Lecture 11 – Digital signatures, UF-CMA, Schnorr, PKI

TEK4500

29.10.2025

Håkon Jacobsen

hakon.jacobsen@its.uio.no

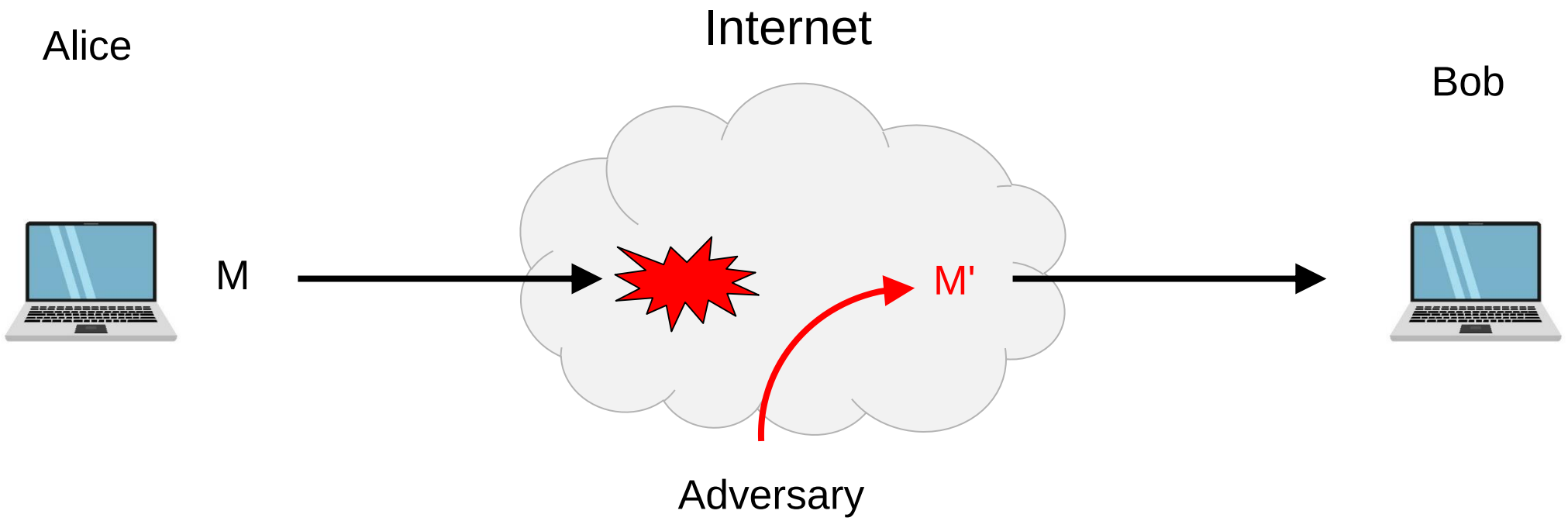
Basic goals of cryptography

	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)

Key Encapsulation Mechanisms (KEM)

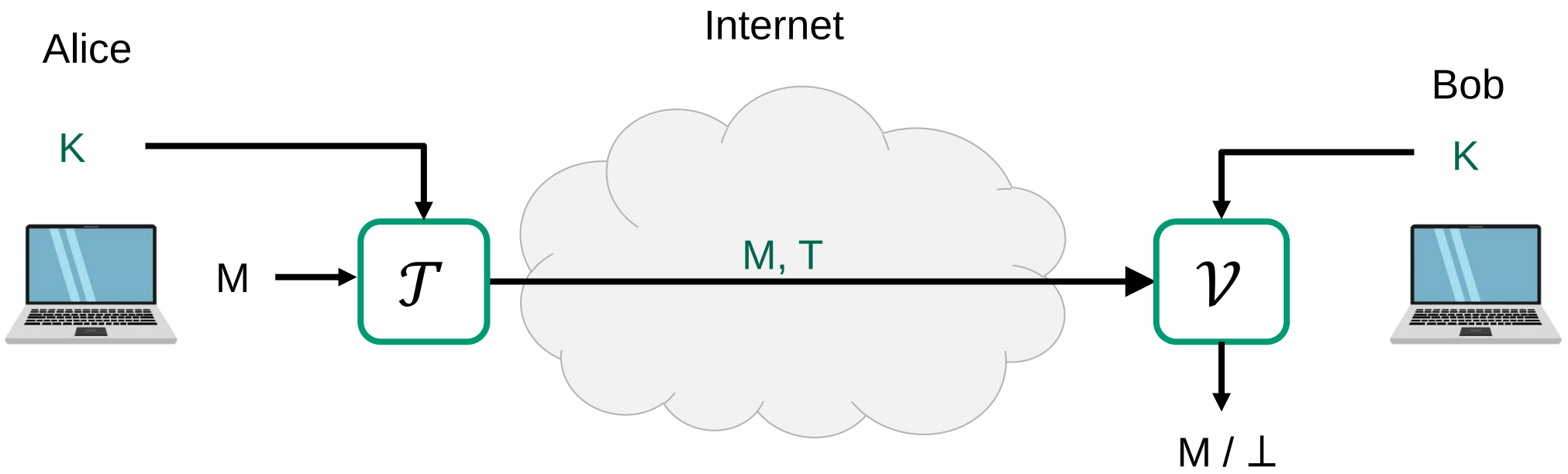
What is cryptography?



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

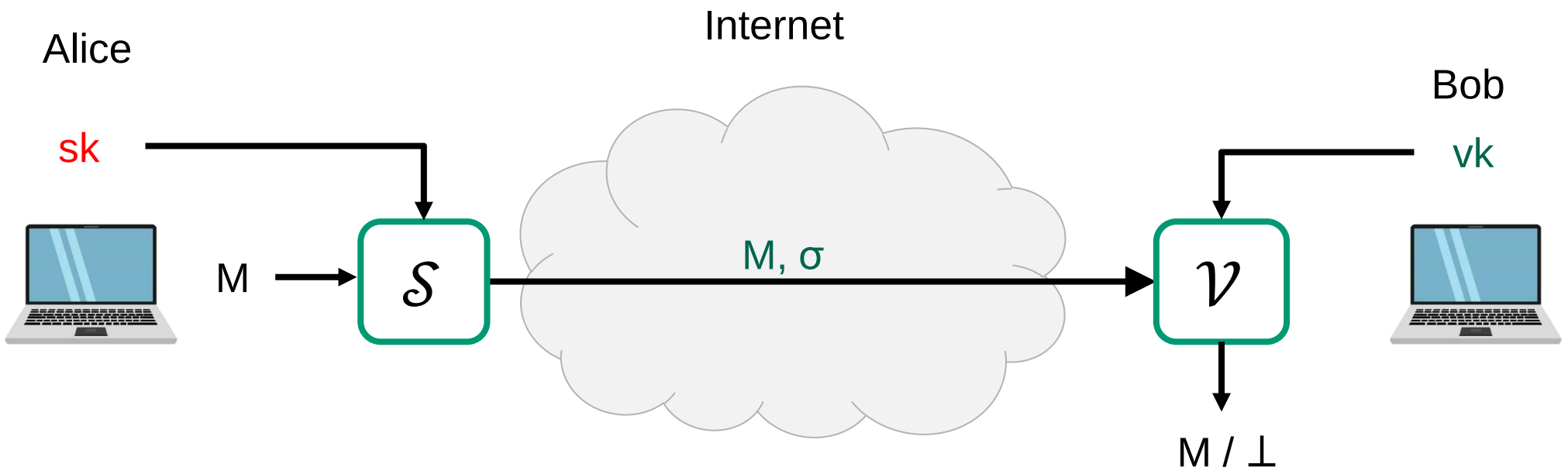


$\mathcal{T}$: tagging algorithm (public)

K : tagging / verification key (secret)

$\mathcal{V}$: verification algorithm (public)

Digital signatures



$\mathcal{S}$: signing algorithm (public)

$\mathcal{V}$: verification algorithm (public)

sk : signing key (secret)

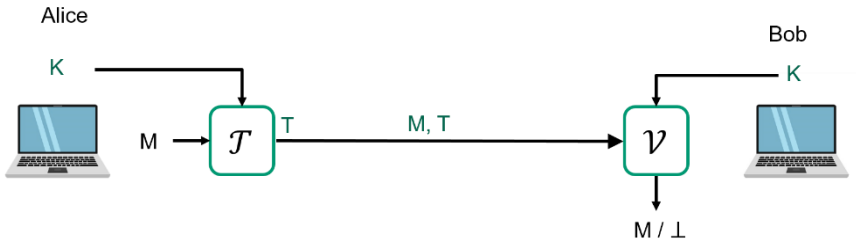
vk : verification key (public)

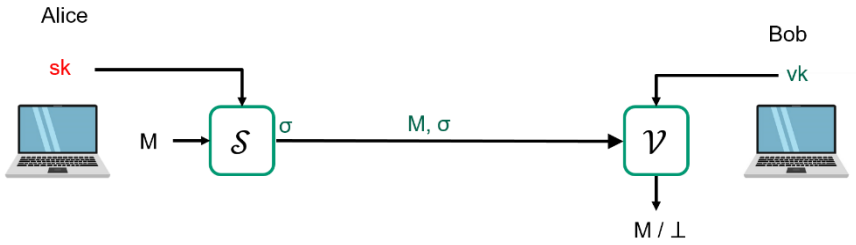
MACs vs. digital signatures

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

- Digital signatures can be verified by *anyone*

MAC



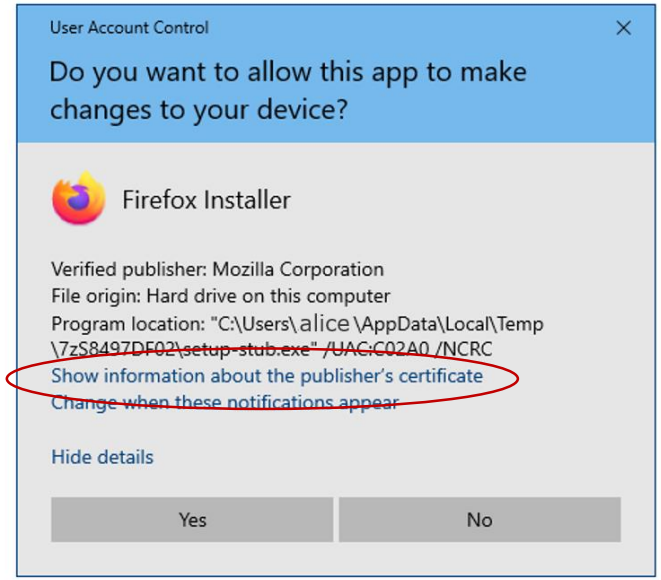


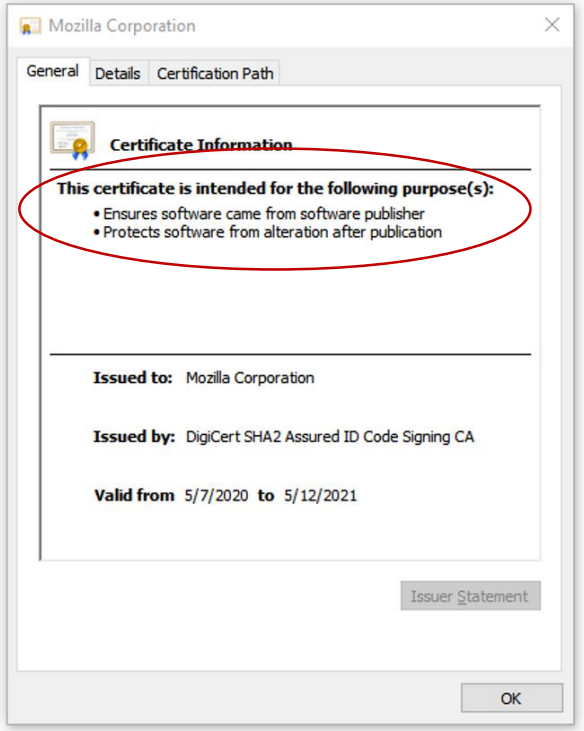
Digital signature

- **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 $\Sigma = (\text{KeyGen}, \text{Sign}, \text{Vrfy})$

