
Lecture 12 – Digital signatures, UF-CMA, RSA, PKI

TEK4500

09.11.2022

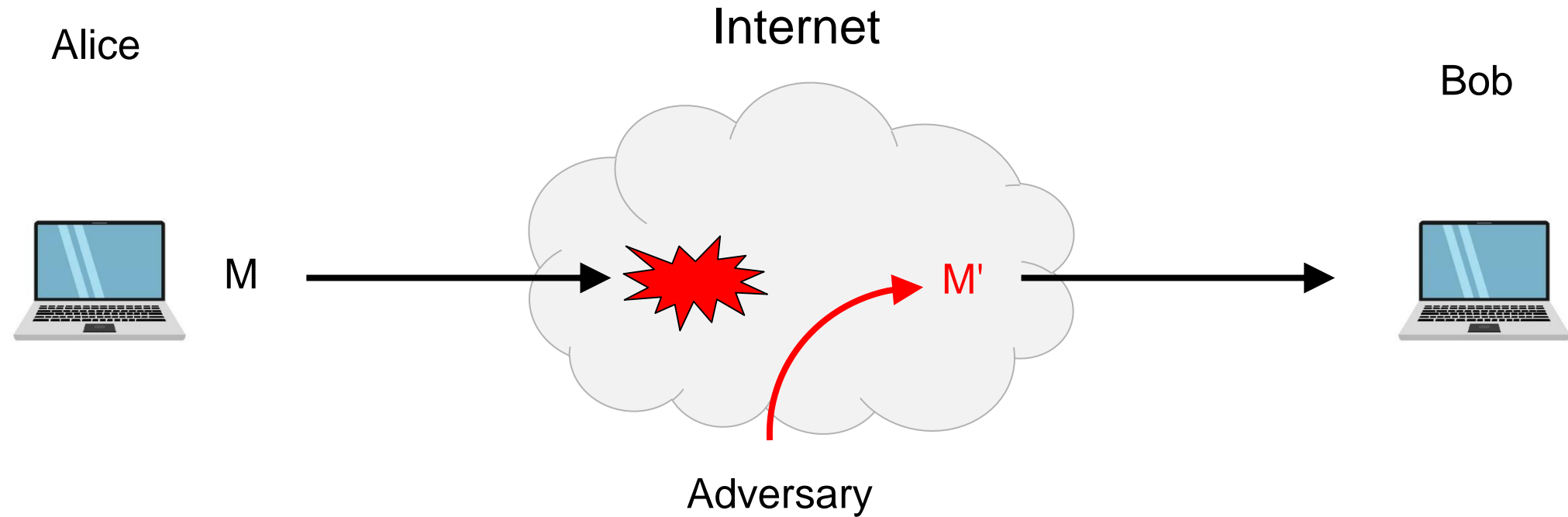
Håkon Jacobsen

hakon.jacobsen@its.uio.no

Basic goals of cryptography

	Message privacy	Message integrity / authentication
Symmetric keys	Symmetric encryption	Message authentication codes (MAC)
Asymmetric keys	Asymmetric encryption (a.k.a. public-key encryption)	Digital signatures (Key exchange)

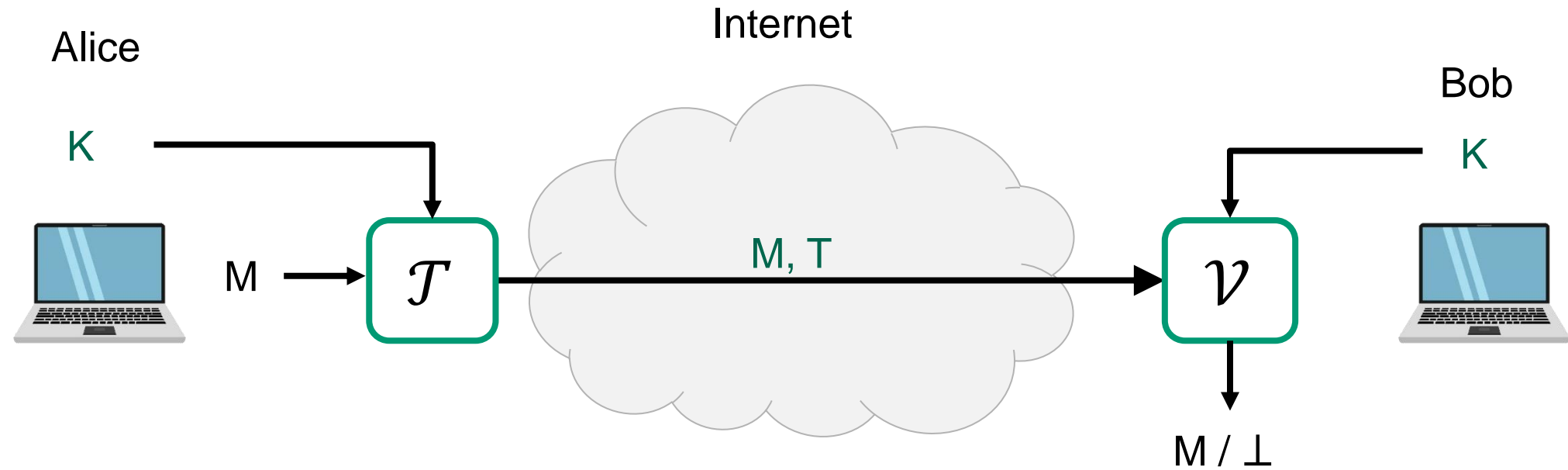
What is cryptography?



Security goals:

- **Data privacy:** adversary should not be able to read message M
- **Data integrity:** adversary should not be able to modify message M
- **Data authenticity:** message M really originated from Alice

MACs

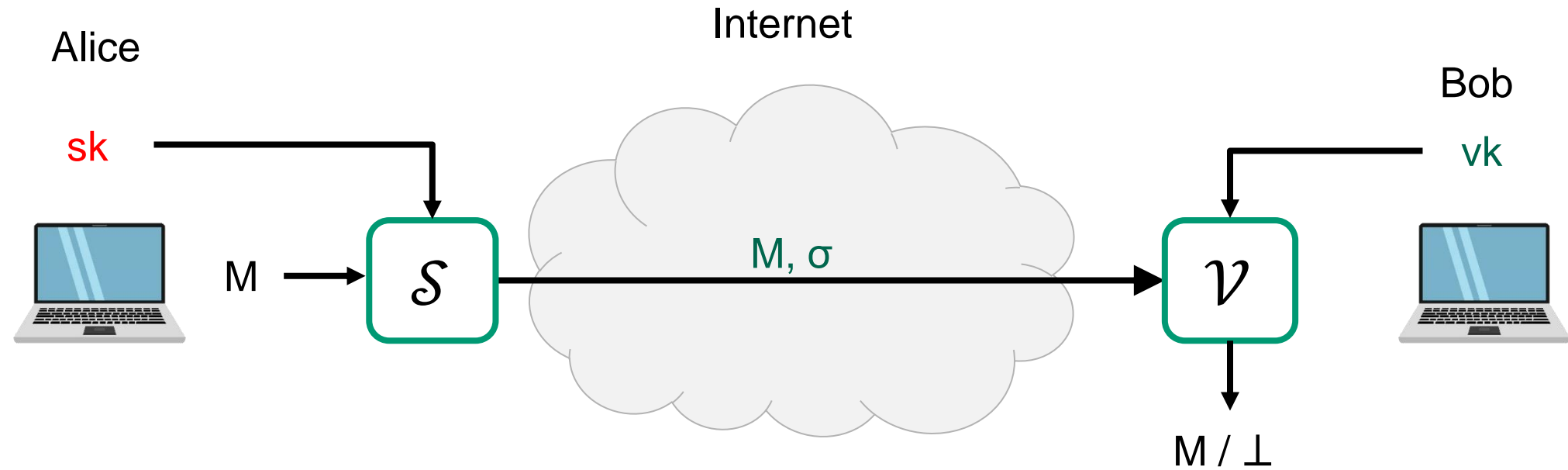


\mathcal{T} : tagging algorithm (public)

K : tagging / verification key (secret)

\mathcal{V} : verification algorithm (public)

Digital signatures



\mathcal{T} : tagging algorithm (public)

