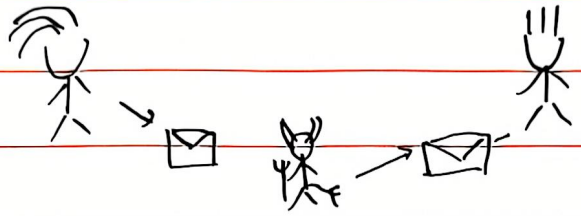
年　　月　　日

Digital Signature



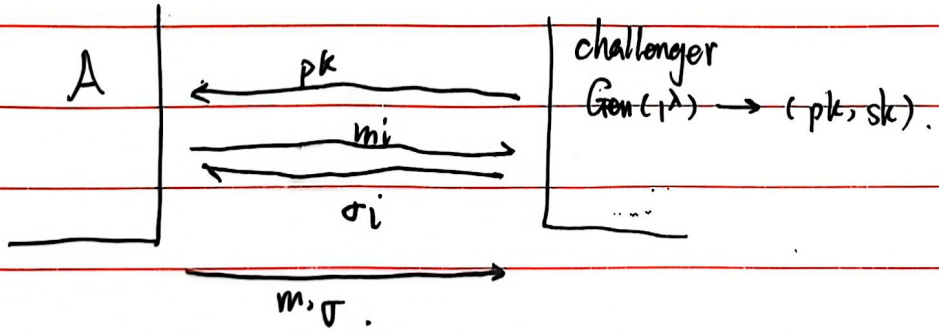
$\Pi = (Gen, Sign, Verify)$

$Gen(1^\lambda) \rightarrow (pk, sk)$.

$Sign(sk, m) \rightarrow \sigma$

$Verify(pk, m, \sigma) \rightarrow Yes / No$.

Security Definition

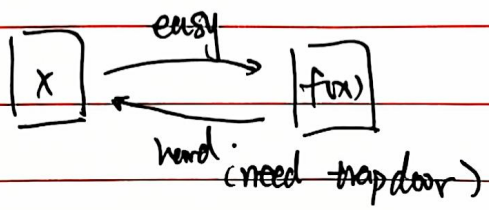


A wins iff $\begin{cases} Verify(pk, m, \sigma) = Yes. \\ m \notin \{m_i\}. \end{cases}$

$(m, \sigma) \notin \{(m_i, \sigma_i)\}$ (strong version).

年　月　日

Trapdoor Permutation.



Candidate: plain RSA.

sample $N=pq$, d,e s.t. $de \equiv 1 \pmod{\varphi(N)}$.

$pk = (N, e)$, $sk = d$.

$Sign(sk, m) = m^d$

$Verify(pk, m, \sigma)$ check if $m = \sigma^e$.

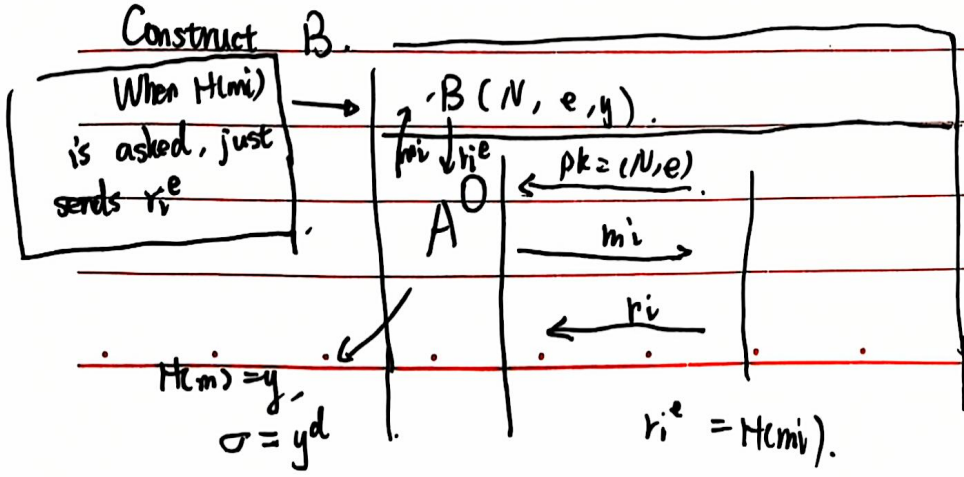
Not secure! $Sign(m_1)\ Sign(m_2) \rightarrow Sign(m_1 m_2)$.

