Digital Signatures

Sushmita Ruj

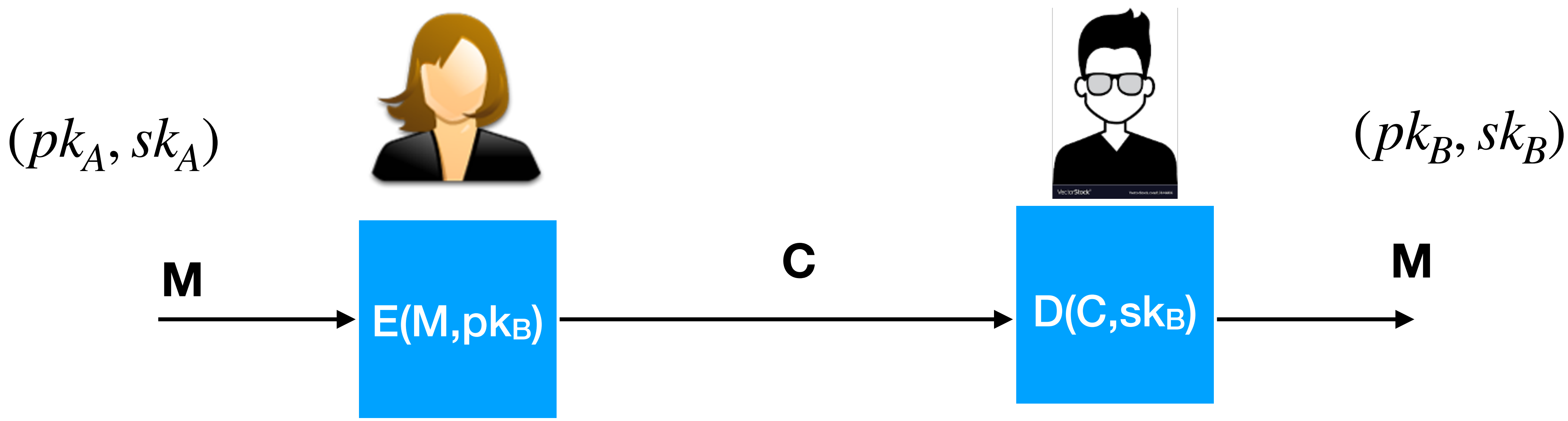
Recap

- Key agreement
- Diffie Hellman Key Exchange
- Discrete Logarithm Problem
- Key Derivation Function
- El Gamal Encryption algorithm

This Lecture

- Digital Signatures
- RSA Signatures
- Digital Signature Algorithm
- Construction of signature algorithms
- Certificate Transparency

Public Key Cryptography



- $\varepsilon = (\mathscr{M}, \mathscr{C}, \mathscr{K})$

Alice wants to send a message to Bob secretly

- $KG(1^k) \rightarrow (pk_A, sk_A), (pk_B, sk_B), \ldots$

- $For\ m \in \mathscr{M}, E(m, pk_B) \rightarrow c$

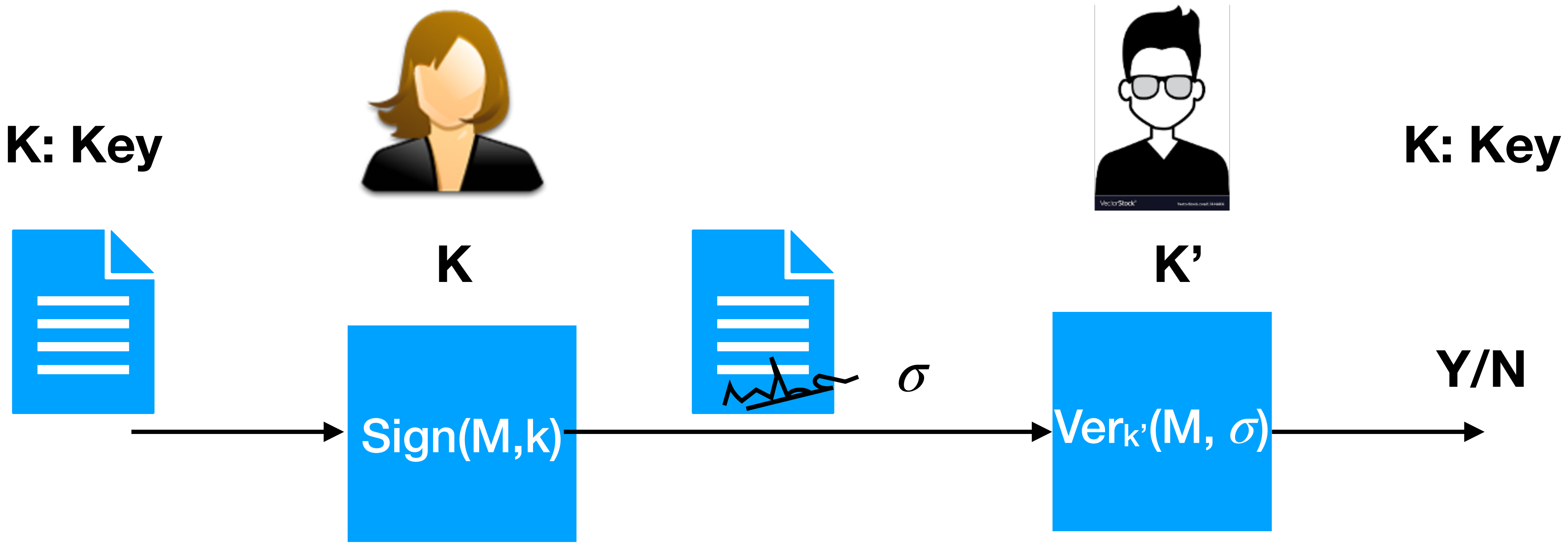
Public key of Bob known to everyone
E is a randomised algorithm