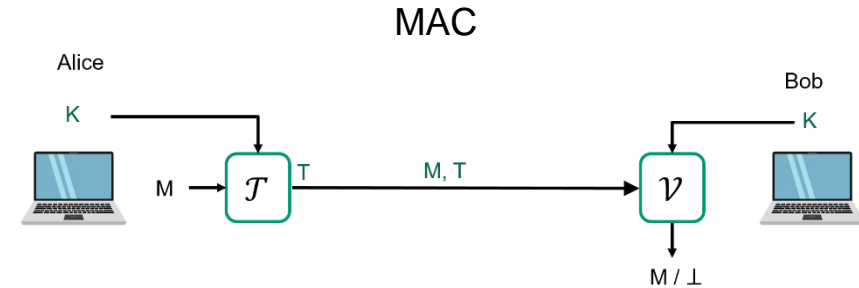
sk : signing key (secret)

\mathcal{V} : verification algorithm (public)

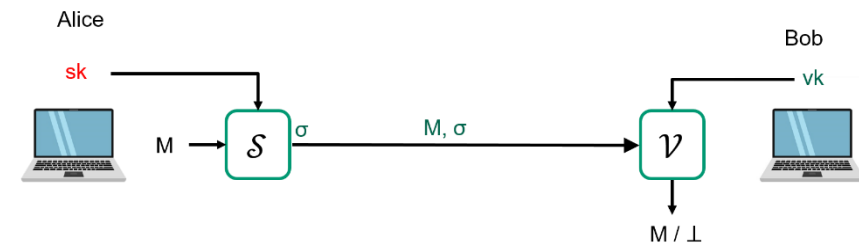
vk : verification key (public)

MACs vs. digital signatures

- MACs can only be verified by party sharing the same key



- Digital signatures can be verified by *anyone*

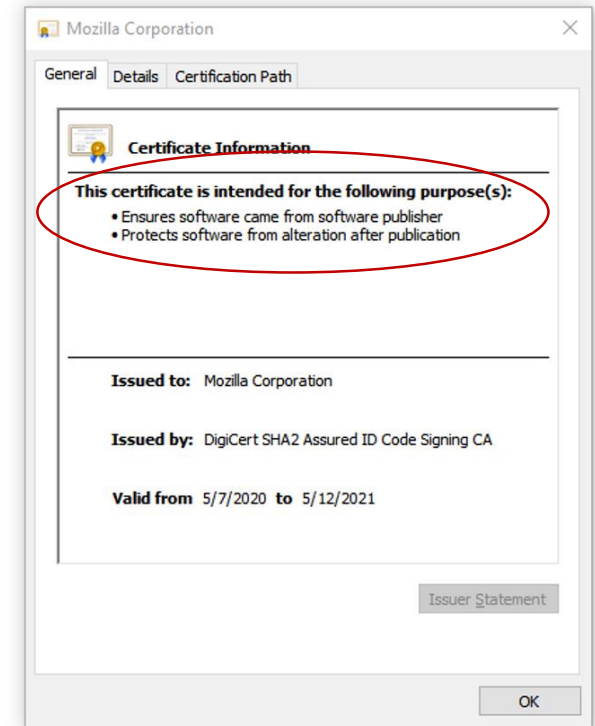
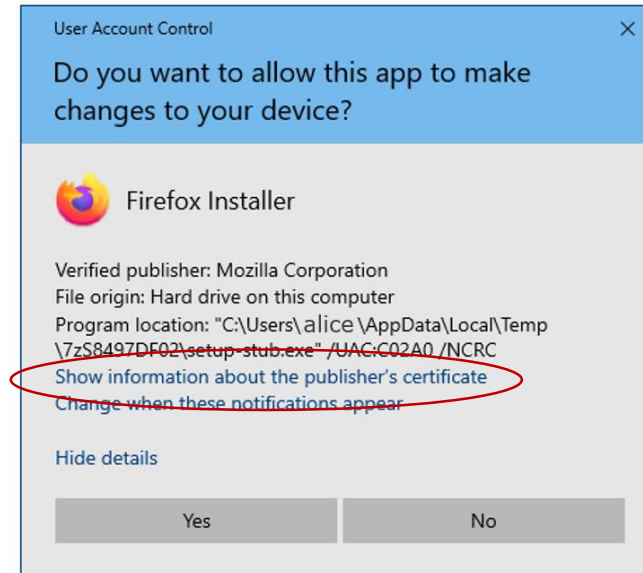


Digital signature

- **Non-repudiation:** Alice cannot deny having created σ
 - But she can deny having created T (since Bob could have done it)

Applications of digital signatures

- Electronic document signing
- HTTPS / TLS certificates
- Software installation
- Email sender authentication
- Bitcoin



Digital signatures – syntax

A **digital signature** scheme is a tuple of algorithms $\Sigma = (\text{KeyGen}, \text{Sign}, \text{Vrfy})$

$$\text{KeyGen} : () \rightarrow \mathcal{SK} \times \mathcal{VK}$$

$$(sk, vk) \stackrel{\$}{\leftarrow} \text{KeyGen}$$

$$\text{Sign} : \mathcal{SK} \times \mathcal{M} \rightarrow \mathcal{S}$$

$$\text{Sign}(sk, M) = \text{Sign}_{sk}(M) = \sigma$$

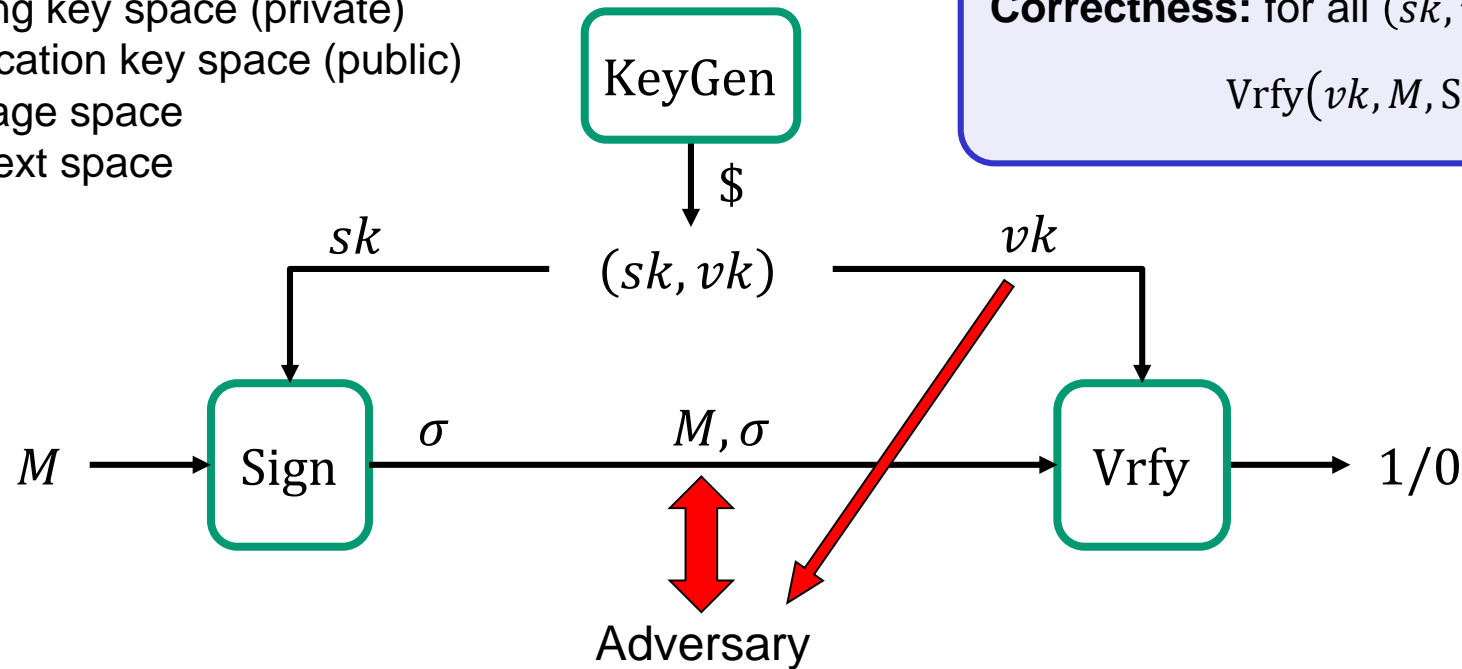
$$\text{Vrfy} : \mathcal{VK} \times \mathcal{M} \times \mathcal{S} \rightarrow \{0,1\}$$

$$\text{Vrfy}(vk, M, \sigma) = \text{Vrfy}_{vk}(M, \sigma) = 1/0$$

- \mathcal{SK} – signing key space (private)
- \mathcal{VK} – verification key space (public)
- \mathcal{M} – message space
- \mathcal{C} – ciphertext space

