Assignment # 2

Generell Systemteori Med Tonvikt på styr- och kontrollfunktioner
(SÄK1a/2I1501del1/2I4118)

Public Key CryptoSystems & RSA Algorithm

Batch: 2005-2006

Group Members
Sadeeq Jan
Charu Gupta
Tariq Saeed

Table of Contents

INTRODUCTION

Cryptography is the art of achieving security by encoding the messages to make them non readable. There are two cryptographic mechanisms depending upon the keys used. If the same key is used for encryption and decryption, we call the mechanism as Symmetric Key Cryptography and if two different keys are used in cryptographic mechanism, wherein one key is used for encryption and another, different key is used for decryption, we call the mechanism as Asymmetric Key Cryptography. Although Symmetric key cryptography is fast and efficient, it suffers from a disadvantage of key exchange. The sender and the receiver share the same key and it is difficult to agree upon the key without letting anyone know about it. Asymmetric key algorithm solves this problem by using two keys-**private key** which remains with the party and the **public key** which is shared by everyone. To communicate securely over any network, all one needs here is to publish one's public key. All these public keys can then be stored in a database that anyone can consult and the private key remains only with the respective individual.

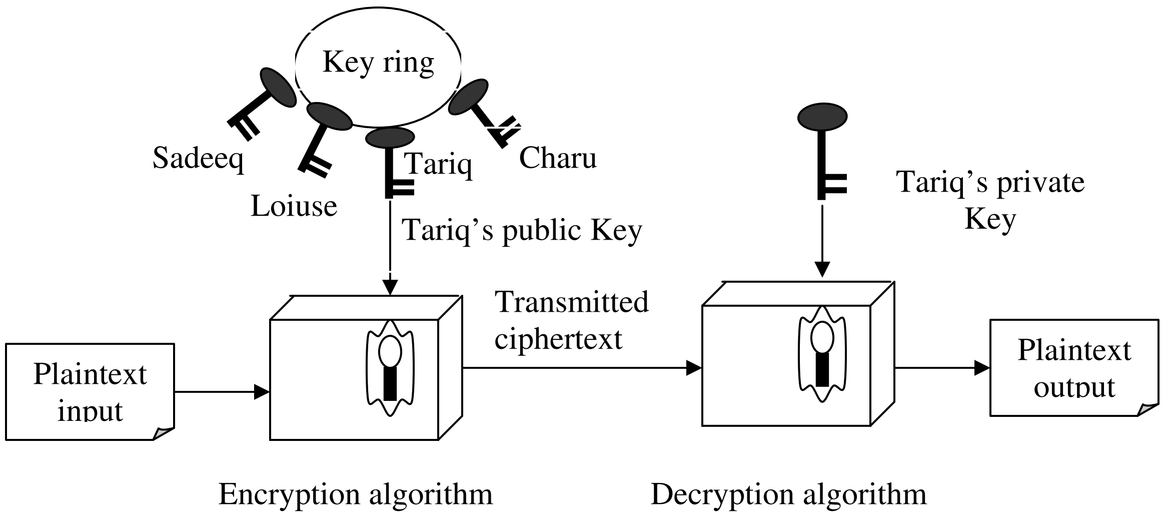
1.1 History of Asymmetric-Key Cryptography

In mid 1970s Whitfield Diffie, a student of Stanford University and Martin Hellman, her professor began to think about the problem of key exchange. After some research and complicated mathematical analysis, they came up with the idea of asymmetric key cryptography and they are now regarded as the fathers of asymmetric key cryptography. However there is lot of debate regarding who should get the credit for asymmetric key cryptography. It is believed that James Ellis of the British Communications Electronic Security Group (CSEG) proposed the idea of asymmetric key cryptography in the 1960's.His ideas were based on an anonymous paper written by at the Bell Labs during the Second World War. However Ellis could not devise a practical algorithm based on his ideas. He met Clifford Cocks, who joined CSEG in 1973 and both of them together came come with the practical algorithm that could work. The following year Malcolm Williamson also working in CSEG developed an asymmetric key cryptographic algorithm. However, CSEG was a secret agency and there findings were never published so these people never got the credit they deserved. Simultaneously the US National Security Agency (NSA) was also working on asymmetric key cryptography and it is believed that the NSA system based on asymmetric key cryptography was operational in the mid 1970's.

Based on the theoretical framework of Diffie and Hellman in 1977, Ron Rivest , Adi-Shamir and Len Adleman at MIT developed the first major asymmetric key cryptography system and published their results in 1978. This method is called as RSA algorithm. The name RSA comes from the first letters of the surnames of the three researchers. Even today RSA is the most widely accepted public key solution. It solves the problem of key agreements and distribution.

1.2 How Asymmetric Key Cryptography Works

In Asymmetric key Cryptography, also called as Public Key Cryptography, two different keys (which form a key pair) are used. One key is used for encryption and only the other corresponding key must be used for decryption. No other key can decrypt the message – not even the original (i.e. the first) key used for encryption. Every communicating party needs just a key pair for communicating with any number of other communicating parties.
The algorithm works as shown in the figure.



