ARM Based Implementation Of RSA Algorithm

Ms. Hemlata Powar [1], Prof. S. S. Patil[2]

[1,2] *Department of Electronics, Tatyasaheb Kore Institute of Engineering & Technology, Warananagar, India*

Abstract

The information that are being transferred through the network of computers being read by other people is very high. We need an efficient way of securing data. Therefore in this paper Implementation of RSA algorithm is presented, to provide maximum security for data over the network. RSA is one of the most-common used algorithms for public-key cryptography applied for encrypting information. Having two keys allows initiating secure communication through public unprotected channels without the need to pass the private key to other side. It is involved encryption, decryption, and key generation. This technique provides more efficiency and reliability over the networks.

Keywords: ***public-key cryptography encryption, decryption, RSA***

1. Introduction

In the present scenarios, data security in international network like internet plays a vital role as the use of internet goes on increasing day-by-day, the number of users also simultaneously increasing. Therefore, it makes sense for all parties involved to secure the Internet from fraudulent use. Security problems can occur in any networked environment. Secured transmission of data can be achieved by using encryption algorithm i.e. RSA algorithm.RSA is an algorithm for public-key cryptography suitable for both signature/verification and encryption/decryption [1]. The encryption and decryption solution can ensure the confidentiality of the information, as well as the integrity of information and certainty, to prevent information from tampering. RSA is used in hundreds of software products and can be used for key exchange, digital signatures, or encryption of small blocks of data**.**RSA uses a variable size encryption block and a variable size key [2]. The challenge of RSA is to develop an algorithm in which it is impossible to determine the private key. RSA gets its security from the difficulty of factoring very large numbers [8]. We used 'n' prime numbers to provide more security throw the networks and it is not easily factorized. Currently, it is suggested that the bit length of N should be at least 1024 for RSA to be considered secure. . RSA is also used in some security protocols to ensure security.

2. Literature review

The idea of RSA algorithm is related with the fact that Diffie & Hellman introduced method of exponential key exchange**.** The Clifford Cocks,a mathematician, was made first ever known description of key exchange system in 1973.but this system was not used because for implementation it requires expensive computers. In 1976 Ron Rivest,Adi Shamir & Leonard Adleman, are working at MIT on novel of cryptographic design. In 1977 they got idea of RSA algorithm for security purpose.

3. RSA: Public Key Cryptography

RSA was publicly described in 1977 by Rivest, Shamir, and Adleman at MIT [11]. A user of RSA creates and then publishes the product of two large prime numbers. In this algorithm 'n' prime number used which is not easily breakable. In RSA cryptographic algorithm two keys are used namely private key and public key. The public key which is not secret key, used for encrypt the messages. The private key is used for decrypt the message .Private key is kept secret. In RSA algorithm we can encrypt a message without the need to exchange a secret key separately. Core arithmetic operation in RSA is exponentiation, which is accomplished by a series of modular multiplications. Therefore, fast modular multiplication is key to achieving fast execution of RSA cryptosystems [12]. According to number theory, it is easy to finds two big prime number, but the factorization of the product of two big prime numbers is very difficult.

4. Implementation Tools of RSA Algorithm

In order to implement RSA we need [4]:

- **Arbitrary precision arithmetic**
  Used to handle large numbers & it provides optimized implementations of arithmetic operations such as modular computation and exponential computation.

- **Pseudo Random Number Generator (PRNG)**
  PRNG is used to provide random numbers for RSA key generation PRNG takes secret input samples (noise, seed) into the PRNG and produces random output. This random output of PRNG is secured with cryptographic function.

- **Prime number generator**
  Prime number is a positive integer and is divisible only by itself and 1. Prime numbers are found with primality testing**.**

5. RSA Algorithm

RSA algorithm involve following three steps i.e. Key generation, encryption, decryption.

5.1 Key Generation [5]:-

- Generate two large random primes, p and q. For security aim, should be of similar bit-length.
- Compute n=p*q
  Where n is called as modulus & its length is required bit length i.e. 1024 bits.
- Calculate $\varphi(n) = (p-1)(q-1)$, where φ is Euler's totient Function
- Choose an integer e, $1 < e < \varphi(n)$, such that $\gcd(e, \varphi(n)) = 1$.
- Compute the *d*, $1 < d < \varphi(n)$, such that $ed \equiv 1 \pmod{\varphi(n)}$, where d is secret component.
- Public Key is (n,e) which is used for encryption
- Private Key is (n,d) used for decryption.

5.2 Encryption:-

We encrypt the message with their public key (e,n). Encryption is done by taking an exponentiation of the message m with the public key e and then taking a modulus of it [6]. The following steps are done in encryption.

1. Obtain public key (n,e)
2. Generate plaintext message m, where m<n
3. Compute the cipher text $c = m^{\wedge e} \bmod n$.
4. Send the cipher text '**c**' to the recipient.

5.3 Decryption:-

Decryption is done using the Private key. The person who is receiving the encrypted message uses his own private key to decrypt the message [6]. Decryption is similar to the encryption except that the keys used are different.