$\text{KeyGen} : \mathcal{SK} \times \mathcal{VK}$

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

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

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

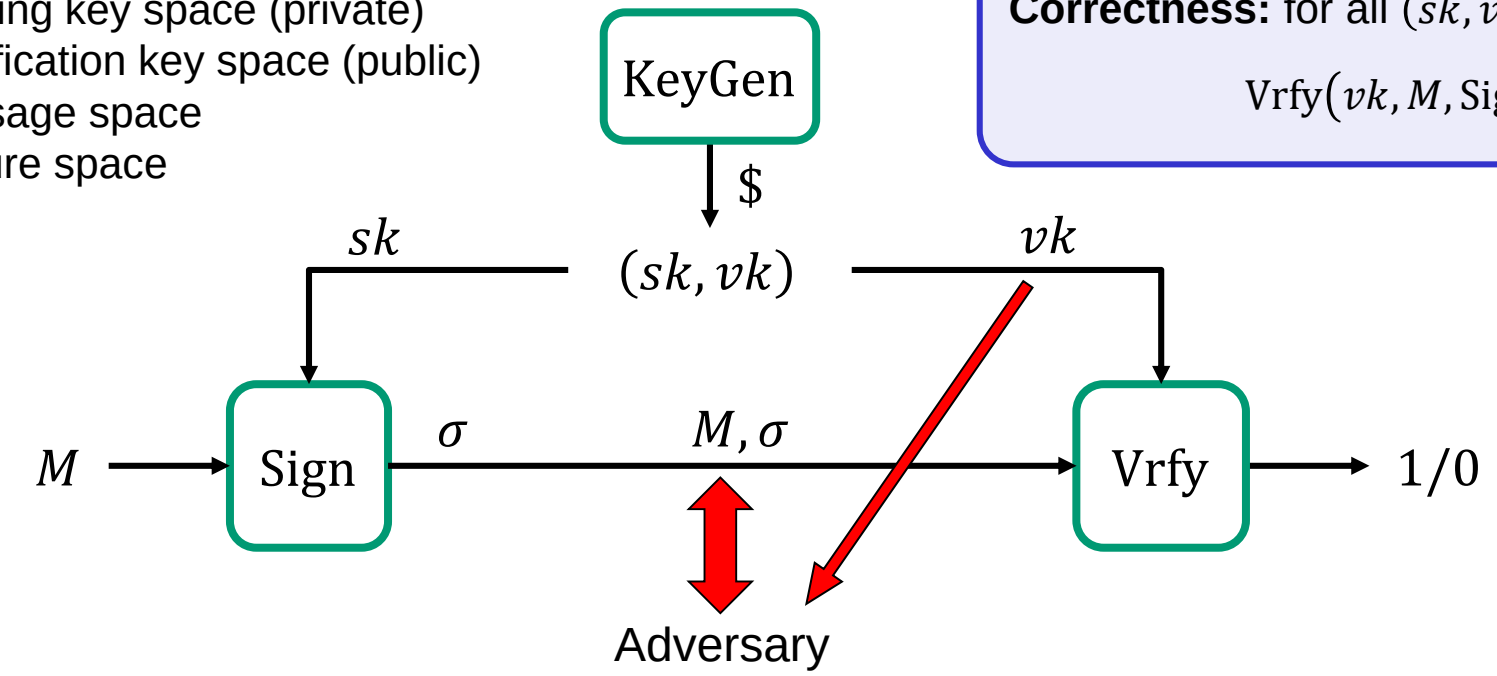
$\text{Vrfy} : \mathcal{VK} \times \mathcal{M} \times \mathcal{S} \to \{0,1\}$

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

- $\mathcal{SK}$ – signing key space (private)
- $\mathcal{VK}$ – verification key space (public)
- $\mathcal{M}$ – message space
- $\mathcal{S}$ –signature space

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

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



8

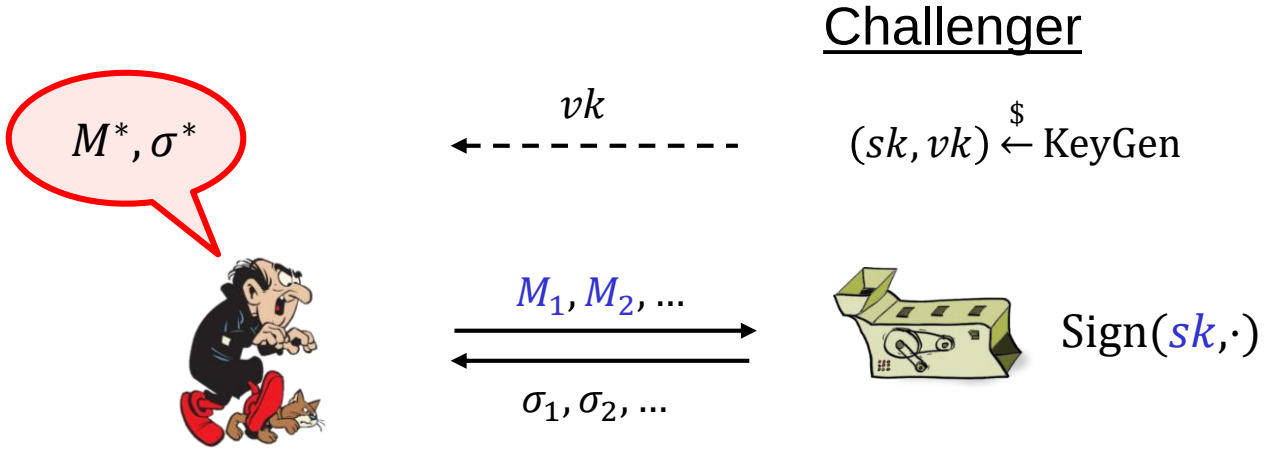
Digital signatures – security: UF-CMA

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

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

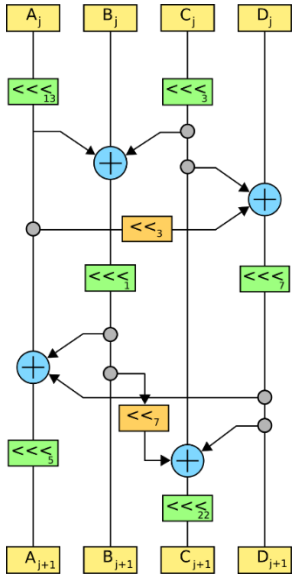


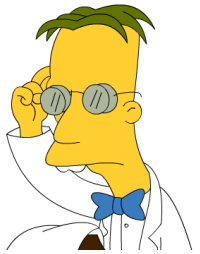
A has **forged** a signature if

- σ^* is a valid signature for M^*
- $M^* \notin \{M_1, M_2, \ldots\}$

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

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





Designing signature schemes

DLOG proof of knowledge

$A = g^a$ $(G,\star) = \langle g \rangle$



a



Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✓
3. No one (including Bob) learns anything about a from the interaction ✗

DLOG proof of knowledge

$A = g^a$ $(\boldsymbol{G}, \star) = \langle g \rangle$



$r \xleftarrow{\$} \{1, \dots, |\boldsymbol{G}|\}$

$a + r \pmod{|\boldsymbol{G}|}$



Proof requirements

1. If Alice knows a she can get Bob to accept ✘
2. If Alice doesn't know a she can't get Bob to accept ?
3. No one (including Bob) learns anything about a from the interaction ✓

DLOG proof of knowledge

$A = g^a$ $(\boldsymbol{G}, \star) = \langle g \rangle$





r

$r \xleftarrow{\$} \{1, \dots, |\boldsymbol{G}|\}$

$a + r \pmod{|\boldsymbol{G}|}$

Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✓
3. No one (including Bob) learns anything about a from the interaction ✗

DLOG proof of knowledge

$A = g^a$ $\quad (\boldsymbol{G}, \star) = \langle g \rangle$



$r \xleftarrow{\$} \{1, \dots, |\boldsymbol{G}|\}$

$R = g^r$

$a + r \pmod{|\boldsymbol{G}|}$

z



accept if $A \star R \stackrel{?}{=} g^z$

Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✗
3. No one (including Bob) learns anything about a from the interaction ✓

$A \star R = g^a g^r = g^{a+r} = g^z$

$R \leftarrow A^{-1} \star g^z \implies A \star R = A \star A^{-1} \star g^z = g^z$

DLOG proof of knowledge

$A = g^a$ $(\boldsymbol{G}, \star) = \langle g \rangle$





$R = g^r$

b

$r \xleftarrow{\$} \{1, \dots, |\boldsymbol{G}|\}$ $b \xleftarrow{\$} \{1, \dots, |\boldsymbol{G}|\}$

$ab + r \pmod{|\boldsymbol{G}|}$

z

accept if $A^b \star R \stackrel{?}{=} g^z$

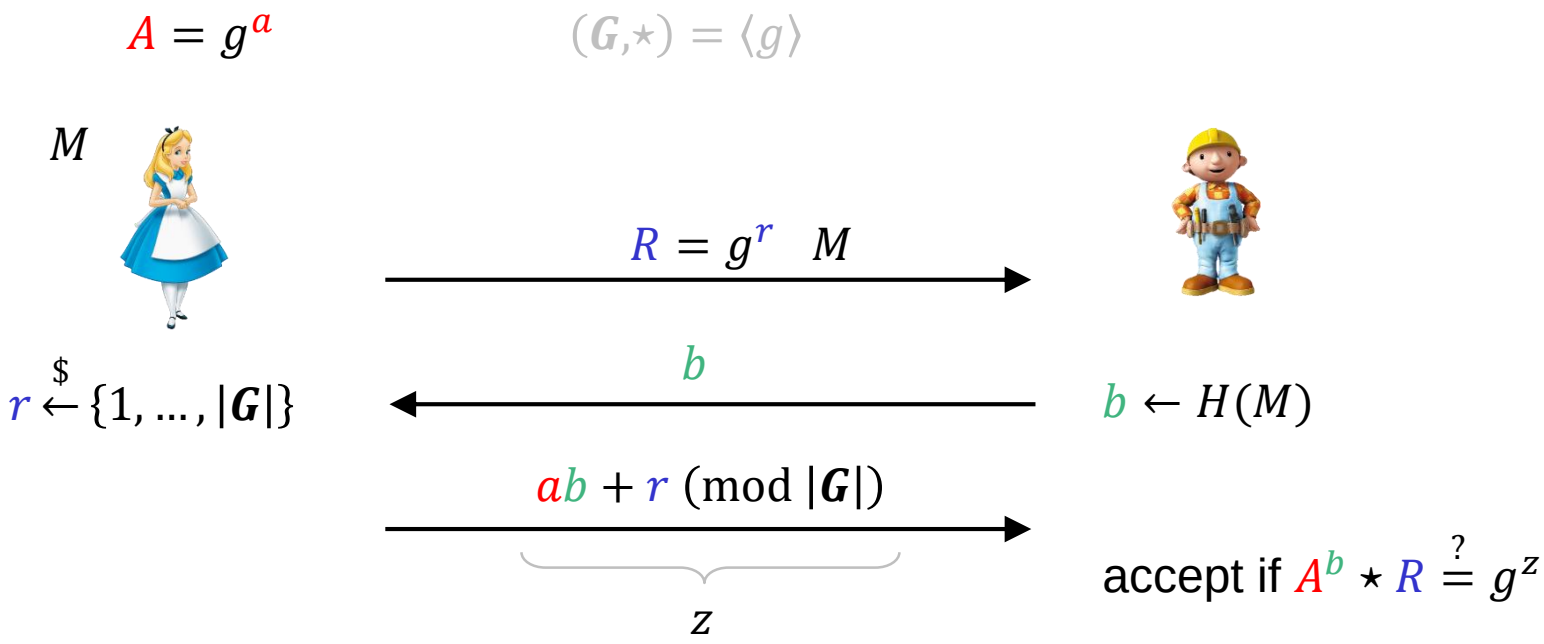
Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✓
3. No one (including Bob) learns anything about a from the interaction ✓

$A^b \star R = g^{ab} g^r = g^{ab+r} = g^z$

$R \leftarrow A^{-1} \star g^z \implies A^b \star R = A^b \star A^{-1} \star g^z \neq g^z$

