© 2015 IJSRSET | Volume 1 | Issue 3 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Implementation and Performance Analysis of RSA Algorithm Using Verilog

A. Manish Kumar*1, Vishal Moyal2
*1Electronics & Telecommunication, SSTC-SSGI [FET], Bhilai, Chhattisgarh, India
2Electronics & Telecommunication, SSTC-SSITM, Bhilai, Chhattisgarh, India

ABSTRACT

This paper proposes an efficient method to implement RSA decryption algorithm. RSA cryptosystem is the most attractive and popular security technique for many applications, such as electronic commerce and secure internet access. There are few end-users today who make use of real security applications. These applications tend to be too complicated, exposing too much detail of the cryptographic process. Users need simple inherent security that doesn't require more of them simply clicking the secure checkbox. Cryptography is a first abstraction to separate specific algorithms from generic cryptographic processes in order to eliminate compatibility and upgradeability problems. The core idea is enhance the performance of RSA algorithm. In this paper public key algorithm RSA and enhanced RSA are compared analysis is made on the three designing parameters i.e. time delay and power consumption. In a word, the total data path time delay of our proposed method is almost times 15.35% less than the already existing method [1]. The proposed method enhances the performance of the RSA operation.

Keywords: Cryptography, Data path time delay, RSA cryptosystem

I. INTRODUCTION

Data communication is an important aspect of our living. So, protection of data from misuse is essential. A cryptosystem defines a pair of data transformations called encryption and decryption. Encryption is applied to the plain text i.e. the data to be communicated to produce cipher text i.e. encrypted data using encryption key. Decryption uses the decryption key to convert cipher text to plain text i.e. the original data. Now, if the encryption key and the decryption key is the same or one can be derived from the other then it is said to be symmetric cryptography. This type of cryptosystem can be easily broken if the key used to encrypt or ecrypt can be found. To improve the protection mechanism Public Key Cryptosystem was introduced in 1976 by Whitfield Diffe and Martin Hellman of Stanford University. It uses a pair of related keys one for encryption and other for decryption. One key, which is called the private key, is kept secret and other one known as public key is disclosed. The message is encrypted with public key and can only be decrypted by using the private key. So, the encrypted message cannot be decrypted by anyone who knows the public key and thus secure communication is possible. RSA (named after its authors – Rivest, Shamir and Adleman) is the most popular public key algorithm. In relies on the factorization problem of mathematics that indicates that given a very large number it is quite impossible in today's aspect to find two prime numbers whose product is the given number. As we increase the number the possibility for factoring the number decreases. So, we need very large numbers for a good Public Key Cryptosystem. GNU has an excellent library called GMP that can handle numbers of arbitrary precision. We have used this library to implement RSA algorithm. As we have shown in this paper number of bits encrypted together using a public key has significant impact on the decryption time and the strength of the cryptosystem.

II. CRYPTOGRAPHY

Cryptography from Greek , "hidden, secret" respectively is the practice and study of techniques for secure communication in the presence of third parties (called adversaries).More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce. Cryptography prior to the modern age was effectively synonymous with encryption, the conversion of information from a readable state to apparent nonsense. The originator of an encrypted message shared the decoding technique needed to recover the original information only with intended recipients, thereby precluding unwanted persons to do the same. Since World War I and the advent of the computer, the methods used to carry out cryptology have become increasingly complex and its application more widespread. Modern cryptography is heavily based on mathematical theory and computer science practice. Cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure; theoretical advances and faster computing technology require these solutions to be continually adapted. There exist information theoretically secure schemes that provably cannot be broken even with unlimited computing power—an example is the one-time pad—but these schemes are more difficult to implement than the best theoretically breakable but computationally secure mechanisms.

2.1 CRYPTOGRAPHY KEY BASICS

The two components required to encrypt data are an algorithm and a key. The algorithm generally known and the key is kept secret. The key is a very large number that should be impossible to guess, and of a size that makes exhaustive search impractical. In a symmetric cryptosystem, the same key is used for encryption and decryption. In an asymmetric cryptosystem, the key used for decryption is different from the key used for encryption.

2.2 KEY PAIR

In an asymmetric system the encryption and decryption keys are different but related. The encryption key is known as the public key and the decryption key is known as the private key. The public and private keys are known as a key pair. Where a certification authority is used, remember that it is the public key that is certified and not the private key. This may seem obvious, but it is not unknown for a user to insist on having his private key certified!

