks,
  4. Each pair of blocks have exactly λ elements in common.
- The above design is a $(10, 6, 3, 5)$-design

Combinatorial basis of the problems

- Take the Dual of the design, interchange between keys and users.

Combinatorial basis of the problems

- Take the Dual of the design, interchange between keys and users.
- $X' = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ Key pool
- $\mathcal{A}' = \{\{0, 1, 2, 3, 4\}, \{0, 1, 5, 6, 7\}, \{0, 2, 5, 8, 9\}, \{1, 3, 6, 8, 9\}, \{2, 4, 6, 7, 8\}, \{3, 4, 5, 6, 9\}$ Users

Combinatorial basis of the problems

- Take the Dual of the design, interchange between keys and users.
- $X' = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ Key pool
- $\mathcal{A}' = \{\{0, 1, 2, 3, 4\}, \{0, 1, 5, 6, 7\}, \{0, 2, 5, 8, 9\}, \{1, 3, 6, 8, 9\}, \{2, 4, 6, 7, 8\}, \{3, 4, 5, 6, 9\}$ Users
- A $Dual - BIBD(v, b, r, k, \lambda)$, is a design which satisfies the following conditions:
  1. $|X'| = b$, $|\mathcal{A}'| = v$, b: size of key pool, v: # users
  2. Each subset in $\mathcal{A}'$ contains exactly r elements, r: size of key chain
  3. Each element in X' occurs in k many blocks,
  4. Each pair of blocks have exactly λ elements in common.
  5. Two user share atleast λ keys
- The above design is a $(10, 6, 3, 5)$-design.

BIBD and Users

Correspondence between a dual of $\lambda - (v, b, r, k)$ design and key predistribution.

- $b =$ key-pool size
- $v =$ number of users
- $k =$ number of users in which a given key occurs
- $r =$ number of keys gievn to one user (key-chain length)
- $\lambda =$ number of common keys between two nodes

  Consider dual of 2-(6,10,5,3) :

$$\text{Key-pool} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$U_1 : \{0, 1, 2, 3, 4\}$ $\quad$ $U_4 : \{1, 3, 6, 8, 9\}$

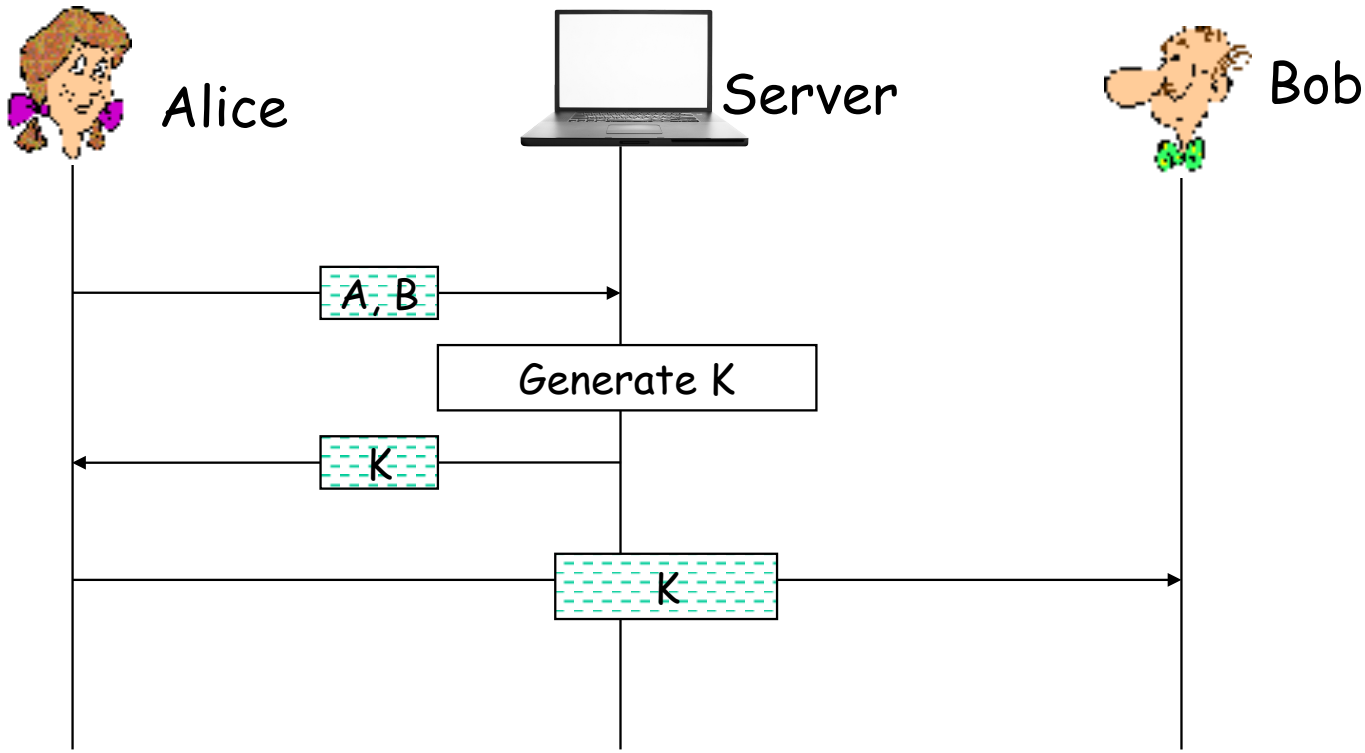
$U_2 : \{0, 1, 5, 6, 7\}$ $\quad$ $U_5 : \{2, 4, 6, 7, 8\}$

