## Tutorial on Public Key Cryptography – RSA

---

### RSA – the Key Generation – Example

1. Randomly choose two prime numbers $p$ and $q$.

   We choose $p = 11$ and $q = 13$.

2. Compute $n = pq$.

   We compute $n = pq = 11 \cdot 13 = 143$.

3. Randomly choose an odd number $e$ in the range $1 < e < \varphi(n)$ which is coprime to $\varphi(n)$ (i.e., $e \in Z^*_{\varphi(n)}$).

   $\varphi(n) = \varphi(p) \cdot \varphi(q) = 10 \cdot 12 = 120$.
   Thus, we choose $e = 7$ ($e \in Z^*_{120}$).

4. Compute $d \equiv e^{-1} \pmod{\varphi(n)}$ by Euclid's algorithm. Thus, $de \equiv 1 \pmod{\varphi(n)}$.

   We compute $d \equiv e^{-1} \equiv 7^{-1} \equiv 103 \pmod{\varphi(143) = 120}$.

   Check that $120 | 7 * 103 - 1 = 721 - 1 = 720$.

---

### RSA – the Key Generation – Example (cont.)

5. Publish $(n, e)$ as the public key, and keep $d$ secret as the secret key.

   We publish $(n, e) = (143, 7)$ as the public key, and keeps $d = 103$ secret as the secret key.

---

### RSA – Encryption/Decryption – Example

**The encryption algorithm $E$:**

Everybody can encrypt messages $m$ ($0 \le m < n_A$) to user $A$ by

$$c = E_A(m) = m^{e_A} \bmod n_A.$$

The ciphertext $c$ ($0 \le c < n_A$) can be sent to $A$, and only $A$ can decrypt.

Encrypt $m = 3$:

$$E_A(m) \equiv m^{e_A} \equiv 3^7 \equiv 42 \pmod{143}$$

---

### RSA – Encryption/Decryption – Example (cont.)

**The decryption algorithm $D$:**

Only $A$ knows his secret key $d_A$ and can decrypt:

$$m = D_A(c) = c^{d_A} \bmod n_A.$$

Decrypt $c = 42$:

$$D_A(c) \equiv c^{d_A} \equiv 42^{103} \equiv 3 \pmod{143}$$

Decrypt $c = 2$:

$$D_A(c) \equiv c^{d_A} \equiv 2^{103} \equiv 63 \pmod{143}$$

---

### Existential Forgery of an RSA Signature

Given a public key $(n_A, e_A)$ of user A, can another user B create a message $m$ and a signature $D_A(m) \equiv m^{d_A} \pmod{n_A}$?

User B can forge a signature in the following way:

B Chooses $y \in Z_n$ and calculates $x \equiv E_A(y) = y^{e_A} \pmod{n_A}$.

Now B can claim that $y \equiv x^{d_A} \pmod{n_A}$ is A's signature on $x$.

---

### Multiplication Property of RSA

**Multiplication Property:**

Given a public key $(n_A, e_A)$ of user A, and $m_1, m_2 \in Z_n$ then

$$E_A(m_1 \cdot m_2) \equiv E_A(m_1) \cdot E_A(m_2) \pmod{n}$$

**Proof:**

$$E_A(m_1) \equiv m_1^{e_A} \pmod{n_A}$$
$$E_A(m_2) \equiv m_2^{e_A} \pmod{n_A}$$

and,

$$E_A(m_1 \cdot m_2) \equiv (m_1 \cdot m_2)^{e_A} \equiv m_1^{e_A} \cdot m_2^{e_A} \equiv E_A(m_1) \cdot E_A(m_2) \pmod{n_A}$$

QED

---

### Random Self Reducibility of RSA

**Problem:**

Given a public key $(n_A, e_A)$ of user A:

Assume we are given an algorithm, called ALG, which given $E_A(m) \equiv m^{e_A} \pmod{n_A}$ can find the message $m$ for $\frac{1}{100}$ of the possible cryptograms.

Show a polynomial random algorithm which given $E_A(m) \equiv m^{e_A} \pmod{n_A}$ finds the message $m$ with probability $\ge \frac{1}{2}$ for every cryptogram in $Z^*_{n_A}$.

