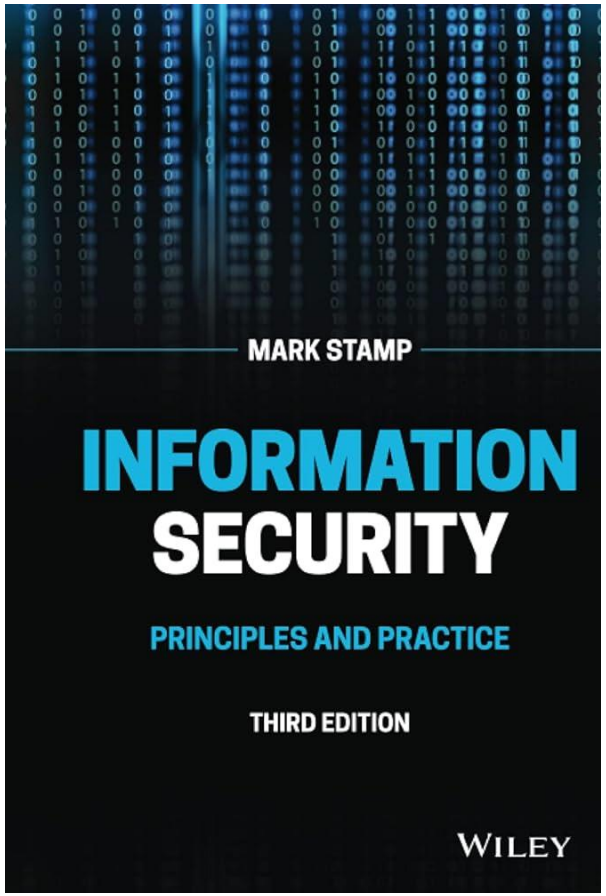
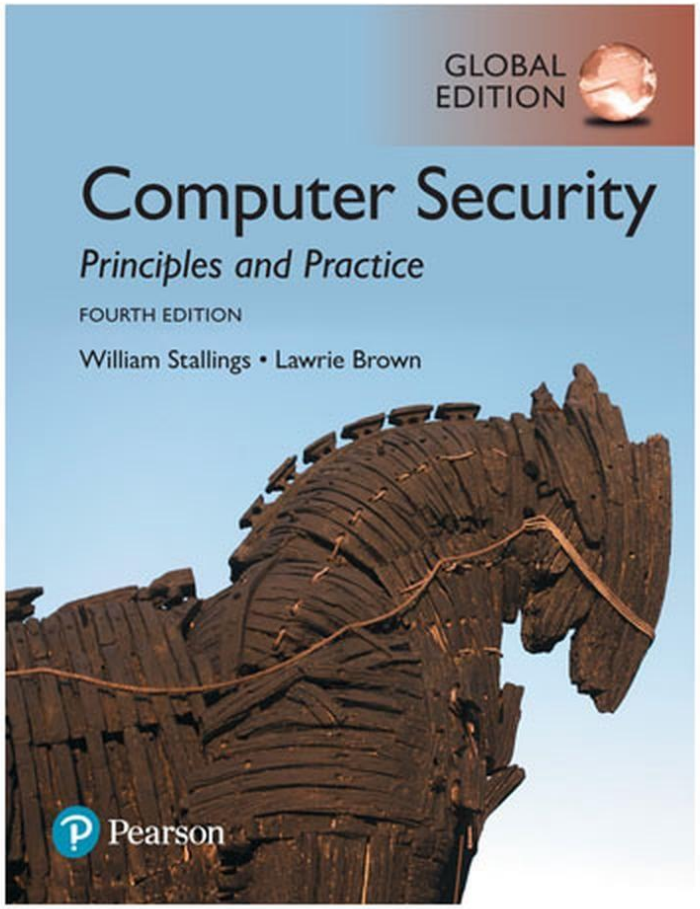
بسم الله الرحمن الرحیم

