Performance Evaluation of Various Symmetric and Asymmetric Cryptosystems for Variable Text Size

Himani Agrawal[1], Dr. (Mrs.) Monisha Sharma[2]

[1]Associate Professor in E & Tc Department, SSGI (FET), Bhilai, Chhattisgarh, India

[2]Professor in E & Tc Department, SSGI (FET), Bhilai, Chhattisgarh, India

Abstract: ***Internet and networks application are growing very fast, so the need to protect such application are increased by using cryptographic methods. Many encryption algorithms are widely available and used in information security. They can be categorized into Symmetric (private) and Asymmetric (public) key encryption. Symmetric key ciphers use the same key for encryption and decryption, or the key used for decryption is easily calculated from the key used for encryption. Symmetric key ciphers can be broadly grouped into block ciphers and stream ciphers. Symmetric encryption has a troublesome drawback i.e., two people who wish to exchange confidential messages must share a secret key. The key must be exchanged in a secure way. Key distribution is difficult among the parties. One thing common to all private key cryptosystems is that if one knows how to cipher a message then he automatically knows how to decipher the message. Thus the method of ciphering must be kept secret. Once the ciphering method is compromised the cryptosystem is useless. Thus this cryptosystem is less secure but having high speed therefore frequently used for sending bulk messages. The ideas of the public key cryptosystems was first introduced by Diffie and Hellman [2] in 1976. Since then, several different protocols for public key cryptography have been presented e.g. RSA, ECC etc. Public key cryptosystems are highly secure but having less speed as compared to private key cryptosystems. Therefore generally used for sending short messages like secret keys. This paper presents the comparative study of some most popular Symmetric cryptosystems i.e. RC4, DES, 3DES, AES, Blowfish and some popular Asymmetric cryptosystems i.e. RSA, RSA PKCS1, RSA SAEP+, RSA OAEP, RSA OAEP+, React RSA, Hybrid React RSA and NTRU. These algorithms are compared on the basis of simulation time required for key generation, encryption and decryption for variable sized text data as input and the results were observed, analyzed and compared so as to identify which method is appropriate to the business needs.***

Keywords: Asymmetric Key, Asymmetric Key, DES, AES, 3DES, Blowfish, RC4, RSA, RSA family, NTRU.

1. Introduction

With the introduction of the computer, the need for automated tools for protecting files and other information stored on the computer became evident. This is especially the case for a shared system, such as a time sharing system, and the need is even more acute for systems that can be accessed over a public telephone network, data network, or the Internet [1]. In Symmetric keys encryption or secret key encryption, only one key is used to encrypt and decrypt data. The key should be distributed before transmission between entities. Keys play an important role. If weak key is used in algorithm then everyone may decrypt the data. Strength of Symmetric key encryption depends on the size of key used. For the same algorithm, encryption using longer key is harder to break than the one done using smaller key. There are many examples of this algorithm like DES, 3DES, RC4, Blowfish, and AES.

Asymmetric key encryption or public key encryption is used to solve the problem of key distribution. In Asymmetric keys, two keys are used; private and public keys. Public key is used for encryption and private key is used for decryption. Examples of this algorithm are RSA, ECC, XTR, NTRU. Because users tend to use two keys: public key, which is known to the public and private key which is known only to the user. There is no need for distributing them prior to transmission. However, public key encryption is based on mathematical functions, computationally intensive and is not very efficient for small mobile devices.

2. Methodology

A brief introduction of various cryptosystems is as follows.

Symmetric Cryptosystem:

DES: DES was created in 1972 by IBM, using the Data Encryption Algorithm. It was adopted by the U.S. Government as its standard encryption method for commercial and unclassified communications in 1977. DES begins the encryption process by using a 64-bit key. The NSA restricted the use of DES to a 56-bit key length, so DES discards 8-bits of the key and then uses the remaining key to encrypt data in 64-bit blocks. DES can operate in CBC, ECB, CFB, and OFB modes, giving it flexibility.

In 1998, the supercomputer DES Cracker, assisted by 100,000 distributed PCs on the Internet, cracked DES in 22 hours. The U.S. Government has not used DES since 1998[2].

