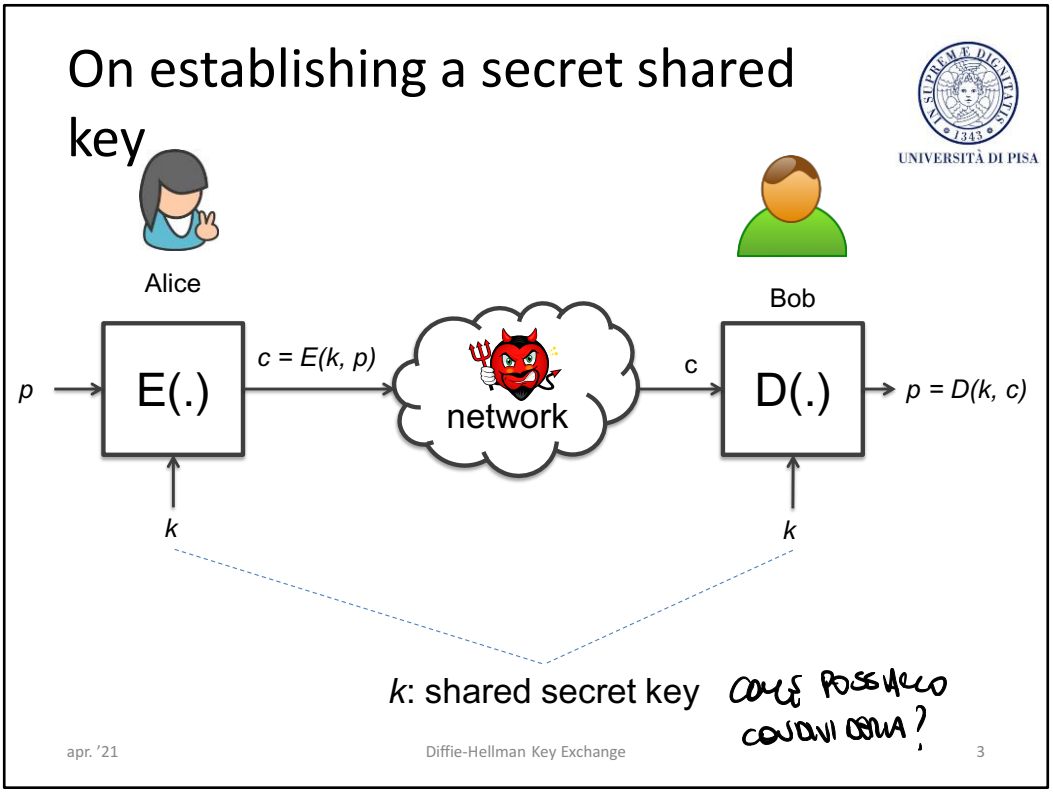
Key Establishment

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
Email: gianluca.dini@.unipi.it
Version: 2021-04-12

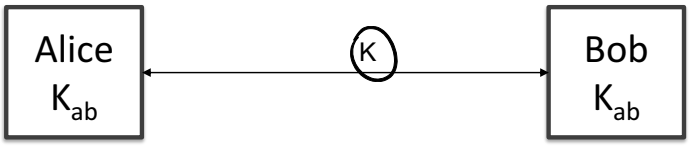
Key Establishment

INTRODUCTION



Symmetric ciphers assume that Alice and Bob have the same key. Alice and Bob have to agree on the same secret key. There must be a way for Alice and Bob to communicate the key to each other without exposing it. They can't send it over the insecure channel. In the old days, keys were distributed physically (off-line distribution. On the Internet this way of distributing keys is not practical. So, we would like to study the problem of Alice and Bob that start communicating without having a shared key. Key establishment refers to cryptographic protocol that makes it possible to establish a secure shared secret between two or more parties.