$U_3 : \{0, 2, 5, 8, 9\}$ $\quad$ $U_6 : \{3, 4, 5, 6, 9\}$

Session key distribution: Objectives

- Alice and Bob should learn the value of the session key K (Effectiveness)
- No other party should know the value of K (implicit key authentication)
- Alice and Bob should believe that the key is freshly generated (Key Freshness)
- Alice should believe that Bob knows K and vice versa (Key confirmation)

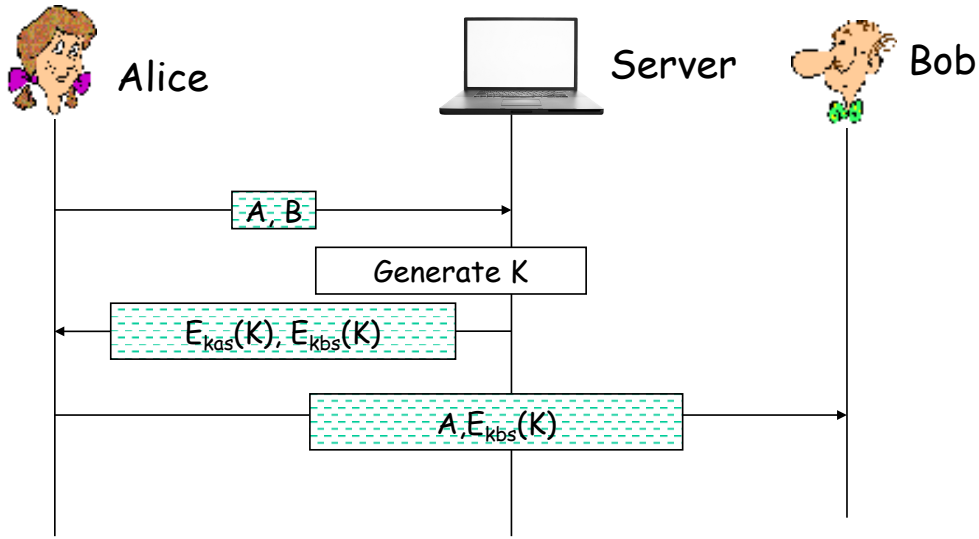
First Attempt at Sesson key distribution



The most obvious problem:

- The adversary can eavesdrop K
- Implicit key authentication is not provided

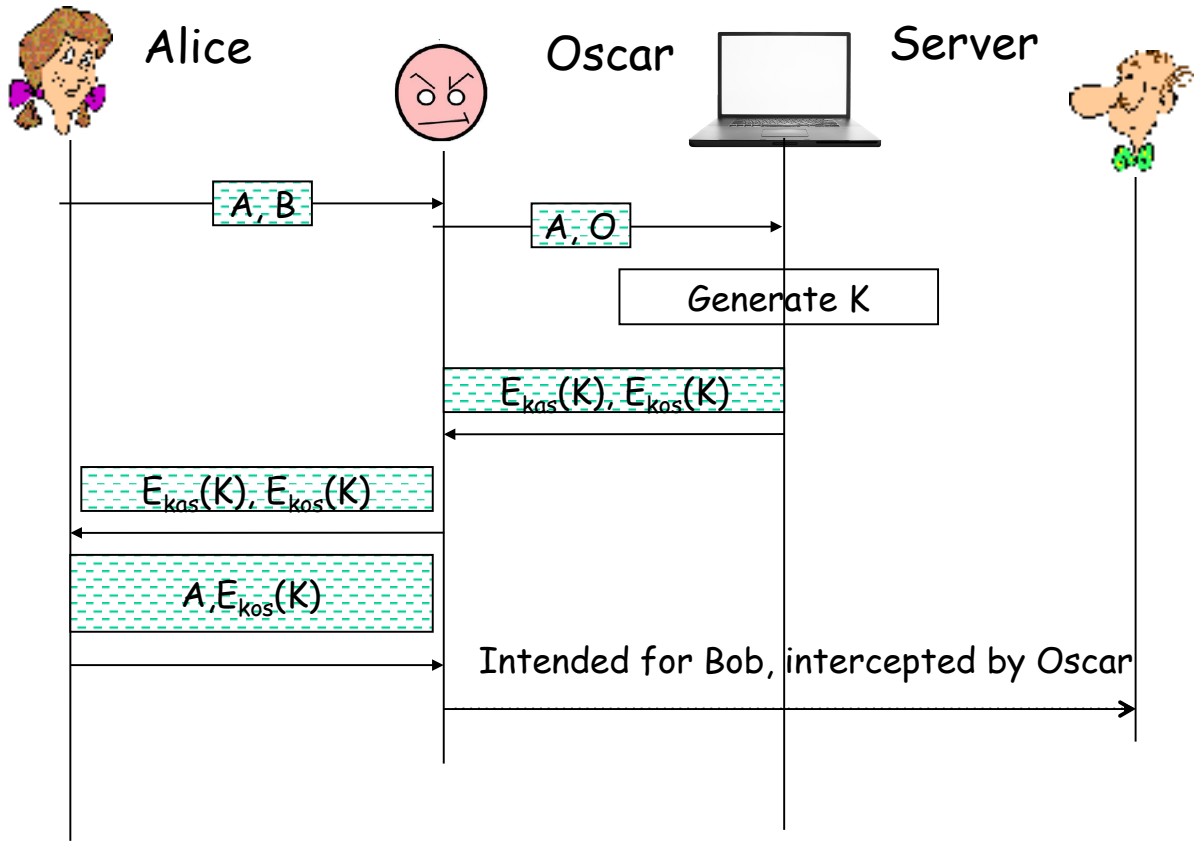
Second attempt at Sesson key distribution



