Software System Security

Unit 3

Network Protocols

Protocols using shared key cryptography: entity authentication protocols - server-less key establishment - server-based key establishment; authentication and key transport using public key cryptography: Entity authentication protocols - key transport protocols; key agreement protocols: Diffie-Hellman key agreement - Diffie Hellman Protocols with Explicit Authentication

* **Authentication**: concerned with knowing that the corred parties are communicating, allows parties active to learn the identity of other parties.
* **Key Establishment**: concerned with obtaining good cryptographic keys to protect the communications, particularly to provide confidentiality and integrity of the data communicated.
* **Authentication and Key Establishment Setup (AKE)**

Objectives
1. Confidentiality
2. Data Integrity
3. Data Origin Authentication
4. Non-repudiation

→ AKE protocols involve 3 entities (called principals/parties)
(i) 2 users
(ii) Trust server S

→ The aim of the protocol is for A and B to establish a new secret key K_{AB} which they can use for subsequent secure communications.

→ The role of S is to generate K_{AB} and transport it to A and B

* Classification of AKE Protocols

→ Three features regarded as architectural criteria are used to classify different protocols:

(i) which keys are already established

(ii) how new keys are generated

(iii) how many users a protocol is designed to serve

* Session Keys

→ A session key is a symmetric cryptographic key generated for each communication session.

→ It is used to encrypt and decrypt data using a single conversation between two parties. It is a temporary password that resets every time after logging in.

→ During the TLS handshake, the client and server generate session keys at the start of the sessions.

The process involves:

① Client Hello - The client initiates the connection by sending a message to the server.

(2) ServerHello - The server responds with a randomly generated number and its server certificate

(3) The client creates a 'pre-master secret' and encrypts it using the server's public key

(4) Both parties then derive a session key from a combination of random numbers and the pre-master secret.

* Protocols using Shared Key Cryptography

→ Refers to 2-party key establishment & authentication protocols based on symmetric algorithms.

→ These protocols can be classified as follows:

1. The cryptographic keys are available a priori
   - The two principals already share a secret key
   - Each principal shares a key with a trusted server.
2. The method of session key generation
   - key transport
   - key agreement
   - hybrid

Notation

1. A & B → The 2 users who wish to share a new session key
2. S → a trusted server
3. ID_A, ID_B, ID_S → The identities of A, B, S
4. $\{m\}_K$ → authenticated encryption of message m with key K
5. $[\![M]\!]_K$ → encryption of msg m with key K to provide confidentiality
6. $\lceil m \rceil_K$ → One way transformation of message m with key K for integrity

* Entity Authentication Protocols with Shared Key Cryptography

Some protocols are:

1. Bird - Gopal - Herzberg - Janson - Kutten - Molva - Yung Protocols
   (Kryptoknight)
2. Bellare Rogaway MAP1 Protocol
3. ISO / IEC 9798-2 Protocols
   (i) one-pass unilateral authentication protocol
   (ii) two-pass unilateral authentication protocol
   (iii) two-pass mutual authentication protocol
   (iv) three-pass mutual authentication protocol
4. ISO / IEC 9798-4 Protocols
   (i) one-pass unilateral authentication protocol
   (ii) two-pass unilateral authentication protocol
   (iii) two-pass mutual authentication protocol
   (iv) three-pass mutual authentication protocol
5. Woo-Lam Authentication Protocol

1. **Bird - Gopal - Herzberg - Janson - Kutten - Molva - Yung Protocols (Kryptoknight)**

→ a family of lightweight authentication and key distribution protocols, built around a common two-way authentication protocol.

→ They emphasize minimal cryptographic processing & compact message sizes.

→ The basic structure of this protocol is:

1. $A \rightarrow B : N_A$
2. $B \rightarrow A : N_B, u(K_{AB}, N_A \ldots)$
3. $A \rightarrow B : v(K_{AB}, N_B \ldots)$

→ u and v are two functions such that K_{AB} is needed to calculate them.

→ u & v must be different from one another so that A's reply in the parallel session cannot be used by the adversary to complete the first run.

2. Bellare Rogaway MAP1 Protocol

→ Bellare & Rogaway proposed a Session Key Distribution Scheme (SKDS)

→ In this scheme, both Alice and Bob select random challenges that are sent to the TA.

→ As a result Bob is already involved in the scheme before the TA grants the session key.

→ The following are the components of the TA's info to Alice:

(i) a session key that has been encrypted with Alice's secret key

(ii) a MAC tag based on the encrypted session key, Alice & Bob's identities, and Alice's challenge.

1. $A \rightarrow B : N_A$

$B \rightarrow A : N_B, [ID_B, ID_A, N_A, N_B]_{K_{AB}}$

$A \rightarrow B : [ID_A, N_B]_{K_{AB}}$

3. ISO / IEC 9798-2 Protocols

→ The international standard ISO / IEC 9798 Part 2 specifies six protocols using symmetric encryption algorithms.

Out of these:

4 protocols → for entity authentication alone - 2 unilateral and 2 for mutual authentication

2 protocols → for entity authentication & key establishment

① One pass unilateral authentication protocol

→ This protocols consists of a single message from claimant A to verifier B.

→ ~~sender~~. It provides unilateral entity authentication of A to B.

$$A \rightarrow B : \{T_A, ID_B\}_{K_{AB}}$$

→ The timestamp T_A allows B to deduce that A is live, while the inclusion of the identity of B ensures that A has knowledge of B as her peer entity.

② Two-pass unilateral authentication protocol

→ This protocol is similar to the one pass protocol ① , but uses a nonce instead of a timestamp

1. $B \rightarrow A : N_B$
2. $A \rightarrow B : \{N_B, ID_B\} K_{AB}$

③ Two-pass mutual authentication protocol

→ The third protocol uses 2 instances of protocol ① . It provides mutual entity authentication between A & B.