Session key



- K_{ab} is a long-term secret shared key
- K is temporary session (ephemeral) key

USATA PER CIFRARE

- KEY TRANSPORT
- KEY AGREEMENT

Assume that Alice and Bob share a *long-term secret* key K_{AB}. Typically they do not use it to communicate. They use the key to establishe a temporary *session*, or *ephemeral*, key. The approach consists in using a key for a limited amount of time and then updating it. This approach brings about several advantages: 1) Less damage if a key is exposed; 2) Less cyphertext available for analytical attacks; 3) An adversary has to recover several keys if (s)he is interested in decrypting larger parts of plaintext.
Exchanging a session key may be done in several ways.

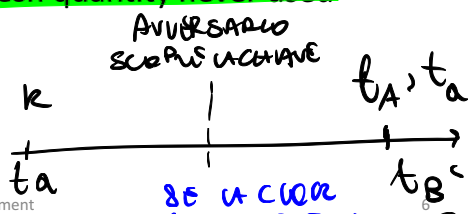
Session key

- Key freshness
  - Use a key for a limited amount of time and then update it
  - Session key or ephemeral key
- Advantages
  - Less damage if a key is exposed
  - Less cyphertext available for analytical attacks
  - An adversary has to recover several keys if (s)he is interested in decrypting larger parts of plaintext

Session transport and agreement

UNIVERSITÀ DI PISA

- One-pass Key transport
  - M1 A → B: $E(K_{ab}, K||t_a)$
  - Where t_a is a timestamp and requires clock synchronization
- Key transport with challenge-response
  - M1 B → A: n_b
  - M2 A → B: $E(K_{ab}, K||n_b)$
  - where n_b is a nonce, i.e., a fresh quantity never used before



Assume that Alice and Bob share a *long-term secret* key K_{AB}. Typically they do not use it to communicate. They use the key to establishe a temporary *session*, or *ephemeral*, key. The approach consists in using a key for a limited amount of time and then updating it. This approach brings about several advantages: 1) Less damage if a key is exposed; 2) Less cyphertext available for analytical attacks; 3) An adversary has to recover several keys if (s)he is interested in decrypting larger parts of plaintext.
Exchanging a session key may be done in several ways.

Session key

- Key agreement
  - M1 B → A: n_b
  - M2 A → B: ~~$E(K', K||n_a||n_b)$~~ $E(K_{ab}, K' || n_a || n_b)$
  - M3 B → ~~A: $E(K'', K||n_a)$~~ $E(K_{ab}, K'' || n_a)$
  - Where n_a and n_b are nonces and K = f(K', K")

$K', K'' \rightarrow K = f(K', K'')$

RANDOM ESSERE

$K = K' \oplus K''$ → K PUO' ESSERE RANDOM ANCHE SE SOLO UNO TRA K' E K'' E' RANDOM

$K = H(K' || K'')$

Assume that Alice and Bob share a *long-term secret* key K_{AB}. Typically they do not use it to communicate. They use the key to establishe a temporary *session*, or *ephemeral*, key. The approach consists in using a key for a limited amount of time and then updating it. This approach brings about several advantages: 1) Less damage if a key is exposed; 2) Less cyphertext available for analytical attacks; 3) An adversary has to recover several keys if (s)he is interested in decrypting larger parts of plaintext.
Exchanging a session key may be done in several ways.

Terminology

- Key Establishment
  - Key Transport
    - One party generates and distributes secret key
  - Key Agreement
    - Parties jointly create a secret key
- Key Establishment is strongly related to identification

Key update

- Run the key establishment protocol
  - Performance impact
- Key derivation function
  - Nonce: random number, counter
  - Cipher or MAC

Key update – random number

Alice (K_{AB})		Bob (K_{AB})
		rnd = RNG()
	< ---------- rnd ----------	
derive key		derive key
$K_{ses} = E_{KAB}(rnd)$		$K_{ses} = E_{KAB}(rnd)$

- HMAC can be used instead of E
- A counter saves the message but requires clock synchronization $K_{ses} = E_{KAB}(cnt++)$

