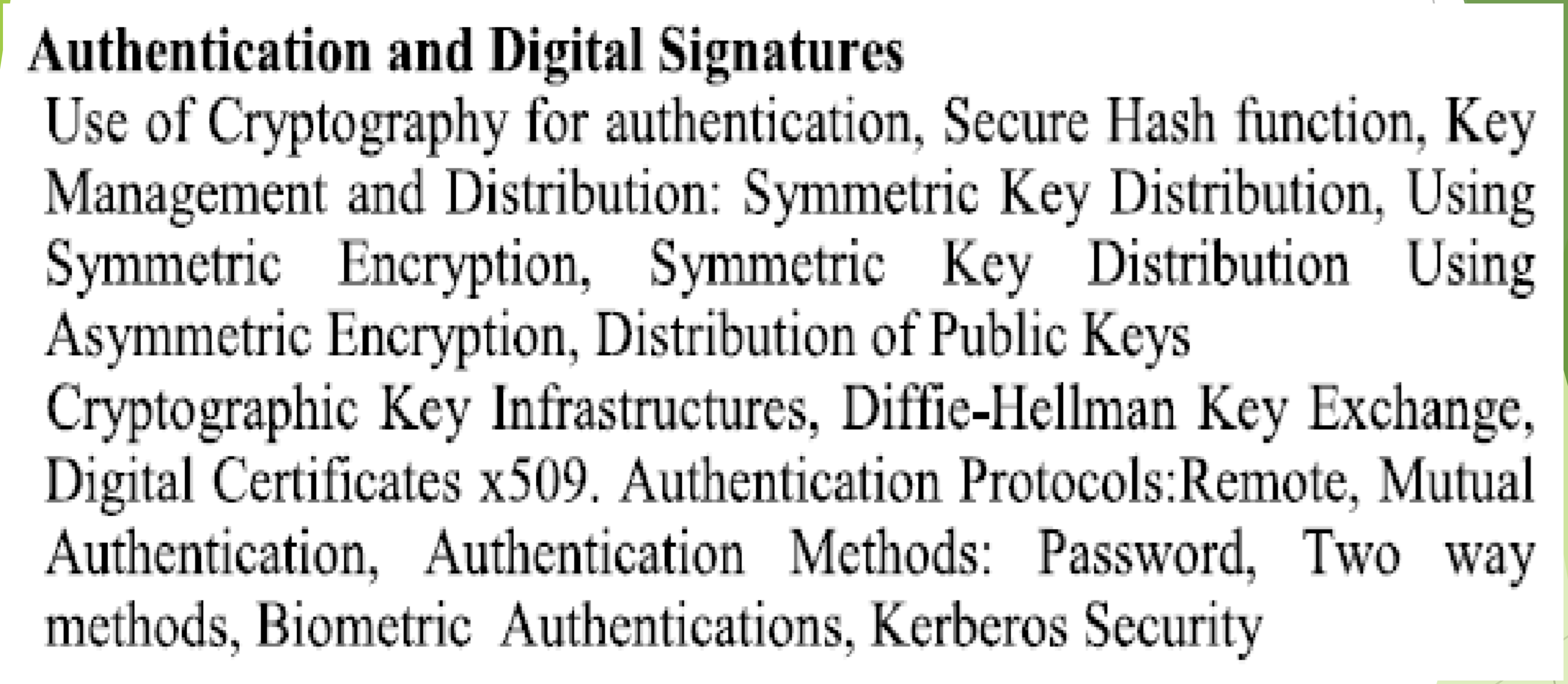
Unit III
Authentication and Digital Signature

Disclaimer

- Information included in these slides came from multiple sources. We have tried our best to cite the sources. Please refer to the references to learn about the sources, when applicable.
- The slides should be used only for preparing notes, academic purposes (e.g. in teaching a class), and should not be used for commercial purposes.

Unit III Syllabus



Authentication and Digital Signatures

Use of Cryptography for authentication, Secure Hash function, Key Management and Distribution: Symmetric Key Distribution, Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys

Cryptographic Key Infrastructures, Diffie-Hellman Key Exchange, Digital Certificates x509. Authentication Protocols:Remote, Mutual Authentication, Authentication Methods: Password, Two way methods, Biometric Authentications, Kerberos Security

Cryptography

Basics

- Cryptography is the science of secret, or hidden writing
- It has two main Components:
  1. Encryption
     - Practice of hiding messages so that they can not be read by anyone other than the intended recipient
  2. Authentication & Integrity
     - Ensuring that users of data/resources are the persons they claim to be and that a message has not been surreptitiously altered

Authentication

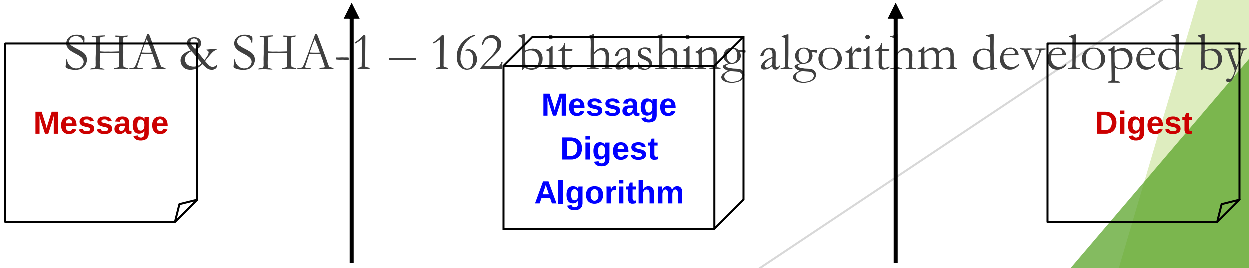
Basics

- Authentication is the process of validating the identity of a user or the integrity of a piece of data.
- There are three technologies that provide authentication
  - Message Digests / Message Authentication Codes
  - Digital Signatures
  - Public Key Infrastructure
- There are two types of user authentication:
  - Identity presented by a remote or application participating in a session
  - Sender's identity is presented along with a message.

Authentication

Message Digests

- A message digest is a fingerprint for a document
- Purpose of the message digest is to provide proof that data has not altered
- Process of generating a message digest from data is called hashing
- Hash functions are one way functions with following properties
  - Infeasible to reverse the function
  - Infeasible to construct two messages which hash to same digest
- Commonly used hash algorithms are
  - MD5 – 128 bit hashing algorithm by Ron Rivest of RSA
  - SHA & SHA-1 – 162 bit hashing algorithm developed by NIST



Message

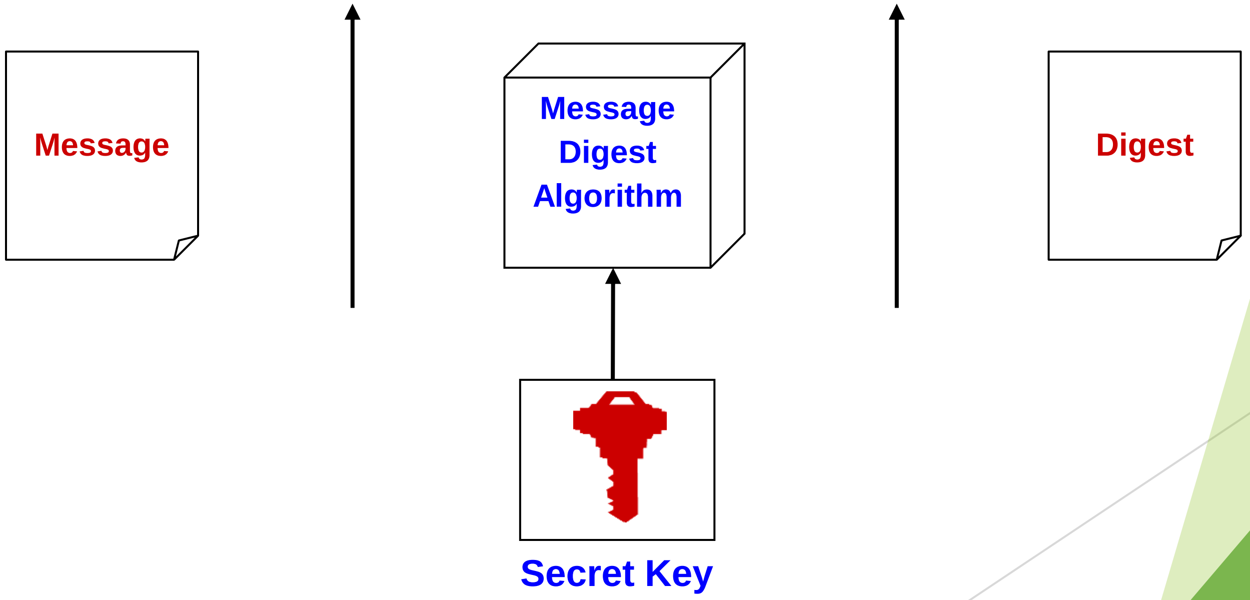
Message Digest Algorithm

Digest

Message Authentication Codes

Basics

- A message digest created with a key
- Creates security by requiring a secret key to be possesses by both parties in order to retrieve the message



Password Authentication

Basics

- Password is secret character string only known to user and server
- Message Digests commonly used for password authentication
- Stored hash of the password is a lesser risk
  - Hacker can not reverse the hash except by brute force attack
- Problems with password based authentication
  - Attacker learns password by social engineering
  - Attacker cracks password by brute-force and/or guesswork
  - Eavesdrops password if it is communicated unprotected over the network
  - Replays an encrypted password back to the authentication server

Authentication Protocols

Basics

- Set of rules that governs the communication of data related to authentication between the server and the user
- Techniques used to build a protocol are
  - Transformed password
    - Password transformed using one way function before transmission
    - Prevents eavesdropping but not replay
  - Challenge-response
    - Server sends a random value (challenge) to the client along with the authentication request. This must be included in the response
    - Protects against replay
  - Time Stamp
    - The authentication from the client to server must have time-stamp embedded
    - Server checks if the time is reasonable
    - Protects against replay
    - Depends on synchronization of clocks on computers
  - One-time password
    - New password obtained by passing user-password through one-way function n times

Authentication Protocols

Kerberos

- Kerberos is an authentication service that uses symmetric key encryption and a key distribution center.
- Kerberos Authentication server contains symmetric keys of all users and also contains information on which user has access privilege to which services on the network

Authentication

Personal Tokens

- Personal Tokens are hardware devices that generate unique strings that are usually used in conjunction with passwords for authentication
- Different types of tokens exist
  - Storage Token: A secret value that is stored on a token and is available after the token has been unlocked using a PIN
  - Synchronous one-time password generator: Generate a new password periodically (e.g. each minute) based on time and a secret code stored in the token
  - Challenge-response: Token computes a number based on a challenge value sent by the server
  - Digital Signature Token: Contains the digital signature private key and computes a computes a digital signature on a supplied data value
- A variety of different physical forms of tokens exist
  - e.g. hand-held devices, Smart Cards, PCMCIA cards, USB tokens

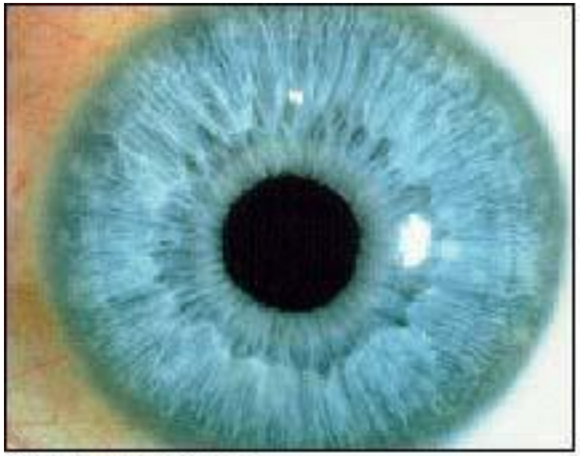
Authentication

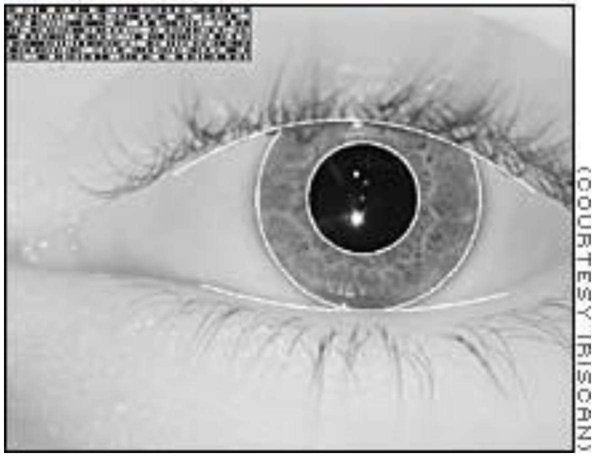
Biometrics

- Uses certain biological characteristics for authentication
  - Biometric reader measures physiological indicia and compares them to specified values
  - It is not capable of securing information over the network
- Different techniques exist
  - Fingerprint Recognition
  - Voice Recognition
  - Handwriting Recognition
  - Face Recognition
  - Retinal Scan
  - Hand Geometry Recognition

Authentication

Iris Recognition





(COURTESY IRISCAN)

The scanning process takes advantage of the natural patterns in people's irises, digitizing them for identification purposes