$A \rightarrow B : \{T_A, ID_B\} K_{AB}$

$B \rightarrow A : \{T_B, ID_A\} K_{AB}$

④ Three-pass mutual authentication protocol

→ The fourth protocol is similar to ③ , but uses a nonce instead.

→ However, it does just not use 2 instances of nonces, but rather the second message has both nonces which binds them together.

$B \rightarrow A : N_B$

$A \rightarrow B : \{N_A, N_B, ID_B\} K_{AB}$

$B \rightarrow A : \{N_B, N_A\} K_{AB}$

4. ISO/IEC 9798-4 Protocols

→ The international standard ISO/IEC 9798 Part 4 specifies 4 protocols using a cryptographic check function, i.e a message authentication code (MAC).

→ These protocols are meant to provide entity authentication alone, and they correspond very closely with the first 4 protocols in the ISO/IEC 9798 Part 2 standard.

→ Two protocols are for unilateral authentication, while the other are concerned with mutual authentication

(1) One-pass unilateral authentication protocol

→ consists of a single message from a claimant A to a verifier B.

→ It provides unilateral entity authentication of A to B.

→ The timestamp T_A allows B to deduce that A is live, and the inclusion of the identity B ensures that A has knowledge of B as her peer entity.

$$A \rightarrow B : T_A, MAC_{K_{AB}}(T_A, ID_B)$$

(2) Two-pass unilateral authentication protocol

→ Same as (1), but uses a nonce instead of a timestamp

$$B \rightarrow A : N_B$$
$$A \rightarrow B : MAC_{K_{AB}}(N_B, ID_B)$$

(3) Two-pass mutual authentication protocol

→ constructed from 2 instances of (1)

$$A \rightarrow B : T_A, MAC_{K_{AB}}(T_A, ID_B)$$

$$B \rightarrow A : T_B, MAC_{K_{AB}}(T_B, ID_A)$$

(4) Three-pass mutual authentication protocol:

→ aims at providing mutual authentication, but uses nonces instead of timestamps.

$$B \rightarrow A : N_B$$

$$A \rightarrow B : N_A, MAC(N_A, N_B, ID_B)$$

$$B \rightarrow A : MAC_{K_{AB}}(N_B, N_A)$$

5. Woo-Lam Authentication Protocol

→ Woo-Lam Authentication is a unilateral authentication protocol using a trusted server as a key translation center.

→ The key translation center has the job of converting messages encrypted with one key into messages encrypted with a different key.

→ The idea is that B chooses a nonce N_B and challenges A to encrypt it with K_{AS}.

→ B receives this encryption, and it asks the trusted server to translate it into an encryption with K_{BS}

→ B decrypts this and checks

1. $A \rightarrow B$: ID_A (A sends B A's identity)

2. $B \rightarrow A$: N_B (send A a nonce)

3. $A \rightarrow B$: $\{N_B\}_{K_{AS}}$ (encrypt nonce with K_{AS})

4. $B \rightarrow S$: $\{ID_A, \{N_B\}_{K_{AS}}\}_{K_{BS}}$ (translate encryption using K_{BS})

5. $S \rightarrow B$: $\{N_B\}_{K_{BS}}$ (server sends B the translated version)

* Server-less Key Establishment

→ This refers to protocols that allow keys to be established directly between two users without the use of a server.

→ The protocols considered require that the two users already share a long-term secret key and may require that either one generates the established key (key transport) or that both users contribute part of the established key (key agreement).

Notation : K_{AB} : The long-term key initially shared by A and B

K'_{AB} : The value of the new session key

Some protocols are:

1. Andrew Secure RPC Protocol
2. Janson - Tsudik 2PKDP Protocol

3. Boyd - Two Pass Protocol

4. ISO/IEC 11770-2 Server-Less Protocols

(i) Key Establishment Mechanism 1

(ii) Key Establishment Mechanism 2

(iii) Key Establishment Mechanism 3

(iv) Key Establishment Mechanism 4

(v) Key Establishment Mechanism 5

(vi) Key Establishment Mechanism 6

(1) Andrew ~~RPC~~ Secure RPC Protocol

→ This protocol has 2 independent components

→ In the first 3 messages, A and B perform a handshake using a key they already share, K_{AB}.

→ In the final message, B sends a new session key K'_{AB} to A.

→ The nonce N_A is chosen by A and nonces N_B, N_B' are chosen by B.

1. $A \rightarrow B : \{N_A\}_{K_{AB}}$
2. $B \rightarrow A : \{N_A + 1, N_B\}_{K_{AB}}$
3. $A \rightarrow B : \{N_B + 1\}_{K_{AB}}$
4. $B \rightarrow A : \{K'_{AB}, N_B'\}_{K_{AB}}$

Clark - Jacob Attack on Andrew RPC Protocol

→ Clark and Jacob proposed a typing attack where an intruder records message 2 and substitutes it in place of message 4.

i.e

1. $A \rightarrow B : \{N_A\}_{K_{AB}}$
2. $B \rightarrow A : \{N_A + 1, N_B\}_{K_{AB}}$
3. $A \rightarrow B : \{N_B + 1\}_{K_{AB}}$ sub here
4. $I_B \rightarrow A : \{N_A + 1, N_B\}_{K_{AB}}$

Thus, in order to overcome this; the revised protocol is:

1. $A \rightarrow B : ID_A, N_A$
2. $B \rightarrow A : \{N_A, K'_{AB}\}_{K_{AB}}$ not important
3. $A \rightarrow B : \{N_A\}_{K'_{AB}}$
4. $B \rightarrow A : N'_B$

(2) Janson - Tsudik 2PKDP Protocol

→ This protocol's distinctive aspect is that it employs two separate cryptographic algorithms : one that provides confidentiality and another that provides authentication.

