Key Management

Celia Li
Computer Science and Engineering
York University

Outline

- Fundamental concept
- Key management based on symmetric key cryptography
  - Key transport protocol
  - Key agreement protocol
- Key management based on public key cryptography
  - Key transport protocol
  - Key agreement protocol

Outline

- Fundamental concept
- Key management based on symmetric key cryptography
  - Key transport protocol
  - Key agreement protocol
- Key management based on public key cryptography
  - Key transport protocol
  - Key agreement protocol

Fundamental Concept

- Key Management
  - Provides shared key between two or more parties, typically for subsequent use as symmetric key for a variety of cryptographic purposes including encryption and authentication.
  - Many protocols involve a centralized or trusted third party
  - Broadly subdivided into two categories:
    - key transport
    - key agreement

Fundamental Concept

- Key Transport Protocol
  - One party creates a shared key and securely transfers it to the other(s)
- Key Agreement Protocol
  - A shared key is derived by two (or more) parties as a function of information contributed by each of these, such that no party can predetermine the key value

Key Management Classification

- Key Management based on Symmetric Key Cryptography
  - Key Transport
  - Key Agreement
- Key Management based on Public Key Cryptography
  - Key Transport
  - Key Agreement

Outline

- Fundamental concept
- Key management based on symmetric key cryptography
  - Key transport protocol
  - Key agreement protocol
- Key management based on public key cryptography
  - Key transport protocol
  - Key agreement protocol

Key Transport based on Symmetric Key Cryptography

- Symmetric Key Transport without a Server
  - (1.1) Key transfer using symmetric encryption
  - (1.2) Key transfer using hash function
  - (1.3) Key transport without a priori shared keys
- Server-based Protocols
  - (2.1) Kerberos
  - (2.2) NS (Needham – Schroeder) shared-key protocol

Key Transport based on Symmetric Key Cryptography

- (1.1) Key transfer using symmetric encryption
  - Key transfer with one pass

$$A \rightarrow B : E\,(Kab : r_A)$$

or $$A \rightarrow B : E\,(Kab : r_A, t_A{}^*, B^*)$$

  - r_A: a random number
  - t_A: timestamp
  - n_A: sequence number
  - The shared key: $K=r_A$
  - E: symmetric encryption algorithm
  - * : optional message fields

Key Transport based on Symmetric Key Cryptography

- (1.1) Key transfer using symmetric encryption
  - Key transfer

$$A \leftarrow B : n_B$$
$$A \rightarrow B : E(Kab : r_A, n_B, B^*)$$

  - If it is required that the key K be a function (such as hash function) of input from both parties

$$A \leftarrow B : n_B$$
$$A \rightarrow B : E(Kab : r_A, n_A, n_B, B^*)$$
$$A \leftarrow B : E(Kab : r_B, n_B, n_A, A^*)$$

  - The shared key $K=f(r_A, r_B)$

Key Transport based on Symmetric Key Cryptography

- (1.2) Key transport using hash functions

Define $T = (B, A, r_A, r_B)$.

$A \rightarrow B:$ r_A
$A \leftarrow B:$ $T,\ h_K(T)$ A authenticates B
$A \rightarrow B:$ $(A, r_B),\ h_K(A, r_B)$ B authenticates A
$W = h'_{K'}(r_B)$

- K, K': A and B shared symmetric keys
- h_K , $h'_{K'}$: a keyed one-way hash function used for authentication
- W: final symmetric key

Key Transport based on Symmetric Key Cryptography

- (1.3) Key transport without a priori shared keys
  - A and B select a prime *p*
  - A and B choose random number a and b seperately
  - A chooses a key K sent to B in message 1

$$A \rightarrow B: K^a \bmod p \quad (1)$$
$$A \leftarrow B: (K^a)^b \bmod p \quad (2)$$
$$A \rightarrow B: (K^{ab})^{a^{-1}} \bmod p \quad (3)$$

  - A calculate a^{-1} based on a x a^{-1} = 1mod *p*
  - K mod *p* : the shared key for A and B
  - A know K mod p.
  - B can calculate b^{-1} based on b x b^{-1} = 1mod *p*
  - (3) $(K^{ab})^{a^{-1}} \bmod p$ = K^b mod p → $(K^b \bmod p)^{b^{-1}}$ = K mod p

