Security (CS4028)

Lecture 5. Asymmetric Cryptography I

Chunyan Mu

chunyan.mu@abdn.ac.uk

The School of Natural and Computing Sciences

Schedule

	Week	Lecture 1	Lecture 2	Tutorial
	1	Intro to course & security	Intro to Crypto	-
	2	Symmetric Crypto	Hash	Math for crypto
$\Rightarrow$	3	Asymmetric Crypto-1	Asymmetric Crypto-2	Symmetric Crypto
	4	Signatures	Zero Knowledge Proof	Asymmetric Crypto
	5	Certificates	Authentication	Signature & certificates
	6	Access Control	AC models	Authentication
	7	Information flow	Security Management	Access control
	8	Protocols	Communications	Concepts & management
	9	Network security	Network security	Protocols and communications
	10	Advanced topics	Advanced topics	Network
	11	Revision		

Lecture 5: Public key cryptography I

Last lecture

- Hash and MACs

This lecture

- understand mathematical ideas behind public key cryptography
- understand how the RSA algorithm works, including:
  - key generation, encryption and decryption

Outline

Outline

Asymmetric-key Cryptography

Public key cryptography

- In symmetric cryto, pairs of communicating peers share the same key to secure the communication (in some way).
- In asymmetric crypto., they use different-but-related keys.

Two Roles of Modern Cryptography

- **Communication**: encrypt/decrypt with key
  - **Symmetric**: different communicating parties use the same secret key for both encryption and decryption
  - **Asymmetric**: different parties use different private keys/secrets
- **Authentication, Data Integrity**: creation of a "fingerprint" or "message digest" for a digital object (e.g. a message or files) and that is hard to fake and that identifies the creator.
  - digital signatures, certificates

Asymmetric-key Cryptography

Public-private key pairs

- Many asymmetric schemes involve public- private key pairs.
- Let the private key be s (for secret).
- Let the public key be v (for visible).

Asymmetric-key Cryptography

Asymmetric Encryption

- Encrypt E with public key and decrypt D with private key.
- E.g., sending a secret, symmetric session key as the message.
- Anyone with the public key can send a confidential message to the owner of the private key
- Need $D(s, E(v, m)) = m$, for all inputs m.

Asymmetric-key Cryptography

Asymmetric Signature

- Encrypt. with the private key, decrypt with public
- Anyone with the public key can verify that the message was signed by someone who has the private key.
- E.g., RSA signatures
- Need $D(v, E(s, m)) = m$, for all inputs m.

Basic ideas of Public-key Cryptography

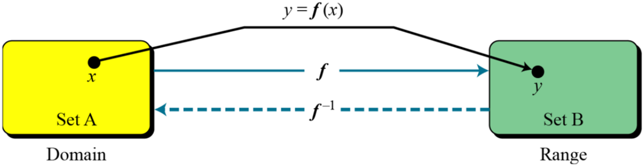
Trapdoor Function

- The main idea behind public-key cryptography is the concept of the trapdoor function:
  - a function that is easy to compute in one direction,
  - but difficult to compute in the opposite direction (finding its inverse) without special information (a secret), called 'trapdoor'.
- Trapdoor functions are widely used in cryptography:
  - the secret relates to (deriving) a private key.

Basic ideas of Public-key Cryptography

Trapdoor Function

- f is easy to compute
- f^{-1} is difficult to compute



Basic ideas of Public-key Cryptography

Trapdoor Function

- Some trapdoors are based on ‘simple’ number theoretic problems.
  - Discrete knapsack, discrete logarithm problem, prime factorisation, ...
- Some based on generalisation of these
  - E.g. to Group-theoretic problems, elliptic curves, shortest-vector problems in lattice-based cryptography, ...

Basic ideas of public-key crypto

Example: Trapdoor Function

When n is large, $n = p \times q$ is a one-way function.

- Given p, q, calculate: $\rightarrow n$ (**easy**)
- Given n, calculate: p, q (**difficult**)

Basic ideas of public-key crypto

Example: Trapdoor Function

When n is large, $n = p \times q$ is a one-way function.