Algo for confidentiality : Bitwise EX-OR

Algo for authentication : MAC algorithm

→ The following steps are followed :

(i) A sends Nonce and ID to B

(ii) B generates a new cession key K'_{AB} and computes values :

ⓐ AUTH = $\lceil N_A, K'_{AB}, B \rceil_{K_{AB}}$

ⓑ MASK = $[\![AUTH]\!]_{K_{AB}}$

i.e

1. $A \rightarrow B : ID_A, N_A$
2. $B \rightarrow A : AUTH, MASK \oplus K'_{AB}$
3. $A \rightarrow B : \lceil N_A, K'_{AB}, A \rceil_{K_{AB}}$

③ Boyd Two Pass Protocol

→ This protocol allows both A and B to contribute to ~~the~~ part of the established key.

→ The messages sent are simply the random numbers chosen by A and B.

1. $A \rightarrow B : N_A$
2. $B \rightarrow A : N_B$

→ The new key is $K'_{AB} = f(N_A, N_B, K_{AB})$ where f is a combining function such that it must be infeasible to find $f(\ldots, \ldots, K_{AB})$ without the knowledge of K_{AB}, even after repeated use.

④ ISO/IEC - 11770-2 Server - Less Protocols

→ The international standard ISO/IEC 11770 part 2 specifies 13 protocols using symmetric encryption standards

6 protocols = server-less

7 protocols = rely on a server

(i) **Mechanism 1**

→ uses only one message pass and provides only implicit key authentication.

→ Mechanism 1 consists only of the encrypted timestamp of A

$$A \rightarrow B : \{T_A\}_{K_{AB}}$$

→ The new session key is derived as : $K'_{AB} = f(K_{AB}, T_A)$

(ii) **Mechanism 2**

→ uses only one message pass and provides only implicit key authentication

$$A \rightarrow B : \{K'_{AB}\}_{K_{AB}}$$

→ The msg. only consist of the encrypted session key. B gains no assurance about its freshness.

(iii) **Mechanism 3**

→ consists of a single message from claimant A to verifier B.

→ The new session key K'_{AB} is chosen by A and encrypted.

→ A and B achieve the good key property, but only B gets key confirmation (IDB is passed)

$$A \rightarrow B : \{T_A, ID_B, K'_{AB}\}_{K_{AB}}$$

(iv) Mechanism 4

→ Mechanism 4 uses a nonce instead of a timestamp

→ The properties achieved are the same as (iii).

1. $B \rightarrow A : N_B$
2. $A \rightarrow B : \{N_B, ID_B, K'_{AB}\}_{K_{AB}}$

(v) Mechanism 5

→ Mechanism 5 is constructed from 2 instances of (iii)

→ Both A and B choose keying material, F_{AB} and F_{BA} respectively

→ The session key is derived as $K'_{AB} = f(F_{AB}, F_{BA})$.

1. $A \rightarrow B : \{T_A, ID_B, F_{AB}\}_{K_{AB}}$
2. $B \rightarrow A : \{T_B, ID_A, F_{BA}\}_{K_{AB}}$

(vi) Mechanism 6

→ 4 nonces are used instead of timestamps

→ Keying material is provided from both parties. ~~and the session~~.

→ Inclusion of both nonces in messages 2 and 3 binds the protocol messages together..

1. $B \rightarrow A : N_B$
2. $A \rightarrow B : \{N_A, N_B, ID_B, F_{AB}\}_{K_{AB}}$
3. $B \rightarrow A : \{N_B, N_A, F_{BA}\}_{K_{AB}}$

* Server - based Key Establishment

→ In these protocols, the server, or one or both users has the responsibility for key generation.

Notation

A & B → Two users wishing to establish a session key

S → server

K_{AS}, K_{BS} → long term keys initially shared by A & S and by B & S

K_{AB} → session key to be shared by A & B.

Some protocols are:

1. Needham Schroeder shared key protocol
2. Otway Rees Protocol
3. Kerberos Protocol
4. ISO/IEC 11770-2 server based protocol
5. Wide mouthed frog protocol
6. Yahalom Protocol
7. Janson Tsudik Protocol
8. Bellare Rogaway
9. Woo Lam
10. Giong Key Agreement
11. Boyd Key Agreement
12. Giong Hybrid

① Needham Schroeder Shared Key Protocol

→ This protocol allows two parties, Alice and Bob, to establish a shared secret key over an inseaure communication channel, which they can then use for further secure communication.

→ The protocol involves a trusted third party, called the Key Distribution Center (KDC), which facillitates the key exchange process.

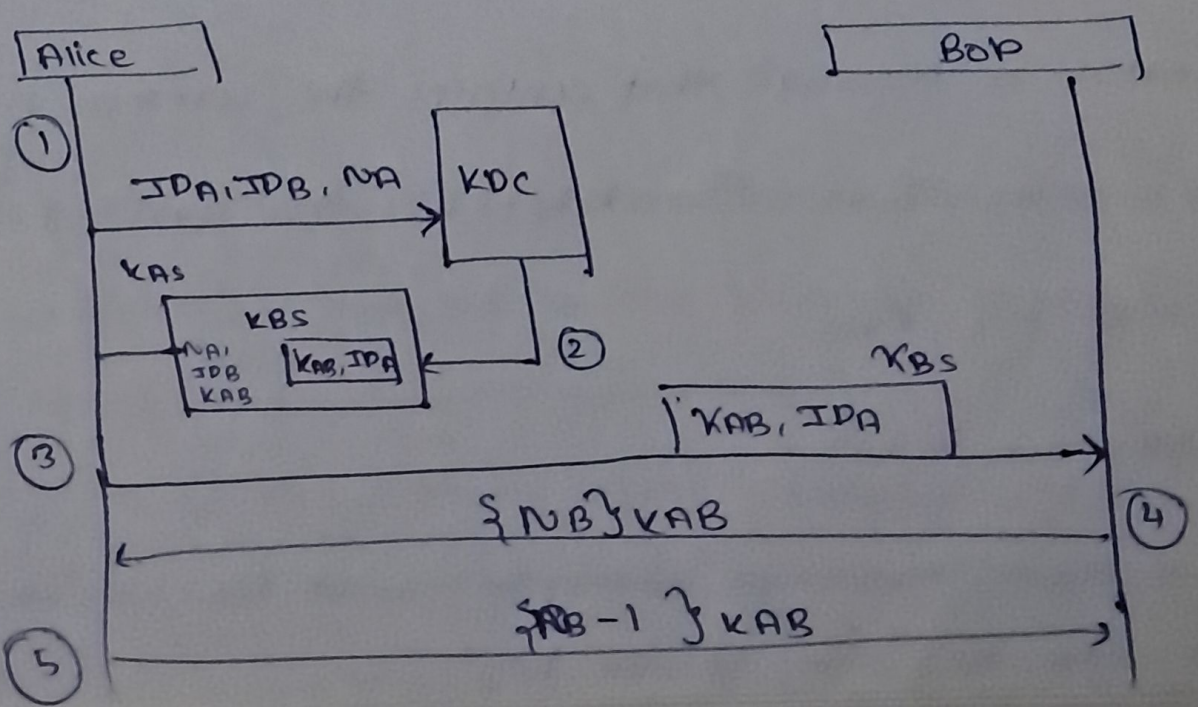
Steps : 1. Alice sends a message to the KDC that includes her nonce, identity and Bob's identity.