## Random Self Reducibility of RSA (cont.)

**The Algorithm**:

Make $t$ iterations of the following:

1. Randomly Choose $z \in Z_{n_A}^*$.

2. Calculate $z^{e_A} \pmod{n_a}$.

3. Let $x = \text{ALG}(m^{e_A} z^{e_A} \pmod{n_A})$.

4. If $x^{e_A} \equiv m^{e_A} z^{e_A} \pmod{n_A}$ then output $x \cdot z^{-1} \pmod{n_A}$ and finish.

## Random Self Reducibility of RSA (cont.)

**Correctness**:

If algorithm ALG succeeds, i.e., outputs $x$ such that

$$x^{e_A} \equiv (mz)^{e_A} \equiv m^{e_A} \cdot z^{e_A} \pmod{n_A}$$

Then, $m \equiv x \cdot z^{-1} \pmod{n_A}$.

For $z \in Z_n^*$

$$\{z \cdot x : x \in Z_n\} = z \cdot Z_n = Z_n$$

because if for $x_1, x_2$ we have:

$$z \cdot x_1 \equiv z \cdot x_2 \pmod{n}$$

which implies

$$x_1 \equiv z^{-1} \cdot z \cdot x_1 \equiv z^{-1} \cdot z \cdot x_2 \equiv x_2 \pmod{n}$$

## Random Self Reducibility of RSA (cont.)

Thus, for every iteration we have $\frac{1}{100}$ probability of success, thus in order to find $m$ with probability $\geq \frac{1}{2}$, $t$ must satisfy:

$$\left(\frac{99}{100}\right)^t \leq \frac{1}{2}$$
$$t \log\left(\frac{99}{100}\right) \leq \log\frac{1}{2}$$

Thus, $t \geq 69$ suffices.

**Note**: Can we find the message when $E_A(m) \notin Z_{n_A}^*$?

When the cryptogram $E_A(m) \notin Z_n^*$, we can find either $p$ or $q$. Thus, we can find $d$ and then $m$.

## Random Self Reducibility of RSA (cont.)

**Note**:

Let $p$ be a prime and $g$ be a generator of $Z_p^*$.

Thus, for $a \in Z_p^*$ there exists $z$ such that $g^z \equiv a \pmod{p}$. This $z$ is called the **discrete logarithm** or **index** of $a$, modulo $p$, to the base of $g$. We denote this value as $\text{ind}_{p,g}(a)$ or $\text{DLOG}_{p,g}(a)$.

DLOG is also random self reducible given a generator $g$ of $Z_p^*$.

## Blind Signatures

Usually when we sign a document we check its contents.

But we might want people to sign documents without ever seeing them.

For example, Bob is a notary. Alice wants him to sign a document, but does not want him to have any idea what he is signing. Bob doesn't care what the document says, he is just certifying that he notarized it at a certain time.

1. Alice takes the document and uses a "**blinding factor**".

2. Alice sends the blinded document to Bob.

3. Bob signs the blinded document.

4. Alice computes the signature on the original document.

## Blind Signatures using RSA

Can Bob, with a public key $(n, e)$, sign a message $m$ (actually signs $H(m)$) without knowing its contents?

Yes.

We choose a random $\alpha \in Z_n^*$ and then ask Bob to sign

$$E(\alpha) \cdot m \equiv \alpha^e \cdot m \pmod{n}$$

we get:

$$D(E(\alpha) \cdot m) \equiv (E(\alpha) \cdot m)^d \equiv (\alpha^e \cdot m)^d \equiv \alpha \cdot m^d \pmod{n}$$

thus we need only to calculate $\alpha^{-1} \pmod{n}$:

$$m^d \equiv \alpha^{-1} \cdot D(E(\alpha) \cdot m) \pmod{n}$$

Note that the function $f(x) \equiv x^e \pmod{n}$ is a permutation of $Z_n$. Moreover, if $\alpha \in Z_n^*$ then $\alpha^e \in Z_n^*$.

## An Example of Using Blind Signatures

**Problem 1**:

Alice wants a virtual 100 dollar note.

**Solution 1**:

Alice goes to the bank and asks for such a note. The bank gives Alice a signature on a virtual 100 dollar check.

Denote the bank's public key by $(n, e)$ and its secret key by $d$.

## An Example of Using Blind Signatures (cont.)

**Problem 2**:

The bank can trace this check to Alice.

**Solution 2**:

Use a blind signature.

