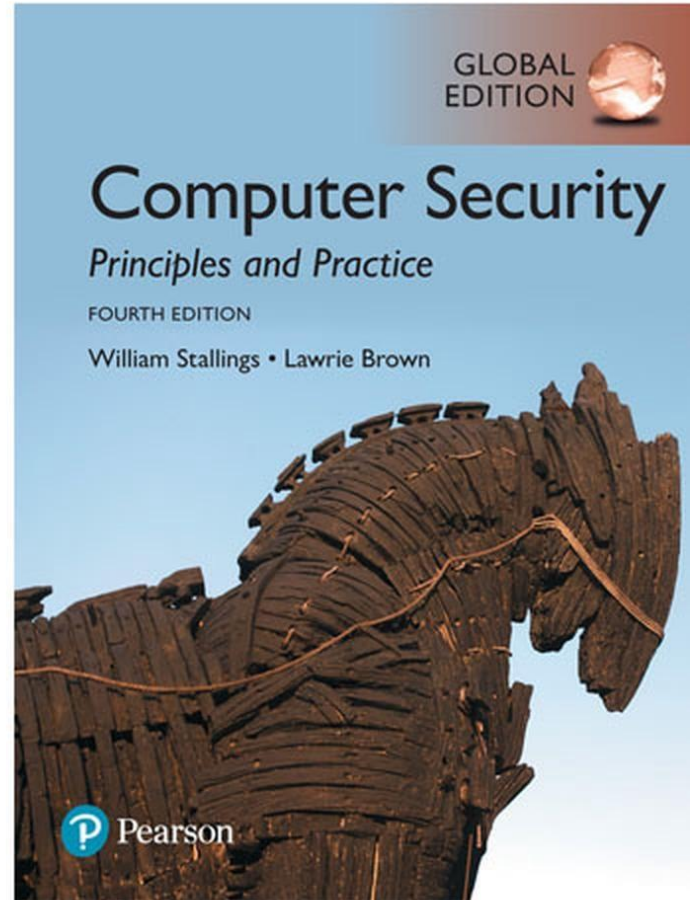
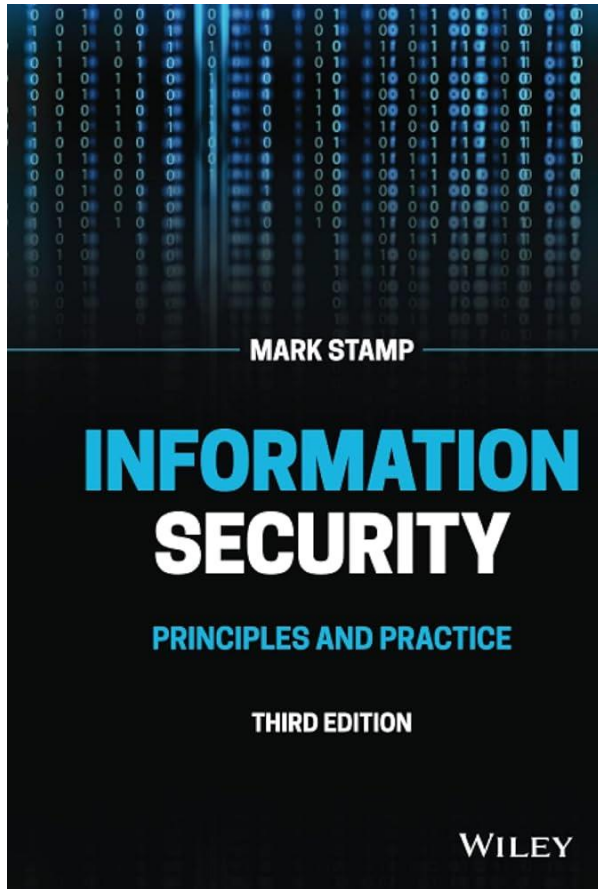


بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

مبانی رایانش امن

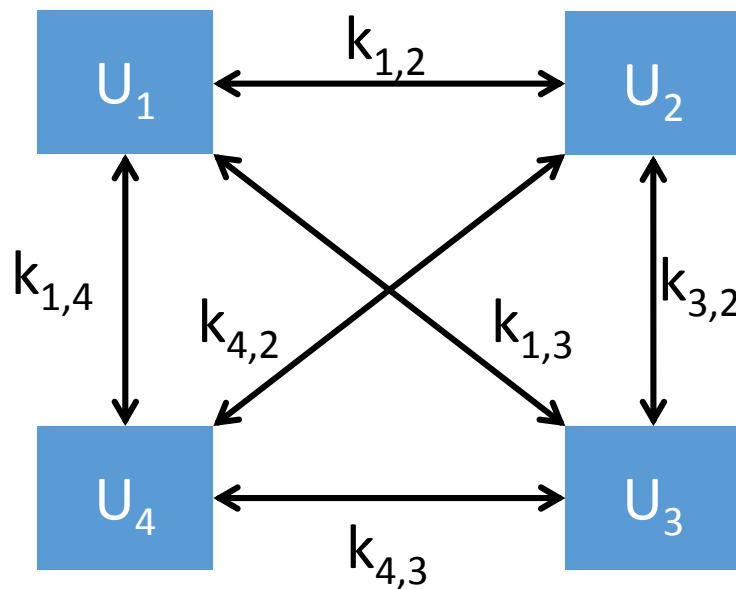
جلسه ۱۰

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان



◀ فصل ۲۲ و ۲۳ استالینگ
◀ فصل ۱۰ استمپ

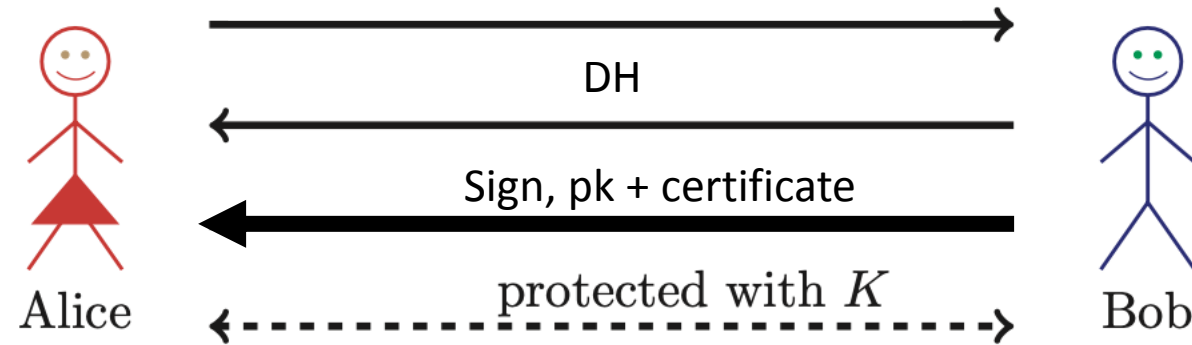
در رمزنگاری متقارن



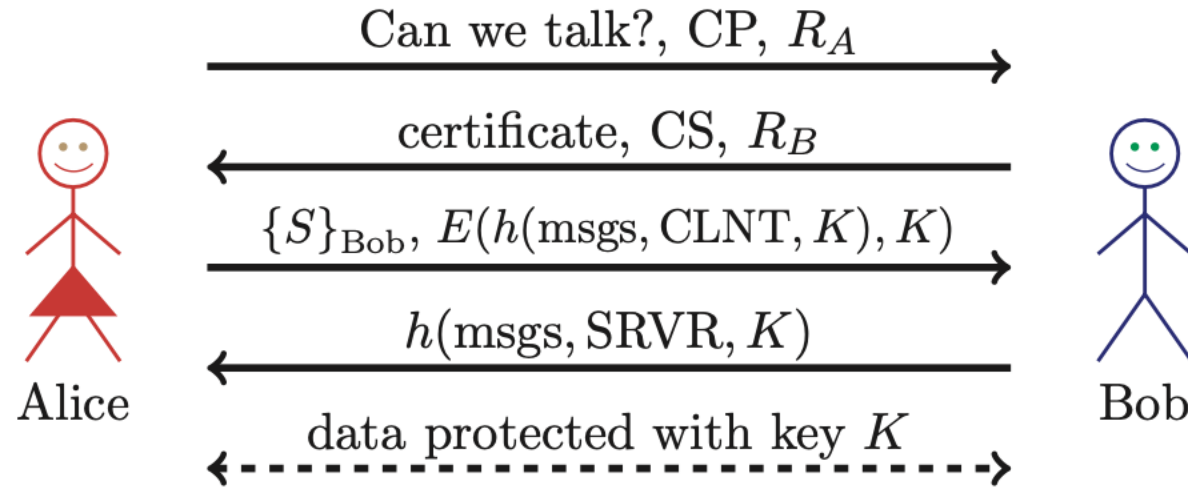
تعداد کلید برای n کاربر؟ ◀

پروتکل امن

Certificate Authority (CA)



پروتکل امن



- S is the so-called **pre-master secret**
- $K = h(S, R_A, R_B)$