- $D(c, sk_B) \rightarrow m'$

Secret key known only to Bob, else only Bob can decrypt
D is a deterministic algorithm

- Correctness: $\forall k \in \mathscr{K}$ and messages $m \in \mathscr{M}$, if we execute $c \overset{R}{\leftarrow} E(m, pk_B)$, $m' \leftarrow D(c, sk_B)$, then with probability 1, $m = m'$

Digital Signatures



Digital Signature

- Definition: A digital signature scheme (KG, Sign,Ver) is a triple of algorithms:

- $KG(1^k) \rightarrow (vk_A, sk_A), (vk_B, sk_B), \ldots$ (Key generation) algorithm)

- $For \;\; m \in \mathcal{M}, Sign(m, sk_A) \rightarrow \sigma$

- $Ver(vk, m, \sigma) \rightarrow 0/1$ if σ is the correct signature on message m

- Consistency: $\forall (sk, vk)$ output by KG :

$$\forall m \in \mathcal{M} : Ver(vk, m, Sign(sk, m)) = 1$$

Security of Signature Schemes

Attacker's power: **chosen message attack**

- for $m_1, m_2, \ldots, m_q$ attacker is given $\sigma_i \leftarrow \mathrm{Sign}(sk, m_i)$

Attacker's goal: **existential forgery**

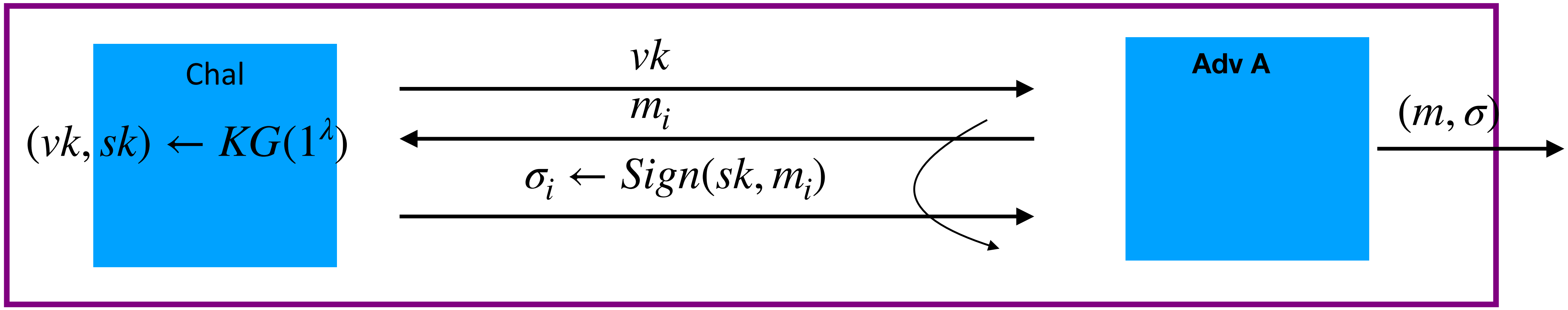
- produce some **new** valid message/signature pair (m, σ).

$$m \notin \{ m_1 , \ldots , m_q \}$$

Security : attacker cannot produce a valid signature for a new message

Security of Signature Schemes

For a sig. scheme (KG,Sign,Ver) and adv. A define a game as:



Adv. wins if **Ver(vk,m,σ) = `accept'** and **$m \notin \{m_1, \ldots, m_q\}$**

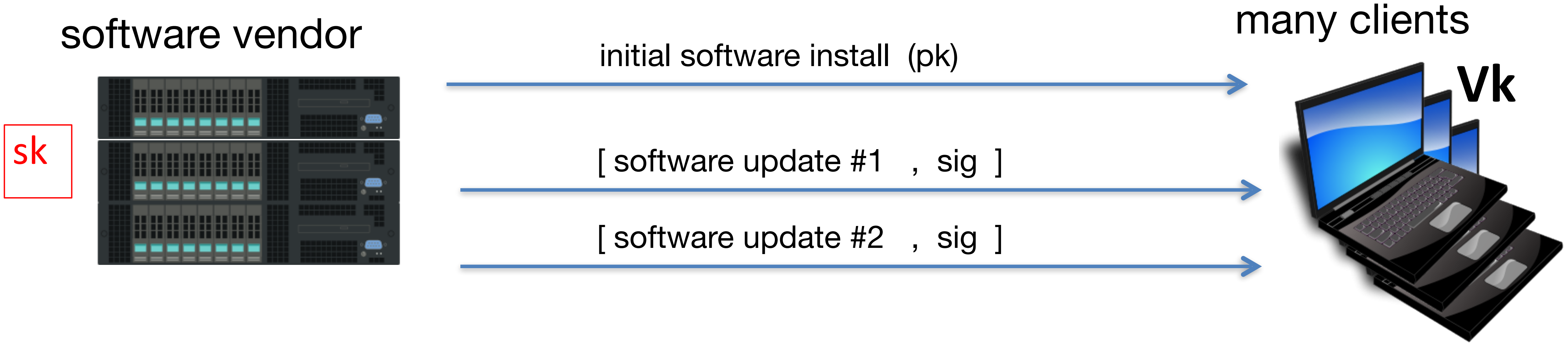
Def: SS=(KG,Sign,Ver) is **secure** if for all "efficient" A:

$Adv_{SIG}[A,SS]$ = Pr[A wins] is "negligible"

Applications

Code signing:

- Software vendor signs code
- Clients have vendor's pk.   Install software if signature verifies.

