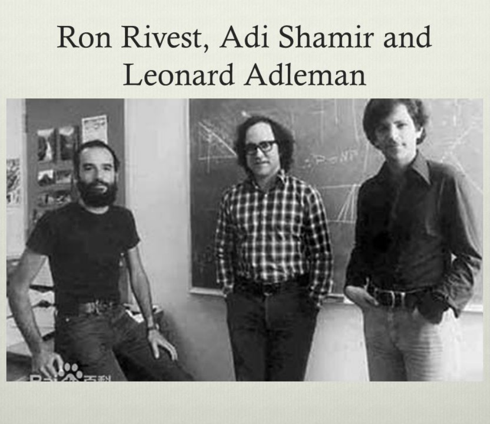


CS 553
CRYPTOGRAPHY

Lecture 24
RSA

Instructor
Dr. Dhiman Saha

Image Source: Google

- In 1977, Ronald Rivest, Adi Shamir and Leonard Adleman proposed a scheme which became the most widely used asymmetric cryptographic scheme, RSA.

RSA Key Generation

Output: public key: $k_{pub} = (n, e)$ and private key: $k_{pr} = (d)$

1. Choose two large primes p and q.
2. Compute $n = p \cdot q$.
3. Compute $\Phi(n) = (p-1)(q-1)$.
4. Select the public exponent $e \in \{1, 2, \ldots, \Phi(n) - 1\}$ such that

$$\gcd(e, \Phi(n)) = 1.$$

5. Compute the private key d such that

$$d \cdot e \equiv \bmod \Phi(n)$$

Given the public key $(n, e) = k_{pub}$ and the plaintext x, the encryption function is:

$$y = e_{k_{pub}}(x) \equiv x^e \bmod n$$

Here where $x, y \in \mathbb{Z}_n$.

- RSA encrypts plaintexts x, where we consider the bit string representing x to be an element in $\mathbb{Z}_n = \{0, 1, \cdots, n-1\}$.
- As a consequence the binary value of the plaintext x must be less than n.
- The value e is sometimes referred to as **encryption exponent** or **public exponent**

Given the private key $d = k_{pr}$ and and the ciphertext y, the decryption function is:

$$x = d_{k_{pr}}(y) \equiv y^d \bmod n$$

Here $x, y \in \mathbb{Z}_n$.

- The private key d is sometimes called **decryption exponent** or **private exponent**

An Example

Alice		**Bob**
message $x = 4$		1. choose $p = 3$ and $q = 11$
		2. $n = p \cdot q = 33$
		3. $\Phi(n) = (3-1)(11-1) = 20$
		4. choose $e = 3$
		5. $d \equiv e^{-1} \equiv 7 \bmod 20$
	$\xleftarrow{k_{pub}=(33,3)}$	
$y = x^e \equiv 4^3 \equiv 31 \bmod 33$		
	$\xrightarrow{y=31}$	
		$y^d = 31^7 \equiv 4 = x \bmod 33$