2.3 KEY COMPONENT

Keys should whenever possible be distributed by electronic means, enciphered under previously established higher-level keys. There comes a point, of course when no higher-level key exists and it is necessary to establish the key manually. A common way of doing this is to split the key into several parts (components) and entrust the parts to a number of key management personnel. The idea is that none of the key parts should contain enough information to reveal anything about the key itself. Usually, the key is combined by means of the exclusive-OR operation within a secure environment. In the case of DES keys, there should be an odd number of components, each component having odd parity. Odd parity is preserved when all the components are combined. Further, each component should be accompanied by a key check value to guard against keying errors when the component is entered into the system. A key check value for the combined components should also be available as a final check when the last component is entered. A problem that occurs with depressing regularity in the real world is when it is necessary to re-enter a key from its components. This is always an emergency situation, and it is usually found that one or more of the key component holders cannot be found. For this reason it is prudent to arrange matters so that the components are distributed among the key holders in such a way that not all of them need to be present. For example, if there are three components (C1, C2, C3) and three key holders (H1, H2, H3) then H1 could have (C2, C3), H2 could have (C1, C3) and H3 could have (C1, C2). In this arrangement any two out of the three key holders would be sufficient.

III. TYPES OF CRYPTOGRAPHY

3.1 Symmetric-key cryptography

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key. Symmetric key ciphers are implemented as either block ciphers or stream ciphers. A block cipher enciphers input in blocks of plaintext as opposed to individual characters, the input form used by a stream cipher. The data Encryption Standard(DES) and the Advanced Encryption Standard(AES) are block cipher designs which have been designated cryptography standards by the US government Despite its deprecation as an official standard, DES remains quite popular, it is used across a wide range of applications, from ATM encryption to email privacy and secure remote access. Many other block ciphers have been designed and released, either considerable variation in quality. Many have been thoroughly broken,, such as FEAL.Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit-by-bit or character –by-character, somewhat like the one time pad. In a stream cipher, the output stream based on a hidden state which changes as the cipher operates.

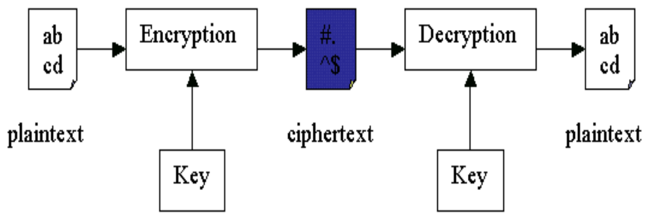


Figure 1: Principle of Cryptography

3.2 Asymmetric key Cryptography

Public-key cryptography refers to a cryptographic system requiring two separate keys, one of which is secret and one of which is public. Although different, the two parts of the key pair are mathematically linked.One key locks or encrypts the plaintext, and the other unlocks or decrypts the cipher text. Neither key can perform both functions by itself. The public key may be published without compromising security, while the private key must not be revealed to anyone not authorized to read the messages. Public-key cryptography uses asymmetric key algorithms (such as RSA), and can also be referred to by the more generic term "asymmetric key cryptography." The algorithms used for public key cryptography are based on mathematical relationships that presumably have no efficient solution. Although it is computationally easy for the intended recipient to generate the public and private keys, to decrypt the message using the private key, and easy for the sender to encrypt the message using the public key, it is extremely difficult (or effectively impossible) for anyone to derive the private key, based only on their knowledge of the public key. This is why, unlike symmetric key algorithms, a public key algorithm does not require a secure initial exchange of one (or more) secret keys between the sender and receiver. The use of these algorithms also allows the authenticity of a message to be checked by creating a digital signature of the message using the private key, which can then be verified by using the public key. In practice, only a hash of the message is typically encrypted for signature verification purposes.

IV. METHODOLOGY

RSA ALGORITHM

1. Choose two distinct prime numbers, p and q.
2. Let n = pq.
3. Let φ(pq) = (p − 1)(q − 1). (φ is totient function).
4. Pick an integer e such that 1 < e < φ(pq), and e and φ(pq) share no divisors other than 1 (e and φ(pq) are coprime).
5. Find d which satisfies
.de≡1 mod φ(pq)
d is a secret private key exponent.

1. The public key consists of e (often called public exponent) and n(often called modulus). The private key consists of e and d (private exponent). The message m is encrypted using formula Where c is the encrypted message. The encrypted message is decrypted using formula Encryption and decryption formulas show how to encode and decode a single integer. Bigger (or different) pieces of information are encoded by converting them into (potentially large) integers first. As RSA is not particularly fast, it is usually only to encrypts the key of some faster algorithm.After RSA decrypts the key, this supplementary algorithm uses it to decrypt the rest of the message. RSA algorithm is fundamentally based on the Euler theorem: Where a and n are positive integers and a is a co-prime to n.