Correctness: for all $(sk, vk) \leftarrow \text{KeyGen}$:

$$\text{Vrfy}(vk, M, \text{Sign}(sk, M)) = 1$$



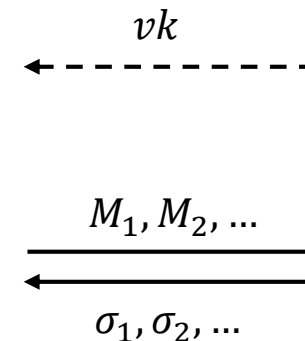
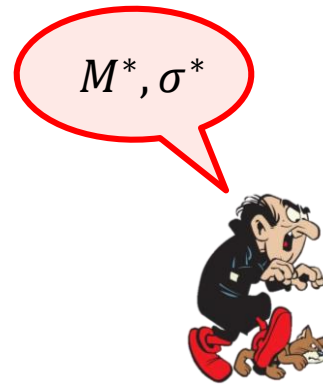
Digital signatures – security: UF-CMA

$\text{Exp}_{\Sigma}^{\text{uf-cma}}(A)$

1. $(sk, vk) \xleftarrow{\$} \Sigma.\text{KeyGen}$
2. $\text{Msgs} \leftarrow []$ // bookkeeping
3. $(M^*, \sigma^*) \leftarrow A^{\mathcal{S}(\cdot)}(vk)$
4. **if** $\Sigma.\text{Vrfy}(vk, M^*, \sigma^*) = 1$ and $M^* \notin \text{Msgs}$:
5. **return** 1
6. **else**
7. **return** 0

$\mathcal{S}(M)$

-
1. $\sigma \leftarrow \Sigma.\text{Sign}(sk, M)$
 2. $\text{Msgs.add}(M)$
 3. **return** σ



Challenger

$(sk, vk) \xleftarrow{\$} \text{KeyGen}$

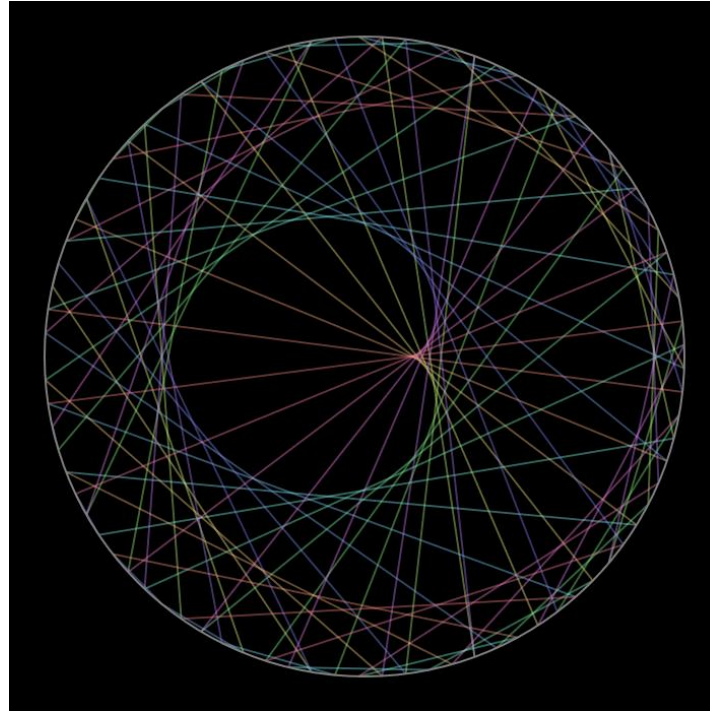


$\text{Sign}(sk, \cdot)$

A has **forged** a signature if σ^* is valid for M^*

Definition: The **UF-CMA-advantage** of an adversary A is

$$\text{Adv}_{\Sigma}^{\text{uf-cma}}(A) = \Pr[\text{Exp}_{\Sigma}^{\text{uf-cma}}(A) \Rightarrow 1]$$



RSA signatures

Textbook RSA signatures

$$C = M^e \pmod n$$

Textbook RSA signatures

$$C^e = M^{de} \bmod n = M$$

Textbook RSA signatures

$$\sigma^e = M^d \bmod n$$

Textbook RSA signatures

RSA.Vrfy($(n, e), M, \sigma$)

1. **if** $\sigma^e = M \bmod N$ **then**
2. **return** 1
3. **else**
4. **return** 0



$M, \sigma = M^d \bmod n$



RSA.KeyGen

1. $p, q \overset{\$}{\leftarrow}$ two random prime numbers
2. $n \leftarrow p \cdot q$
3. $\phi(n) = (p - 1)(q - 1)$
4. **choose** e such that $\gcd(e, \phi(n)) = 1$
5. $d \leftarrow e^{-1} \bmod \phi(n)$
6. $sk \leftarrow (n, d)$ $pk \leftarrow (n, e)$
7. **return** (sk, pk)

RSA.Sign($(n, d), M$)

1. $\sigma \leftarrow M^d \bmod N$
2. **return** σ

Textbook RSA signatures – (in)security

A₁

- Output $(M, \sigma) = (1, 1)$

$$\text{Adv}_{\text{RSA}}^{\text{uf-cma}}(A_1) = 1$$

A₂

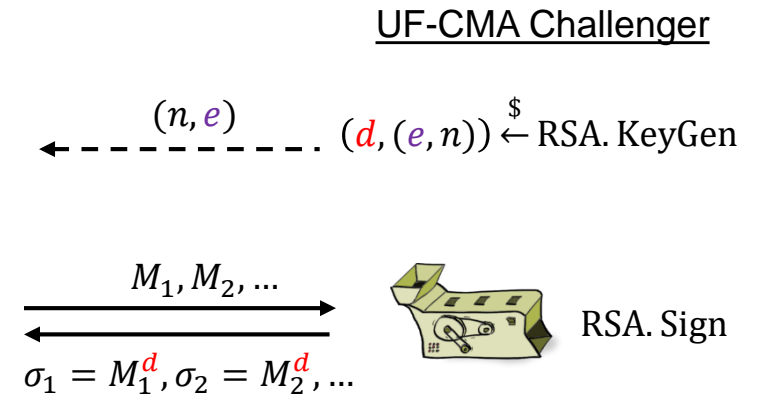
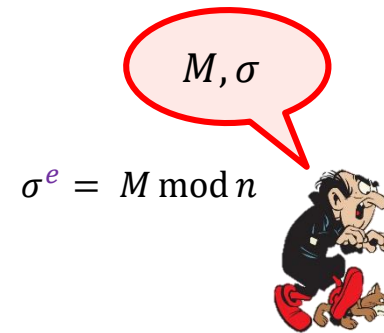
- Pick any $X \neq M$

Z_n^{} is group!*

$$\text{Adv}_{\text{RSA}}^{\text{uf-cma}}(A_2) = 1$$

$$(\sigma_1 \cdot \sigma_2)^e \stackrel{?}{=} M \pmod n$$

$$(X^d \cdot Y^d)^e = X^{ed} \cdot Y^{ed} = X \cdot Y = X \cdot X^{-1} \cdot M = M \pmod n$$



Textbook RSA signatures

$$H : \{0,1\}^* \rightarrow \mathbb{Z}_n^*$$

RSA.Vrfy($(n, e), M, \sigma$)

1. if $\sigma^e = M \bmod N$ then
2. return 1
3. else
4. return 0



$$M, \sigma = M^d \bmod n$$



RSA.KeyGen

1. $p, q \xleftarrow{\$}$ two random prime numbers
2. $n \leftarrow p \cdot q$
3. $\phi(n) = (p - 1)(q - 1)$
4. choose e such that $\gcd(e, \phi(n)) = 1$
5. $d \leftarrow e^{-1} \bmod \phi(n)$
6. $sk \leftarrow (n, d)$ $pk \leftarrow (n, e)$
7. return (sk, pk)

RSA.Sign($(n, d), M$)