مبانی رایانش امن

جلسه ۱۰

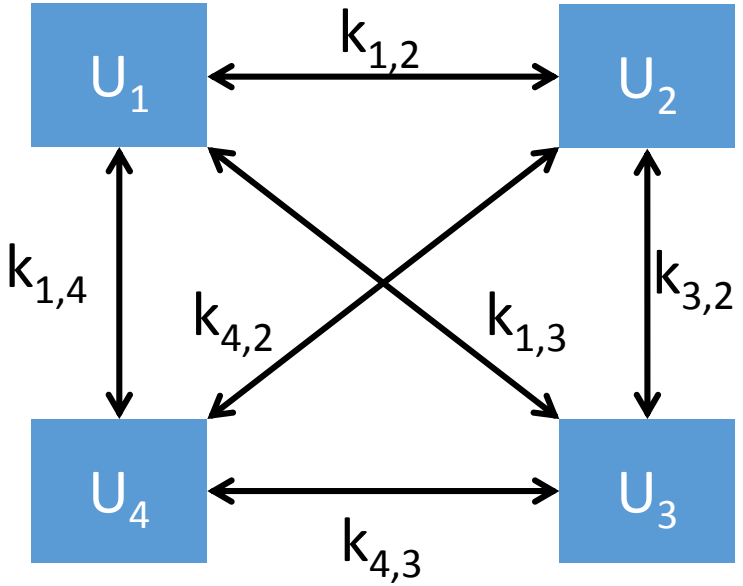
مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان





- فصل ۲۲ و ۲۳ استالینگ
- فصل ۱۰ استمپ

در رمزنگاری متقارن

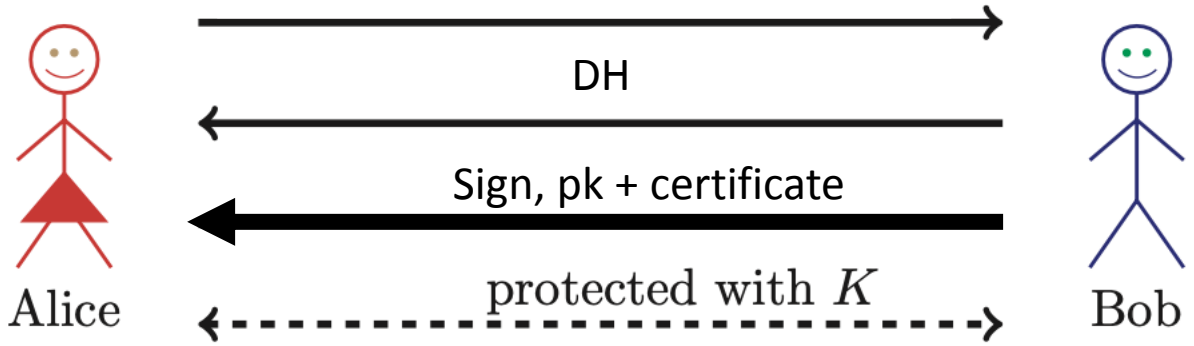


- تعداد کلید برای n کاربر؟

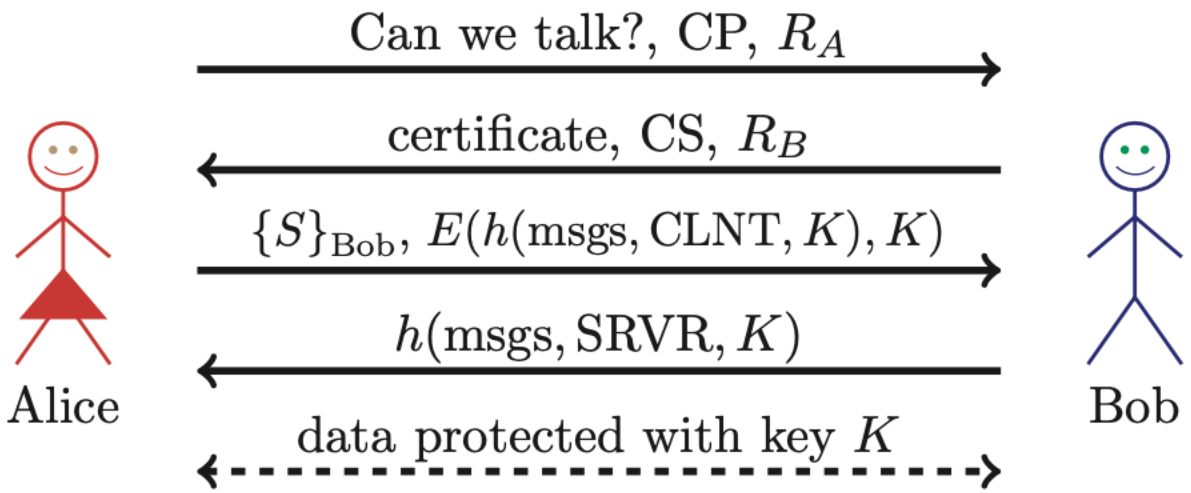
پروتکل امن

IUT-ECE

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

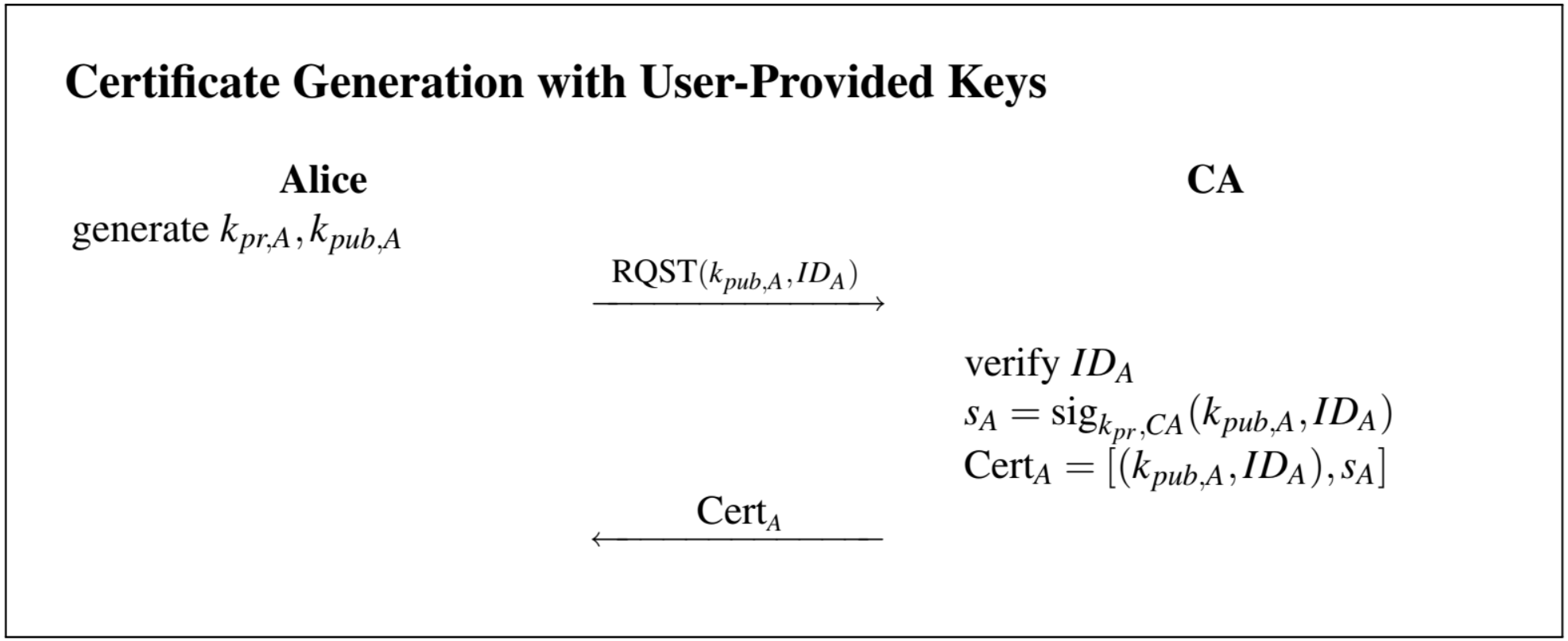
Certificate

- 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

Certificate Generation with CA-Generated Keys

Alice		CA
request certificate	$\xrightarrow{\text{RQST}(ID_A)}$	
		verify ID_A generate $k_{pr,A}, k_{pub,A}$ $s_A = \text{sig}_{k_{pr},CA}(k_{pub,A}, ID_A)$ $\text{Cert}_A = [(k_{pub,A}, ID_A), s_A]$
	$\xleftarrow{\text{Cert}_A, k_{pr,A}}$	