To break the algorithm from the mathematical side, one needs to solve the factoring problem (find the two prime numbers that, when multiplied, produce the given result). When the picked numbers are large enough, the problem cannot be easily solved by brute force and at least currently it also does not have easier analytic solution

Encryption

Sender A does the following:-
1. Obtains the recipient B's public key (n, e).
2. Represents the plaintext message as a positive integer m, 1 < m < n .
3. Computes the cipher text c = me mod n.
4. Sends the cipher text c to B.

Decryption

The plaintext message can be quickly recovered when the decryption key d, an inverse of e modulo (p-1)(q-1) is known. (Such an inverse exists since gcd(e,(p-1)(q-

1))=1). To see this, note that if d e≡1 (mod (p-1)(q-1)), there is an integer k such that d e = 1 + k(p-1)(q-1).

Working Principle of RSA Algorithm

1. To generate the primes p and q, generate a random number of bit length b/2 where b is the required bit length of n; set the low bit (this ensures the number is odd) and set the two highest bits (this ensures that the high bit of n is also set); check if prime (use the Rabin-Miller test); if not, increment the number by two and check again until you find a prime. This is p. Repeat for q starting with a random integer of length b-b/2. If p<q, swop p and q (this only matters if you intend using the CRT form of the private key). In the extremely unlikely event that p = q, check your random number generator.

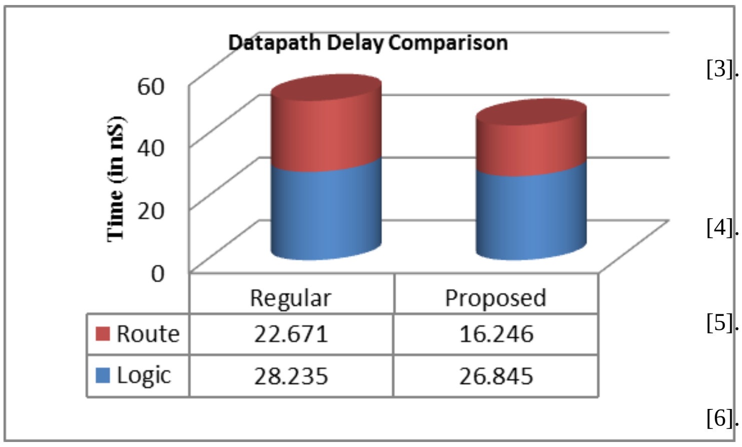
   Alternatively, instead of incrementing by 2, just generate another random number each time. There are stricter rules in ANSI X9.31 to produce strong primes and other restrictions on p and q to minimize the possibility of known techniques being used against the algorithm. There is much argument about this topic. It is probably better just to use a longer key length.
2. In practice, common choices for e are 3, 17 and 65537 (216+1). These are Fermat primes, sometimes referred to as F0, F2 and F4 respectively (Fx=2^(2^x)+1). They are chosen because they make the modular exponentiation operation faster. Also, having chosen e, it is simpler to test whether gcd(e, p)=1 and gcd(e, q-1)=1 while generating and testing the primes in step 1. Values of p or q that fail this test can be rejected there and then. (Even better: if e is prime and greater than 2 then you can do the less-expensive test (p mod e)!=1 instead of gcd(p-1,e)==1.
3. To compute the value for d, use the Extended Euclidean Algorithm to calculate d = e-1 mod phi, also written d = (1/e) mod phi. This is known as modular inversion. Note that this is not integer division. The modular inverse d is defined as the integer value such that ed = 1 mod phi. It only exists if e and phi have no common factors.
4. When representing the plaintext octets as the representative integer m, it is usual to add random padding characters to make the size of the integer m large and less susceptible to certain types of attack. If m = 0 or 1 or n-1 there is no security as the cipher text has the same value. For more details on how to represent the plaintext octets as a suitable representative integer m, see PKCS#1 Scheme below or the reference itself [PKCS1]. It is important to make sure that m < n otherwise the algorithm will fail. This is usually done by making sure the first octet of m is equal to 0x00.
5. Decryption and signing are identical as far as the mathematics is concerned as both use the private key.Similarly, encryption and verification both use the same mathematical operation with the public key.That is, mathematically, for m < n,m = (me mod n)d mod n = (md mod n)e mod n
6. The original definition of RSA uses the Euler totient function φ(n) = (p-1)(q-1). More recent standards use the Charmichael function λ(n) = lcm(p-1, q-1) instead. λ(n) is smaller than φ(n) and divides it. The value of d' computed by d' = e-1 mod λ(n) is usually different from that derived by d = e-1 mod φ(n), but the end result is the same. Both d and d' will decrypt a message me mod n and both will give the same signature value s = md mod n = md' mod n. To compute λ(n), use the relation7. λ(n) = (p-1)(q-1) / gcd(p-1, q-1)

V. RESULTS AND DISCUSSION

The proposed system architecture and the existing architecture of RSA implemented in given by [1] were compared by modeling them using Verilog and simulating for 16 bit inputs. The performance is compared in terms of area, speed and power. The analysis results were obtained using Xilinx 13.2 ISE & I-Simulator. Prime numbers from the generated random numbers were identified and again stored in a FIFO memory. The first two prime numbers stored in the FIFO is selected for encryption. But the proposed architecture eliminates the need for memories by checking for primality and selects two prime numbers simultaneously while the random numbers were being generated. Also the generation stops as soon as the system detects two prime numbers.