If one party wants to send a message to other party, then the 1st party (sender) must have the public key of the 2nd party (receiver). For example Sadeeq wants to send a message to Tariq, Sadeeq first encrypts the message with Tariq's Public Key and then transmits it. Tariq receives the encrypted message and decrypts it with its own private key. No one except Tariq can decrypt the message because no one knows the private key of Tariq.

There is a simple mathematical basis for this scheme. If one has an extremely large number that has only two factors which are prime numbers, one can generate a pair of keys. For example, consider a number 10. The number 10 has only two factors, 5 and 2, If we apply 5 as an encryption factor; only 2 can be used as the decryption factor. Nothing else- even 5 cannot do the decryption. Since 10 is a small number it can be broken easily however for large numbers even years of computation cannot break the scheme.
Suppose A wants to send a message to B without having to worry about its security. Then A and B should each have a private key and a public key.

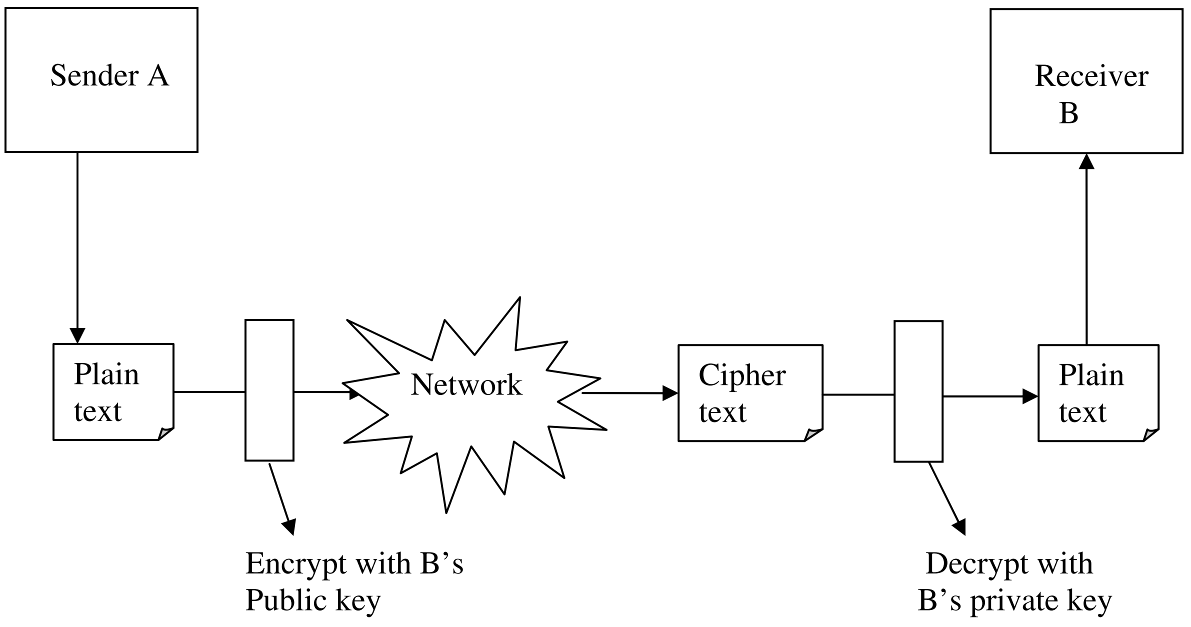
- A should keep her private key secret.
- B should keep her private key secret.
- A should inform B about her public key.
- B should inform A about her public key.

Thus we have the following matrix

Key Details	**A** Should Know	**B** Should Know
A's Private key	Yes	No
A's Public Key	Yes	Yes
B's Private Key	No	Yes
B's Public Key	Yes	Yes

Asymmetric key cryptography works as follows:

1. When A wants to send a message to B, A encrypts the message using B's public key. This is possible because A knows B's public key.
2. A sends the message encrypted with B's public key to B.
3. B decrypts A's message with B's private key. Only B knows her private key and the message could be decrypted only with B's private key. No one else would be able to make the sense of the message even if one is able to intercept the message.



Similarly when B wants to send a message to A, exactly reverse steps take place. B encrypts the message using A's public key. Therefore, only A can decrypt the message back to its original form, using her private key.