2. The KDC sends an encrypted message to Alice that includes Alice's nonce, Bob's identity, the session key, and an encrypted ticket for Bob. The whole message is encrypted with Alice's key.

3. Alice sends Bob's ticket to him

4. Bob sends his challenge to Alice (N_B), encrypted with the session key.

5. Alice responds to Bob's challenge.



1. $A \rightarrow S$: ID_A, ID_B, N_A

2. $S \rightarrow A$: $\{N_A, ID_B, K_{AB}, \{K_{AB}, ID_A\}_{K_{BS}}\}_{K_{AS}}$

3. $A \rightarrow B$: $\{K_{AB}, ID_A\}_{K_{BS}}$

4. $B \rightarrow A$: $\{N_B\}_{K_{BS}}$

5. $A \rightarrow B$: $\{N_B - 1\}_{K_{AB}}$

(2) Otway Rees Protocol

→ aims to provide mutual authentication & session key establishment between two parties, with the help of a trusted KDC.

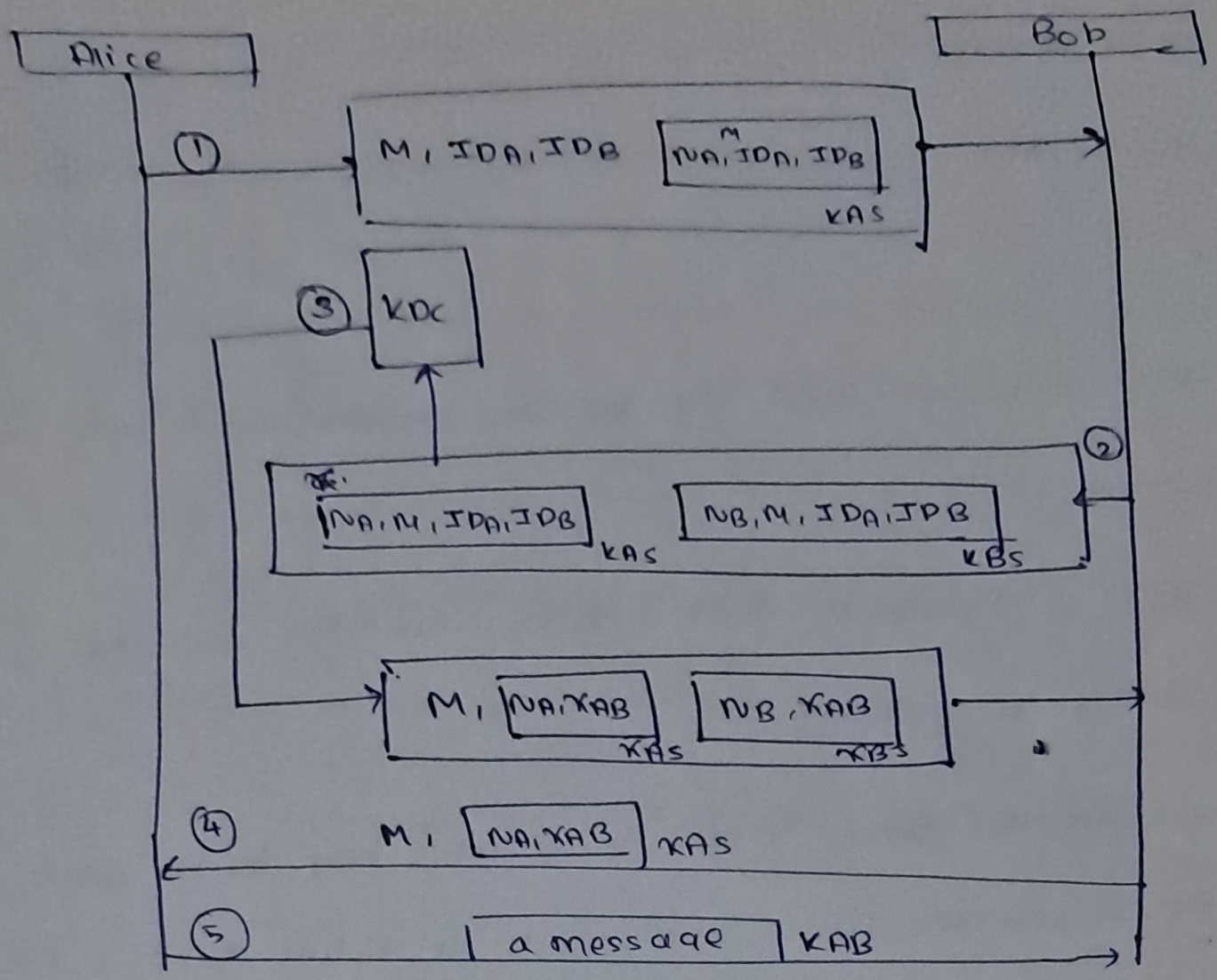
Steps : 1. Alice sends a message to Bob that includes a common nonce (M), the identities of A & B, and a ticket for the KDC that includes Alice's nonce, the common nonce and the identities of A & B

