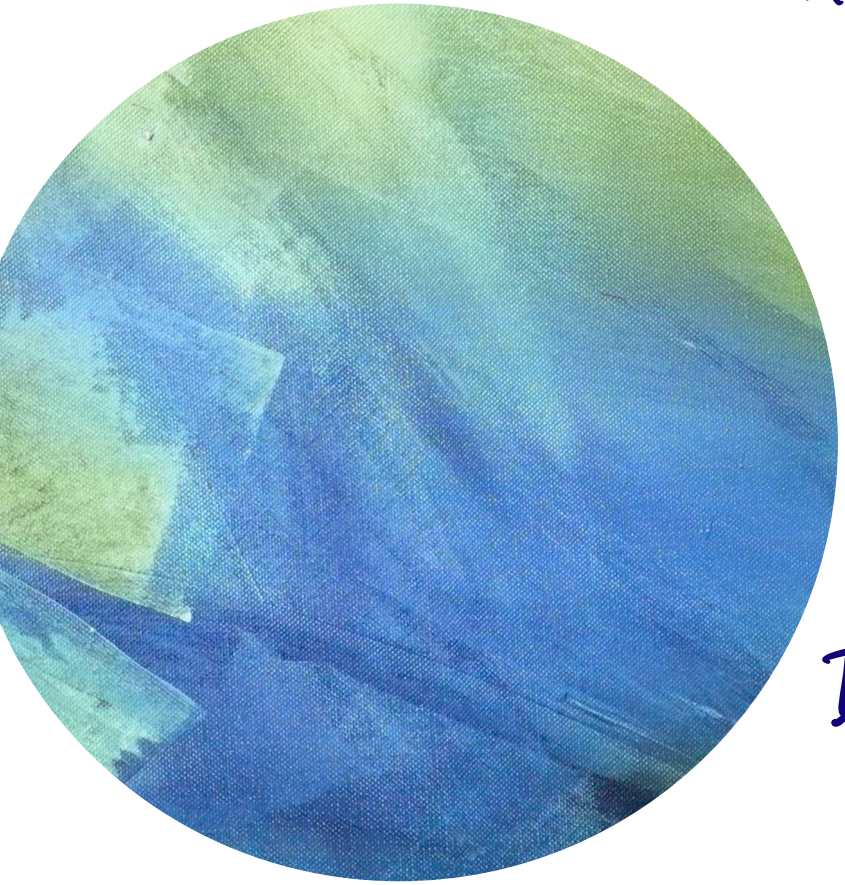
Lecture 15

Scribe: Kartik

Outline



Digital

~~Commitments~~

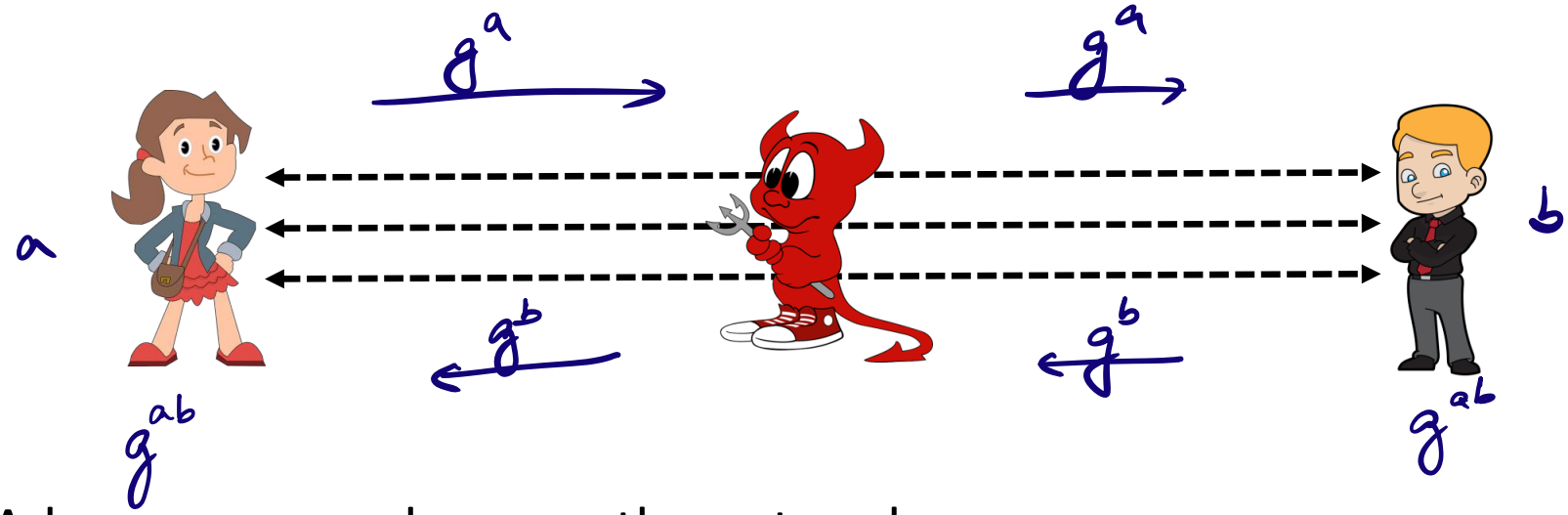
Signatures

~~Zero Knowledge~~

Authenticated Key Exchange

Authenticated Key Exchange

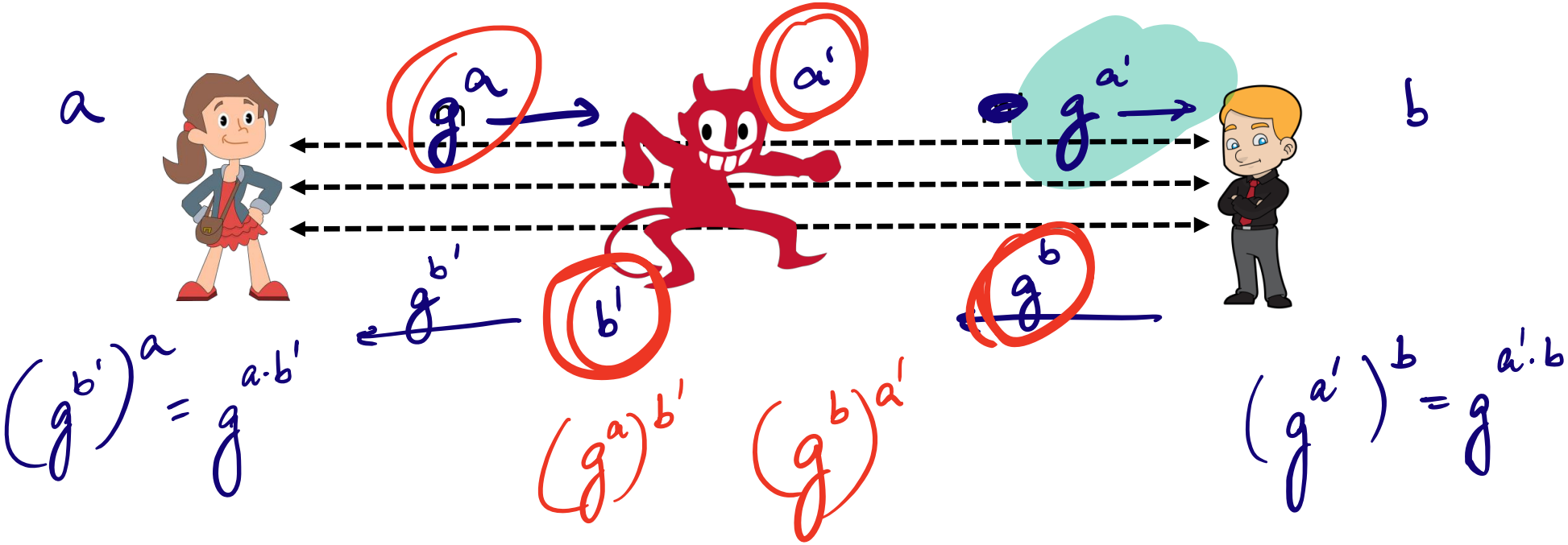
- Recall key exchange



- Adversary eavesdrops on the network

Authenticated Key Exchange

- Adversary has complete control of the network
  - Modify/inject/delete packets



Trusted Third Parties

- Adversary has complete control of the network
  - Modify/inject/delete packets

Authenticated Key Exchange

Important Component: Signature Scheme

Public-key MAC

verification key. is public

signing key is hidden

sk.