Alice shares a secret key *Ksa* with the TA and Bob shares a secret key *Ksb* with the TA Problem:

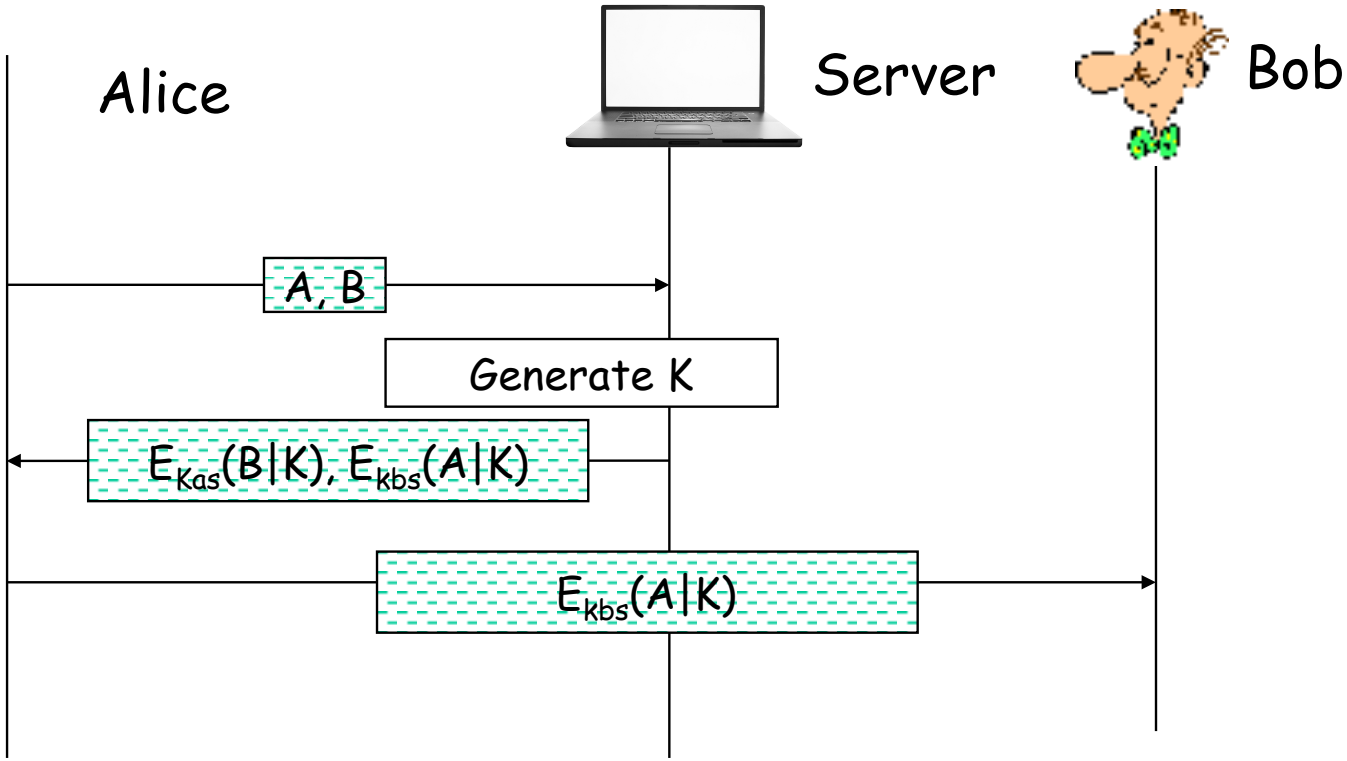
- Alice cannot be sure that K has been constructed for the session between herself and Bob.
- Similarly, Bob is not sure that K has been constructed for the session between herself and Bob.
- Implicit key authentication not provided.