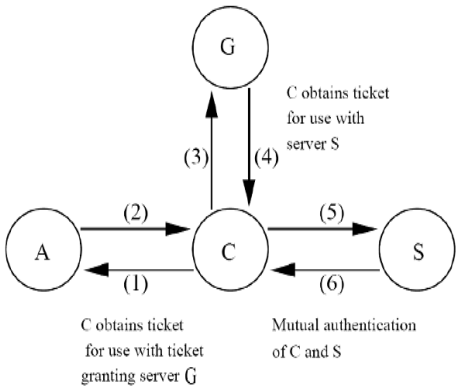
Key Transport based on Symmetric Key Cryptography

Server-based protocols

- (2.1) Kerberos
  - Message 1 & 2:

  C gets the shared key Kcg generated by A
  - Message 3 & 4:

  C gets the shared key Kcs generated by G



Key Transport based on Symmetric Key Cryptography

- (2.2) NS (Needham–Schroeder) shared-key protocol

$$
\begin{aligned}
&(1)\ A \rightarrow S : A, B, Na \\
&(2)\ S \rightarrow A : E(Kas : Na, B, Kab, E(Kbs : Kab, A)) \\
&(3)\ A \rightarrow B : E(Kbs : Kab, A) \\
&(4)\ B \rightarrow A : E(Kab : Nb) \\
&(5)\ A \rightarrow B : E(Kab : Nb - 1)
\end{aligned}
$$

- Message2:
- A gets a shared key Kab generated by S
- Message 3:
- B gets shared key Kab generated by S

Outline

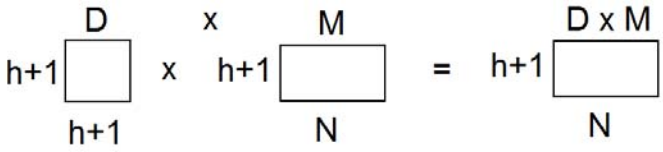
- Fundamental concept
- Key management based on symmetric key cryptography
  - Key transport protocol
  - Key agreement protocol
- Key management based on public key cryptography
  - Key transport protocol
  - Key agreement protocol

Key Agreement on Symmetric Key Cryptography

- Key Agreement Protocol
  - A shared secret is derived by two (or more) parties as a function of information contributed by, or associated with, each of these, such that no party can predetermine the resulting value.
  - Example:
    - Blom's symmetric key system

Key Agreement on Symmetric Key Cryptography

- Blom's symmetric key system
  - Let M be a matrix of size (h+1)*N
    - N: the number of nodes in the network
    - h: a value chosen by a trusted third party
  - Matrix M is a **public information** shared among the participants.
  - In the key generation phase, trusted third party
    - Creates a random *symmetric* matrix D of size (h+1)*(h+1)
    - Computes an matrix $A = (D * M)^T$ *where* $(D * M)^T$ *is the transpose of D*M.*
  - **Matrix A must be kept securely**.



Key Agreement on Symmetric Key Cryptography

- Blom's symmetric key system

▪ If we let *K = A*M, we know that K is a symmetric matrix of size N*N* because of the symmetric property of D.

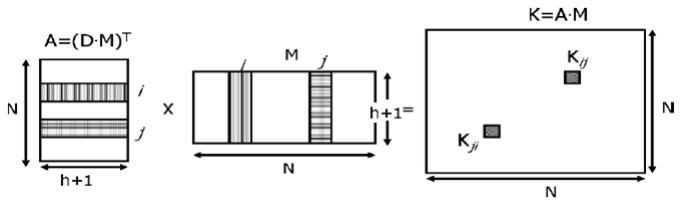
Prove:

(1) $K=A*M = (D*M)^T*M = M^T * D^T *M = M^T * D * M$

(D is a symmetric matrix)

(2) $(A*M)^T = (((D*M)^T*M)^T = M^T * ((D * M)^T)^T = M^T * D * M$

Thus, $K = K^T$



Key Agreement on Symmetric Key Cryptography

- Blom's symmetric key system

How the shared key is generated?