$\boxed{vk}$

$$\xrightarrow{m,\ tag} \quad = Sign_{sk}(m)$$

PUBLIC ALGORITHM

$\boxed{Verify_{vk}(m, tag)}$

PUBLIC

Digital Signatures

$$\mathrm{Gen}(1^n; r) \to (vk, sk)$$

$$\mathrm{Sign}(m, sk, r) \to \sigma$$

$$\mathrm{Ver}(m, \sigma, vk) \to \underset{\text{reject}}{0} \mid \underset{\text{accept}}{1}$$

Correctness: $\forall m$, $\mathrm{Verify}(m, \sigma, vk) = 1$

whenever $\sigma = \mathrm{Sign}(m, sk, r)$ and $(vk, sk) \leftarrow \mathrm{Gen}$

Digital Signatures

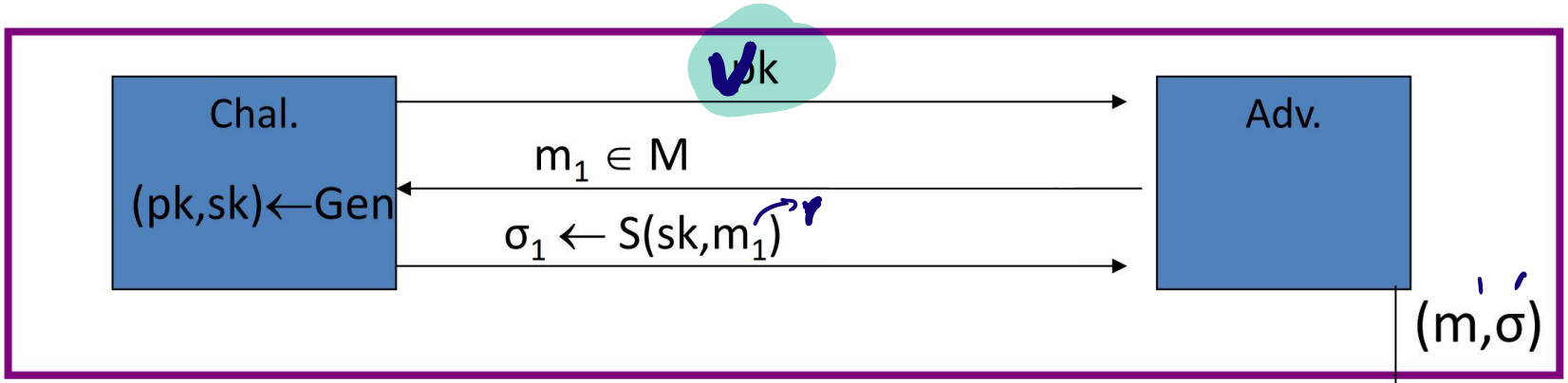
Public-key version of “MACs”

One-Time Signatures

: A can come up with a new signature σ' on $m' = m$.

- Definition of security

For a sig. scheme (Gen,S,V) and adv. A define a game as:



Adv. wins if ~~V(pk,m,σ) = 'accept'~~ and

Verify $(vk, m', \sigma') =$ accept AND $m' \neq m$

STRONG One-time sign. : Adv wins if Verify$(vk, m', \sigma') = 1$ AND $(m', \sigma') \neq (m, \sigma)$.

Lamport One-Time Signatures (from OWF)

- vk =

y_0	$f(x_1^0)$	$f(x_2^0)$	$f(x_3^0)$			$f(x_n^0)$
y_1	$f(x_1^1)$	$f(x_2^1)$	$f(x_3^1)$			$f(x_n^1)$

f: one-way function

- sk =

x_1^0 x_2^0 . . . . . x_n^0

x_1^1 x_2^1 x_n^1

each x_i^b is 256-bits.