- Given p, q, calculate: $\rightarrow n$ (**easy**)
- Given n, calculate: p, q (**difficult**)

Discrete Logarithm Problem (DLP)

When n is large, the function $y = x^k \mod n$ is a trapdoor one-way function.

- Given x, k and n, calculate: $\rightarrow y$ (**easy**)
- Given y, x and n, calculate: k (**difficult**)

Outline

Modular arithmetic

Definition: congruence modulo n

Given an integer $n > 1$, called a modulus, two integers a and b are said to be congruent modulo n, if there is an integer k such that $a - b = kn$.

- Congruence modulo n is denoted by:

$$a \equiv b \pmod{n}$$

- If a is non-negative, and $0 < b < n$, you can think of b as the remainder of a when divided by n, or the residue of a modulo n

Modular arithmetic

Examples

$$
\begin{aligned}
25 &\equiv 1 \pmod{4} \\
25 &\equiv 1 \pmod{3} \\
38 &\equiv 8 \pmod{15} \\
43 &\equiv 1 \pmod{3} \\
43 &\equiv 7 \pmod{12} \\
43 &\equiv 13 \pmod{15} \\
43 &\equiv 14 \pmod{29}
\end{aligned}
$$

Modular arithmetic

Properties

Modular arithmetic is just like normal arithmetic: commutative; associative; distributive

$(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$
$(a - b) \bmod n = ((a \bmod n) - (b \bmod n)) \bmod n$
$(a \times b) \bmod n = ((a \bmod n) \times (b \bmod n)) \bmod n$
$(a \times (b + c)) \bmod n = (((a \times b) \bmod n) + ((a \times c) \bmod n)) \bmod n$

Modular arithmetic

Example

$$
\begin{aligned}
& (39 + 57) \bmod 12 \\
= \;& ((39 \bmod 12) + (57 \bmod 12)) \bmod 12 \\
= \;& ((3 \bmod 12) + (9 \bmod 12)) \bmod 12 \\
= \;& 12 \bmod 12 \\
= \;& 0 \bmod 12
\end{aligned}
$$

Try one

$$17424 \bmod 12 = ?$$

Prime numbers

Definition

- A prime number p is an integer greater than 1 whose only factors are 1 and itself: no other number evenly divides it.
- A prime number p is an integer greater than 1 such that $p \equiv 0 \pmod{n}$ iff $n = 1$ or $n = p$

Examples

$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, \ldots, 73, 79, \ldots,$
$2591, \ldots, 2365347734339, \ldots, 2^{756839} - 1$

Relatively prime numbers

Definition

- Two numbers a and b are relatively prime <u>iff.</u> they have no factors in common other than 1
- i.e., , their greatest common divisor is equal to 1: $gcd(a, b) = 1$

Examples

$$(2, 3),\ (3, 4),\ (4, 5),\ (5, 6)\,; \ldots,\ (16, 81),\ \ldots$$

Relatively prime numbers

Euclid's algorithm to compute $gcd(a, b)$

gcd(a, b) recursive version

```
if b = 0 then
    return a
else
    return gcd(b, a mod b)
end if
```

gcd(a, b) iterative version

```
g = b
while (a > 0)
    g = a
    a = b mod a
    b = g
print "g = ", g
```

Example

a	b	a mod b
2322	654	360
654	360	294
360	294	66
294	66	30
66	30	6
30	6	0
6	0	-

a	b	g	$a > 0$
12	18	18	T
6	12	12	T
0	6	**6**	F

Extended Euclid Algorithm: EEA

What does it do?