3DES: DES was superseded by Triple DES (3DES) in November of 1998. 3DES is exactly what it is named – it performs three iterations of DES encryption on each block. It can do this in a number of ways, but the most common method is the Minus Encrypt-Decrypt-Encrypt (-EDE) method. Each iteration of 3DES using –EDE will encrypt a block using a 56-bit key. After encryption, use a different 56-bit key to decrypt the block. On the last pass, a 56-bit key is used to encrypt the data again. This is equivalent to using a 168-bit encryption key. Another method that can be used is Minus Encrypt-Encrypt-Encrypt (-EEE). This is three successive encryptions using a different 56-bit key. There are several keying methods that 3DES uses. All three keys can be independent of each other, or the first and third keys can be identical, with the second key being unique. All three keys can also be identical, which provides the least security, but is also the fastest to encrypt with. 3DES is still approved for use by U.S. Governmental systems, but has been replaced by the Advanced Encryption Standard (AES) [2].

AES: AES was approved for use by the U.S, Government for the encryption of sensitive, but unclassified data in 2000. It was developed by two Belgian cryptographers, Joan and Vincent Rijmen, Rijndael is a portmanteau of the names of the two inventors [3]. AES uses the Rijndael block cipher. Rijndael is a very resilient algorithm that has shown resistance to all known cryptographic attacks thus far. Rijndael key and block length can be 128, 192 or 256-bits. If both the key-length and block length are 128-bit, Rijndael will perform 9 processing rounds. If the block or key is 192-bit, it performs 11 processing rounds. If either is 256-bit, Rijndael performs 13 processing rounds. These round counts do not include an extra round of encipherment performed at the end.

Each processing round involves four steps:

1. Substitute bytes – Uses an S-box to perform a byte by byte substitution of the block.
2. Shift Rows – A simple permutation
3. Mix Column – A substitution method where data in each column from the shift row step is multiplied by the algorithm's matrix.
4. Add Round Key – The key for the processing round is XORed with the data. The algorithm itself is designed to exponentially increase the time it would take to mount a brute force attack on the cipher to several billion years [2].

BLOWFISH: It was developed by Bruce Schneier. It has a key length that can vary from 32 bits to a maximum of 448 bits (one to fourteen 32-bit words).That key is used to generate 18; 32 bit subkeys and four 8X 32 S-boxes containing a total of 1024 32-bit entries. The total is 4168 bytes. Blowfish is very fast since it encrypts data on 32-bit microprocessors at a rate of 18 clock cycles per byte, and can run in less than 5K of memory. The variable length key allows a tradeoff between speed and security. Blowfish is one of the most formidable conventional encryption algorithms. So far, the security of Blowfish is unchallenged [4].

RC4: It is probably the best known symmetric encryption algorithm that uses a stream cipher. It is designed in 1987 by Ron Rivest for RSA security [1]. RC4 keys can be anywhere from 1 to 2048 bits in length. These are used to initiate a 256-byte state table. This table is used by RC4 to generate a psudo-random byte stream used to encrypt the plaintext. This stream is transformed with the plaintext data using an XOR (Exclusive OR Boolean Operator). The function will result in "True" only when a plaintext bit and its corresponding psudo-random stream bit are opposite in value. In addition, every element in the state table is swapped at least once. This is what produces the ciphertext [2].

ASymmetric Cryptosystem

RSA: RSA is one of the oldest and the most widely used public key cryptographic algorithms. It was the first algorithm known to be suitable for signing as well as encryption. The system works on two large prime numbers, from which the public and private keys will be generated. RSA was developed by Ron Rivest, Adi Shamir, and Leonard Adleman, in 1977. RSA derives its name from the initials of the last name of each of its developers. It is commonly used with key strengths of 1024-bits, but its real strength relies on the prime factorization of very large numbers [2]. The RSA scheme is a block cipher in which the plaintext and the ciphertext are integers between 0 and n-1 for some modulus n.

RSA-PKCS1-v1.5: To overcome RSA from the simple CCA attack, practical RSA-based cryptosystems randomly pad the plaintext prior to encryption. This randomizes the ciphertext. This type of padding is defined in PKCS #1 v1.5 standard [5].
RSA-PKCS1-v1_5 can operate on messages of length up to *k* – 11 octets (*k* is the octet length of the RSA modulus), although care should be taken to avoid certain attacks on low-exponent RSA due to Coppersmith, Franklin, Patarin, and Reiter when long messages are encrypted. As a general rule, the use of this scheme for encrypting an arbitrary message, as opposed to a randomly generated key, is not recommended. It is possible to generate valid ciphertexts without knowing the corresponding plaintexts, with a reasonable probability of success [6].