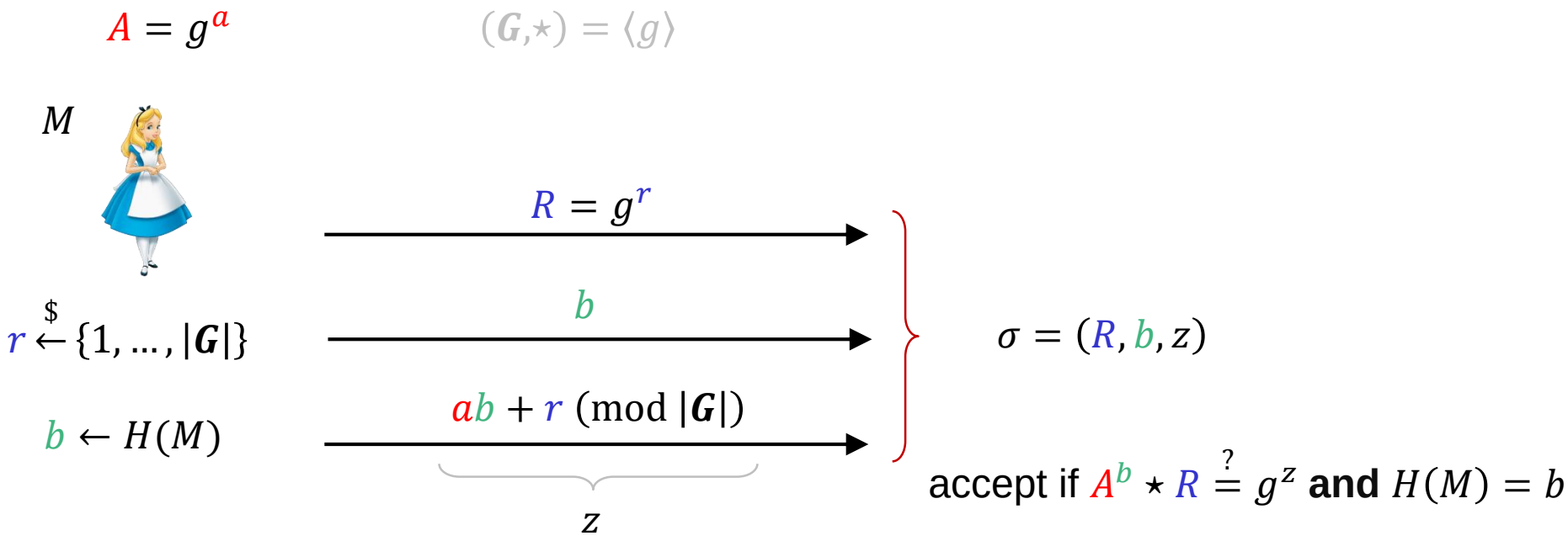
Interactive signature scheme



Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✓
3. No one (including Bob) learns anything about a from the interaction ✓

Non-interactive signature scheme



Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✓
3. No one (including Bob) learns anything about a from the interaction ✓

Non-interactive signature scheme

$A = g^a$ $(G,\star) = \langle g \rangle$

M



$r \xleftarrow{\$} \{1, \dots, |G|\}$

$b \leftarrow H(M)$

$R = A^{-b} \star g^z$

b

z

z

$\sigma = (R, b, z)$

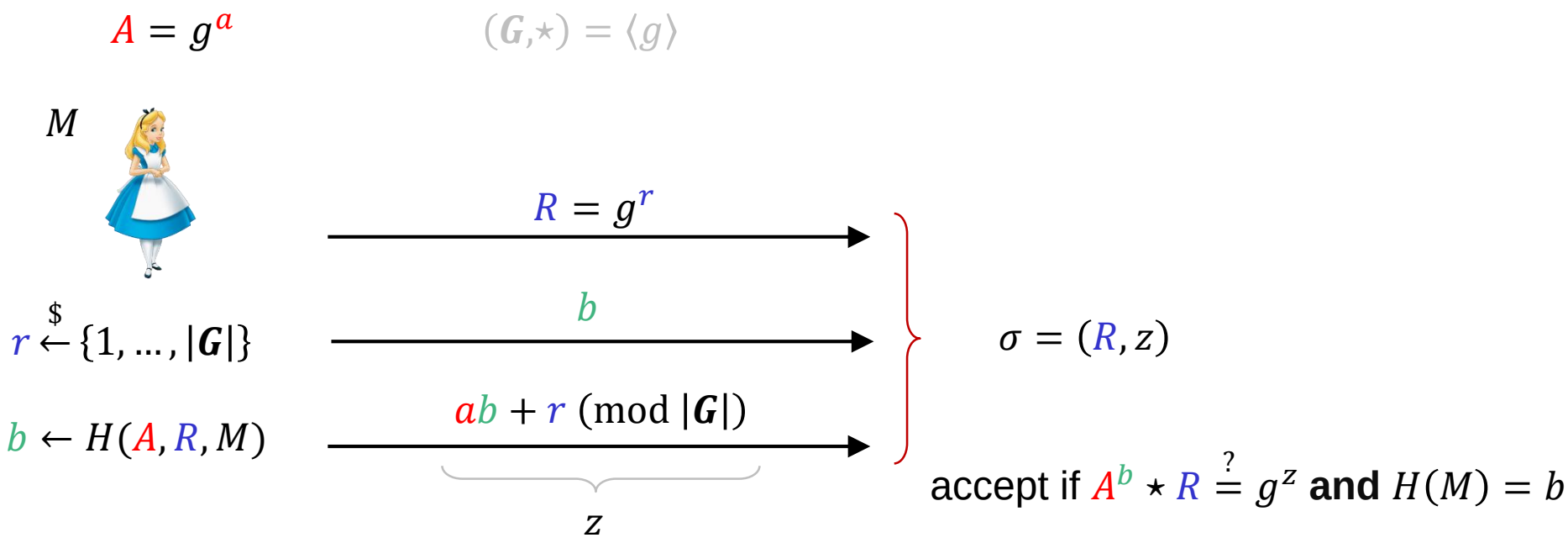
accept if $A^b \star R \stackrel{?}{=} g^z$ **and** $H(M) = b$

Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✘
3. No one (including Bob) learns anything about a from the interaction ✓

$A^b \star R = g^{ab} g^{-ab} \star g^z = g^z$

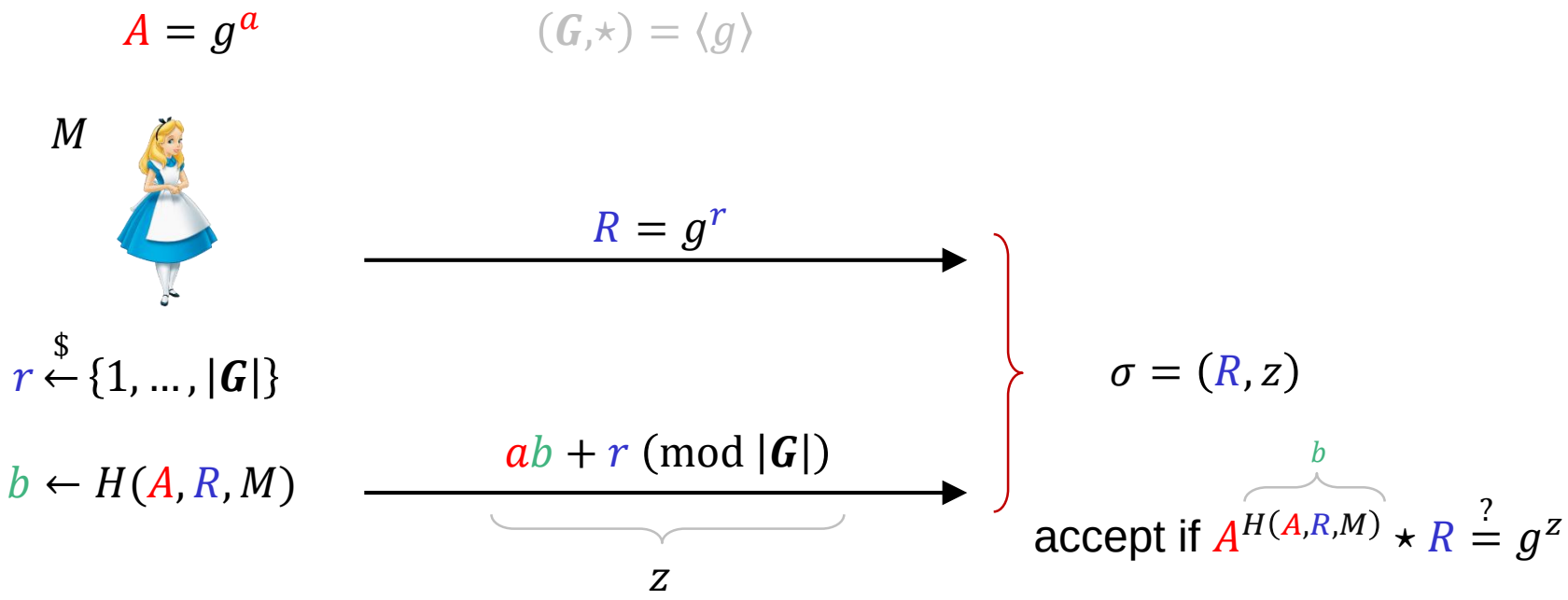
Non-interactive signature scheme



Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✓
3. No one (including Bob) learns anything about a from the interaction ✓

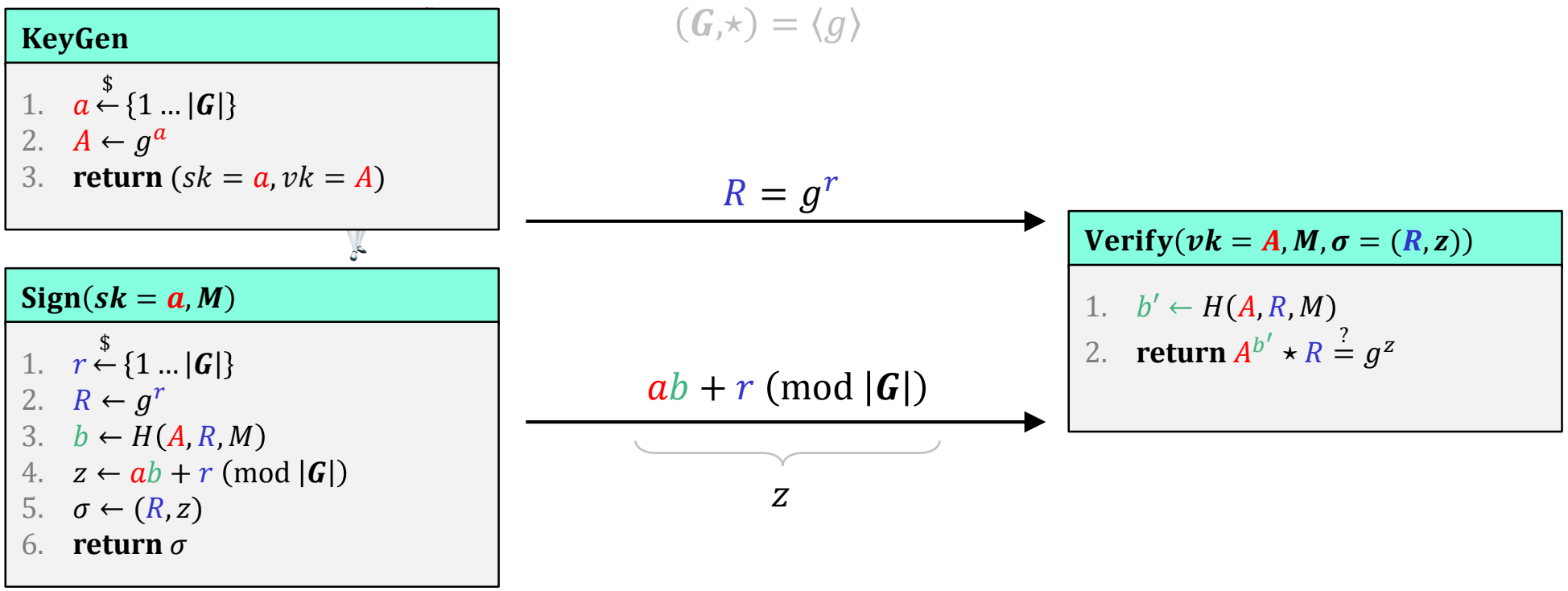
Non-interactive signature scheme



Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✓
3. No one (including Bob) learns anything about a from the interaction ✓

Schnorr signature scheme



Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✓
3. No one (including Bob) learns anything about a from the interaction ✓

Correctness:

$$A^{H(A,R,M)} \star R = g^{ab} \star g^r = g^{ab+r} = g^z$$

Schnorr signature scheme

$(\boldsymbol{G},\star) = \langle g \rangle$

KeyGen

1. $a \xleftarrow{\$} \{1 \dots |\boldsymbol{G}|\}$
2. $A \leftarrow g^a$
3. **return** $(sk = a, vk = A)$

Sign$(\boldsymbol{sk} = \boldsymbol{a}, \boldsymbol{M})$

1. $r \xleftarrow{\$} \{1 \dots |\boldsymbol{G}|\}$
2. $R \leftarrow g^r$
3. $b \leftarrow H(A, R, M)$
4. $z \leftarrow ab + r \pmod{|\boldsymbol{G}|}$
5. $\sigma \leftarrow (R, z)$
6. **return** σ

$R = g^r$

$ab + r \pmod{|\boldsymbol{G}|}$

z

Verify$(\boldsymbol{vk} = \boldsymbol{A}, \boldsymbol{M}, \boldsymbol{\sigma} = (\boldsymbol{R}, \boldsymbol{z}))$

1. $b' \leftarrow H(A, R, M)$
2. **return** $A^{b'} \star R \stackrel{?}{=} g^z$

r must be picked new *every time*!