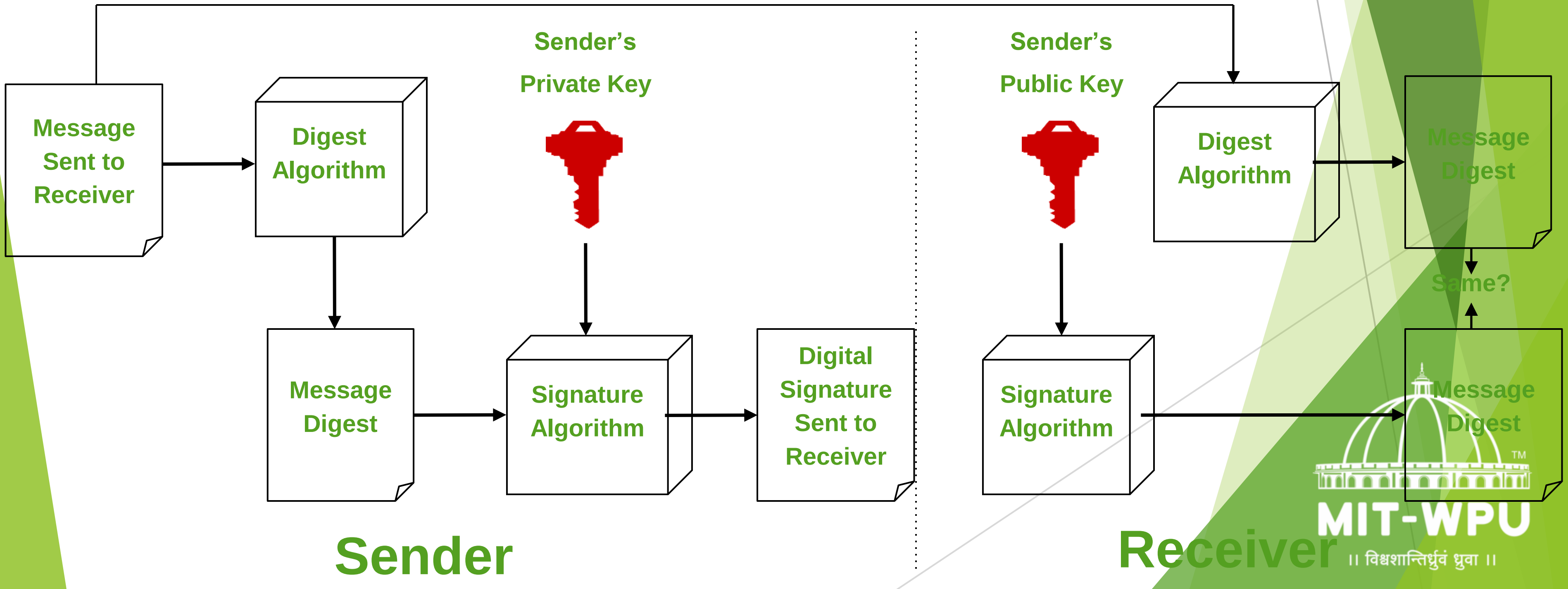
Facts

- Probability of two irises producing exactly the same code: 1 in 10 to the 78th power
- Independent variables (degrees of freedom) extracted: 266
- IrisCode record size: 512 bytes
- Operating systems compatibility: DOS and Windows (NT/95)
- Average identification speed (database of 100,000 IrisCode records): one to two seconds

Authentication

Digital Signatures

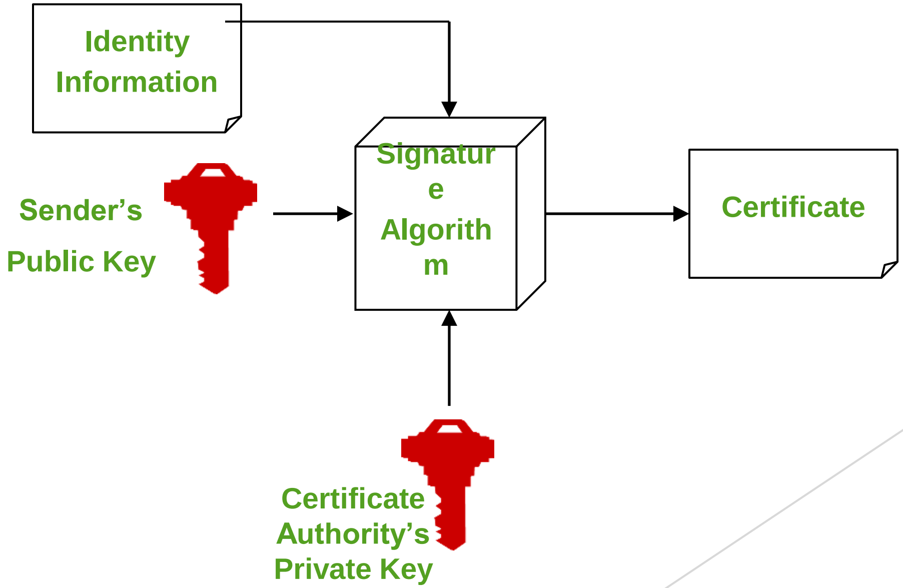
- A digital signature is a data item which accompanies or is logically associated with a digitally encoded message.
- It has two goals
  - A guarantee of the source of the data
  - Proof that the data has not been tampered with



Authentication

Digital Cerftificates

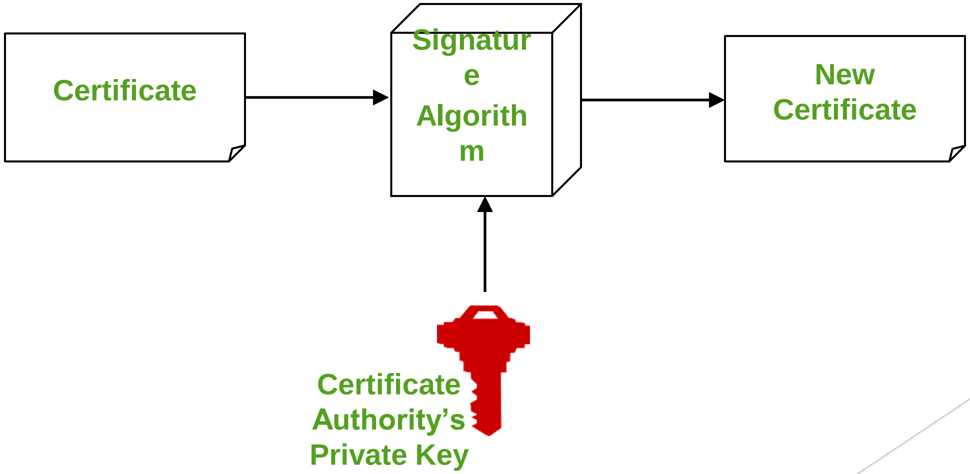
- A digital certificate is a signed statement by a trusted party that another party's public key belongs to them.
  - This allows one certificate authority to be authorized by a different authority (root CA)
- Top level certificate must be self signed
- Any one can start a certificate authority
  - Name recognition is key to some one recognizing a certificate authority
  - Verisign is industry standard certificate authority



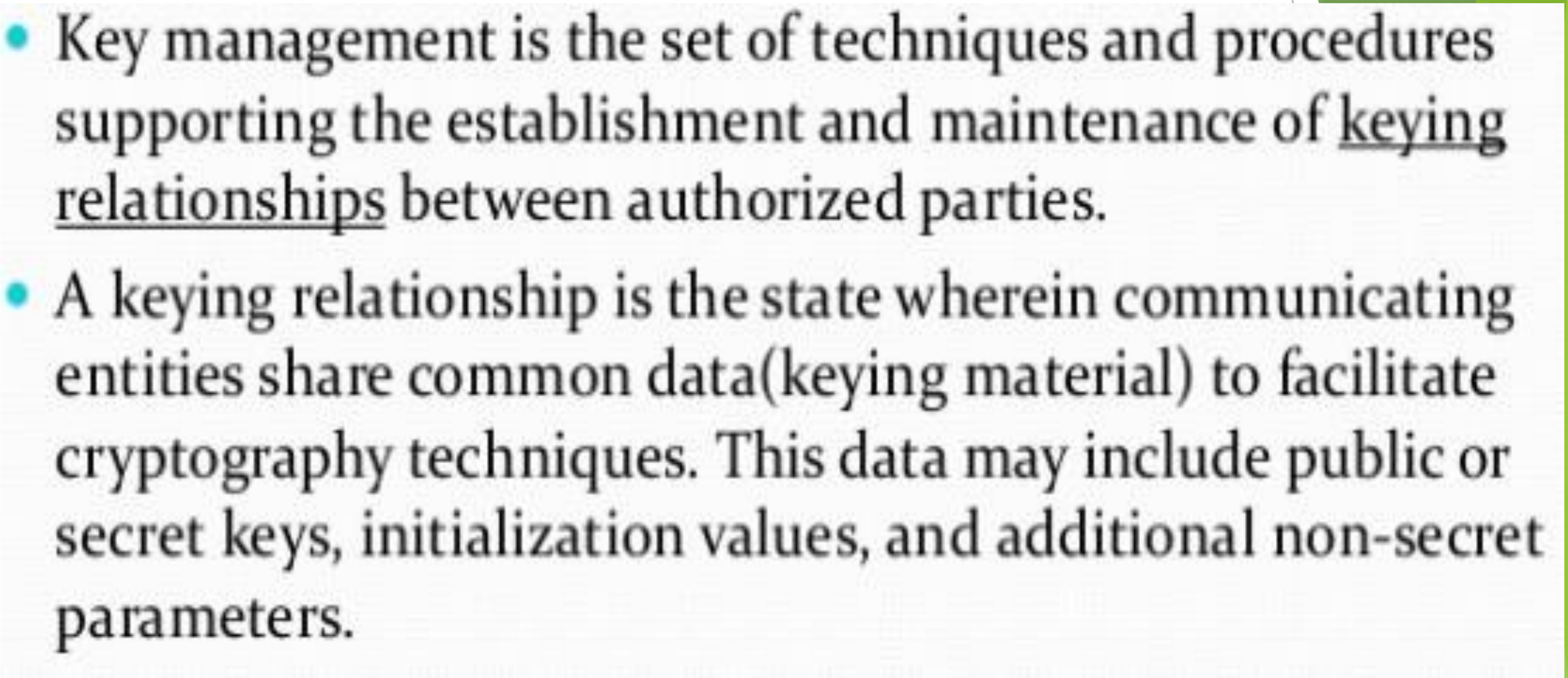
Authentication

Cerftificates Chaining

- Chaining is the practice of signing a certificate with another private key that has a certificate for its public key
  - Similar to the passport having the seal of the government
- It is essentially a person's public key & some identifying information signed by an authority's private key verifying the person's identity
- The authorities public key can be used to decipher the certificate
- The trusted party is called the certificate authority



Key Management



- Key management is the set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties.
- A keying relationship is the state wherein communicating entities share common data(keying material) to facilitate cryptography techniques. This data may include public or secret keys, initialization values, and additional non-secret parameters.

Types of keys

- Session Key
- Master Key

- Public and Private Keys

Key management techniques

- Symmetric Key encryption
- Public-key encryption

Key Management and Distribution

- topics of cryptographic key management / key distribution are complex
  - cryptographic, protocol, & management issues
- Symmetric schemes require both parties to share a common secret key
- Public key schemes require parties to acquire valid public keys
  - issue is how to securely distribute this key
  - whilst protecting it from others

Key Distribution

- symmetric schemes require both parties to share a common secret key
- issue is how to securely distribute this key
- whilst protecting it from others
- frequent key changes can be desirable
- often secure system failure due to a break in the key distribution scheme

Key Distribution using

1. Symmetric Key Distribution Using Symmetric Encryption

2. Symmetric Key Distribution Using Asymmetric Encryption

3. Distribution Of Public Keys

1. Symmetric Key Distribution Using Symmetric Encryption

- ❖ Given parties A and B have various **key distribution** alternatives:

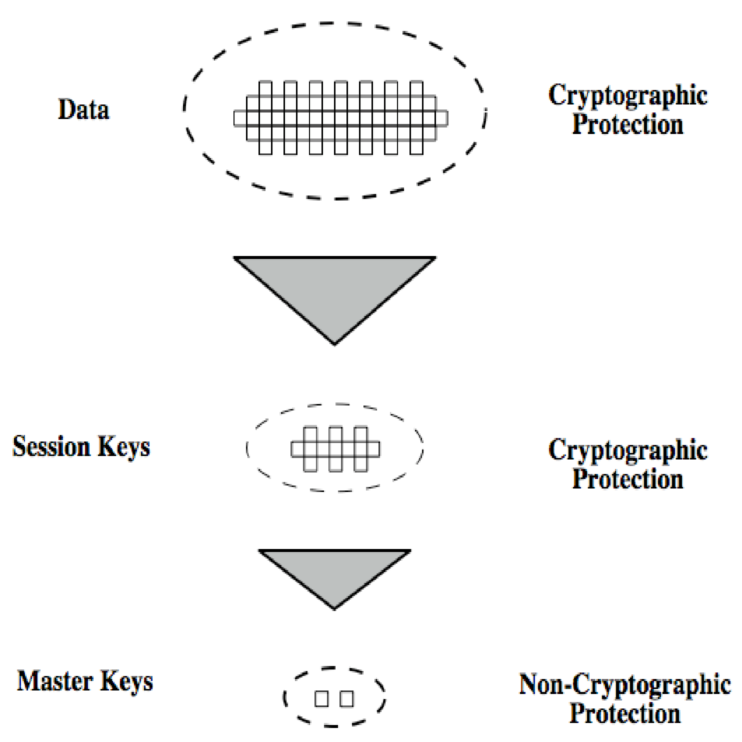
1. A can select key and physically deliver to B
2. Third party can select & physically deliver key to A & B
3. If A & B have communicated previously can use previous key to encrypt a new key
4. If A & B have secure communications with a third party C, C can relay key between A & B

❖ A **key distribution center** is responsible for distributing keys to pairs of users (hosts, processes, applications) as needed.