2. B creates the same type of ticket with his nonce, and both tickets are sent to the KDC.

3. The KDC creates a message that contains the common nonce, a ticket for A, and a ticket for B. Each ticket has the corresponding nonce and the session key K_{AB}

4. Bob sends Alice her ticket.

5. Alice sends a short message encrypted with her session key K_{AB} to show that she has the session key.



1. $A \rightarrow B : M, ID_A, ID_B, \{N_A, M, ID_A, ID_B\}_{K_{AS}}$

2. $B \rightarrow S : M, ID_A, ID_B, \{N_A, M, ID_A, ID_B\}_{K_{AS}}, \{N_B, M, ID_A, ID_B\}$

3. $S \rightarrow B : M, \{N_A, K_{AB}\}_{K_{AS}}, \{N_B, K_{AB}\}_{K_{BS}}$

4. $B \rightarrow A : M, \{N_A, K_{AB}\}_{K_{AS}}$

(3) Kerberos

→ It is both an authentication protocol, and at the same time a KDC

→ Three servers are involved in the Kerberos protocol:

(i) an authentication server (AS)

(ii) a ticket granting server (TGS)

(iii) a real data server that provides services to others

Authentication Server

→ The authentication server (AS) is the KDC in the Kerberos protocol.

→ Each user registers with the AS and is granted a user identity and password

→ The AS has a database with these identities and the corresponding passwords

→ The AS verifies the user, issues a session key to be used between Alice and the TGS, and sends a ticket for the TGS.

TGS

→ The ticket granting server issues a ticket for the real server. It also provides the session key K_{AB} between Alice and Bob

→ Kerberos has separated user verification from the issuing of tickets.

→ In this way, though Alice verifies her identity once with the AS, she can contact the TGS multiple times to obtain tickets for different real servers.

Real Server

→ The real server (Bob) provides services for the user (Alice)

→ Kerberos is designed for a client-server program, such as FTP, in which a user uses the client process to access the server process.

Steps : 1. Alice sends her request to the AS in plain text using her registered identity

2. (i) The AS sends a message encrypted with Alice's permanent symmetric key. K_{A-AS}.

(ii) The message has 2 items:

(a) a session key - K_{A-TGS} - that is used by Alice to contact the TGS

(b) a ticket for the TGS that is encrypted with the TGS symmetric key, K_{AS-TGS}.

3. Alice does not know K_{A-AS}, but when the message arrives, she types her symmetric password. The password and the appropriate algorithm together create K_{A-AS} if the password is correct.

4. Alice now sends 3 items to the TGS. The first is the ticket received from the TGS. The second is the name of the real server (Bob) and the third is a timestamp that is encrypted by K_{A-TGS}.

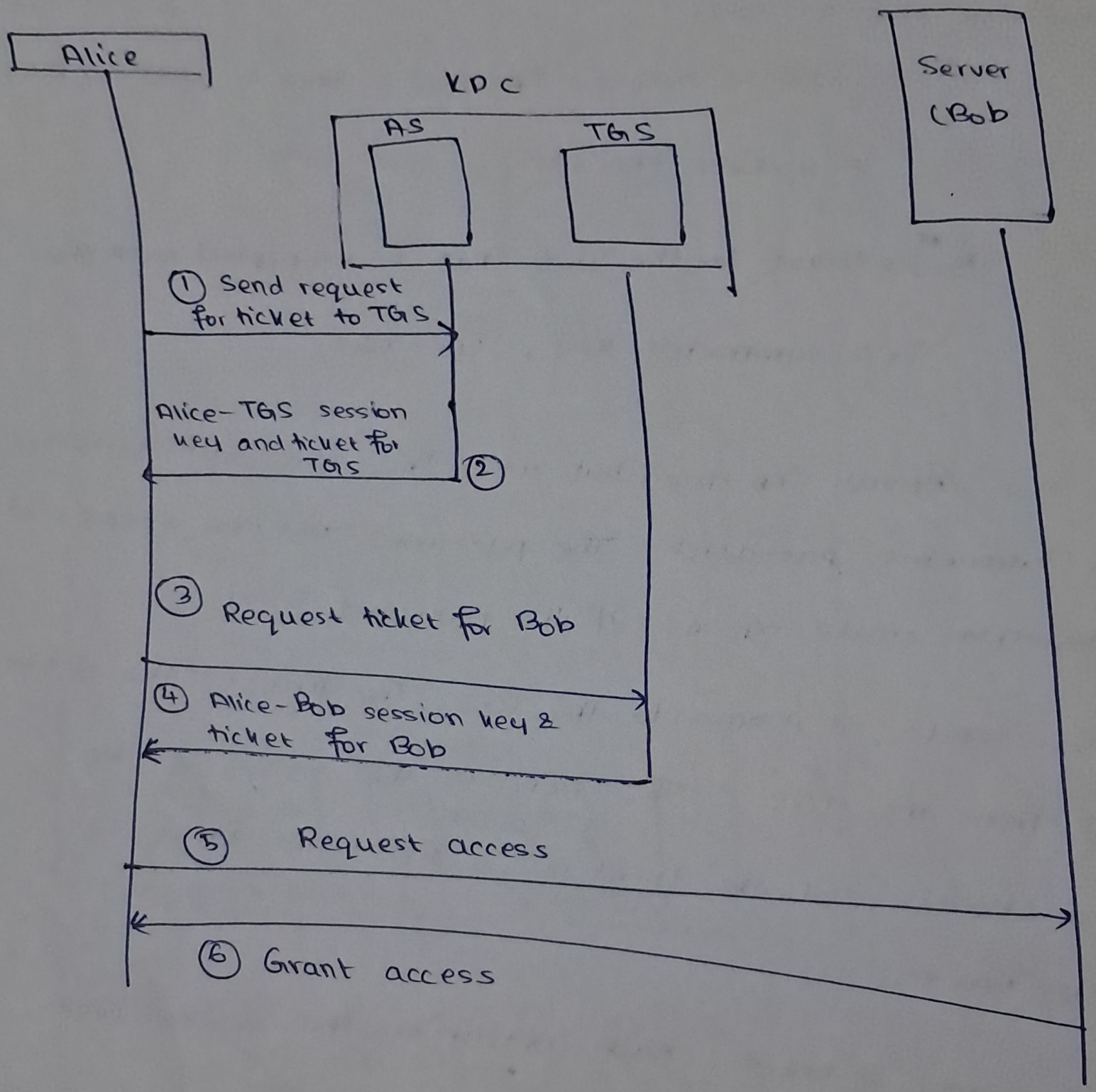
5. The TGS sends 2 tickets, each containing the session key between Alice and Bob K_{AB}.

