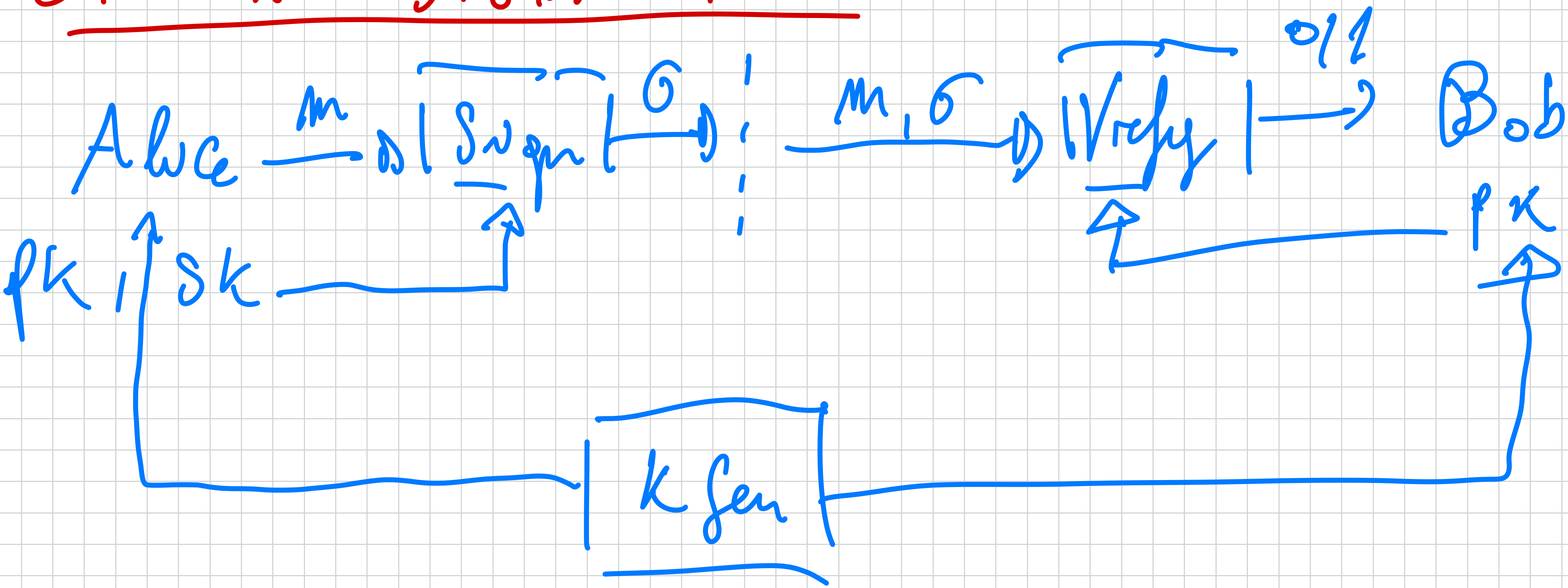
# DIGITAL SIGNATURES




**Def** $\Pi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ is UFCMA if $\forall$ PPT $A$: $\Pr[\mathrm{GAME}^{\mathrm{ufcma}}_{\Pi, A}(\lambda) = 1] \leq \mathrm{negl}$

$\mathcal{A}$ $\qquad\qquad\qquad\qquad$ $\mathcal{C}$

$$\xleftarrow{\quad pk \quad}$$

$\#\text{poly}$

$$\xrightarrow{\quad m \quad}$$

$$\xleftarrow{\quad \sigma \quad}$$

$$\xrightarrow{\quad m^*, \sigma^* \quad}$$

$(pk, sk) \leftarrow KGen(1^\lambda)$

$\sigma = Sign(sk, m)$

Output 1 iff:

$m^* \notin \{m_i\}$

$Vrfy(pk, m^*, \sigma^*) = 1$

Disclaimer: Bob must know pk is the public key of Alice.

Remember the DH key exchange?

$$\xrightarrow{\quad g^x \quad}$$

$$\xleftarrow{\quad g^y \quad}$$

To avoid MIM attacks parties must sign the protocol messages.