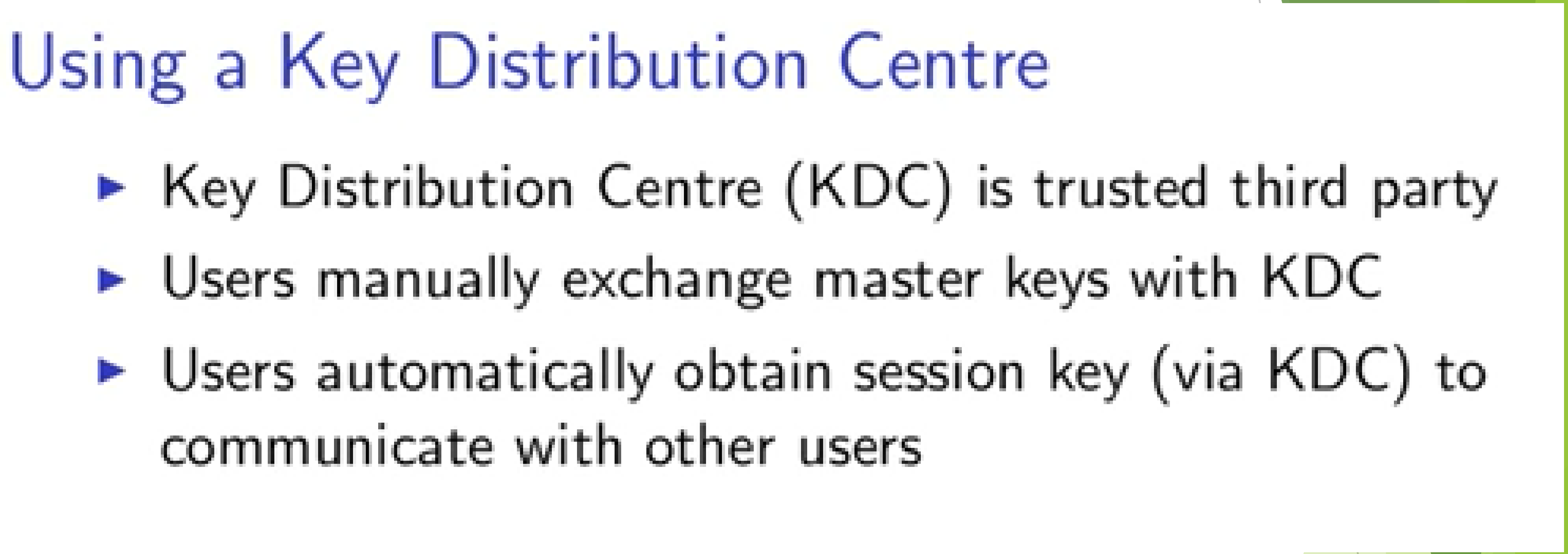
Key Hierarchy

- ❖ The use of a KDC is based on the use of a hierarchy of keys

- ❖ typically have a hierarchy of keys
- ❖ session key
  - temporary key
  - used for encryption of data between users
  - for one logical session then discarded
- ❖ master key
  - used to encrypt session keys
  - shared by user & key distribution center



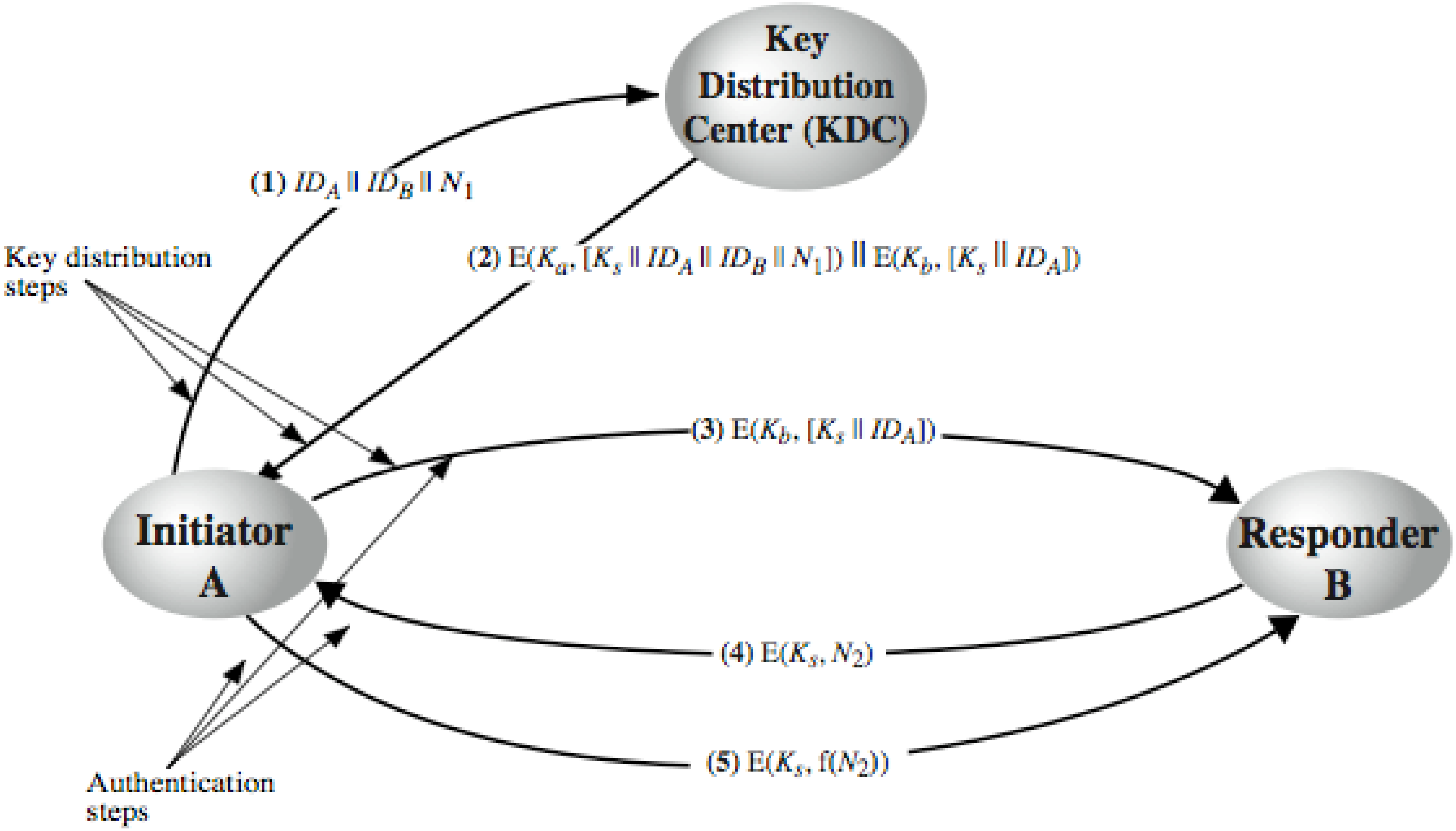
KDC

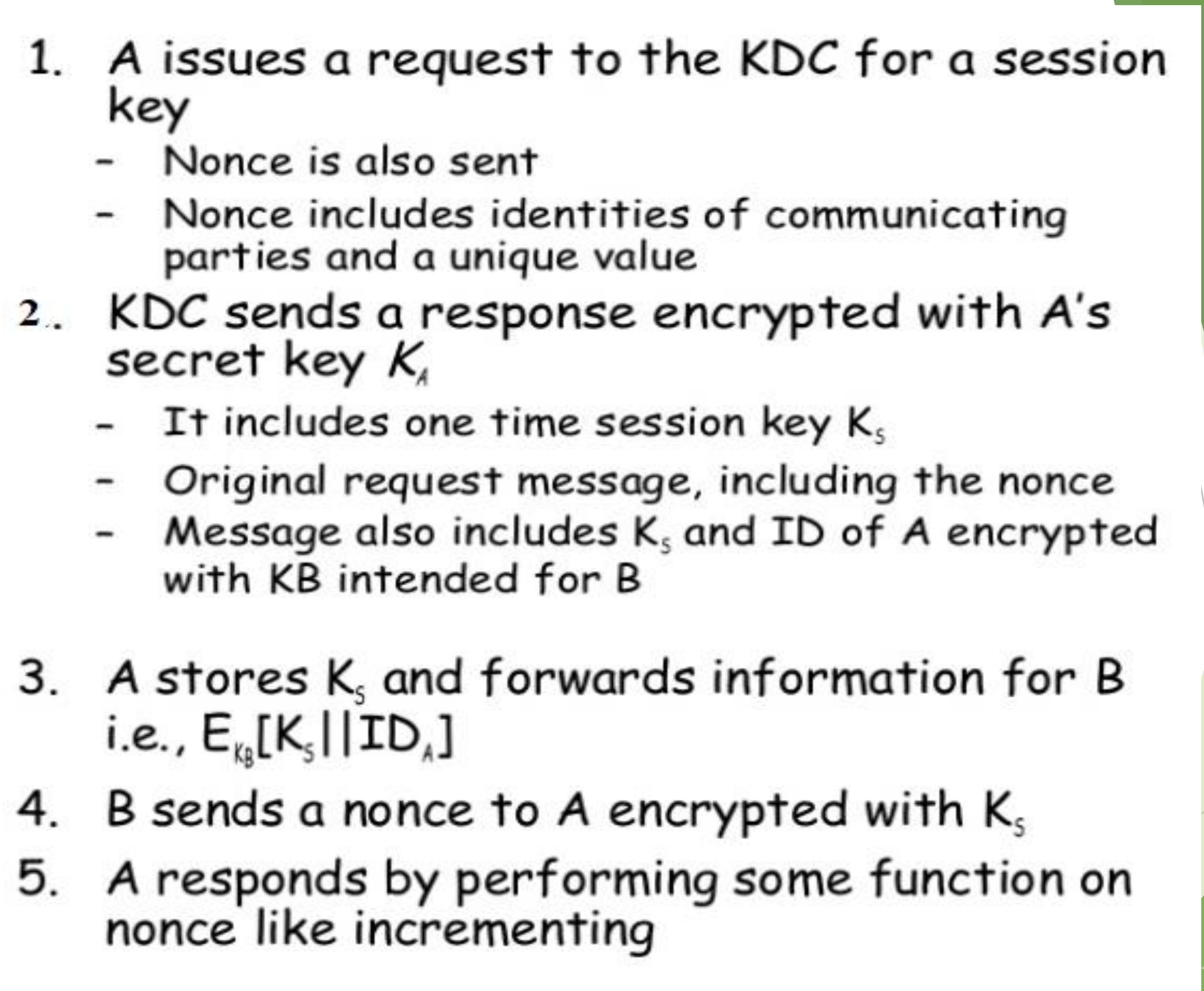


Using a Key Distribution Centre

- Key Distribution Centre (KDC) is trusted third party
- Users manually exchange master keys with KDC
- Users automatically obtain session key (via KDC) to communicate with other users

Key Distribution Scenario


Key Distribution Center (KDC)
(1) $ID_A \| ID_B \| N_1$
Key distribution steps
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s \| ID_A])$
(3) $E(K_b, [K_s \| ID_A])$
Initiator A
Responder B
(4) $E(K_s, N_2)$
(5) $E(K_s, f(N_2))$
Authentication steps



1. A issues a request to the KDC for a session key
   - Nonce is also sent
   - Nonce includes identities of communicating parties and a unique value
2. KDC sends a response encrypted with A's secret key K_A
   - It includes one time session key K_S
   - Original request message, including the nonce
   - Message also includes K_S and ID of A encrypted with KB intended for B
3. A stores K_S and forwards information for B i.e., $E_{K_B}[K_S||ID_A]$
4. B sends a nonce to A encrypted with K_S
5. A responds by performing some function on nonce like incrementing

Key Distribution Issues

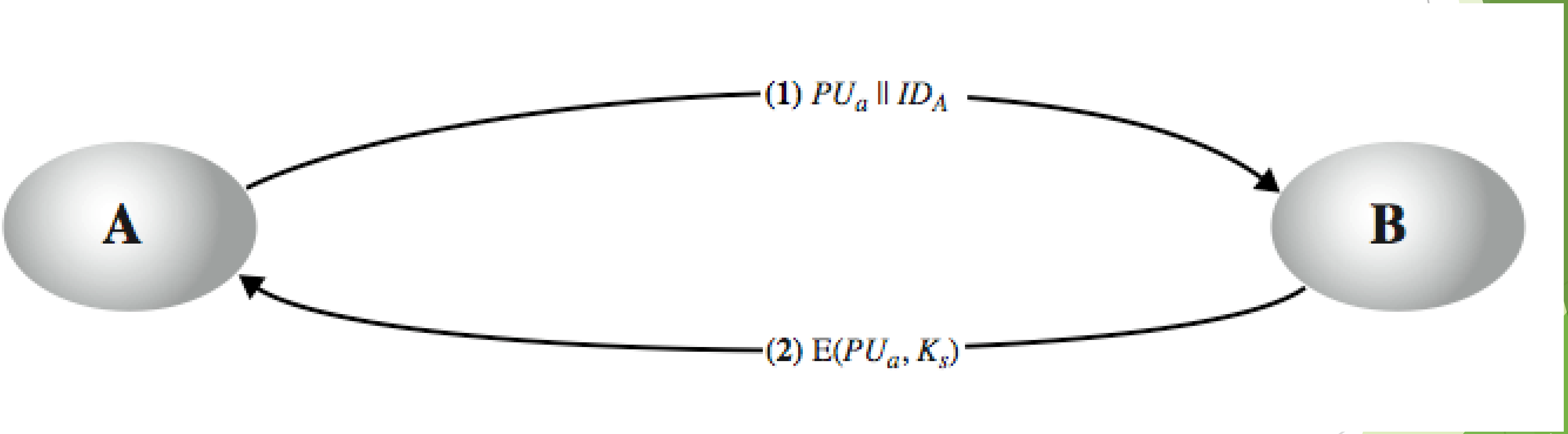
- hierarchies of KDC’s required for large networks, but must trust each other
- session key lifetimes should be limited for greater security
- use of automatic key distribution on behalf of users, but must trust system