The bank signs a check $m$ by using a blind signature:

Alice wants to get $m^d$. Thus, Alice chooses a random $\alpha \in Z_n^*$ and then asks the bank to sign:

$$E(\alpha) \cdot m \equiv \alpha^e \cdot m \pmod{n}$$

now Alice needs only to calculate $\alpha^{-1} \pmod{n}$:

$$m^d \equiv \alpha^{-1} \cdot D(E(\alpha) \cdot m) \pmod{n}$$

## An Example of Using Blind Signatures (cont.)

**Problem 3**:

Alice can trick the bank into giving her a check worth more than 100 dollars.

**Solution 3**:

- Alice prepares 100 versions of the check $m_1, \ldots, m_{100}$.

- Alice chooses random $\alpha_1, \ldots, \alpha_{100} \in Z_n^*$ at random.

- Alice gives $y_i = \alpha_i^e \cdot m_i \pmod{n}$ to the bank.

- The bank asks Alice to reveal 99 of the $\alpha$'s. Denote the remaining $\alpha$ by $\alpha_k$.

- The bank signs $y_k$.

- Alice calculates $m^d \equiv \alpha_k^{-1} \cdot (\alpha_k^e \cdot m)^d \pmod{n}$.

## An Example of Using Blind Signatures (cont.)

Assume Alice tries to cheat by using one check $m_j$ which is worth a million dollars.

If the bank does not ask for $\alpha_j$ then Alice gets million dollars. On the other hand, if the bank does ask for $\alpha_j$, giving it the real $\alpha_j$ exposes her deceit.

Alice would prefer to reveal $\beta_j$ such that

$$m_j' \equiv y_j \cdot \beta_j^{-e} \pmod{n}$$

where $m_j'$ is a check on 100 dollars.

$$y_j \equiv m_j \cdot \alpha_j^e = m_j' \cdot \beta_j^e \pmod{n}$$

thus,

$$\beta_j^e \equiv \alpha_j^e \cdot m_j \cdot m_j'^{-1} \pmod{n}$$

or

$$\beta_j \equiv \alpha_j \cdot m_j^d \cdot m_j'^{-d} \pmod{n}$$

Which means we can find $(m_j \cdot m_j'^{-1})^d$.

## An Example of Using Blind Signatures (cont.)

**Another approach:**:

The bank will use different public keys for different bill values, i.e., the bank will use $(e_{100}, n_{100})$ for 100 dollar bills, $(e_{20}, n_{20})$ for 20 dollar bills, etc.

For a bill to be valid, it must be formatted like $m =$ "This is a 100 dollar bill, serial number: x", where the serial number is a very long random chosen by Alice, and the bill must be signed using the appropriate public key.

## Examples

**Question**:

We are given the following implementation of RSA:

A trusted center chooses $p$ and $q$, and publishes $n = pq$. Then, $n$ is used by all the users.

He gives the $i$'th user a private key $d_i$ and a public key $e_i$, such that $\forall_{i \neq j} e_i \neq e_j$.

Show that if two users, $i$ and $j$, for which $gcd(e_i, e_j) = 1$, receive the same message $m$, it is possible to reconstruct $m$ by using $n, e_i, e_j, m^{e_i}, m^{e_j}$.

**Solution**:

$$\gcd(e_i, e_j) = 1 \Rightarrow \exists x, y \; x e_i + y e_j = 1$$

Thus,

$$(m^{e_i})^x \cdot (m^{e_j})^y \equiv m^{x e_i + y e_j} \equiv m \pmod{n}$$

**Exercise**: Show that even one user can reconstruct a message $m$ without cooperation of any other user.

## Examples (cont.)

**Question**:

Let $p$ and $q$ be prime, $n = pq$.

Alice wishes to send messages to Bob using the RSA cryptosystem. Unwisely she does not choose her own keys, but allows Eve to choose them for her. The only precaution that Alice takes is to check that $e \neq 1 \pmod{\varphi(n)}$.

Show that Eve can still choose a pair of keys $e, d$ such that encryption and decryption can be accomplished, but $m^e \equiv m \pmod{n}$, for every $m \in Z_n^*$.

## Examples (cont.)

**Solution**:

Denote $\ell = \text{lcm}(p - 1, q - 1)$. Thus,

$$m^{\ell+1} \equiv m^{a(p-1)+1} \equiv 1^a \cdot m = m \pmod{p}$$

and

$$m^{\ell+1} \equiv m^{b(q-1)+1} \equiv 1^b \cdot m = m \pmod{q}$$

thus by the Chinese reminder theorem

$$m^{\ell+1} \equiv m \pmod{n}.$$