Alice's key is encrypted with K_{A-TGS}.

Bob's key is encrypted with K_{B-TGS}.

6. Alice sends Bob's ticket with the timestamp encrypted by K_{AB}.

7. Bob confirms the receipt by adding 1 to the timestamp. The message is encrupted with K_{AB} and sent to Alice.



Q3

Alice

KDC

AS

TGS

Server (Bob)

① Alice ID

K_{A-AS}

A-TGS

K_{AS-TGS}

Alice, A-TGS

② Ticket for TGS

③

K_A-TGS

T

Bob

K_{AS-TGS}

Alice, A-TGS

Ticket for TGS

K_{A-TGS}

Bob, AB

K_{TGS-B}

Alice, AB

④ Ticket for Alice

Ticket for B

③

K_{A-B}

T

K_{TGS-B}

Alice, AB

Ticket for Bob

K_{A-B}

T-1

(4) ISO/IEC - 11770 - 2 Server - Based Protocols

→ There are 7 server - based protocols in this standard.

In 4 : the server chooses the session key and acts as a key distribution center.

In 3 : the session key is chosen by either A or B and the server acts as a key translation center to make that key available to the other party.

(i) Mechanism 7

→ The server simply encrypts and sends the K_{AB} to both parties, including the identity of the peer entity.

→ Freshness is not guaranteed.

→ no attack reported

1. $A \rightarrow B : ID_A, N_A$

2. $B \rightarrow S : ID_A, N_A, ID_B, N_B$

3. $S \rightarrow B : \{K_{AB}, ID_A, N_B\}_{K_{BS}}$,

$\{K_{AB}, ID_B, N_A\}_{K_{AS}}$

4. $B \rightarrow A : \{K_{AB}, ID_B, N_A\}_{K_{AS}}$

(ii) Mechanism 10

→ Send an encrypted message to S which checks the authenticity and freshness of the timestamp.

→ Only authentic parties are able to request new keys.

1. $A \rightarrow S : \{T_A, ID_B\}_{K_{AS}}$
2. $S \rightarrow A : \{T_S, K_{AB}, ID_B\}_{K_{AS}}$
3. $S \rightarrow B : \{T_S', K_{AB}, ID_A\}_{K_{BS}}$

→ no attacks reported

(iii) Mechanism 8

→ Principal sends a nonce to S and the key is returned encrypted for A together with A's nonce and the identity of B

→ In addition, the key is encrypted for B together with a timestamp or counter and the identity of principal A.

1. $A \rightarrow S : N_A, ID_B$
2. $S \rightarrow A : \{N_A, K_{AB}, ID_B\}_{K_{AS}}, \{T_S, K_{AB}, ID_A\}_{K_{BS}}$
3. $A \rightarrow B : \{T_S, K_{AB}, ID_A\}_{K_{BS}}$

(iv) Mechanism 9

→ Both principals send their nonces to S and they are returned with the encrypted key and the identity of the peer entity.

1. $A \rightarrow B : ID_A, N_A$
2. $B \rightarrow S : ID_A, N_A, ID_B, N_B$
3. $S \rightarrow B : \{K_{AB}, ID_A, N_B\}_{K_{BS}}, \{K_{AB}, ID_B, N_A\}_{K_{AS}}$
4. $B \rightarrow A : \{K_{AB}, ID_B, N_A\}_{K_{AS}}$

(v) Mechanism 10

→ similar to mechanism 11, but add a nonce for A to check that the key was received by S, and a timestamp for B to check freshness

→ The session key K_{AB} is chosen by A.

1. $A \rightarrow S \quad \{N_A, ID_A, K_{AB}\}_{K_{AS}}$
2. $S \rightarrow A \quad \{N_A, ID_B\}_{K_{AS}}, \{T_S, K_{AB}, ID_A\}_{K_{BS}}$
3. $A \rightarrow B: \quad \{T_S, K_{AB}, ID_A\}_{K_{BS}}$

(vi) Mechanism 13

→ Both parties send their nonces to S, while K_{AB} is chosen by B

1. $A \rightarrow B: N_A$
2. $B \rightarrow S: \{N_A, N_B, ID_A, K_{AB}\}_{K_{BS}}$
3. $S \rightarrow B: \{N_B, ID_A\}_{K_{BS}}, \{N_A, K_{AB}, ID_B\}_{K_{AS}}$
4. $B \rightarrow A: \{N_A, K_{AB}, ID_B\}_{K_{AS}}$

(3) Wide-Mouthed Frog Protocol

→ This protocol is intended for environments where one user trusts the other to choose the key

→ The server simply makes the key chosen by one user available to the other.

→ T_A and T_S are timestamps generated by A & S respectively.