Computes $gcd(a, b)$, as well as number x and y such that $ax + by = d$

Example

- Note hat $gcd(3, 5) = 1$, find x and y s.t. $3x + 5y = 1$
- by trial and error, we find the following pairs:
  - $x = 2, y = -1$: $3 \times 2 + 5 \times (-1) = 1$
  - $x = -3, y = 1$: $3 \times (-3) + 5 \times (2) = 1$
  - $x = -13, y = 8$: $3 \times (-13)2 + 5 \times (8) = 1$

Extended Euclid Algorithm: EEA

The algorithm

Find integers x and y such that $ax + by = d$ where $d = gcd(a, b)$

EEA(a, b)

if $b = 0$ **then**
    return (a, 1, 0)
else
    $(d', x', y') \leftarrow \text{EEA}(b, a \bmod b)$
    $(d, x, y) \leftarrow (d', y', (x' - \lfloor a/b \rfloor y'))$
    return (d, x, y)
end if

Extended Euclid Algorithm: EEA

Example

Find the d, x and y, by running EEA for the numbers $a = 2322$ and $b = 654$.

a	b	a mod b	$\lfloor a / b \rfloor$	d	x	y
2322	654	360	3	6	20	$-11 - 20 \times 3 = -71$
654	360	294	1	6	-11	$9 - (-11) \times 1 = 20$
360	294	66	1	6	9	$-2 - 9 \times 1 = -11$
294	66	30	4	6	-2	$1 - (-2) \times 4 = 9$
66	30	6	2	6	1	$0 - 2 \times 1 = -2$
30	6	0	5	6	0	$1 - 0 \times 5 = 1$
6	0	-	-	6	1	0

EEA is used to find inverses, when they exist.

Extended Euclid Algorithm: EEA

Remarks

- Find y such that $ay = 1 \pmod{b}$ with given $a < b$
  - if a and b are relatively prime, there exists a unique solution y
  - if a and b are non-relatively prime, there is no solution
- Equivalent to finding y and z such that $ay + bz = 1$

Inverse modulo a number

What is it?

- Remember inverse?
  - the inverse of 4 is $\frac{1}{4}$ since $4 \times \frac{1}{4} = 1$
- In modular world, the problem is more complicated: $4 \times ? \equiv 1 \pmod{7}$
  - by trial and error we can see $x = 2$
  - the equation is equivalent to finding an x and y s.t.

$$4x + 7y = 1$$

  - the general problem is finding an x s.t.

$$1 \equiv (a \times x) \bmod n$$

  - this is also written as: $a^{-1} \equiv x \bmod n$

Inverse modulo a number

The modular inverse problem is a lot more difficult to solve

- sometimes it has a solution, sometimes not, e.g.,
  - the inverse of 5 mod 14: $1 \equiv (5 \times ?) \bmod 14 \ \rightarrow 3$
  - however, the inverse of 2 mod 14: $1 \equiv (2 \times ?) \bmod 14 \ \rightarrow$ no solution!

Inverse modulo a number

The modular inverse problem is a lot more difficult to solve

- sometimes it has a solution, sometimes not, e.g.,
  - the inverse of 5 mod 14: $1 \equiv (5 \times ?) \bmod 14 \ \rightarrow 3$
  - however, the inverse of 2 mod 14: $1 \equiv (2 \times ?) \bmod 14 \ \rightarrow$ no solution!

How to find it?

- EEA is used to find **inverses** of a modulo n.
- We call $EEA(a, n)$ and we get the result (d, x, y):
  - if $d = 1$ then the inverse of a modulo n is x,
  - if $d \neq 1$, then a has no inverse modulo n.

Inverse modulo a number

Example: 11^{-1} mod 35

Equivalent to find x, y s.t. $11x + 35y = 1$ by running EEA.

a	b	a mod b	$\lfloor$ a / b $\rfloor$	d	x	y
11	35	11	0	1	16	-5-0 $\times$ 16 = -5
35	11	2	3	1	-5	1-3 $\times$ (-5) = 16
11	2	1	5	1	1	0-5 $\times$ 1 = -5
2	1	0	2	1	0	1-2 $\times$ 0 = 1
1	0	-	-	1	1	0

What is 11^{-1} mod 35 then?

