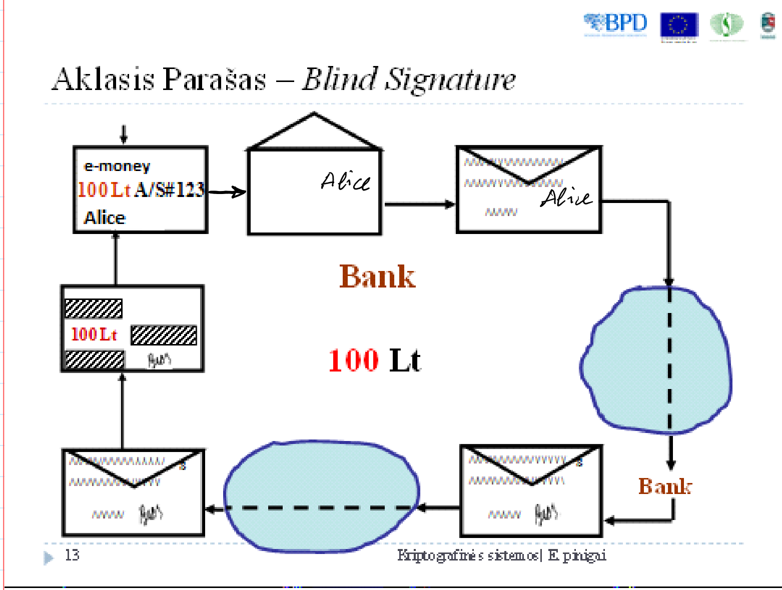


Chaum e-money system
e-coin

```
>> e=2^16+1
e = 65537
>> isprime(e)
ans = 1
```

RSA cryptosystem

B: $p, q \leftarrow$ genprime

$n = p \cdot q$

$\phi = (p-1)\cdot(q-1)$ $\quad PuK = (n, e)$

$e = 2^{16}+1$

$d = e^{-1} \bmod \phi$ $\Rightarrow$ $e\,d = 1 \bmod \phi$ $\quad PrK = d$

If $e = 2^{16}+1$ – it is prime

1) $1 < e < \phi$

2) $\gcd(e, \phi) = 1$ since e is prime

$>> d = mulinv(e, fy)$ % $fy = \phi$

Since numbers e and d are presented in exponent, then exponent value is computed mod ϕ according to Euler theorem:

If $\gcd(z, n) = 1 \Rightarrow z^{\phi} \bmod n = 1$

Any computations performed in the exponent are computed mod ϕ: $\quad z^{e\cdot d} \bmod n = z^{e\cdot d \bmod \phi} \bmod n = z^1 \bmod n = z$

if $z < n$

RSA signature creation:

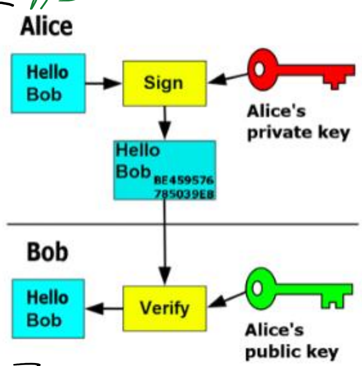
On message M encoded by decimal number $m < n$.

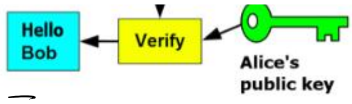
$Sign(PrK = d, m) = \sigma = m^d \bmod n.$

RSA signature verification:

$Ver(PuK = (e, n), \sigma) = \sigma^e \bmod n = m.$

Correctness: $\sigma^e \bmod n = (m^d)^e \bmod n = m^{de \bmod \phi} \bmod n =$ ... $de \bmod \phi = 1$





$Ver(PuK=(e,n), \sigma) = \sigma^e \bmod n = m.$

Correctness: $\sigma^e \bmod n = (m^d)^e \bmod n = m^{\overbrace{de \bmod \phi}^{=1}} \bmod n =$

$= m \bmod n \underset{\text{if } m<n}{=} m$

A: $PrK_A = d_A$, $PuK_A = (n_A, e)$

$\xleftarrow{PuK=(n,e)}$

B: $PrK = d$, $PuK = (n, e)$.

A: $m = 100$; is masking value m:

$t \leftarrow randi$; $1 < t < n$: $\underline{\gcd(t,n)=1} \Rightarrow \underline{\exists!\, t^{-1} \bmod n}$.

