An Efficient Implementation of RSA Data Encryption Algorithm In 8051

M.Suresh[1], P.Saravana Kumar[2], S.M.Ramesh[3], Dr.T.V.P.Sundararajan[4], S.Muthusamy[5]

Abstract— **Main objective is building a RSA data encryption using the 8051 microcontroller. Encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encryption doesn't prevent hacking but it reduces the likelihood that the hacker will be able to read the data that is encrypted. In an encryption scheme, the message or information, referred to as plaintext, is encrypted using an encryption algorithm, turning it into an unreadable cipher text. This is usually done with the use of an encryption key, which specifies how the message is to be encoded. Any adversary that can see the cipher text should not be able to determine anything about the original message. An authorized party, however, is able to decode the cipher text using a decryption algorithm that usually requires a secret decryption key that adversaries do not have access to. In this age of information technology and with the fast advancement of computer technology, the need to secure information especially digital information has become immense [8]. Nowadays, most encryption and decryption processes are done by the computer. The objective of this project is to write an encryption and decryption program for the microcontroller and AT89C52 microcontroller as an external encryption [1] and decryption platform and the computer will be used as the interface platform between the user and the microcontroller. Programming language that is used to write the program is C language.**

***Index Terms*—Cryptography; Trojan horse; key distribution, RSA.**

I. INTRODUCTION

Encryption, by itself, can protect the confidentiality of messages, but other techniques are still needed to protect the integrity and authenticity of a message; for example, verification of a Message Authentication Codes (MAC) or a digital signature. Standards for cryptographic software and hardware to perform encryption are widely available, but successfully using encryption to ensure security may be a challenging problem. A single slip-up in system design or execution can allow successful attacks. Sometimes an adversary can obtain unencrypted information without directly undoing the encryption. e.g. Traffic analysis, TEMPEST, or Trojan horse. Digital signature and encryption must be applied at message creation time (i.e. on the same device it has been composed) to avoid tampering. Otherwise Any node between the sender and the encryption agent could potentially tamper it.

Encryption is also used to protect data in transit, for example data being transferred via networks (e.g. the Internet, e-commerce), mobile telephones, wireless microphones, wireless intercom systems, Bluetooth devices and bank automatic teller machines. There have been numerous reports of data in transit being intercepted in recent years. Encrypting data in transit also helps to secure it as it is often difficult to physically secure all access to networks.

II. RSA ALGORITHM SYSTEM

The RSA algorithm involves three steps: key generation, encryption and decryption.

A. KEY GENERATION.

RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated the following way:

1) Choose two distinct prime numbers p and q.
For security purposes, the integer's p and q should be chosen at random, and should be of similar bit-length. *Prime integers* can be efficiently found using a primarily test.

2) Compute n = pq.
n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.

3) Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1)$, where φ is Euler's totient function.

4) Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are coprime.
- ✓ e is released as the public key exponent.
- ✓ e having a short bit-length and small Hamming weight results in more efficient encryption – most commonly 216 + 1 = 65,537. However, much smaller values of e

Manuscript received Aug 15, 2012.

***M.Suresh,**, PG Scholar, Department of ECE, BannariAmman Institute of Technology, Erode, T.N, India.*
***P.Saravana Kumar**, Assistant Professor, Department of ECE, BannariAmman Institute of Technology, Erode, T.N, India.*
***S.M.Ramesh,** Associate Professor, Department of ECE, BannariAmman Institute of Technology, Erode, T.N, India.*
***Dr.T.V.P.Sundararajan,** Professor, Department of ECE, BannariAmman Institute of Technology, Erode, T.N, India.*
***S.Muthusamy,** PG Scholar, Department of IT, BannariAmman Institute of Technology, Erode, T.N, India.*

(such as 3) have been shown to be less secure in some settings.

5) Determine d as d ≡ e−1 (mod φ(n)); i.e., d is the multiplicative inverse of e (modulo φ(n)).
   - ✓ This is more clearly stated as: solve for d given d·e ≡ 1 (mod φ(n)).
   - ✓ This is often computed using the extended Euclidean algorithm. Using the pseudo code in the Modular integers section, inputs a and n correspond to e and φ(n), respectively.
   - ✓ d is kept as the private key exponent.