1.3 Comparison between Asymmetric Key Cryptography and Asymmetric-Key

Characteristic	Symmetric	Asymmetric
Key used for en/decryption	same for both	two keys used.One for Encryption and one for decryption.
Speed of en/decryption	Very Fast	Slower
Size of Resulting encrypted text	usually same or less than original	More than original
Key agreement/Exchange	Big Problem	No Problem
No. of keys required	Equal about the square of number of participants	Same as number of participants
Use	Mainly for encryption and Decryption, cannot be used For digital signatures.	Can be used for encryption and Decryption as well as Digital Signature

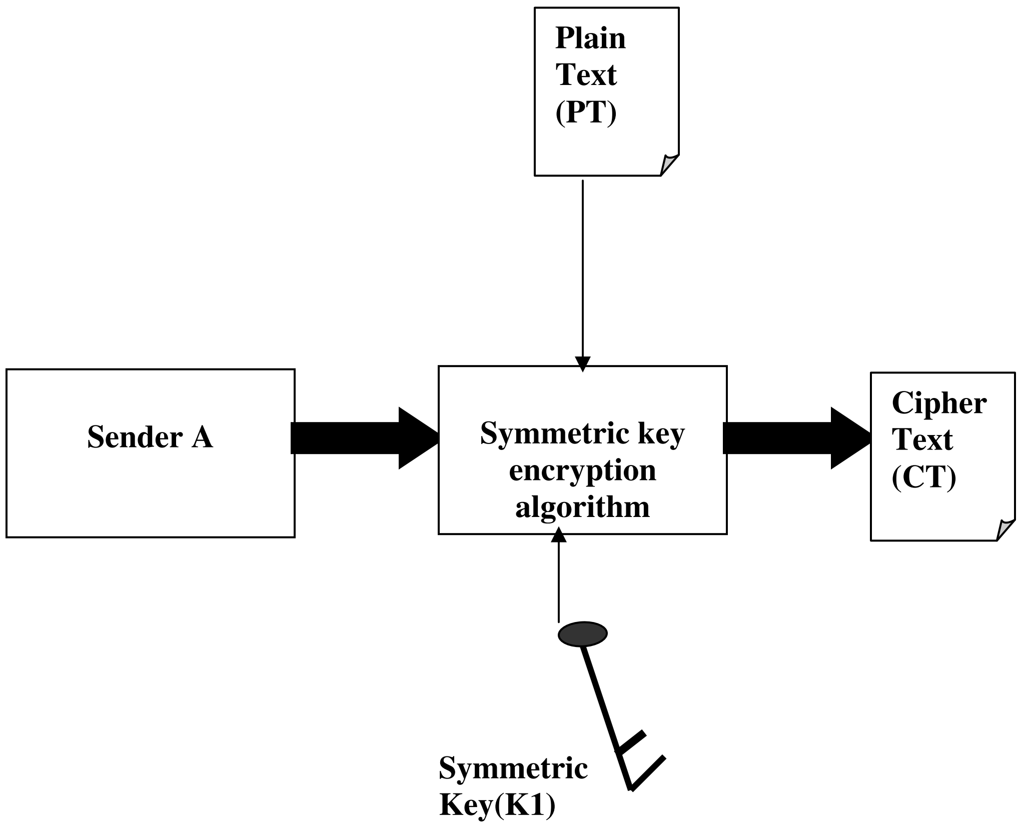
1.4 The Best of Both

It would be really wonderful if both the cryptography techniques are combined so that we can achieve the better of the two and do not compromise on any feature. Most specifically we need to ensure that the following objectives are met:

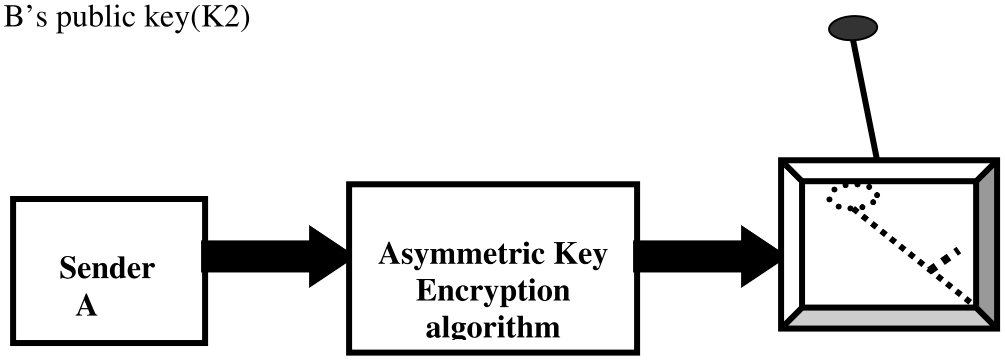
1. The solution should be completely secure.
2. The encryption and decryption process must not take a long time.
3. The generated cipher text should be compact in size.
4. The solution should scale to a large number of users easily without introducing any additional complications.
5. The key distribution problem must be solved by the solution.

In practice symmetric key cryptography and asymmetric key cryptography are combined to have a very efficient security solution. The way it works is as follows, assuming A is the sender of the message and B is the receiver.