Proof requirements

1. If Alice knows a she can get Bob to accept ✓
2. If Alice doesn't know a she can't get Bob to accept ✓
3. No one (including Bob) learns anything about a from the interaction ✓

$\sigma = (R, z)$

$\sigma' = (R, z')$

$\Rightarrow \quad z - z' = (ab + r) - (ab' + r) = a \cdot (b - b') \quad \Rightarrow \quad a = (z - z') \cdot (b - b')^{-1} \bmod |\boldsymbol{G}|$

Security: The Schnorr signature is UF-CMA secure if:

- The DLOG problem is hard in $(\boldsymbol{G},\star)$
- The hash function H is modelled as a *random oracle*

Learned private long-term key!

Discrete-log-based signatures

- Schnorr
  - Elegant design
  - Has formal security proof (based on DLOG problem and H assumed perfect)
  - Was patented

- (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
    - Reuse of r leaks long-term signing key
    - Broke all PlayStation 3's produced by Sony


CNBC

MARKETS BUSINESS INVESTING TECH POLITICS CNBC TV WATCHLIST

VIDEO GAMES

Sony: PlayStation Breach Involves 70 Million Subscribers

Chris Morris | @MorrisatLarge

Published 4:30 PM ET Tue, 26 April 2011 | Updated 5:26 PM ET Tue, 26 April 2011



Public-key infrastructure (PKI)

What are identities?



"Alice", g^a

"Bob", g^b



$K \leftarrow g^{ab}$

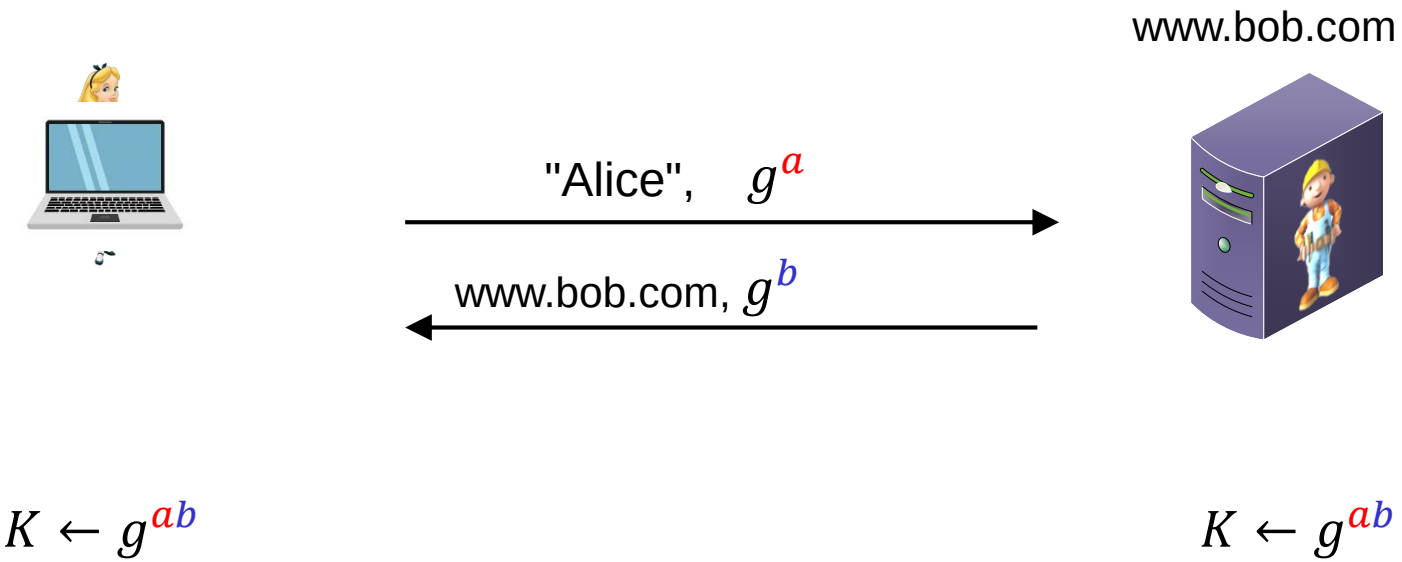
$K \leftarrow g^{ab}$

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

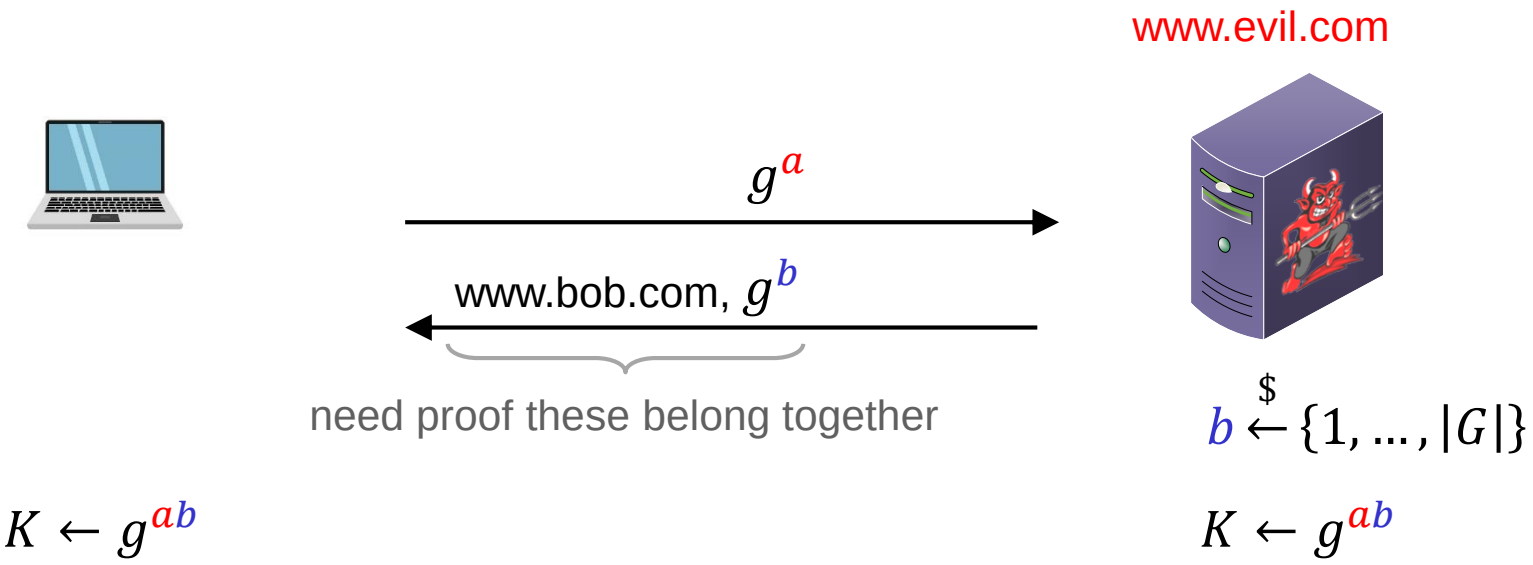


There are many Alice's and many Bob's

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

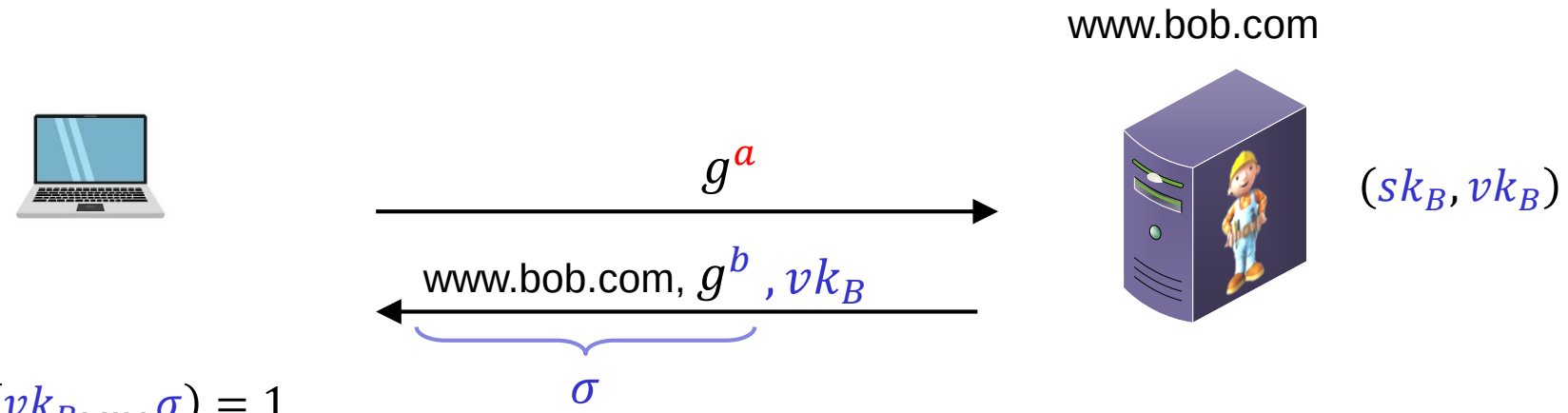


There are many Alice's and many Bob's

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



if $\text{Vrfy}(vk_B, \ldots, \sigma) = 1$

$K \leftarrow g^{ab}$

else

abort

$K \leftarrow g^{ab}$

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

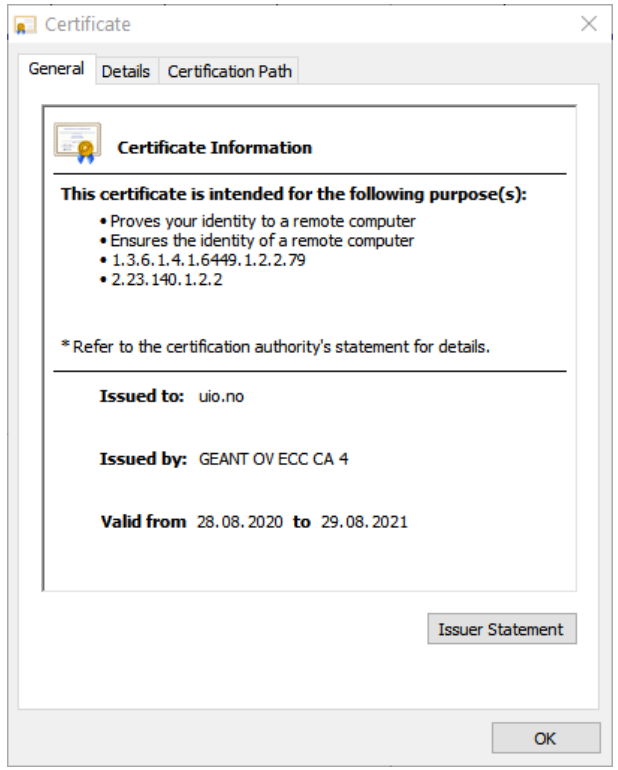
Digital certificates

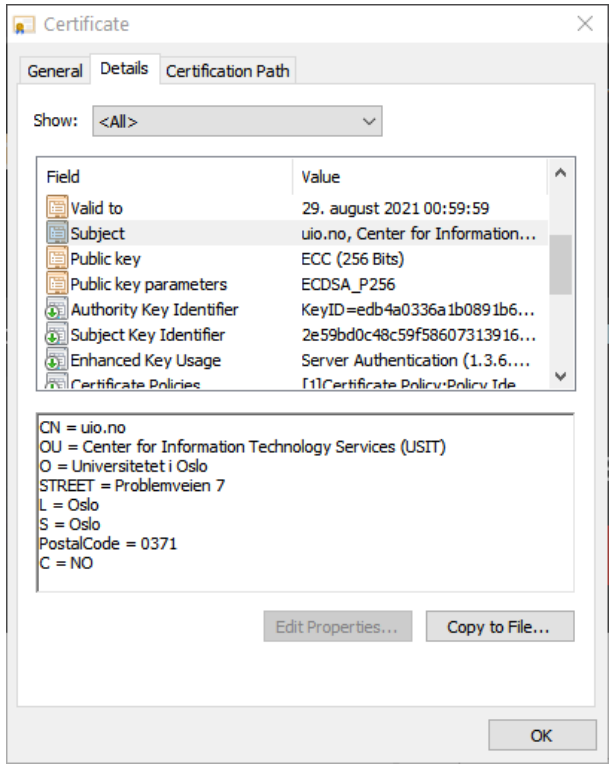
- **Digital certificate:** a way of binding a public key to an entity