$m = 0110101$

$m' = 0111101$

- Sign$(m = b, sk) \longrightarrow \sigma = x_b^0$
- Verify$(m = b, \sigma, vk) \longrightarrow$ check if $y_b = f(\sigma)$

Lamport One-Time Signatures

strong s.s.

- vk =

y_1^0	y_2^0				y_n^0
y_1^1	y_2^1				y_n^1

- One-time security (definition):

Ch

$\xrightarrow{vk}$

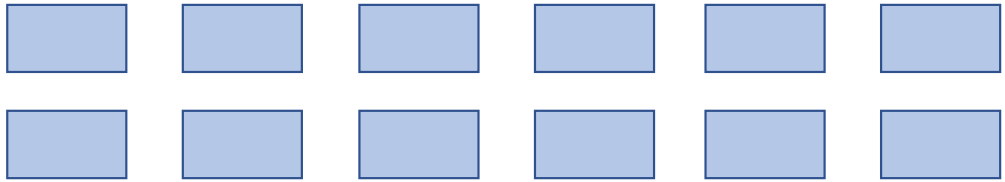
$\xleftarrow{m_i}$

$\xrightarrow{\{x_i^{m_i}\}_{i \in [n]}}$

$\xrightarrow{(m', \sigma')}$ s.t. $m' \neq m$

Lamport One-Time Signatures

- vk =



- One-time security (proof):

$\sigma' = \{x_i^{m_i'}\}_{i \in [n]}$

for $m_i' \neq m_i$ for some i

$\Downarrow$

$\left(x_i^{m_i'} \text{ is the inverse of } y_i^{m_i'} \text{ for } m_i' \neq m_i \right)$

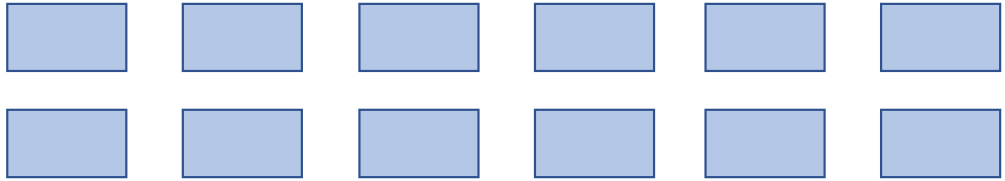
$\Leftarrow$

Attacker is inverting OWF

$\Downarrow$

contradiction.

Lamport One-Time Signatures



- What about two-time security?

$0 \ldots 0$

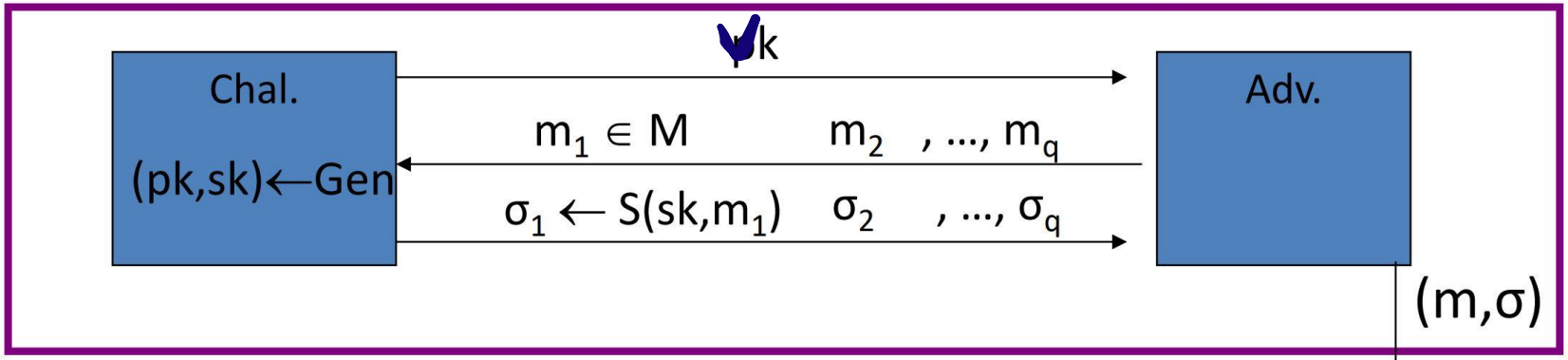
$\{x_i^0\}_{i \in [n]}$

$1 \ldots 1$

$\{x_i^1\}_{i \in [n]}$.

