Computer System Security

ASSOC. PROF. NOHA A. HIKAL

INFORMATION TECHNOLOGY DEPT.

FACULTY OF COMPUTERS AND INFORMATION SCIENCES

SUBJECT CODE:IT424

4TH LEVEL –IT&IS

WEEK 9 - 8 APRIL 2020

Data Authentication

Outlines:

1. **Why Data Authentication?**
2. **Digital Fingerprints**
3. **Cryptographic Hash Functions**
4. **Digital Signatures**
5. **Network Authentication procedure**

1- Why Data Authentication?

1- Certify the origin of the data

2- Convince the user that the data has not been modified or fabricated

3- Provide data non-reputation

Data authentication:

To authenticate a long data string *M*, it suffices to compute a short representation *h* of *M* and encrypt *h*

2-Digital Fingerprints

A short representation of *M* generated without using secret key is referred to as a *digital digest* or a *digital fingerprint*

Digital fingerprint can be obtained using a *cryptographic hash function*, also called *one-way hash function*

A short representation of *M* generated using a secret key is referred to as a *message authentication code* (*MAC*) or a *tag*

MAC can be obtained using an *encrypted checksum algorithm*

Keyed-hash message authentication code (*HMAC*) is the combination of cryptographic hash function and encrypted checksum algorithm

3- Cryptographic Hash Functions

A hash function takes a long string as input, breaks it into pieces, mixes them up, and produces a new shorter string

Not every hash function is suitable for generating a digital fingerprint. For example, let

$$M = M_1\ M_2\ \ldots\ M_k$$

where M_i is a 16-bit binary string

Define a hash function $H_{\oplus}$ by

$$H_{\oplus}(M) = M_1 \oplus M_2 \oplus \ldots \oplus M_k$$

It is straightforward to find sentences with different meanings that have the same hash value under $H_{\oplus}$

- S_1: "He likes you but I hate you" and S_2: "He hates you but I like you"
- Encoding English letters using 8-bit ASCII codes and removing spaces between words, we get $H_{\oplus}(S_1) = H_{\oplus}(S_2)$

Design Criteria

1. **One-Wayness**: Computing a digital fingerprint for a given string is easy, but finding a string that has a given fingerprint is hard
2. **Computational Uniqueness**: It is computational difficult to find two different strings with the same fingerprint

   Collision Resistance – Given a string x, it is intractable to find a different string y such that:

   $H(x) = H(y)$ (Note that such strings y exist)

   Strong Collision Resistance – It is intractable to find two binary strings x and y such that :