Modified: $Sign(sk, m) = H(m)^d$. (e.g. H is a random oracle.)

Assume A with Oracle. O attacks it. Idea: B can simulate O itself.

Construct B.

When H(mi) is asked, just sends r_i^e

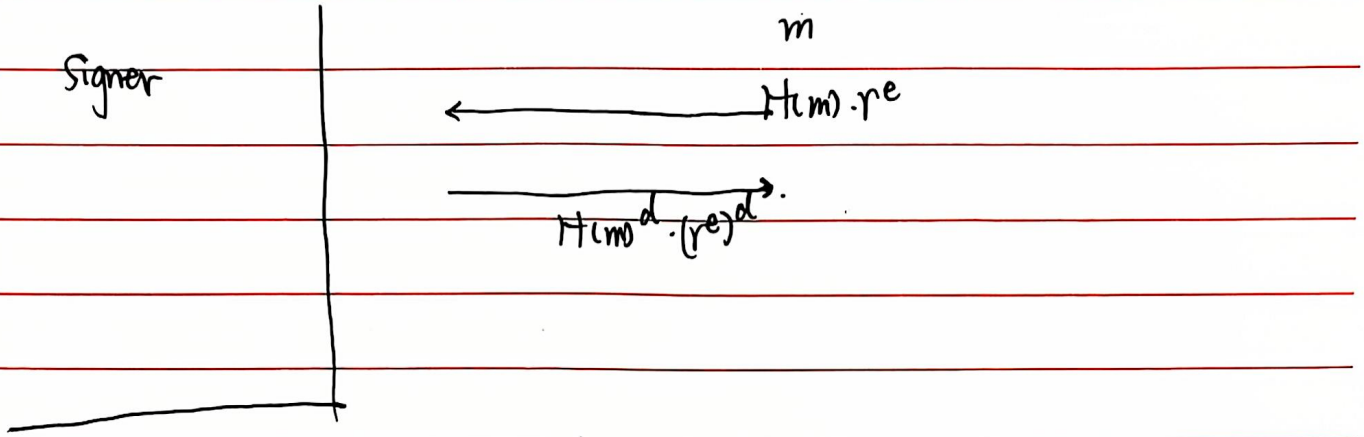


① Intuitively we gain $\sigma^e = y$ from scratch.

② To formalize this, force A ask H(m), and randomly choose $r_i = y$.

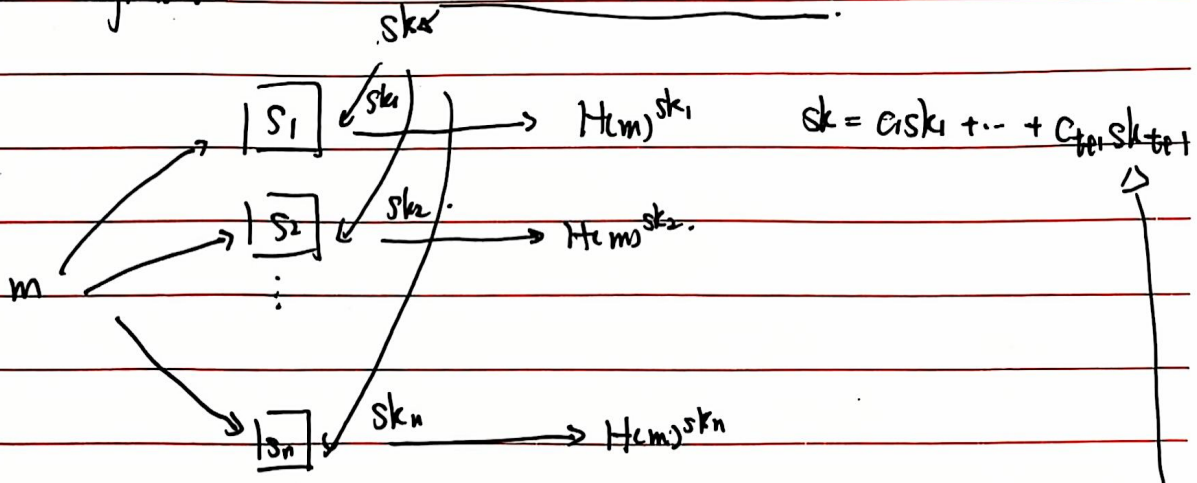
年　月　日

Blind Signature.