$A \rightarrow S : \quad A, \{T_A, B, K_{AB}\} K_{AS}$

$S \rightarrow B : \quad \{T_S, A, K_{AB}\} K_{BS}$

⑥ Yahalom Protocol

→ uses BAN Logic . A typical BAN logic sequence includes the following 3 steps :

(i) verification of message origin .

(ii) verification of message freshness .

(iii) verification of the origin's trustworthiness .

BAN Logic Components

(i) Belief : represents what the participant believes about the protocol execution

(ii). Assumption : specifies the assumptions made by the participants about the environment.

(iii) Nonce : a unique value used only once to prevent replay attacks

(iv) Intruder : represents an adversary who can intercept and manipulate messages

(v) Secrecy : ensures that only authorized parties can understand the message.

(vi) Authentication . ensures that the communicating parties are who they claim to be.

(viii) Freshness : ensures that nonces & other values are fresh and not replayed.

Protocol

1. $A \rightarrow B$: ID_A, N_A

2. $B \rightarrow S$: $ID_B, \{ID_A, N_A, N_B\}_{K_{BS}}$

3 $S \rightarrow A$: $\{ID_B, K_{AB}, N_A, N_B\}_{K_{AS}}, \{ID_A, K_{AB}\}_{K_{BS}}$

4. $A \rightarrow B$: $\{ID_A, K_{AB}\}_{K_{BS}}, \{N_B\}_{K_{AB}}$

(7) Janson - Tsudik Protocol

3 PKDP

Server-based Key Establishment Protocols (Contd).

7. Janson- Tsudik 3PKDP

The 3PKDP protocol proposed by Janson and Tsudik is a server-based protocol that uses two executes of the 2PKDP as a building block.

The protocol has two executions of the 2PKDP: first between A and S, and then between B and S

It achieves both the good key goal and the mutual authentical goal.

1. $A \rightarrow S: A, B, N_A$
2. $S \rightarrow A: AUTH_A, MASK_A \oplus K_{AB}$
3. $A \rightarrow S: [N_A, K_{AB}, A]_{K_{AS}}$
4. $B \rightarrow S: B, A, N_B$
5. $S \rightarrow B: AUTH_B, MASK_B \oplus K_{AB}$
6. $B \rightarrow S: [N_B, K_{AB}, B]_{K_{BS}}$
7. $A \rightarrow B: A, N'_A$
8. $B \rightarrow A: [N'_A, N'_B, B]_{K_{AB}}, N'_B$
9. $A \rightarrow B: [N'_A, N'_B, A]_{K_{AB}}$

Protocol 3.33: Janson–Tsudik 3PKDP protocol

8. **Bellare Rogaway 3PKD Protocol**

This protocol has key establishment as its goal. The 3PKD uses two distinct cryptographic transformations: a symmetric key encryption algorithm and a MAC.

1. $A \rightarrow B: N_A$
2. $B \rightarrow S: N_A, N_B$
3. $S \rightarrow A: [\![K_{AB}]\!]_{K_{AS}}, [A, B, N_A, [\![K_{AB}]\!]_{K_{AS}}]_{K_{AS}}$
4. $S \rightarrow B: [\![K_{AB}]\!]_{K_{BS}}, [A, B, N_B, [\![K_{AB}]\!]_{K_{BS}}]_{K_{BS}}$

9. **Woo Lam Key Transport Protocol**

1. $A \rightarrow B : N_A$
2. $B \rightarrow A : N_B$
3. $A \rightarrow B : \{A, B, N_A, N_B\}_{K_{AS}}$
4. $B \rightarrow S : \{A, B, N_A, N_B\}_{K_{AS}}, \{A, B, N_A, N_B\}_{K_{BS}}$
5. $S \rightarrow B : \{B, N_A, N_B, K_{AB}\}_{K_{AS}}, \{A, N_A, N_B, K_{AB}\}_{K_{BS}}$
6. $B \rightarrow A : \{B, N_A, N_B, K_{AB}\}_{K_{AS}}, \{N_A, N_B\}_{K_{AB}}$
7. $A \rightarrow B : \{N_B\}_{K_{AB}}$

Protocol 3.36: Woo–Lam key transport protocol

Authentication and Key Transport using Public Key Cryptography

Entity Authentication Protocols

a. ISO/IEC 9798-3

1. $A \rightarrow B : T_A, B, Sig_A(T_A, B)$

Protocol 4.1: ISO/IEC 9798-3 one-pass unilateral authentication

1. $B \rightarrow A : N_B$
2. $A \rightarrow B : N_A, N_B, B, Sig_A(N_A, N_B, B)$

Protocol 4.2: ISO/IEC 9798-3 two-pass unilateral authentication

1. $A \rightarrow B : T_A, B, Sig_A(T_A, B)$
2. $B \rightarrow A : T_B, A, Sig_B(T_B, A)$

Protocol 4.3: ISO/IEC 9798-3 two-pass mutual authentication

1. $B \rightarrow A : N_B$
2. $A \rightarrow B : N_A, N_B, B, Sig_A(N_A, N_B, B)$
3. $B \rightarrow A : N_B, N_A, A, Sig_B(N_B, N_A, A)$

Protocol 4.4: ISO/IEC 9798-3 three-pass mutual authentication

1. $A \rightarrow B : N_A$
1'. $B \rightarrow A : N_B$
2. $A \rightarrow B : N_A, N_B, B, Sig_A(N_A, N_B, B)$
2'. $B \rightarrow A : N_B, N_A, A, Sig_B(N_B, N_A, A)$

Protocol 4.6: ISO/IEC 9798-3 two-pass parallel authentication

b. ISO/IEC 9798-5

- is devoted to entity authentication mechanisms using zero knowledge techniques. Six classes of protocols are specified, depending mainly on the algebraic setting.
- discrete logarithms in the integers modulo a prime;
- discrete logarithms in the integers modulo a composite;
- discrete logarithms in elliptic curve groups;
- the identity-based setting;
- public key encryption;
- integer factorisation.