$m' = m \cdot t^e \bmod n \xrightarrow{m'}$

$Ver(PuK=(n,e), \sigma', m') = m' \xleftarrow{\sigma'}$

B:

$Sign(PrK=d, m') = \sigma'$

$\sigma' = (m')^d \bmod n =$

$= (m \cdot t^e)^d \bmod n =$

$= m^d \cdot t^{\overbrace{ed \bmod \phi}^{\to 1}} \bmod n =$

$= m^d \cdot t \bmod n$

$\xleftarrow{\sigma' = m^d \cdot t \bmod n}$

A: unmasks signed m'

$(\sigma')^e \bmod n = ((m')^d)^e \bmod n = (m')^{\overbrace{ed \bmod \phi}^{=1}} \bmod n =$

$= m' \bmod n \underset{\text{if } m'<n}{=} m' \Rightarrow$ Signature is valid. $=$ True

A: wants to find a valid signature σ of B on $m = 100$:

$\sigma = m^d \bmod n$

A extracts (unmasks) $m^d \bmod n = \sigma$ from σ':

$\sigma' \cdot t^{-1} \bmod n \longrightarrow$ if $\gcd(t, n) = 1 \Rightarrow t^{-1} \bmod n$ exists.

$\sigma' \cdot t^{-1} \bmod n = \underbrace{m^d \cdot \cancel{t}} \cdot \underbrace{\cancel{t^{-1}}} \bmod n = \underline{m^d \bmod n = \sigma.}$

But $m^d \bmod n$ – is a B's signature on the actual amount of money $m = 100$.

$\sigma = m^d \bmod n$.

A: (m, σ) $\xrightarrow[\text{to the Vendor } \mathcal{V}]{(m,\sigma)}$

$PuK=(n,e)$ B's

$\mathcal{V}$: verifies is B's signature on the money amount $m = 100$ is true

$Ver(PuK=(n,e), \sigma, m) = True$

$\sigma^e \bmod n = (m^d)^e \bmod n = m^{de \bmod \phi} \bmod n = m \bmod n = m$ if $m < n$

E-coin properties.

1.**Anonimity**.

2.**Untraceability**.

3.**Double-spending prevention**.

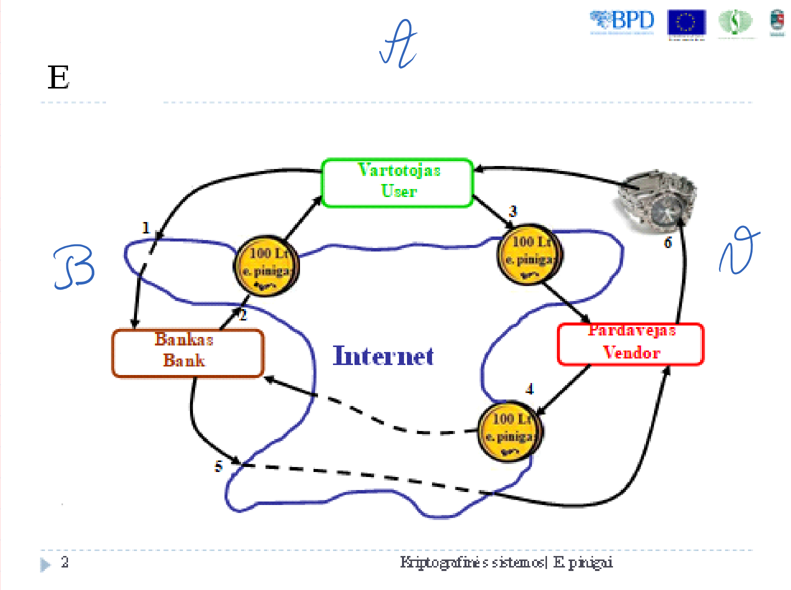
4.**Divisibility**.

Chaum

Divisible coins (e-money) are growing is size.

$A: \xrightarrow{(m,\sigma),\, AD_1} V_1 \quad \xrightarrow{(m,\sigma),\, AD_1,\, AD_2} V_2$ ———

$\xrightarrow{(m,\sigma),\, \underbrace{AD_1,\, AD_2,\, AD_3}_{\text{growing in size}}} V_3$ — — — — —



e-money anonimity

A: 50 claims to withdraw e-money from B.

$m_1 = 100,\ m_2 = 100, \ldots, m_{50} = 100.$

$r_1 \leftarrow randi,\ r_2 \leftarrow randi,\ \ r_{50} \leftarrow randi.$

$m_1' = m_1 \cdot r_1^e \bmod n, \ldots, m_{50}' = m_{50} \cdot r_{50}^e \bmod n.$

$\xrightarrow{m_1',\, m_2',\, \ldots,\, m_{50}'} B: m_i' \leftarrow rand\{m_1', \ldots, m_{50}'\}$

$\xleftarrow{m_1', \ldots, m_{i-1}',\, m_{i+1}', \ldots, m_{50}'}$

$\xrightarrow{r_1', \ldots, r_{i-1}',\, r_{i+1}', \ldots, r_{50}'}$ Since $m_j' = m_j \cdot r^e \bmod n$