SPEED COMPARISON

The performance matrix which is being compared in this paper is the speed. The time delay is obtained which clearly shows that the time delay in case of proposed design is lesser as compared to the regular design[1] and hence the speed is improved. Following is the comparison of time delay in both the designs:



	Regular	Proposed
🟥 Route	22.671	16.246
🟦 Logic	28.235	26.845

* At slow process corner

Figure 2: Comparison of datapath delay between Regular & Proposed Design

From the above graph shown in fig 8 it can be observed that the total datapath time delay in existing regular design is 50.906 nsec which is reduced to 43.091 nsec in the proposed design showing reduction of almost 15.35%. Fig 9 also shows comparison of logic datapath and route datapath delay of both the designs. This reduction in delays henceforth shows increase in speed of operation of circuit.

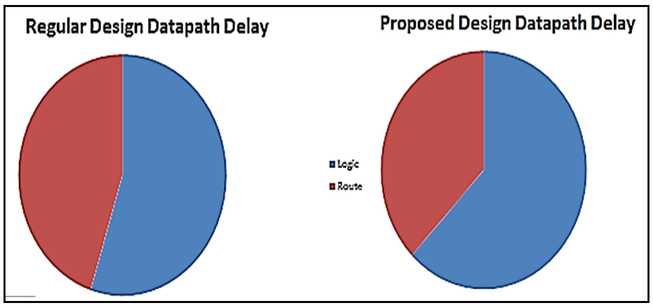


Figure 3: Comparison showing logic & route data path delay between Regular & Proposed Design

Strongly encouraged not to call out multiple figures or tables in the conclusion these should be referenced in the body of the paper.

VI. REFERENCES

[1]. Jainath Nasreen.P, Denila.N, Emy Ramola.P "A Novel Architecture for VLSI Implementation of RSA Cryptosystem" 2012 International Conference on Computing, Electronics and Electrical Technologies [ICCEET],IEEE

[2]. Md. Ali-Al-Mamun, Mohammad Motaharul Islam, S.M. Mashihure Romman and A.H. Salah Uddin Ahmad "Performance Evaluation of Several Efficient RSA Variants" IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.7, July 2008 7

[3]. Ren-Junn Hwang, Feng-Fu Su, Tamsui, Taipei, Yi-Shiung Yeh, Chia-Yao Chen "An Efficient Decryption Method for RSA Cryptosystem" Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05) 2005 IEEE

[4]. Guang Gong and Lein Harn "Public-Key Cryptosystems Based on Cubic Finite Field Extensions" IEEE Transactions On Information Theory, VOL. 45, NO. 7, NOVEMBER 1999

[5]. Mohammad Tehranipoor, Mehrdad Nourani, Nisar Ahmed "Low-Transition LFSR for BIST-Based Applications" 14th Asian Test Symposium, 2005 Proceedings.

[6]. W. Diffie and M. E. Hellman, New directions in cryptography, IEEE Trans. Inform. Theory, Nov. 1976,22: 644-654.

[7]. R.L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM, Feb. 1978, 21(2): 120-126.

[8]. E. F. Brickell, A fast modular multiplication algorithm with application to two-key cryptography, in Proc. CRYPTO'82 Advances Cryptology,Plenum Press, New York and London, 1982: 51-50.

[9]. M. Preetha, M. Nithya "A Study and Performance Analysis of RSA Algorithm" IJCSMC, Vol. 2, Issue. 6, June 2013

[10]. R.L.Rivest,A.Sharmir,L.Adleman : A method for obtaining digital signatures and public key Cryptosystems", Tata McGraw-Hill

[11]. W. Stallings, Cryptography and Network Security: Principles and Practice 3/e. Prentice Hall

[12]. Cormen , Thomas H, Charles E.Leiserson, aronald L.Rivest Clifford Stein "Introduction to algorithms". MIT Press and McGraw-Hill

[13]. www.rsa.com

[14]. www.cc.gatech.edu

[15]. www.symmtech.com

[16]. www.rsasecurity.com

[17]. www.cryptotools.com