The n^2 Key Distribution Problem

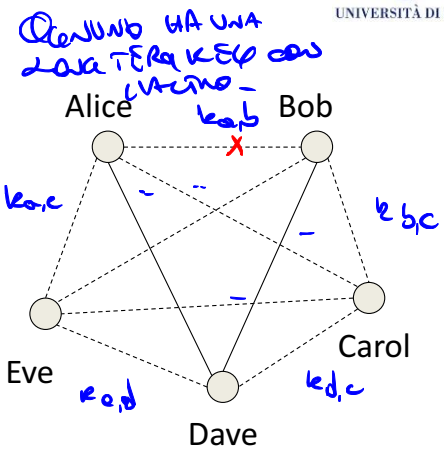
- n users where each party securely communicates with everyone
- Each pair of users shares a long-term secret pairwise key
  - Key pre-distribution
  - Out-of-band transmission

In a pairwise key management scheme, every pair of users share a pairwise key. It follows that Every user has to store n – 1 keys. The total number of keys is: (n-1)+(n-2)+…+1 = n(n-1)/2 that is in the order to n^2.

The n^2 Key Distribution Problem

UNIVERSITÀ DI PISA

- Every user stores (n -1) keys
- There are $\binom{n}{2} = \frac{n \cdot (n-1)}{2}$ symmetric key pairs in the system which is in the order of n^2.



In a pairwise key management scheme, every pair of users share a pairwise key. It follows that
Every user has to store n – 1 keys;
The total number of keys is: (n-1)+(n-2)+…+1 = n(n-1)/2 that is in the order to n^2.

The n^2 Key Distribution Problem

- Pros: Security
  - If a subject is compromised only its communications are compromised; communications between two other subjects are not compromised
  - We cannot do any better!
- Cons: Poor scalability
  - The number of keys is quadratic in the number of subjects
  - A new member's joining/leaving affect all current members

The pairwise scheme has pros and cons. In terms of security, if a subject is compromised only its communications are compromised. The remaining users can keep communicating. However, the scheme scales poorly.

The n^2 Key Distribution Problem

- Pre-distribution does not work for large dynamic networks
- Pre-distribution works for small networks where the number of users does not change frequently
  - E.g., branches of a company SMALL AND STATIC

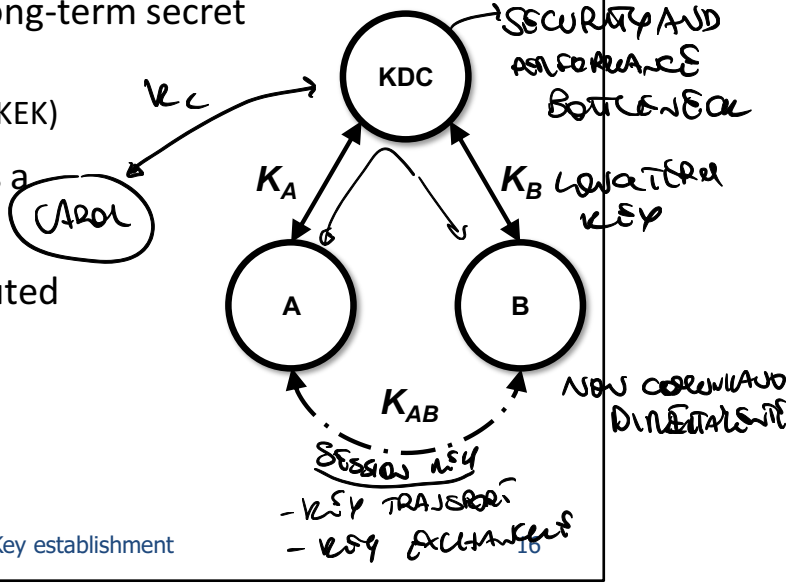
The pairwise scheme has pros and cons. In terms of security, if a subject is compromised only its communications are compromised. The remaining users can keep communicating. However, the scheme scales poorly.

Key Establishment

KEY ESTABLISHMENT USING SYMMETRIC-KEY TECHNIQUES

Key Distribution Center

- Each user shares a long-term secret key with KDC
  - Key Encryption Key (KEK)
- Each KEK constitutes a secure channel
- KEKs are pre-distributed



In a TTP-based scheme, TTP is a *trusted third-party that*

1. Maintain a database <U, K_U>
2. Guarantee integrity and secrecy of the database
3. Correctly play *the key management protocol*

In addition, Each user shares a *long-term, a priori key* with TTP. The overall number of long-term keys is O(n). TTP allows a pair of users to establish a session key (*key management protocol*). When a new member joins (or leaves), we have to update only the TTP (add or delete the member's key: just one key). If a member is compromised, then only its communications are compromised. However, if the TTP is broken the whole system is broken.