Many-Time Signatures

For a sig. scheme (Gen,S,V) and adv. A define a game as:

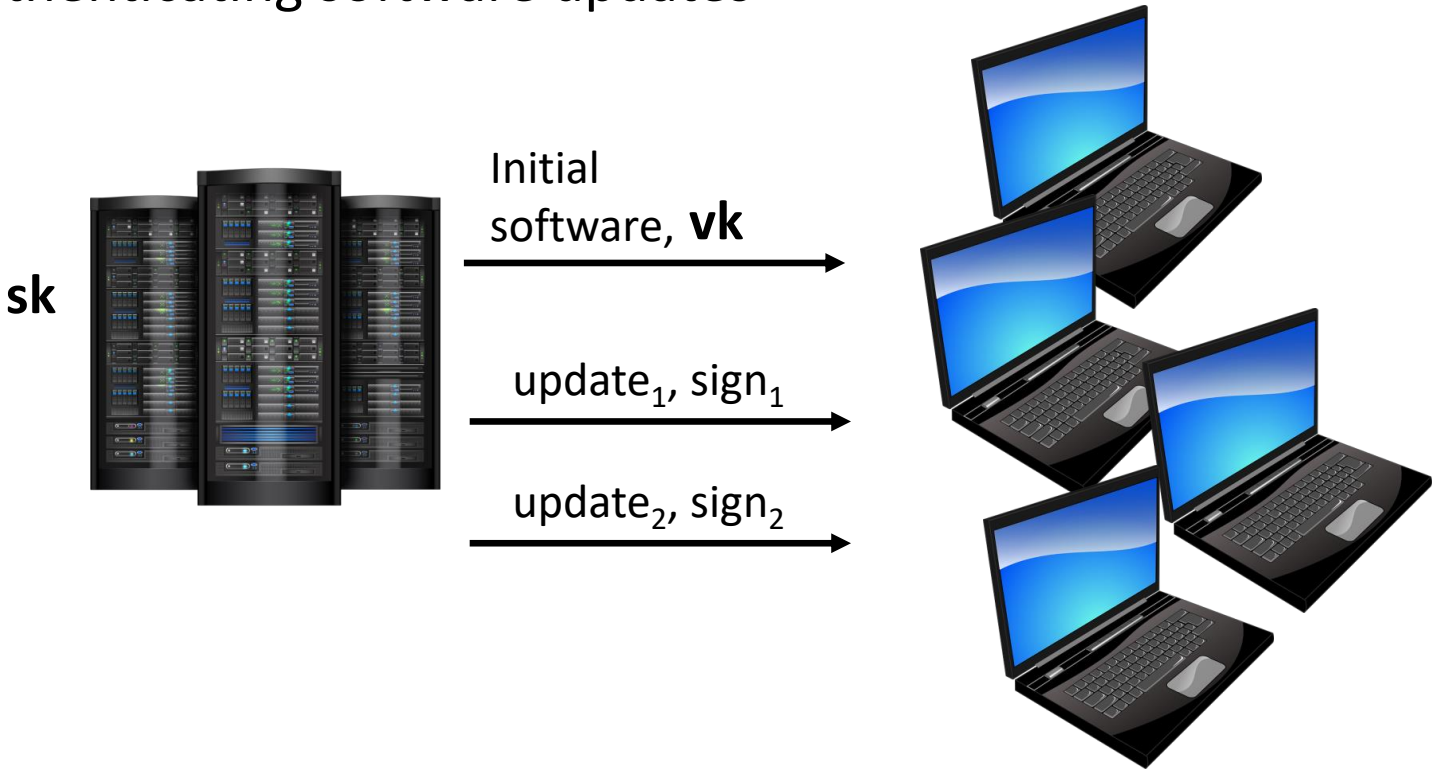


Adv. wins if **$V(pk,m,\sigma)$ = `accept'** and **$m \notin \{m_1, \dots, m_q\}$**

