

Cryptography and Network Security

PUBLIC KEY CRYPTOGRAPHY
RSA



Session Meta Data

Author	Dr T Sree Sharmila
Reviewer	
Version Number	1.0
Release Date	5 JuLY 2018

Revision History

Revision Date	Details	Version no.
		1.0

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

Introduction

Every Egyptian received two names, which were known respectively as the true name and the good name, or the great name and the little name; and while the good or little name was made public, the true or great name appears to have been carefully concealed.

—The Golden Bough, Sir James George Frazer

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

Private-Key Cryptography

- traditional **private/secret/single key** cryptography uses **one** key
- shared by both sender and receiver
- if this key is disclosed communications are compromised
- also is **symmetric**, parties are equal
- hence does not protect sender from receiver forging a message & claiming it sent by sender

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

Public-Key Cryptography

- probably most significant advance in the 3000 year history of cryptography
- uses **two** keys – a public & a private key
- **asymmetric** since parties are **not** equal
- uses clever application of number theoretic concepts to function
- complements **rather than** replaces private key crypto

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

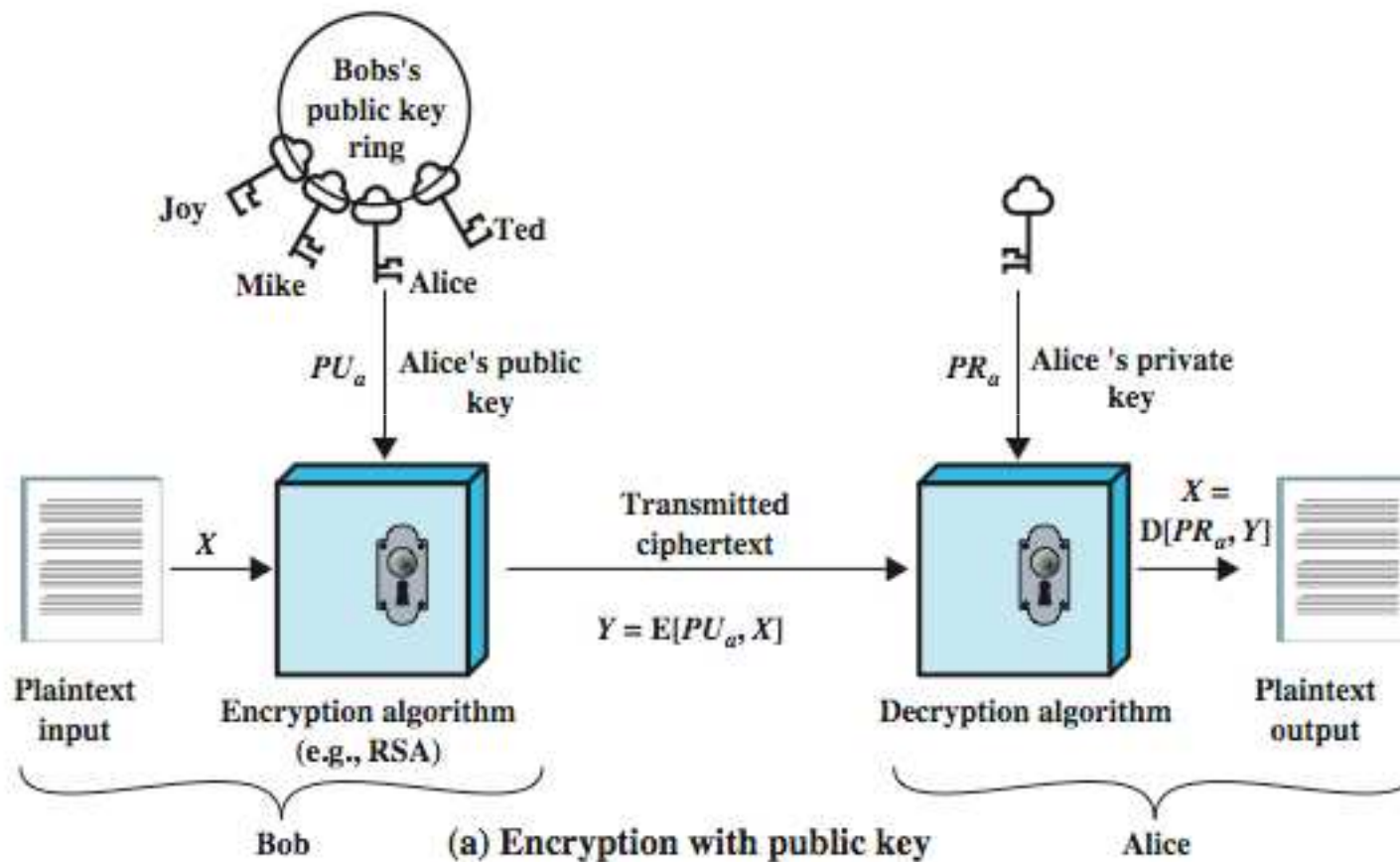
Why Public-Key Cryptography?