Threshold Signature.

Secret Sharing to distribute



Secret Sharing.

Share (s) = (s1, ..., sn).

↑ Secret　↑ Shares

deg-t

Random f s.t. f(0) = s.

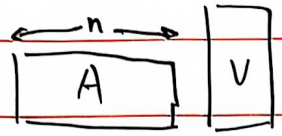
Let s(i) = f(i).

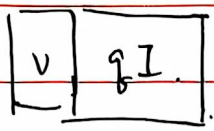
Refresh f if ≥ t+1 Si.

f(0) has linear relations with $s_{i_1}, \cdots s_{i_{t+1}}$

年　　月　　日

LWE-based.





$\{\vec{x} \in \mathbb{Z}^n \mid A\vec{x} \equiv 0 \pmod q\}$. spanned by = span(S). (on $\mathbb{Z}$).

pk: A

sk: S. — small elements. full rank. s.t. $AS = 0$.

$qI \subseteq \text{span}(S)$ (on $\mathbb{Z}$).

First sample S., (can prove A is close to uniform distribution).

$f(\vec{x}) = A\vec{x}$

$\vec{x} \xrightarrow{\text{easy}} A\vec{x} \pmod q$

$A\vec{x} \pmod q \xrightarrow{\text{easy}}$ some $\vec{x}$

$A\vec{x} \pmod q \xrightarrow{\text{hard}}$ small $\vec{x}$

Given $\vec{y}$, 1) find $\vec{t}$, s.t. $A\vec{t} = \vec{y} \pmod q$.

2). find small $\vec{x} = \vec{t} + \text{Lattice}(S)$

thus $\vec{y} = A\vec{x} = A\vec{t} \pmod q$.

With S this can be easy

年　　月　　日

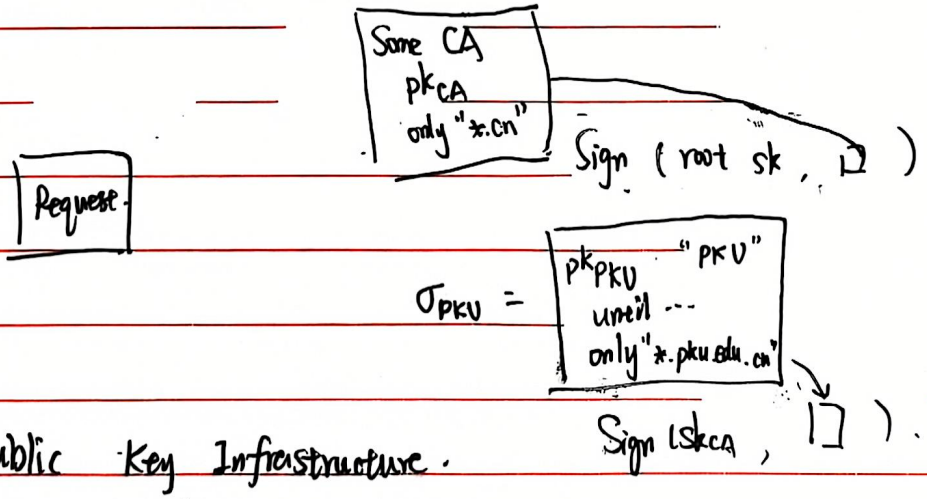
Sign(m): short $\vec{x}$ s.t. $A\vec{x} = H(m)$, (H is modeled as random oracle)

$\vec{x} = \sigma_m$.

Pairing ⟹ IBE.

pk	pk	$g, G, g^s, H, g_T, G_T, e$.
sk	msk	s.
m	ID.	
σ	~~sk_{ID}~~.	$H(ID)^s$.

Verify: $e(H(m), g^s) \stackrel{?}{=} e(\sigma = H(m)^s, g)$.



Public Key Infrastructure.

Key released: CA broadcasts.

年　月　日

Hash-based Signature. Lamport Signature. (Quantum-resistant).

Sign 1 bit:

pk: $H(s_0)$, $H(s_1)$.

sk: $s_0, s_1 \in \{0,1\}^\lambda$

Sign(b) = s_b.

Sign n bits: Repeat n times

pk: $H(s_{i,b})$, $i \in [n]$, $b \in \{0,1\}$.

sk: $s_{i,b}$.

Sign 2 n-bit, stateful: Repeat 2 times. $(pk_1, sk_1; pk_2, sk_2)$.