Symmetric Key Distribution Using Asymmetric/public Keys

- public key cryptosystems are inefficient
  - so almost never use for direct data encryption
  - rather use to encrypt secret keys for distribution

Simple Secret Key Distribution

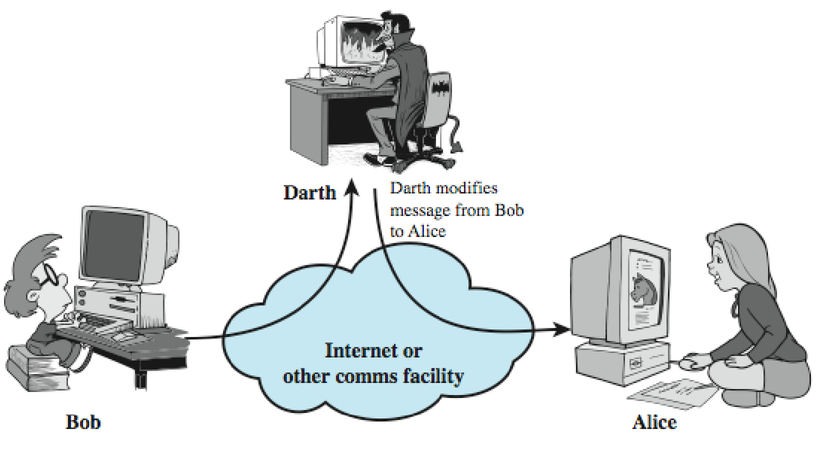
- Merkle proposed this very simple scheme
  - allows secure communications
  - no keys before/after exist



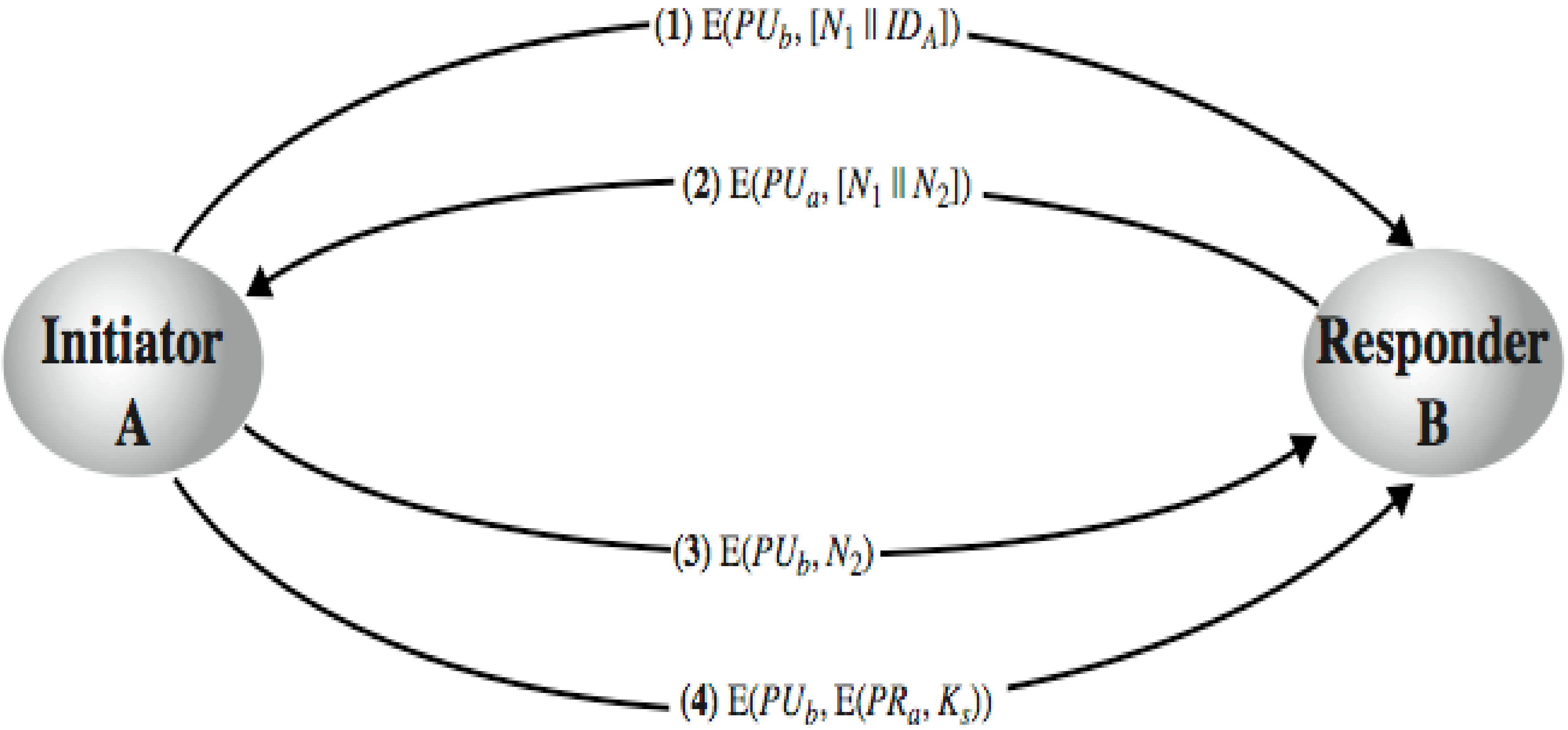
Man-in-the-Middle Attack

- this very simple scheme is vulnerable to an active man-in-the-middle attack

1. A generates a public/private key pair $\{PU_a, PR_a\}$and transmits a message intended for B consisting of and an identifier of A, ID_A .
2. E intercepts the message, creates its own public/private key pair $\{PU_e, PR_e\}$ and transmits $PU_e \,||\, ID_A$ to B.
3. B generates a secret key, K_s, and transmits E(PU_e, K_s).
4. E intercepts the message and learns by computing D(PR_e, E(PU_e, K_s)).
5. E transmits E(PU_a, K_s) to A.


Darth
Darth modifies message from Bob to Alice
Internet or other comms facility
Bob
Alice

Secret Key Distribution with Confidentiality and Authentication



(1) E(PU_b, [N_1 ‖ ID_A])
(2) E(PU_a, [N_1 ‖ N_2])
Initiator A
Responder B
(3) E(PU_b, N_2)
(4) E(PU_b, E(PR_a, K_s))

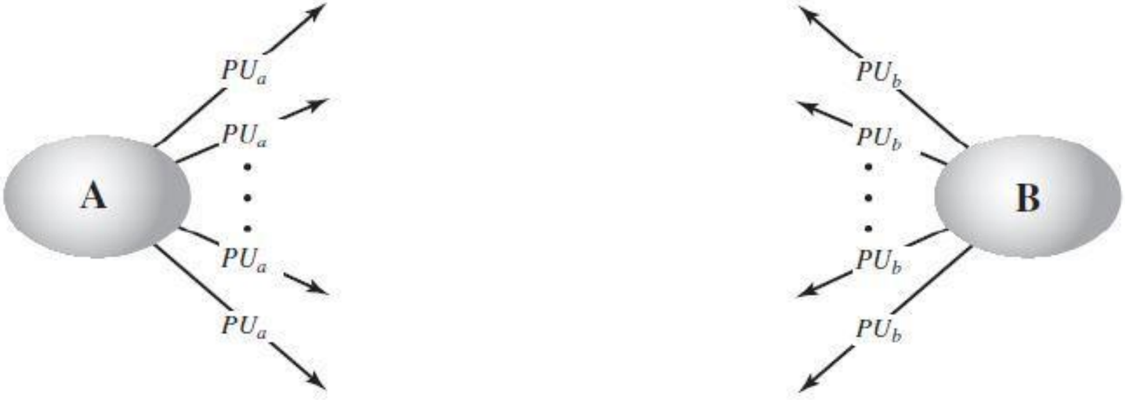
Distribution of Public Key

- Two aspects to the use of public-key encryption:
  - The distribution of public keys.
  - The use of public-key encryption to distribute secret keys.

- ❖ Public announcement
- ❖ Publicly available directory
- ❖ Public-key authority
- ❖ Public-key certificates

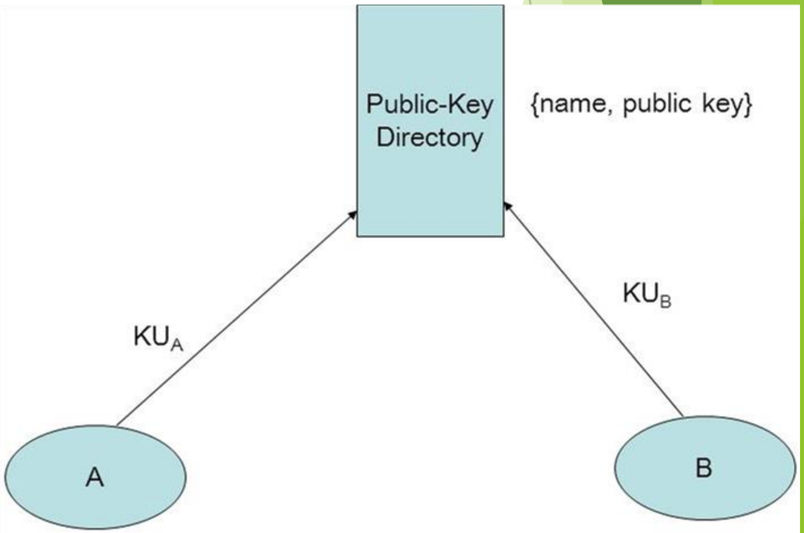
Public announcement

- users distribute public keys to recipients or broadcast to community at large
  - eg. append PGP keys to email messages or post to news groups or email list
- major weakness is forgery
  - anyone can create a key claiming to be someone else and broadcast it
  - until forgery is discovered can masquerade as claimed user



Publicly available directory

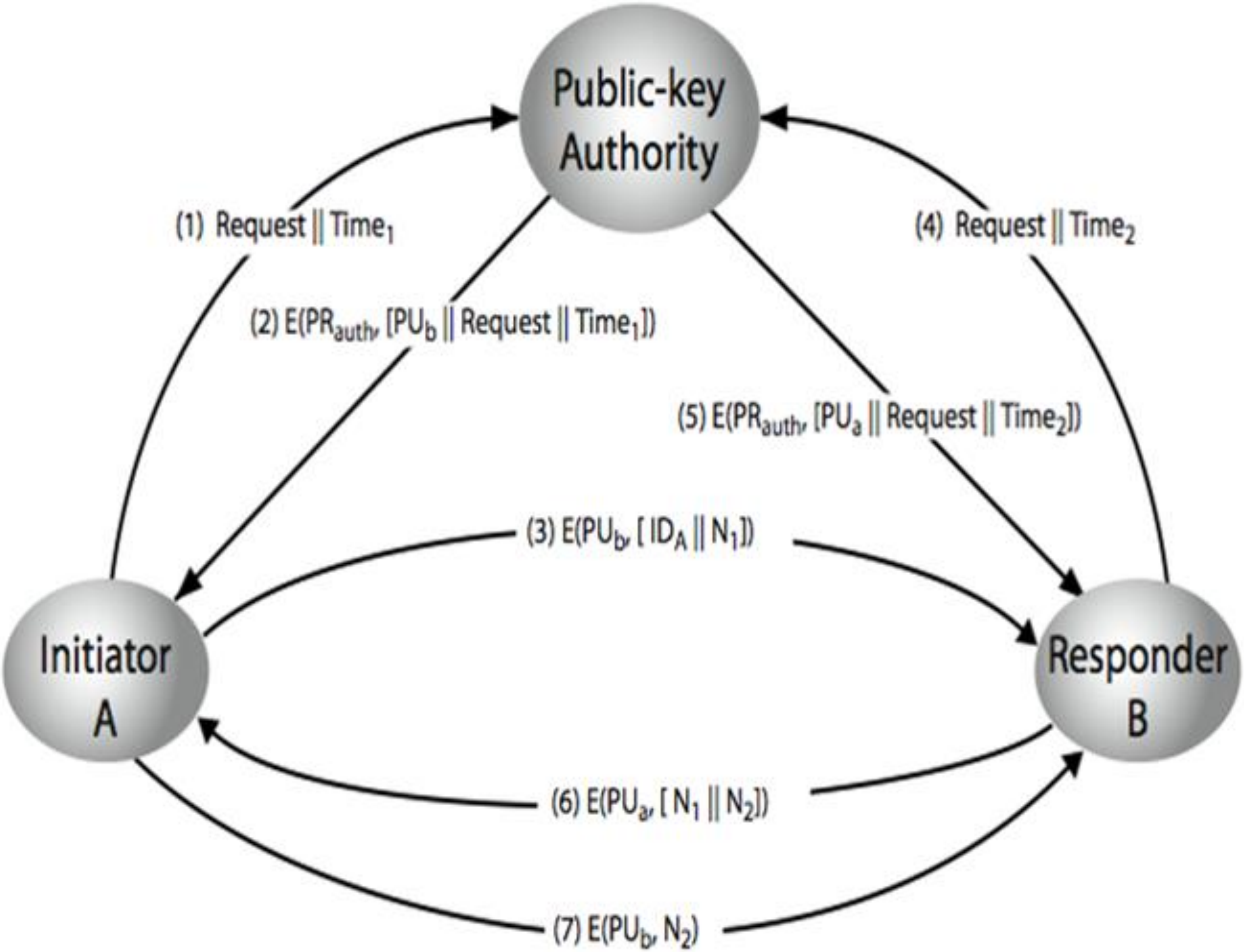
- ❖ can obtain greater security by registering keys with a public directory
- ❖ directory must be trusted with properties:
  - contains {name,public-key} entries
  - participants register securely with directory
  - participants can replace key at any time
  - directory is periodically published
  - directory can be accessed electronically
- ❖ still vulnerable to tampering or forgery