- developed to address two key issues:
 - **key distribution** – how to have secure communications in general without having to trust a KDC with your key
 - **digital signatures** – how to verify a message comes intact from the claimed sender
- public invention due to Whitfield Diffie & Martin Hellman at Stanford Uni in 1976
 - known earlier in classified community

Public-Key Cryptography

- **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**
 - a related **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**
- **infeasible to determine private key from public**
- **is asymmetric** because
 - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

Public-Key Cryptography



Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

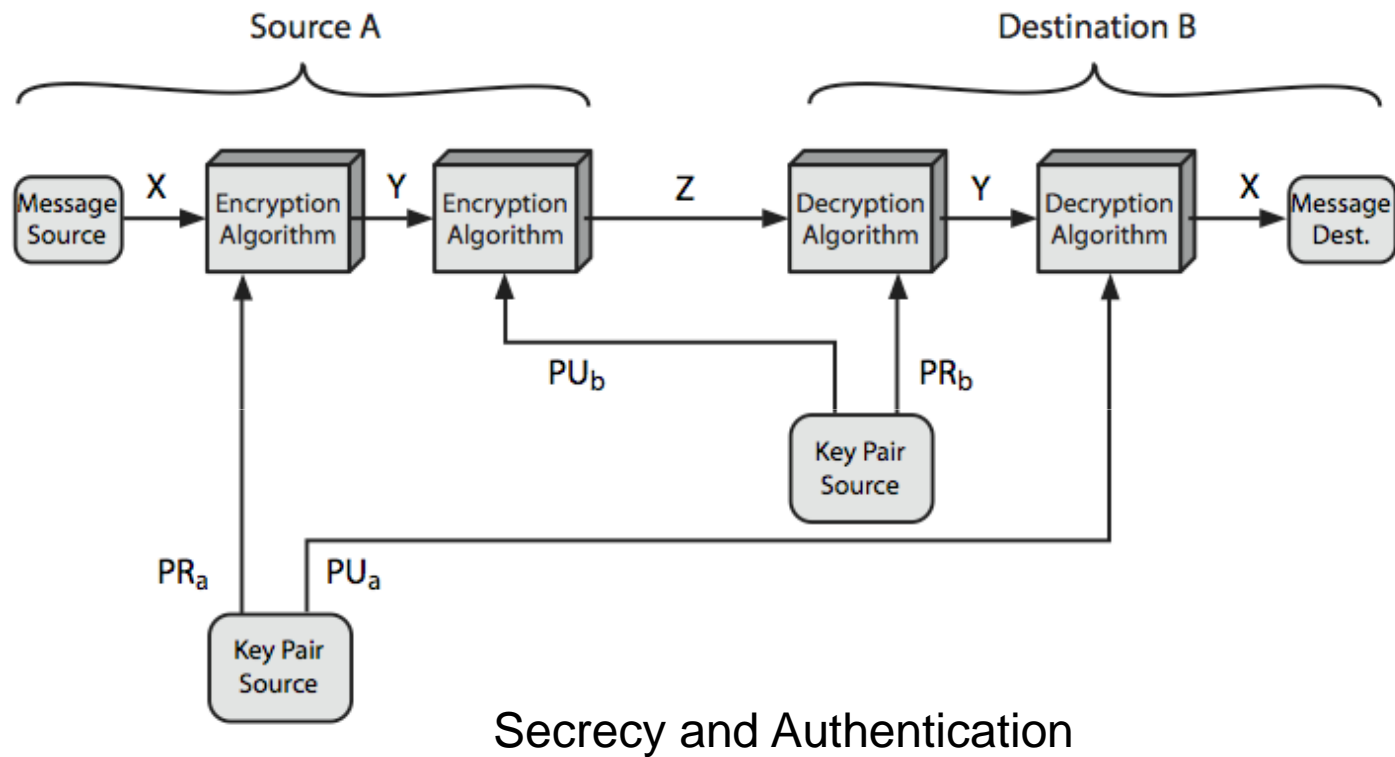
Symmetric vs Public-Key

Conventional Encryption	Public-Key Encryption
<p><i>Needed to Work:</i></p> <ol style="list-style-type: none">1. The same algorithm with the same key is used for encryption and decryption.2. The sender and receiver must share the algorithm and the key. <p><i>Needed for Security:</i></p> <ol style="list-style-type: none">1. The key must be kept secret.2. It must be impossible or at least impractical to decipher a message if no other information is available.3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	<p><i>Needed to Work:</i></p> <ol style="list-style-type: none">1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption.2. The sender and receiver must each have one of the matched pair of keys (not the same one). <p><i>Needed for Security:</i></p> <ol style="list-style-type: none">1. One of the two keys must be kept secret.2. It must be impossible or at least impractical to decipher a message if no other information is available.3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

Public-Key Cryptosystems



Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

Public-Key Applications

- can classify uses into 3 categories:
 - **encryption/decryption** (provide secrecy)
 - **digital signatures** (provide authentication)
 - **key exchange** (of session keys)
- some algorithms are suitable for all uses, others are specific to one