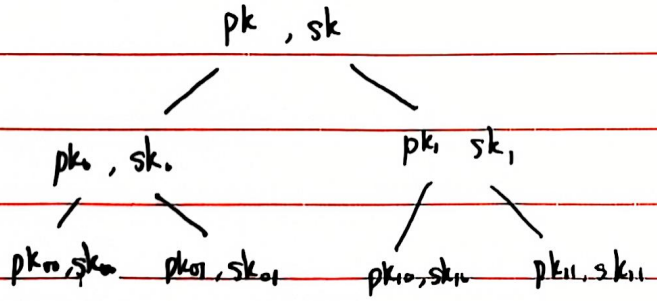
Sign arbitrary-number of λ-bit msg, stateful.

pk, sk, pk_1, sk_1

$\sigma_1 = (Sign(sk, m_1 || sk_1), pk_1)$.

$\sigma_2 = (Sign(sk_1, m_2 || sk_2), pk_2, \sigma_1, m_1)$.

Can use hash to solve the problem of length.



Use (pk, sk) to sign its children, only need the information in the path.

年　　月　　日

Use decisional methods to generate each key.

choose a seed of a PRF

(May contain repay).

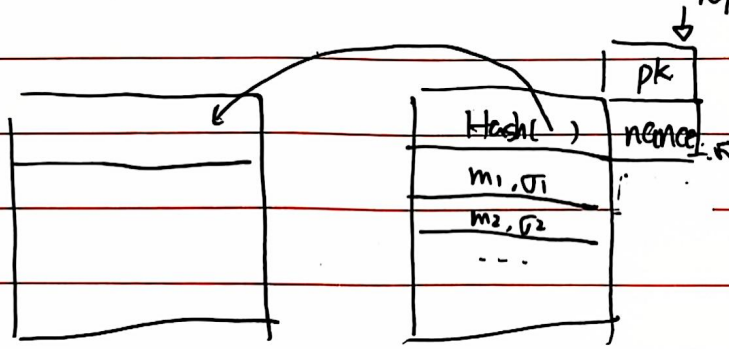
$ID_0 (= pk_0)$

$ID_1 (= pk_1)$

$m = \text{Trans } 2\$ \text{ from } pk_0 \text{ to } pk_1.$

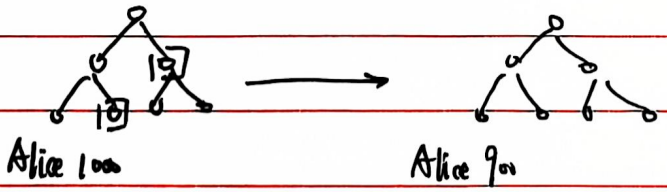
modified → $m = (pk_0, \Delta_0, pk_1, \Delta_1, pk_2, \Delta_2 \cdots)$

$\sigma = \text{Sign}(sk_0, m)$



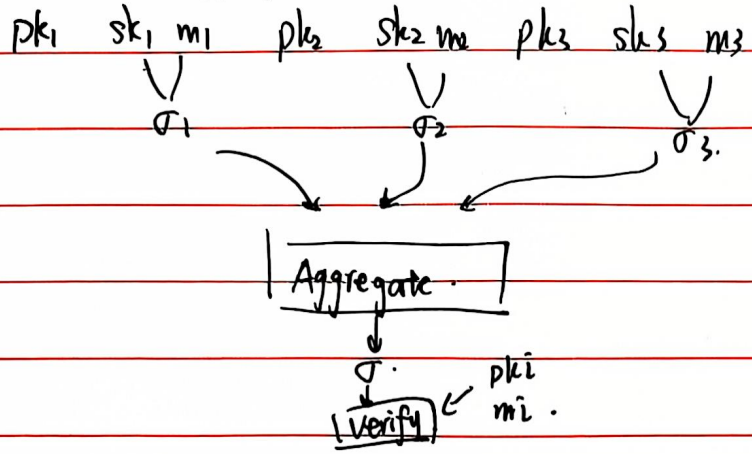
Choose the "Legest chain" when there are branchs

Merkel Tree.



年　　月　　日

Signature Aggregation



$H(m_1)^{s_1} \cdot H(m_2)^{s_2} \cdot H(m_3)^{s_3}$.

Other ideas about bitcoin:

- Need huge memory.
- Need many bitcoins.