$(m_j')^d = m_j \cdot r \bmod n$

$$(m_j')^{d} \cdot r^{-1} = m_j \bmod n$$

By collecting all m_j, $j = 1, 2, \ldots, i-1, i+1, \ldots, 50$,
$\mathcal{B}$ verifies: 1) if all m_j has the same value ?
2) if $\mathcal{A}$ account sum $s > m_j$?
If Yes then $\mathcal{B}$ blindly signs remaining value m_i'

$$\sigma_i' = (m_i')^d \bmod n = (m \cdot r^e)^d = m^d\, r \bmod n$$

$1 \bmod \Phi$

The probability for $\mathcal{A}$ to cheat is : $Pr(\text{cheating}) = \frac{1}{50}$

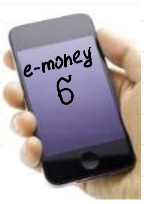
$\mathcal{A}$: is unmasking σ_i' and obtains

$$\sigma_i = \sigma_i' \cdot r^{-1} \bmod n = m_i^d \bmod n.$$

$\mathcal{A}$: verifies σ_i on m_i : $Ver(PuK = (n, e), \sigma_i, m) = T$

$$m_i = (\sigma_i)^e \bmod n = m_i^{de \bmod \Phi} \bmod n = m_i^1 \bmod n = m_i \quad \text{if } m_i < n.$$

1. Coin withdrawal Protocol 1. Untraceability.



e-wallet

$\sigma = m^d \bmod n$

$m = 100$ Lt

e-purse
wallet

off-line +
on-line −

1'. Coin withdrawal Protocol 1'. Untraceability + off-line payment.
+ Double spending preven.

$\mathcal{A}$: creates Random Identification String RIS for every m_j':
Then $\mathcal{A}$ encodes her name by some binary string $A = 1010$.

$x_{j1} \leftarrow \text{randbin} \rightarrow x_{j1} = 0110$

$\rightarrow x_{j1}' = A \oplus x_{j1} \rightarrow$

$$\begin{array}{rr} & A \quad 1010 \\ \oplus & x_{j1} \quad 0110 \\ \hline & x_{j1}' = 1100 \end{array}$$

2) Payment protocol

3) Deposit protocol

$\mathcal{A}$ computes:

x_{j1}, x_{j1}' ; x_{j2}, x_{j2}' ; ... ; $x_{j,50}, x_{j,50}'$.

If x_{jk} and x_{jk}' is revealed, then the identity of $\mathcal{A}$ will be revealed.

E.g. Let x_{j1} and x_{j1}' is known, then

E.g. Let x_{j1} and x'_{j1} is known, then

$$A = x_{j1} \oplus x'_{j1} \longrightarrow \oplus \begin{array}{r} 0110 \\ 1100 \\ \hline 1010 \end{array} = A$$

$y_{j1} = H(x_{j1})$, $y'_{j1} = H(x'_{j1})$.

$m'_1 = m_1 \cdot r_1^e \bmod n, \ldots, m'_{50} = m_{50} \cdot r_{50}^e \bmod n$.

$\Pi'_1 = (m'_1; y_{11}, y'_{11}; \ldots; m'_{1,50}; y_{1,50}, y'_{1,50})$

$\Pi'_2 = \ldots$

$\ldots$

$\Pi'_{50} = \ldots$

$\xrightarrow{\Pi'_1, \Pi'_2, \ldots, \Pi'_{50}}$ B: $\Pi'_i \leftarrow \text{rand}\{\Pi'_1, \ldots, \Pi'_{50}\}$

$\xleftarrow{\Pi'_1, \ldots, \Pi'_{i-1}, \Pi'_{i+1}, \ldots, \Pi'_{50}}$

$\xrightarrow{r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_{50}}$

Verifies if:

1) all m_j have the same value

2) A account $s > m_j$

B blindly signs e-coin Π'_i

$\text{Sig}(PrK = d, \Pi'_i) = \sigma'_i$

$\xleftarrow{\sigma'_i}$

A: unmasks σ'_i in the same way by computin σ_i on the sum m_i and hence A has e-coin Π_i consistin of the following:

$\Pi_i = (m_i, \sigma_i, y_{i1}, y'_{i1}; \ldots; y_{i,50}, y'_{i,50})$

↑ not necessary to include since having signature σ_i the value m_i can be computed during the verification phase.

$\sigma_i = M^d \bmod n$; $M_i = {}' m_i; y_{i1}, y'_{i1}; \ldots; y_{i,50}, y'_{i,50}{}'$

$\text{Ver}(PuK = (n, e), \sigma_i, M_i) = T$

Instead of Π_i we will use the notation Π of e-coin.