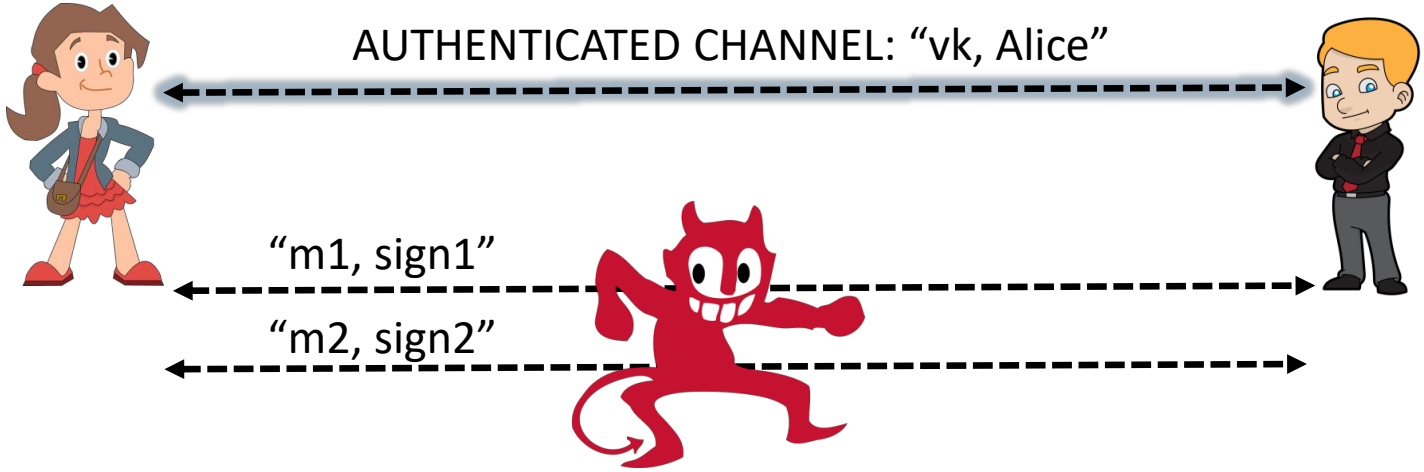
STRONG : $(m,\sigma) \notin \{(m_1,\sigma_1) \dots (m_q,\sigma_q)\}$

Applications?

- Authenticating software updates



One-time => Many-time authenticated channel



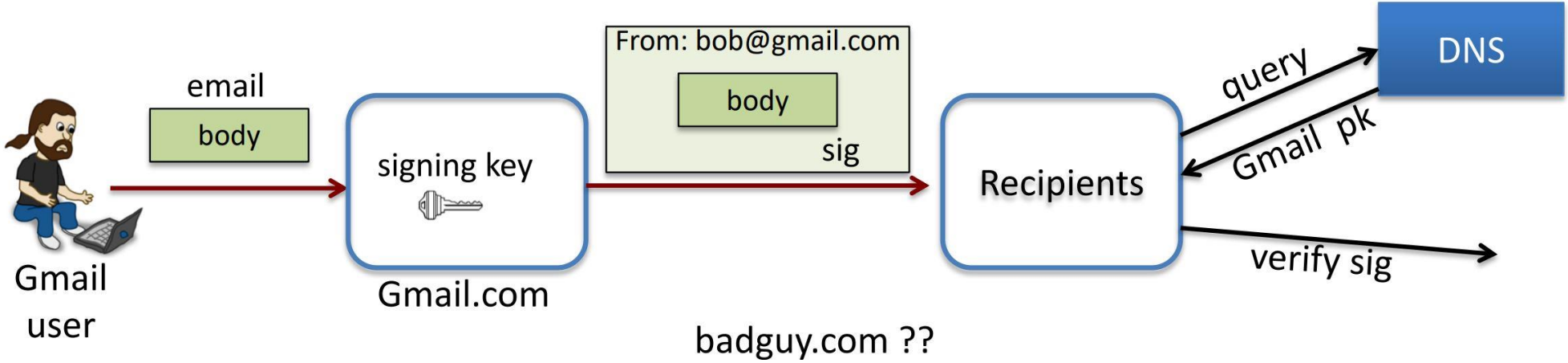
Signing email: DKIM (domain key identified mail)

