Research Journal of Applied Sciences, Engineering and Technology 11(8): 847-853, 2015
DOI: 10.19026/rjaset.11.2094
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: May 16, 2015 Accepted: June 22, 2015 Published: November 15, 2015

Research Article

A Survey and Analysis of Security Issues on RSA Algorithm

Kunal Gagneja and K. John Singh
School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu, India

Abstract: RSA is used for remote login session, multimedia, credit card payment systems and email security. Key lengths are getting longer in RSA at an exponential phase without solving security issues. According to NIST key management guidelines, 15360 bit RSA is equivalent to 256-bit symmetric key in terms of strength. In case of small messages, even key would be longer than the message. Generally, security is calculated only on the basis of brute force attack and advancements in algorithms, hardware and software architectures are ignored. In this study, problems are identified based upon mathematical architecture of RSA and same observations are used in removal of defects.

Keywords: Acoustic signals, boneh-durfee attack, Chinese remainder theorem, coppersmith attack, quantum computer, wiener attack

INTRODUCTION

Public key cryptography is a concept in which there is one public key and one private key. Key used for encryption is called public key and is known to everyone and the key which is used for decryption is called private key and is known only to receiver. Security can be enhanced by combining public key with enveloped public key encryption but it complicates and slows the algorithm. Public key cryptosystem is a decades old concept and every public key cryptosystem has one or the other vulnerability. Public key encryption systems need a certification authority which can be a bank, charted accountant, lawyer or government. If certification authority gets compromised or is unfaithful, then entire system becomes insecure.

Architecture of Public key cryptosystems which involve factoring large numbers, discrete logarithms and elliptic curves makes them slower than symmetric key algorithms. There is a long list of insecure public key algorithms, for example Diffie Hellman (1976) which had been put forward in 1976. It is only limited to key exchange and does not have any method for encryption or decryption. ElGamal is another public key algorithm which is insecure in case key length is shorter than 2048 bits. Paillier is one of public key encryption algorithms which is insecure against adaptive chosen ciphertext attack. DSA has a disadvantage of doing only signatures and is patented.

MD2 is another public key cryptosystem which is vulnerable to pre-image attack having time complexity equivalent to 2^{104} applications of compression function. In 2008, preimage attack had been further refined with time complexity of 2^{73}. MD2 had been found vulnerable to collision attack in 2009 with time complexity of $2^{63.3}$ as compared to $2^{65.5}$ for birthday attack. Similarly MD4, MD5 and MD6 had one or the other vulnerabilities. Basis of RSA had been public key cryptography which is very old and dates back to 60's. A system equivalent to RSA has been developed i.e., ancestor of RSA is also too old. It is impossible for RSA to match with modern day improvements in computer architecture. It is just a matter of years when public key cryptography will be outdated because of lattice cryptography, quantum computing, DNA computing and photon computing. It is possible to break public key cryptography by quantum computer (Johnson *et al*., 2011).

LITERATURE REVIEW

Mathematical weaknesses in RSA: RSA had been broken for the first time just nineteen days after the announcement of RSA factoring challenge. Prizes from one hundred to fifty thousand dollars were offered to cryptographers who could break RSA, but it proved insecure. Brute force, mathematical attacks, timing attacks and chosen ciphertext attacks are possible approaches to attacking RSA. Security of RSA depends on correct implementation of algorithm, adequate use of algorithm, key generation and value of modulo (n), where n is modulus for both public and private keys.

It requires a key distributer or big brother to distribute keys. It can be a bank, charted accountant or government. Big brother can eavesdrop the entire communication or even secretly sell keys. They all

Corresponding Author: Kunal Gagneja, School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu, India

work after taking license from government. There were rumours about involvement of NSA in design of RSA right from its invention. Interestingly RSA became popular in spite of being slow and security issues. In 2013, RSA security LLC issued a warning against use of its own Dual_EC_DRBG random number generator which has been default in RSA's BSafe cryptographic toolkit because of backdoor by National Security Agency. Not a surprise that company implemented it even if it had been slow. RSA is 100 to 1000 times slower than DES and 10,000 times slower than AES. It puts limitations on implementation of RSA on devices with less hardware. NSA had been in controversy over inserting backdoors in public key algorithms known only to them. NSA had put restrictions on export of cryptographic products which exist even today on some algorithms. Interestingly, export restrictions were relaxed in 2000 when RSA had become too old. Thus it is easy for public key cryptosystems to get compromised either by NSA or by key issuing authority. RSA has the property that the product of two ciphertexts is equal to the encryption of the product of the respective plaintexts. Just apply permutation and combination with combinations of plaintexts and ciphertexts to create new attack.