An attack against the second attempt



- (Wo)Man in the middle attack
- Alice thinks that she shares K with Bob, but shares it with Oscar

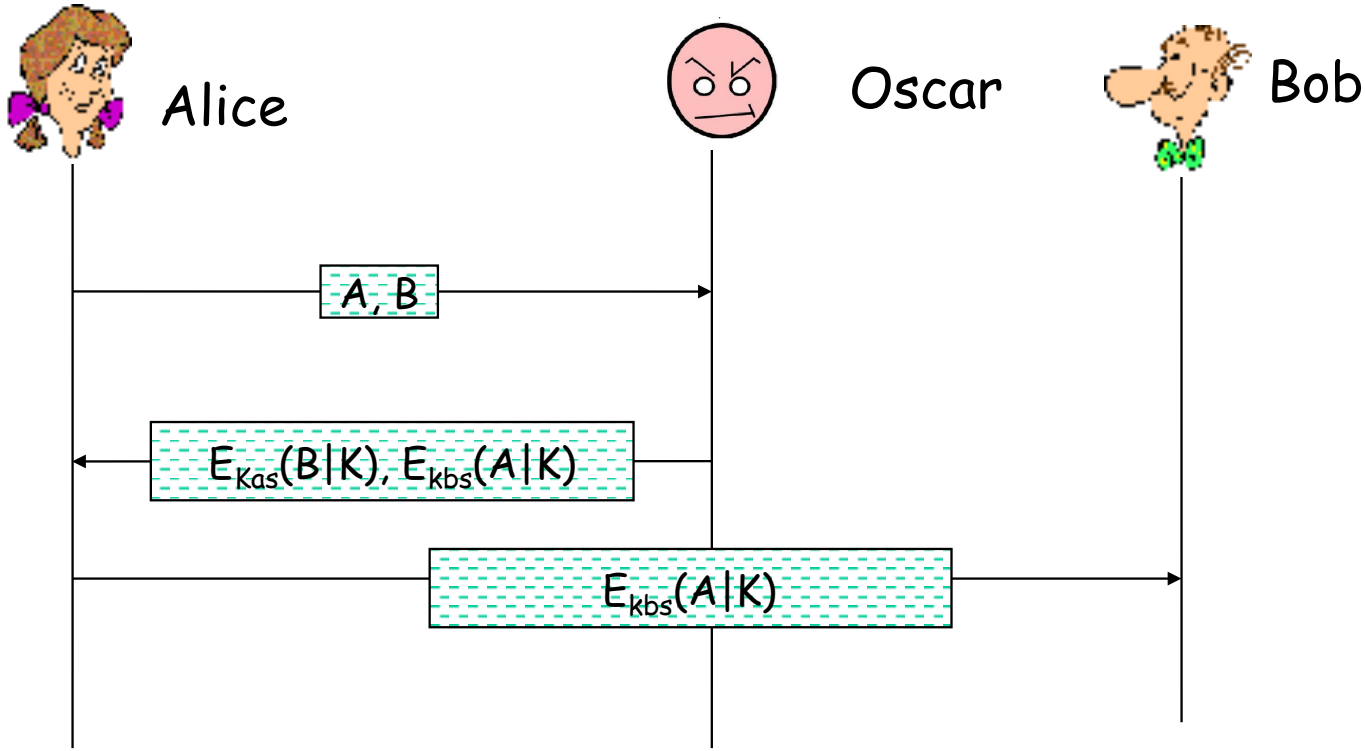
Third attempt



Problem:

- Neither Alice nor Bob can be sure that K is fresh
- No key freshness is provided