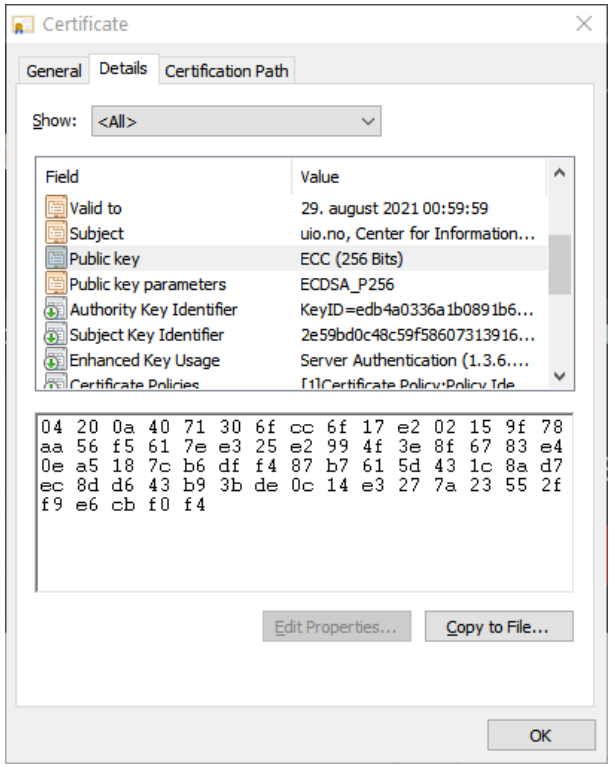


- A certificate consists of:
  - The public key of the entity
  - A bunch of information identifying the entity
    - Name
    - Address
    - Occupation
    - URL
    - Email-address
    - Phone number
    - …
  - A *digital signature* on all the above by a **certificate authority (CA)**

Digital certificates







Certificate authorities (CA)

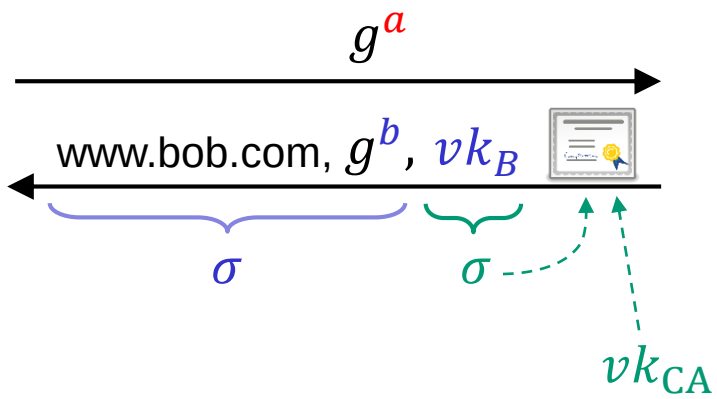
- **CA:** an issuer of digital certificates
- Acts as a trusted third-party, certifying (i.e., signing) the public keys of other entities
  - Verifies the identity of a claimed public-key owner
- The basis of a **public-key infrastructure (PKI)**



Authenticated key exchange + PKI



vk_{CA}



www.bob.com



(sk_B, vk_B)

if $\mathrm{Vrfy}(vk_B, \ldots, \sigma) = 1$

and $\mathrm{Vrfy}(vk_{\mathrm{CA}}, \ldots, \sigma) = 1$

$K \leftarrow g^{ab}$

else

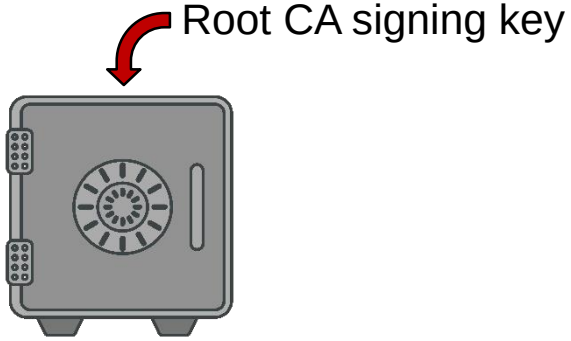
abort

$K \leftarrow g^{ab}$

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



COMODO

IdenTrust part of HID Global

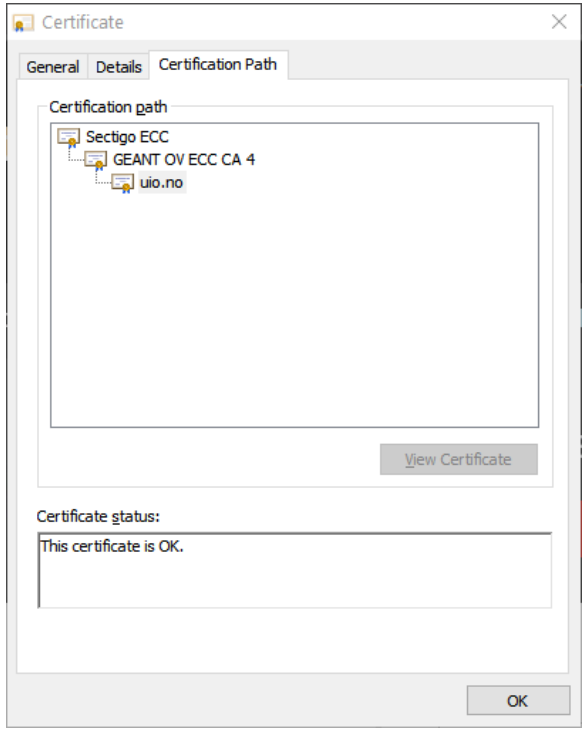
digicert

GoDaddy

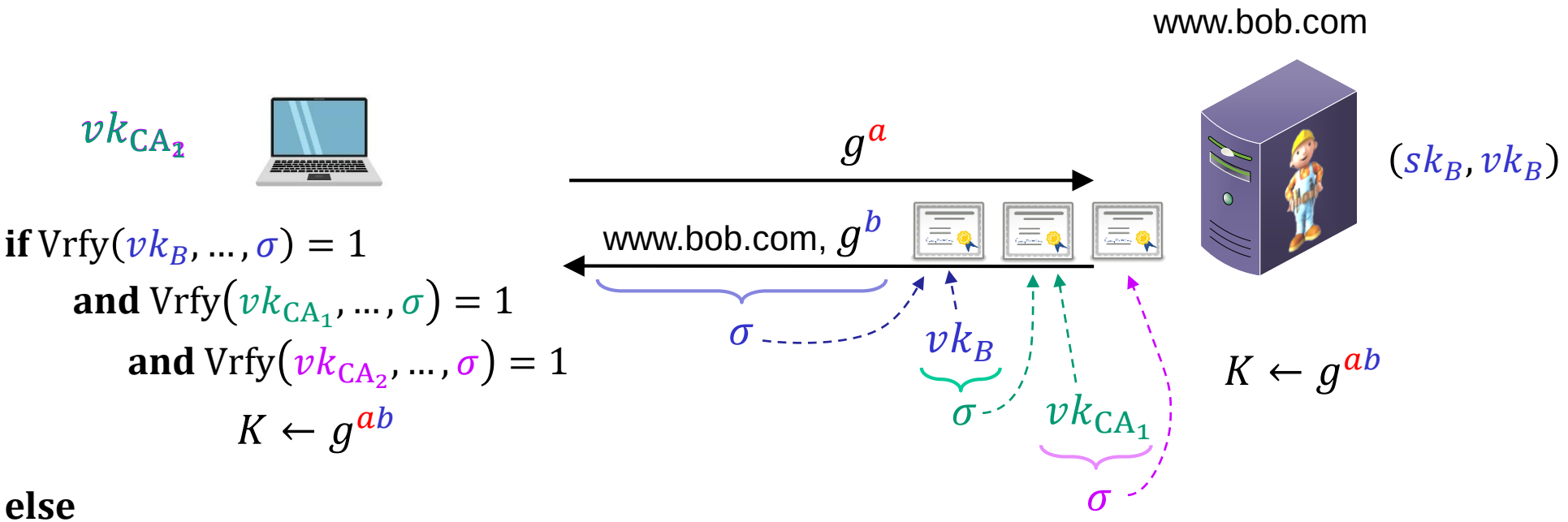


Let's Encrypt

Certificate chains



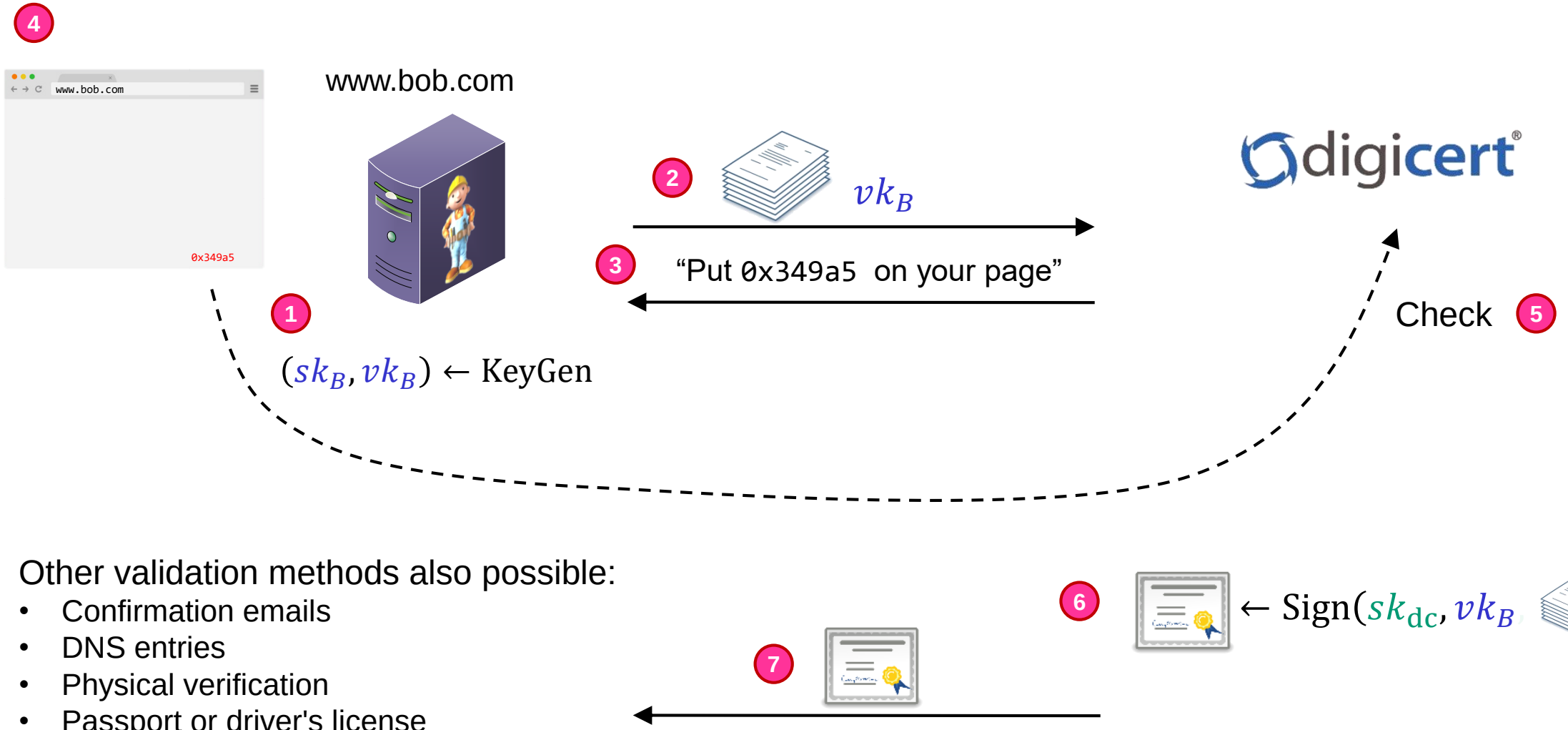
HTTPS / TLS + PKI



if $\mathrm{Vrfy}(vk_B, \ldots, \sigma) = 1$
  and $\mathrm{Vrfy}(vk_{\mathrm{CA}_1}, \ldots, \sigma) = 1$
    and $\mathrm{Vrfy}(vk_{\mathrm{CA}_2}, \ldots, \sigma) = 1$
      $K \leftarrow g^{ab}$

else
  abort

How to get a signed certificate?