Public-key authority

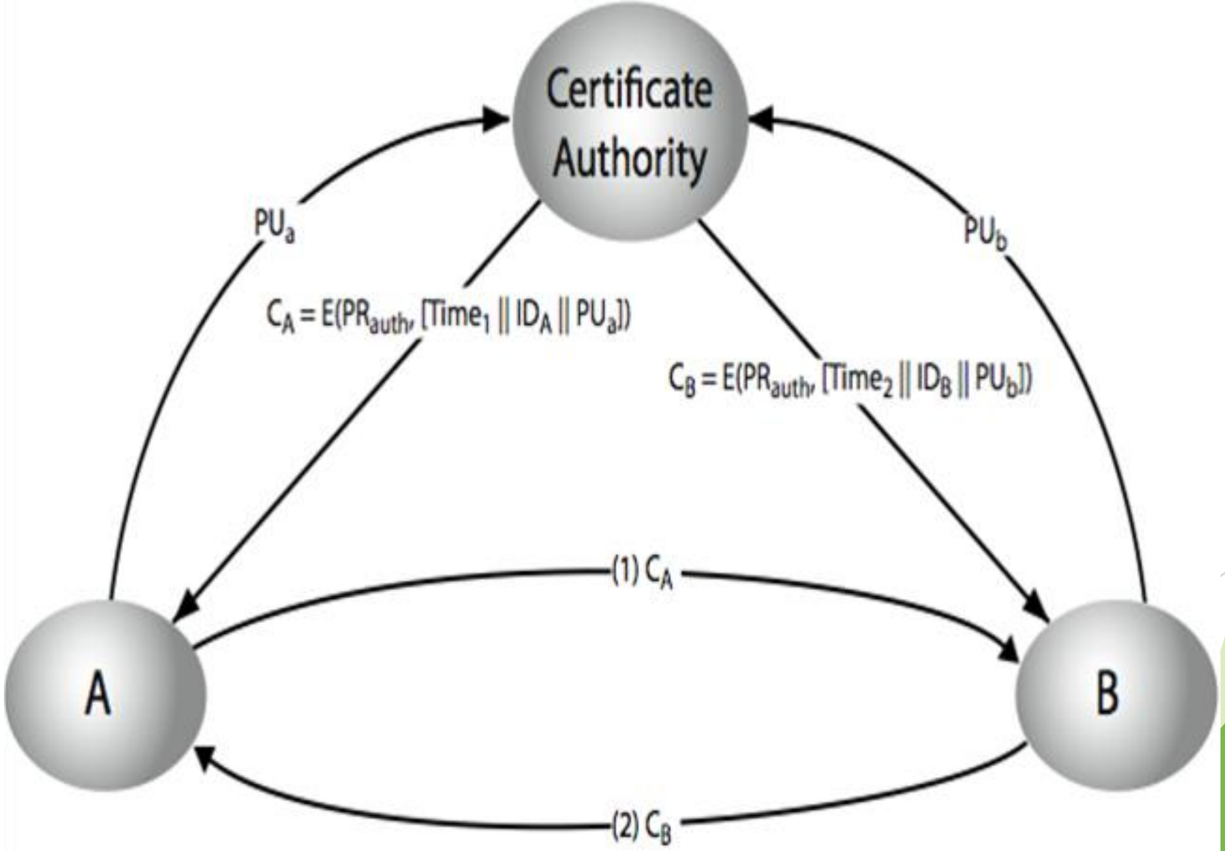
- improve security by tightening control over distribution of keys from directory
- has properties of directory
- and requires users to know public key for the directory
- then users interact with directory to obtain any desired public key securely
  - does require real-time access to directory when keys are needed
  - may be vulnerable to tampering

● User must appeal to the authority for a public key for every other user that it wishes to contact which makes the **system slow.**


Public-key Authority
(1) Request || $Time_1$
(4) Request || $Time_2$
(2) $E(PR_{auth}, [PU_b || Request || Time_1])$
(5) $E(PR_{auth}, [PU_a || Request || Time_2])$
(3) $E(PU_b, [ID_A || N_1])$
Initiator A
Responder B
(6) $E(PU_a, [N_1 || N_2])$
(7) $E(PU_b, N_2)$

Public-key certificates

- certificates allow key exchange without real-time access to public-key authority
- a certificate binds identity to public key
  - usually with other info such as period of validity, rights of use etc
- with all contents signed by a trusted Public-Key or Certificate Authority (CA)
- can be verified by anyone who knows the public-key authorities public-key



Certificate Authority
PU_a
PU_b
$C_A = E(PR_{auth}, [Time_1 || ID_A || PU_a])$
$C_B = E(PR_{auth}, [Time_2 || ID_B || PU_b])$
(1) C_A
(2) C_B
A
B

Diffie-Hellman Key Exchange

1. Choose two prime numbers **n** and **g** where g is a primitive root of n.

2. User A selects X_A as his **private key randomly**. i.e. $X_A < n$

3. User B selects X_B as his **private key randomly**. i.e. $X_B < n$

4. User A computes his **public key** i.e. $Y_A = (g^{XA}) \bmod n$

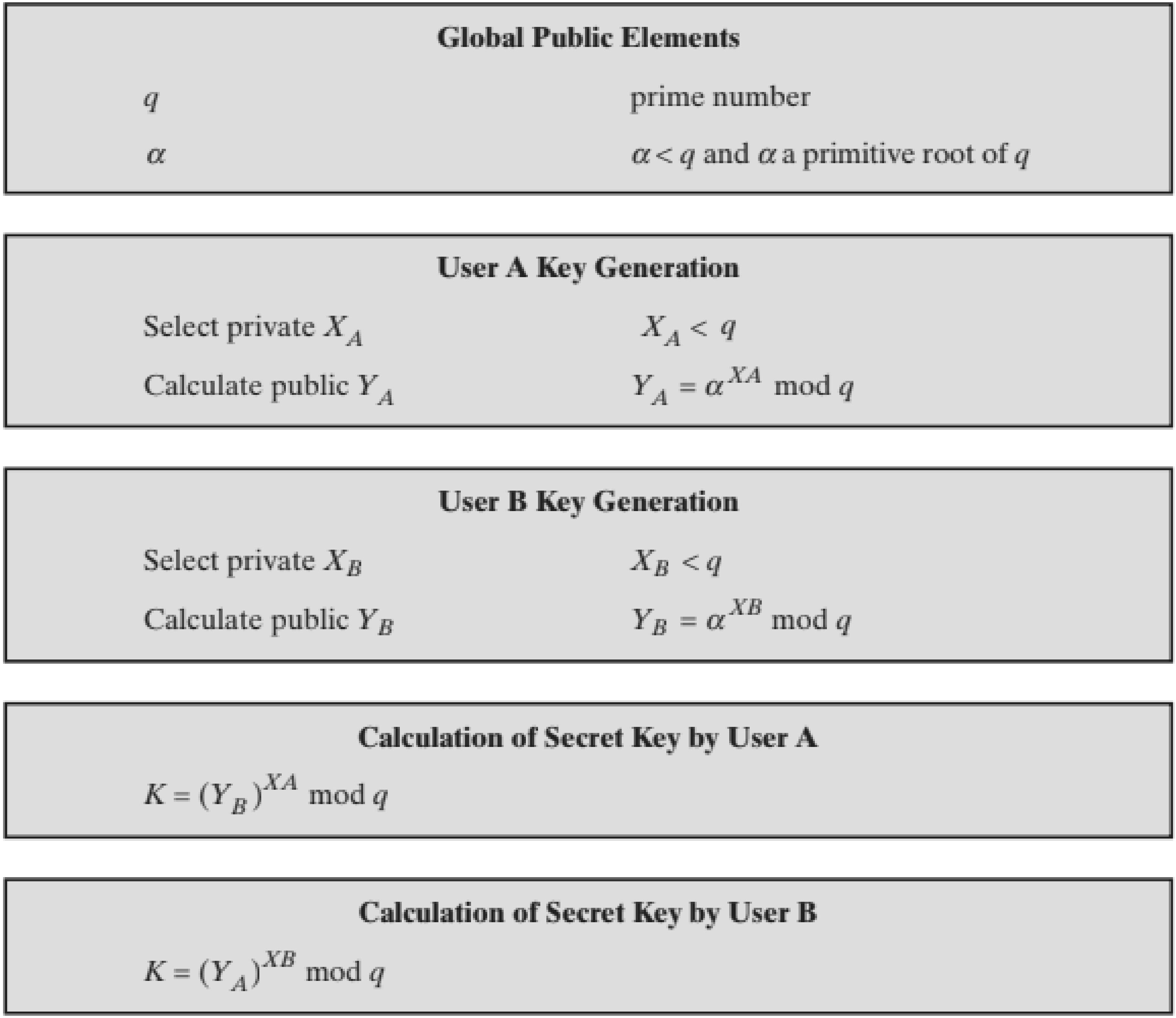
5. User B computes his **public key** i.e. $Y_B = (g^{XB}) \bmod n$

6. Exchange their public keys

7. User A computes key called **shared secret key**. i.e. $k = (Y_B{}^{XA}) \bmod n$

8. User B computes key called **shared secret key**. i.e. $k = (Y_A{}^{XB}) \bmod n$

9. Both user communicate each other using one of the symmetric encryption technique. They use shared secret key as the encryption key for selected algorithm.



Global Public Elements

q	prime number
α	$\alpha < q$ and α a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{XA} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{XB} \bmod q$