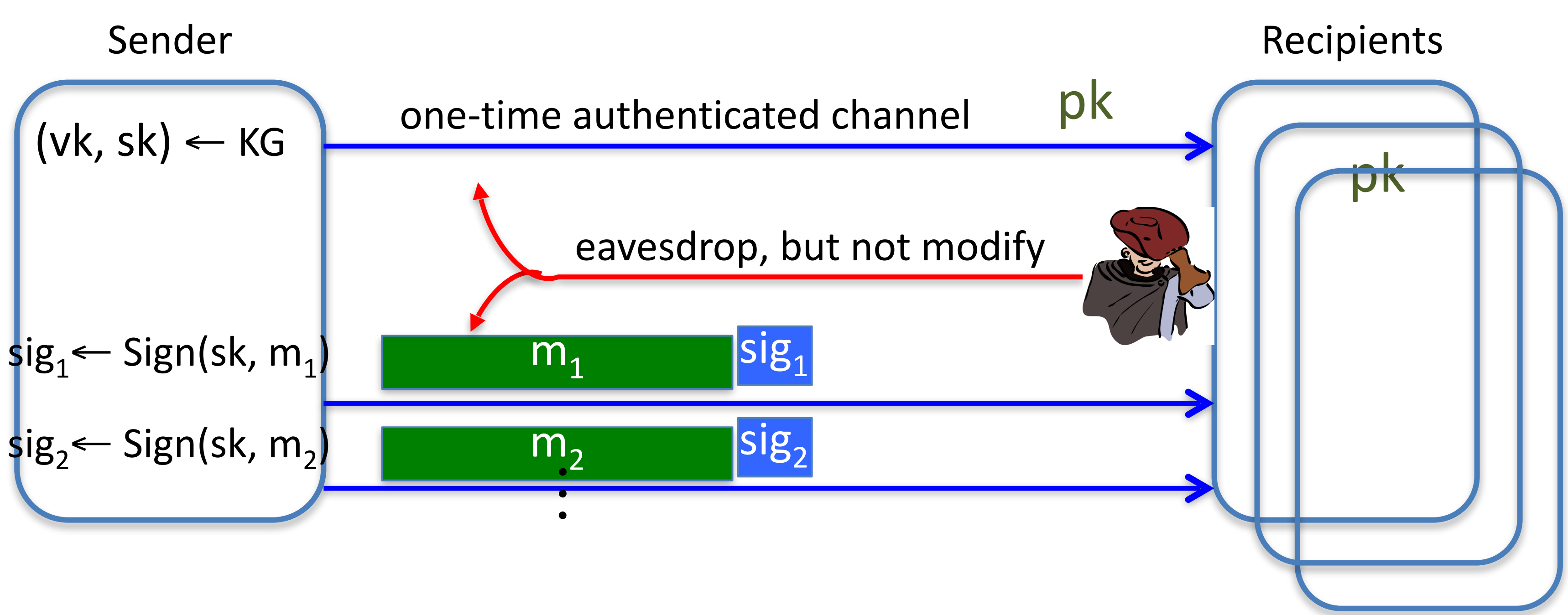


Applications

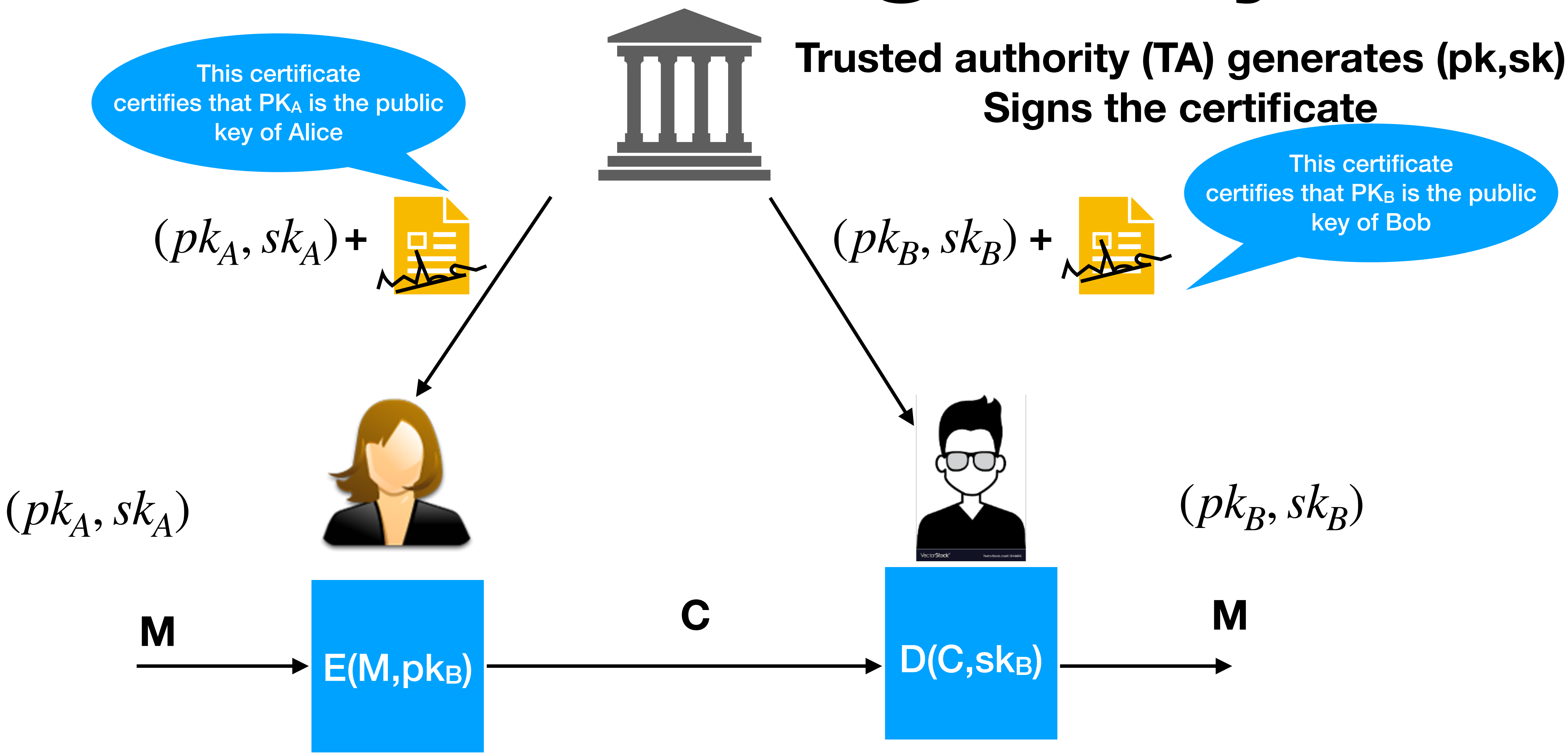
One-time authenticated channel (non-private, one-directional)