Other validation methods also possible:

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

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



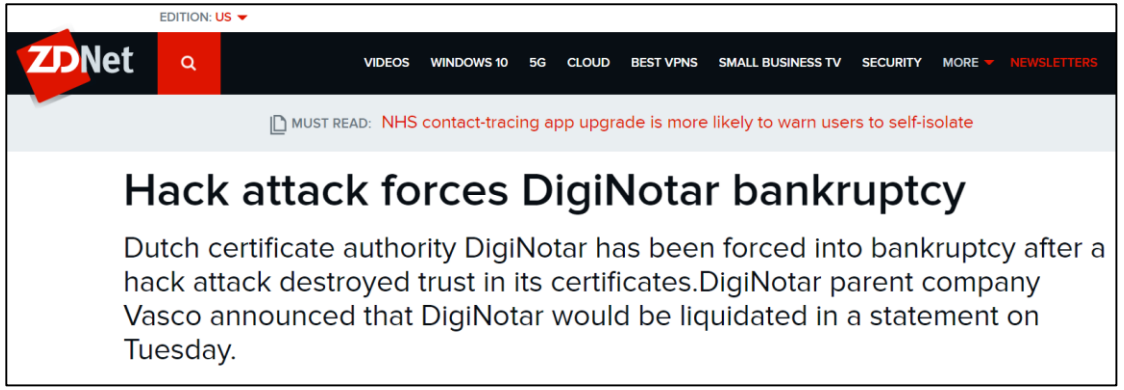
- Lot's of auditing and paperwork

- Many formal technical and non-technical security requirements
  - CA/Browser forum
  - Baseline Requirements v1.7.3

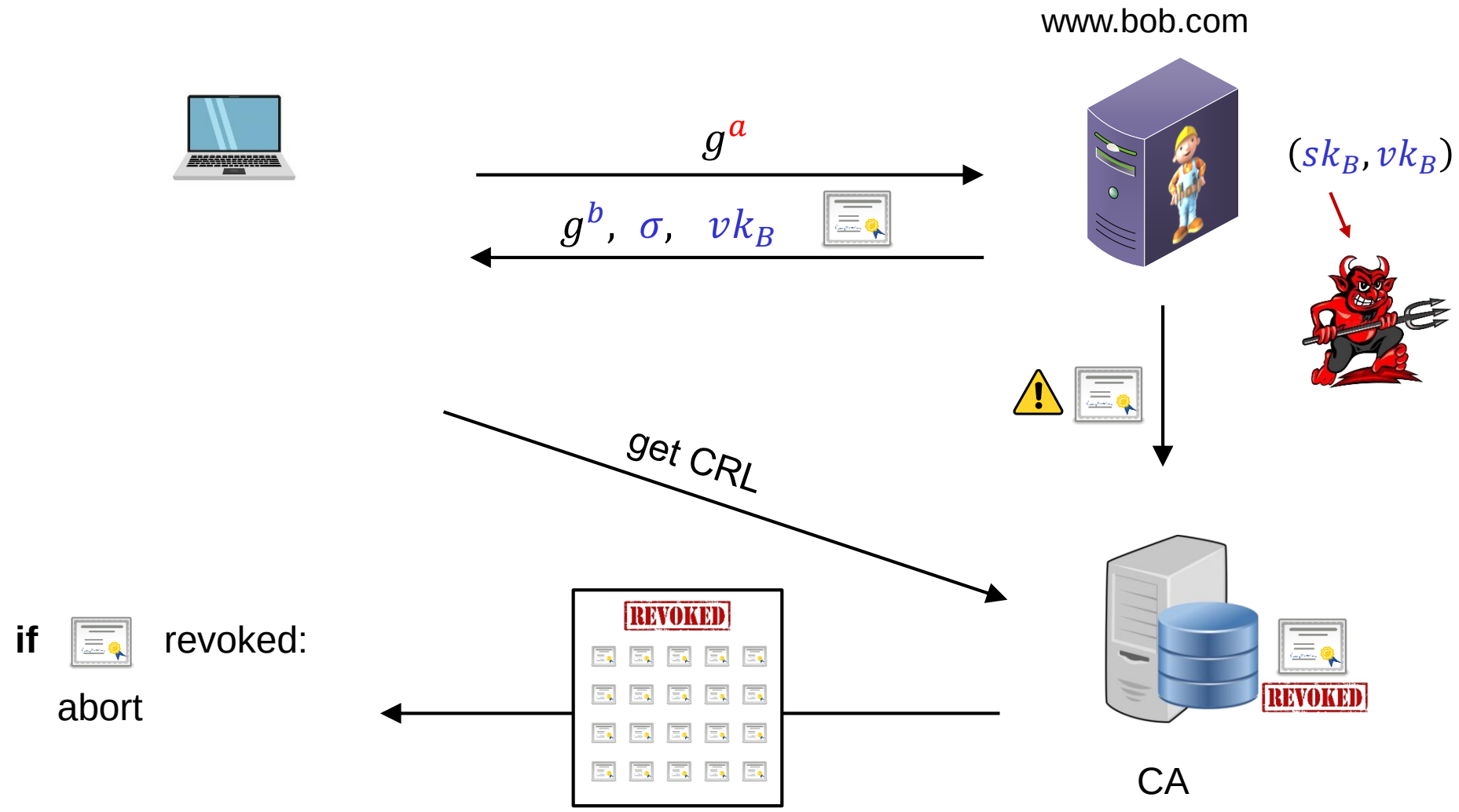
DigiNotar

- 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
- Triggered the creation of Certificate Transparency

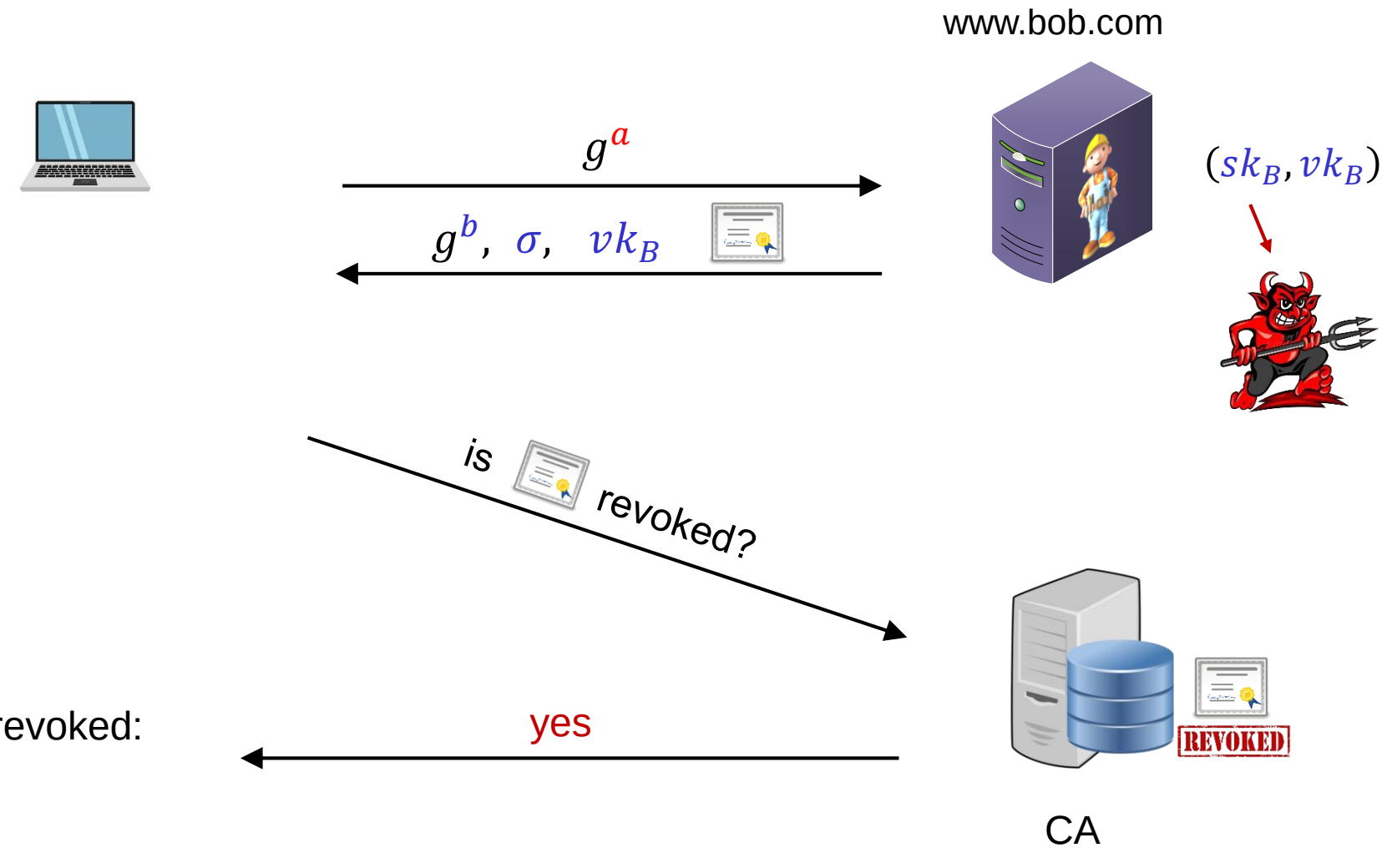




Certificate revocation



Certificate revocation – OCSP (Online Certificate Status Protocol)



Certificate revocation – OCSP



g^a

$g^b,\ \sigma,\ vk_B$

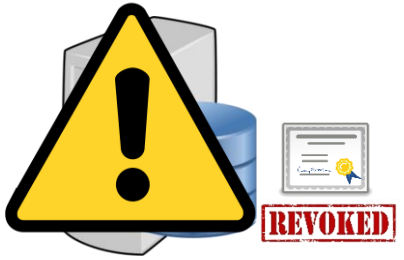


www.bob.com

(sk_B, vk_B)



is revoked?



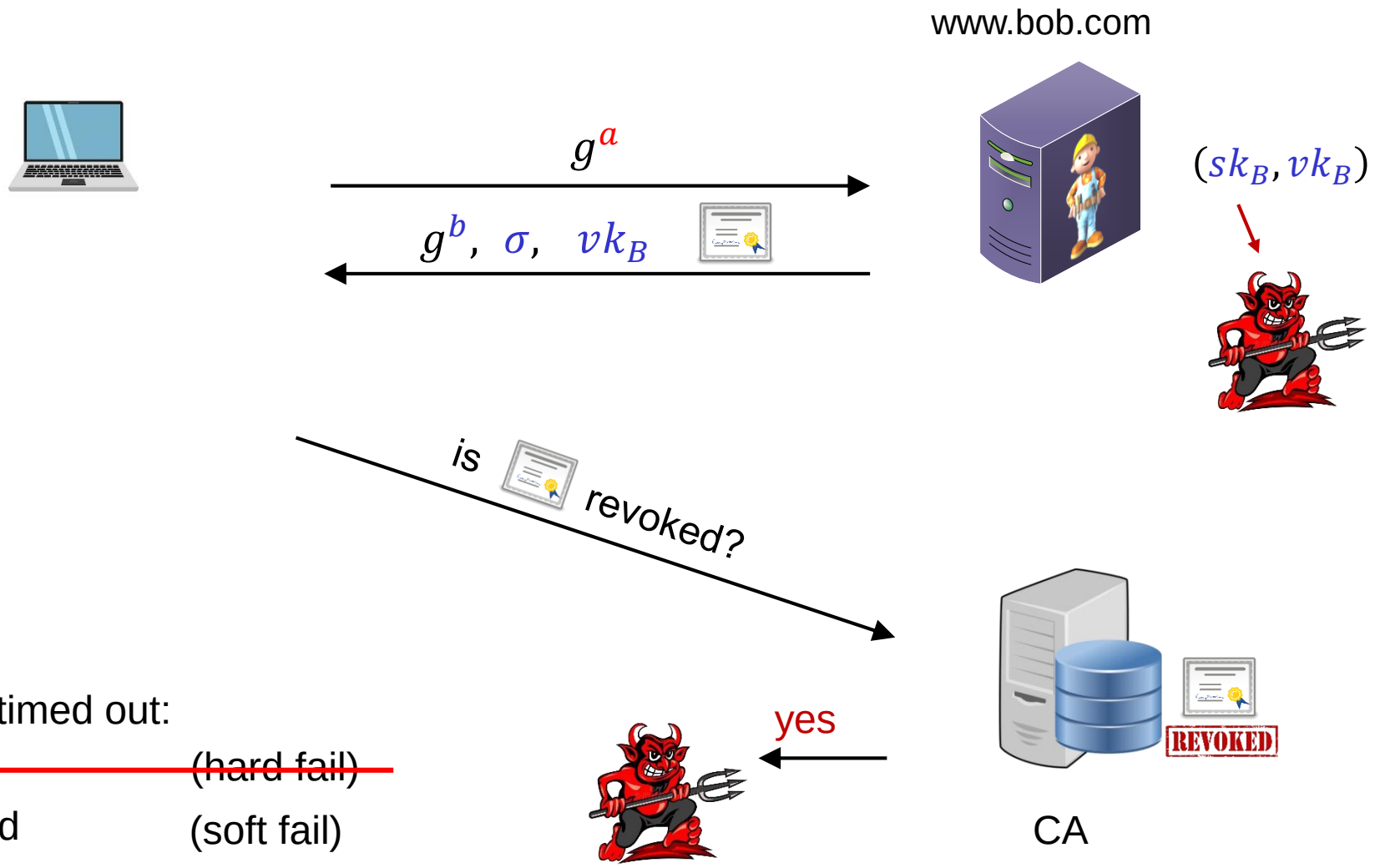
CA

if request timed out:

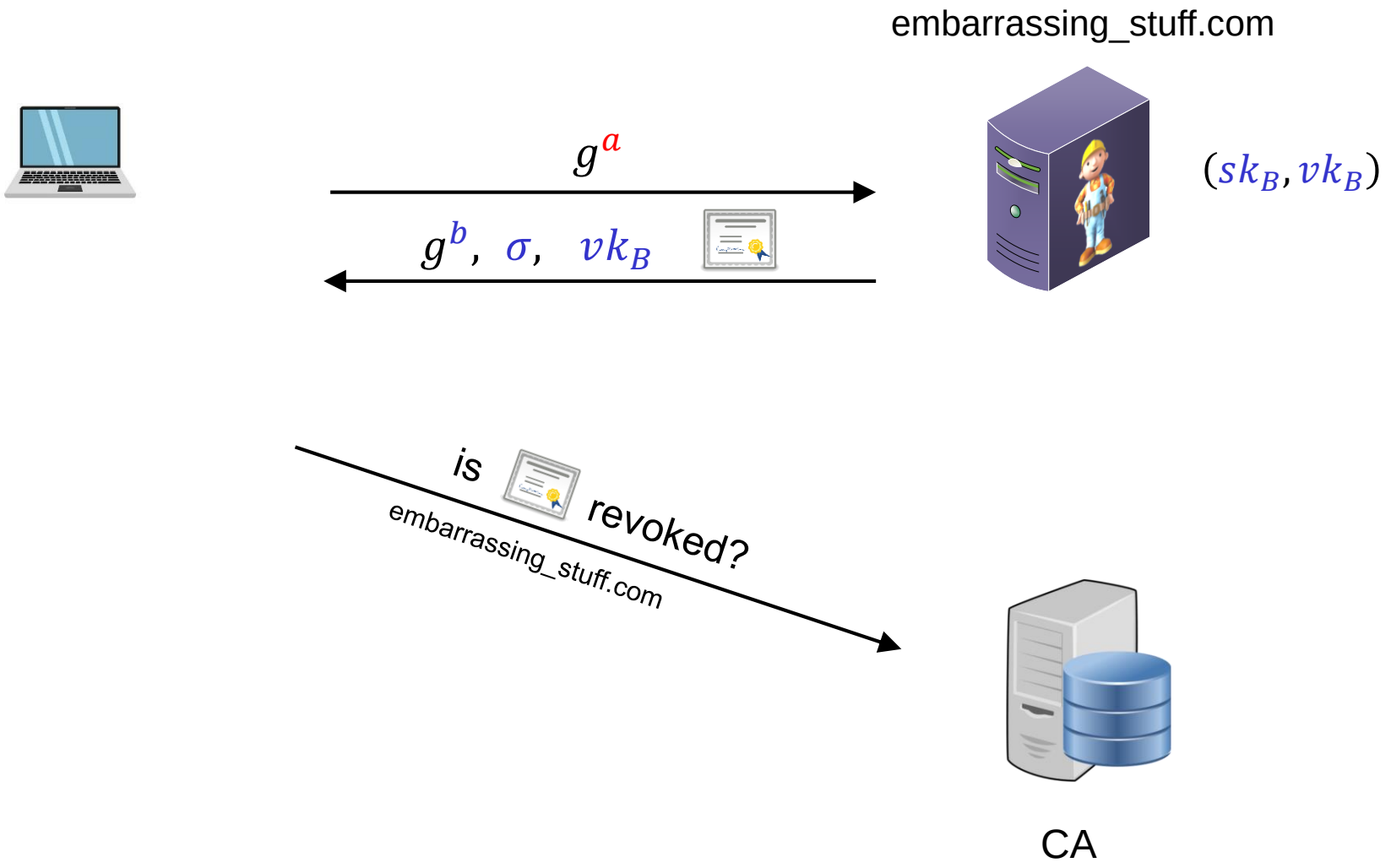
~~abort~~ ~~(hard fail)~~

cert valid (soft fail)

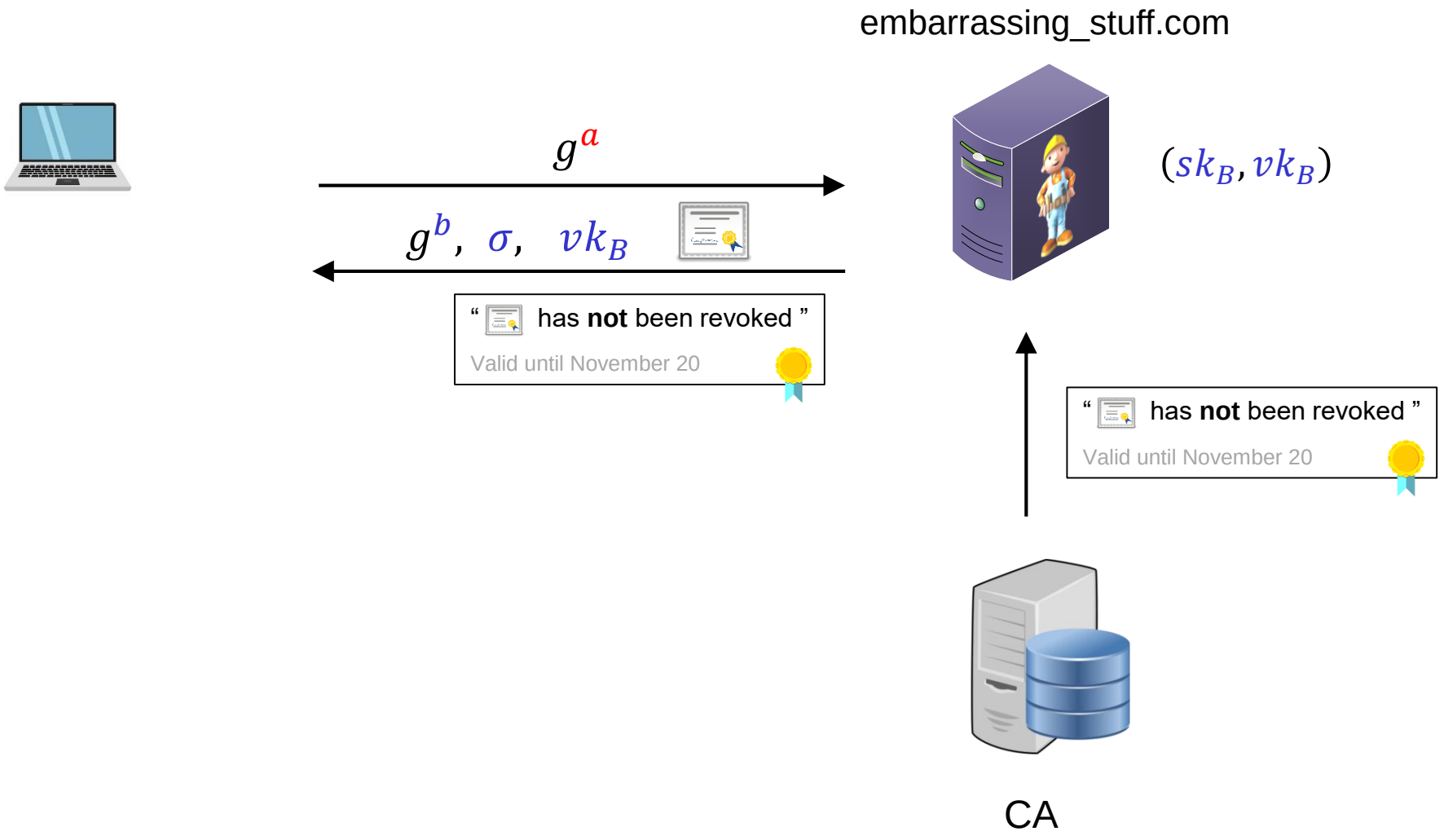
Certificate revocation – OCSP



Certificate revocation – OCSP

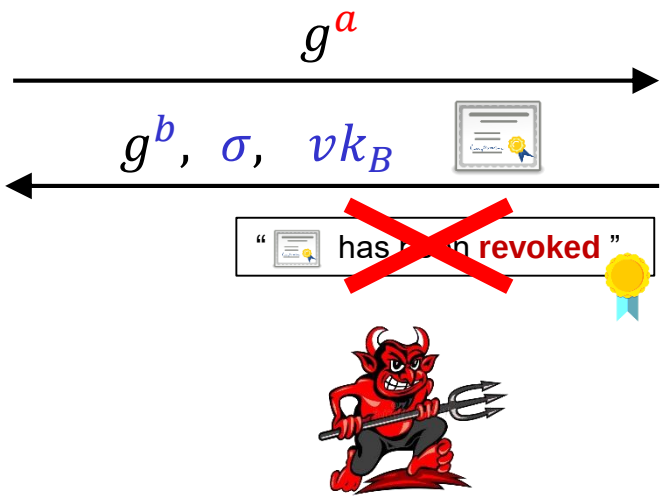


Certificate revocation – OCSP stapling



Certificate revocation – OCSP stapling





embarrassing_stuff.com



(sk_B, vk_B)





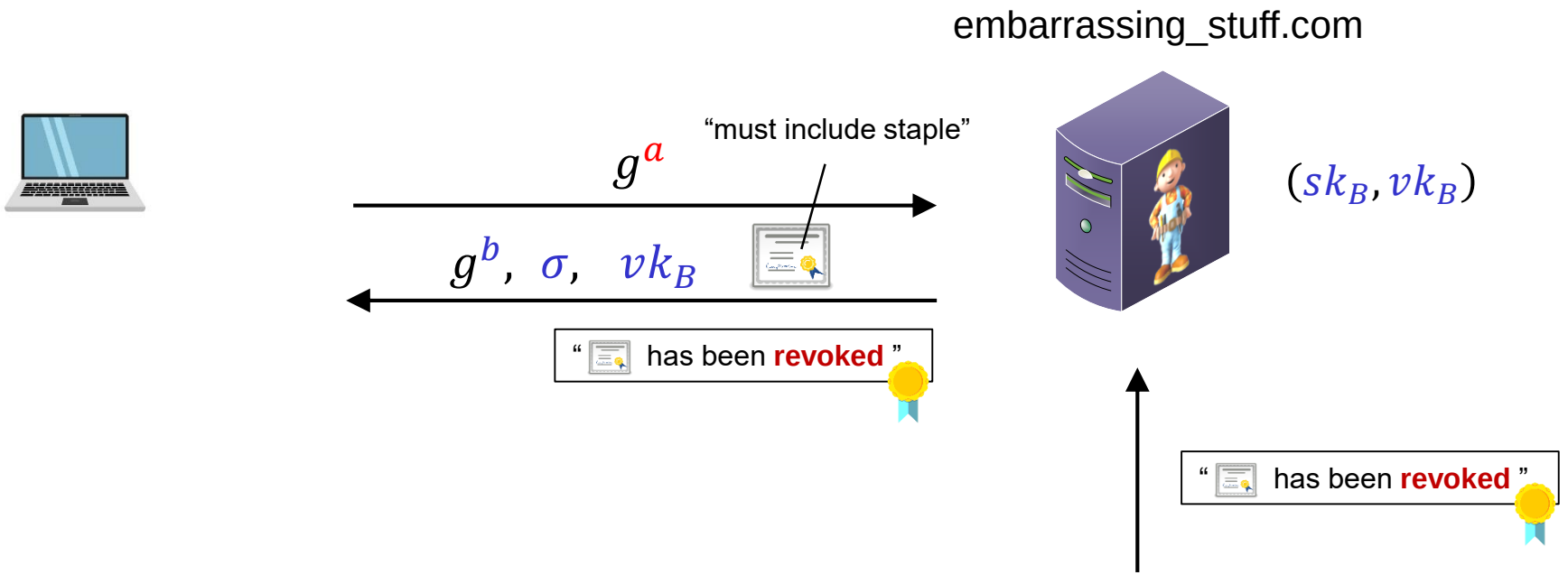
CA

if OCSP staple not present:

~~abort (hard fail)~~

cert valid (soft fail)

Certificate revocation – OCSP stapling + must_staple

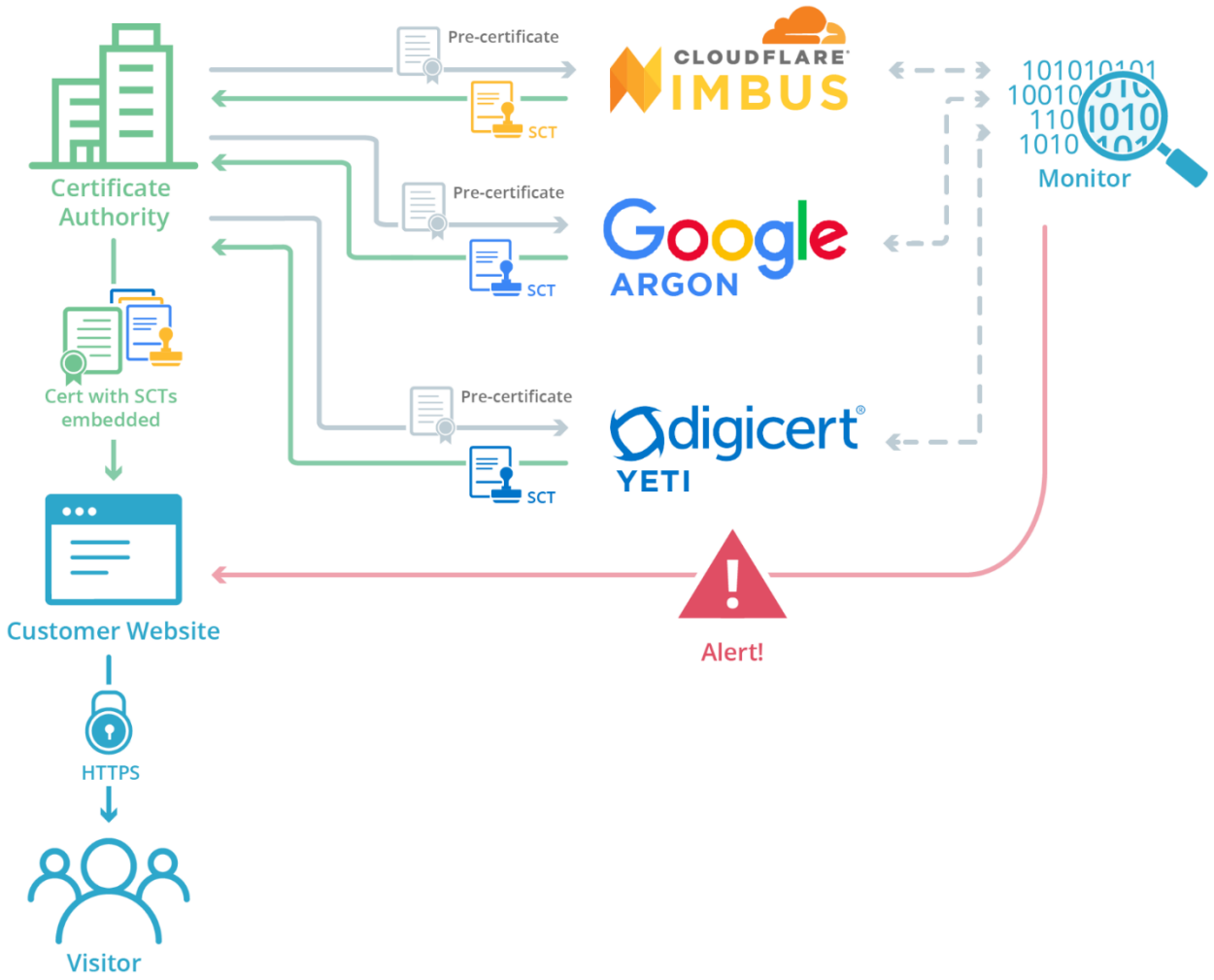


if OCSP staple not present:

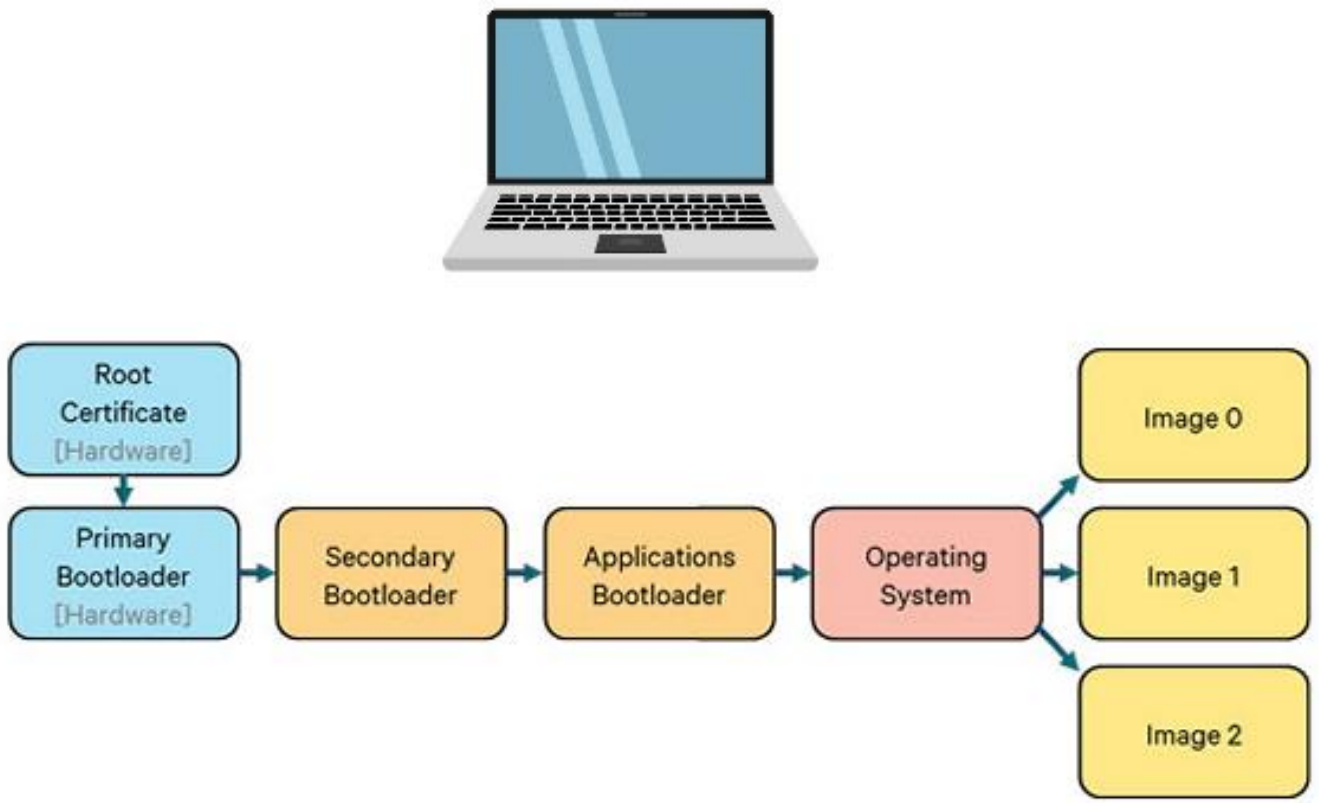
~~abort (hard fail)~~

~~cert valid (soft fail)~~

Certificate Transparency

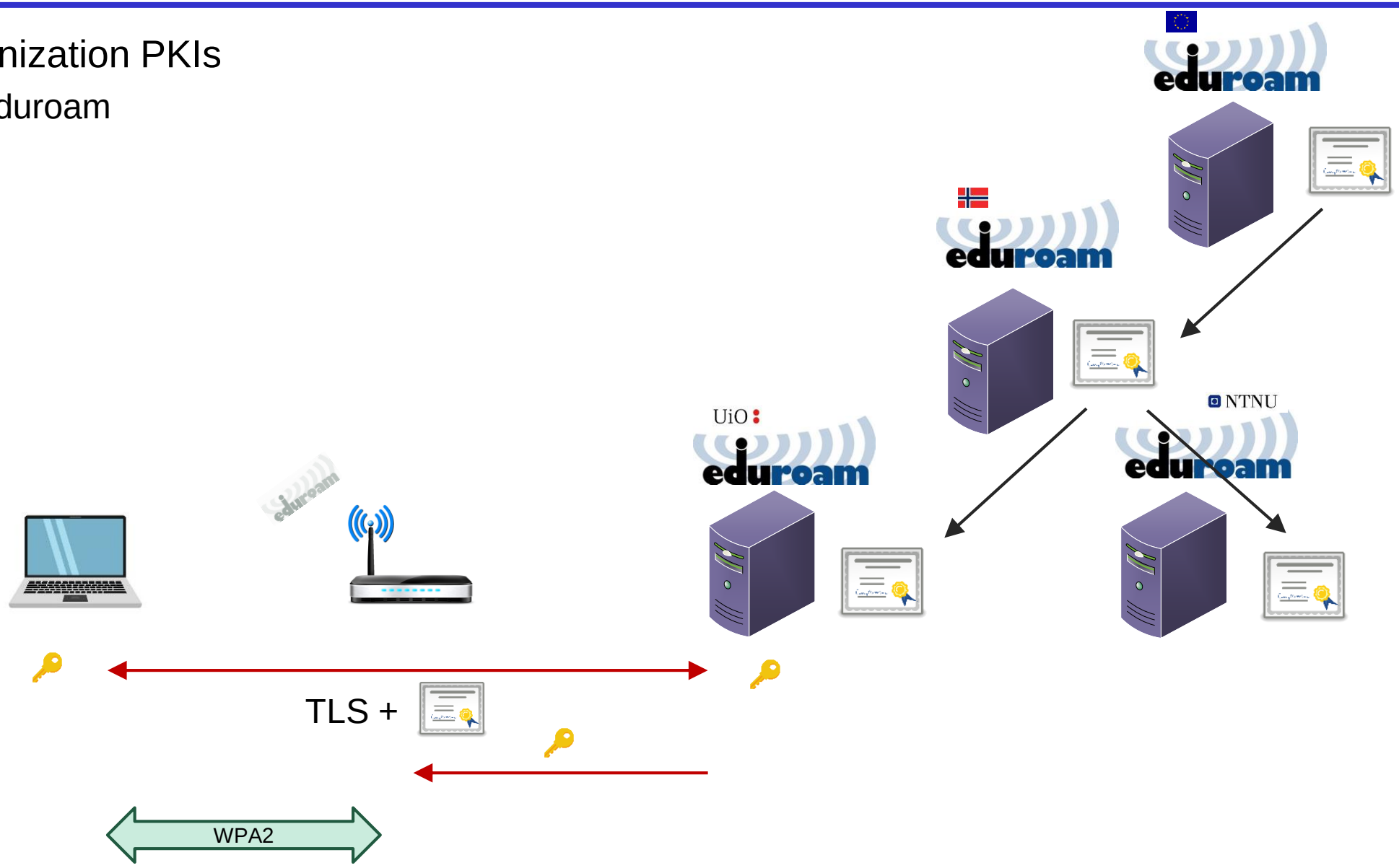


Hardware Root of Trust



Other PKIs exist

- Organization PKIs
  - eduroam



Next week

- Quantum computers