RSA-SAEP+: Boneh has proposed this new padding scheme to be used with RSA. It is enhanced version of Simplified Asymmetric Encryption Padding. It is simpler than OAEP whereas OAEP is a two-round Feistel network, SAEP+ is a single- round [5]. It uses two random hash functions for encryption and decryption [7].

RSA-OAEP: In cryptography, **Optimal Asymmetric Encryption Padding** (**OAEP**) is a padding scheme often used together with RSA encryption. OAEP was introduced by Bellare and Rogaway.The OAEP algorithm is a form of Feistel network which uses a pair of random oracles G and H to process the plaintext prior to asymmetric encryption. When combined with any secure trapdoor one-way permutation *f*, this processing is proved in the random oracle model to result in a combined scheme which is semantically secure under chosen plaintext attack. When implemented with certain trapdoor permutations (e.g., RSA), OAEP is also proved secure against chosen plaintext attack. OAEP satisfies the following two goals:

1. Add an element of randomness which can be used to convert a deterministic encryption scheme (e.g., traditional RSA) into a probabilistic scheme.
2. Prevent partial decryption of ciphertexts (or other information leakage) by ensuring that an adversary

cannot recover any portion of the plaintext without being able to invert the trapdoor one-way permutation *f* [8].

OAEP gives an efficient RSA encryption scheme with a strong security guarantee (semantic security against chosen-ciphertext attacks). After Bleichenbacher's devastating attack on RSA-PKCS #1 v1.5 in 1998, RSA-OAEP became the natural successor (RSA-PKCS #1 v2.0) and thus a de facto international standard [5].

RSA-OAEP+: OAEP+ is essentially just as efficient as OAEP, and even has a tighter security reduction [15]. OAEP+ provides adequate security for key sizes used in practice. It uses the variable redundancy instead of the constant seed. It is thus a bit more intricate than the original OAEP. The security reduction for OAEP+ is efficient, but still runs in quadratic time [5].

REACT-RSA: This Rapid Enhanced Security Asymmetric Cryptosystem Transform is proposed by Okamoto. Another alternative to OAEP is the REACT construction. The RSA-REACT scheme is IND-CCA secure under the RSA assumption. Furthermore, the security reduction is very efficient [4]. Seven years after OAEP which makes chosen-ciphertext secure encryption scheme from any trapdoor one-way permutation, REACT-RSA is a new conversion which applies to any weakly secure cryptosystem, in the random oracle model: it is optimal from both the computational and the security points of view. Indeed, the overload is negligible, since it just consists of two more has hings for both encryption and decryption, and the reduction is very tight. Furthermore, advantages of REACT beyond OAEP are numerous:

1. It is more general since it applies to any partially trapdoor one-way function (a.k.a. Weakly secure public-key encryption scheme) and therefore provides security relative to RSA but also to the Diffie- Hellman problem or the factorization;
2. It is possible to integrate symmetric encryption (block and stream ciphers) to reach very high speed rates;
3. It provides a key distribution with session key encryption, whose overall scheme achieves chosen-ciphertext security even with weakly secure symmetric scheme. Therefore, REACT could become a new alternative to OAEP, and even reach security relative to factorization, while allowing symmetric integration [9].

HYBRID-REACT-RSA: It allows integration of a symmetric encryption scheme to achieve very high encryption rates [4]. In this REACT conversion; one can improve efficiency, replacing the one-time pad by any symmetric encryption scheme. One can use any symmetric encryption scheme that is just semantically secure (under no plaintext nor ciphertext attacks). Indeed, the one-time pad achieves perfect semantic security, against this kind of very weak attacks. But one can tolerate some imperfection. AES itself, resisted to more powerful attacks, and thus can be considered strongly secure in our scenario. Therefore, plaintexts of any size could be encrypted using this conversion, with a very high speed rate [10].

