Public Key Cryptosystem based on Pell's Equation Using The Gnu Mp Library

K V S S R S S Sarma[1], P S Avadhani[2]
kss@uohyd.ernet.in[1], psavadhani@yahoo.com[2]
University of Hyderabad[1], Hyderabad, Andhra University[2], Visakhapatnam, India

Summary

Protection of data is the utmost thing for any company related to digital information. There are several malicious methods adapted, based on the priority of demand of that piece of information. There are several cryptosystems implementing various algorithms. We are developing a cryptosystem based on Pell's Equation. RSA algorithm is extensively used in the popular implementations of Public Key Infrastructures. In this paper, we have done an efficient implementation of RSA algorithm and Pell's Equation using GMP library from GNU. We have also analyzed the changes in the performance of the algorithm by changing the number of bits of message and keys.

Key Words :
Pell's Equation , Public key cryptosystem, Cryptography, RSA, PKI, MP, GNU.

1. 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 transmutated 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 decrypt 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 arbitrar2y precision. We have used this library to implement RSA algorithm and Pell's Equation. 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.

The basic issue in cryptography is to communicate securely over an insecure channel. The two communicating parties need to authenticate and maintain confidentiality in order to protect the privacy of their messages. There are many algorithms proposed in the literature to provide the confidentiality and authentication. There are many cryptosystems [1], [2], [3], [4] proposed in this direction of which the public key encryption methods are more prominent. In this paper, based on the previous work of [5], [6], we propose a Pell equation based public key cryptosystem for encryption/decryption which can deal with high bit keys.

To deal with large numbers, GNU has an excellent library called GMP that can handle numbers of arbitrary precision. We have used this library to implement our Pell's Equation and RSA algorithm. Number of bits of the key for encryption has a significant impact on the decryption time and the strength of the cryptosystem.

The significance of the proposed cryptosystem is that it is not only two times faster than the RSA Scheme but also as secure as RSA against chosen cipher text attack[30]. The new model is also as secure as RSA against partially known plain text attack [6]. Moreover it is more secure than RSA scheme when purely

common plain texts are encrypted in the broadcast application. Also, the addition operation in the proposed model is computationally less expensive than the schemes/models based on the cubic curve [7], [8]. Thus the encryption processesses are more efficient than the other RSA type cryptosystems based on the cubic curve.

2. REVIEW OF EXISTING LITERATURE

Authentication protocols and their implications are discussed in [1]. Computing inverse of a shared secret modulus, which involves mathematical formulation of RSA, is discussed in [2]. Application of hash function in the field of cryptography is discussed in [3]. The strength of RSA algorithm is discussed in [4]. A survey of fast exponentiation method is done in [5]. Cryptosystem for sensor networks is studied in [6]. Security proofs for various digital signature scheme is studied in [7]. Multiparty authentication services and key agreement protocols are discussed in [8]. Various fast RSA implementations are described in [9]. An efficient implementation of RSA is discussed in [10]. The basic RSA algorithms and other cryptography related issues are discussed in [11]. Implementation of the RSA algorithm using the GNU MP library has been done in [31].

3. SCOPE OF OUR PRESENT WORK

Our work in this paper is focused primarily on the implementation of RSA and PELL'S EQUATION. For efficient implementation we have used the GMP library, we have explored the behavior and feasibility of the algorithm with the change of various input parameters, and finally a user interface is developed to provide an application of our analysis. Apart from the analysis application, we have also developed a cryptosystem for basic encryption / decryption of text using either of the methods mentioned in this paper.

4. PELL'S EQUATION

Introduction

We describe a cyclic group G_p over the pell equation $x^2 - D y^2 \equiv 1 \pmod{P}$, where P is an odd prime. Some properties of the group G_p are then deduced. These properties are also found in the group G_N over the pell equation $x^2 - D y^2 \equiv 1 \pmod{N}$, where N is a product of two primes. This group G_N then developed to be a public key crypto scheme based on Pell's equations over the ring Z_N^*. From the group G_N, we find a group isomorphism mapping $f : G_N \rightarrow Z_N^*$ such that a solution (x, y) of the Pell's equation $x^2 - D y^2 \equiv 1 \pmod{N}$, can easily be transformed to unique element u in Z_N^*. This implies that the plain texts/cipher texts in the in the group G_N can easily transformed to the corresponding plain texts/cipher texts in the RSA scheme.