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

IUT-ECE

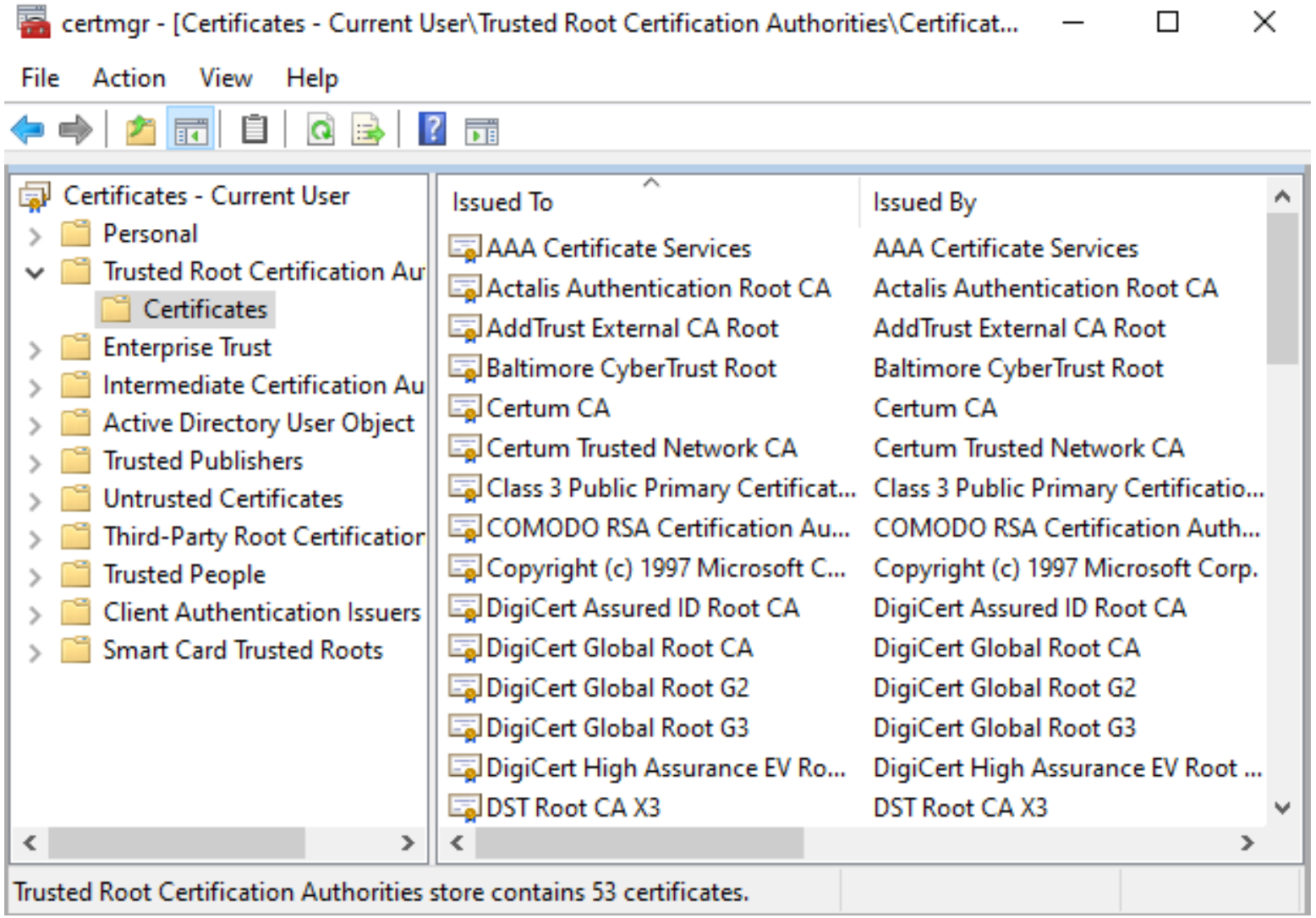
Diffie–Hellman Key Exchange with Certificates

Alice		Bob
$a = k_{pr,A}$		$b = k_{pr,B}$
$A = k_{pub,A} \equiv \alpha^a \bmod p$		$B = k_{pub,B} \equiv \alpha^B \bmod p$
$\text{Cert}_A = [(A, ID_A), s_A]$		$\text{Cert}_B = [(B, ID_B), s_B]$
	$\text{Cert}_A \longrightarrow$	
	$\longleftarrow \text{Cert}_B$	
verify certificate:		verify certificate:
$\text{ver}_{k_{pub,CA}}(\text{Cert}_B)$		$\text{ver}_{k_{pub,CA}}(\text{Cert}_A)$
compute session key:		compute session key:
$k_{AB} \equiv B^a \bmod p$		$k_{AB} \equiv A^b \bmod p$