- “msgs” means all previous messages
- CLNT and SRVR are constants

Certificate

- A certificate for a user Alice (or Bob) in its most basic form is the following structure:

$$\text{Cert}_A = [(k_{pub,A}, ID_A), \text{sig}_{k_{pr}}(k_{pub,A}, ID_A)]$$



CA

Certificate

Certificate Generation with User-Provided Keys

Alice

generate $k_{pr,A}, k_{pub,A}$

CA

$\xrightarrow{\text{RQST}(k_{pub,A}, ID_A)}$

verify ID_A

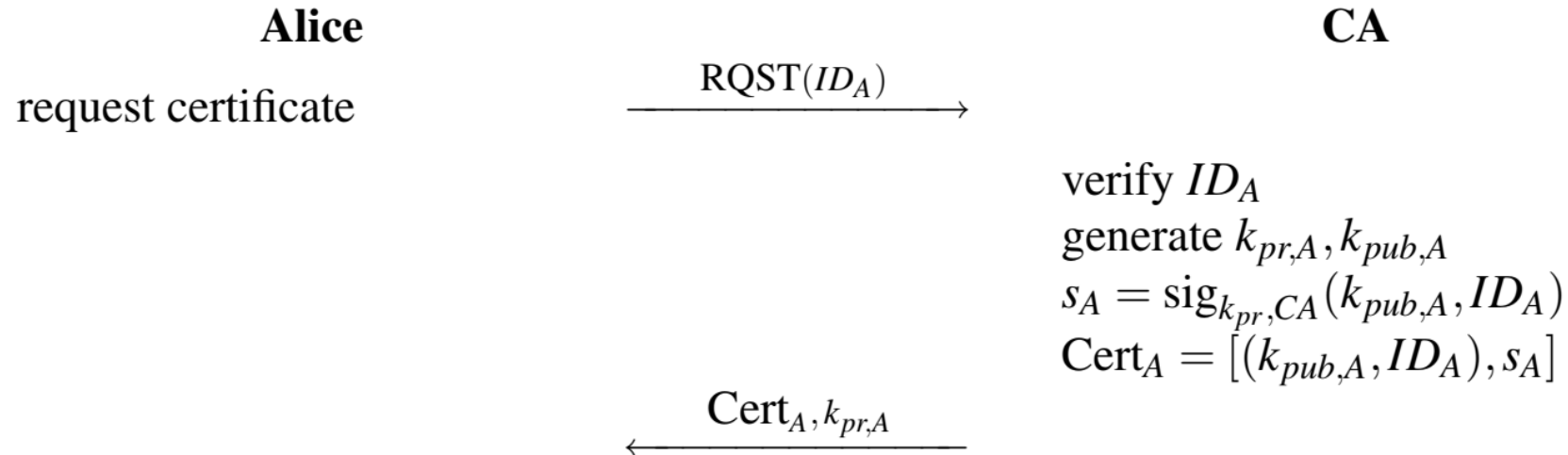
$s_A = \text{sig}_{k_{pr,CA}}(k_{pub,A}, ID_A)$