Problem: bad email claiming to be from **someuser@gmail.com**

but in reality, mail is coming from domain **baguy.com**

⇒ Incorrectly makes gmail.com look like a bad source of email

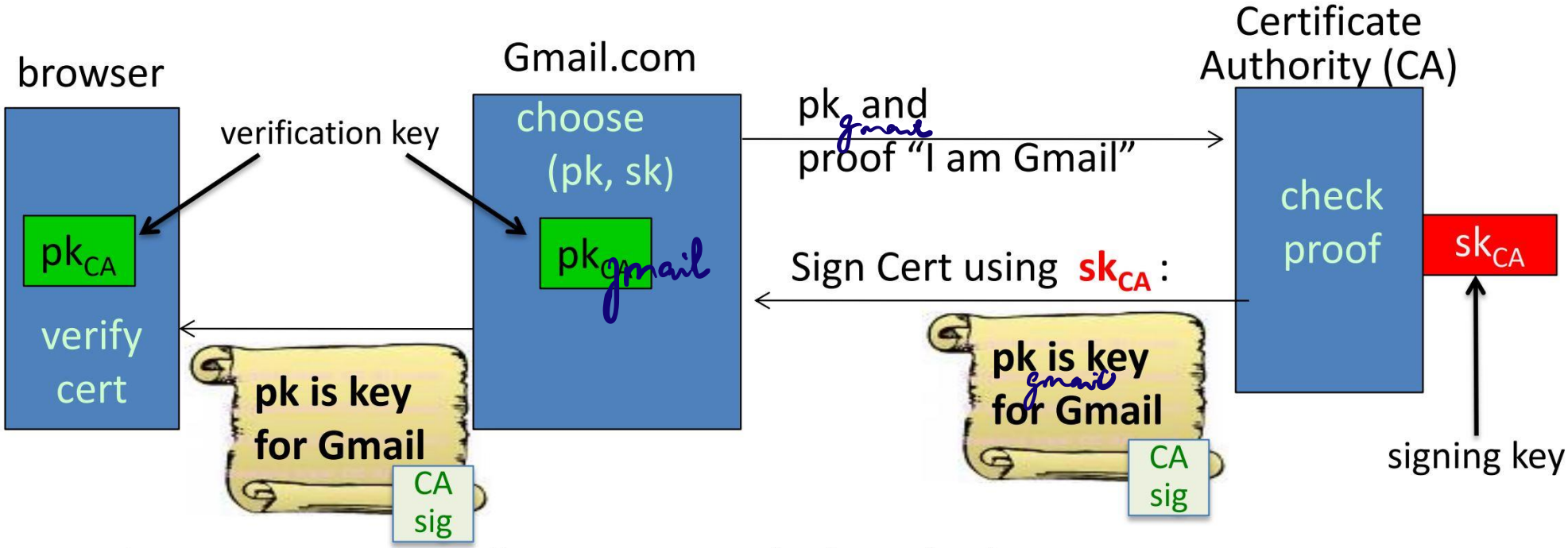
Solution: **gmail.com** (and other sites) sign every outgoing mail



Important application: Certificates

Problem: browser needs server's public-key to setup a session key

Solution: server asks trusted 3rd party (CA) to sign its public-key pk



Server uses Cert for an extended period (e.g. one year)

MACs versus signatures

- Use MAC when
  1. signer & verifier are same
  2. single signer & single verifier

- Use signature when
  1. many verifiers for same signer
  2. public verification.

Can you use MACs here?

- Authenticating software updates