RSA Parameters for 1024 bit-length modulus

$$
\begin{aligned}
p = {} & E0DFD2C2A288ACEBC705EFAB30E4447541A8C5A47A37185C5A9 \\
& CB98389CE4DE19199AA3069B404FD98C801568CB9170EB712BF \\
& 10B4955CE9C9DC8CE6855C6123_h \\
q = {} & EBE0FCF21866FD9A9F0D72F7994875A8D92E67AEE4B515136B2 \\
& A778A8048B149828AEA30BD0BA34B977982A3D42168F594CA99 \\
& F3981DDABFAB2369F229640115_h \\
n = {} & CF33188211FDF6052BDBB1A37235E0ABB5978A45C71FD381A91 \\
& AD12FC76DA0544C47568AC83D855D47CA8D8A779579AB72E635 \\
& D0B0AAAC22D28341E998E90F82122A2C06090F43A37E0203C2B \\
& 72E401FD06890EC8EAD4F07E686E906F01B2468AE7B30CBD670 \\
& 255C1FEDE1A2762CF4392C0759499CC0ABECFF008728D9A11ADF_h \\
e = {} & 40B028E1E4CCF07537643101FF72444A0BE1D7682F1EDB553E3 \\
& AB4F6DD8293CA1945DB12D796AE9244D60565C2EB692A89B888 \\
& 1D58D278562ED60066DD8211E67315CF89857167206120405B0 \\
& 8B54D10D4EC4ED4253C75FA74098FE3F7FB751FF5121353C554 \\
& 391E114C85B56A9725E9BD5685D6C9C7EED8EE442366353DC39_h \\
d = {} & C21A93EE751A8D4FBFD77285D79D6768C58EBF283743D2889A3 \\
& 95F266C78F4A28E86F545960C2CE01EB8AD5246905163B28D0B \\
& 8BAABB959CC03F4EC499186168AE9ED6D88058898907E61C7CC \\
& CC584D65D801CFE32DFC983707F87F5AA6AE4B9E77B9CE630E2 \\
& C0DF05841B5E4984D059A35D7270D500514891F7B77B804BED81_h
\end{aligned}
$$

Correctness of RSA

To Show

Decryption is the inverse function of encryption

$$d_{k_{pr}}(e_{k_{pub}}(x)) = x$$

Points to Ponder

- $d \xleftarrow{\texttt{Computationally Infeasible}} (e, n)$
- Since x is only unique up to the size of the modulus n, we cannot encrypt more than l bits with one RSA encryption, where l is the bit length of n.
- Easy encryption/decryption
  - Implies need for fast exponentiation
- For a given n, there should be **many** private-key/public-key pairs, otherwise an attacker might be able to perform a brute-force attack.

The RSA Setting

Setup Let p and q be large primes, let $n = pq$, and let e and y be integers

Problem Solve the congruence $x^e \equiv y \bmod n$ for the variable x.

Easy Bob, who knows the values of p and q, can easily solve for x

Hard Eve, who does not know the values of p and q, cannot easily find x.

Trapdoor Solving $x^e \equiv y \bmod n$ is **easy** for a person who possesses certain **extra** information, but it is apparently **hard** for all other people.

Setup Let p and q be large primes, let $n = pq$, and let e and y be integers

Problem Solve the congruence $x^e \equiv y \bmod n$ for the variable x.

Easy Bob, who knows the values of p and q, can easily solve for x

Hard Eve, who does not know the values of p and q, cannot easily find x.

Trapdoor Solving $x^e \equiv y \bmod n$ is **easy** for a person who possesses certain **extra** information, but it is apparently **hard** for all other people.

Setup Let p and q be large primes, let $n = pq$, and let e and y be integers

Problem Solve the congruence $x^e \equiv y \bmod n$ for the variable x.

Easy Bob, who knows the values of p and q, can easily solve for x

Hard Eve, who does not know the values of p and q, cannot easily find x.

Trapdoor Solving $x^e \equiv y \bmod n$ is **easy** for a person who possesses certain **extra** information, but it is apparently **hard** for all other people.

Setup Let p and q be large primes, let $n = pq$, and let e and y be integers

Problem Solve the congruence $x^e \equiv y \bmod n$ for the variable x.

Easy Bob, who knows the values of p and q, can easily solve for x

Hard Eve, who does not know the values of p and q, cannot easily find x.

Trapdoor Solving $x^e \equiv y \bmod n$ is **easy** for a person who possesses certain **extra** information, but it is apparently **hard** for all other people.

The RSA Setting

Setup Let p and q be large primes, let $n = pq$, and let e and y be integers

Problem Solve the congruence $x^e \equiv y \bmod n$ for the variable x.

Easy Bob, who knows the values of p and q, can easily solve for x

Hard Eve, who does not know the values of p and q, cannot easily find x.

Trapdoor Solving $x^e \equiv y \bmod n$ is **easy** for a person who possesses certain **extra** information, but it is apparently **hard** for all other people.