NTRU: NTRU is one of the public key cryptosystems. NTRU (Nth degree truncated polynomial ring units) is a collection of mathematical algorithms based on manipulating lists of very small integers. It was first introduced by Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman in 1998 [11]. NTRU is the first secure public key cryptosystem not based on factorization or discrete logarithmic problems. The keys are generated by having small potent polynomials from the ring of truncated polynomials given by $Z[X]/(XN - 1)$. The security of the NTRU cryptosystem is based on the difficulty of finding short vectors in a certain lattice. The larger the parameter *N*, the more secure the system is. NTRU is a probabilistic cryptosystem. The encryption process includes a random element and therefore one message has several possible encryptions. The advantage of NTRU over other cryptosystems is that it is highly random in nature, Encryption and decryption are very fast, the key sizes are relatively small and the key generation is fast and easy[12,13].

3. Result

After the implementation of above mentioned cryptosystem we prepared the comparison tables as follows:

Table 1: Comparison of various cryptosystems based on simulation time (in seconds) for different length of messages

Sr.No.	Cryptosystem	3 bytes	85 bytes	117 bytes	362 bytes	1432 bytes
1	RC4	0.000	0.000	0.000	0.015	0.062
2	DES	0.109	0.344	0.437	1.235	7.735
3	3DES	0.188	0.953	1.250	3.891	33.875
4	AES	1.172	1.297	1.359	1.797	3.672
5	BLOWFISH	2.032	2.094	2.140	2.375	3.422
6	**NTRU**	**0.344**	**0.437**	**0.500**	**0.906**	**2.687**
7	RSA	0.890	0.984	1.125	1.953	4.422
8	RSA PKCS1	10.234	10.250	10.265	NA	NA
9	RSA SAEP+	11.265	11.094	NA	NA	NA
10	RSA OAEP	11.234	11.219	NA	NA	NA
11	RSA OAEP+	11.250	11.343	11.781	14.797	33.453
12	REACT RSA	11.531	13.234	13.922	19.406	41.906
13	H REACT RSA	10.625	12.515	13.282	19.235	44.547

*NA: Not Applicable

4. Conclusion

From the above table it is clear that RC4 is the fastest algorithm. But this is a Symmetric stream cipher therefore at a time we can send only a single bit or byte and also its randomness is not very high. All the other Symmetric ciphers are block ciphers. If we compare the simulation speed of these Symmetric block ciphers and Asymmetric Ciphers then we conclude that for shorter messages DES is faster but for long messages the fastest cryptosystem is the

Asymmetric Cryptosystem NTRU. Since we know that Asymmetric cryptosystems are highly secure as compared to Symmetric Cryptosystems therefore we conclude that NTRU is a highly secure high speed cryptosystem.

Reference

[1] William Stalling, "Cryptography and Network Security", 3rd edition
[2] An Introduction to Cryptography, and Common Electronic Cryptosystems – Part I", EnterpriseITplanet.com
[3] From Wikipedia, the free encyclopedia
[4] Takako Nakashima & tatsuaki Okamoto, "Key size evaluation of provably secure RSA based encryption schemes", Natural science report Ochanomizu University, vol 57, Sep 2006, page 37-55.
[5] David Pointcheval, "How to Encrypt Properly with RSA ", RSA Laboratories' CryptoBytes. Volume 5, No. 1 - Winter/Spring 2002, pages 9-19.
[6] PKCS #1 v2.1:RSA Cryptography Standard, RSA Laboratories, June 14, 2002
[7] Natural Science Report, Ochanomizu University, Vol. 57, No.1 (2006)
[8] OAEP, From Wikipedia, the free encyclopedia
[9] Tatsuaki Okamoto and David Pointcheval "REACT: It is Rapid Enhanced Security Asymmetric Cryptosystem Transform ‘' ,1st JAN 2001
[10] David Pointcheval , " Asymmetric cryptography and practical security" , journal of telecommunication and information technology, April 2002
[11] J. Hoffstein, J. Pipher and J. H. Silverman, NTRU: A Ring-Based Public Key Cryptosystem. Algorithmic Number Theory (ANTS III), Portland, OR, June 1998, J.P. Buhler (ed.), LNCS 1423, Springer-Verlag, Berlin, 267-288, 1998
[12] Tommy Meskanen,"On the NTRU CryptoSystem", TUCS Dissertations No 63, June 2005
[13] From Wikipedia browsed on 15.7.14