ITS64304 Theory of Computation

School of Computer Science
Taylor's University Lakeside Campus

Lecture 9: Cryptography

Learning outcomes

At the end of this topic students should be able to:

- Describe a public-key cryptosystem.
- Describe the key aspects of the RSA system.
- *Explain the significance of intractable problems in public-key cryptosystems.**

* Relates to Course Learning Outcome 3

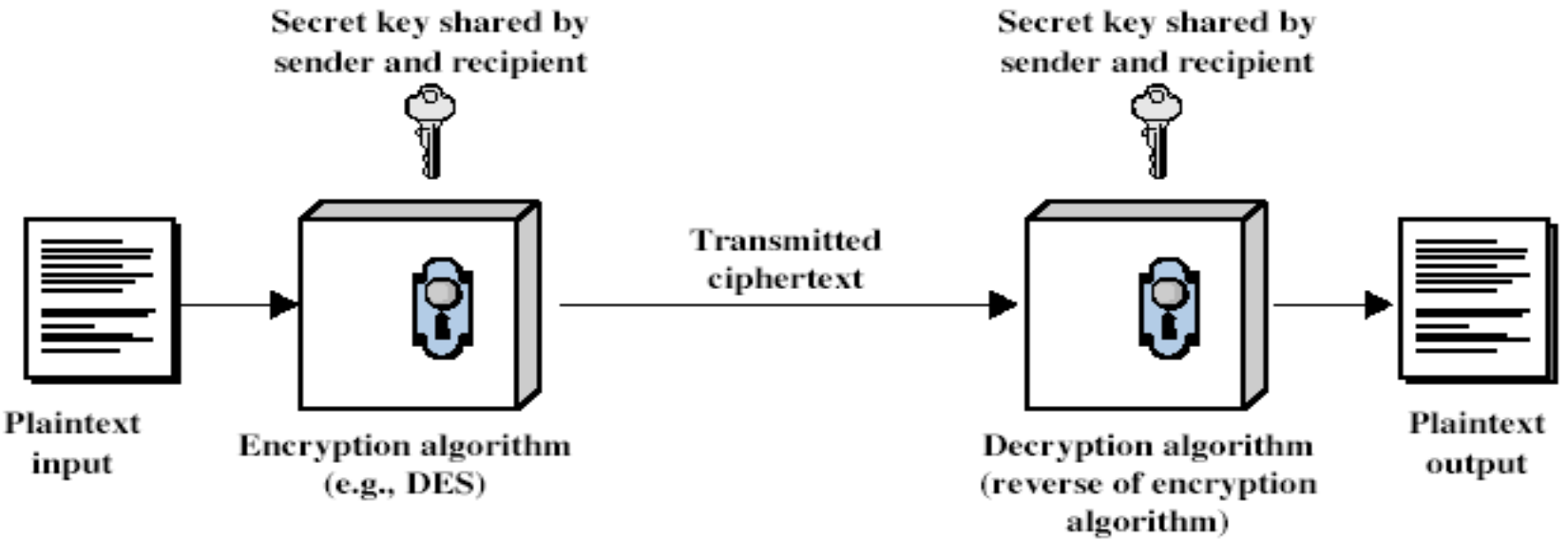
Cryptography

- Problems which cannot easily be solved can be useful.
  - Intractable problems

- Consider the problem of encryption.
  - Write Message, **M**, as a (large) number -- encoding letters as numbers as in ASCII.
  - Message M, encrypted by applying **E**,
    - encrypted message is **E(M).**
  - To decrypt, we apply a decryption key, **D**,
    - **M = D(E(M)).**

- The idea is that no “unauthorized” person can see the message M.
  - ensure that “cracking” the system is sufficiently hard

Cipher Model



- Classical examples: *Caesar cipher* (*Monoalphabetic cipher), Vigenere Cipher (Polyalphabetic Cipher), Playfair Cipher* (Multiple letter Cipher)
- Modern examples: *Data Encryption Standard (DES), AES, Blowfish, RC5*

