TUTORIAL 2

CSCI361 – Computer Security

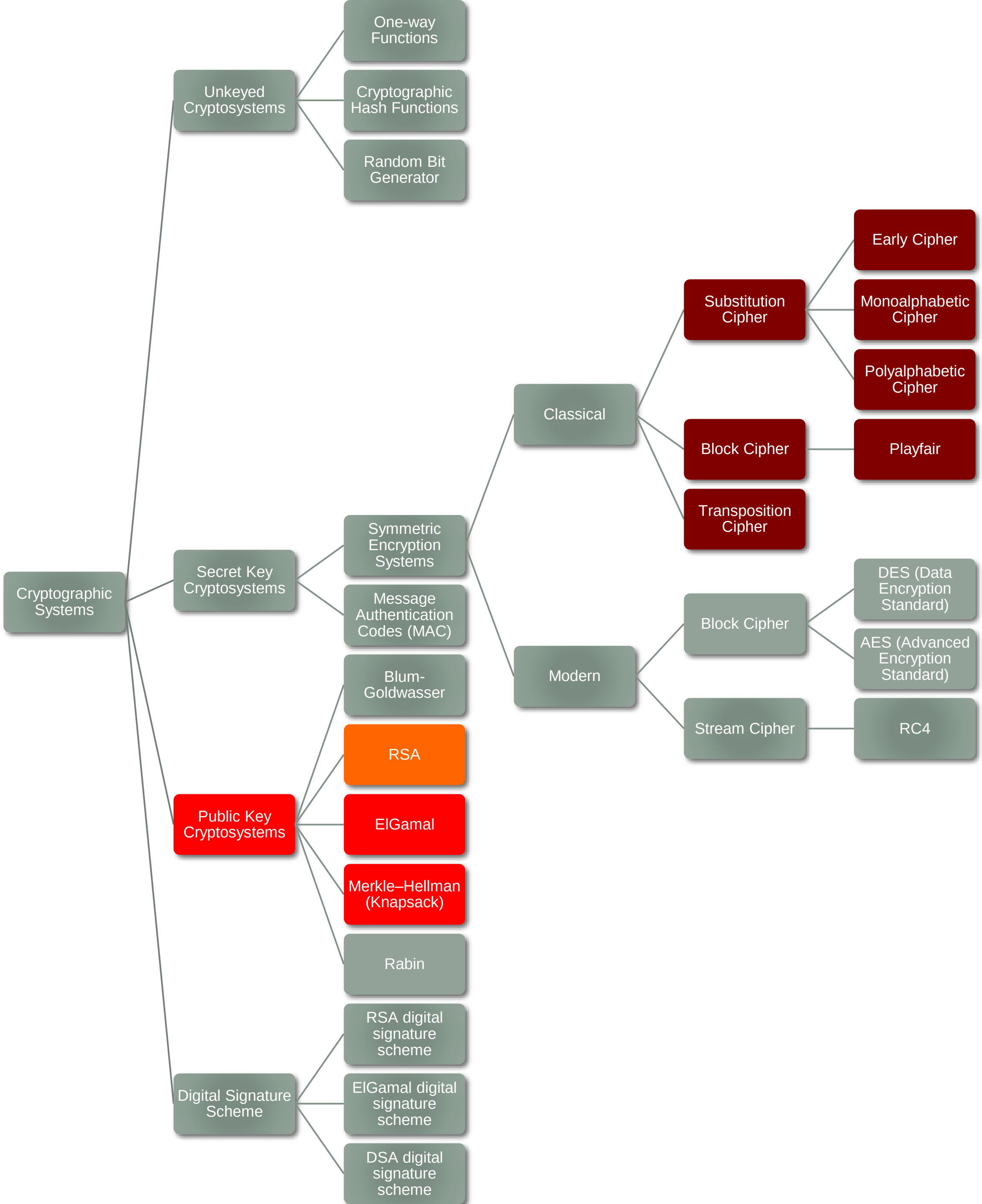
Sionggo Japit

sjapit@uow.edu.au

12 February 2024

TUTORIAL 2

RSA



Cryptographic Systems
Unkeyed Cryptosystems
One-way Functions
Cryptographic Hash Functions
Random Bit Generator
Secret Key Cryptosystems
Symmetric Encryption Systems
Message Authentication Codes (MAC)
Classical
Substitution Cipher
Early Cipher
Monoalphabetic Cipher
Polyalphabetic Cipher
Block Cipher
Playfair
Transposition Cipher
Modern
Block Cipher
DES (Data Encryption Standard)
AES (Advanced Encryption Standard)
Stream Cipher
RC4
Public Key Cryptosystems
Blum-Goldwasser
RSA
ElGamal
Merkle–Hellman (Knapsack)
Rabin
Digital Signature Scheme
RSA digital signature scheme
ElGamal digital signature scheme
DSA digital signature scheme

RSA

- Joinly invented by Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman at MIT in 1977.
- RSA’s strength lies in the tremendous difficulty in factorization of large numbers.
- Unlike other public cryptosystem, RSA can be both an asymmetric encryption system and a digital signature system (DSS), that we will discuss in Tutorial 4.