The RSA Setting

Setup Let p and q be large primes, let $n = pq$, and let e and y be integers

Problem Solve the congruence $x^e \equiv y \bmod n$ for the variable x.

Easy Bob, who knows the values of p and q, can easily solve for x

Hard Eve, who does not know the values of p and q, cannot easily find x.

Trapdoor Solving $x^e \equiv y \bmod n$ is **easy** for a person who possesses certain **extra** information, but it is apparently **hard** for all other people.

The Number $\Phi(n)$

- Crucial to RSA's security

Why?

Finding $\Phi(n) \implies$ breaking RSA

- Thus p, q must be **secret**
- As knowing $p, q \implies$ knowing $\Phi(n)$

```
❶ sage: p = random_prime(2^32); p
  103222539
❷ sage: q = random_prime(2^32); q
  17870599
❸ sage: n = p*q; n
  c
❹ sage: phi = (p-1)*(q-1); phi
  3656723004526064
❺ sage: e = random_prime(phi); e
  1377192787721470
❻ sage: d = xgcd(e, phi)[1]; d
  1541797006342885
❼ sage: mod(d*e, phi)
  1
```

```
❶ sage: x = 1234567
❷ sage: y = power_mod(x, e, n); y
  1904832305575590 4
❸ sage: power_mod(y, d, n)
  1234567
```

How hard is it to find x without the **trapdoor** d?

- An attacker who can **factor** big numbers can break RSA by recovering p and q and then $\Phi(n)$ in order to compute d from e.
- An attackers ability to compute x from $x^e \bmod n$, or e^{th} roots modulo n, **without necessarily factoring** n.

Both risks seem closely connected, though we don't know for sure whether they are equivalent.

RSAs security level depends on three factors

- The size of modulus n
- The choice of p and q, and
- How the trapdoor permutation is used.

- What if n is too small?
- What if p, q are related of very different sizes?
- What if p, q are too close/small?
- Should the RSA trapdoor permutation be used directly for encryption or signing?

Textbook RSA Encryption

Textbook RSA Encryption

Used to describe the simplistic RSA encryption scheme

- The plaintext contains **only** the message you want to encrypt

Textbook RSA Encryption is **deterministic**

- Same plaintext $\implies$ same ciphertext

Malleability

Informally, ability to generate related ciphertexts without actually querying the encryption oracle.

Given two textbook RSA ciphertexts

- $y_1 = x_1^e \bmod n$
- $y_2 = x_2^e \bmod n$

What can you say about the ciphertext of $(x_1 \times x_2)$?

Multiplying the ciphertexts

$y_1 \times y_2 \bmod n = x_1^e \times x_2^e \bmod n = (x_1 \times x_2)^e \bmod n$

Breaking Malleability

Malleability

Informally, ability to generate related ciphertexts without actually querying the encryption oracle.

Given two textbook RSA ciphertexts

- $y_1 = x_1^e \bmod n$
- $y_2 = x_2^e \bmod n$

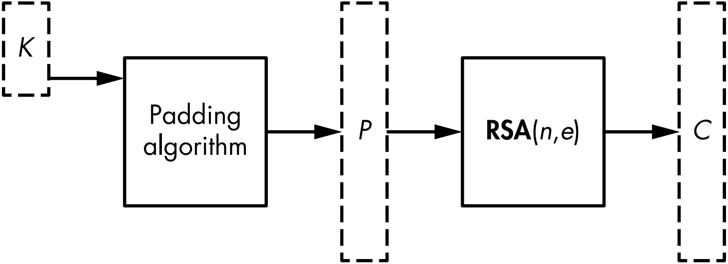
What can you say about the ciphertext of $(x_1 \times x_2)$?

Multiplying the ciphertexts

$y_1 \times y_2 \bmod n = x_1^e \times x_2^e \bmod n = (x_1 \times x_2)^e \bmod n$

Strong RSA Encryption: OAEP

Optimal Asymmetric Encryption Padding (OAEP)



RSA-OAEP

This scheme involves creating a bit string as large as the modulus by **padding** the message with extra data and **randomness** before applying the RSA function.