Public key cryptology, RSA, ElGamal, Elliptic Curve

Jason Pearson and

November 27, 2015

Key Terms

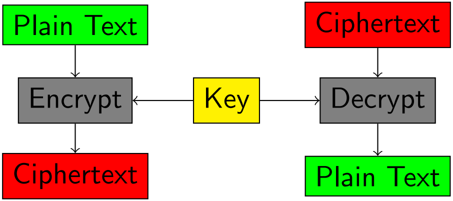
Plain text: typically a simple text such as this line
Cipher text: a message after it has been encrypted
Prime Number: a whole number that can only be divided evenly by one and itself also it is greater than one

Types of Encryption

Symmetric Encryption
Asymmetric Encryption
Hashing
Hybrid Encryption

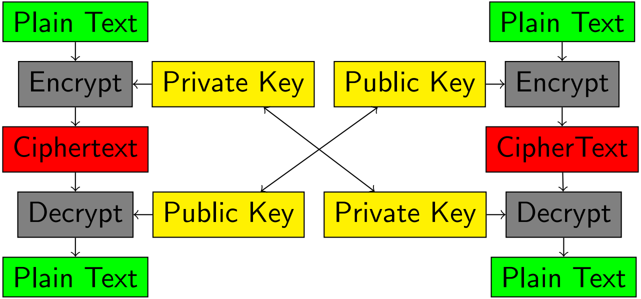
Symmetric Encryption

Encryption and Decryption use the same key



Asymmmetric Encryption

Public key and private key pair
Public key is used to encrypt a message
Private key is used to decrypt a message
Creating the key tends to be computationally expensive



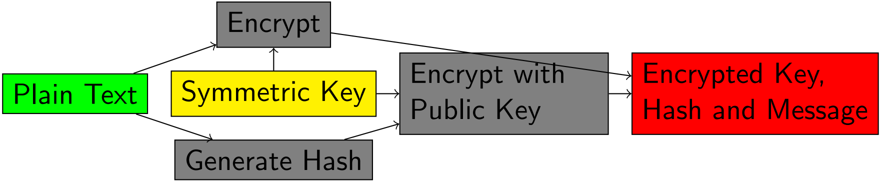
Hashing

We may or may not want to talk about this?

Hybrid Encryption

Uses ideas from symmetric and asymmetric encryption methods

An asymmetric cryptosystem is used for key encapsulation and an symmetric system is used for data encapsulation



Padding Schemes

RSA Cryptosystem

First designed in 1973 and declassified in 1997.
Named after its founders Ron Rivest, Adi Shamir and Leonar Adleman
Uses large prime numbers to create a private and public key
Security arises from the presumed difficulty of factoring large prime numbers

RSA Generating Public and Private Keys

Generate two large prime numbers and use to calculate n
Compute Euler's Totient Function
Create Public Key and Private Key

Generating Large Prime Numbers

Generate two large prime numbers.
Typically uses AKS testing and/or the Miller-Rabin test for prime numbers
These two values we will call p and q
$p = 991$
$q = 821$
We next calculate $n = p * q$
So we have $n = 991 * 821 = 813611$

Public Information	Secret Information
$n = 813611$	$q = 821$
	$p = 991$

We then can determine Euler's totient value by using the following equation.
$\phi(n) = \phi(p)\,\phi(q) = (p-1)(q-1) = n - (p+q-1)$
And for $\phi(n) = \phi(813611) = 811800$

Public Information	Secret Information
n = 813611	q = 821
	p = 991
	$\phi(n) = 811800$

Create Public and Private Keys

To create a public key we pick an arbitrary number e between $1 < e < \phi(n)$ for example we will use e $= 7423$
To create a private key we need to find the modular multiplicative inverse of e.
This is commonly done using the Extended Euclidean Algorithm.

$d \equiv e^{-1} \pmod{\phi(n)}$
This makes our value of d $= 788287$

Public Information	Secret Information
n = 813611	q = 821
e = 7423	p = 991
	$\phi(n) = 811800$
	d = 788287

Using these values we can create a cipher text c and decrypt it using the following equations
$c \equiv m^e \,(mod\,(n))$ and $m \equiv c^d \,(mod\,(n))$
Finishing up our example we will encrypt "Hi" using our new values
plain text m = 72105, e = 7423, d = 788287, n = 813611

Public Encrypt	Private Encrypt
$c \equiv m^e \,(mod\,(n))$	$c \equiv m^d \,(mod\,(n))$
$c \equiv 72105^{7423} mod(813611)$	$c \equiv 72105^{788287} mod(813611)$
$c = 707473$	$c = 616895$

Private Decrypt	Public Decrypt
$m \equiv c^d \,(mod\,(n))$	$m \equiv c^e \,(mod\,(n))$
$m \equiv 707473^{788287} mod(813611)$	$m \equiv 616895^{7423} mod(813611)$
$m = 72105$	$m = 72105$

RSA Practical Usage

ElGamal Cryptosystem

This method for cryptogphy uses discrete logarithms with a large prime modulus

The first step in creating is to create a large prime number

Then we create a Public and Private key

These can be used to encrypt and decrypt information

Generating Large Prime Numbers

First we generate a large prime number p

For us p = 17

We then create a generator g of multiplicative group $\mathbb{Z}_p^*$ of integers modulo p

For this example g = 6

Public Information	Private Information
p = 17	
g = 6	

ElGamal Creating Public and Private Keys

We then select a private key a where $1 \leq a \leq p - 2$
For this example a $= 5$
We can then use this to generate the last part of the public key
$g^a \mathrm{mod} p = 6^5 \mathrm{mod} 17 = 7$

Public Information	Private Information
p $= 17$	a $= 5$
g $= 6$	
$g^a \mathrm{mod} p = 7$	

Encrypting a Message

We will have our message m = 13

A public sender to send a message to the private key holder picks a random value k for this example k = 10

We then compute $c_1 = g^k modp = 15$

Now $c_2 = m * g^k \mathrm{mod} p = 13 * 6^{10} mod 17 = 8$

Cipher text sent through c_1 and c_2 to private key holder

Public Information	Private Information
p = 17	a = 5
g = 6	
$g^a \mathrm{mod} p = 7$	

Decrypting a Message

First we must calculate the shared secret
$s = \left(c_1{}^a\right) * c_2 \mathrm{mod} p = \left(15^5\right) * 8 \mathrm{mod} 17 = 16$
We then take the modular inverse of s and multiply it by c_2
Finding the modular inverse is commonly done with the Extended Euclidean Algorithm
$m = \left(c_{2*} s^{-1}\right) \mathrm{mod} p = (8 * 8) \mathrm{mod} 17 = 64 \mathrm{mod} 17 = 13$

Public Information	Private Information
p = 17	a = 5
g = 6	
$g^a \mathrm{mod} p = 7$	

ElGamal Practical Usage

Cipher text double the size in bits than the message

Commonly used in hybrid Cryptosystems

Elliptic Curve

Conclusion

References

http://caislab.kaist.ac.kr/lecture/2010/spring/cs548/basic/B02.pdf
http://doctrina.org/Why-RSA-Works-Three-Fundamental-Questions-Answered.html
http://doctrina.org/How-RSA-Works-With-Examples.html