RSA

- The same set of algorithms can be used to encrypt and decrypt messages, as well as to digitally sign message and verify digital signatures.
  - If the recipient's public key is used to encrypt a plaintext message, then the RSA public key cryptosystem yields an asymmetric encryption system. In this case, the recipient's private key must be used to decrypt the ciphertext.
  - If the sender's private key is used to encrypt a plaintext message, then the RSA key cryptosystem yields a DSS. In this case, the sender's public key must be used to verify the digital signature.

RSA

RSA works as follows:

Setup:

- Take two large primes, p and q, and compute their product $n = p \times q$; n is called the modulus.
- Choose a number e, less than n and relatively prime to $(p-1) \times (q-1)$; *i.e.*, have no common factors except 1.
- Find another number d such that $((e \times d) - 1)$ is divisible by $(p-1) \times (q-1)$.
- The values e and d are called the public and private exponents, respectively.

RSA

- The public key is the pair (n, e);
- The secrete key is (n, d).
- The factors p and q may be kept with the private key, or destroyed.

RSA

Encryption:

- Sender has a message M which he/she splits into a sequence of blocks $M_1, M_2, \dots, M_t$ where each M_i satisfies $0 \leq Mi < n$. The sender then encrypts these blocks as

$$C_i = M_i^e \; mod \; n,$$

And sends the encrypted blocks to the intended recipients.

RSA

Decryption:

The recipient using his/her private key d by calculating

$$M_i = C_i^d \, mod \, n.$$

RSA

Example: Suppose Alice wants to sent message 4 to Bob using Bob's public key (77, 7).

Key generation:

- Bob chooses p = 7 and q = 11, and compute the modulus; that is

  $$n = p \times q = 7 \times 11 = 77$$

- Next Bob chooses e = 7 such that $gcd(e,(p-1)(q-1)) = 1$

  $$(p-1)(q-1) = (7 - 1)(11 - 1) = (6 * 10) = 60$$

RSA

- Check if gcd(7, 60) = 1?

n1	n2	r	q
60	7	4	8
7	4	3	1
4	3	1	1
3	1	0	3

Thus gcd(7,60) = 1

- Next Bob finds another number *d* such that *(ed – 1)* is divisible by *(p-1)(q-1)*.

RSA

- Using the extended gcd algorithm, Bob can find *d* such that *gcd(ed, (p-1)(q-1)) = 1*, in other words, *ed = 1 mod (p-1)(q-1). Thus d can be found using the extended Euclidean algorithm since d is the multiplicative inverse of e modulo Φ(n).*

n1	n2	r	q	a1	b1	a2	b2
60	7	4	8	1	0	0	1
7	4	3	1	0	1	1	-8
4	3	1	1	1	-8	-1	9
3	1	0	3	-1	9	2	-17

gcd(60, 7) = a2 (n1) + b2 (n2) = 2 (60) + (-17)(7)

Thus d = -17 mod (60) = 43

Hint: Use extended gcd instead of gcd to save one step ☺

RSA

- Thus Bob's public key is the pair **(n, e) = (77, 7)**, and private key is the pair **(n, d) = (77, 43).**

RSA

Encryption:

- Alice want to send the message *m = 4* to Bob.
- Alice encrypt the message *m* using Bob's public key pair (77, 7) as

$$
\begin{aligned}
C &= M^e \bmod n \\
&= 4^7 \bmod 77 \\
&= 16384 \bmod 77 \\
&= 60
\end{aligned}
$$

- Alice sends 60 to Bob.

RSA

Decryption:

- To decrypt, Bob uses his private key (*d = 43*) to recover the message

$$
\begin{aligned}
M &= C^d \bmod n \\
&= 60^{43} \bmod 77 \\
&= 4
\end{aligned}
$$

WEAKNESS IN RSA

- If some one can factor *n* in polynomial time, RSA is not secure.
- Common modulus attack.
- Important attack.

Weakness in RSA

If an attacker has a polynomial algorithm to factor *n*, which is a large arbitrary integer, why this makes RSA based public key cryptography insecure?