Inverse modulo a number

Example: $11^{-1} \bmod 35$

Equivalent to find x, y s.t. $11x + 35y = 1$ by running EEA.

a	b	a mod b	$\lfloor$ a / b $\rfloor$	d	x	y
11	35	11	0	1	16	-5-0 $\times$ 16 $=$ -5
35	11	2	3	1	-5	1-3 $\times$ (-5) $=$ 16
11	2	1	5	1	1	0-5 $\times$ 1 $=$ -5
2	1	0	2	1	0	1-2 $\times$ 0 $=$ 1
1	0	-	-	1	1	0

What is 11^{-1} mod 35 then?

Try one:

$$8^{-1} \bmod 25, \qquad 4^{-1} \bmod 9, \qquad 4^{-1} \bmod 10$$

The Euler's Totient Function

Euler's totient function

- also called "Euler Phi function", written as $\phi(n)$
- $\forall n > 1$, $\phi(n)$ is the number of positive integers $< n$, that are relatively prime to n.
- e.g., $\phi(24) = 8$, since there are 8 totatives of 24: $1, 5, 7, 11, 13, 17, 19, 23$

How to compute $\phi(n)$ in general case?

if $n = p_1^{e_1} \cdot p_2^{e_2} \cdot \cdots \cdot p_k^{e_k}$, where $p_1, \ldots, p_k$ are distinct primes, then

$$\phi(n) = (p_1 - 1) \cdot p_1^{e_1 - 1} \cdot \ldots \cdots (p_k - 1) \cdot p_k^{e_k - 1}$$

e.g., $n = 24 = 2^3 \cdot 3$, hence $\phi(24) = 1 \cdot 2^2 \cdot 2 \cdot 3^0 = 8$

The Euler's Totient Function

Special cases

- if n is prime, then $\phi(n) = n - 1$,
  - e.g., $n = 13$, $\phi(13) = 12$
- if $n = pq$ where p, q are different prime numbers, then $\phi(n) = (p-1)(q-1)$,
  - e.g., $n = 143 = 13 \times 11$, $\phi(143) = 120$
- if $n = p^a$ where p is prime then $\phi(n) = (p-1)p^{a-1}$,
  - e.g., $n = 9 = 3^2$, $\phi(9) = (3-1) \times 3^1 = 6$

Modular Exponentiation

Basic idea

Given number a, b, n, compute $a^b \bmod n$

Two ways to compute it

- Brute force: $\overbrace{a \cdot a \cdot \cdots \cdot a}^{b} \bmod n$ - slow for large value of b!
- Repeated squaring:
  1. write the exponent b in binary
  2. ignore the first digit
  3. read the digit from left to right: if 0 then square the number; if 1 then square the number and multiply the result by a

Modular Exponentiation

Example: 3^{11} by repeated squaring

- write 11 in binary: 1011_2.
- ignore the first digit, so we have: 011.
- now from left to right we see:
  - 0, so we square the number: $3^2 = 9$ (1 multiplication)
  - 1, so we square the number and then multiply with 3: $3^5 = 9^2 \times 3 = 81 \times 3 = 243$ (2 multiplications)
  - 1, so we square the number and then multiply with 3: $3^{11} = 243^2 \times 3 = 59049 \times 3 = 177147$ (2 multiplications)
- we computed 3^{11}, with only 5 multiplications instead of 10.

Modular Exponentiation

Example: 3^{11} by repeated squaring

- We can represent this example in a table:

b	1	0	1	1
z	3	$3^2 = 9$	$9^2 \times 3 = 243$	$243^2 \times 3 = 177147$

- We can reduce the intermediate results modulo n in modular arithmetic

b	1	0	1	1
z	3	3^2=9 mod 12 =9	$9^2 \times 3$=243 mod 12 =3	$3^2 \times 3$ mod 12 =3

Outline

RSA: Rivest-Shamir-Adleman

History

- Clifford Cocks, a British mathematician, described an equivalent system in an internal document in 1973 (not revealed until 1998 due to its top secret classification).
- Rivest, Shamir, and Adleman devised RSA independently of Cocks' work.
- The RSA algorithm was publicly described in 1978 by Ron Rivest, Adi Shamir, and Leonard Adleman at MIT