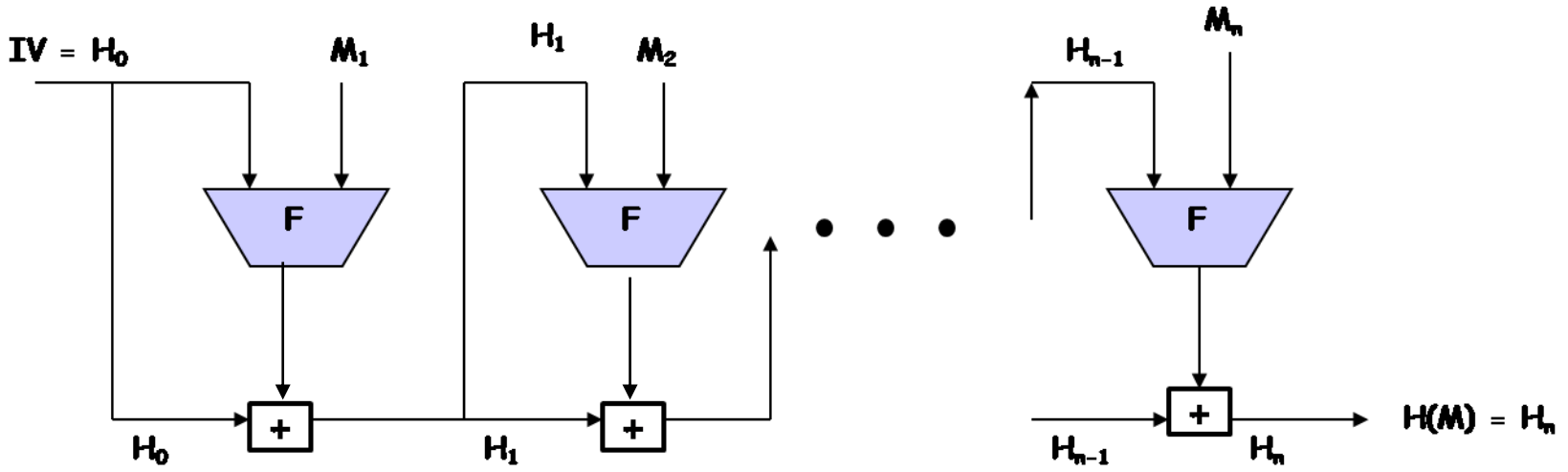
   $H(x) = H(y)$

   Note that failing the strong collision resistance does not imply failing the collision resistance

Basic Structure

The heart of this basic structure is a *compression function F*

- Different hash algorithms use different compression functions
- Use a CBC mode of repeated applications of *F* without using secret keys

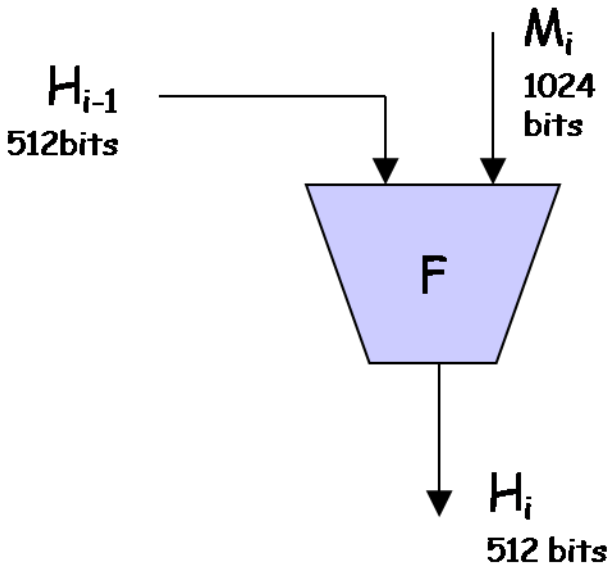


M is a plaintext block, IV is an initial vector, F is a compression function, and “+” is some form of modular addition operation

3- One-Way Hash Function Examples

- Several hash functions that were believed to be cryptographically strong, including
  MD4, MD5, HACAL-128 and RIPEMD,
  fail to satisfy the strong collision resistance
- Another commonly-used hash function SHA-1's collision resistance was proven weaker than expected
- Checksums are commonly used to detect transmission errors in network communication (also called Message Authentication Codes (MAC))
- Two standard hash functions: SHA-512 and WHIRLPOOL

3-1 SHA-512 Initial Process (I)



- SHA-512 uses a 512-bit IV
- Let r_1, r_2, r_3, r_4, r_5, r_6, r_7, and r_8 be eight 64-bit registers
  - Initially they are set to, respectively, the 64-bit binary string in the prefix of the fractional component of the square root of the first 8 prime numbers:

    $\sqrt{2}$, $\sqrt{3}$, $\sqrt{5}$, $\sqrt{7}$, $\sqrt{11}$, $\sqrt{13}$, $\sqrt{17}$, $\sqrt{19}$,

3-2 Exclusive-OR Cryptographic Checksums

Let E denote the AES-128 encryption algorithm and K an AES-128 secret key

$$H_{\oplus}(M) = M_1 \oplus M_2 \oplus \cdots \oplus M_k$$

$$MAC(M) = E_K(H_{\oplus}(M))$$

This method is insecure. It is vulnerable to a man-in-the-middle attack.

For example, suppose Alice and Bob share the same AES-128 key K.

If Alice sends (M, $E_K(H_{\oplus}(M))$) to Bob to authenticate M and Malice intercepts it,

then Malice can use $E_K(H_{\oplus}(M))$ to impersonate Alice .