1. Recipient uses his private key (n,d) to compute $m = c^{\wedge d} \bmod n$.
2. Obtain the plaintext '**m** '.

6. Key length for RSA

The key length for a secure RSA transmission is typically 1024 bits. For more security use 2048 or even 4096 bits. The longer your information is needed to be kept secure, the longer the key you should use. Keep up to date with the latest recommendations in the security journals. To encrypt a 256-bit key with a 1024-bit RSA modulus means we only need a single representative message integer.

7. RSA Algorithm Example:-

Key Generation

- Choose p = 3 and q = 11
- Compute $n = p * q = 3 * 11 = 33$
- Compute $\varphi(n) = (p - 1) * (q - 1) = 2 * 10 = 20$
- Choose e such that $1 < e < \varphi(n)$. Let e = 7

- Compute value for d such that (d*e)% φ(n)=1 Select d=3. i.e., d is the multiplicative inverse of e (modulo φ(n)).
- Public key is (e, n) = (7, 33)
- Private key is (d, n) =(3, 33)

Encryption

- Select message m=2
- Compute ciphertext c = 2^7 % 33 = 29

Decryption

- Compute m = 29^3 % 33 = 2
- Obtain original message m=2

8. Block Diagram

Here we implement RSA algorithm using ARM processor i.e. LPC 2138. For programming we are using Keil software. Figure 1 represents encoding of message & figure 2 represents decoding procedure for generation of original message

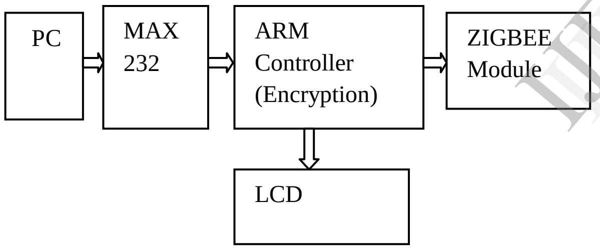

Figure 1. Transmitter Section

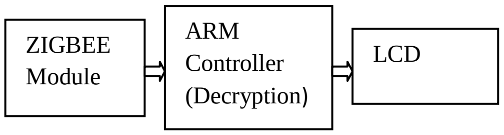

Figure 2. Receiver Section

Steps for implementation:

1. First develop pc interfacing code with ARM Controller.
2. Generate public key for encryption & private key for decryption.
3. Using public key develop encryption code in Embedded c.
4. ARM controller interfaced with zigbee.
5. Send this data to its destination through network.
6. At receiver side, using private key develop decryption code.
7. ARM CONTROLLER interfacing with LCD in embedded c & Display original message on LCD.

9. Theoretical Result

Following tables illustrates the results of encryption & decryption of message

Message	Encrypted data	Display
0	0	0
1	1	1
2	29	29
3	9	9
4	33	33
5	16	16
6	30	30
7	28	28
8	2	2
9	15	15

Table 1. Encryption at Transmitter

Received Data	Decrypted data	Display
0	0	0
1	1	1
29	2	2
9	3	3
33	4	4
16	5	5
30	6	6
28	7	7
2	8	8
15	9	9

Table 2. Decryption at Receiver

10. Conclusion

In this paper, we implemented an algorithm of encoding and decoding of information based on the RSA algorithm and intended for secure data transmission. This interface was implanted under keil software. In which we endeavored to get the quality that make easier the cryptography to have a good use of prime numbers. Primary advantage of RSA- public key cryptography algorithm is increased security and convinces. Therefore it is used in web browsers, email programs, mobile phones, virtual private networks and secure shells etc. RSA algorithms provide the security to the network and data.

11. References

[1] W. Diffie and M. E. Hellman, “New directions in cryptography”, IEEE Trans. Inform.Theory, Nov1976, 22:644-654

[2]HTTP://WWW.GARYKESSLER.NET/LIBRARY/CRYPTO.HTML

[3] RSA LABORATORIES: HTTP://WWW.RSA.COM/RSALABS/NODE.ASP?ID=2167

[4] Anderson, J. A. and Bell, J. M. 1997. *Number Theory with Applications*. New Jersey: Prentice-Hall

[5] Chang, C. C. and Hwang, S. J. 1997. A Simple Approach for Generating RSA Keys. *Information Processing Letters*. 63 (1), 19-21

[6] Xin Zhou and Xiaofei Tang, "Research and Implementation of RSA Algorithm for Encryption and Decryption", the 6th International Forum on Strategic Technology, pp. 1118 – 1121, 2011

[7] Uma Somani, Kanika Lakhani and Manish Mundra, “Implementing Digital Signatures with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing”, 1st International Conference on Parallel, Distributed and Grid Computing (PDGC), pp. 211-216, 2010

[8] Steve Burnett and Stephen Paine, “The RSA Security's Official Guide to Cryptography”, CA USA: Osborne/McGraw-Hill, 2001

[9] Public-key cryptographic standards. http://www.rsasecurity.com, 2004.
[10]http://mathworld.wolfram.com/RSAEncryption.html

[11] R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signature and public-key cryptosystems,” *Commun. ACM*, vol. 21, pp. 120–126, Feb. 1978
[12] C. McIvor,M.McLoone, and J. V. McCanny, “Fast Montgomery modular multiplication and RSA cryptographic processor architectures,”