The solution to certify public keys is the so-called PKI.

$$g^x, \ \mathrm{Sign}_{sk}(g^x), (\mathrm{Cert}_{pk}, pk)$$

What is $\mathrm{Cert}_{pk}$? Just a signature on pk! Under which key??? It looks like a circular problem...

Trust assumption: There is a so-called CA, that is in charge to certify pks.

A moron     pk, I'm AMAZON     CA

$sk_{CA}, pk_{CA}$

$\mathrm{cert}_{pk}$

$$cert_{pk} = Sign(sk_{CA}, pk \| A\ \text{maron})$$

"X.509 standard"

The public key $pk_{CA}$ is hard-wired in the browser. In practice, there are many CA's. But this is just an approximation. From now on, we just assume $pk$ is authentic.

Two constructions:

1) FDH - Full Domain Hash $\approx$ how to sign with any TDP (RSA).

2) Fiat-Shamir signatures $\approx$ signatures from IDENTIFICATION SCHEMES. Many instantiations (DL, RSA, but even post-quantum ... )

Made with Goodnotes

1) FDH. The basic idea is:

$KGen(1^\lambda) \longrightarrow (pk, sk)$ $\big(pk = (N, e)\,;\, sk = (d, N)\big)$

$Sign(sk, m) = f^{-1}_{sk}(m)$ $\big(\sigma = m^d \bmod N\big)$

$Vrfy(pk, m, \sigma): f_{pk}(\sigma) \overset{?}{=} m \longrightarrow$ If YES output 1. Else, 0.

$\big(\sigma^e \equiv (m^d)^e \equiv m \bmod N$ by EULER.$\big)$

But not UF-CMA! Why?

1. $A \quad \underset{\longleftarrow\ \sigma}{\xrightarrow{\ m\ }} \quad \sigma = m^d$

2. $A \xrightarrow{\sigma}$

$\xleftarrow{\sigma^d = m^{e \cdot d}}$ Not sure...

let's just take any $\sigma^*$. Then, let

$$m^* = f_{pk}(\sigma^*)$$

$$(m^* = (\sigma^*)^e \bmod n).$$

Output $(m^*, \sigma^*)$.

Can you forge a chosen message $m^*$ (with RSA)? Exercise. Just use the fact

That RSA is homomorphic:

$(m_1, \sigma_1)$ ; $(m_2, \sigma_2)$

$\sigma_1 \cdot \sigma_2$ is a signature on $m_1 \cdot m_2$.

<u>FDH</u>: Kill the attack by first hashing m and then apply the TDP.

$$\sigma = f^{-1}_{SK}(H(m)) \qquad : \text{Sign}$$

$$f_{PK}(\sigma) \stackrel{?}{=} H(m) \qquad : \text{Verify}$$

As a bonus: It also works for VIL messages.

Can we prove it UF-CMA? Yes. Under what assumptions? Ideally: TDP + CRH. We don't know how to do this.

( **Remark**: If Sign is a secure UF-CMA signature on $\{0,1\}^n$, then assuming H is a CRH then

$$\mathrm{Sign}_{sk}(H(m)) \text{ is also UF CMA.})$$

We will give the simplest proof, under a strong assumption on $H$: $H$ is a RANDOM ORACLE. Basically $H$ corresponds to a truly random table, and the only way to evaluate it on $x$ is to ask an oracle to give $H(x)$.

(Actually, we can prove it secure in the standard model, no RANDOM ORACLES, using strong tools such as OBFUSCATION.)

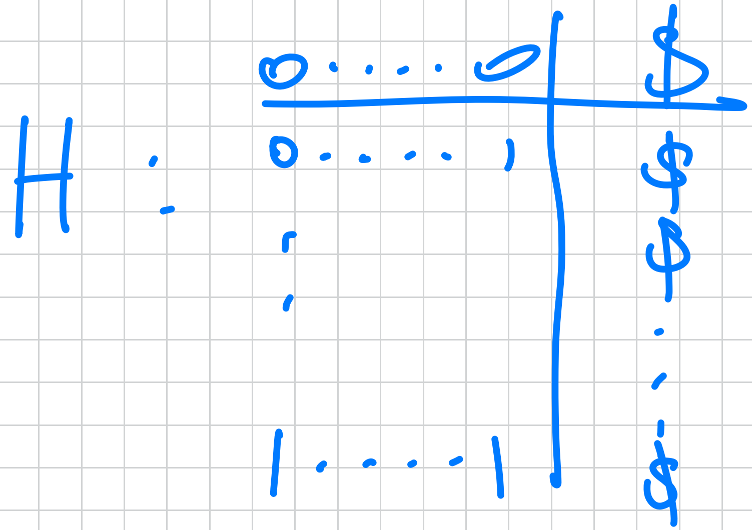
**THM.** FDH is UF-CMA in the

ROM (RANDOM ORACLE MODEL)

assuming $(f, f^{-1})$ is a TDP.