4- Digital Signatures

have looked at message authentication

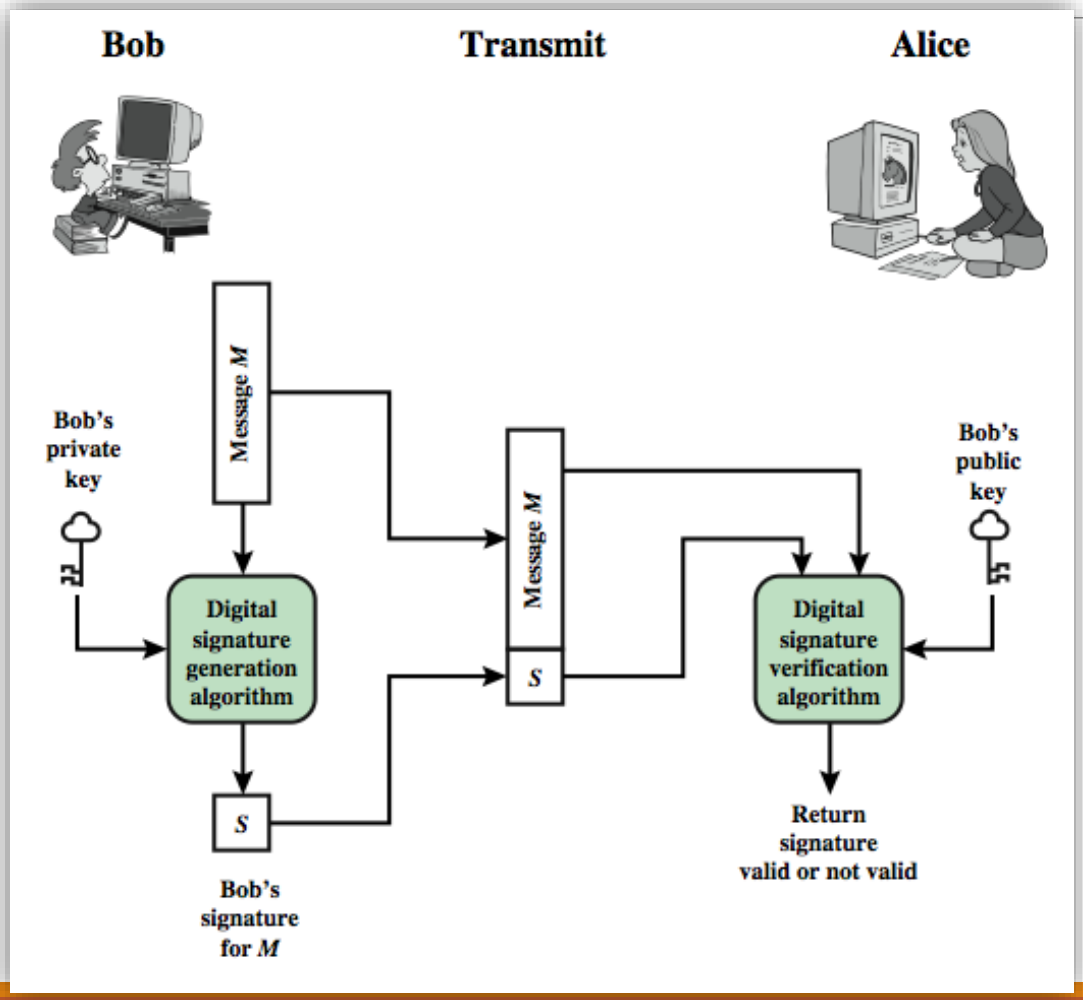
- but does not address issues of lack of trust