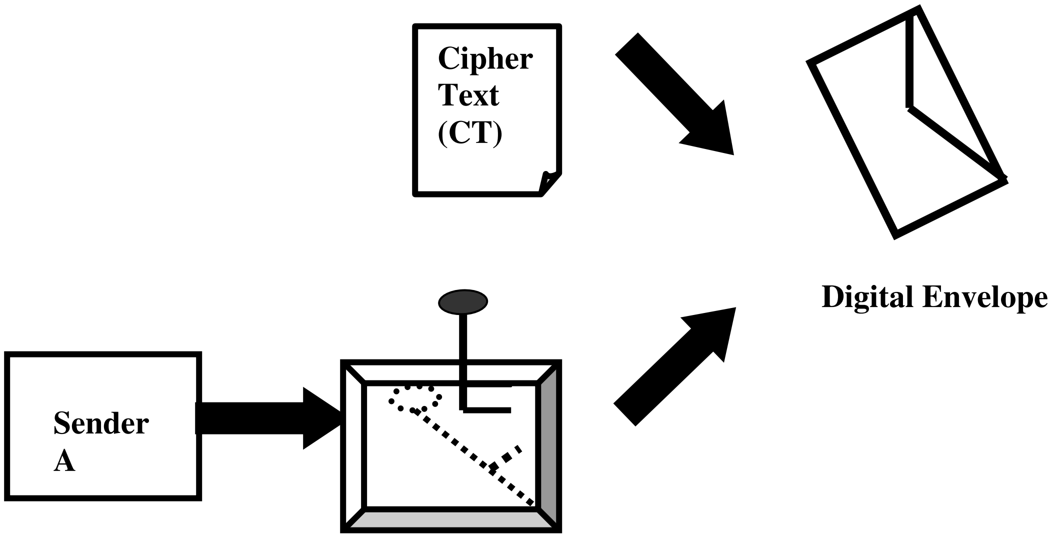
1. A encrypts the original plain text message with the help of standard key cryptography algorithm such as DES, IDEA etc. This produces cipher text message. The key used in this operation is called as one time symmetric key as it is used and then discarded.



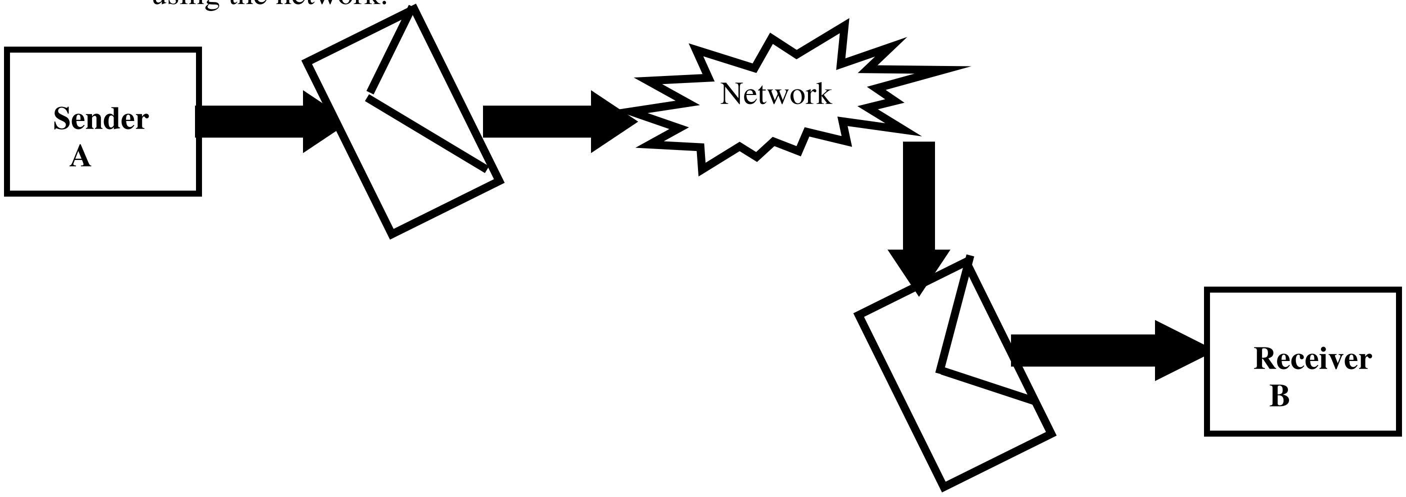
2. A now takes one time symmetric key of step 1(K1) and encrypts K1 with B’s public key (K2) .The process is called as key wrapping of the symmetric key. As shown in the figure the symmetric key K1 goes inside the box which is sealed by B’s public key(K2)