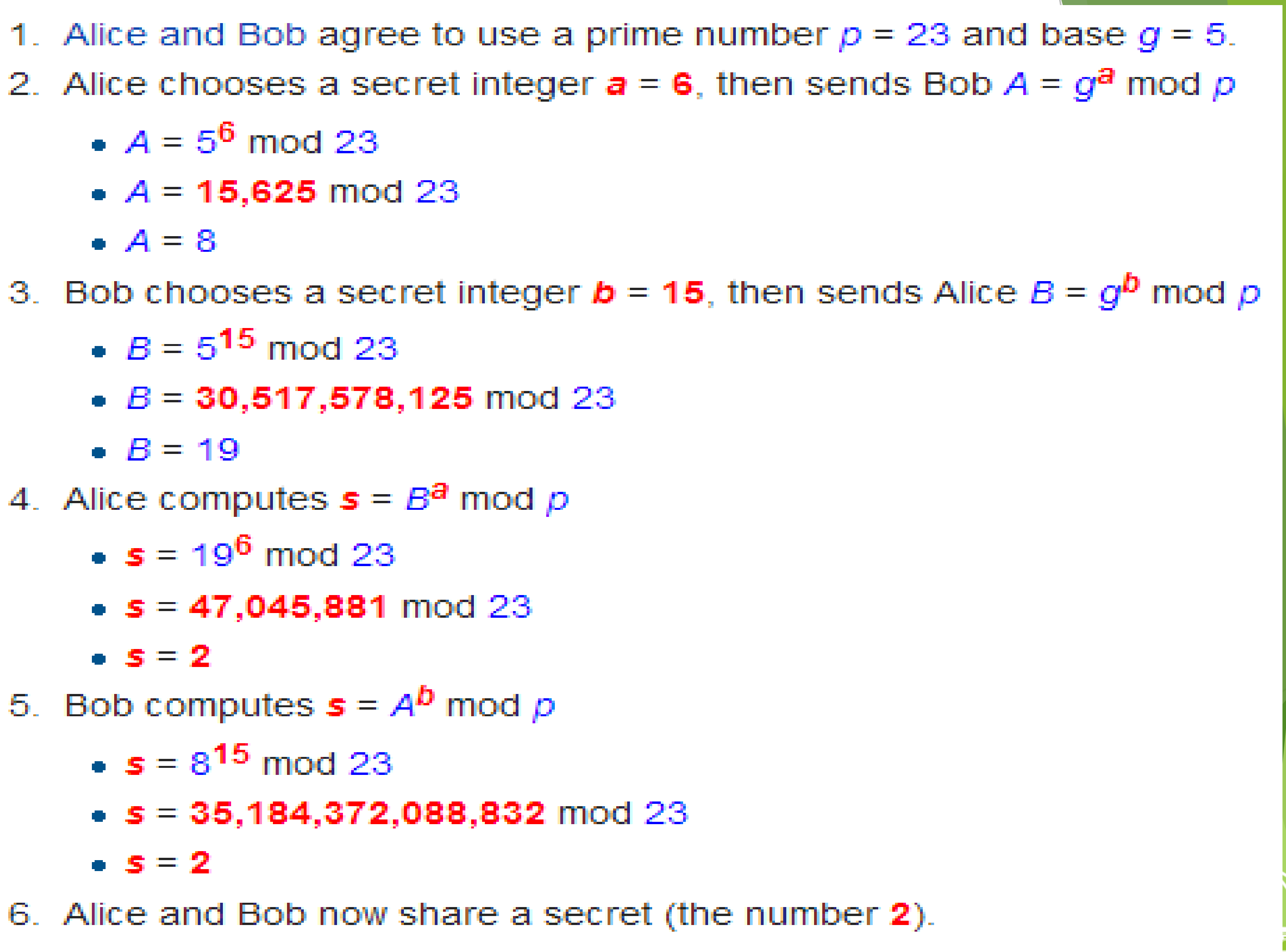
Calculation of Secret Key by User A

$K = (Y_B)^{XA} \bmod q$

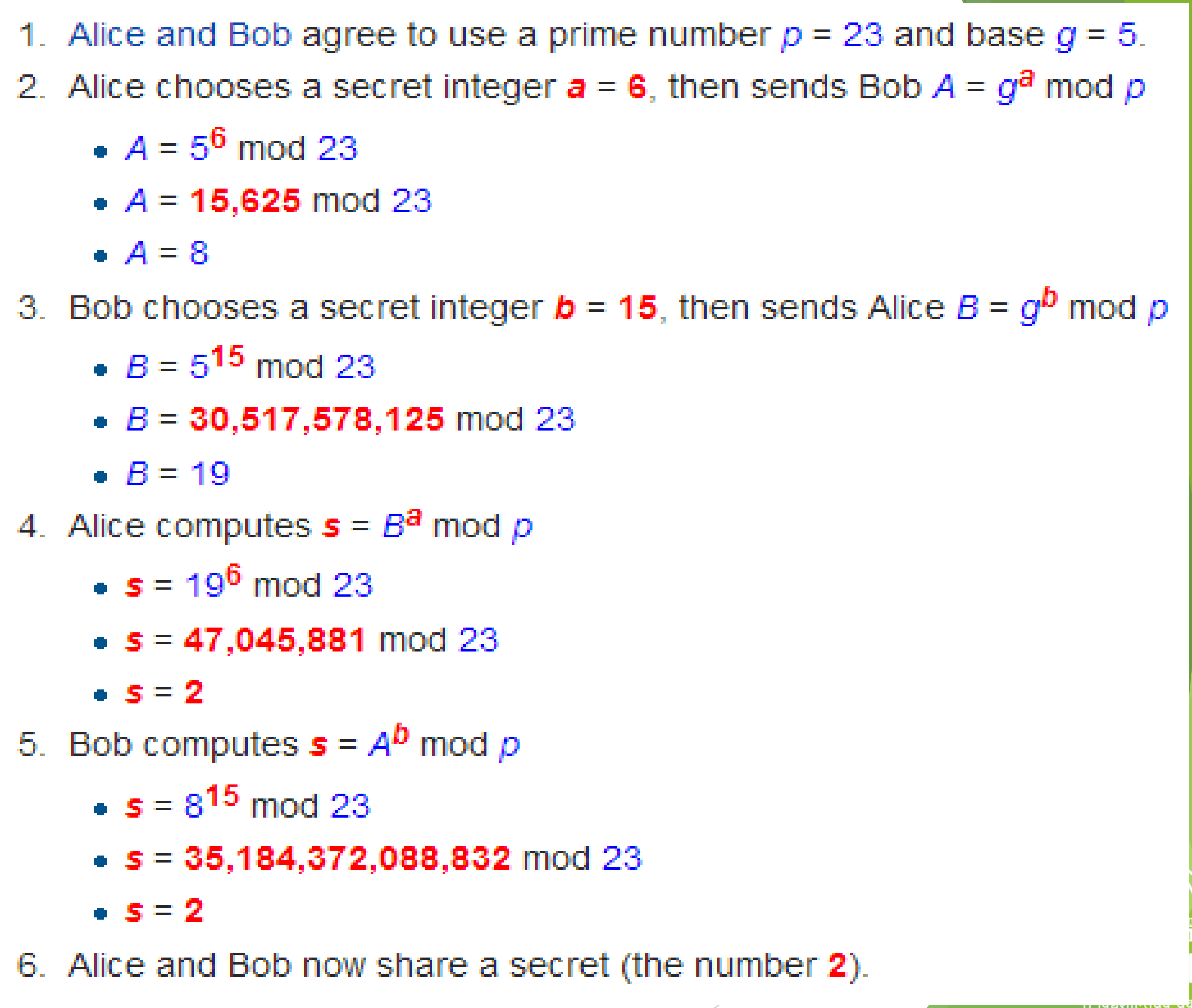
Calculation of Secret Key by User B

$K = (Y_A)^{XB} \bmod q$

Figure 10.1 The Diffie-Hellman Key Exchange Algorithm



1. Alice and Bob agree to use a prime number p = 23 and base g = 5.
2. Alice chooses a secret integer $\mathbf{a}$ = **6**, then sends Bob $A = g^{a} \bmod p$
   - $A = 5^{6} \bmod 23$
   - A = **15,625** mod 23
   - A = 8
3. Bob chooses a secret integer $\mathbf{b}$ = **15**, then sends Alice $B = g^{b} \bmod p$
   - $B = 5^{15} \bmod 23$
   - B = **30,517,578,125** mod 23
   - B = 19
4. Alice computes $\mathbf{s} = B^{a} \bmod p$
   - $\mathbf{s} = 19^{6} \bmod 23$
   - $\mathbf{s}$ = **47,045,881** mod 23
   - $\mathbf{s}$ = **2**
5. Bob computes $\mathbf{s} = A^{b} \bmod p$
   - $\mathbf{s} = 8^{15} \bmod 23$
   - $\mathbf{s}$ = **35,184,372,088,832** mod 23
   - $\mathbf{s}$ = **2**
6. Alice and Bob now share a secret (the number **2**).



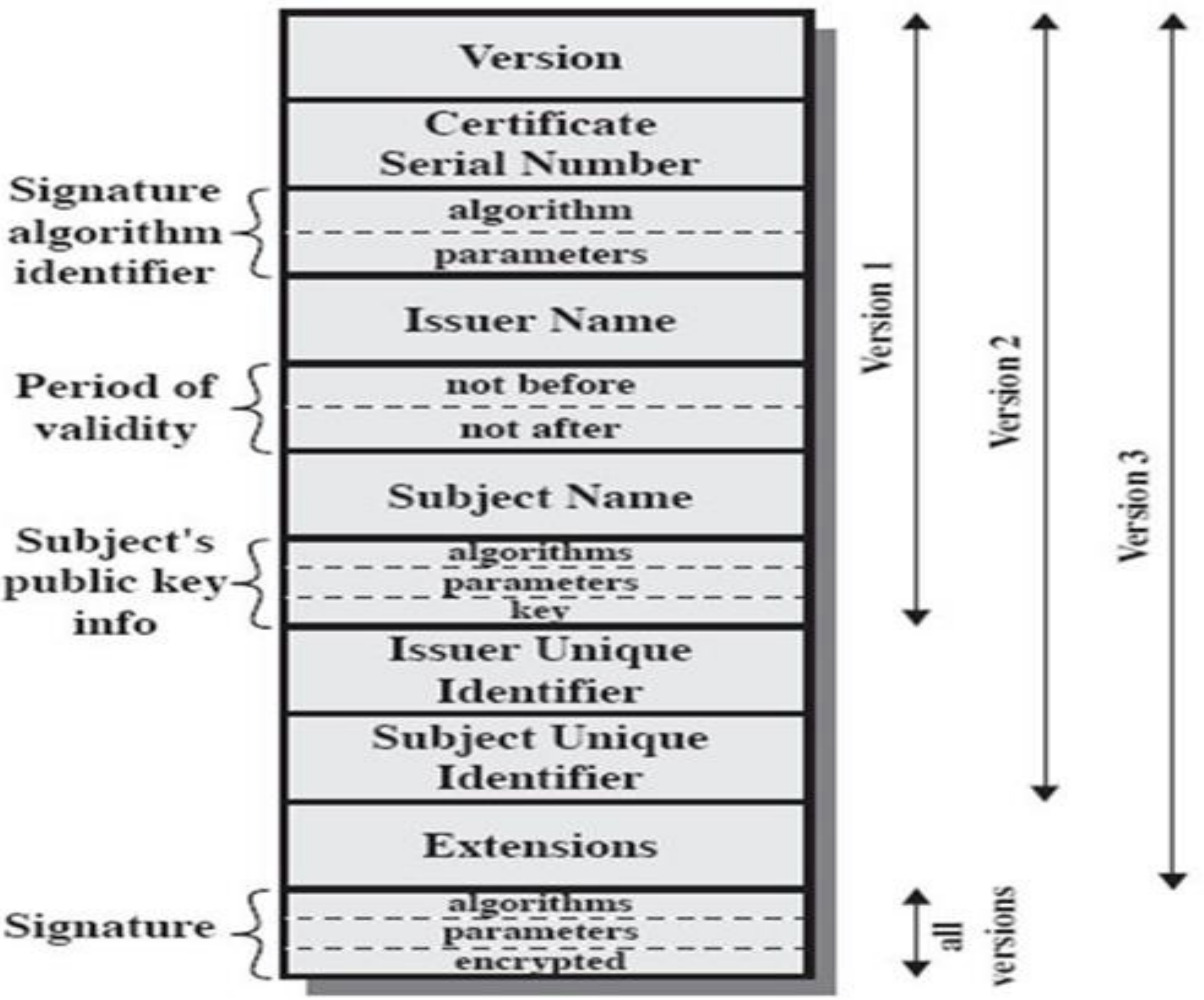
1. Alice and Bob agree to use a prime number $p = 23$ and base $g = 5$.
2. Alice chooses a secret integer $a = 6$, then sends Bob $A = g^a \bmod p$
   - $A = 5^6 \bmod 23$
   - $A = 15{,}625 \bmod 23$
   - $A = 8$
3. Bob chooses a secret integer $b = 15$, then sends Alice $B = g^b \bmod p$
   - $B = 5^{15} \bmod 23$
   - $B = 30{,}517{,}578{,}125 \bmod 23$
   - $B = 19$
4. Alice computes $s = B^a \bmod p$
   - $s = 19^6 \bmod 23$
   - $s = 47{,}045{,}881 \bmod 23$
   - $s = 2$
5. Bob computes $s = A^b \bmod p$
   - $s = 8^{15} \bmod 23$
   - $s = 35{,}184{,}372{,}088{,}832 \bmod 23$
   - $s = 2$
6. Alice and Bob now share a secret (the number **2**).

Digital Certificate

- Certificates are the framework for identification information, and bind identities with public keys.
- They provide a foundation for
  - identification,
  - authentication and
  - non-repudiation.
- Trusted organization (i.e. **Certificate Authority** (CA)) that issues certificates and maintains status information about certificates.
- The most popular CA's are Verisign and Entrust.
- CA issues new certificates, maintain old ones, and revoke the certificate that has become invalid for some sort of reasons, etc.
- The CA can delegate some of its tasks to this third-party called as a **Registration Authority (RA).**
- A Standard called **X.509** define a structure of a digital certificate.

BASIS FOR COMPARISON	DIGITAL SIGNATURE	DIGITAL CERTIFICATE
Basic	It verifies the authenticity and source of a particular document.	It creates an identity of a website and also increases its trustworthiness.
Process	The document is encrypted at the sending end and decrypted at the receiving end using asymmetric keys.	A certificate is issued by a trusted agency known as CA which follow particular steps to do so that are - key generation, registration, verification and creation.
Security	It provides authentication, non-repudiation and integrity.	It provides identification, authentication, non-repudiation and security.

Structure of X.509 digital certificate.