In RSA algorithm public key is the key used for encryption and private key is used for decryption (Quisquater and Couvreur, 1982). Thus an identity of sender cannot be kept secret in network because an adversary can still learn who is transmitting even if he can't see which public key it is. However digital signature can be applied to RSA but it does not solve impersonation. Impersonation can be done in digital envelope which is secure electronic data container that protects data through encryption and authentication. There is no authentication even if AES and RSA are used together (Shuang-Hua, 2011). Secret key and public key are just two layers of digital envelope. An eavesdropper can simply buy prime numbers and multiply them and use permutation and combination to match with private key.

Sender should check the functions used in implementation of padding (RSAES-OAEP should be preferred). Otherwise public key will encrypt to the same output every time (James, 2011). RSAES-OAEP is an encryption scheme in which padding is used and it is securing against adaptive chosen cipher text attacks. Security of RSAES-OAEP depends on random nature of the output of the mask generation function. It further depends on random nature of the underlying hash function. A padding scheme like OAEP must be used to increase security. Unfortunately it is missing in many implementations of RSA and is prone to chosen cipher text attacks. OAEP adds an integrity check and hides structure of the message. RSAES-OAEP only becomes insecure if the octet-aligned OAEP process is translated to integers modulo n in RSA has a particular mathematical structure that can potentially be exploited without solving the RSA problem directly. A procedure based on Chinese remainder theorem can accelerate the decryption process because it makes use of mod function. Two large primes can be multiplied to match with public key using hit and trial method and one of them is private key. Distance between two prime numbers increases as the numbers become bigger. Thus it is easy to apply theory of probability as prime numbers grow bigger. RSA becomes vulnerable when size of one of the prime numbers used in calculation of public key is small. It is vital for RSA securities that two very large prime numbers be generated that are quite far apart. Generating composite numbers, or even prime numbers that are close to each other makes RSA totally insecure. Same value of mod (n) should not be used against large number of users (Bruce, 1995). If public key and algorithm are known then can RSA simply be reversed to get plaintext. Thus it is difficult and insecure to implement RSA in environment with constrained hardware because there is limit to size of prime numbers which can be applied.

Public keys should be authenticated: It is difficult to specify particular public key belongs to which user without authentication. Authentication procedure is another problem with RSA. It happens when signature S is not valid cryptosystem of sender's enciphering key (Alida, 1991). Increase in key length does not necessarily mean increase in protection against brute force attack. It is because of law of diminishing returns as shown in Fig. 1 in which X axis shows increase in key length and y axis shows corresponding increase in work factor in doing brute force attack.

RSA slowed down when it is implemented in hardware (Bruce, 1995). It puts restriction on key length and message length because of constraints of time. It can be concluded that RSA will become insecure against brute force attack because of Moore's law. RSA is slower than secret key method, but can be used in conjunction with the secret key to make it more

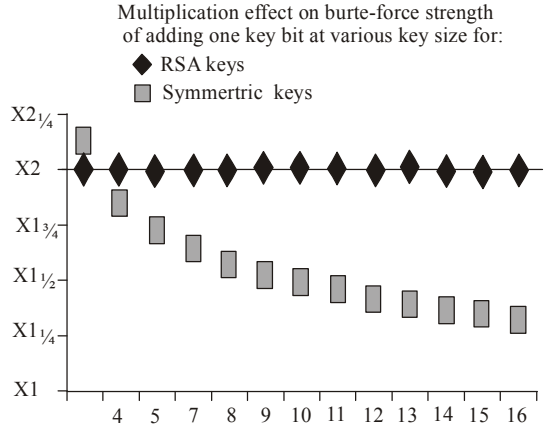

Fig. 1: Law of diminishing returns of RSA

efficient. Generally it is assumed that algorithm is known to everyone but key is kept secret, but public key algorithms have less speed. Public key and secret key algorithms can be combined to get advantages of both. However it is easy to do brute force attack on secret key algorithms. The security of RSA is dependent on the assumption that it is difficult to generate the private key from the public keys and the modulus n.