B. ENCRYPTION.

Alice transmits her public key (n, e) to Bob and keeps the private key secret. Bob then wishes to send message M to Alice. He first turns M into an integer m, such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$$c \equiv m^e \pmod{n} \quad (1)$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice. Note that at least nine values of m will yield a cipher text c equal to m, but this is very unlikely to occur in practice.

C. DECRYPTION.

Alice can recover *m* from *c* by using her private key exponent *d* via computing

$$m \equiv c^d \pmod{n} \quad (2)$$

Given *m*, she can recover the original message *M* by reversing the padding scheme.

Suppose Alice uses Bob's public key to send him an encrypted message. In the message, she can claim to be Alice but Bob has no way of verifying that the message was actually from Alice since anyone can use Bob's public key to send him encrypted messages. In order to verify the origin of a message, RSA can also be used to sign a message.

III. HARDWARE IMPLEMENTATION

A. CAESAR CIPHER

In the Earlier system Caesar cipher was used. In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A; E would become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence.

B. BREAKING THE CAESAR CIPHER

The Caesar cipher can be easily broken even in a cipher text-only scenario. Two situations can be considered:

1. An attacker knows (or guesses) that some sort of simple substitution cipher has been used, but not specifically that it is a Caesar scheme.
2. An attacker knows that a Caesar cipher is in use, but does not know the shift value.

C. MICROCONTROLLER

AT89C51 features are:

1. 4 Kbytes of In-System Reprogrammable Flash Memory. Endurance 1,000 Write/Erase Cycles
2. Fully Static Operation: 0 Hz to 24 MHz
3. Three-Level Program Memory Lock
4. 128 x 8-Bit Internal RAM
5. 32 Programmable I/O Lines
6. Two 16-Bit Timer/Counters
7. Six Interrupt Sources
8. Programmable Serial Channel

Description:

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4 Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters.

IV. SYSTEM IMPLEMENTATION

A. External vs. internal encryption

There are many applications and software out there today that can perform this encryption process. However, most or

ISSN: 2278 – 909X

even all of these programs and applications are made to run on the computer. This means that the encryption process is done internally within the computer. There is a huge drawback to having the encryption process to be done internally, which is the algorithms and keys for encrypting and decrypting information reside somewhere in the computer memory or hard disk and the secret keys are exposed in the medium or the program. This makes these encryption programs and applications that run on the computer and do the encryption and decryption process internally vulnerable to attacks from hackers. The same cannot be said if the encryption and decryption process is done externally. Externally here means that the encryption and decryption process is done on a different device or hardware independent to the computer. By doing so, the algorithms, encryption and decryption keys are always locked in a highly secured hardware environment.

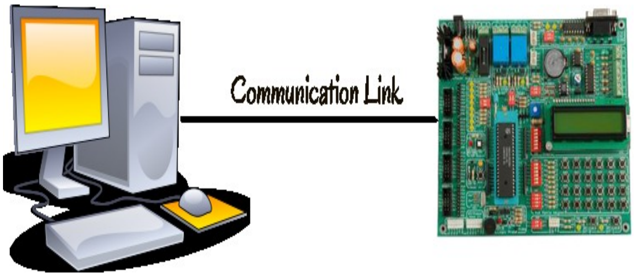


Fig.4.1 Hardware System Model

Advantages of external encryption and decryption are:

a. Algorithms, encryption and decryption keys are always locked in a highly secured hardware environment.
b. Safe from hackers.
c. In the worst circumstances, to maintain secrecy, the hardware can be destroyed. Then, the algorithms, encryption and decryption keys will remain a secret forever.

Disadvantages of external encryption and decryption are:

a. Requires more work.
b. Could be more expensive since there are many free encryption and decryption programs for the computer out there today.

Advantages of internal encryption and decryption are:

a. Easier to develop.
b. Cheap and in fact, there are many free encryption software available out there

Disadvantages of internal encryption and decryption are:

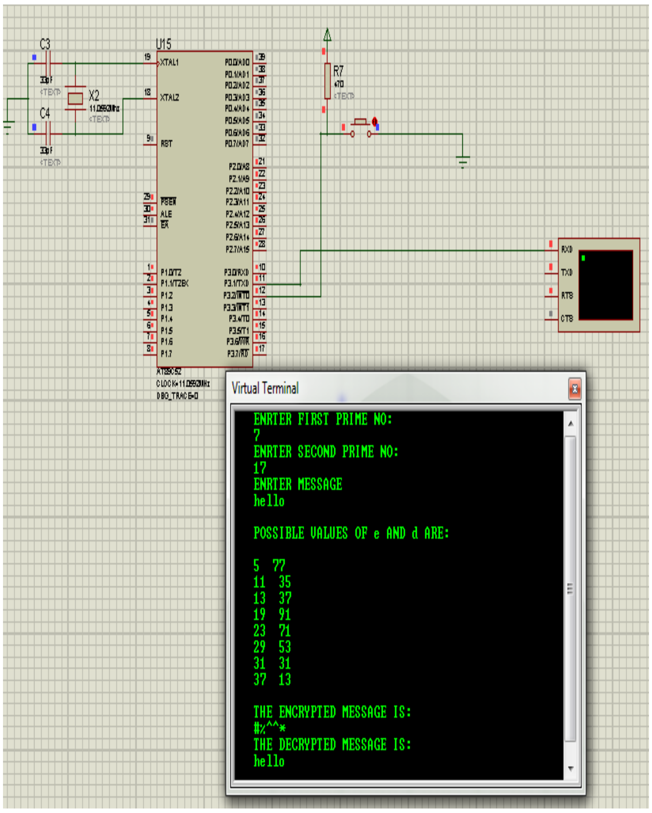
a. Vulnerable to hackers
b. Not as secure as external encryption.

External encryption is carried out. A computer system is connected with AT89C51 development board, whenever data transmission is carried out first send to the board using communication link. The text is converted using RSA encryption and then sends back to the PC. At the receiver side same process carried out RSA decryption to retrieve the original text.

B. *Experimental Results*

Present paper is designed using 8051 microcontroller. It is proposed to design an embedded system which is used for information security. In this paper AT89C52 microcontroller is used for interfacing to various hardware peripherals. A serial driver IC is used for converting TTL voltage levels to RS-232 voltage levels.

The simulation is carried out by using PROTEUS design tool. The current design is an embedded system platform, which asks for a prime number input to compute the output.



```
ENRTER FIRST PRIME NO:
7
ENRTER SECOND PRIME NO:
17
ENRTER MESSAGE
hello

POSSIBLE VALUES OF e AND d ARE:

5  77
11  35
13  37
19  91
23  71
29  53
31  31
37  13

THE ENCRYPTED MESSAGE IS:
#%^^*
THE DECRYPTED MESSAGE IS:
hello
```

Fig.4.2 Implementation Model

V. CONCLUSION

RSA data encryption using the 8051 microcontroller tends to provide more security than others. Externally, that the encryption and decryption process is done on a different device or hardware independent to the computer. By doing so, the algorithms, encryption and decryption keys are always locked in a highly secured hardware environment than the Caesar cipher text algorithm. Experimental results shows that it will work for any system and can put forward to practical applications.

REFERENCES

[1] Prasant Singh Yadav, Pankaj Sharma, Dr K. P Yadav Implementation of RSA algorithm Using elliptic curve algorithm for Security and performance Enhancement, International Journal of Scientific & Technology Research Volume 1, Issue 4, May 2012.

[2] Sambhu Prasad Panda, Madhusmita Sahu, Umesh Prasad Rout, Surendra Kumar Nanda Encryption and Decryption algorithm using two dimensional cellular automata rules in Cryptography. International Journal of Communication Network & Security, Volume-1, Issue-1, 2011.

[3] Leif Uhsadel, Markus Ullrich, Ingrid Verbauwhede and Bart Prenee HW/SW co-design of RSA on 8051.

[4] Leif Uhsadel, Markus Ullrich,Ingrid Verbauwhede and Bart Prenee Cryptographic authentication on the communication from an 8051 based development board over UDP.

[5] Sushanta Kumar,Sahu Manoranjan Pradhan FPGA Implementation of RSA Encryption System International Journal of Computer Applications (0975 – 8887) Volume 19– No.9, April 2011.

[6] Thomas Wollinger, Jorge Guajardo, and Christof Paar Cryptography in Embedded Systems: An Overview. Proceedings of the Embedded World 2003 Exhibition and Conference, pp. 735-744,

[7] Sushanta Kumar, Sahu Manoranjan Pradhan FPGA Implementation of RSA Encryption System, in International Journal of Computer Applications (0975 – 8887) Volume 19– No.9, April 2011

[8] M. Preetha, M. Nithya, A study and performance Analysis of RSA algorithm, IJCSMC, Vol. 2, Issue. 6, June 2013, pg.126 – 139.