$\Longrightarrow$ many-time authenticated channel

Initial software install is authenticated, but not private



How to Manage Keys?



TA in real are Digicert, Lets Encrypt, Identrust, GoDaddy etc.

Problem is that TA can get corrupted, what happens then?

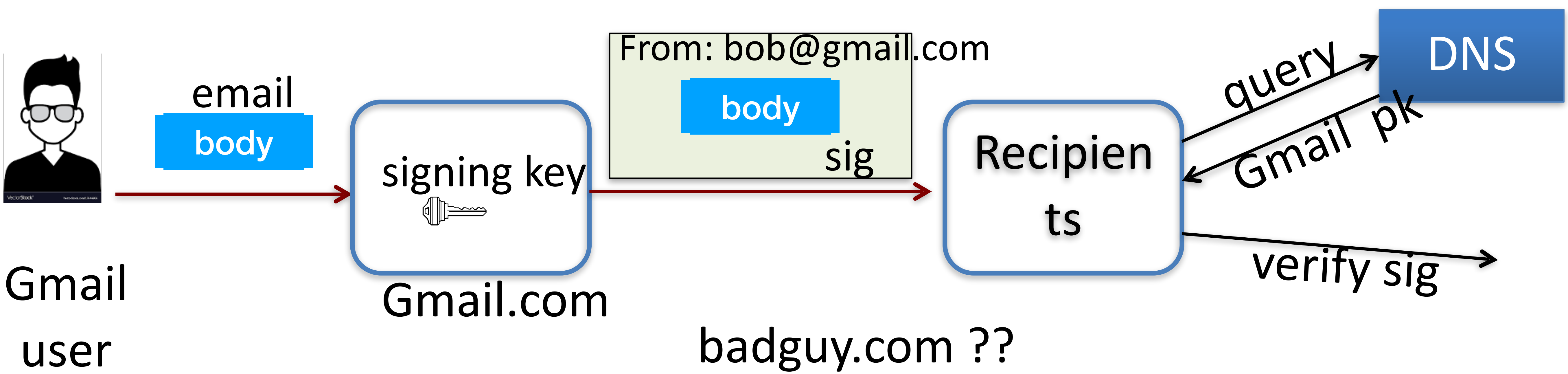
Signing email: DKIM
(domain key identified mail)

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



Example DKIM header from gmail.com

Example DKIM header from gmail.com

X-Google-DKIM-Signature: v=1; **a=rsa-sha256**; c=relaxed/relaxed;

d=1e100.net; s=20130820;

h=x-gm-message-state:mime-version:in-reply-to:references:from:date:
message-id:subject:to:content-type;

bh=MDr/xwte+/JQSgCG+T2R2Uy+SuTK4/gxqdxMc273hPQ=;

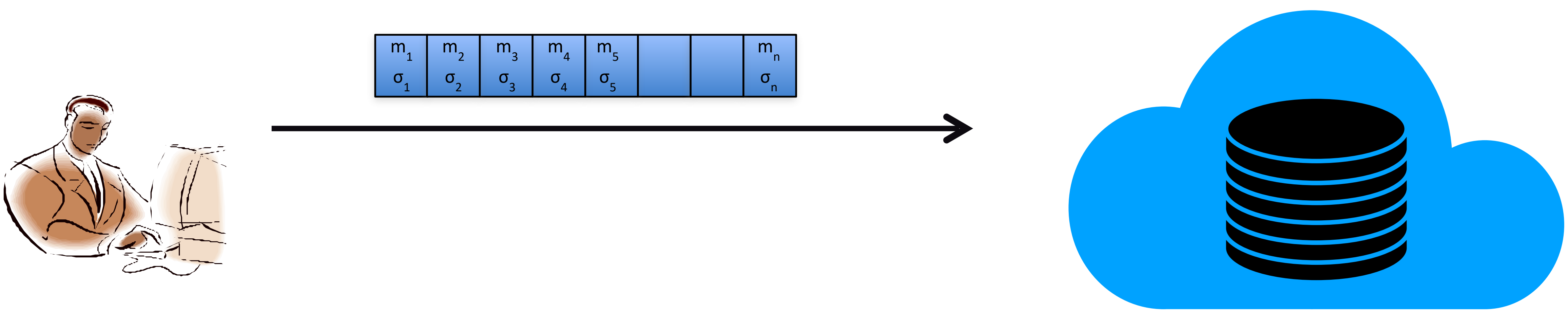
b=**dOTpUVOaCrWS6AzmcPMreo09G9viS+sn1z6g+GpC/ArkfMEmcffOJ1s9u5Xa5KC+6K
XRzwZhAWYqFr2a0ywCjbGECBPIE5ccOi9DwMjnvJRYEwNk7/sMzFfx+0L3nTqgTyd0ED
EGWdN3upzSXwBrXo82wVcRRCnQ1yUITddnHgEoEFg5WV37DRP/eq/hOB6zFNTRBwkvfS
0tC/DNdRwftspO+UboRU2eiWaqJWPjxL/abS7xA/q1VGz0ZoI0y3/SCkxdg4H80c61DU
jdVYhCUd+dSV5flSouLQT/q5DYEjINQbi+EcbL00Iiu4o623SDEeyx2isUgcvi2VxTWQ
m80Q==**