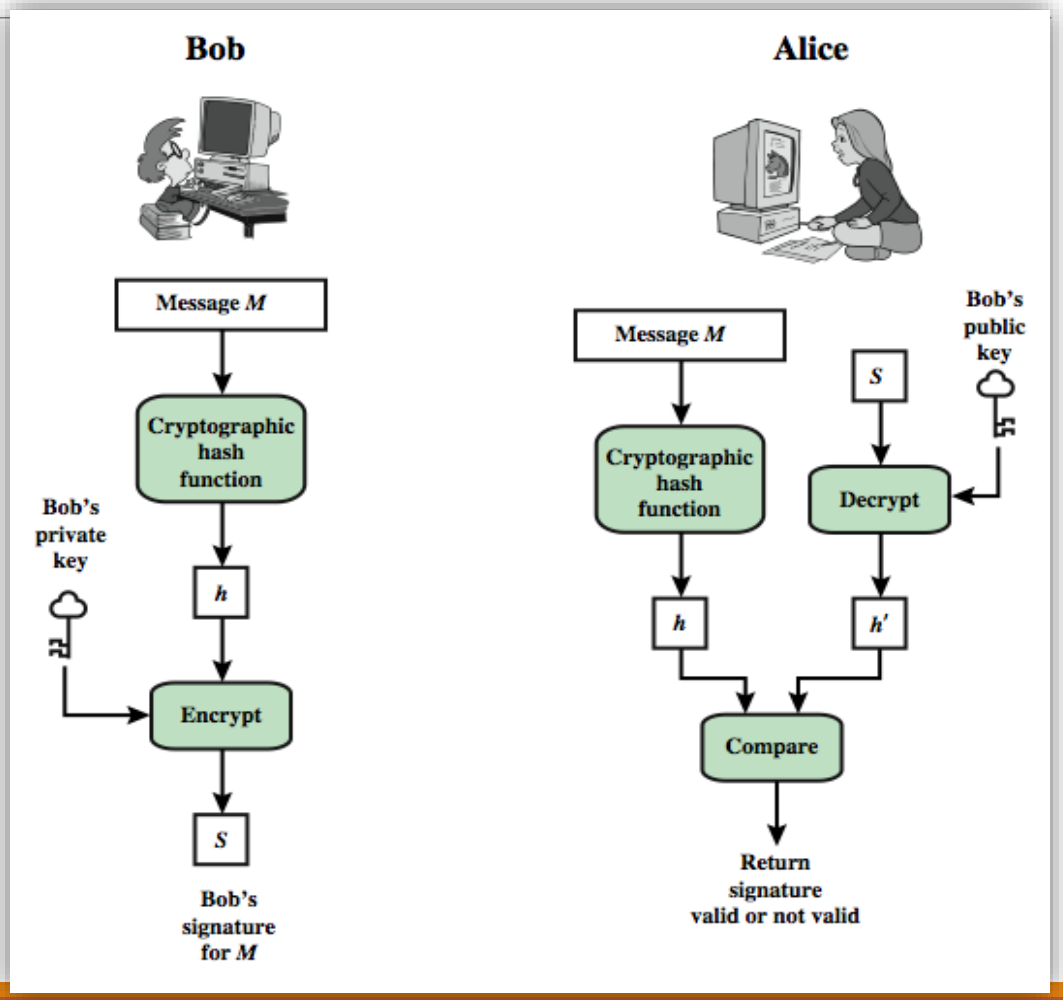
digital signatures provide the ability to:

- verify author, date & time of signature
- authenticate message contents
- be verified by third parties to resolve disputes

hence include authentication function with additional capabilities

Digital Signature Model





Examples:

- Direct Digital Signature
- ElGamal Digital Signatures
- Digital Signature Standard (DSS)
- Digital Signature Algorithm (DSA)