Mod function is not one-to-one (William, 2005; Sengadir, 2009). It is possible for multiple mods to give same result. For example 1998*1999*2000 mod 7 = 4 and 2^2006 mod 7 = 4. Thus the mod value of both gives same output.

It is not practical to use longer key lengths to encrypt short messages or files. One time pad is broken because the Soviets repeated the keys. There are two ways to solve this problem. First is to add padding for short messages which is dependent on generation of pseudo random numbers and are predictable. Second is to increase number of public keys with increase in number of users and number of messages. This can possibly lead to repetition of prime numbers which is itself insecure. Backbone of RSA security is length of prime numbers.

A power fault attack on RSA is successful in 2010. RSA digital signatures are based on CRT which can be subjected to power and fault attacks. Modular exponentiation when combined with CRT is prone to both the attacks (Sung-Kyoung *et al*., 2011). Chosen plaintext attack is possible in RSA (Tal, 1998). Chosen text attack can also be done (Desmedt and Odlyzko, 1986). In this attack, cryptanalyst has to obtain from plaintext versions of some carefully chosen cipher texts. It can be further processed to obtain further plaintext without additional resources. Till now, there is no such proof that security of RSA is only dependent on factoring public key. There are ways in which RSA can be crypt analysed without factoring public key. If small encryption exponent is used and same message is forwarded to multiple destinations, then RSA becomes insecure according to research done by Knuth. Suppose n is the public modulus of user A and public encryption exponent is E. Assuming that ciphertext has been intercepted and cryptanalyst wishes to recover plaintext m. A random integer x is chosen as:

$$C' \equiv x^e \, mod n$$

If it is sent to receiver to decrypt by using the pair (E, n) for signatures. The eavesdropper must also agree to sign challenge messages. Alternatively user receiver can discard decrypted messages which appear meaningless. Then we obtain Eq. (1) as follows:

$$(c')^d \equiv c^d x \equiv mx \, mod \, n \quad (1)$$

Thus n can be recovered. Any single ciphertext can be deciphered by using receiver decryption mechanism.

Coppersmith partial key exposure attack is possible on RSA. In it if few bits are discovered, then entire plaintext can be discovered (Mollin, 2002). Assume n = p*q is modulus of length l. If least significant bit or most significant bit of is known. Its value can be used to factor n. Partial key exposure attacks to have been successfully implemented on RSA (Mollin, 2002). RSA is insecure if Carmichael numbers are used as values of p and q. Probability of their occurrence is rare but increasing. Nowadays, messages and keys are getting longer. Factoring a number to check if it is Carmichael number or testing it for Carmichael numbers is another overhead.

Computing attack is possible in RSA as shown in Eq. (2):

$$n = p * q \quad (2)$$

Also Eq. (3) holds true:

$$\emptyset(n) = (p-1)(q-1) \quad (3)$$

Substituting q = n/p we get Eq. (4):

$$p^2 - (n - \emptyset(n) + 1)p + n = 0 \quad (4)$$

The two roots of equation are p and q, factors of n (Rosen, 1999). Because of better algorithms, it has become easier to factor number and test them for primarily. Smart card implementations of RSA are slow. RSA is slower than symmetric key algorithms (Bruce, 1995). An adaptive chosen-ciphertext attack on a smart card implementation of the RSA decryption algorithm in the presence of side-channel information leakage has been performed successfully. Information leakage through power consumption variation has been studied. Simple Power Analysis (SPA) of the smart card revealed macro characteristics caused by improper implementation of Chinese remaindering (Roman, 2002). Smart cards can be reverse engineered using chip testing equipment. A glitch attack against RSA has been performed in which implementation based on the Chinese Remainder Theorem (CRT) could recover the private key using only one message and corresponding faulty signature. Furthermore, the implementation often leaks additional side-channel information. Non-invasive attacks have been proposed based on timing information, a device's power consumption and electromagnetic radiation.