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

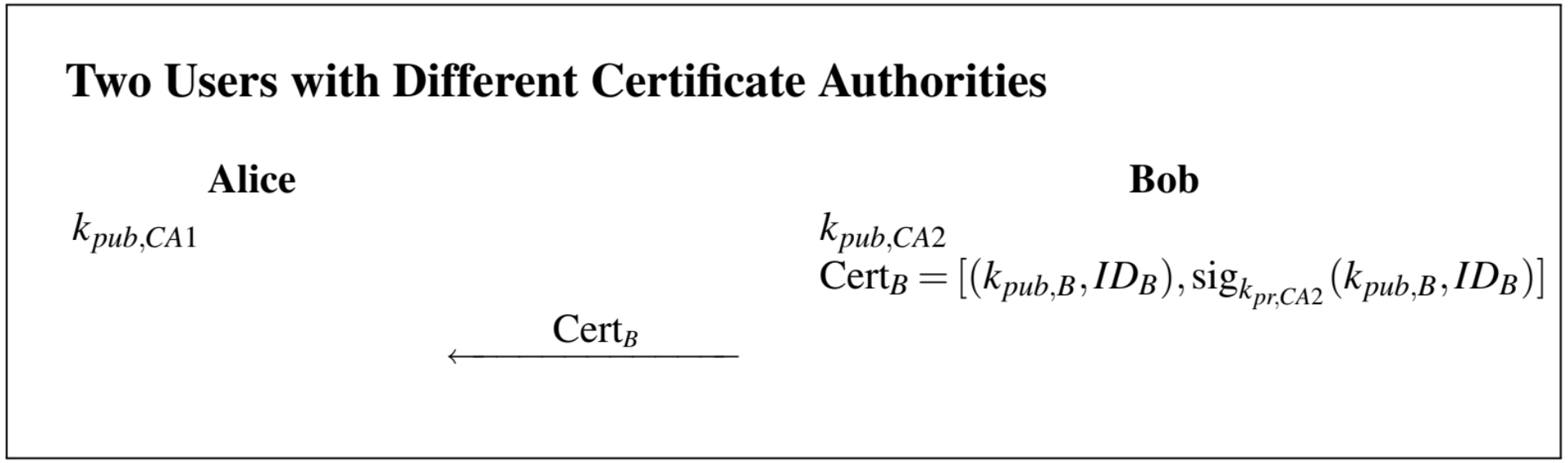
- In practice, certificates not only include the ID and the public key of a user, they tend to be quite complex structures with many additional fields. As an example, we look at the a X.509 certificate in Fig. 13.4. X.509 is an important standard for network authentication services, and the corresponding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

Serial Number
Certificate Algorithm: - Algorithm - Parameters
Issuer
Period of Validity: - Not Before Date - Not After Date
Subject
Subject's Public Key: - 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:



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

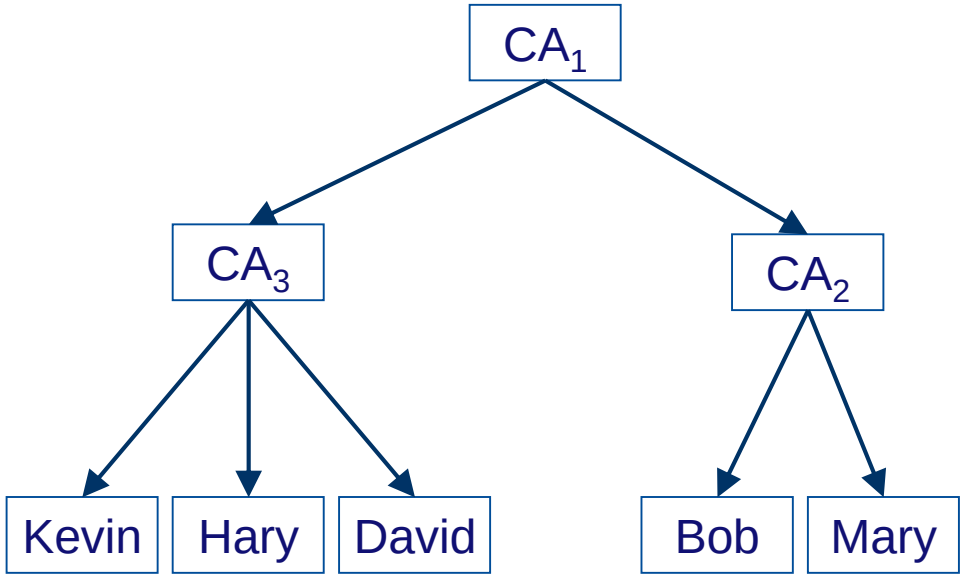
Alice		**CA2**
	$\xrightarrow{\text{RQST}(\text{Cert}_{CA2})}$	
	$\xleftarrow{\text{Cert}_{CA2}}$	