Table 9.3 Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

5- Network Authentication procedure

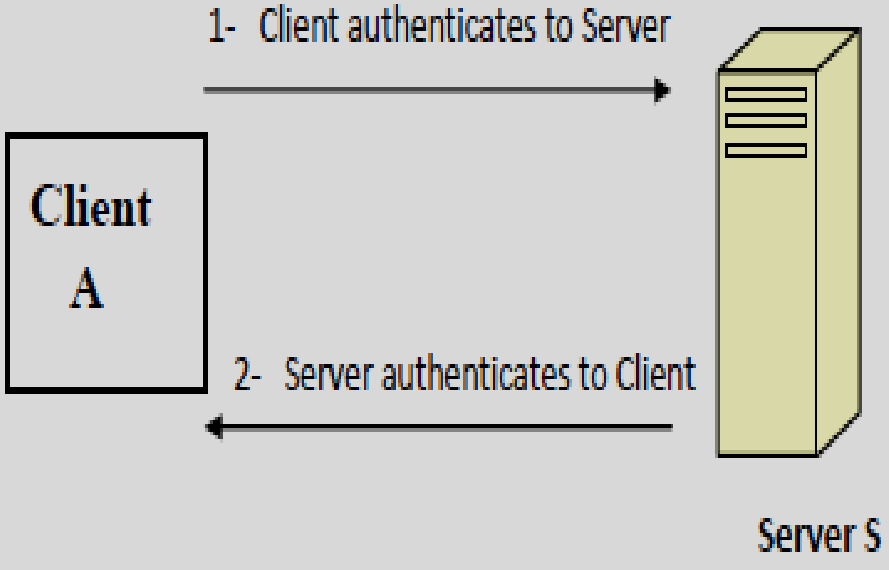


Figure 2: Two –Party Authentication

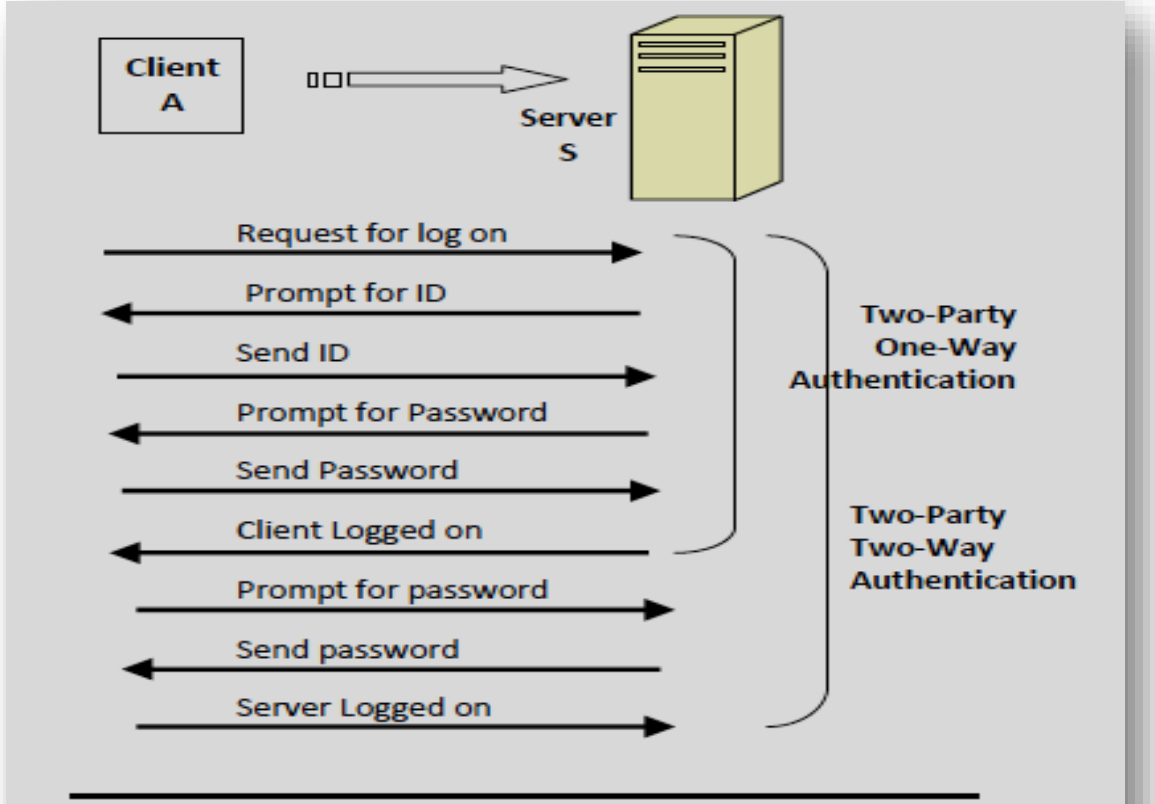


Figure 3: Two-Party One-Way Authentication and Two-Way Authentication.