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

Public-Key Requirements

- Public-Key algorithms rely on two keys where:
 - it is computationally infeasible to find decryption key knowing only algorithm & encryption key
 - it is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
 - either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)
- these are formidable requirements which only a few algorithms have satisfied

Public-Key Requirements

- need a trapdoor one-way function
- one-way function has
 - $Y = f(X)$ easy
 - $X = f^{-1}(Y)$ infeasible
- a trap-door one-way function has
 - $Y = f_k(X)$ easy, if k and X are known
 - $X = f_k^{-1}(Y)$ easy, if k and Y are known
 - $X = f_k^{-1}(Y)$ infeasible, if Y known but k not known
- The development of a practical public-key scheme depends on a suitable trap-door one-way function

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

Security of Public Key Schemes

- like private key schemes brute force **exhaustive search** attack is always theoretically possible
- but keys used are too large (>512bits)
- security relies on a **large enough** difference in difficulty between **easy** (en/decrypt) and **hard** (cryptanalyse) problems
- more generally the **hard** problem is known, but is made hard enough to be impractical to break
- requires the use of **very large numbers**
- hence is **slow** compared to private key schemes

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

RSA

- by Rivest, Shamir & Adleman of MIT in 1977
- best known & widely used public-key scheme
- based on exponentiation in a finite (Galois) field over integers modulo a prime
 - exponentiation takes $O((\log n)^3)$ operations (easy)
- uses large integers (eg. 1024 bits)
- security due to cost of factoring large numbers
 - Factorization takes $O(e^{\log n \log \log n})$ operations (hard)

RSA En/decryption

- to encrypt a message M the sender:
 - obtains **public key** of recipient $PU = \{e, n\}$
 - computes: $C = M^e \bmod n$, where $0 \leq M < n$
- to decrypt the ciphertext C the owner:
 - uses their private key $PR = \{d, n\}$
 - computes: $M = C^d \bmod n$
- note that the message M must be smaller than the modulus n (block if needed)