Performance and security issues

- Performance: better scalability than pairwise scheme
  - The overall number of KEKs is n
  - Each user stores 1 KEK
  - Each member's joining/leaving only 1 KEK has to be established/removed
- Security
  - If a user is compromised, its communications are compromised
  - If KDC is compromised, all communications are compromised

Key Distribution Center

- KDC is a single point of failure
  - Performance
    - KDC must be available
    - KDC must be efficient
  - Security
    - KDC knows all the keys
    - KDC can read all msg between Alice and Bob
    - KDC can impersonate any party
    - KDC must a trusted third party

Basic KE using KDC (1/2)

Alice	**KDC**	**Bob**
KEK: K_A	KEK: K_A, K_B	KEK: K_B
|-- REQ(A, B) -- >		
	K_{AB} <- RNG()	
	$y_A = E_{KA}(K_{AB})$	
	$y_B = E_{KB}(K_{AB})$	
< ---- y_A ----	---- y_B ---- >	
$k_{AB} = D_{KA}(y_A)$		$K_{AB} = D_{KB}(y_B)$
< ---- $E_{KAB}(session)$ --- >		

The Key Encryption Keys (KEKs) are long-term keys used to encrypt other keys or secrets.

Basic KE using KDC (2/2)

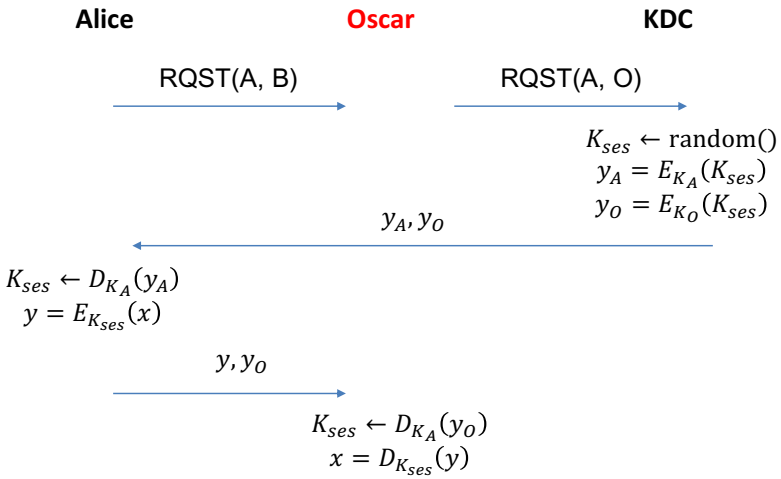
Alice	**KDC**	**Bob**
KEK: K_A	KEK: K_A, K_B	KEK: K_B
|-- REQ(A, B) -- >		
	K_{AB} <- RNG()	
	$y_A = E_{KA}(K_{AB})$	
	$yB = E_{KB}(K_{AB})$	
< ---- y_A , y_B ---|		
$k_{AB} = D_{KA}(y_A)$		
|---------- yB ---------- >		
		$K_{AB} = D_{KB}(y_B)$
< ------- E_{KAB}(*session*) ------ >		

Al alternative communication scheme. From the security standpoint, the tow protocols are equivalent because it is the KDC that communicates witj Alice and Bob by means of yA and yB.
This protocol is simplified and suffers from two problems: replay attack and key confirmation attack.

Security issues

- Replay Attack
  - The adversary records the key establishment protocol
  - The adversary replays y_A and/or y_B
  - The adversary make users to use an old session key
    - An old session can be replied (the session has to be recorded)
    - A compromised session key can be reused
- Key Confirmation attack
  - MIM attack performed by a legitimate but malicious user

Key confirmation attack



Assume that Oscar (the adversary) is able to control Alice's communications.