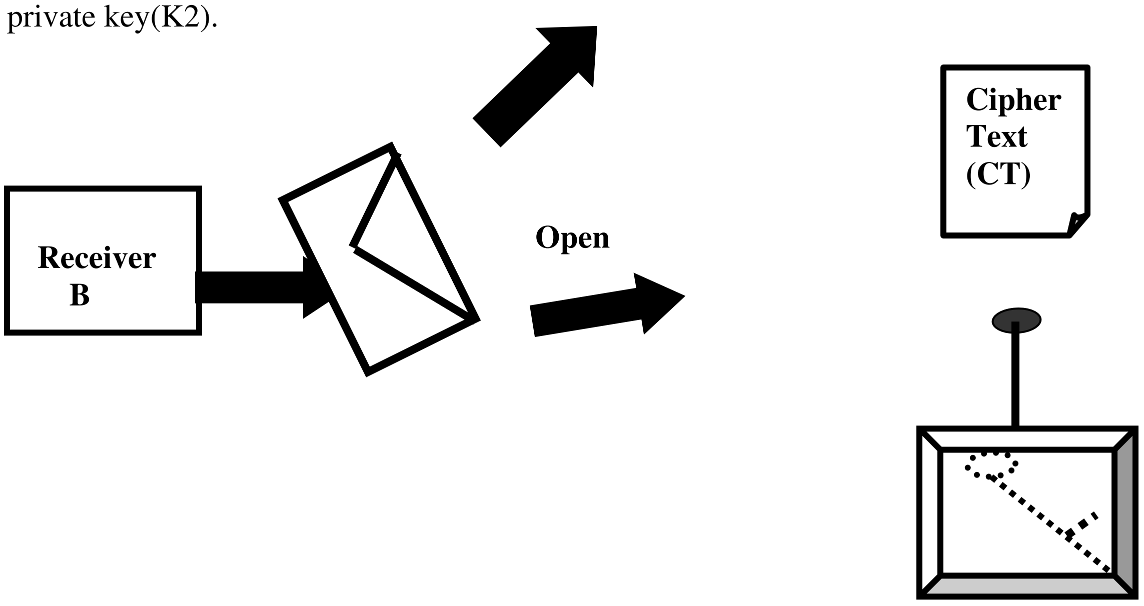
3. Now A puts the cipher text (CT) and the encrypted symmetric key together inside a digital envelope.



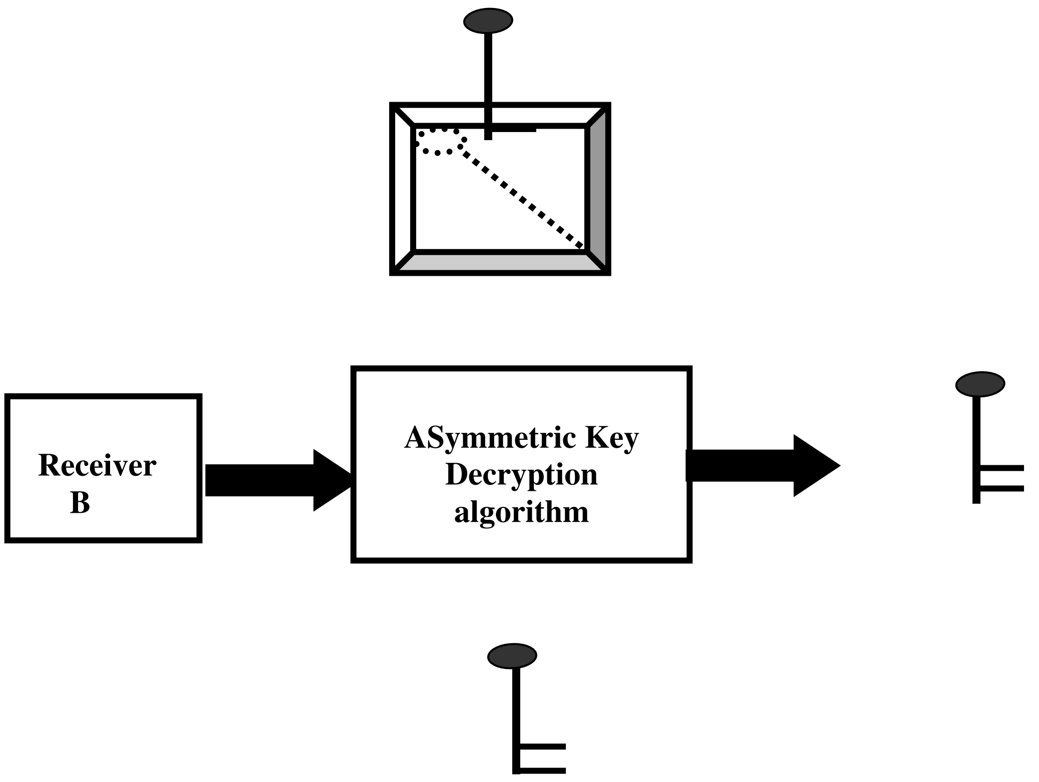
4. The sender (A) now send the digital envelope which contains the cipher text (CT) and the one time symmetric key (K1) encrypted with B's public key (K2) to B using the one time symmetric key (K1) encrypted with B's public key,(K2) to B using the network.