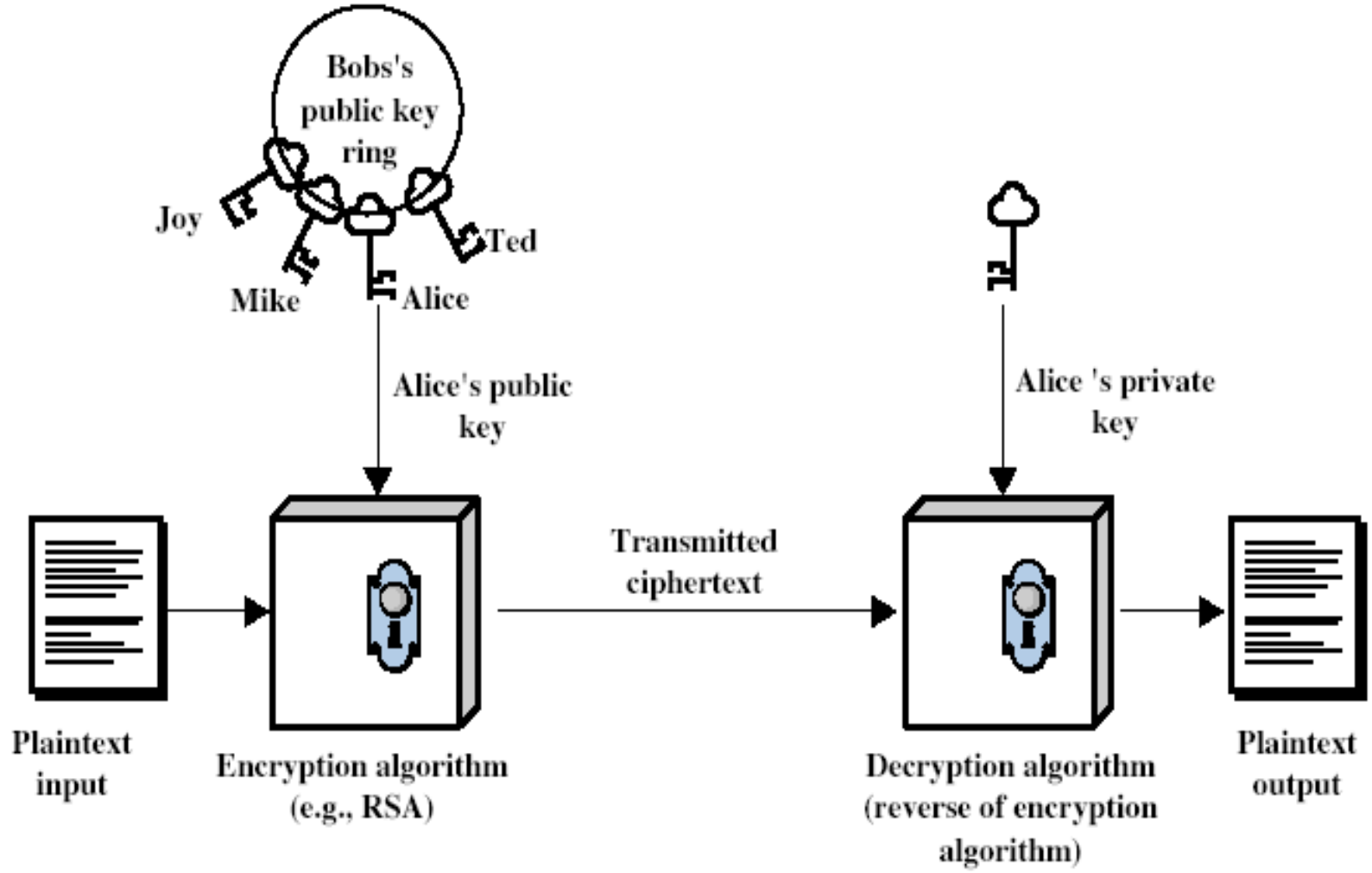
Symmetric Keys

- Symmetric key algorithms/ciphers
  - Keys are related for both encryption and decryption
  - There is secret key, single key, shared key or only one key
  - The key should be shared secretly by both sender and receiver

Public key cryptosystems

- Used to ensure secure communication between two arbitrary users.
- **public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - a **public-key**,
    - which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
    - encryption keys **E** published or made public
  - a **private-key**,
    - known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**
    - decryption keys **D** are kept secret
  - is **asymmetric** because
    - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures
- Hence we publish E, and keep D secret.
- To work, this requires that D be not deducible from E.

Public-Key Cryptography



Public key cryptosystems

- We can ensure D (private key) is not (easily) deducible from E (public key) if the task of **computing D from E** is **intractable**.

- This gives us confidence that **no-one** can find D from E.
  - D must not be (easily) deduced from E
  - Increase the numbers until this happens
  - Computing E(M) must be simple enough

- Hence, we want E(M) to be a **one-way trapdoor function**.

