Cryptographic Protocols

Chapter 3

Blind Digital Signatures

3.1 General

- Protocol between a user $\mathbb{A}$ and signer $\mathbb{B}$ (with a signature scheme)
- $\mathbb{A}$ inputs a message
- $\mathbb{A}$ obtains a signature of $\mathbb{B}$ on m
- $\mathbb{B}$ will not learn message he signs and not see any association between information in the protocol and a signature seen later

- blind signature must be unforgeable as ordinary dig.sig.
- anyone can verify signature using pk
- $\mathbb{B}$ learns nothing about messages he signs, except for total count

3.2 Blind Signatures for RSA

$\textsc{KeyGen}()$ as in RSA

$$
\begin{array}{lcl}
\mathbb{A}(m \in \{0,1\})^{\star} & & \mathbb{B}(sk) \\
\quad r \leftarrow \mathbb{Z}_N & & \\
\quad \overline{h} \leftarrow \mathbb{H}(m) \cdot r^{-e} \ \mathit{mod}\ N & & \\
 & \xrightarrow{\overline{h}} & \\
 & & \overline{s} \leftarrow \overline{h}^{d} \ \mathit{mod}\ N) \\
 & \xleftarrow{\overline{s}} & \\
\quad s \leftarrow \overline{s}/r & & \\
\quad //\ s \text{ is Rsa Signature on } m & &
\end{array}
$$

$$
\begin{array}{l}
\textsc{Verifier} \\
s^e \stackrel{?}{\equiv} \mathbb{H}(m)
\end{array}
$$

3.2.1 Completeness

$$
\begin{aligned}
s &\equiv \overline{s}^e \cdot r^{-e} \\
&\equiv (\overline{h}^{de}) \cdot r^{-e} \\
&\equiv (\overline{h}) \cdot r^{-e}
\end{aligned}
$$

3.2.2 Blindness

- B signs a random

3.3 Blind Schnorr Signature Scheme

hello

3.3.1 Signing Protocol

$$
\begin{array}{lcl}
\text{User } \mathbb{A}(m \in \{0,1\})^\star & & \text{Signer } \mathbb{B}(pk(y = g^x), sk) \\
 & & \quad r \leftarrow \mathbb{Z}_q \\
 & & \quad t \leftarrow g^r \\
 & \xleftarrow{t} & \\
\quad \alpha, \beta \leftarrow \mathbb{Z}_q & & \\
\quad \overline{t} \leftarrow t \cdot g^{-\alpha} \cdot y^{-\beta} & & \\
\quad \overline{c} \leftarrow \mathbb{H}(m \| \overline{t}) & & \\
\quad c \leftarrow \overline{c} + \beta & & \\
 & \xrightarrow{c} & \\
 & & \quad s \leftarrow r - c \cdot x \\
 & \xleftarrow{s} & \\
\quad \overline{s} \leftarrow s - \alpha & & \\
\quad \text{return } (\overline{c}, \overline{s}) & &
\end{array}
$$

Verification as in the ordinary scheme: $\begin{array}{l} \textsc{Verify}(pk, m, (\overline{c}, \overline{s})) \\ \quad \text{return } (\overline{c} \stackrel{?}{=} \mathbb{H}(m \| g^{\overline{s} \cdot y^{\overline{c}}})) \end{array}$

3.3.2 Completeness

$$
\begin{aligned}
\hat{t} &= g^{\overline{s}} \cdot y^{\overline{c}} \\
&= g^{s-\alpha} g^{x \cdot \overline{c}} \\
&= g^{r-cx-\alpha+x\overline{c}} \\
&= g^{r+x(\overline{c}-c)-\alpha} \\
&= t \cdot g^{x \cdot -\beta} \cdot g^{-\alpha} \\
&= t \cdot y^{-\beta} \cdot g^{-\alpha} \\
&= \overline{t}
\end{aligned}
$$

Signature is ordinary Schnorr Signature.

3.3.3 Blindness

- $\mathbb{B}$ sees only $\mathbb{H}(m\|...)$
- $\mathbb{B}$ sees $\mathbb{H}(m\|...) + \beta$, where β is a random blinding factor
- $\mathbb{B}$ sees $(\overline{c}, \overline{s})$, where:

$$\overline{c} = c - \beta$$

$$\overline{s} = s - \alpha$$

- Signature is unlinkable with signing protocol.

3.4 Anonymous Digital Cash (Chaum, 1985)

User $\mathbb{A}$: wallet
Shop $\mathbb{S}$: exchanges service for payment
Bank $\mathbb{B}$: creates coins, stores balance for $\mathbb{A}$ and $\mathbb{S}$

3.4.1 Security Goals

Completeness

If $\mathbb{A}$ withdraws a coin from $\mathbb{B}$, then $\mathbb{B}$ debits it from balance of $\mathbb{A}$.
If $\mathbb{A}$ transfers this coin to $\mathbb{S}$, then $\mathbb{B}$ will credit coin to balance of $\mathbb{S}$

Security

$\mathbb{B}$ does not credit a coin to $\mathbb{S}$ unless $\mathbb{B}$ has issued the coin to some user $\mathbb{X}$ and user $\mathbb{X}$ has transferred coin to $\mathbb{S}$.

Anonymity

If $\mathbb{B}$ credits a coin to some $\mathbb{Y}$, then $\mathbb{B}$ cannot link this coin to any withdrawal by a user.

3.4.2 Protocol to withdraw (issue) coin

User $\mathbb{A}$

$m \leftarrow \{0,1\}^\star$

$\overline{m} \leftarrow$ blinded m

send message (BLING-SIG, $\overline{m}$, m) to $\mathbb{B}$ and run blind sig. protocol

wait for (SIG, $\overline{m}$, $\overline{\sigma}$) message from $\mathbb{B}$

$\sigma \leftarrow$ unblind σ

store(u, m, σ)

Bank $\mathbb{B}$

upon receiving msg (BLING-SIG, $\overline{m}$, m) from $\mathbb{A}$

$bal_A \leftarrow bal_A - u$

run blind sig. protocol with $\mathbb{A}$

send message (SIG, $\overline{m}$, $\overline{\sigma}$) to $\mathbb{A}$

3.4.3 Protocol to withdraw (issue) coin

- User $\mathbb{A}$ spends coin to $\mathbb{S}$
- Each coin can only be spent once

User $\mathbb{A}$

send message (SPEND, u, m, σ) to $\mathbb{S}$

wait for (ACK) or (NACK message from $\mathbb{S}$

Shop $\mathbb{S}$

upon receiving msg (SPEND, u, m, σ) from $\mathbb{A}$

send message (DEPOSIT, u, m, σ) to $\mathbb{B}$

wait for msg (RESULT,m, b) from $\mathbb{B}$

if $b =$ TRUE

  deliver goods or service to $\mathbb{A}$ and send msg (ACK) to $\mathbb{A}$

else

  send msg (NACK) to $\mathbb{A}$

Bank $\mathbb{B}$

upon receiving msg (DEPOSIT, u, m, σ)

if VERIFY(pk, u, m, σ) and $m \notin \mathbb{M}$

  $\mathbb{M} \leftarrow \mathbb{M} \cup \{m\}$

  $bal_S \leftarrow bal_S + u$

  send msg. (RESULT,m, TRUE)

else

  send msg. (RESULT,m, FALSE)