Acoustic attacks had been performed on RSA. It had been successful with 4096 bit RSA in 2013. Attack had been carried out with a microphone and a rudimentary hardware. This side channel attack has been carried out by Daniel *et al*. (2013) (who co-invented RSA) and Eran Tromer. Researchers listened

Fig. 2: Equipment used for capturing noise from decryption computer

to the high-pitched sounds from 10 to 150 KHz produced by computer as data had been decrypted. Low and band pass filters were used to separate sounds from background. Acoustic signals were generated by CPU's voltage regulator and were heard at a distance of thirteen feet. Even a smartphone could successfully get acoustic signals at a distance of thirty centimetres. The hardware used is shown in Fig. 2 (Daniel *et al*., 2013) which is equipment used for capturing noise from decryption computer. Figure 3 (Daniel *et al*., 2013) is an illustration of acoustic signals captured after the attack. Bruel and Kjaer 4939 microphone capsule had been used. Horizontal axis is frequency (0-310 KHz) and vertical axis is time (3.7 sec). Intensity is proportional to instantaneous energy in that frequency band.

Wiener attack and exponent attacks are also possible on RSA (Mollin, 2002). Decryption exponent d satisfies ed = 1mod$\emptyset(n)$.The value of $\emptyset(n)$ is equal to (p-1)(q-1) according to Euler phi function. Where p and q are two prime numbers chosen by Bob. For modular inverse to exist, encryption exponent e and $\emptyset(n)$ must be relatively prime to each other. Assuming that private key pair is (d, n). Applying fermat's little theorem, ciphertext can be obtained by applying Eq. (5):

$$C^d \equiv (M^e)^d \equiv M^{ed} \equiv Mmodn \quad (5)$$

Applying Euclidean algorithm, secret key can be recovered if factorization of n is known. According to Wiener's theorem, value of n = pq. If value of p lies between q and 2q:

$$q < p < 2q$$

And if value of $d < (n^{\frac{1}{4}})/3$, value of d can be recovered.

RSA is prone to decryption exponent attack and Weiner's low decryption exponent attack (Rosen, 1999). An exponent attack on RSA can be performed if two or three exponents are known. Assuming x, y, z are unknown parameters and two or three exponents (e1, e2, e3.....) are known with modulus (n). Decryption exponent attack is continuation of Guo's continued fraction attack and Blomer and May lattice-reduction basis attack. The following equation needs to be satisfied:

$$ex - \emptyset(n)y = z \quad (6)$$

It is easier to perform this attack if value of d is small. According to this attack RSA can be totally broken if the following equation is satisfied:

$$d < n^{0.292} \quad (7)$$

Boneh-Durfee attack had been successfully done on RSA. Partial Key exposure attacks are also possible in RSA. They are based on knowledge of some bits of secret decryption exponent d (Wenbo, 2003). Coppersmith attacks are possible in RSA (Menezes *et al*., 1996). Coppersmith theorem is used for factoring polynomials. It provides all roots of (f) modulo (n) that are less than $X = n^{\left(\frac{1}{d}\right)}$ for some $\epsilon \geq 0$. According to Coppersmith theorem n, is an integer and $f \in Z[x]$ is a monic polynomial of degree d. This theorem can find all integers satisfying $f(x) = 0 \, mod \, n$. Running time is less for smaller values of X. Coppersmith theorem can find small roots of polynomial modulo a composite n. If key modulus takes selected is long, then processor takes more time to generate keys (James, 2006). Timing attacks are possible on some implementations of RSA decryption. RSA is prone to common modulus attack,