1. $\sigma \leftarrow M^d \bmod N$
2. return σ

RSA message space:

$$\mathcal{M} = \mathbb{Z}_n^*$$

$$\mathcal{M} = \{0,1\}^*$$



Actually want

Hashed-RSA

$$H : \{0,1\}^* \rightarrow \mathbb{Z}_n^*$$

RSA.Vrfy($(n, e), M, \sigma$)

1. **if** $\sigma^e = H(M) \bmod N$ **then**
2. **return** 1
3. **else**
4. **return** 0



$$M, \sigma = H(M)^d \bmod n$$



RSA.KeyGen

1. $p, q \stackrel{\$}{\leftarrow}$ two random prime numbers
2. $n \leftarrow p \cdot q$
3. $\phi(n) = (p - 1)(q - 1)$
4. **choose** e such that $\gcd(e, \phi(n)) = 1$
5. $d \leftarrow e^{-1} \bmod \phi(n)$
6. $sk \leftarrow (n, d)$ $pk \leftarrow (n, e)$
7. **return** (sk, pk)

RSA.Sign($(n, d), M$)

1. $\sigma \leftarrow H(M)^d \bmod N$
2. **return** σ

Hashed-RSA – security

A₁

1. ~~Output (M, σ) = (1,1)~~ Output (X, 1) // H(X) = 1

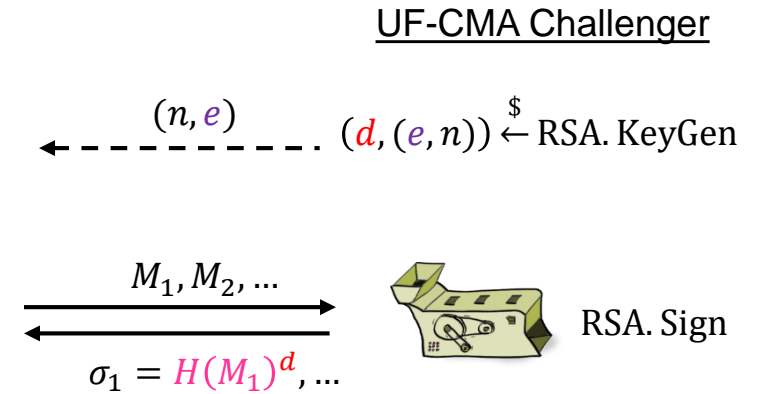
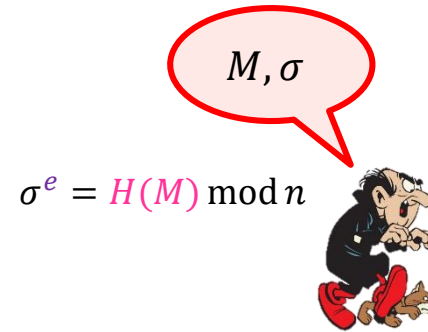
Hard to find!

A₂

1. Output (M, σ) = (M, σ₁ · σ₂)

$$(\sigma_1 \cdot \sigma_2)^e \stackrel{?}{=} H(M) \bmod n$$

$$(H(X)^d \cdot H(Y)^d)^e = \underbrace{H(X) \cdot H(Y)}_{\text{Hard to find!}} = H(M) \bmod n$$



Hashed-RSA – security

- Factoring + RSA-problem must be hard
- What are the requirements of H ?
 - Must be collision-resistant:

$$H(X) = H(Y) \Rightarrow H(X)^d = H(Y)^d = \sigma$$

- Is this enough?
 - Unknown
 - However, if H is a random oracle then

RSA. KeyGen

1. $p, q \stackrel{\$}{\leftarrow}$ two random prime numbers
2. $n \leftarrow p \cdot q$
3. $\phi(n) = (p - 1)(q - 1)$
4. **choose** e such that $\gcd(e, \phi(n)) = 1$
5. $d \leftarrow e^{-1} \pmod{\phi(n)}$
6. $sk \leftarrow (n, d)$ $pk \leftarrow (n, e)$
7. **return** (sk, pk)

Theorem: if the RSA problem is hard and H is a random oracle, then Hashed-RSA is UF-CMA secure

RSA. Vrfy($(n, e), M, \sigma$)

1. **if** $\sigma^e = H(M) \pmod N$ **then**
2. **return** 1
3. **else**
4. **return** 0

Discrete log based signatures

- Schnorr (see problem set 10 for details)
 - Elegant design
 - Has formal security proof (based on DLOG problem and H assumed perfect)
 - Was patented
 - One sharp edge: requires randomness during signing \Rightarrow reuse of randomness leaks private key
- (EC)DSA
 - Non-patented alternative
 - More complicated design than Schnorr
 - No security proof
 - Standardized by NIST (designed by NSA)
 - Very widely used
 - Same sharp edge as Schnorr
 - Broke all PlayStation 3's produced by Sony





Public-key infrastructure (PKI)

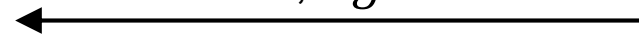
What are identities?



"Alice", g^a



"Bob", g^b



$$K \leftarrow g^{ab}$$

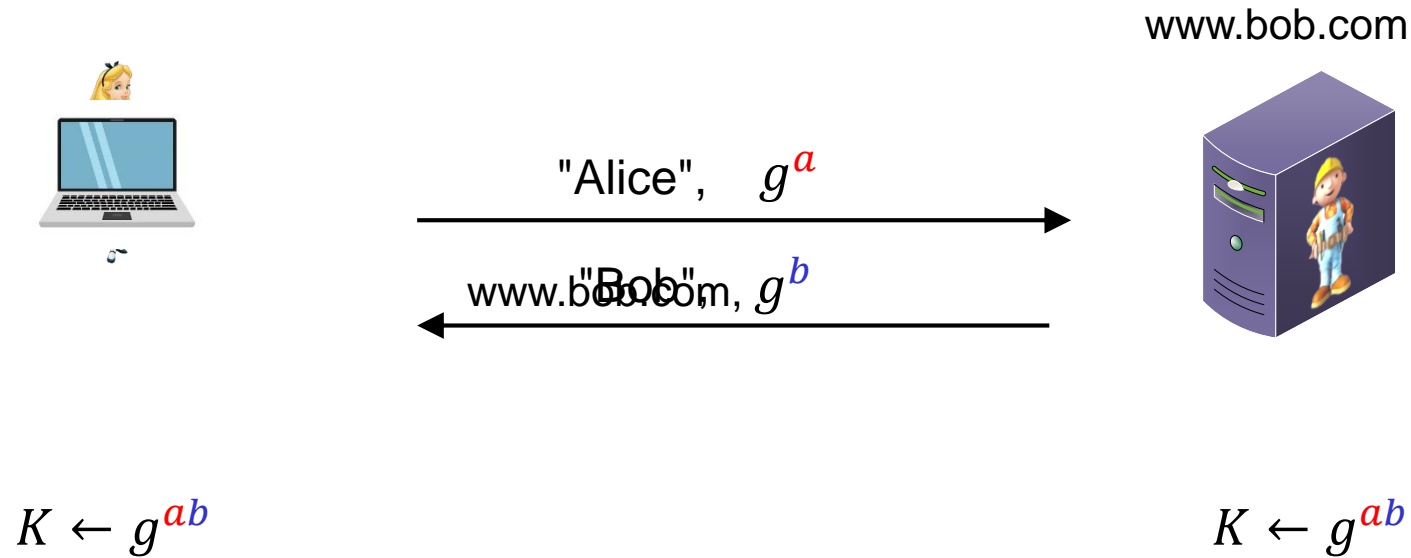
$$K \leftarrow g^{ab}$$

There are many Alice's and many Bob's

How do we know that g^a belongs to *this* particular Alice, and g^b to this particular Bob?