$\text{ver}_{k_{pub,CA1}}(\text{Cert}_{CA2})$
$\Rightarrow k_{pub,CA2}$ 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:

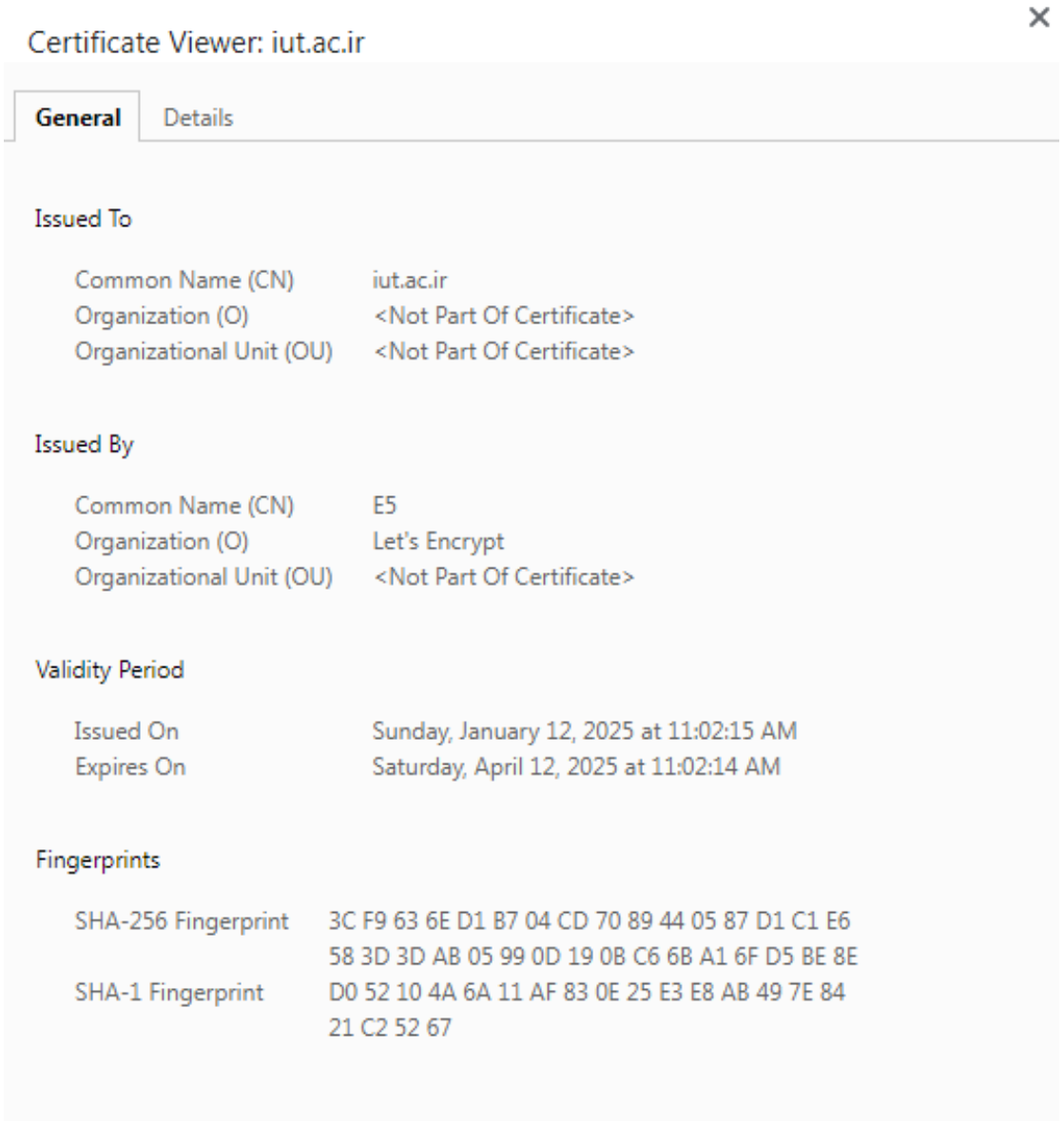
$$\text{Cert}_{CA2} = [(k_{pub,CA2}, ID_{CA2}), \text{sig}_{k_{pr,CA1}}(k_{pub,CA2}, ID_{CA2})]$$