RSA cryptosystem

- RSA is a way of finding appropriate one-way trapdoor functions
- Prime number (divisible only by 1 and itself) – form the basis of the RSA algorithm
- Underlying technique
  - it is easy to find and multiply large prime numbers together (p x q = n)
  - but extremely difficult to factor their product n.

The RSA Algorithm

1. Choose two large prime numbers, **P** and **Q (private, chosen)** typically 1024 bits
2. Calculate N = P x Q (public, calculated)
3. Calculate ø(n)=(p-1)(q-1)
4. Select the public key (i.e. the encryption key) **E** such that

   gcd(ø(n), E) = 1; 1< E < ø(n) (public, chosen)
5. Select the private key (i.e. the decryption key) **D** such that the following equation is true: (private, calculated)

   (D x E) mod ø(n) = 1, equivalently, $D = E^{-1} \bmod ø(n)$
6. For **encryption**, calculate the cipher text CT from the plain text PT as follows: $CT = PT^E \bmod N$
7. Send CT as the cipher text to the receiver
8. For **decryption**, calculate the plain text PT from the cipher text CT as follows: $PT = CT^D \bmod N$

RSA Example

1. Select primes: $p=47$ & $q=71$
2. Compute $n = pq = 47 \times 71 = 3337$
3. Compute $\varnothing(n)=(p-1)(q-1)=46 \times 70 = 3220$
4. Select **E** : $\gcd(\mathbf{E}, 3220)=1$; choose **E**$=79$
5. Determine **D**: $\mathbf{(Dx79) \bmod 3220 = 1}$

   Value is **D**$=1019$ since $1019 \times 79 = 80501 \bmod 3220 = 1$
6. Publish public key $\{79, 3337\}$
7. Keep secret private key $\{1019, 47, 71\}$

RSA cryptosystem - revisited

- Chooses two large primes p and q
- Compute $n = p \times q$ and $r = (p-1) \times (q-1)$
- Find e such that e and r are co-prime, where $1<e< ((p-1) \times (q-1))$

- Find d such that; $(d \times e) \bmod ((p - 1) \times (q - 1)) = 1$ or
  $d \times e \equiv 1 \pmod{r}$
- e and n are made public
- d, p and q are kept secret

Note: Two integers are said to be co-prime or relatively prime if they have a common factor as 1 or their greatest common divisor is 1

RSA cryptosystem

- To break this system, it is necessary to determine *d* from *n* and *e*.
- If we can factor *n* into *p* and *q*, *d* can easily be determined from *e*, *p* and *q* using Euclid's algorithm
- Fortunately, factoring large numbers or computing *d* from *e* and *n* (*p* and *q*) is intractable

More RSA Example

1. Select primes: $p=17$ & $q=11$
2. Compute $n = pq = 17 \times 11 =$
3. Compute $\phi(n) = (p-1)(q-1) = 16 \times 10 =$
4. Select **E** : $\gcd(E, 160) = 1$; choose **E**=
5. Determine **D**: ***(D x E) = 1 mod 160***

   Value is **D**=
6. Publish public key:
7. Keep secret private key:

RSA example

p = 3, q = 11
n = 3 x 11 = 33
z = 2 x 10 = 20

Number relatively prime to z = 7
e = (1 mod z)/d
7e = 1 mod 20
e = 3

$$C = P^e = P^3$$

$$P = C^d = C^7$$

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

Sender's computation

Receiver's computation

An example of the RSA algorithm.

RSA exercise…

- P = 7, Q = 17
  - Encrypt plaintext alphabet F = 6

N = 7*17 = 119

$\phi(n)$ = 6*16 = 96

E = GCD (96, E) = 1

E = 5

D = (D*5)mod 96 = 1

D = 77

Encryption: 6^5 mod 119 = 41

Decryption: 41^{77} mod 119 = 6

Conclusion

- Security depends on not being able to factor *n* in any reasonable time.
- By making *n* large enough to defeat the best algorithms known, the method is secure.
- **Factoring** is known to be a very **intractable** problem, and so it is unlikely that a "smart" algorithm will be able to defeat the system.
- Note: probabilistic methods can be used to easily find appropriate primes, and so setting up the system is computationally tractable.

Start Revising ToC in preparations for the final examination

Question

How could you write a program to determine when a decryption has succeeded?