c. SPLICE/AS

1. $A \rightarrow B : A, B, T_A, L, E_B(N_A), Sig_A(A, T_A, L, E_B(N_A))$
2. $B \rightarrow A : B, A, E_A(B, N_A + 1)$

Protocol 4.7: SPLICE/AS protocol

Key Transport Protocols

a. ISO/IEC-11770-3

1. $A \rightarrow B : E_B(A, K_{AB}, T_A)$

Protocol 4.10: ISO/IEC 11770-3 Key Transport Mechanism 1

1. $A \rightarrow B : B, T_A, E_B(A, K_{AB}), Sig_A(B, T_A, E_B(A, K_{AB}))$

Protocol 4.11: ISO/IEC 11770-3 Key Transport Mechanism 2

1. $A \rightarrow B : E_B(B, K_{AB}, T_A, Sig_A(B, K_{AB}, T_A))$

Protocol 4.12: ISO/IEC 11770-3 Key Transport Mechanism 3

1. $A \rightarrow B : N_A$
2. $B \rightarrow A : A, N_A, N_B, E_A(B, K_{AB}), Sig_B(A, N_A, N_B, E_A(B, K_{AB}))$

Protocol 4.14: ISO/IEC 11770-3 Key Transport Mechanism 4

1. $A \rightarrow B : N_A$
2. $B \rightarrow A : N_B, N_A, A, E_A(B, K_{BA}), Sig_B(N_B, N_A, A, E_A(B, K_{BA}))$
3. $A \rightarrow B : N_A, N_B, B, E_B(A, K_{AB}), Sig_A(N_A, N_B, B, E_B(A, K_{AB}))$

Protocol 4.15: ISO/IEC 11770-3 Key Transport Mechanism 5

1. $A \rightarrow B : E_B(A, K_{AB}, N_A)$
2. $B \rightarrow A : E_A(B, K_{BA}, N_A, N_B)$
3. $A \rightarrow B : N_B$

Protocol 4.16: ISO/IEC 11770-3 Key Transport Mechanism 6

b. Needham Schroeder Public Key Protocol

1. $A \rightarrow B : E_B(N_A, A)$
2. $B \rightarrow A : E_A(N_A, N_B)$
3. $A \rightarrow B : E_B(N_B)$

Protocol 4.19: Needham–Schroeder public key protocol

c. X.509 Standard

There are three protocols specified, with one, two and three message flows respectively. Each protocol extends the previous one by adding an extra message. The goal of each protocol is transport of a session key from A to Band, for the two and three flow protocols, transport of a session key from B to A.

1. $A \rightarrow B : T_A, N_A, B, E_B(K_{AB}), Sig_A(T_A, N_A, B, E_B(K_{AB}))$

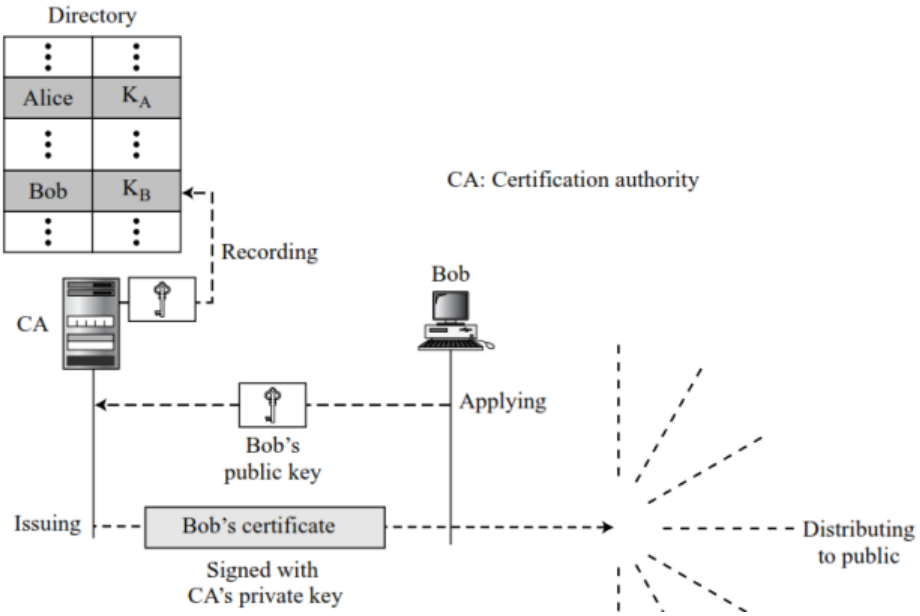
Protocol 4.21: X.509 one-pass authentication

1. $A \rightarrow B : T_A, N_A, B, E_B(K_{AB}), Sig_A(T_A, N_A, B, E_B(K_{AB}))$
2. $B \rightarrow A : T_B, N_B, A, N_A, E_A(K_{BA}), Sig_B(T_B, N_B, A, N_A, E_A(K_{BA}))$

Protocol 4.22: X.509 two-pass authentication

1. $A \rightarrow B : T_A, N_A, B, E_B(K_{AB}), Sig_A(T_A, N_A, B, E_B(K_{AB}))$
2. $B \rightarrow A : T_B, N_B, A, N_A, E_A(K_{BA}), Sig_B(T_B, N_B, A, N_A, E_A(K_{BA}))$
3. $A \rightarrow B : N_B, B, Sig_A(N_B, B)$

Protocol 4.23: X.509 three-pass authentication



d. Beller-Chang-Yacobi Protocols

- proposed hybrid protocols using a combination of asymmetric and symmetric cryptographic algorithms.
- These protocols were designed to satisfy the requirements of the mobile communications environment. They were intended to provide security between a mobile station and a base station of the fixed network, rather than to provide end-to-end security between mobile users.
- KA public key of alice

1. $A \rightarrow B : ID_A, K_A$
2. $B \rightarrow A : \text{Enc}_A(K_{AB}), \{ID_B, SC_B\}_{K_{AB}}$