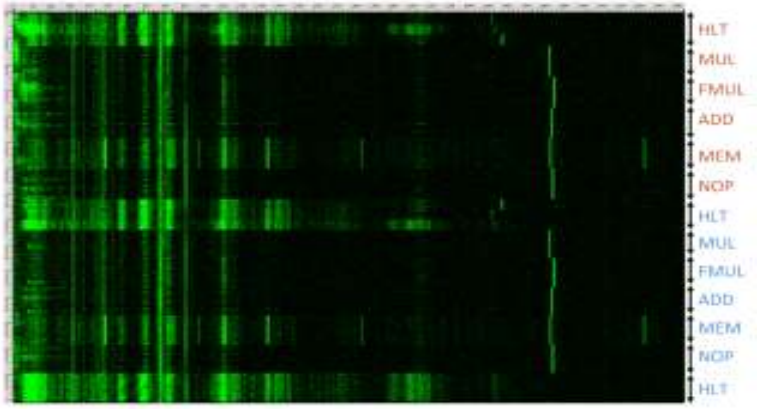

Fig. 3: An illustration of acoustic signals received

Table 1: Performance of RSA on different platforms

4kbit CPU seconds	1.8 GHz Celeron T3000(HP laptop	0.7 GHz ARMV6 (Raspberry Pi)
Key-gen	1.4-12.3	50.2-219
Sign	0.124-0.136	1.95-1.98
Verify	0.004-0.012	0.3
Encrypt	<0.001-0.004	0.317-0.320
Decrypt	0.132-0.136	1.93
Sign--encrypt	0.132-0.140	1.97-1.98
Verify-decrypt	0.132-0.140	1.96-1.96
Sign--encrypt (10kbit recipient)	0.13-0.14	2.04
Verify decrypt (10kbit receiver)	0.14-0.14	2.03
4kbit CPU seconds	(0.6 GHz ARMv7 OMAP 3430 ARM CortexA8(Nokia N900)	0.26 GHz MIPS Broadcom BCM 4704(Asus WL500gP)
Key-gen	29-221	139-334
Sign	1.57-1.64	3.50-3.58
Verify	0.016-0.039	0.07-0.08
Encrypt	0.016-0.023	0.06-0.09
Decrypt	1.55-1.60	3.47-3.52
Sign--encrypt	1.59-1.62	3.55-3.59
Verify-decrypt	1.58-1.60	3.50-3.54
Sign--encrypt (10kbit recipient)	1.66-1.71	3.79-3.86
Verify decrypt (10kbit receiver)	1.67-1.69	3.74-3.78

Table 2: Performance of RSA on different hardware

10 bit key CPU seconds	1.8GHz Celeron T300 (HP laptop)	0.7 GHz ARM (Raspberry Pi)
Key-gen	47-430	>2000
Sign	1.692-1.704	18.9
Verify	0.004-0.012	0.368
Encrypt	0.004-0.012	0.374
Decrypt	1.680-1.688	18.8
Sign--encrypt	1.696-1.712	18.9
Verify decrypt	1.68-1.70	20.3
Sign--encrypt (4kbit recipient)	1.69-1.70	18.9
Verify decrypt (4kbit sender)	1.68-1.70	18.8
10 bit key CPU seconds	0.6GHz ARMv7 OMAP 3430 ARM Cortex A8 (Nokia N900)	0.26GHz MIPS Broadcom BCM4704 (Asus WL500 gP)
Key-gen	448-6375	2135-17609
Sign	18.8-19.0	44.61-44.77
Verify	0.078-0.102	0.21-0.25
Encrypt	0.086-0.109	0.23-0.26
Decrypt	18.8-18.9	43.60-43.72
Sign--encrypt	18.9-19.3	43.89-44.07
Verify decrypt	18.83-19.03	44.41-44.70
Sign--encrypt (4kbit recipient)	18.81-19.25	44.38-44.52
Verify decrypt (4kbit sender)	18.84-18.95	44.26-44.42

chosen ciphertext attack, low encryption exponent attack and low decryption exponent attack (Bruce, 1995). Bit-security is low for RSA (Van Tilborg and Jajodia, 2011).

Slow speed and CPU performance of RSA: Experimentally, key works fine on a 3 GHz desktop, it may be slow on smartphone, or an access point. The following table had been drawn after running every test ten times. Minimum and maximum runtime had been observed. Table 1 shows how performance of RSA degrades in terms of execution time in seconds with change in hardware. RSA is totally inconvenient to use in low hardware devices. The following tables shows performance of RSA on different platforms. Table 2 is another comparison of RSA in terms of execution time in seconds when RSA had been executed on different hardware. In addition to above, key sizes are growing at exponential phase making it difficult to implement RSA in future, which is illustrated in Fig. 4. Key length is shown on y axis and year is shown in x axis. From these tables, it can be concluded that RSA is slow, putting limitations on it's implementation.