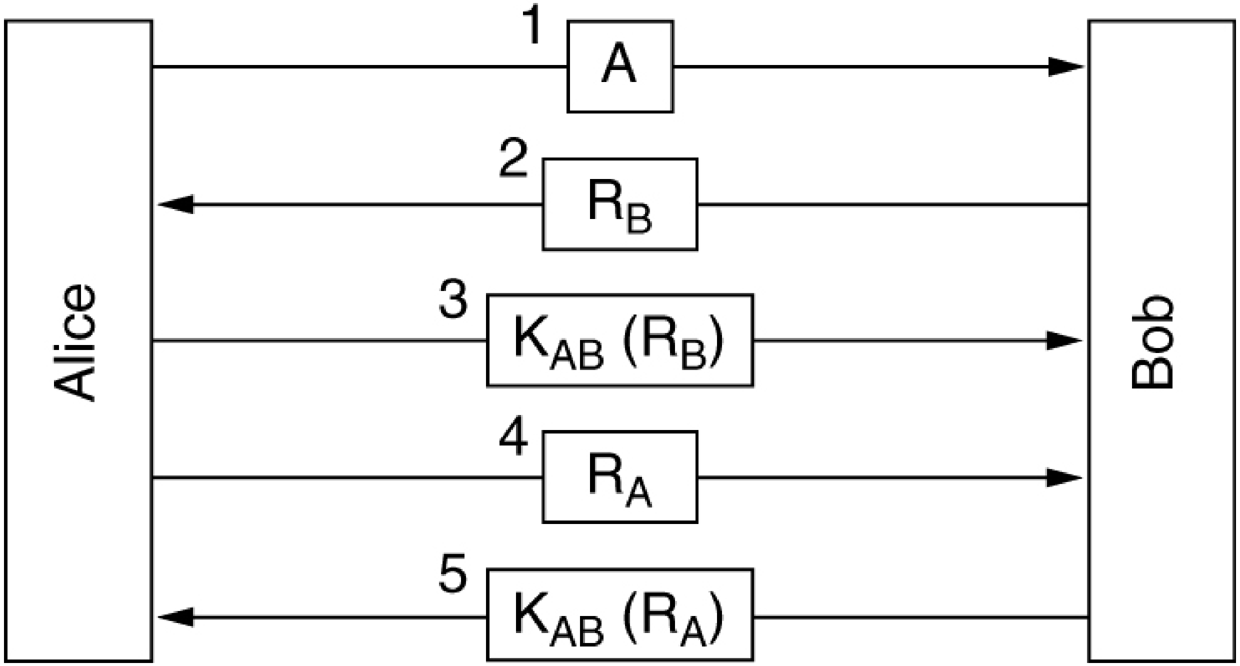
Authentication Protocols

Authentication

Who are you? User, Process, Device, Hardware

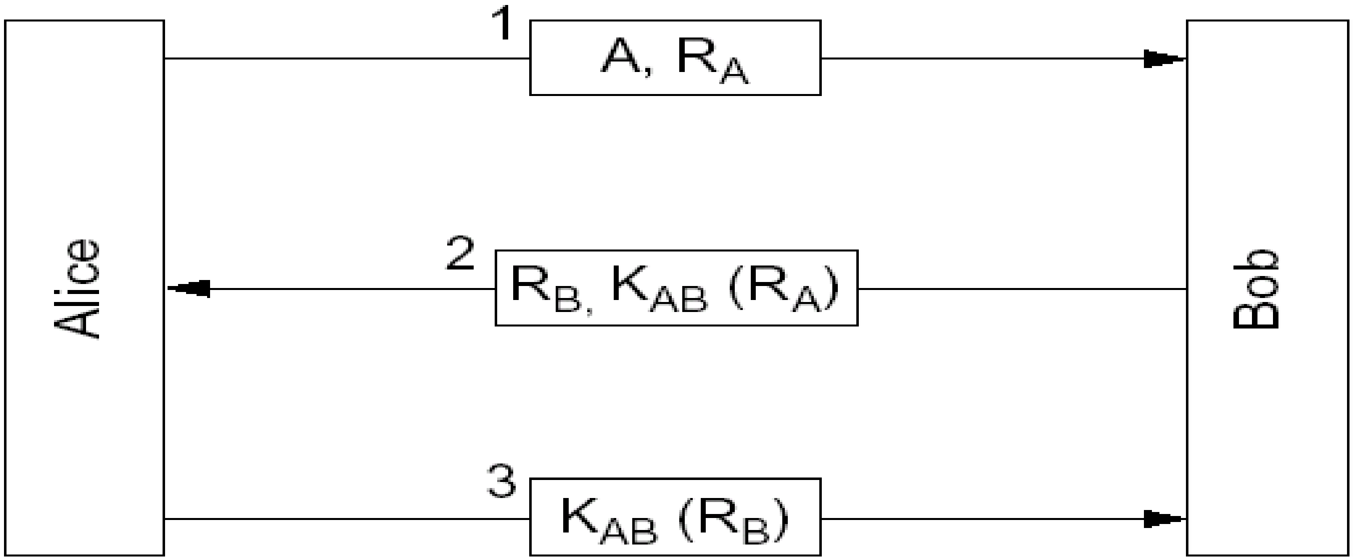
1. A --I am alice □ B but trudy can also say I am alice
2. A --My IP is a.b.c.d -> B but trudy can spoof
3. Secret password , Encrypted pass, Reply!
4. Nonce
5. Public key

Authentication Based on a Shared Secret Key



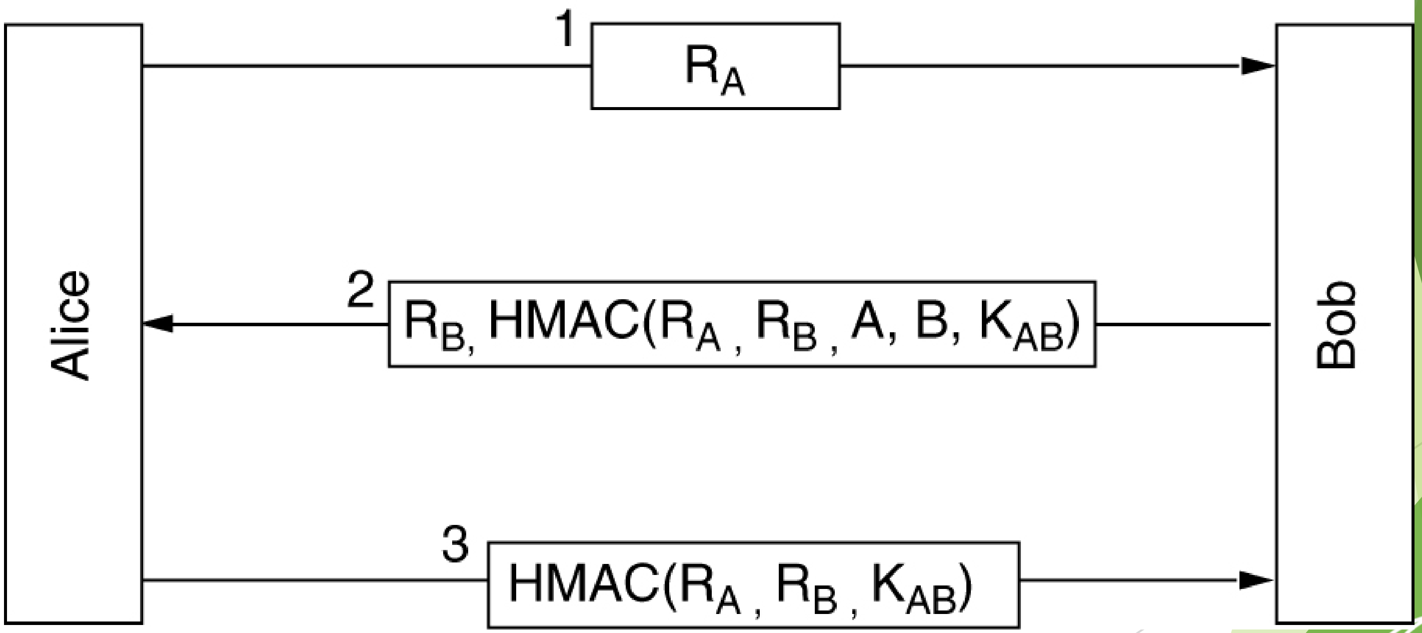
- ▣ Two-way authentication using a challenge-response protocol.

Authentication Protocol



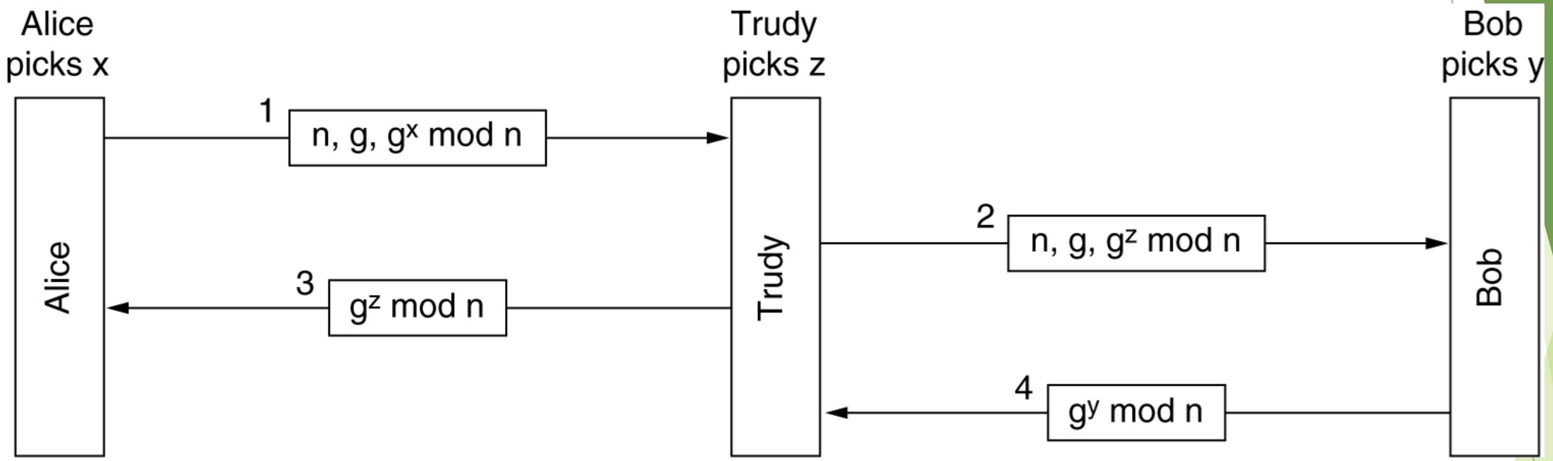
Authentication Based on a Shared Secret Key

- Authentication using HMACs.


Alice
Bob
1
R_A
2
R_B, HMAC(R_A , R_B , A, B, K_{AB})
3
HMAC(R_A , R_B , K_{AB})

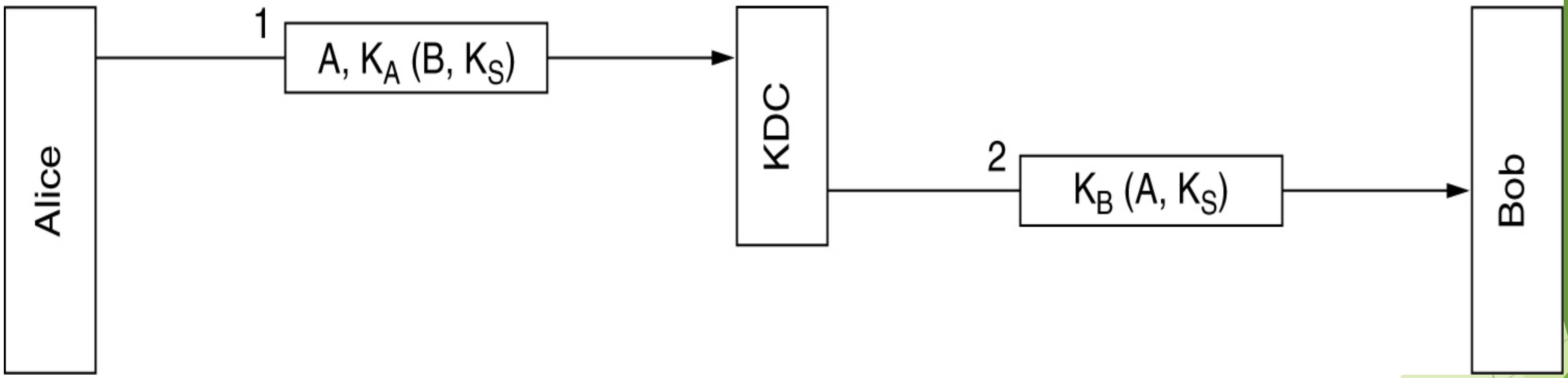
Establishing a Shared Key: The Diffie-Hellman Key Exchange

- The bucket brigade or man-in-the-middle attack.



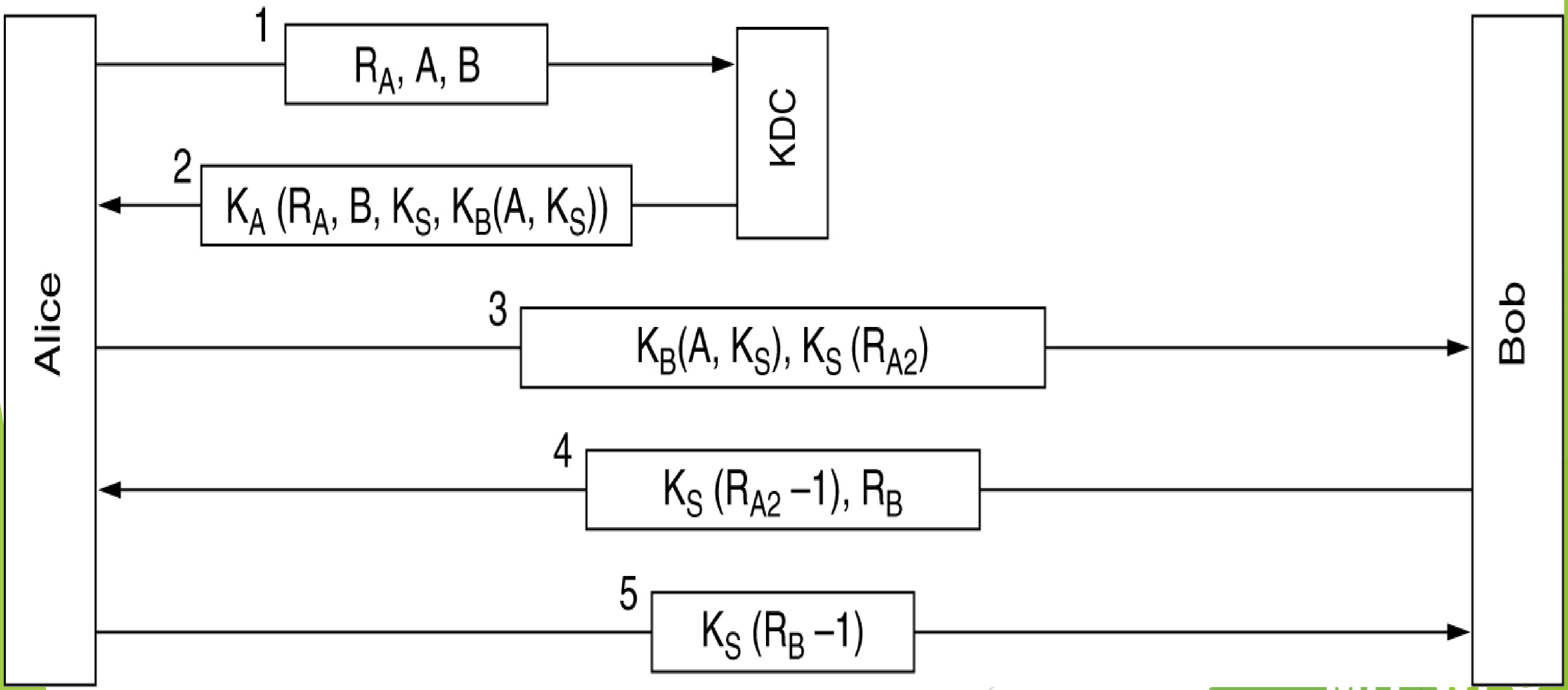
Authentication Using a Key Distribution Center

- A first attempt at an authentication protocol using a KDC.



