Hash-Based Signatures

yes, there is a reading for Monday

Previously:

full-domain hash RSA: $\text{Sign}(m) = H(m)^d \bmod pq$

Rabin signatures: $\text{Sign}(m) =$ compute sqrt of $m \| r \bmod pq$

Question:

can we construct signature schemes without wild number theory stuff?

One-time Signatures

= Attacker only sees signature of 1 msg

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

Adv gets vk

Adv choose m

Adv gets $\text{Sig}(sk, m)$

Adv tries to generate forgery: (m', σ') : $\text{Ver}(vk, m', \sigma') = 1$ AND $m' \neq m$

Lamport Signature:

message space is $\{0,1\}^n$

need a function f that is hard to invert

(e.g. target-collision resistant hash)

(given $f(x)$, hard to find x' : $f(x') = f(x)$)

KeyGen:

for $i = 1$ to n

  for $b = 0$ to 1

    $sk[i,b] \leftarrow \{0,1\}^\lambda$

    $vk[i,b] = f(sk[i,b])$

$sk = (sk[1,0], \dots)$

$vk = (vk[1,0], \dots)$

$sk[1,0]$	$sk[2,0]$	$\dots$	$sk[n,0]$
$sk[1,1]$	$sk[2,1]$	$\dots$	$sk[n,1]$

f $\quad$ f $\quad$ f

$vk[1,0]$	$vk[2,0]$	$\dots$	$vk[n,0]$
$vk[1,1]$	$vk[2,1]$	$\dots$	$vk[n,1]$

$\mathrm{Sign}(sk, m \in \{0,1\}^n)$:

for $i = 1$ to n

  reveal $sk[i, m_i]$

  i^{th} bit of m

$m =$	0	1	$\dots$	1
	$sk[1,0]$	$sk[2,0]$	$\dots$	$sk[n,0]$
	$sk[1,1]$	$sk[2,1]$	$\dots$	$sk[n,1]$

$\mathrm{Ver}(vk, m, \sigma)$:

check for $i = 1$ to n:

  does $f(\sigma_i) = vk[i, m_i]$?

If yes to all, return 1

Adv sees Sign(sk, m), tries to forge Sig on $m' \neq m$

$m = \quad 0 \quad 1 \quad \cdots \quad 1$

sk[1,0]	sk[2,0]	...	sk[n,0]
sk[1,1]	sk[2,1]	...	sk[n,1]

$m' = \quad 0 \quad 0 \quad \cdots \quad 1$

↑ would have to find sk[2,0] or any other preimage of vk[2,0]

Winternitz Signature:

message space = $\{1, \ldots, n\}$

KeyGen:

$A_0, B_0 \leftarrow \{0,1\}^\lambda$

$sk = (A_0, B_0)$

$vk = (A_n, B_n)$

sk $\qquad\qquad\qquad\qquad$ vk

$$A_0 \xrightarrow{f} A_1 \xrightarrow{f} A_2 \xrightarrow{f} \cdots \xrightarrow{f} A_n$$

$$B_0 \xrightarrow{f} B_1 \xrightarrow{f} B_2 \xrightarrow{f} \cdots \xrightarrow{f} B_n$$

Sign(sk, $m \in \{1, \ldots, n\}$):

give A_{n-m}

and B_{m-1}

$m = 2$ $\qquad\qquad\qquad\qquad$ A_{n-2}

$$A_0 \xrightarrow{f} A_1 \xrightarrow{f} A_2 \xrightarrow{f} \cdots \xrightarrow{f} A_n$$

$$B_0 \xrightarrow{f} B_1 \xrightarrow{f} B_2 \xrightarrow{f} \cdots \xrightarrow{f} B_n$$