Need to **bind** public keys to entities

Identities on the internet

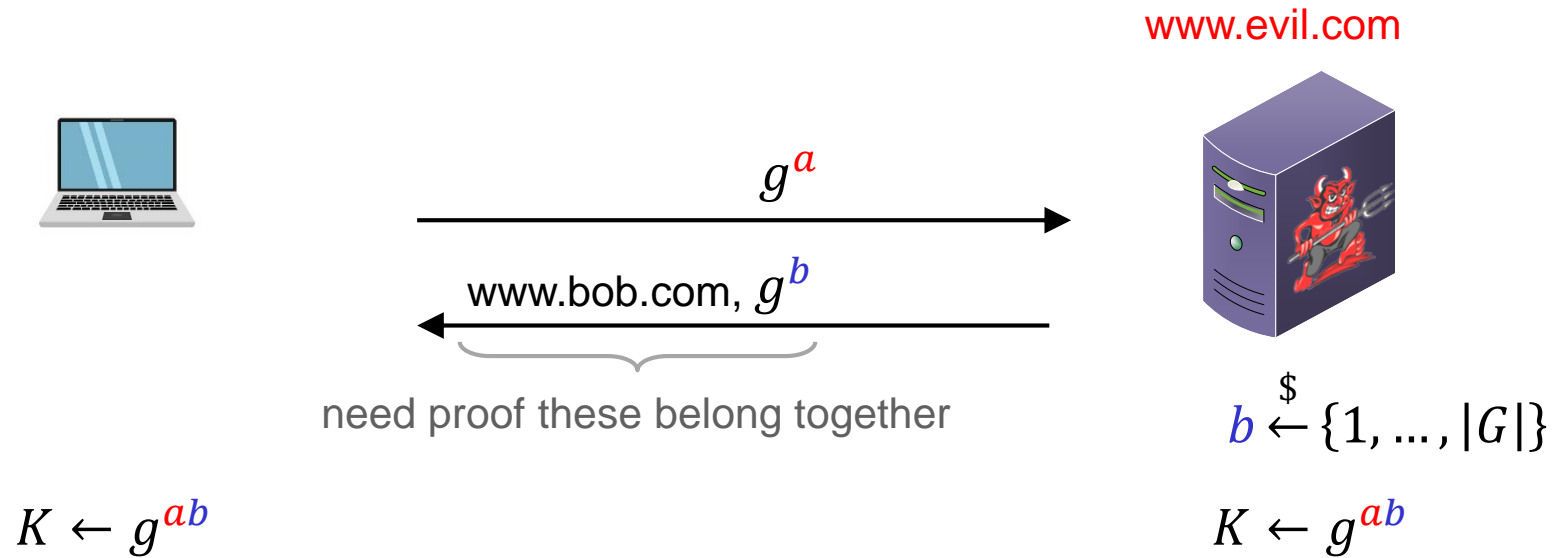


There are many Alice's and many Bob's

How do we know that g^a belongs to *this* particular Alice, and g^b to this particular Bob?

Need to **bind** public keys to entities – internet: bind public keys to **domain names**

Identities on the internet

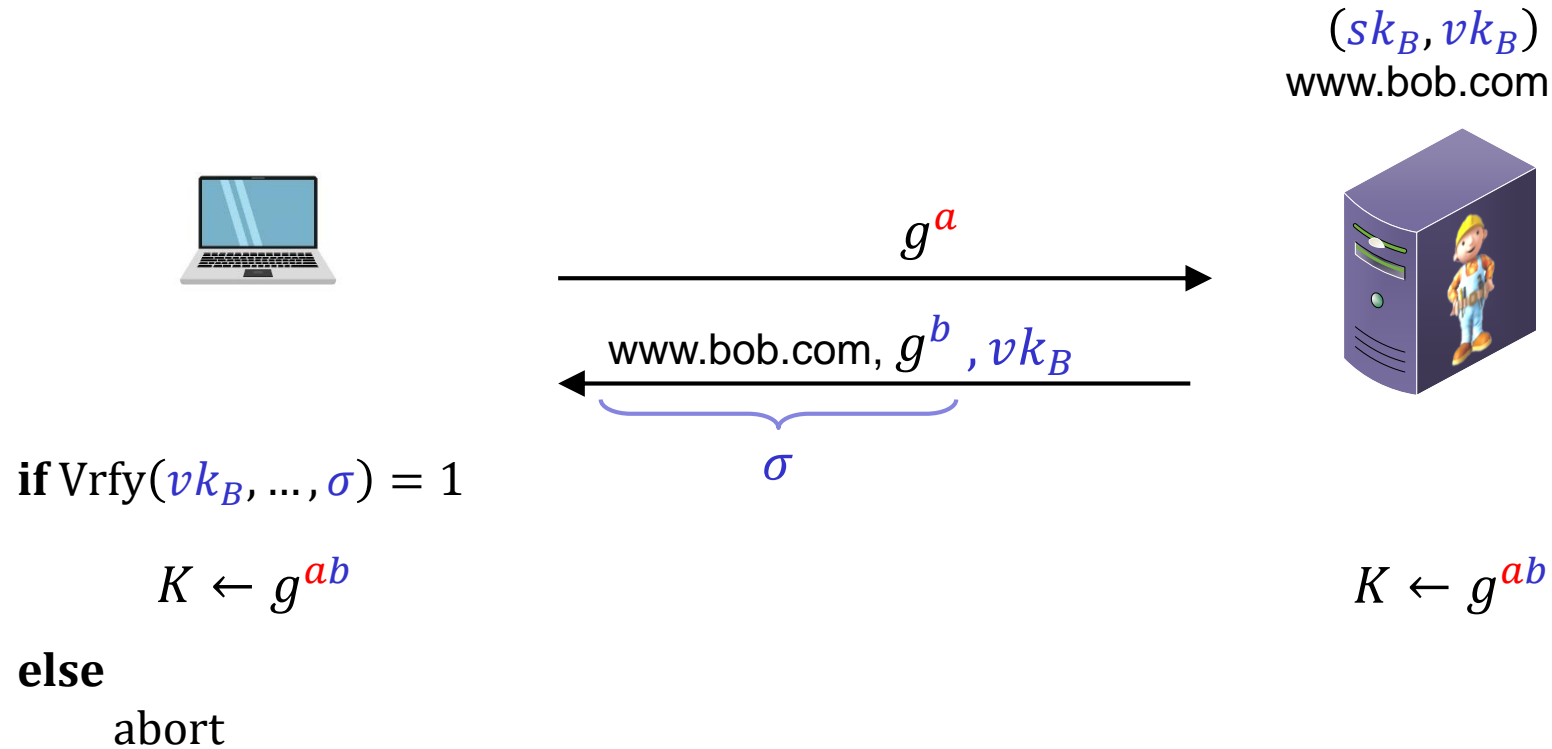


There are many Alice's and many Bob's

How do we know that g^a belongs to *this* particular Alice, and g^b to this particular Bob?

Need to **bind** public keys to entities – internet: bind public keys to **domain names**

Authenticated key exchange



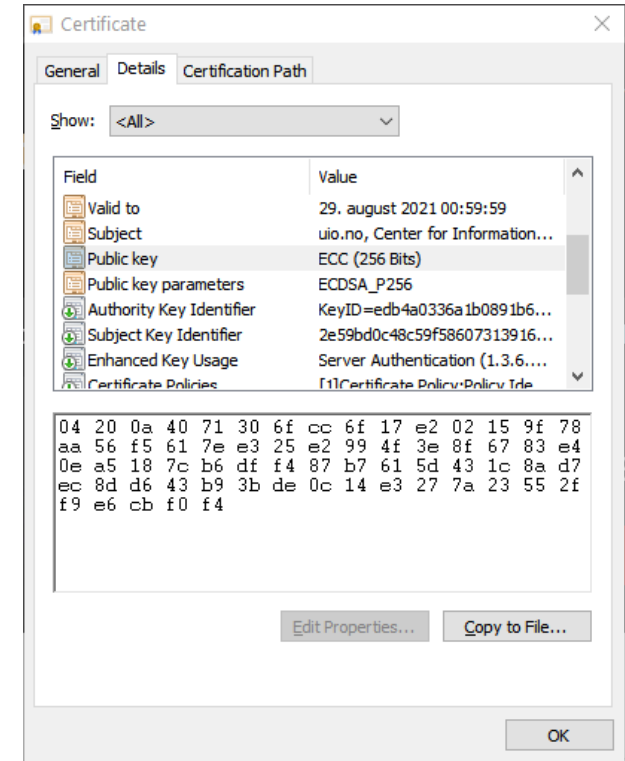
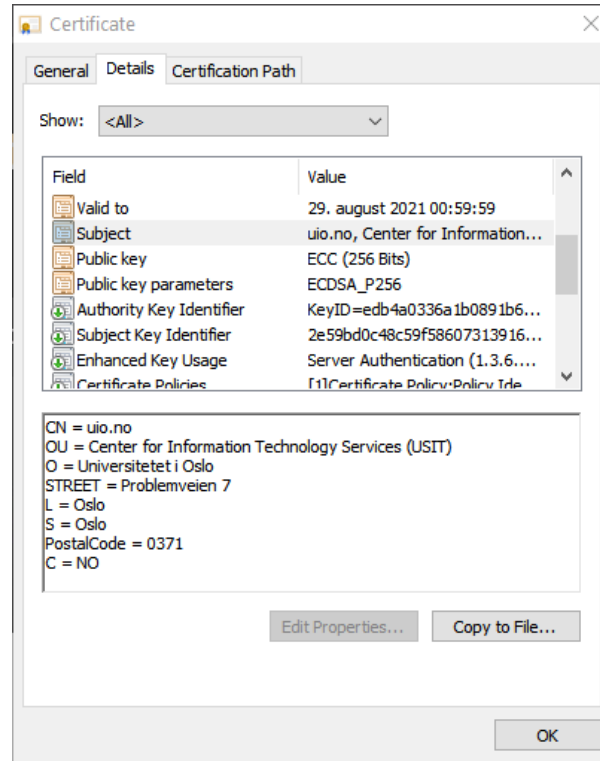
But why should we trust this vk_B ? Could have been created by the adversary itself