Gmail's signature on headers, including DKIM header (2048 bits)

MAC Vs Signature

- Private vs Public
- When the verifier is the one who creates the tag, then MAC
- If Verification is public, use Signatures

Data Auditing



σ_i is the tag (unforgeable)

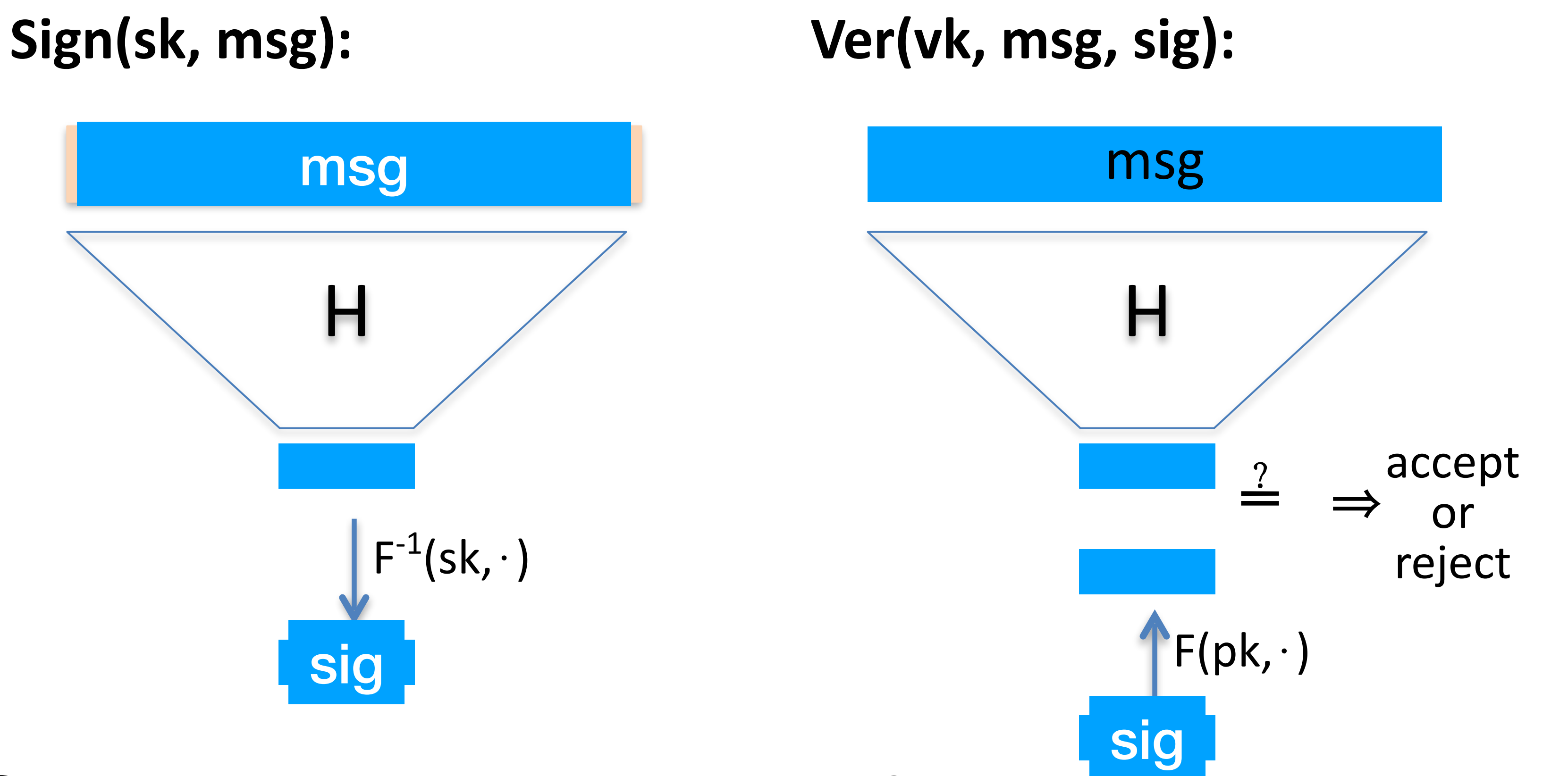
If data owner audits: Use MAC
If you need third party auditing: use signatures

Data Integrity

- Collision resistent hashing
- Signing
- MAC

Signature Construction

Full Domain Hash Functions



Let **Sig**=(Gen, S, V) be a signature scheme for short messages, say M = $\{0,1\}^{256}$

Let $H : M \to M'$ be a hash function (s.g. SHA-256)

Def: **SC** = (KG, Sign , Ver) for messages in M as:

Sign**(sk, m) = S(sk,H(m))** ; **Ver(vk, m, σ) = V(pk,H(m),σ)**

Thm: If Sig is a secure sig scheme for M' and H is collision resistant hash function then SC is a secure sig scheme for M

Signatures from One-Way-Functions

A function $f : X \to Y$ is a **one-way function** (OWF) if:

- for all $x \in X$ it is easy to compute compute $f(x)$
- Given $f(x)$, it is hard to find x

Example: f(x) = AES(x, 0) (x is secret)

Examples: Lamport Signatures (PQC Signatures)