RSA Key Setup

- each user generates a public/private key pair by:
- selecting two large primes at random: p, q
- computing their system modulus $n=p \cdot q$
 - note $\phi(n) = (p-1)(q-1)$
- selecting at random the encryption key e
 - where $1 < e < \phi(n)$, $\gcd(e, \phi(n)) = 1$
- solve following equation to find decryption key d
 - $e \cdot d = 1 \pmod{\phi(n)}$ and $0 \leq d \leq n$
- publish their public encryption key: $PU = \{e, n\}$
- keep secret private decryption key: $PR = \{d, n\}$

RSA Algorithm

Key Generation by Alice

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption by Bob with Alice's Public Key

Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption by Alice with Alice's Public Key

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

Why RSA Works

- because of Euler's Theorem:
 - $a^{\phi(n)} \bmod n = 1$ where $\gcd(a, n) = 1$
- in RSA have:
 - $n = p \cdot q$
 - $\phi(n) = (p-1)(q-1)$
 - carefully chose e & d to be inverses mod $\phi(n)$
 - hence $e \cdot d = 1 + k \cdot \phi(n)$ for some k

- hence :

$$\begin{aligned} C^d &= M^{e \cdot d} = M^{1+k \cdot \phi(n)} = M^1 \cdot (M^{\phi(n)})^k \\ &= M^1 \cdot (1)^k = M^1 = M \bmod n \end{aligned}$$

RSA Example - Key Setup

1. Select primes: $p=17$ & $q=11$
2. Calculate $n = pq = 17 \times 11 = 187$
3. Calculate $\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e : $\gcd(e, 160) = 1$; choose $e=7$
5. Determine d : $de = 1 \pmod{160}$ and $d < 160$
Value is $d=23$ since $23 \times 7 = 161 = 10 \times 160 + 1$
6. Publish public key $PU = \{7, 187\}$
7. Keep secret private key $PR = \{23, 187\}$

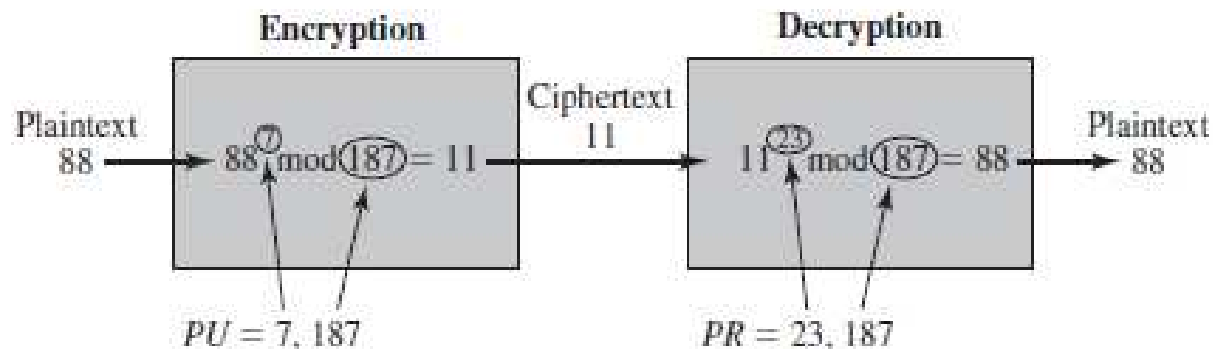
RSA Example - En/Decryption

- sample RSA encryption/decryption is:
- given message $M = 88$ (nb. $88 < 187$)
- encryption:

$$C = 88^7 \bmod 187 = 11$$

- decryption:

$$M = 11^{23} \bmod 187 = 88$$



Exponentiation

- can use the Square and Multiply Algorithm
- a fast, efficient algorithm for exponentiation
- concept is based on repeatedly squaring base
- and multiplying in the ones that are needed to compute the result
- look at binary representation of exponent
- only takes $O(\log_2 n)$ multiples for number n
 - eg. $7^5 = 7^4 \cdot 7^1 = 3 \cdot 7 = 10 \pmod{11}$
 - eg. $3^{129} = 3^{128} \cdot 3^1 = 5 \cdot 3 = 4 \pmod{11}$