RSA: Rivest-Shamir-Adleman

Operation

The RSA algorithm involves three steps:

- key generation
- encryption
- decryption

RSA: overview

The keys

- Keys: public key (encryption), and private key (decryption)
- Keys generation:
  - Choose two different large random prime numbers p, q
  - Compute $n = p \times q$
  - Calculate totient $\phi(n) = (p - 1) \times (q - 1)$
  - Choose the encryption key e such that $e < \phi(n)$, and e and $\phi(n)$ are relatively prime
  - Compute the decryption key d such that $ed = 1 \bmod \phi(n)$
    - d is the inverse of e modulo $\phi(n)$, can be calculated by using Extended Euclid's Algorithm (EEA)
  - e and n are public, and d is private

An online inverse of modulo calculator: `https://planetcalc.com/3311/`

RSA: Encryption and Decryption

Encryption of a message m

1. Divide m into numerical blocks m_i which is smaller than n
2. compute $c_i = m_i^e \bmod n$
3. aggregate all c_i to obtain the final ciphertext c (represented by a set of blocks (c_i)).

Decryption

1. Obtain c in numerical blocks c_i
2. Compute $m_i = c_i^d \bmod n$
3. Aggregate all m_i to obtain the plaintext m.

RSA: how it works

Why the RSA decipher works?

$$
\begin{aligned}
M^{\phi(n)} &\equiv 1 \pmod{n} && \text{(by Fermat's little theorem)} \\
M^{k\cdot\phi(n)} &\equiv 1 \pmod{n} \\
M \cdot M^{k\cdot\phi(n)} &\equiv M \cdot 1 \pmod{n} \\
M^{k\cdot\phi(n)+1} &\equiv M \pmod{n} \\
M^{ed} &\equiv M \pmod{n} && (\text{since } ed \equiv 1 \pmod{\phi(n)})
\end{aligned}
$$

RSA: an example

A complete example: key, encryption and decryption

1. Consider $p = 37$ and $q = 59$ then

$$n = p \times q = 2183 \text{ (PUBLIC)}$$

   and we compute

$$\phi(n) = (p-1) \times (q-1) = 36 \times 58 = 2088$$

2. Randomly pick up $e = 83$ (PUBLIC) then $d = 1283$, such that $83 \times d = 1 \bmod 2088$ by using Extended Euclid's Algorithm (EEA).
   - you could also use an online inverse of modulo calculator, e.g., : `https://planetcalc.com/3311/`

RSA: an example

Example: key, encryption and decryption

3. Consider $m = 57747639217438$, $c = E_{RSA(83,2183)}(m)$:
   - $m_1 = 577$, $m_2 = 476$, $m_3 = 392$, $m_4 = 174$, $m_5 = 038$ (padding)
   - $c_1 = 577^{83} \bmod 2183 = 425$
   - $c_2 = 476^{83} \bmod 2183 = 1478$
   - ... ...

   $e = 83 = \langle 1010011 \rangle_2$, we compute c as follows:

i	6	5	4	3	2	1	0
b_i	1	0	1	0	0	1	1
m_1	577	1113	938	95	293	420	$425 \to c_1$
m_2	476	1727	316	1621	1492	94	$1478 \to c_2$
$\vdots$		$\vdots$				$\vdots$	

You could also use the online PowerMod calculator to compute these, e.g., :

`https://www.mtholyoke.edu/courses/quenell/s2003/ma139/js/powermod.html`

RSA: an example

Example: key, encryption and decryption

4. Consider $c = 425, 1478, 1350, ...,\ m = D_{RSA}(c)$
   - $m_1 = 425^{1283} \bmod 2183 = 577$
   - $m_2 = 1478^{1283} \bmod 2183 = 476$
   - ... ...

Exercise

- Compute c_3, c_4 and c_5
- Compute m_3, m_4 and m_5

Outline

Summary

This lecture

- Mathematical background
- RSA: key generation, encryption and decryption

Next lecture

- Diffie-Hellman key exchange
- ElGamal: key generation, signature, encryption and decryption