Digital certificates

- **Digital certificate:** a way of binding a public key to an entity
- A certificate consists of:
 - The public key of the entity
 - A bunch of information identifying the entity
 - Name
 - Address
 - Occupation
 - URL
 - Email-address
 - Phone number
 - ...
 - A *digital signature* on all the above by a **certificate authority (CA)**



Digital certificates



Certificate authorities (CA)

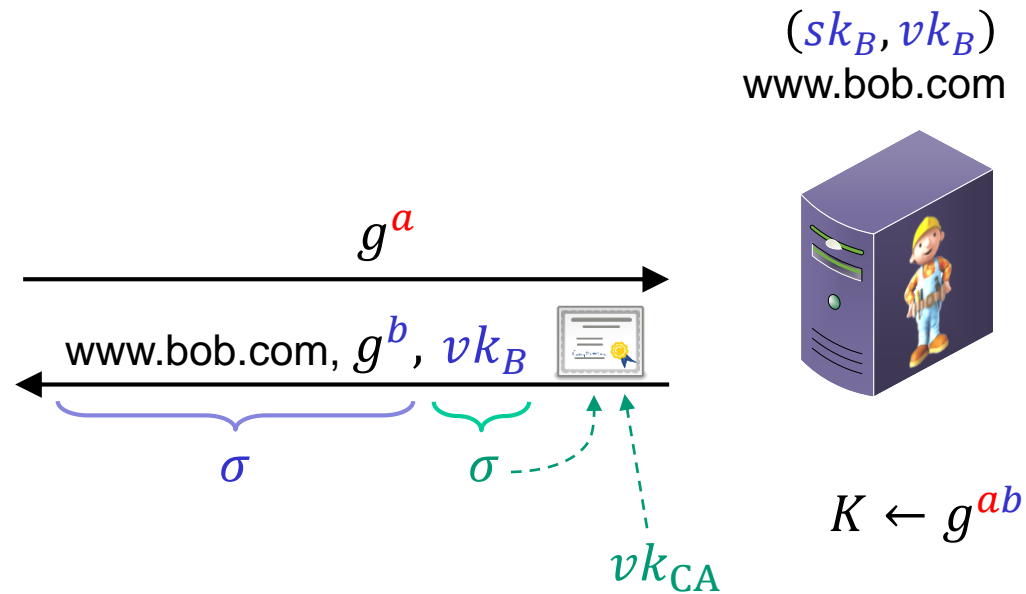
- **CA:** an issuer of digital certificates
- Acts as a trusted third-party, certifying (i.e., signing) the public keys of other entities
 - Verifies the identity of a claimed public-key owner
- The basis of a **public-key infrastructure (PKI)**



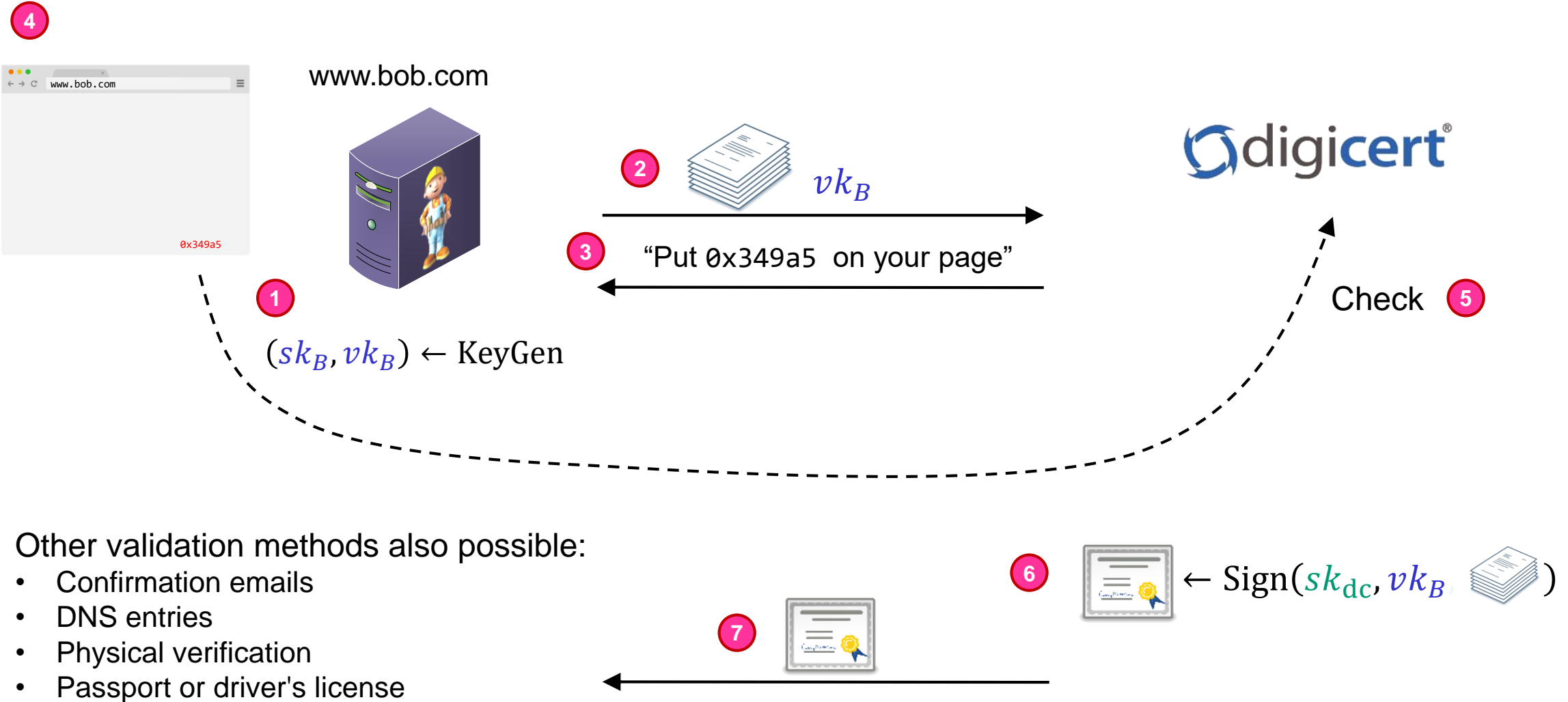
Authenticated key exchange + PKI



```
if Vrfy( $vk_B, \dots, \sigma$ ) = 1
  and Vrfy( $vk_{CA}, \dots, \sigma$ ) = 1
     $K \leftarrow g^{ab}$ 
else
  abort
```



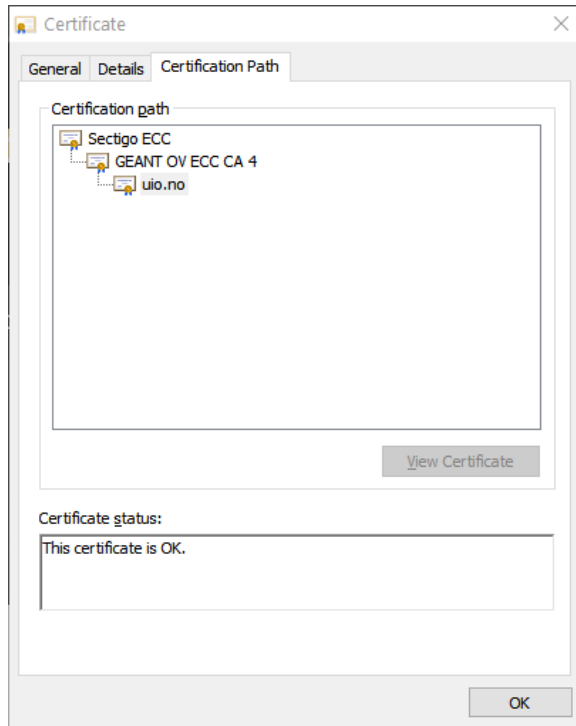
How to get a signed certificate?