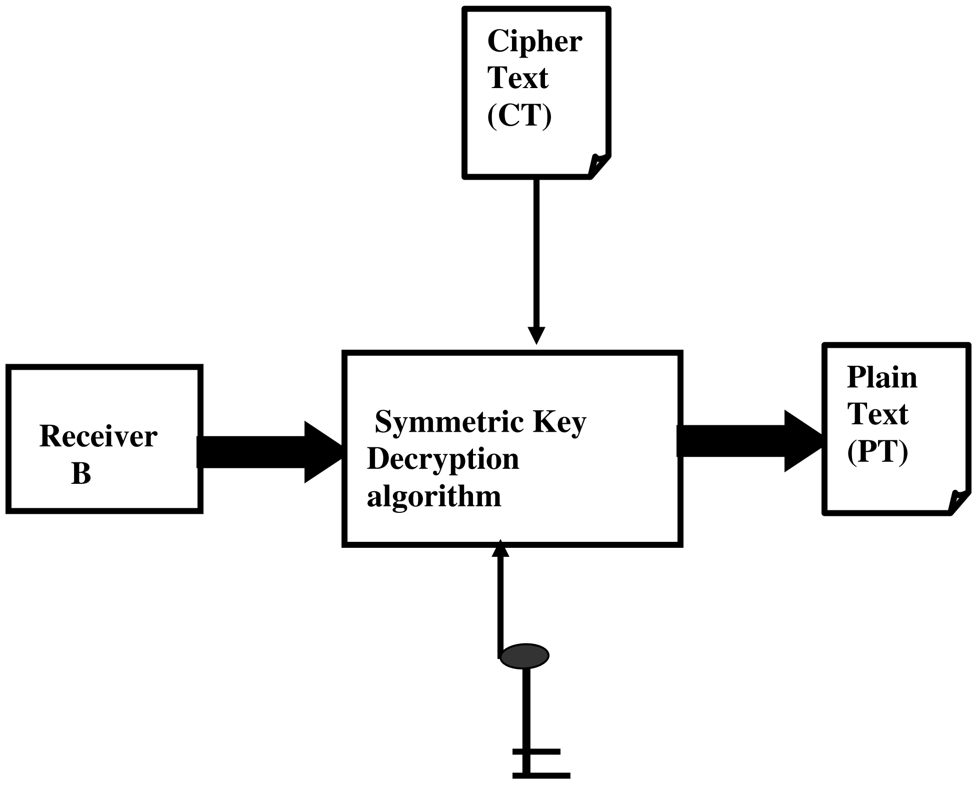
5. B receives and opens the digital envelope. After B opens the envelope, it receives two things: Cipher text (CT) and the one time session key (K1) encrypted using B's private key(K2).



6. Now using the same asymmetric key algorithm as used by A and her private key (K3) to decrypt the logical box that contains the symmetric key(K1) which was encrypted with B's public key(K2). Thus the output of the process is the one time symmetric key K1.



7. Finally B applies the same symmetric key algorithm as used by A and the symmetric key K1 to decrypt the cipher text CT). This process yields the original plain text (PT).



The algorithm is efficient due to following reasons.

1. Firstly we encrypt the plain text (PT) with the one time session key using a symmetric key cryptographic algorithm.The symmetric key encryption is fast and the generated text is of same size as original text. If we had used asymmetric key encryption the operation would have been quite slow and the encrypted text would have been of greater size.
2. We encrypted the one time session key with B's public key. Since size of K1 is going to be small, this asymmetric encryption process would not take long and the resulting encrypted key would not also consume a lot of space.
3. We have been able to solve the problem of key exchange with this scheme without losing the advantage of either symmetric or asymmetric key cryptography.

1.5 Requirements of Public Key Cryptosystem

The Public Key Cryptosystem depends on a cryptographic algorithm based on two related keys. Diffie and Hellman postulated this system without demonstrating that such algorithms exist. However they did lay out the conditions that such algorithms must fulfill. [1]

1. it is computationally easy for a party B to generate a pair(public key $KU_{b,}$ private key KR_b)
2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext.
   $$C=E_{KUb}(M)$$
3. It is computationally easy for a receiver B to decrypt the resulting ciphertext using the private key to recover the original message.
   $$M=D_{KRb}(C)= D_{KRb}[E_{KUb}(M)]$$
4. It is computationally infeasible for an opponent, knowing the public key, $KU_{b,}$ to determine the private key, KR_b.
5. It is computationally infeasible for an opponent, knowing the public key, KU_b, and a ciphertext, C, to recover the original message, M.

1.6 Applications for Public Key Cryptosystems

The public key system is characterized by the use of a cryptographic type of algorithm with two keys, one is public key and other is private key.
Depending on the application, sender uses either the sender's private key or the receiver's public key, or both, to perform some type of cryptographic function.