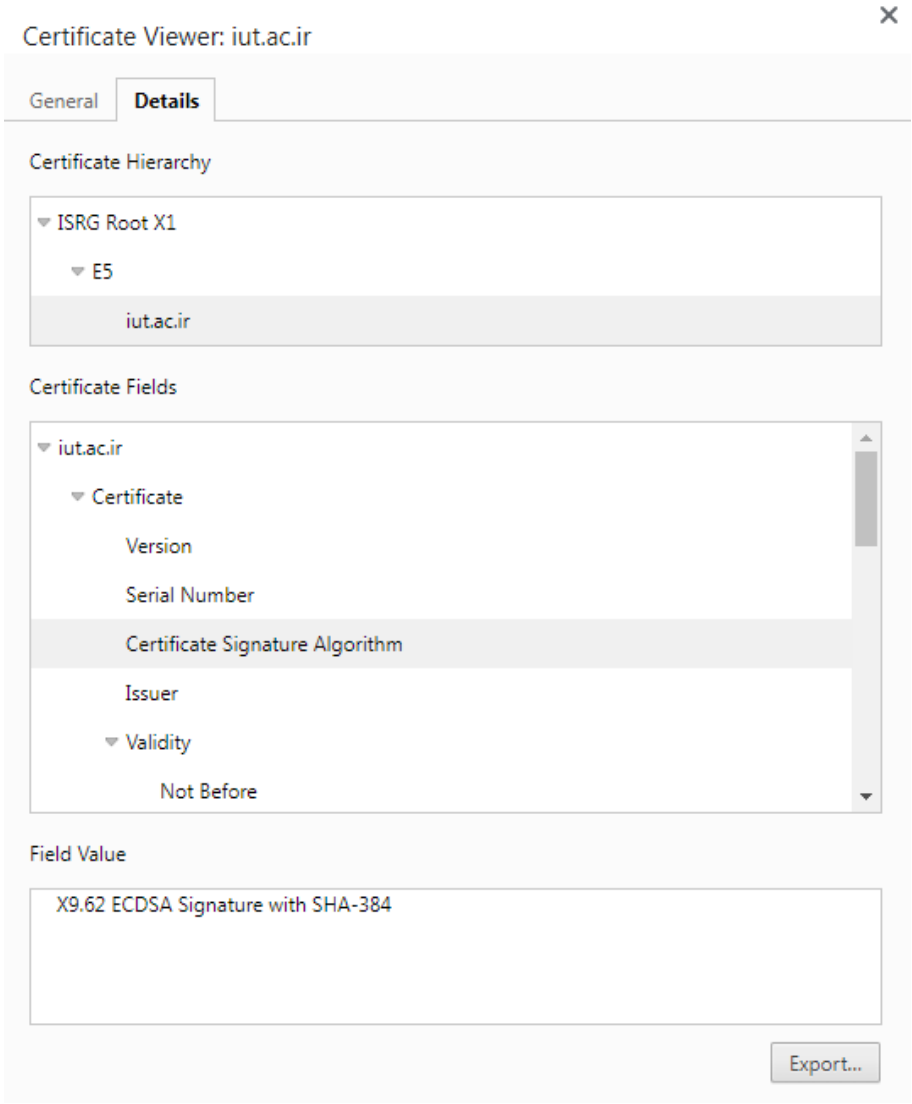
Certificate



Certificate

IUT-ECE





Mojtaba Khalili