Other validation methods also possible:

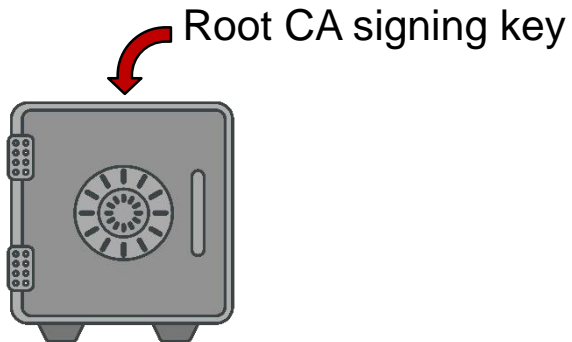
- Confirmation emails
- DNS entries
- Physical verification
- Passport or driver's license

Certificate chains

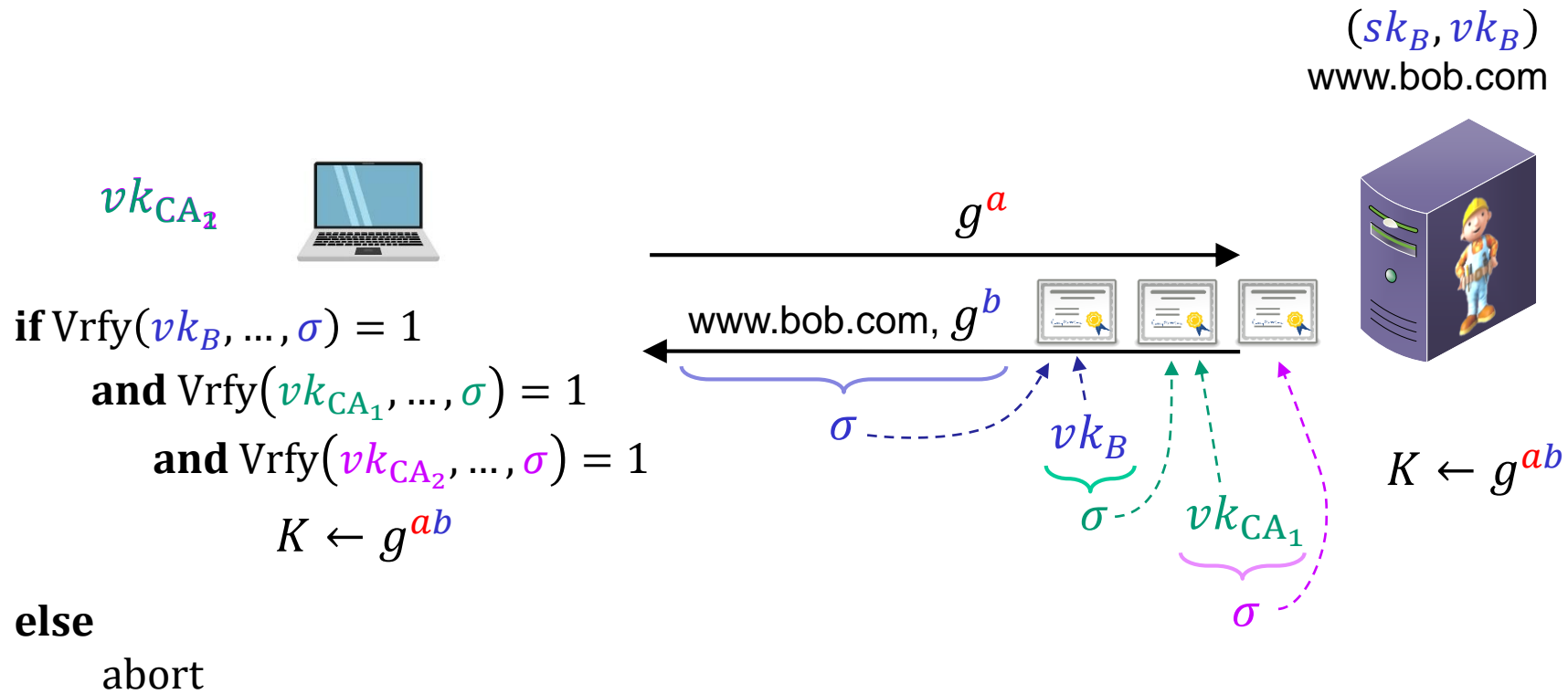


Root CAs

- **Root CAs:** CAs that sign other CAs' public keys
 - + only a few root CAs need to be trusted by end-users
 - + root CAs can distribute the signing + verification load to smaller CAs
 - single point of failure; private key must be *very heavily* guarded
- Root CAs for the internet: a few large multinational corporations



HTTPS / TLS + PKI



How to become an internet root CA?

- Need to prove yourself (trust)worthy to browser and OS vendors
 - [Microsoft Root Certificate Program](#)
 - [Mozilla CA Certificate Program](#)
 - [Apple Root Certificate Program](#)
 - [Chrome Root CA Program](#)
- Lot's of auditing and paperwork
- Many formal technical and non-technical security requirements
 - CA/Browser forum
 - [Baseline Requirements v1.7.3](#)

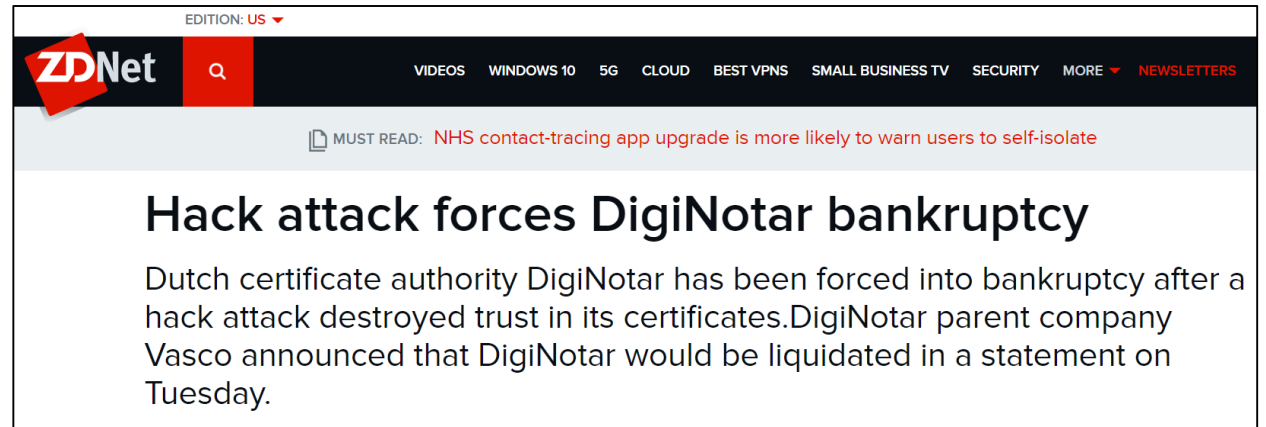


DigiNotar

- Dutch root CA
- Lost control of their private signing key in 2011
- Fraudulent certificates issued for Gmail, Yahoo!, Mozilla, WordPress, ...
- 30 000 Iranian Gmail users targeted