Stateful: Size > 40 kb

Stateless:: Size < 4 kb

Signatures from Trapdoor Permutations

A function $f : X \to X$ is a trapdoor **permutation (TDP)** if:

- for all $x \in X$ it is easy to compute compute $f(x)$
- Given $f(x)$, it is hard to find x, unless trapdoor is known

Example: RSA

RSA Signatures

Most common and used for signing certificates

Full Domain Hash (FDH) Signatures

(G_{TDP}, F, F^{-1}):   Trapdoor permutation on domain X

$H: M \longrightarrow X$ hash function   (FDH)

$(KG, Sign, Ver)$ signature scheme:

- **KG**:   run G_{TDP} and output   pk, sk
- **Sign(sk, m∈M)**:   output $\sigma \longleftarrow F^{-1}(sk, H(m))$
- **Ver(vk, m, σ):**   output   'accept' if $F(pk, \sigma) = H(m)$
  'reject'   otherwise

RSA-FDH

KG: generate an RSA modulus $N = p \cdot q$ and $e \cdot d = 1 \bmod \phi(N)$

construct CRHF $H: M \longrightarrow Z_N$

output $pk = (N,e,H)$, $sk = (N,d,H)$

- $Sign(sk, m \in M)$: output $\boldsymbol{\sigma \longleftarrow H(m)^d \bmod N}$
- $Ver(vk, m, \sigma)$: output. 'accept' if $\boldsymbol{H(m) = \sigma^e \bmod N}$

Problem: having H depend on N is slightly inconvenient

Signatures from DLOG

Choose group cyclic G of order p and

Choose generator of $g \in G$, i.e. G (i.e. G = $\{1, g, g^2, g^3, \cdots, g^{p-1}\}$)

Discrete-log in G is hard if f(x) = g^x is a one-way function

- note: f(x+y) = f(x) · f(y)

Examples: = (multiplication mod p) for a large prime p

= (group of points on an elliptic curve mod p)

Signatures from DLOG: ElGamal, Schnorr, Digital Signature Algorithm (DSA), Elliptic Curve DSA, ECDSA etc.

Digital Signature Standard (DSS)/ Digital Signature Algorithm (DSA)

- KG: Generates (p,q,g). Let $H : \{0,1\}^* \rightarrow \mathbb{Z}_q$ be a function. Choose $x \xleftarrow{R} \mathbb{Z}_q$, set $y = g^x \bmod p$. $vk = \langle H, p, q, g, y \rangle, sk = \langle H, p, q, g, x \rangle$, Such that

  (1) p and q are primes with ||q|| = n;

  (2) $q \mid (p-1)$ but $q^2 \nmid (p-1)$; and

  (3) g is a generator of the subgroup of Z_p^*; having order q.

- Sign: : let $m \in \{0,1\}^*$, choose $k \xleftarrow{R} \mathbb{Z}_q^*$, compute $r = (g^k \bmod p) \bmod q$, compute $s = (H(m) + xr) \,.\, k^{-1} \bmod q$, output $\sigma = (r, s)$

- Ver: Compute $u_1 = H(m) \,.\, s^{-1} \bmod q$, $u_2 = r \,.\, s^{-1} \bmod q$

- Output 1 if $r = (g^{u_1} y^{u_2} \bmod p) \bmod q$, and 0 o.w.

- Does not have security proof

Digital Certificate (X.509 v3)

- Certificate
  - Version Number
  - Serial Number
  - Signature Algorithm ID
  - Issuer Name
  - Validity period
    - Not Before
    - Not After
  - Subject name
  - Subject Public Key Info
    - Public Key Algorithm
    - Subject Public Key
  - Issuer Unique Identifier (optional)
  - Subject Unique Identifier (optional)
  - Extensions (optional)
    - ...
- Certificate Signature Algorithm
- Certificate Signature

Rouge TA and Certificate Transparency


Print subscriptions | Sign in | Search jobs | Search | Australia edition

Support the Guardian
Fund independent journalism with $17 per month
Support us

The Guardian
A decade of making a difference

News | Opinion | Sport | Culture | Lifestyle | More

Australia World AU politics Environment Climate crisis Indigenous Australia Immigration Media Business Science Tech Podcasts Newsletters