Exponentiation

```
c = 0; f = 1
for i = k downto 0
    do c = 2 x c
        f = (f x f) mod n
    if  $b_i == 1$  then
        c = c + 1
        f = (f x a) mod n
return f
```

Efficient Encryption

- encryption uses exponentiation to power e
- hence if e small, this will be faster
 - often choose $e=65537$ ($2^{16}-1$)
 - also see choices of $e=3$ or $e=17$
- but if e too small (eg $e=3$) can attack
 - using Chinese remainder theorem & 3 messages with different moduli
- if e fixed must ensure $\gcd(e, \phi(n)) = 1$
 - ie reject any p or q not relatively prime to e

Efficient Decryption

- decryption uses exponentiation to power d
 - this is likely large, insecure if not
- can use the Chinese Remainder Theorem (CRT) to compute mod p & q separately. then combine to get desired answer
 - approx 4 times faster than doing directly
- only owner of private key who knows values of p & q can use this technique

RSA Key Generation

- users of RSA must:
 - determine two primes at random - p , q
 - select either e or d and compute the other
- primes p , q must not be easily derived from modulus $n=p \cdot q$
 - means must be sufficiently large
 - typically guess and use probabilistic test
- exponents e , d are inverses, so use Inverse algorithm to compute the other

RSA Security

- possible approaches to attacking RSA are:
 - brute force key search - infeasible given size of numbers
 - mathematical attacks - based on difficulty of computing $\phi(n)$, by factoring modulus n
 - timing attacks - on running of decryption
 - chosen ciphertext attacks - given properties of RSA

Factoring Problem

- mathematical approach takes 3 forms:
 - factor $n=p \cdot q$, hence compute $\phi(n)$ and then d
 - determine $\phi(n)$ directly and compute d
 - find d directly
- currently believe all equivalent to factoring
 - have seen slow improvements over the years
 - as of May-05 best is 200 decimal digits (663) bit with LS
 - biggest improvement comes from improved algorithm
 - cf QS to GHFS to LS
 - currently assume 1024-2048 bit RSA is secure
 - ensure p, q of similar size and matching other constraints

Timing Attacks

- developed by Paul Kocher in mid-1990's
- exploit timing variations in operations
 - eg. multiplying by small vs large number
 - or IF's varying which instructions executed
- infer operand size based on time taken
- RSA exploits time taken in exponentiation
- countermeasures
 - use constant exponentiation time
 - add random delays
 - blind values used in calculations

Chosen Ciphertext Attacks

RSA is vulnerable to a Chosen Ciphertext Attack (CCA)
attackers chooses ciphertexts & gets decrypted plaintext
back

choose ciphertext to exploit properties of RSA to provide
info to help cryptanalysis

can counter with random pad of plaintext

or use Optimal Asymmetric Encryption Padding (OASP)

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

Summary

- principles of public-key cryptography
- RSA algorithm, implementation, security

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

Test your understanding

1. What are the principal elements of a public-key cryptosystem?
2. What are the roles of the public and private key?
3. What are three broad categories of applications of public-key cryptosystems?
4. What requirements must a public-key cryptosystems fulfill to be a secure algorithm?
5. What is a one-way function?
6. What is a trap-door one-way function?
7. Describe in general terms an efficient procedure for picking a prime number.

Agenda

- Introduction
- Private key cryptography
- Public key cryptography
 - Why public key cryptography?
 - Symmetric Vs public key
 - Public key cryptosystems
 - Application
 - Requirement
 - Security
- RSA
- Summary
- Test your understanding
- References

References

1. William Stallings, Cryptography and Network Security, 6th Edition, Pearson Education, March 2013.
2. Charlie Kaufman, Radia Perlman and Mike Speciner, "Network Security", Prentice Hall of India, 2002.