$\text{Cert}_A = [(k_{pub,A}, ID_A), s_A]$

$\xleftarrow{\text{Cert}_A}$

Certificate

Certificate Generation with CA-Generated Keys



Certificate

- The signatures for certificates are provided by a mutually trusted third party. This party is called the *Certification Authority* commonly abbreviated as *CA*.
- It is said that **certificates bind the identity of a user to their public key**.

Certificate

Diffie–Hellman Key Exchange with Certificates

Alice

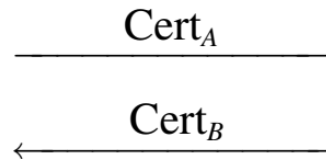
$$\begin{aligned}a &= k_{pr,A} \\ A &= k_{pub,A} \equiv \alpha^a \text{ mod } p \\ \text{Cert}_A &= [(A, ID_A), s_A]\end{aligned}$$

verify certificate:
 $\text{ver}_{k_{pub,CA}}(\text{Cert}_B)$
compute session key:
 $k_{AB} \equiv B^a \text{ mod } p$

Bob

$$\begin{aligned}b &= k_{pr,B} \\ B &= k_{pub,B} \equiv \alpha^B \text{ mod } p \\ \text{Cert}_B &= [(B, ID_B), s_B]\end{aligned}$$