1.6.1 Encryption/Decryption

The sender encrypts a message with the recipient's public key. For example in disk encryption programs encrypt your entire hard disk so that you don't have to worry about unencrypted data on your disk.

Pretty Good Privacy is a software package originally developed by Phil Zimmerman. This software package provides encryption and authentication for e-mail and file storage applications. Zimmerman developed his freeware program using existing encryption techniques. It runs on multiple platforms. It provides message encryption, digital signatures, data compression, and e-mail compatibility. PGP uses the user's private key and user-supplied password to encrypt the file using IDEA. The receiver use same password and key to unlock the file.

1.6.2 Authentication/Digital signature

Authentication and digital signatures are a very important application of public-key cryptography in which sender “signs” a message with its private key. For example, if you receive a message from me that I have encrypted with my private key and you decrypt it using my public key, you should feel that the message did come from me. If I think that it necessary to keep the message secret, I may encrypt the message with my private key and then with your public key, that way only you can read the message, and you will know that the message came from me. The only requirement is that public keys are associated with their users by a trusted manner, for example a trusted directory. Due to this weakness, the standards community has used an object called a certificate. It contains, the certificate issuer's name, the subject name for which the certificate is being issued, the public key of the subject, and some time stamps. You know the public key is good, because the certificate issuer has a certificate too. [4]

1.6.3 Key exchange

We can use Asymmetric Key Cryptography for Key exchange. i.e. the session key can be transmitted using Asymmetric Key Cryptography.

1.6.4 Time Stamping

Time stamping is a technique that can certify that a certain electronic document was delivered at a certain time. Time stamping uses the blind signature scheme encryption model. Blind signature schemes allow the sender to get a message receipted by another party without revealing any information about the message to the other party.
Time stamping is working alike to sending a registered letter through the ordinary mail, but with additional level of proof etc recipient received a specific document. Following applications include patent applications, copyright archives, and contracts. [4]

1.6.5 Electronic Money

Electronic money is also called electronic cash or digital cash. It is still evolving. The transaction carried out electronically with a net transfer of funds from one party to another. It may be either debit or credit and can be either anonymous or identified. Anonymous applications work on blind signature schemes. Anonymous schemes are the electronic analog of cash, while identified schemes are the electronic analog of a debit or credit card.

Encryption is used for conventional transaction data in electronic money schemes like account numbers and transaction amounts. Digital signatures can replace handwritten signatures and public-key encryption can provide confidentiality. There are several systems of this range of applications, from transactions mimicking conventional paper transactions with values of several dollars and up. The various micropayment schemes that consignment extremely low cost transactions into amounts that will bear the overhead of encryption and clearing the bank. [4]

Some algorithms are suitable for all three applications; others can be used only for one or two of these applications. As shown in the following table:

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

1.7 Public key Cryptanalysis

In symmetric encryption, a public key encryption scheme is vulnerable to a brute force attack. The countermeasure is the same: Use large keys. Public key is depending on the use of some sort of invertible mathematical function. The complexity of calculating these functions may not scale linearly with the number of bits in the key but grow more rapidly than that. If you use the size of key large then brute force attack is impractical but small enough for practical encryption and decryption. If brute force is impractical to key size than encryption or decryption speed becomes too slow for general purpose use. Public key encryption is confined to key management and signature applications.

Another form of attack is to find some way to compute the private key given the public key. No one has proved yet that this form of attack is infeasible for a particular public key algorithm. The history of cryptanalysis shows a problem that seems insoluble from one perspective can be found to have a solution if looked at in an entirely different way.

Finally, there is a form of attack that is peculiar to public key systems. This is in essence, a probable message attack. For example a message was to be sent that consisted solely of a 56-bit DES key. A hacker could encrypt all possible keys using the public key and could decipher any message by matching the transmitted ciphertext. Thus, no matter how large the key size of the public key scheme, the attack is reduced to a brute force attack on a 56 bit key. This attack can be thwarted by appending some random bits to such simple messages.

THE RSA CRYPTOSYSTEM

RSA is a cryptosystem or means of transporting information in a secure and encrypted way.

It is based on the principles of public key cryptography. i.e. it uses two keys: Public Key and Private Key. Everyone which is involved in communication generates two keys. One key(Public Key) is sent to other parties involved in communication public and the other key is kept secret. If someone wants to send you an encrypted message, he must know your public key. He/She takes your public key, encrypts the message with it and then sends it to you. Unless someone is able to factor 128-bit numbers in less than 100 years, your message is relatively safe. After receiving the message, you decrypt the message using your private key. Someone else holding your public key will not be able to decrypt your message

2.1 RSA Algorithm

The RSA Algorithm was invented by Rivest, Shamir & Adleman of MIT in 1977. It is the best known & widely used public-key scheme. The details of the algorithm are as follows:

2.1.1 Key Generation

The first step of RSA Algorithm is key generation. Each user that wish to communicate must generate a public-private key pair. The following are the steps used for key generation. [1]