$\Pi = (m; \sigma; y_1, y'_1; \ldots; y_{50}, y'_{50})$

2. Payment protocol.

$\mathcal{A}$: $\xrightarrow{\quad \Pi \quad}$ $\mathcal{V}$: Victor – vendor verifies

1) If signature on m is a valid $\mathcal{B}$ signature

$Ver(PuK=(n,e), \sigma, m) = \mathbf{T}$

2) If m value is equal to the price of silver wath.

3) $\mathcal{V}$ generates random bit string – RBS consisting of 50 bits

E.g. $RBS = 1\ 0\ 1\ 1, \ldots, 0$ with bits $b_1\ b_2\ b_3\ b_4 \ldots b_{50}$

$\mathcal{A}$: is taking RBS $\xleftarrow{\quad RBS \quad}$

and reveals either x_1 if $b_1 = 1$ or x_1' if $b_1 = 0$

x_2 if $b_2 = 1$ or x_2' if $b_2 = 0$

– – – – – –

x_{50} if $b_{50} = 1$ or x_{50}' if $b_{50} = 0$

$x_1, x_2', x_3, x_4, \ldots, x_{50}'$ $\longrightarrow$ $\mathcal{V}$: verifies

$\mathcal{A}$: $\longleftarrow$

$$\left\{\begin{array}{l} \text{if } H(x_1) = y_1 \\ \text{if } H(x_2') = y_2' \\ \text{– – – – – – –} \\ \text{if } H(x_{50}') = y_{50}' \end{array}\right\} \quad \text{If it is } \mathbf{T}$$

3. Deposit protocol. Vendor deposits his e-coins to his bank account.

$\mathcal{V}$: $\Pi, (x_1, x_2', x_3, x_4, \ldots, x_{50}')$ $\longrightarrow$ $\mathcal{B}$: Verifies: 1) if σ on Π is valid?

2) if the same string of $(y_1, y_1'; \ldots; y_{50}, y_{50}')$ didn't deliver to him?

If it is $\mathbf{T}$, the $\mathcal{B}$ deposits e-coin Π to the $\mathcal{V}$ account.

4. $\mathcal{Lo}$ impersonates $\mathcal{A}$ and is double spending Π.

To protect $\mathcal{A}$ honour we assume that $\mathcal{Lo}$ together with Π

seized also $RIS = (x_1, x_1'; x_2, x_2'; \ldots; x_{50}, x_{50}')$

$\mathcal{A}$: $\xrightarrow{\Pi}$ $\mathcal{V}$: generates a different RBS_2,

$RBS \neq RBS_2 = 1101, \ldots, 0$

$\xleftarrow{RBS_2}$ $Pr(RBS = RBS_2) = 1/2^{50}$

$\mathcal{A}$ knows the actual RIS, hence she reveals to $\mathcal{V}$ required values

$\xrightarrow{x_1, x_2, x_3', x_4, \ldots, x_{50}'}$ $\mathcal{V}$: 1) Verifies signature σ on m

2) If m value is correct

3)

$$\left.\begin{cases} \text{if } H(x_1) = y_1 \\ \text{if } H(x_2) = y_2 \\ \text{-- -- -- -- -- --} \\ \text{if } H(x_{50}') = y_{50}' \end{cases}\right\} T$$

$\mathcal{A}$ $\xleftarrow{\quad}$ 

$\mathcal{V}$: $\xrightarrow{\Pi, (x_1, x_2, x_3', x_4, \ldots, x_{50}')}$ $\mathcal{B}$: Verifies:

1) If σ on Π is valis? T

2) If the same coin Π with the same $(y_1, y_1', \ldots, y_{50}, y_{50}')$ is already received previously: Yes

$\mathcal{B}$: discloses the identity of e-coin Π holder.

$$\begin{array}{rl} \oplus & x_1, x_2', x_3, x_4, \ldots, x_{50}' \\ & x_1, x_2, x_3', x_4, \ldots, x_{50}' \\ \hline & \vec{0}, A, A, \vec{0}, \ldots, \vec{0} \end{array}$$

$\downarrow$

$\mathcal{A}$ identity $A = 1010$

so $\mathcal{A}$ due to distraction has a problems with law enforcement.

Property: the only customer **Alice** can create and is responsible for Random Identification String - RIS during the Withdrawal protocol.

Questions:

1.Is it possible for **Alice** to modify e-coin ∏.

1.How vendor **Victor** can cheat against **Bank** and how it is prevented?

E-coin properties.

1.**Anonimity**.

2.**Untraceability**.

3.**Double-spending prevention**.

4.**Divisibility**.

International Association for Cryptographic Research - IACR Barcelona, 2008, announced results:
1.Divisible e-money can be trully anonymous.
2.Divisible and trully anonymous e-money grow in size during their transfers.