- Private information of participant i and j
  - A(i) and A(j) rows of matrix A
- Public information of participant i and j
  - M(i) & M(j) columns of matrix M
- Shared key of i and j
  - exchange their columns M(i) and M(j)
  - participant i computes*: Kij= A(i)*M(j)*
  - *participant j computes: Kji= A(j)*M(i)*
  - Kij = Kji
  - K is symmetric.

Outline

- Fundamental concept
- Key management based on symmetric key cryptography
  - Key transport protocol
  - Key agreement protocol
- Key management based on public key cryptography
  - Key transport protocol
  - Key agreement protocol

Key Transport based on Public Key Cryptography

- One party chooses a key and transfers it to a second party using that party's public key.
  - (3.1) Key transport using public key encryption without signatures
  - (3.2) Key transport using public key encryption with signatures

Key Transport based on Public Key Cryptography

- (3.1) Key transport using public key encryption without signatures
  - One pass key transport by public key encryption

$A \rightarrow B : E(Kb : k, A)$

Kb : public key of B

K: A encrypts a randomly generated key *k*, and sends the result to B.

Key Transport based on Public Key Cryptography

- (3.2) Key transport protocols using public key encryption with signature
  - (3.2.1) Sign the key, encrypt the signed key using public key
  - (3.2.2) Sign the key, encrypt the unsigned key using public key
  - (3.2.3) Encrypt the key using public key, sign the encrypted key

Key Transport based on Public Key Cryptography

Notation

- *For data input y,*
  - $S_A(y)$ *:* Signature operation on y using A's private key,
- Kb: public key of B

Key Transport based on Public Key Cryptography

- *(3.2.1) Encrypt signed key*

$$A \rightarrow B : E(Kb :\ k,\ t_A{}^*,\ S_A(B, k, t_A{}^*))$$

- *(3.2.2) Encrypt and Sign separately*

$$A \rightarrow B : E(Kb : k,\ t_A{}^*),\ S_A(B, k, t_A{}^*)$$

The asterisk denotes that the timestamp t_A of A is optional

Key Transport based on Public Key Cryptography

- *(3.2.3) Sign the encrypted key*

$$A \rightarrow B :\quad t_A{}^*,\ E(Kb : A, k),\ S_A(B, t_A{}^*,\ E(Kb : A, k))$$

Outline

- Fundamental concept
- Key management based on symmetric key cryptography
  - Key transport protocol
  - Key agreement protocol
- Key management based on public key cryptography
  - Key transport protocol
  - Key agreement protocol

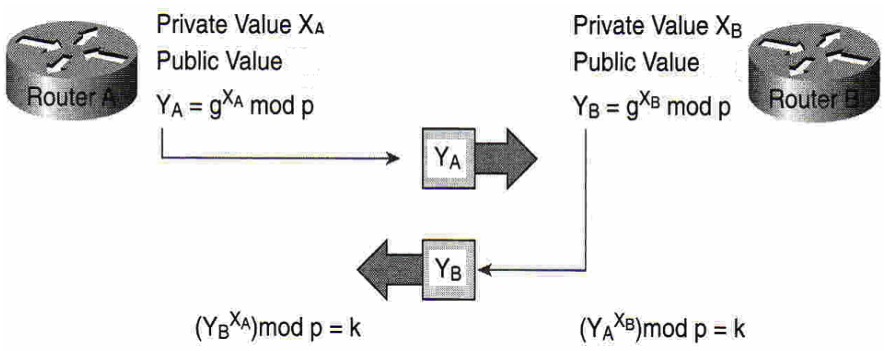
Key Agreement based on Public Key Cryptography

- Diffie-Hellman
  - A protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared key over an insecure communications channel.
  - This key can be used to encrypt subsequent communications using a symmetric key encryption algorithm.

Key Agreement based on Public Key Cryptography

- Diffie-Hellman

  Alice and Bob agree on two prime numbers, g and p.



Reference

- [1] R. *Blom*, "An optimal class of *symmetric key* generation systems", Report LiTH-ISY-I-. 0641, Linköping University, 1984.
- [2] Krawczyk H.: HMQV: A high-performance secure Diffie-Hellman protocol. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 546-566. Springer, Heidelberg (2005). Full version available at http://eprint.iacr.org/2005/176.