Comparative analysis of severity of these attacks: Since, there is only one big brother for key distribution, there were many instances in past when keys were sold or leaked i.e., threat from it is real. Prime numbers are getting larger and number of known prime numbers is also increasing but the rate discovery of new prime numbers has remained almost constant. In future, it will be more difficult to match product of two prime numbers with public key. Power fault attacks were implemented on other encryption algorithms too. It is highly severe and can only be solved only by using DC current provided by battery instead of AC current. Chosen text attacks and chosen plaintext attacks are not a serious concern as they can be easily solved by padding. Adaptive chosen-cipher text attack on smart card can be a serious threat as it was done on less

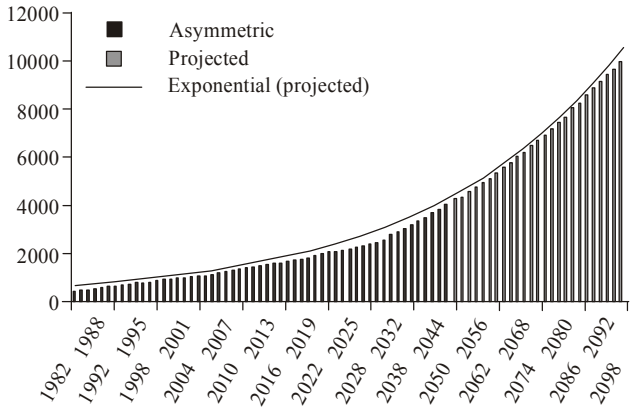

Fig. 4: Key lengths of RSA

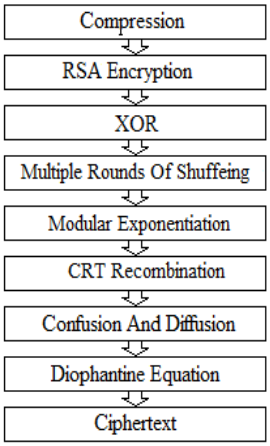

Fig. 5: Stages in secure implementation of RSA

hardware and needs DC power and padding together. It is difficult to have DC power every time smart card is used. Acoustic attacks were done in complete silence and it would be difficult to achieve the same in real life situations. Exponent attack can be avoided by longer non repeating keys which is easy to implement. Coppersmith attack can only work for small value of n.

Enhanced RSA: Generally encryption and decryption are done in base 2. Data had been converted into a random base known only to sender and reviver. RSA is a block cipher in which padding is applied. Random bits were added in every block to make padding more effective. XOR and shuffling were added because many of the standardised algorithms are also using these mathematical functions. Multiple compression had been be done before encryption. It makes cryptanalysis difficult and gives better compression ratio as compared to ordinary compression algorithms. Confusion and diffusion must be added in the end to avoid statistical attacks. Power and fault attacks can be solved by modular exponentiation and CRT recombination algorithms. Arithmetic operations were replaced by logical ones to stop errors in CRT recombination step. Because CRT-RSA algorithm does not need knowledge of public exponent, it is implemented in better way. After adding confusion and diffusion, Diophantine equation had been added for more security. In the proposed new RSA algorithm, it had been checked at the time of encryption if it could be broken using Diophantine equation. It is a polynomial equation with two or more unknowns such that only integer solutions are found (it is applicable to RSA as we will only use integers as keys). Diophantine equation can be used for cryptanalysis of RSA. Chinese remainder theorem provides important example of linear systems of Diophantine equations. Let y1, y2 ..., yk be k pairwise coprime integers greater than one, a1, ..., ak be k arbitrary integers and X be the product y1 ⋯ y2. The Chinese remainder theorem asserts that the following linear Diophantine system has exactly one solution (y, y1, ..., yk) such that $0 \leq x < Y$ and that the other solutions are obtained by adding to x a multiple of Y:

$$x = a1+x1*y1$$
$$\vdots$$
$$x = ak+xk*yk$$

Congruence has been used for error detection and correction in next phase. A small code has been added in next step to modify encrypted file to avoid rubber-hose cryptanalysis or purchase key attack. Cryptanalysis proved it to be more secure than earlier implementation. Figure 5 shows stages of implementation of RSA after removing security drawbacks.