Kerberos Authentication Protocol

Goals:

- Authenticate users on a local-area network without PKI
- Allow users to access to services without re-entering password for each service

It uses **symmetric-key encryption** and electronic passes called tickets

It uses **two different types of tickets**:

- TGS-ticket: issued to the user by AS
- V-ticket (server ticket): issued to the user by TGS

Requires **two special servers** to issue tickets to users:

- AS: Authentication Server. AS manages users and user authentication
- TGS: Ticket Granting Server. TGS manages servers

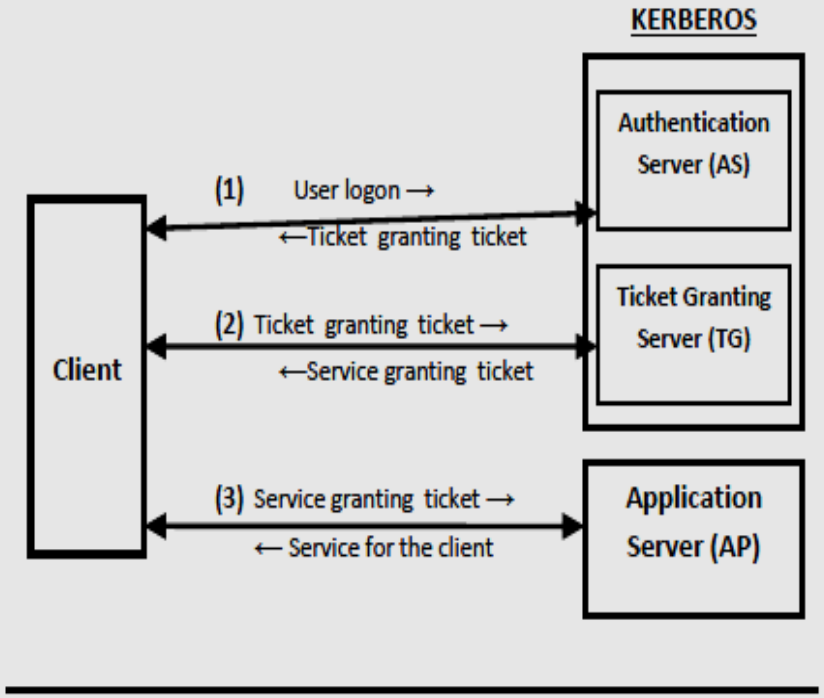


Figure 4: KERBEROS system

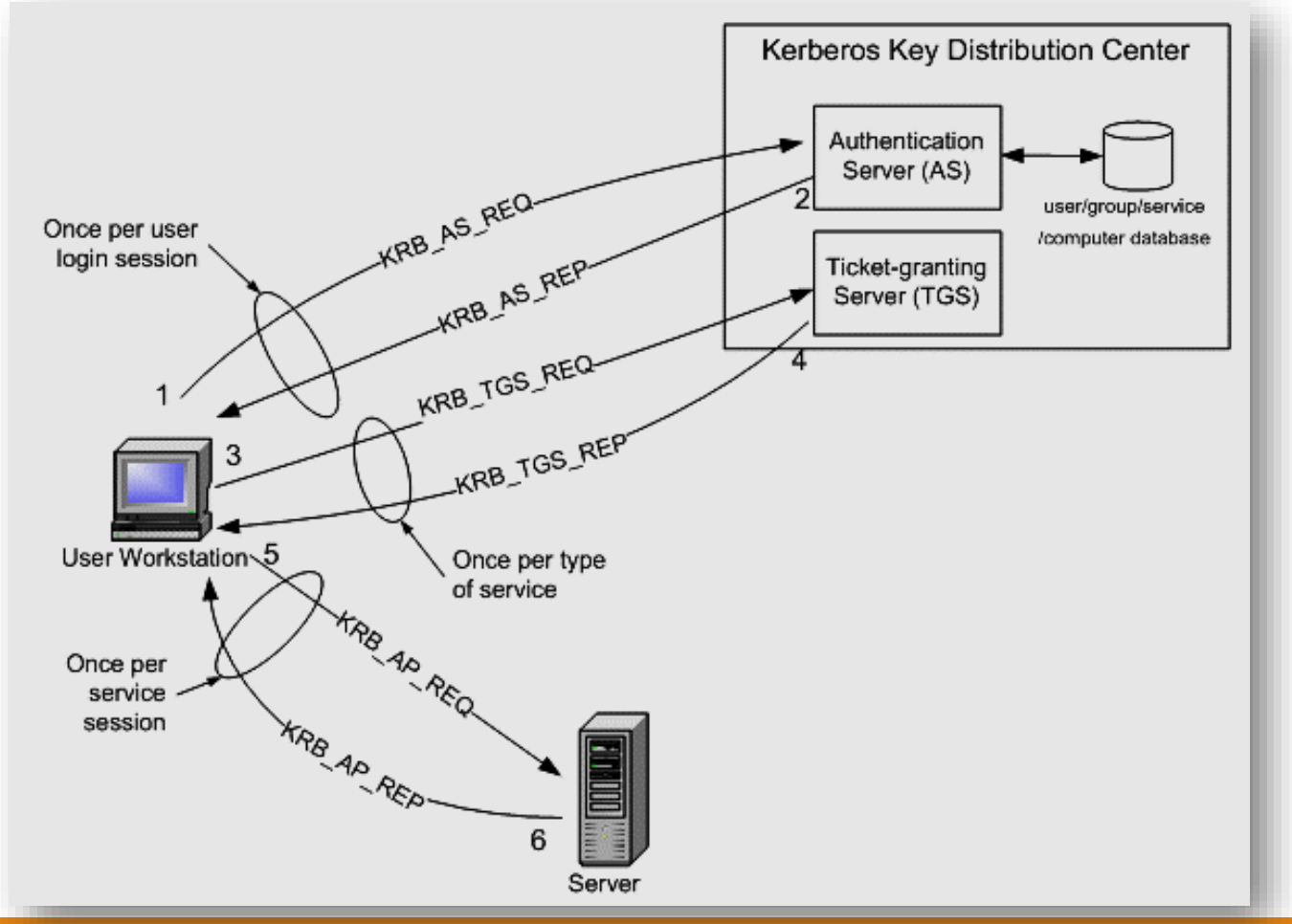
How Does Kerberos Work?

- At first logon, the user provides username and password to AS
- AS then authenticates the user and provides a TGS ticket to the user
- When the user wants to access a service provided by server V, the user provides the TGS its TGS-ticket
- The TGS then authenticates the user's TGS-ticket and issues a V-ticket (server ticket) to the user
- The user provides the V-ticket to server V to obtain service

Kerberos Notations

Notation	Meaning
U	User
V	Server
ID_U	U's ID
ID_{TGS}	TGS's ID
t_i	Time stamp
E_K	Symmetric-key encryption with secret key K
K_U	The secret key derived from user U's password
$K_{U,TGS}$	The session key generated by AS to be used by U and TGS
K_{TGS}	The master key shared by AS and TGS
K_V	The master key shared by TGS and V
$K_{U,V}$	The session key generated by TGS to be used by U and V
LT_i	Expiration time
$Ticket_{TGS}$	TGS-ticket issued to U by AS
$Ticket_V$	Server ticket for using server V issued to U by TGS
AD_U	U's MAC address
$Auth_{U,TGS}$	Authentication code generated using secret key $K_{U,TGS}$
$Auth_{U,V}$	Authentication code generated using secret key $K_{U,V}$

KERBEROS System



THANK YOU

Reference:

- Textbook: Security in Computing, 5th Edition , 2015 by Charles P. Pfleeger.
- Stallings W, Brown L, Bauer MD, Bhattacharjee AK. Computer security: principles and practice. Upper Saddle River, NJ, USA: Pearson Education; 2012.
- Internet resources.