Weakness in RSA

- The reason is that the attacker can compute the victim's private key from the victim's public key.
- It is noted that $ed=1 \bmod \phi(n)$, and $\phi(n)=(p-1)(q-1)$. If the attacker has a polynomial algorithm, then the attacker can compute $n = (p \times q)$. Knowing the values of p and q, the attacker can then compute $\phi(n)$, *and hence* d can be computed from the victim's public key e using $d = e^{-1} \bmod \phi(n)$. Thus the message can be revealed.

Weakness in RSA (due to weak implementation)

Common modulus attack:

Adam and Barbie share the same modulus *n* for RSA to generate their encryption key e_A and e_B. Charlie sends them (Adam and Barbie) the same message m encrypted with e_A and e_B respectively. The resulting ciphertexts are c_A and c_B. Eve intercepts both c_A and c_B. Show how Eve can use the ***common modulus attack*** to compute the plaintext or the message m sent by Charlie if $\gcd(e_A, e_B) = 1$?

Weakness in RSA (due to weak implementation)

Eve knows the ciphertext $c_A \circ\ m^{e_A} \bmod n$ and $c_B \circ\ m^{e_B} \bmod n$. Eve also knows that the $\gcd(e_A, e_B) = 1$. Thus Eve can compute the inverse multiplicative of e_A and e_B using extended Euclidean algorithm to get $(e_A)(a) + (e_B)(b) = 1$.

Weakness in RSA (due to weak implementation)

Eve then computes $(c_A)^a \times (c_B)^b \mod n$ which she can obtain the message m as follow:

$$= (m^{e_A})^a \times (m^{e_B})^b \bmod n$$

$$= (m^{e_A \times a}) \times (m^{e_B \times b}) \mod n$$

$$= m^{(e_A)(a)+(e_B)(b)} \mod n$$

$$= m \mod n$$

$$= m$$

Weakness in RSA (due to weak implementation)

- In an organization where the boss uses the same modulus N for all his/her employee, show that the employee can actually decrypt a message even though the message is not intended for him/her.

Weakness in RSA (due to weak implementation)

- Although each employee has his/her own public and private keys (e_i, d_i), an employee can decrypt a message that was encrypted using someone else public key. For example:

$\text{Since } gcd(e_1, d_1) = 1, \text{we have } (e_1 \times d_1) = 1\ mod\ \phi(n)$.
$\text{Equivalently}, (e_1 \times d_1) = 1\ mod\ \phi(n)\ means$
$\phi(n) | (e_1 \times d_1) - 1.\ \text{Hence } (e_1 \times d_1) - 1 \equiv k \times \phi(n)$.

Let $\mathrm{V} = k \times \phi(n)$. If an employee knows e_1 and d_1, he/she can calculate V.

Weakness in RSA (due to weak implementation)

Using extended gcd, the employee can now calculate α and β (the multiplicative inverses) as follow:

$$\alpha \times e_2 + \beta \times V = 1$$

In the calculation above, α is an inverse multiplicative modulo e_2, and it meets the requirement of a private key correspond to $e_2 (i.e., d_2)$ because V is a multiple of $\phi(n)$. Hence, the employee can use α to decrypt the encrypted message.

Weakness in RSA

Important attack:

If an attacker can compute $\Phi(n)$ efficiently, then the attacker can break RSA. This is known as an ***important attack***. Show how the attacker can carry out this attack, in other words, how $\Phi(n)$ can be computed.

Weakness in rsa

- The public key of RSA is (e, n), where e is the encryption key (public) and $n = p$´ q.

- $f(n) = (p-1)(q-1)$.

  Expending the equation, we have

$$f(n) = pq - p - q + 1 \qquad \text{(eq. 1)}$$

Weakness in RSA

If n=p × q, it can be re-written as

$$q = \frac{n}{p} \quad \text{(eq. 2)}$$

Substituting eq. 2 into eq. 1, we have

$$f(n) = p\left(\frac{n}{p}\right) - p - \left(\frac{n}{p}\right) + 1$$

$$f(n) = n - p - \frac{n}{p} + 1$$

$$f(n) - n + p + \frac{n}{p} - 1 = 0$$

Weakness in RSA

Multiply both sides of the equation by p, we have

$$f(n)(p) - (n)(p) + (p)^2 + n - p = 0$$

$$p^2 + p(f(n) - n - 1) + n = 0$$

Since the attacker can compute $f(n)$, thus the polynomial equation can be solved for p as followed:

$$p_{1,2} = \frac{-(f(n) - n - 1) \pm \sqrt{(f(n) - n - 1)^2 - 4(1)(n)}}{2(1)}$$

Knowing p, q can easily obtained.