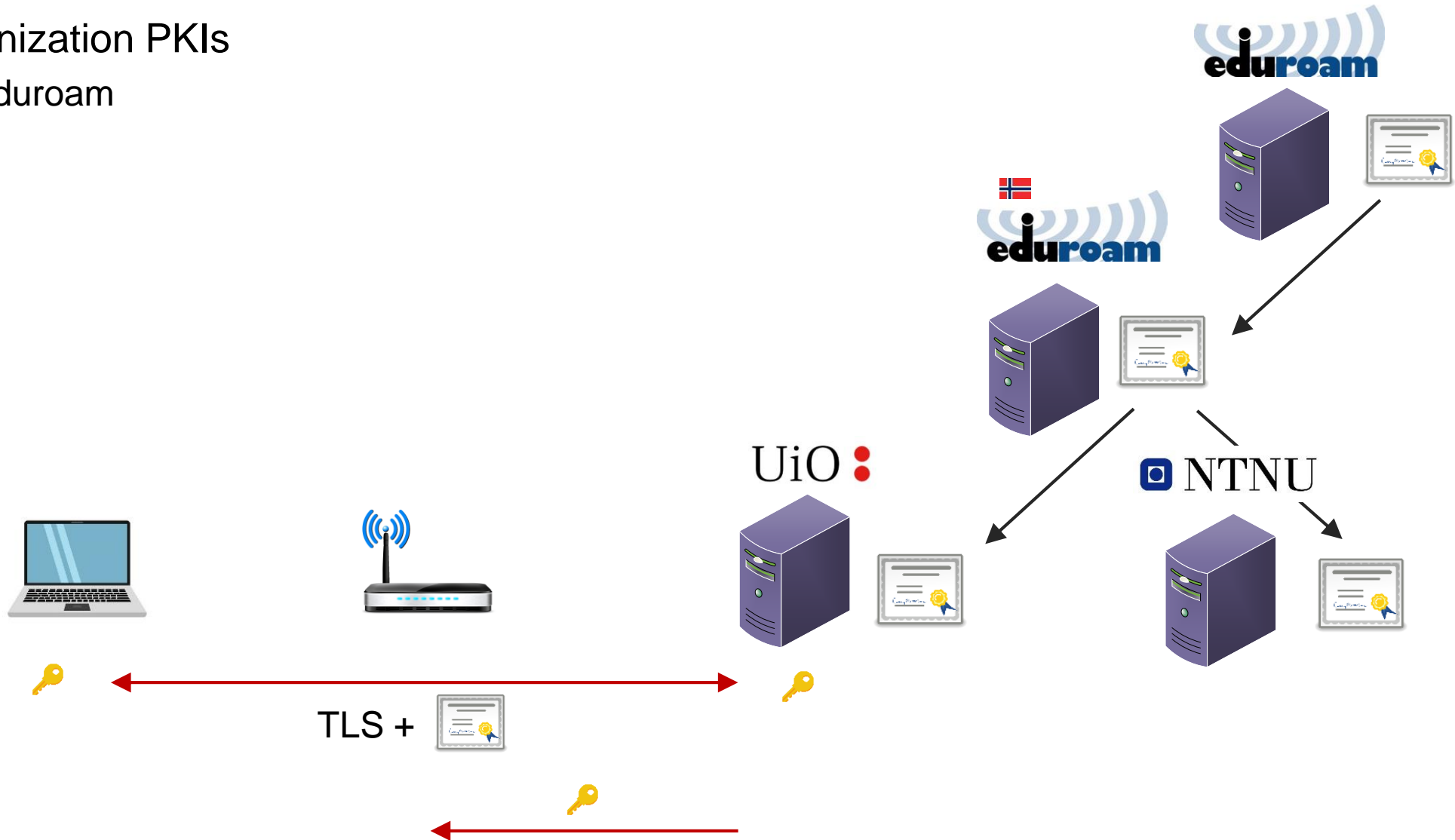
The screenshot shows the ITPro website header with navigation links for Business, Cloud, Hardware, Infrastructure, Security, Software, and Technology. The article title is "DigiNotar goes bankrupt after hack" and the sub-headline is "The Dutch CA goes into bankruptcy following the significant hacks claimed by ComodoHacker." The author is Tom Brewster, dated 20 Sep 2011. The image shows a close-up of a computer keyboard.

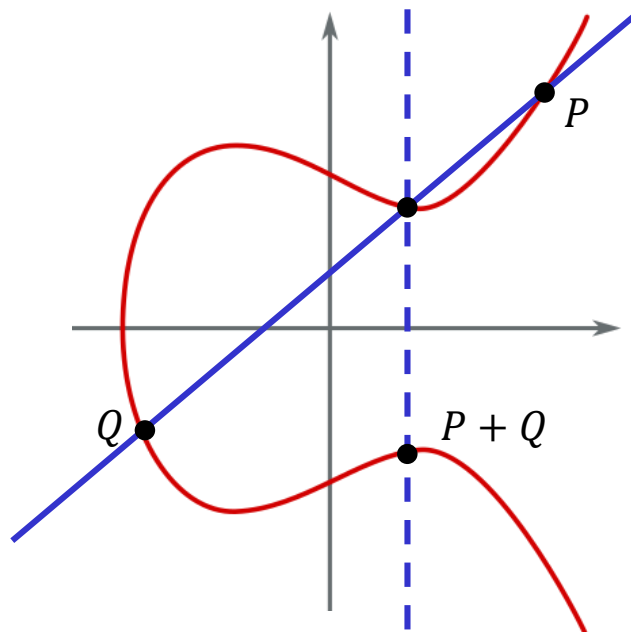


The screenshot shows the ZDNet website header with navigation links for VIDEOS, WINDOWS 10, 5G, CLOUD, BEST VPNS, SMALL BUSINESS TV, SECURITY, MORE, and NEWSLETTERS. The article title is "Hack attack forces DigiNotar bankruptcy" and the sub-headline is "Dutch certificate authority DigiNotar has been forced into bankruptcy after a hack attack destroyed trust in its certificates. DigiNotar parent company Vasco announced that DigiNotar would be liquidated in a statement on Tuesday." A "MUST READ" banner above the article title reads "NHS contact-tracing app upgrade is more likely to warn users to self-isolate".

Other PKIs exist

- Organization PKIs
 - eduroam





$$y^2 = x^3 + ax + b$$

$$a, b, x, y \in \mathbf{R}$$

End of Part II
(Asymmetric crypto)

Summary of asymmetric cryptography

Primitive	Functionality + syntax	Hardness assumption / security goal	Acronym	Examples
Diffie-Hellman	Derive shared value (key) in a cyclic group $A^b = g^{ab} = B^a$	Discrete logarithm (DLOG) Diffie-Hellman (DH)	DH	(\mathbf{Z}_p^*, \cdot) –DH $(E(\mathbf{F}_p), +)$ –DH
RSA function	One-way trapdoor permutation	Factoring problem RSA-problem		Textbook RSA
Public-key encryption	Encrypt variable-length input $\text{Enc} : \mathcal{PK} \times \mathcal{M} \rightarrow \mathcal{C}$	Confidentiality: attacker should learn nothing about plaintext (except length) from ciphertexts	IND-CPA IND-CCA	ElGamal Hashed/Padded RSA Fujisaki-Okamoto-transform
Digital signatures	Create signature on variable length input $\text{Sign} : \mathcal{SK} \times \mathcal{M} \rightarrow \mathcal{S}$ $\text{Vrfy} : \mathcal{VK} \times \mathcal{M} \times \mathcal{S} \rightarrow \{1,0\}$	Integrity: attacker shouldn't be able to forge messages, i.e., create new messages with valid signatures	UF-CMA	Schnorr Hashed-RSA ECDSA

Cryptographic groups	Comment	Computational problem	Best-known attack	Common sizes
(\mathbf{Z}_p^*, \cdot)	p prime $ \mathbf{Z}_p^* = p - 1$	Discrete logarithm	General number field sieve (GNFS)	$ p \approx 2000\text{--}3000$ bits
Subgroups $H < (\mathbf{Z}_p^*, \cdot)$	$ H = q$ (typically prime)	Discrete logarithm	GNFS	$ q \approx 256$ bits
$(E(\mathbf{F}_p), +)$	p prime $ E(\mathbf{F}_p) = q$ (typically) prime $p \neq q$	Discrete logarithm	Generic attacks: Baby-step giant-step, Pollard-rho, Pohlig-Hellman	$ E(\mathbf{F}_p) \approx 256$ bits $ p \approx 256$ bits
(\mathbf{Z}_n^*, \cdot)	n not prime $ \mathbf{Z}_n^* = \phi(n)$	Factoring	GNFS	$ n \approx 2000\text{--}4000$ bits

Next week

- Quantum computers