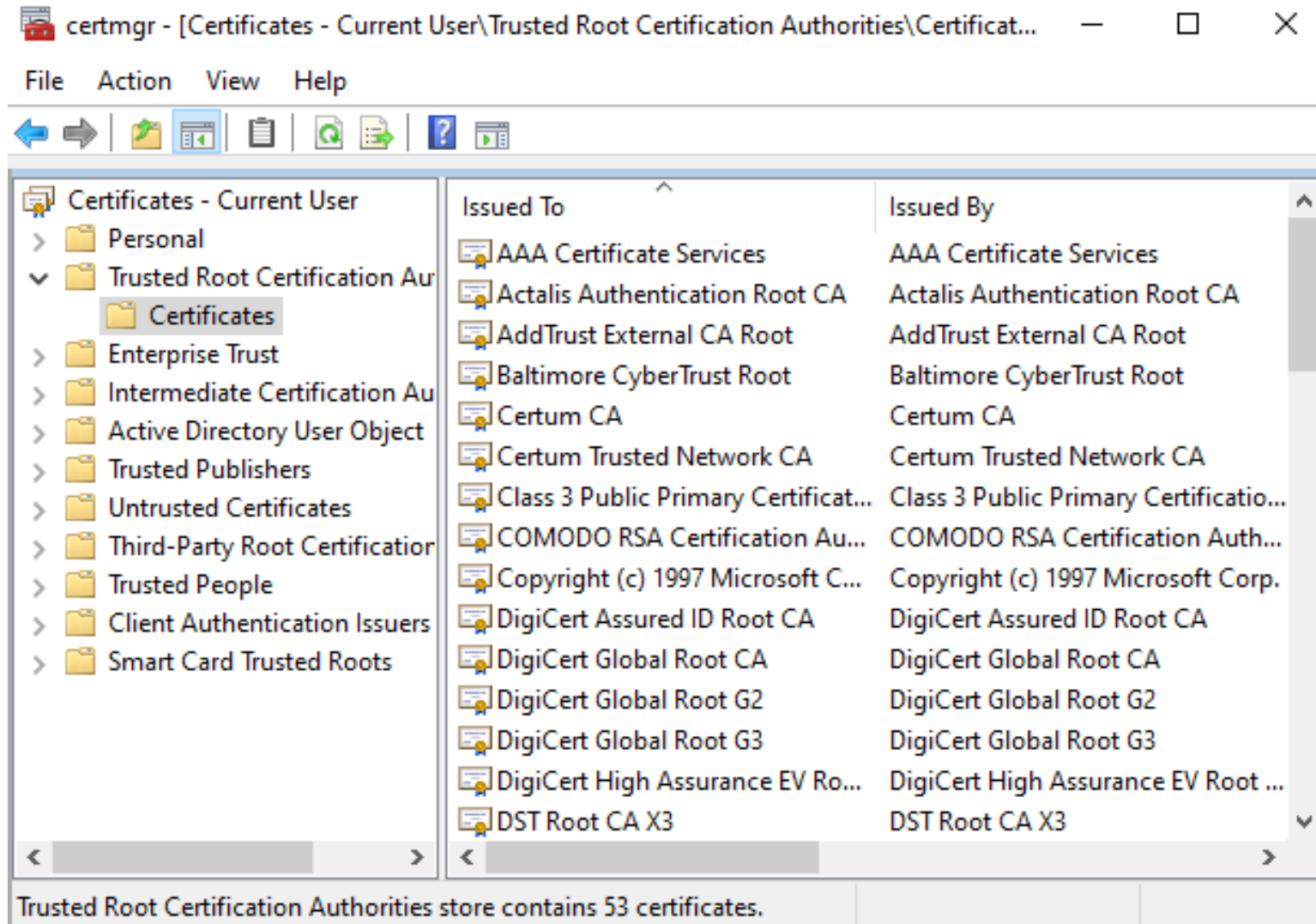
verify certificate:
 $\text{ver}_{k_{pub,CA}}(\text{Cert}_A)$
compute session key:
 $k_{AB} \equiv A^b \text{ mod } p$



Certificate

- Public verification keys are nowadays often included in PC software such as Web browsers or Microsoft software products. The authenticated channel is here assumed to be given through the installation of original software which has not been manipulated.

Certificate



public-key infrastructure

The entire system that is formed by CAs together with the necessary support mechanisms is called a *public-key infrastructure*, usually referred to as *PKI*.

Certificate

- In practice, certificates not only include the ID and the public key of a user, they tend to be quite complex structures with many additional fields. As an example, we look at the a X.509 certificate in Fig. 13.4. X.509 is an important standard for network authentication services, and the corresponding certificates are widely used for Internet communication.

Serial Number
Certificate Algorithm: <ul style="list-style-type: none">- Algorithm- Parameters
Issuer
Period of Validity: <ul style="list-style-type: none">- Not Before Date- Not After Date
Subject
Subject's Public Key: <ul style="list-style-type: none">- Algorithm- Parameters- Public Key
Signature

Fig. 13.4 Detailed structure of an X.509 certificate

public-key infrastructure

Let's look at an example where Alice's certificate is issued by CA1 and Bob's by CA2. At the moment, Alice is only in possession of the public key of "her" CA1, and Bob has only $k_{pub,CA2}$. If Bob sends his certificate to Alice, she cannot verify Bob's public key. This situation looks like this:

Two Users with Different Certificate Authorities

Alice

$k_{pub,CA1}$

Bob

$k_{pub,CA2}$

$Cert_B = [(k_{pub,B}, ID_B), \text{sig}_{k_{pr,CA2}}(k_{pub,B}, ID_B)]$

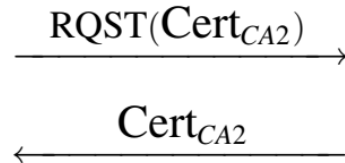
$\leftarrow Cert_B$

Alice can now request CA2's public key, which is itself contained in a certificate that was signed by Alice's CA1:

Verification of a CA Public Key

Alice

CA2

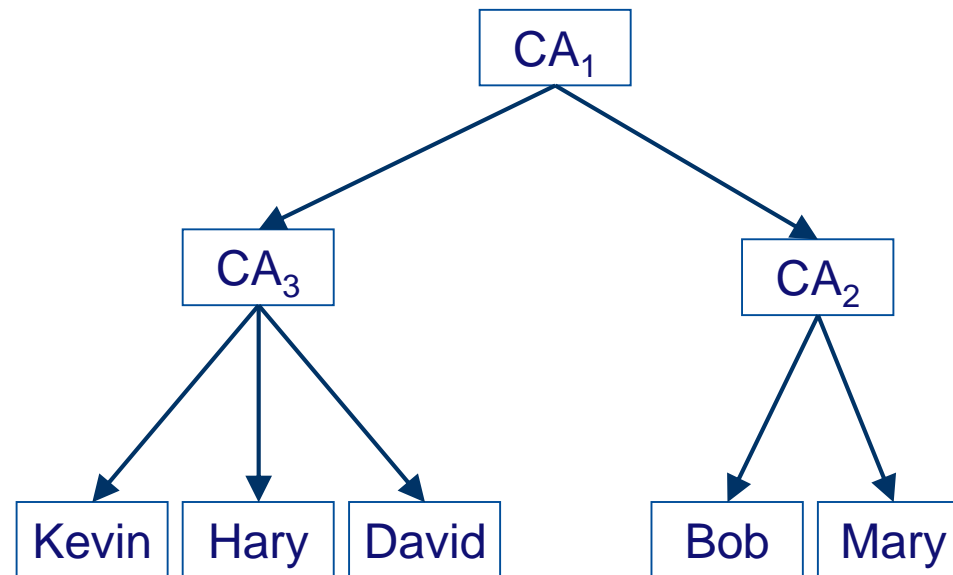


$\text{ver}_{k_{pub,CA1}}(\text{Cert}_{CA2})$
 $\Rightarrow k_{pub,CA2}$ is valid
 $\text{ver}_{k_{pub,CA2}}(\text{Cert}_B)$
 $\Rightarrow k_{pub,B}$ is valid

The structure Cert_{CA2} contains the public key of CA2 signed by CA1, which looks like this:

$$\text{Cert}_{CA2} = [(k_{pub,CA2}, ID_{CA2}), \text{sig}_{k_{pr,CA1}}(k_{pub,CA2}, ID_{CA2})]$$

Certificate



Certificate



Certificate Viewer: iut.ac.ir

General

Details

Issued To

Common Name (CN) iut.ac.ir
Organization (O) <Not Part Of Certificate>
Organizational Unit (OU) <Not Part Of Certificate>

Issued By

Common Name (CN) E5
Organization (O) Let's Encrypt
Organizational Unit (OU) <Not Part Of Certificate>

Validity Period

Issued On Sunday, January 12, 2025 at 11:02:15 AM
Expires On Saturday, April 12, 2025 at 11:02:14 AM

Fingerprints

SHA-256 Fingerprint 3C F9 63 6E D1 B7 04 CD 70 89 44 05 87 D1 C1 E6
58 3D 3D AB 05 99 0D 19 0B C6 6B A1 6F D5 BE 8E
SHA-1 Fingerprint D0 52 10 4A 6A 11 AF 83 0E 25 E3 E8 AB 49 7E 84
21 C2 52 67



Certificate Viewer: iut.ac.ir

General

Details

Certificate Hierarchy

ISRG Root X1

E5

iut.ac.ir

Certificate Fields

iut.ac.ir

Certificate

Version

Serial Number

Certificate Signature Algorithm

Issuer

Validity

Not Before

Field Value

X9.62 ECDSA Signature with SHA-384

Export...