To improve security, use of same key had been avoided for encryption and signing. A blind message had been prevented from getting encrypted or signed. Input was formatted before getting encrypted or signed. It had been ensured that length of padding had been at least 8 bits and same padding had not been repeated. If any error had been discovered in digital signature or

decryption by receiver, then only error message had been returned instead of decrypted or signed message.

CONCLUSION

Dual_EC_DRBG random number generator had been removed to make it more difficult for NSA to spy. Quantum computing is still in experimental stage, so RSA can be assumed to be secure against it for at least few years but not decades. Moore's law will be replaced be mew Moore's law somewhere between 2020 and 2030. Calculations based on RSA on quantum computer will only be possible after its implementation. RSA has got security advantages of other algorithms like aes, des etc because of introduction of xor and shuffling. It is expected that file size would increase in future which would make adding confusion and diffusion easier. Since RSA is slow, doing brute force attack on it will take more time.

REFERENCES

Alida, S., 1991. New Trends in Cryptology. 2nd Edn., Lulu Publishers, New York.

Bruce, S., 1995. Applied Cryptography: Protocols, Algorithms and Source Code in C (cloth). 2nd Edn., John Wiley and Sons, New York.

Daniel, G., S. Adi and T. Eran, 2013. RSA key extraction via low-bandwidth acoustic cryptanalysis. IACR cryptology ePrint Archive, December 18.

Desmedt, Y. and A.M. Odlyzko, 1986. A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes. In: Williams, H.C. (Ed.), Advances in Cryptology-CRYPTO' 1986. LNCS 218, Springer-Verlag, Berlin, Heidelberg, pp: 516-522.

James, H.C., 2006. IPSec Virtual Private Network Fundamentals. 1st Edn., Cisco Press, Indianapolis.

James, M., 2011. A chosen ciphertext attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as standardized in PKCS #1 v2.0. Proceeding of the 21st Annual International Cryptology Conference on Advances in Cryptology-CRYPTO, August 2, pp: 230-238.

Johnson, M.W., M.H.S. Amin, S. Gildert, T. Lanting, F. Hamze *et al*., 2011. Quantum annealing with manufactured spins. Nature, 473: 194-198.

Menezes, A.J., P.C. Oorschot and S.A. Manstone, 1996. Handbook of Applied Cryptography. 1st Edn., CRC Press, Boca Raton, London.

Mollin, R.A., 2002. RSA and Public Key Cryptography. 2nd Edn., Chapman and Hall/CRC, Boca Raton, FL.

Quisquater, J.J. and C. Couvreur, 1982. Fast decipherment algorithm for RSA public-key cryptosystem. Electron. Lett., 18(21): 905-907.

Roman, N., 2002. Public Key Cryptography. 1st Edn, Springer, Berlin.

Rosen, K.H., 1999. Discrete Maths And its Applications. 4th Edn., McGraw-Hill Inc., New York.

Sengadir, T., 2009. Discrete Mathematics and Combinatorics. 5th Edn., Pearson Education, Chennai, India, pp: 568, ISBN: 8131714055.

Shuang-Hua, Y., 2011. Internet-Based Control Systems: Design and Applications. 1st Edn., Springer-Verlag, London, pp: 204.

Sung-Kyoung, K., H.K. Tae, H. Dong-Guk and H. Seokhie, 2011. An efficient CRT-RSA algorithm secure against power and fault attacks. J. Syst. Software, 84(10): 1660-1669.

Tal, R., 1998. A simplified approach to threshold and proactive RSA. Proceeding of the 18th Annual International Cryptology Conference, NCS 1462, August 23-27, pp: 89-104.

Van Tilborg, H.C.A. and S. Jajodia, 2011. Encyclopedia of Cryptography and Security. 2nd Edn., Springer, US, ISBN: 978-1-4419-5905-8.

Wenbo, M., 2003. Modern Cryptography: Theory and Practice. 1st Edn., Prentice Hall, Upper Saddle River, NJ, ISBN: 013288741X, pp: 707.

William, S., 2005. Cryptography and Network Security Principles and Practices. 4th Edn., Pearson Education, New York.