Attack against the third attempt

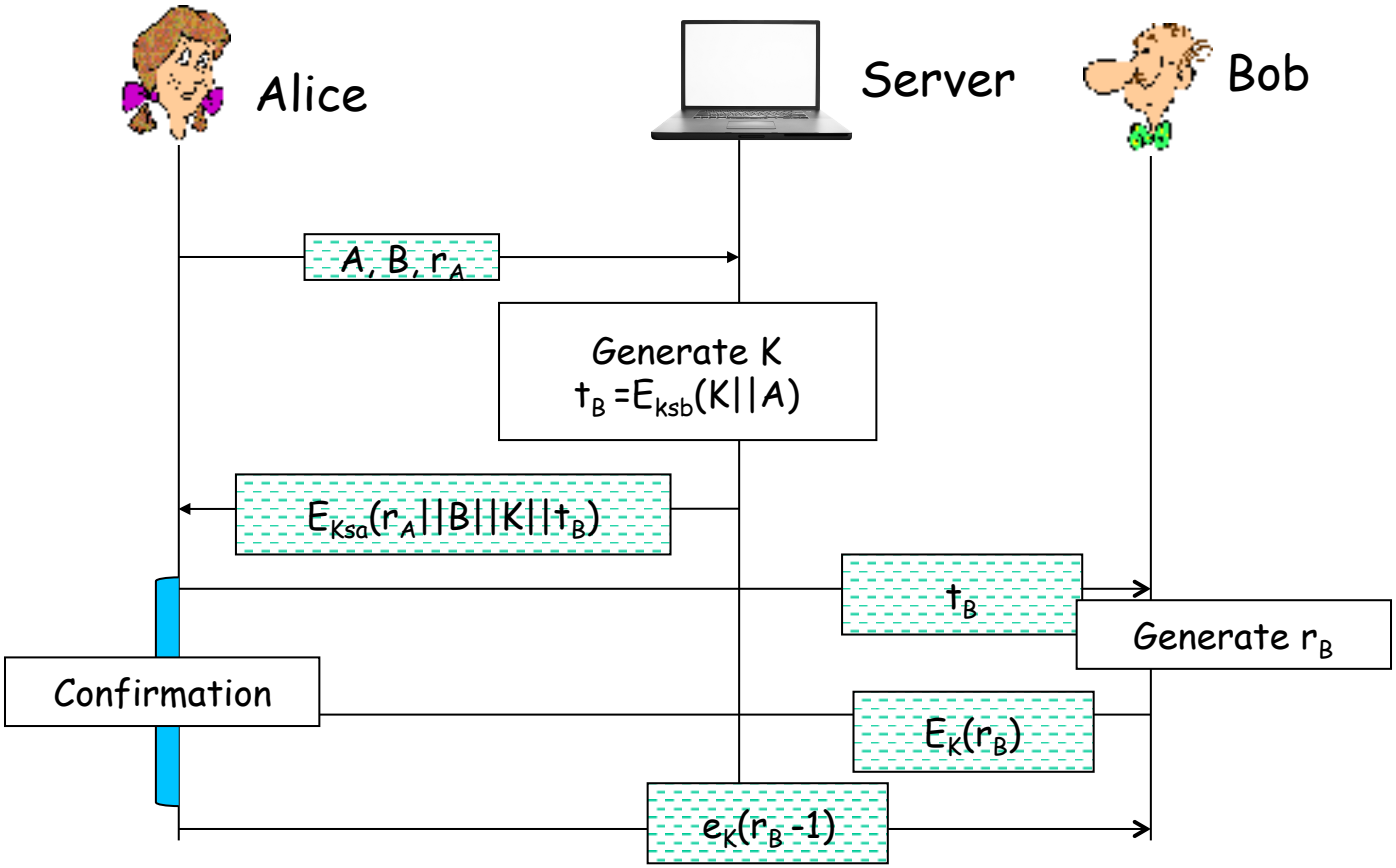


- Replay attack
- If K is compromised by adversary, then she can decrypt follow up information between Alice and Bob
- Even if K is not compromised, the adversary can replay encrypted messages to Alice and Bob from past session where K was used.

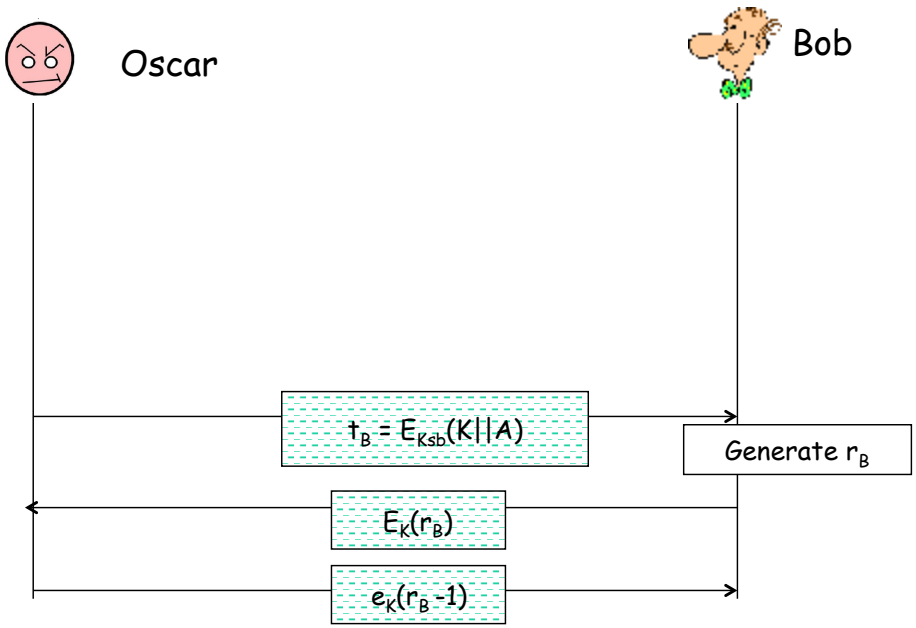
Key features in existing protocols

- Security services
  - Implicit key authentication
  - Key freshness
  - Key confirmation
- Reciprocity: guarantees are provided unilaterally and mutually
- Efficiency
  - Number of messages exchanged
  - Number of bits exchanged
  - Complexity of computation by each party
  - Possibility of pre-computation
- Third party requirement: If required and type of trust