OBIS : Z6IGXXE6

Proof. ROM: We assume all parties including the adversary can ask RO queries:

Some conventions: $A$ asks $q_s$ signature queries $m_1, \dots, m_{q_s}$ and $q_h$ RO queries. Of course, $q_s, q_h = poly(\lambda)$.

WLOG, assume that queries are not repeated. Before asking for a signature on $m_i$ or forging on $m^*$, $A$ makes a RO query with $m_i$ or $m^*$. Adding these queries does not decrease $A$'s prob. of success.

Assume $\exists$ PPT $A$ as above in the UF-CMA that succeeds w.p. $\varepsilon(\lambda) \geq 1/poly(\lambda)$

Build a PPT $\mathcal{B}$ breaking the TDP.

$\mathcal{A}$ $\qquad\qquad$ $\mathcal{B}$ $\qquad\qquad$ $\mathcal{C}$

$\mathcal{B} \xleftarrow{pk,\, y} \mathcal{C}$

$\mathcal{A} \xleftarrow{pk} \mathcal{B}$

$\mathcal{C}$: $pk, sk$; $x \leftarrow_\$ \mathcal{X}$; $y = f_{pk}(x)$

$$\left( \begin{array}{l} pk = (N, e) \\ sk = (N, d) \\ y = x^e \bmod N \end{array} \right)$$

↳ RSA

$\mathcal{B}$: Sample $i \leftarrow [q_h]$ (0)

(1) $\mathcal{A} \xrightarrow{m_i} \mathcal{B}$, $\mathcal{A} \xleftarrow{y_i} \mathcal{B}$

(2) $\mathcal{A} \xrightarrow{m_i} \mathcal{B}$, $\mathcal{A} \xleftarrow{\sigma_i} \mathcal{B}$

(3) $\mathcal{A} \xrightarrow{m^*,\, \sigma^*} \mathcal{B} \xrightarrow{???} \mathcal{C}$

Trick (only possible in the ROM):
The reduction can simulate the output of RO queries arbitrarily, so long as it looks like a random table to A.
In the above picture:

⓪ Think of $j$ as the index corresponding to the RO query $m^*$.

① Upon RO query $m_i$:
- If $i \neq j$, pick $x_i \leftarrow \mathcal{X}$ and return $y_i = f_{pk}(x_i)$. $(H(m_i) \stackrel{!}{=} y_i)$

$(y_i = x_i^e \bmod n.)$

- If $i = j$, return $y$

② Upon signature query $m_i$, return $\sigma_i = x_i$ to $A$, unless $m_i = m_j$ in which case ABORT.

③ Upon $m^*, \sigma^*$, if $m_j = m^*$ output $x = \sigma^*$.

Analysis:

- The pk is perfectly simulated.
- Simulation of RO queries is also

good, because $y_i$ is RANDOM and also $y$ is RANDOM

- Assuming $\mathcal{B}$ never aborts, the signatures are perfectly simulated.

Indeed: $\text{Vrfy}(pk, m_i, \sigma_i)$:

$$f_{pk}(\sigma_i) = f_{pk}(x_i) = y_i = H(m_i)$$

as $x_i$ is the pre-image of $y_i$

- Assuming $\mathcal{B}$ does not abort, for the same reason $x = \sigma^*$ is the pre-image of $y$.

Finally:

$$\Pr[\mathcal{B} \text{ wins}] \geq \Pr[\mathcal{A} \text{ wins} \wedge m^* = m_i]$$

$$\geq \frac{1}{poly} \cdot \varepsilon(\lambda) =$$

$$= \frac{1}{poly} \cdot 1/poly = 1/poly \quad \blacksquare$$