1. Select two large prime numbers. Lets call them 'p' and 'q' such that p != q. (at least 512 bits each)
2. Compute their system modulus N= p*q
   where $ø(N)=(p-1)(q-1)$)
3. Compute our 'n' to be n = p*q.
4. Select a small, odd integer (e) that is relatively prime to ø (N) and not 1
   where $1<e<ø(N)$, $\gcd(e,ø(N))=1$
5. Compute 'd' to be the multiplicative inverse of 'e' modulo ' ø (N)'
   Where $e.d=1 \bmod ø(N)$ and $0 \le d \le N$

6. The ordered pair (e, n) is your RSA public-key (Encryption Key). Publish this key.
7. The ordered pair (d, n) is your RSA private-key (Decryption Key). Keep Secret this key.
8. Make sure that 'p' and 'q' are completely annihilated. Otherwise the security of your cryptosystem may be compromised.

2.1.2. RSA Encryption

For encrypting a message M with RSA, the sender first obtains the public key of the recipient i.e. KU={e,N}. Once the sender has the public key of the recipient, he computes the encrypted message as follows

$C=M^e \bmod N$
where $0 \le M < N$

Thus C is the Encrypted message which is sent to the recipient over public network. No one other person except the original recipient can decrypt the encrypted message C to get the original message M.

2.1.3. RSA Decryption

When the recipient receives the encrypted message C, he/She can decrypt the message with his/her private key. He/She can decrypt it only if the message was encrypted with his/her Public Key. Thus the recipient uses his/her private key KR={d, N} to get the original message M in the following way.

M=Cd mod N
where $0 \le M < N$

The message M must be smaller than the modulus N

2.1.4. Example of RSA

Here is an example of RSA Algorithm.

1. Select primes: p=5 & q=11
2. Compute $n = pq$ =5×11=55
3. Compute ø(n)=(p–1)(q-1)=4×10=40
4. Select e *:* gcd(e,40)=1; choose e=3
5. Determine d*:* de=1 mod 40 and d < 40 Value is d=27 since 27×3=81= 40 x 2+1
6. Publish public key KU={3,55}
7. Keep secret private key KR={27,55}

After generating the keys, the encryption/decryption is performed as follows:

- given message M = **30** (30<55)
- encryption:
  C = 30^3 mod 55 = 27000 mod 55= **50**
- decryption:
  M = 50^27 mod 55 = **30**

2.2 The Security of RSA

Three possible approaches to attacking the RSA algorithm are as follows:

2.2.1 Brute force

A brute force attack involves of trying every possible code, combination, or password until you find the right one. Brute force attack is not an easy method. Following factors determining the difficulty of a Brute force attack:

1. How long can the key be?
2. How many possible values can each component of the key have?
3. How long will it take to attempt each key?
4. Is there a mechanism which will lock the attacker out after a number of failed attempts?

We explain it with example if a system which only allows 4 digits PIN codes. This means there are a maximum of 1000 possible PIN combinations. PIN security could be increased by increasing the length of the PIN, allowing the PIN to contain characters other than numbers, such as * or #, imposing a 30 second delay between failed authentication attempts and locking the account after 5 failed authentication attempts. A brute force attack will always succeed, eventually but brute force attacks against systems with sufficiently long key sizes may require billions of years to complete.

2.2.2 Mathematical attacks

There are several approaches, all equivalent in effect to factoring the product of two primes. The security of RSA depends of factoring being difficult. There are several methods to try factoring, but as long as the keys are long enough there is small risk of having your RSA encoded message broken. 256 bits can be broken relatively easily, 512 bits is probably insecure and breakable by major governments, 1024 bits should be secure for decades according to today's information. 2048 bits will most probably remain safe for a long time.

2.2.3 Timing attacks

These depend on the running time of the decryption algorithm. The way to break RSA is to find a technique to compute e^{th} roots mod n. Since $c = m$^e, the e^{th} root of c is the message m. This attack would allow someone to recover encrypted messages and forge signatures even without knowing the private key. This attack is not known to be equivalent to factoring. No methods are currently known that attempt to break RSA in this way.

CONCLUSIONS

Public Key Cryptography solves the problem of Key Distribution and verification of Messages from the sender which exists in Symmetric Key Ciphers. But the most of the algorithms (e.g. RSA) are slow as compared to DES and other Symmetric Ciphers. Thus Public Key Cryptography is most suitable for transmission of smaller messages (e.g. exchange of Session Keys).

A combination of Symmetric and Asymmetric Ciphers produces good results.

REFERENCE & BIBLIOGRAPHY

1. Cryptography and Network Security Third Edition by William Stallings
2. Cryptography and Network Security Third Edition by Atul Kahate
3. http://docs.sun.com/source/816-6154-10/contents.htm Date: 2005-09-15
4. http://www.eco.utexas.edu/faculty/Norman/BUS.FOR/course.mat/SSim/life.html Date: 2005-09-16