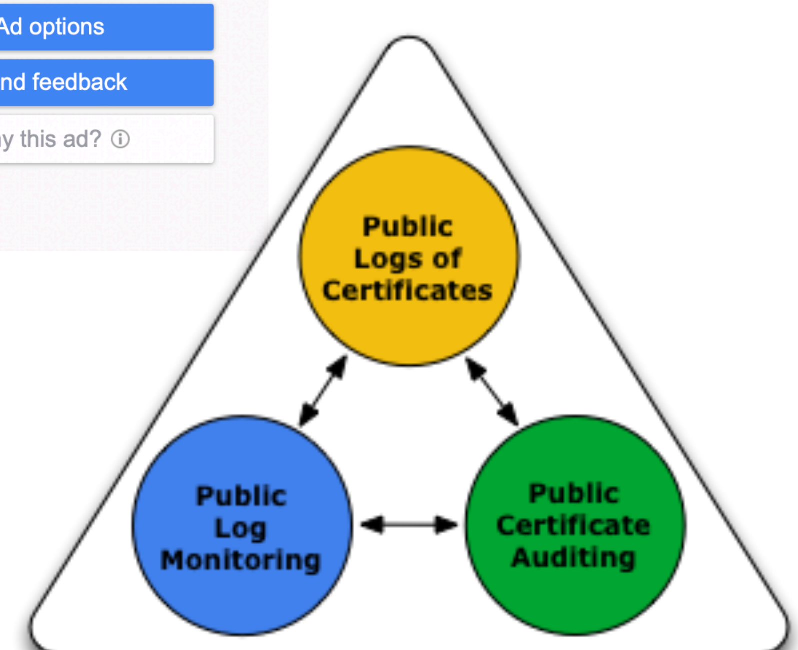
Hacking

This article is more than 12 years old

DigiNotar SSL certificate hack amounts to cyberwar, says expert

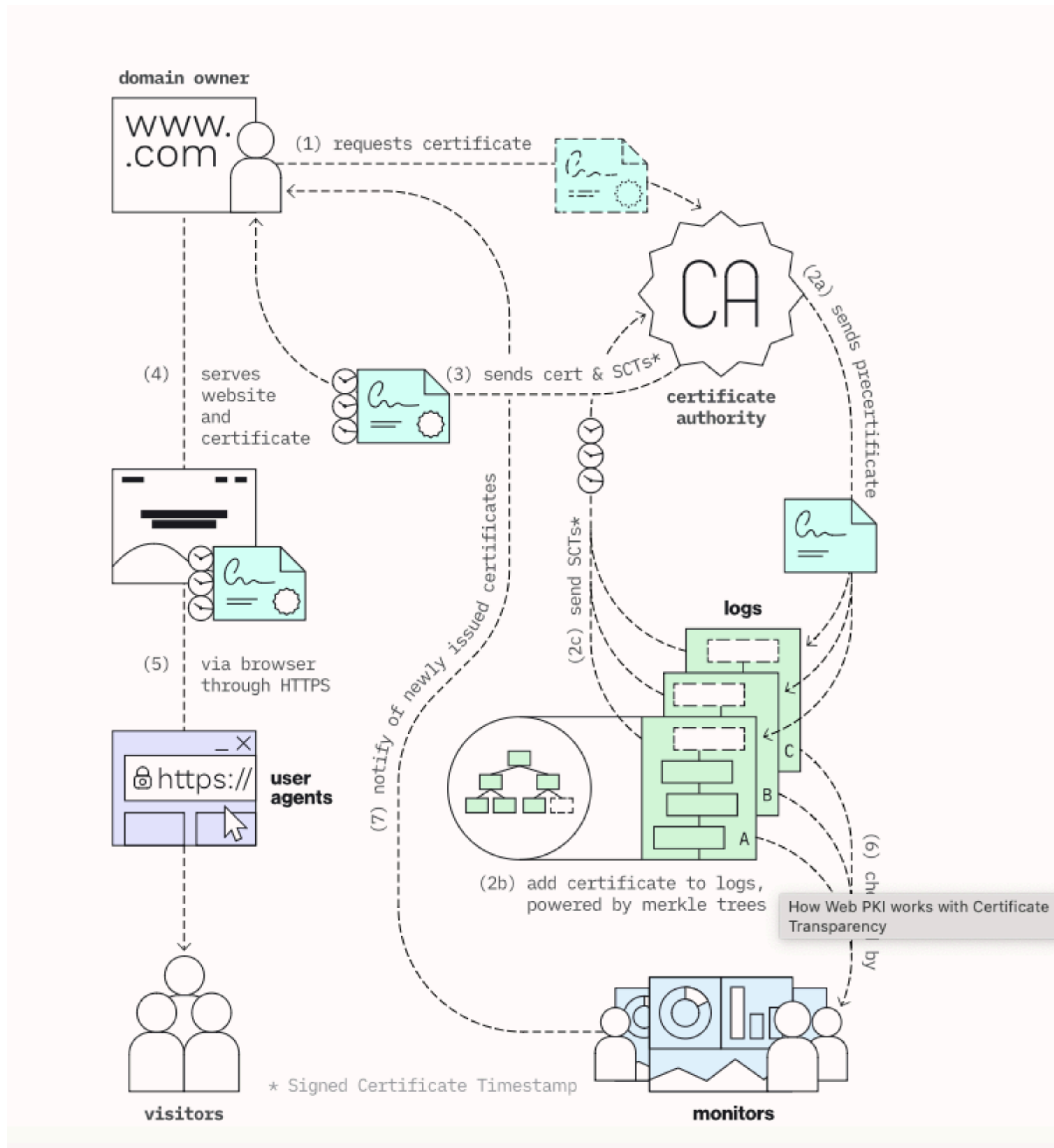
Dutch government revokes certificates used for all its secure online transactions, while CIA, Google, Microsoft and others affected by hack called 'worse than Stuxnet'

Advertisement
Ad served by Google
Ad options
Send feedback
Why this ad?



Certificate Management in Week 7

Certificate Transparency



https://certificate.transparency.dev/

Thank you

Optional proofs

PKCS1 v1.5 signatures

RSA trapdoor permutation: pk = (N,e) , sk = (N,d)

- $Sign(sk, m \in M)$

$$\text{EM} = \underbrace{\boxed{\overset{\text{16 bits}}{01}}\boxed{\text{0xFF 0xFF 0xFF ... 0xFF 0xFF}}\boxed{00}\boxed{\overbrace{\text{H(m)}}^{\text{256 bits}}}}_{\text{RSA modulus size (e.g. 2048 bits)}}$$