Needham-Schroeder Prorocol

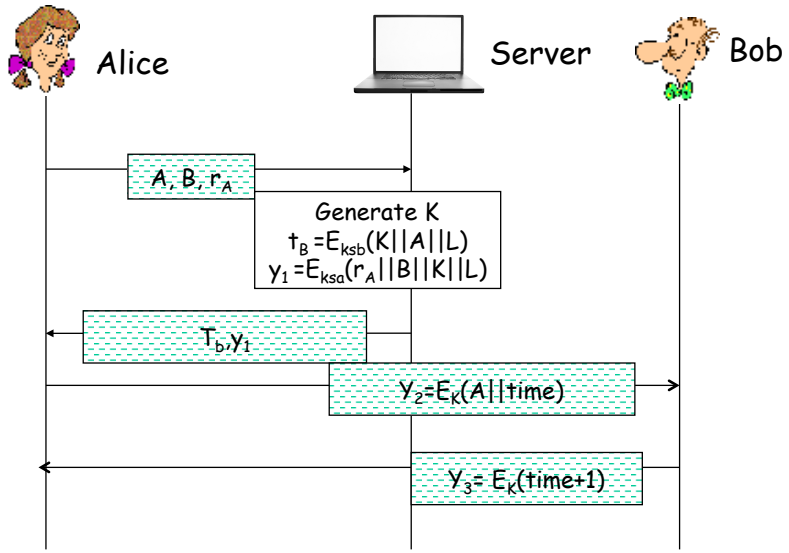


Attack on Needham-Schroeder Prorocol



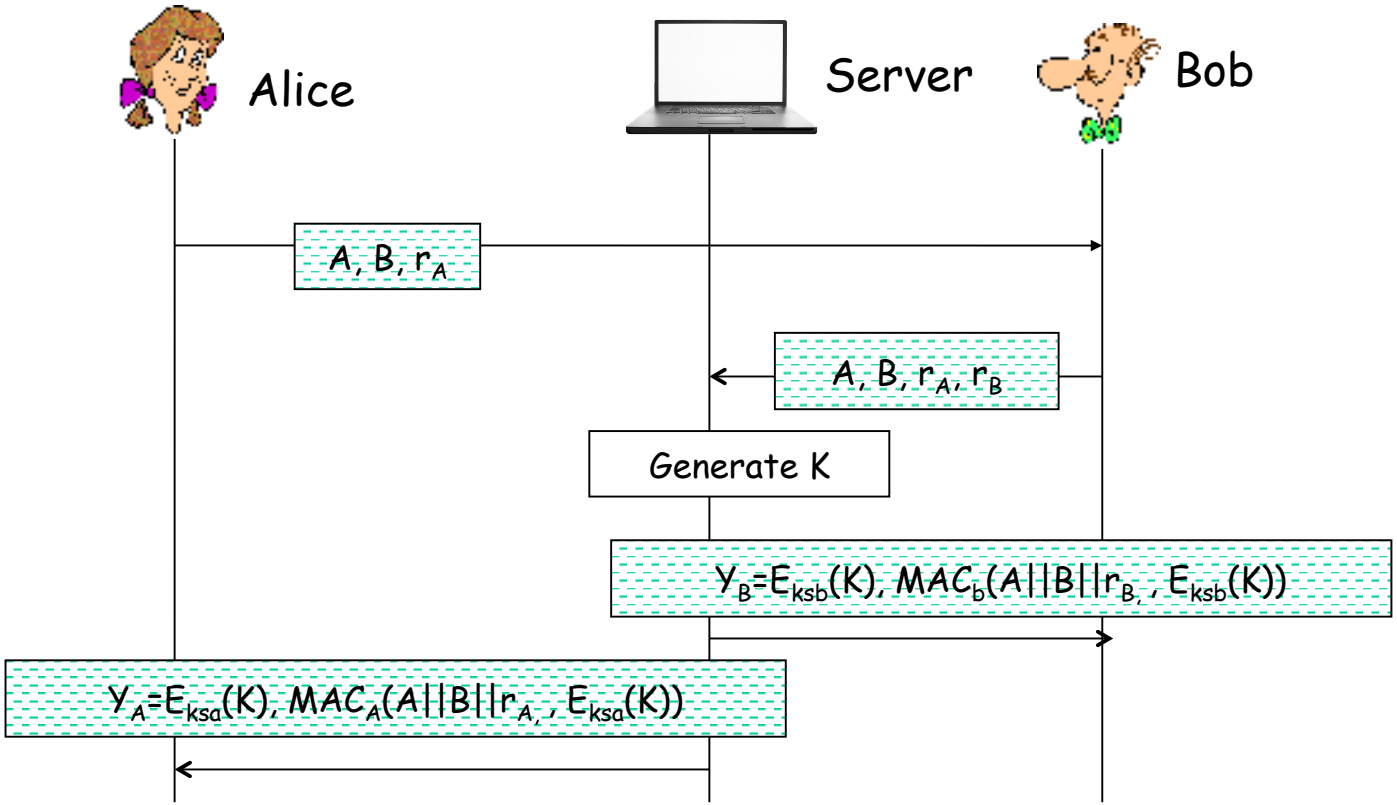
- Oscar knows the session key K of a particular session.
- Known session key attack.
- Oscar initiates a new session S' with Bob.
- Problem 1: The key K is not known to Bob's intended peer, Alice for the session S'.
- Problem 2: The key for session S' is known to someone other than Alice, ie, Oscar.