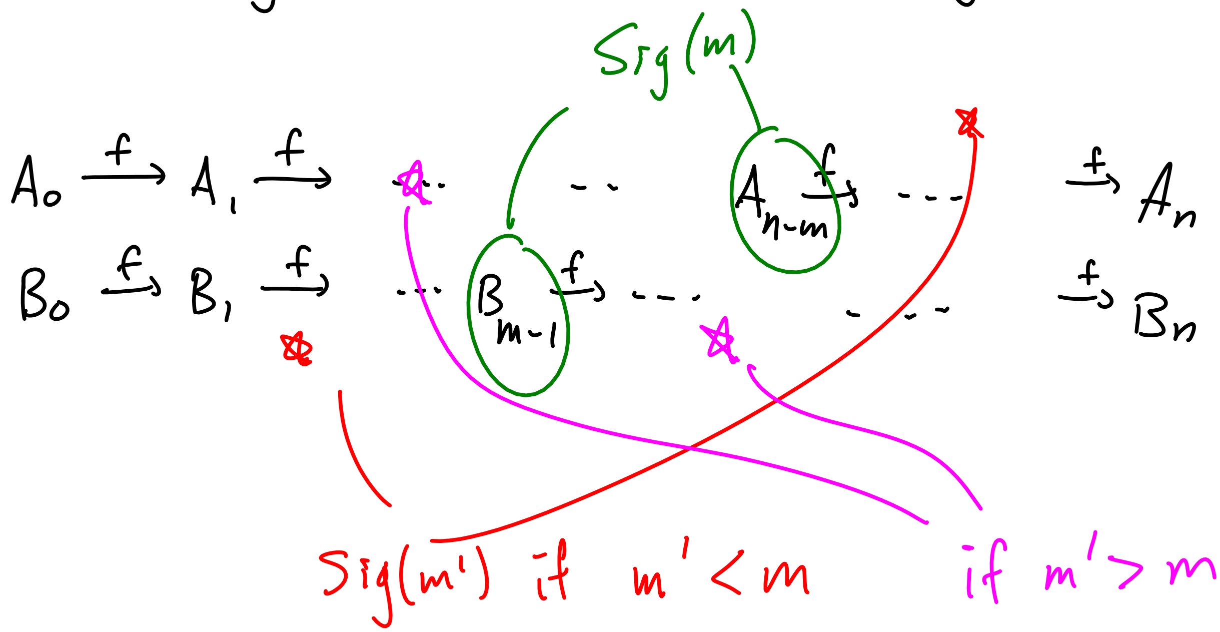
$\underline{Ver(vk, m, (A^*, B^*))}$:

check $\underbrace{f(f(f}_{m \text{ times}}(\cdots A^*))) = A_n$

$\underbrace{f(f(}_{n-m+1 \text{ times}}\cdots B^*))) = B_n$

forgery:

Adv sees Sig(m), tries to forge Sig(m')



either way, must "reverse" one of the 2 CHAINZ

Comparision:

Lamport: faster verification (fewer calls to f)

Winternitz: smaller keys

Full-fledged signature:

consider message space of $\{0,1\}^2$

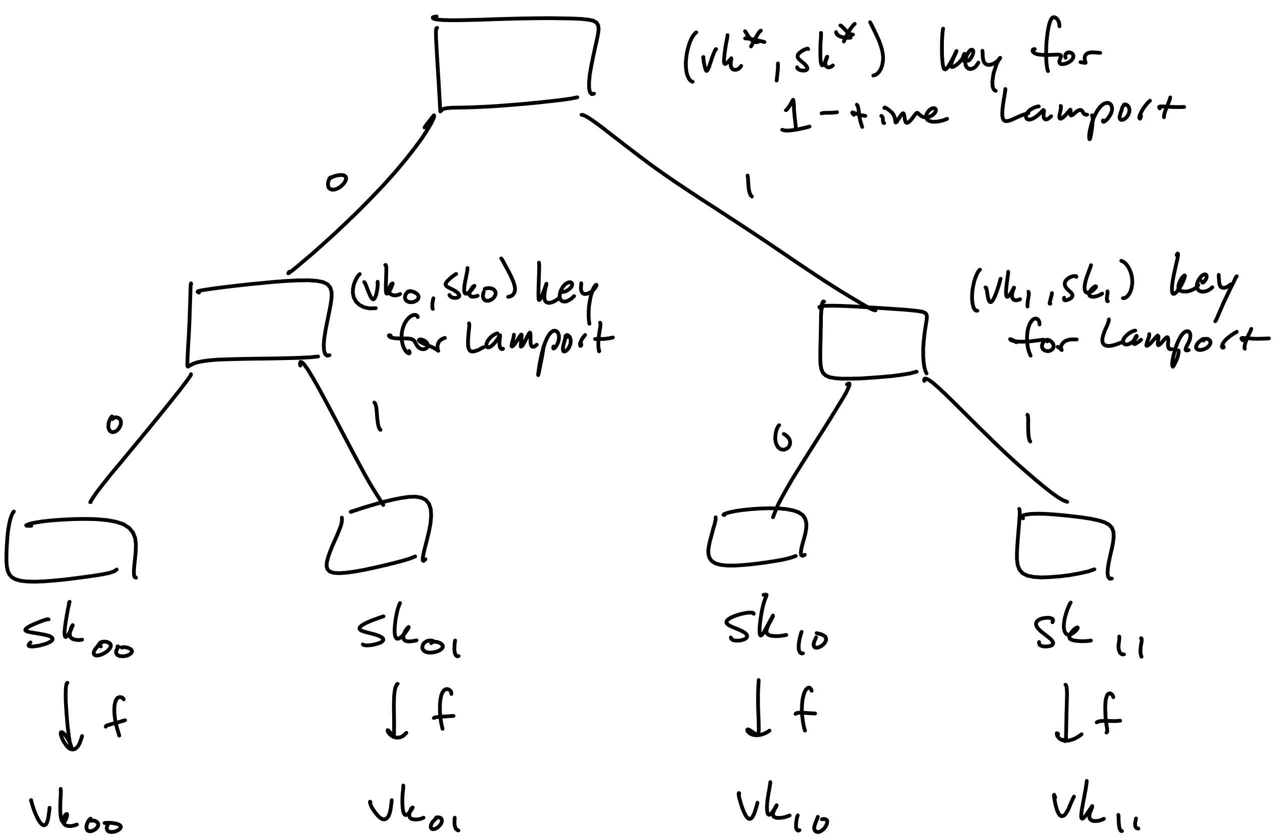
Dumb attempt: generalize Lamport

Signature of $m = 10$

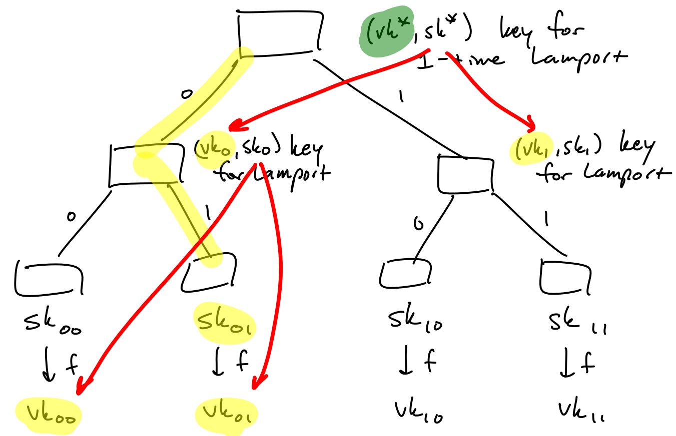
$$\begin{array}{ccccc} sk_{00} & sk_{01} & sk_{10} & sk_{11} & = sk \\ \downarrow f & \downarrow f & \downarrow f & \downarrow f & \\ vk_{00} & vk_{01} & vk_{10} & vk_{11} & = vk \end{array}$$

But size of keys is exponential in msg length

Better attempt:



give out ONLY vk^*



Sign$(m = 01)$ contains

- (vk_0, vk_1)
- Signature of $vk_0 \| vk_1$ under sk^*
- vk_{00}, vk_{01}
- signature of $vk_{00} \| vk_{01}$ under sk_0
- sk_{01}

Note: sk^* used only to sign $vk_0 \| vk_1$

sk_0 used only once, to sign $vk_{00} \| vk_{01}$

Note: Signer doesn't have to generate all key pairs up-front

Can generate on-demand using PRF for randomness of each key pair