output: $\sigma \longleftarrow (\text{EM})^d \bmod N$

- $Ver(vk, m, \sigma)$: verify that $\sigma^e \bmod N$ has the correct format

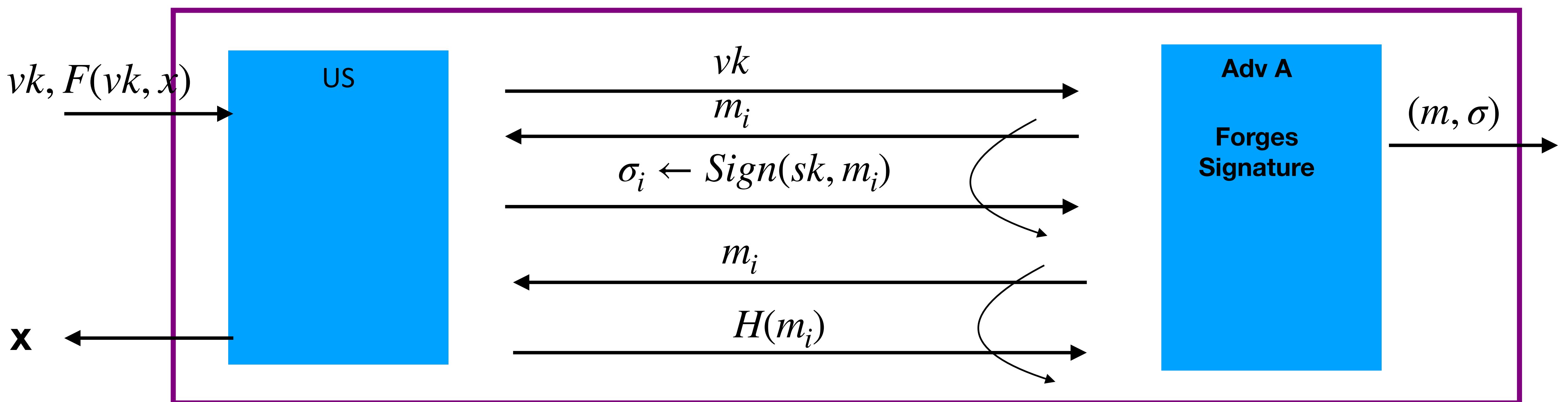
Security: no security analysis, not even with ideal hash functions

Security of RSA-FDH

(G, F, F^{-1}): secure TDP with domain X,

Recall FDH sigs: **Sign(sk, m) = F^{-1}(sk, H(m))** where $H: M \longrightarrow X$

We will show: TDP is secure ⇒ FDH is secure, when H is a random function



How to use forger?
Solution: “we” will know sig. on **all-but-one** of m where adv. queries H(). Hope adversary gives forgery for that single message.

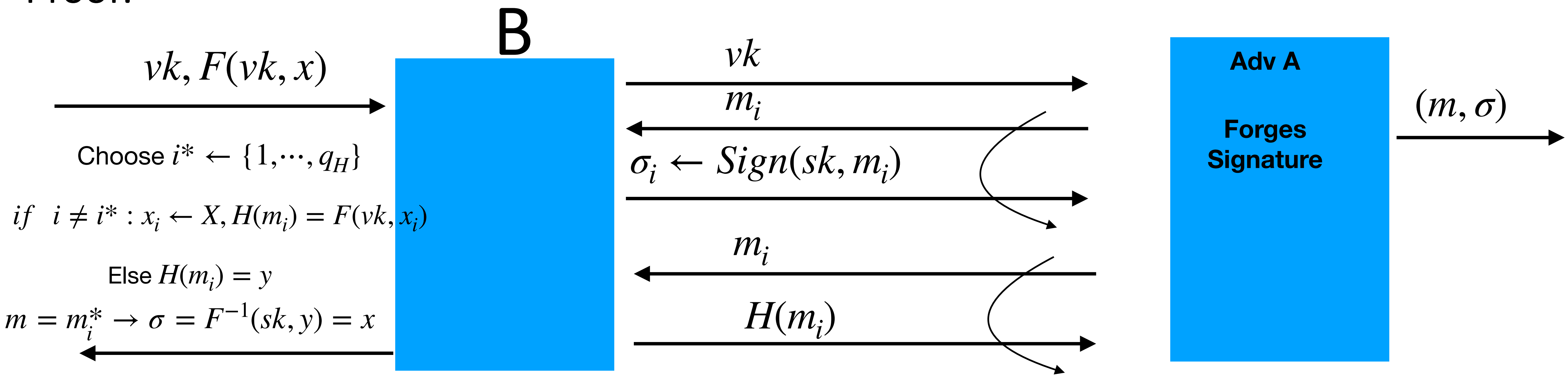
Security Proof

Thm [BR]: (KG_{TDP}, F, F^{-1}) secure TDP $\Rightarrow$ $(KG_{TDP}, Sign, Ver)$ secure signature

when **H:** M $\longrightarrow$ X is modeled as a random oracle.

$\forall$A $\exists$B: $Adv_{SIG}[A,FDH] \leq q_H \cdot Adv_{TDP}[B,F]$

Proof:



$$Pr[m = m_i^*] = 1/q_H$$

Proving security

Thm [BR]: (G_{TDP}, F, F^{-1}) secure TDP $\Rightarrow$ $(KG_{TDP}, \mathbf{Sign}, \mathbf{Ver})$ secure signature

when **H:** $M \longrightarrow X$ is modeled as a random oracle.

$\forall A\ \exists B$: $\quad \overset{(RO)}{} \mathrm{Adv}_{SIG}[A,FDH] \le q_H \cdot \mathrm{Adv}_{TDP}[B,F]$

Proof:

$\Uparrow$

So: $$\underbrace{\mathrm{Adv}_{TDP}[B,F]}_{\text{Prob. B outputs x}} \ge \underbrace{(1/q_H)}_{\Pr[m=m_{i^*}]} \cdot \underbrace{\mathrm{Adv}_{SIG}[A,FDH]}_{\text{Prob. forger A outputs valid forgery}}$$

How B answers queries?

Alg. B has table:

$m_1, \quad x_1 : \quad H(m_1) = F(vk, x_1)$

$m_2, \quad x_2 : \quad H(m_2) = F(vk, x_2)$

$\vdots$

$m_{i*}, \qquad\qquad H(m_{i*}) = y$

$\vdots$

$m_q, \quad x_q : \quad H(m_q) = F(vk, x_q)$