Kerberos Prorocol



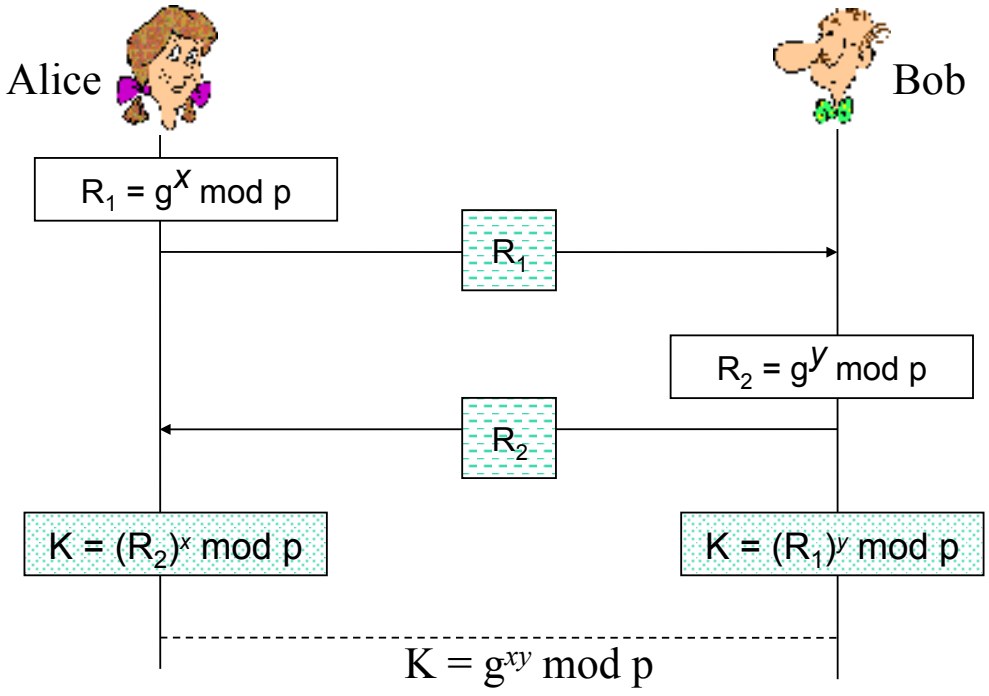
- L is the lifetime, during which K is valid.
- Alice checks if $time \leq L$.
- This prevent replay attacks, gives partial protection from the previous attack.
- In Needham-Schroeder's protocol t_b is doubly encrypted, which is not required.
- Timestamps need time synchronization which is not easy to achieve.