Authentication Using a Key Distribution Center

- The Needham-Schroeder authentication protocol.



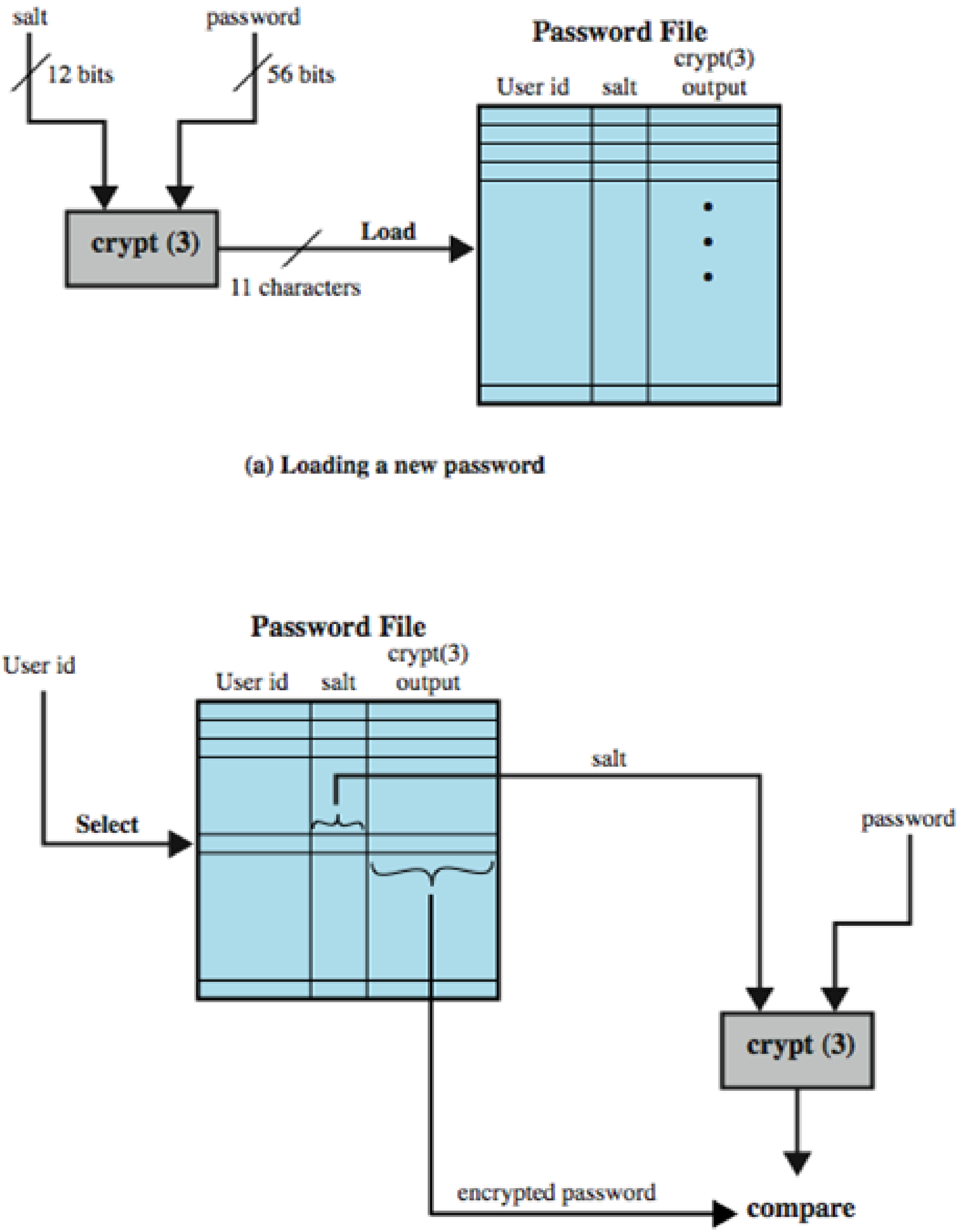
Password attacks/vulnerabilities

- offline dictionary attack
- specific account attack
- popular password attack (against a wide range of IDs)
- password guessing against single user (previous knowledge about the user)
- workstation hijacking
- exploiting user mistakes
- exploiting multiple password use
- vulnerable to eavesdropping

Countermeasures/ Defense for password vulnerability

- stop unauthorized access to password file
- intrusion detection measures
- account lockout mechanisms
- policies against using common passwords but rather hard to guess passwords
- training & enforcement of policies
- automatic workstation logout
- encrypted network links

Use of hashed passwords



(a) Loading a new password

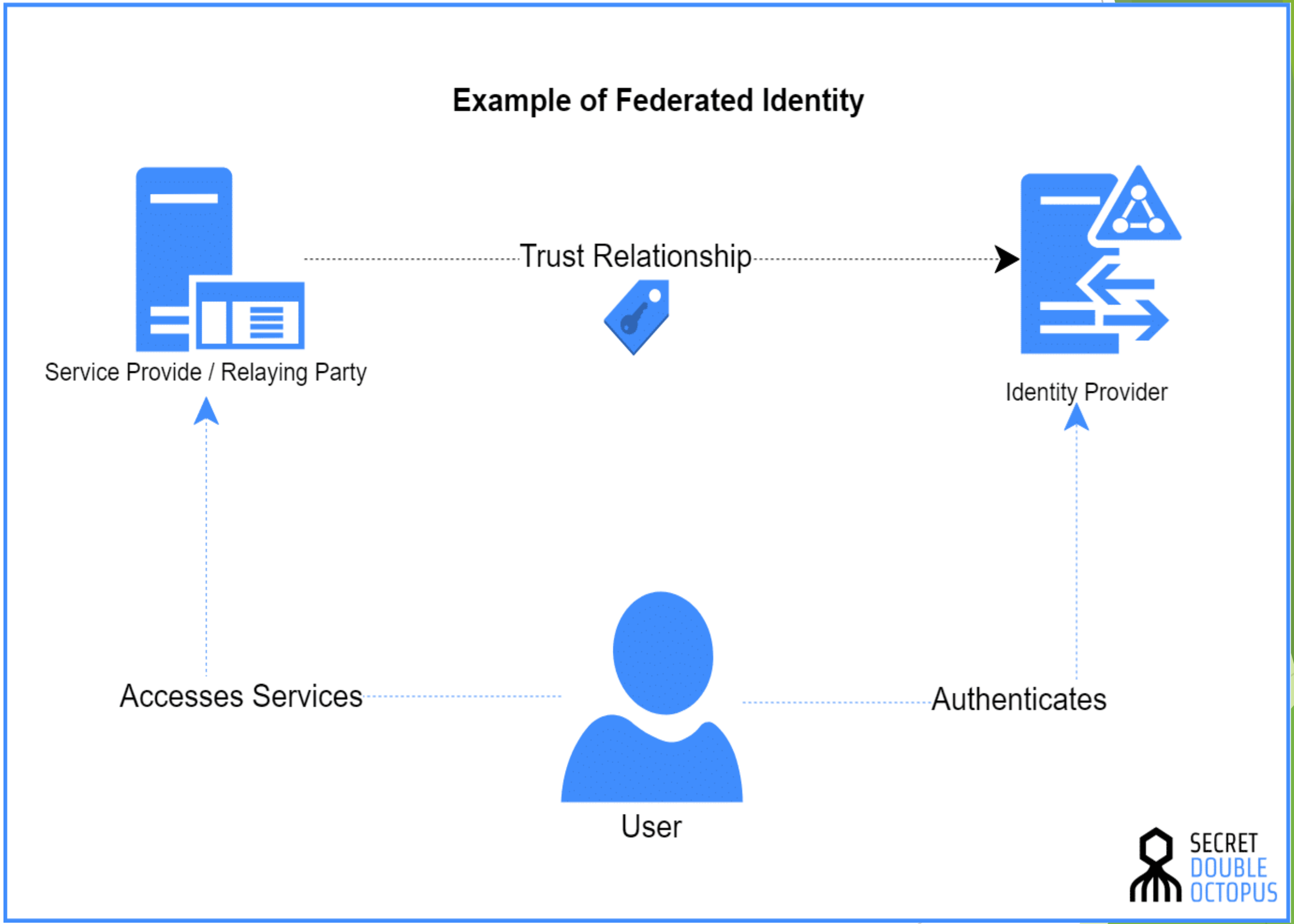
(b) Verifying a password

Why a salt value?

- Prevents duplicate passwords from being visible in the password file
- Increases the difficulty of offline dictionary attacks
- Nearly impossible to tell if a person used the same password on multiple systems

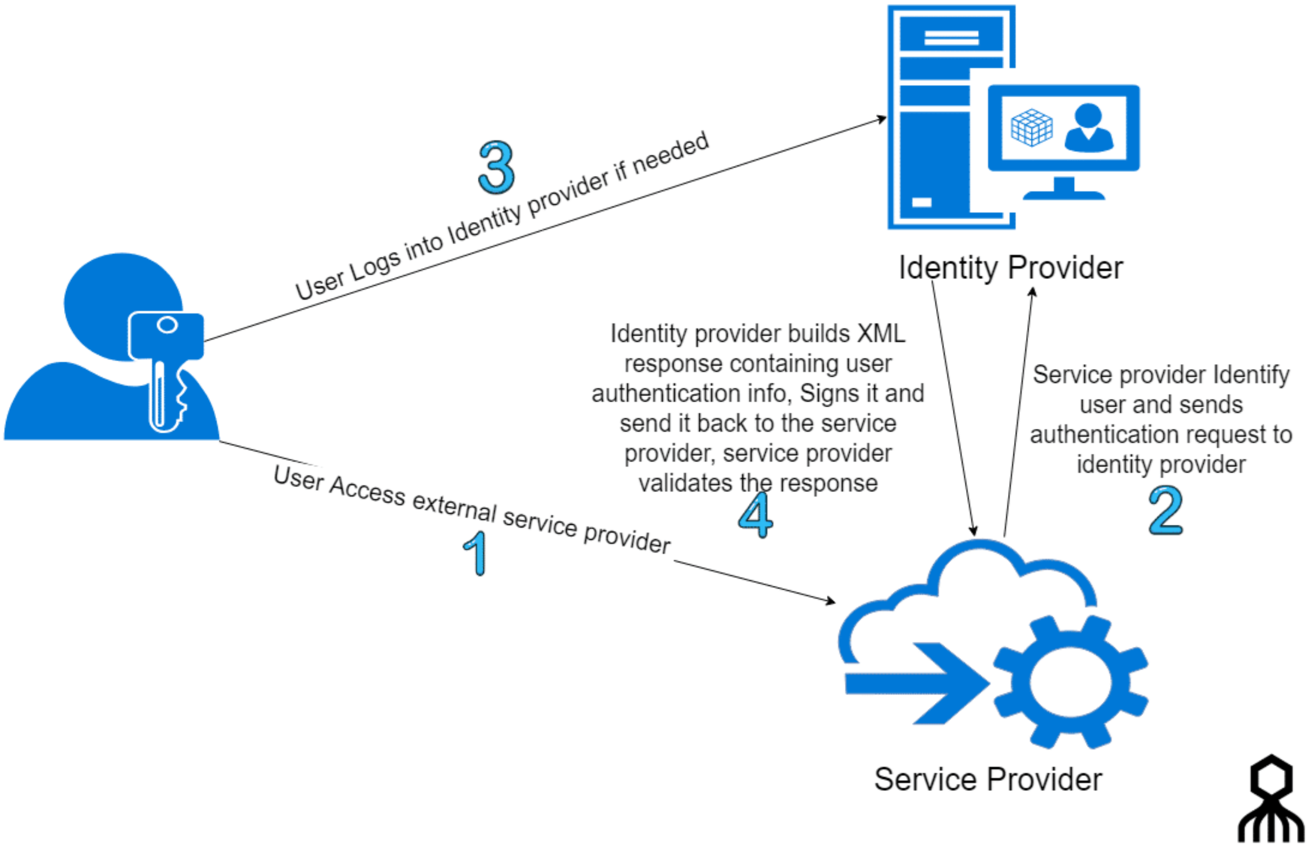
Federated Authentication

- **Federated identity management** (FIM) is an arrangement that can be made between multiple enterprises to let subscribers use the same identification data to obtain access to the networks of all the enterprises in the group. The use of such a system is sometimes called **identity federation**.
- **Federated authentication** allows members of one organization to use their **authentication** credentials to access a web application in another institution. The two are often combined to "stack" the benefits of both technologies.



Example of Federated Identity

Single Sign On Process


3
User Logs into Identity provider if needed
Identity Provider
Identity provider builds XML response containing user authentication info, Signs it and send it back to the service provider, service provider validates the response
4
Service provider Identify user and sends authentication request to identity provider
2
User Access external service provider
1
Service Provider

SECRET DOUBLE OCTOPUS