Bellare-Rogaway scheme



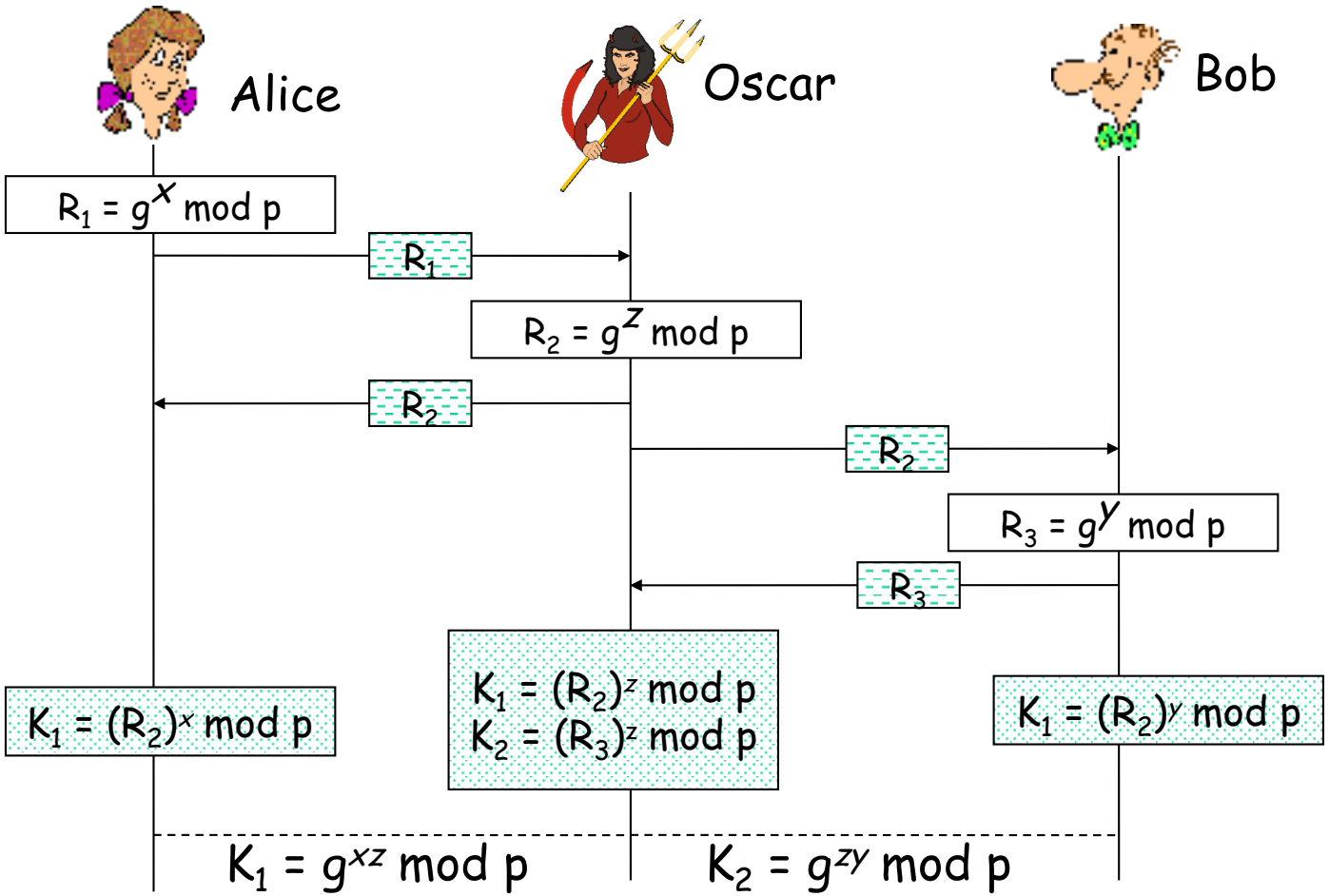
Diffie Hellman Protocol

- p is a large prime, g is a generator of Z_p^*, both are publicly known



- Based on the security of Discrete logarithm problem
- Easy to calculate $g^x \mod p$, difficult to obtain x, given $g^x \mod p$

Diffie Hellman Protocol: Attack

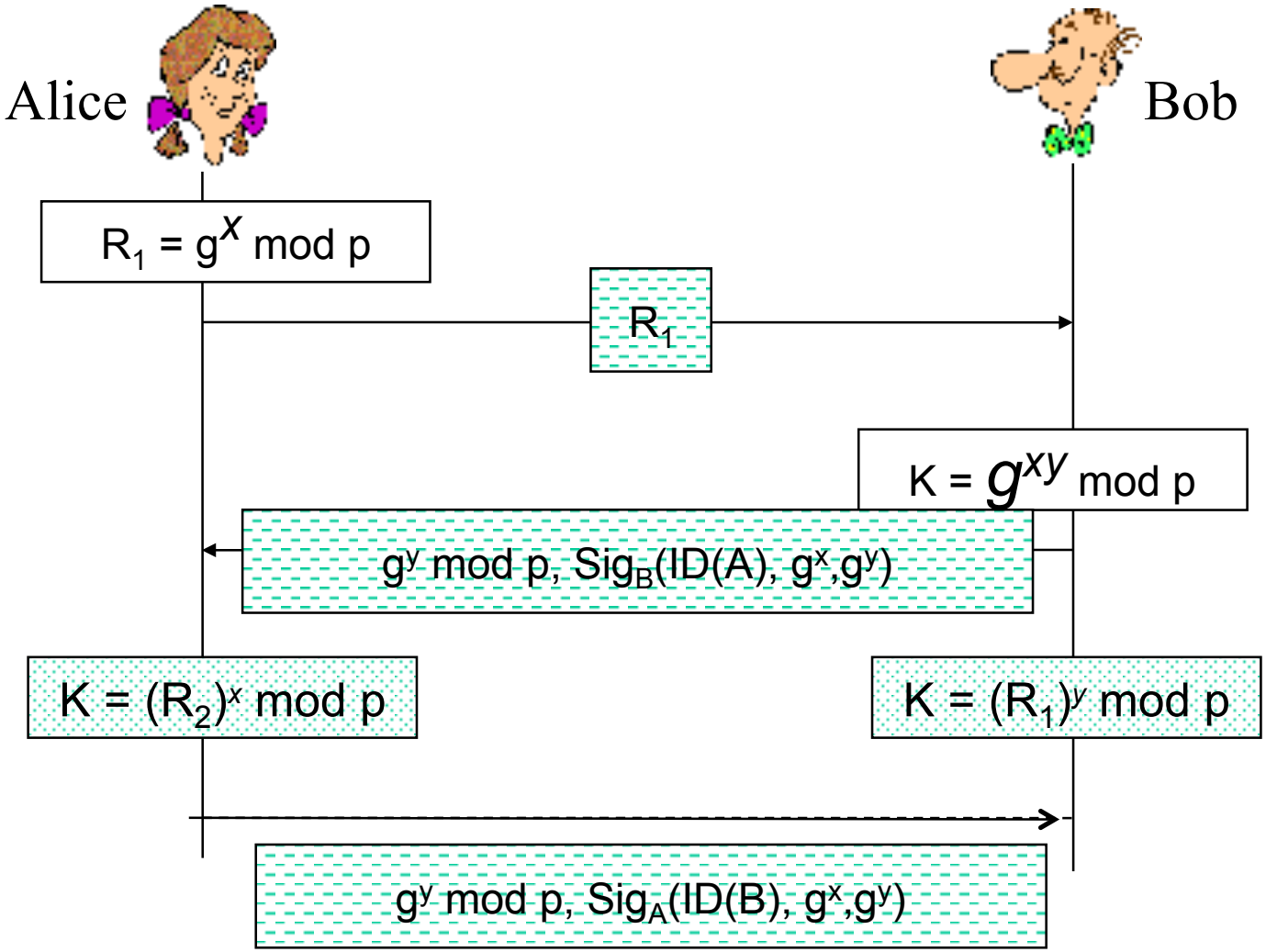


(Wo)man in the middle attack

Features

- No authentication
- Key freshness with randomly selected exponents
- No party can control the key
- No need for TA

Station to Station Protocol



End

Curiouser and curiouser!!