Key Generation

Recipient (R) chooses two large primes p and q. Let N = p*q and N = lcm (p − 1, q − 1). R determines an integer e satisfying gcd (e, N) = 1. Decryption keys d is computed from encryption key e as $d = e^{-1} \bmod N$ by using the Euclidean algorithm. The pair (e, n) is the public key and private key is (p, q, d).

Pell's Encryption Scheme :

The Sender S performs the following operations. Z_1 is computed such that $Z_1 \equiv MxMy$ and Y is taken as My. He solves the equation $X - aY \equiv Z_1$ and $X + aY \equiv Z_1^{-1}$. Get $X \equiv (Z_1 + Z_1^{-1})/2$ and $a \equiv (Z_1^{-1} - X)/Y$ and $D \equiv a^2$. Hence (X,Y) is the solutions for the Pell's Equation. Next we find $M \equiv (X - aY)$ and C=Me. Then the Cipher text (C,a) is send to the recipient R.

Pell's Decryption Scheme :

After receiving the Cipher text (C, a) the recipient R proceeds as follows. He computes $M \equiv C\ d$. Then using M, he computes X and Y by $X \equiv (M^{-1} + M)/2$ and $Y \equiv (M^{-1} - M)/2a$. This implies that $Z1 \equiv M$. Therefore, My=Y and $Mx \equiv M/Y$. Above two schemes are not semantically secure. To get semantically secure public key cryptosystem we generalize the scheme-II as below.

5. GNU GMP

Introduction

GNU MP is a portable library written in C for arbitrary precision arithmetic on integers, rational numbers, and floating-point numbers. It aims to provide the fastest possible arithmetic for all applications that need higher precision than is directly supported by the basic C types.

Many applications use just a few hundred bits of precision; but some applications may need thousands or even millions of bits. GMP is designed to give good performance for both, by choosing algorithms based on the sizes of the operands, and by carefully keeping the overhead at a minimum.

The speed of GMP is achieved by using full words as the basic arithmetic type, by using sophisticated algorithms, by including carefully optimized assembly code for the most common inner loops for many different CPUs, and by a general emphasis on speed (as opposed to simplicity or elegance).

Description to the special operations of GMP

There is a gmp routine specifically for a computation having the prototype **void mpz_powm (mpz_t ROP, mpz_t BASE, mpz_t EXP, mpz_t MOD)** which sets ROP to (BASE raised to EXP) modulo MOD. Thus invoking moz_powm(c, m, e,n) stores the encrypted partial message in the integer c.

There is a gmp routine specifically to compute the product of an integer and inverse of the integer called **int mpz_invert (mpz t rop, mpz t op1, mpz t op2)**. Which computes the inverse of op1 modulo op2 and put the result in rop.

There is a gmp routine specifically to compute the modulus which is extremely important for our implementation. **void mpz_mod (mpz t r, mpz t n, mpz t d)** which sets r to n mod d. The sign of the divisor is ignored and the result is always non-negative.

6. Results

Timings for algorithms with varying bit strengths (512 to 2048)

RSA Scheme

Key Generation	Encryption	Decryption
0.057984	0.054362	0.90318
0.194653	0.065302	2.098904
0.465994	0.078851	3.365591
0.657473	0.089712	4.437929
1.613467	0.105439	5.804798
2.057411	0.116585	7.430849
4.052181	0.126361	9.001286

Table I

Pell's Scheme

Key Generation	Encryption Time	Decryption Time
0.057984	0.044362	0.70498
0.194653	0.059502	1.797604
0.465994	0.069651	3.012591
0.657473	0.079212	4.242814
1.613467	0.099439	4.989204
2.057411	0.109938	6.630849
4.052181	0.113361	7.689686

Table 2

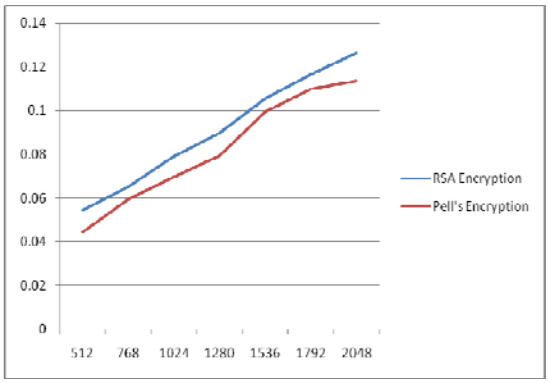

Fig 1

(Comparison of the Encryption algorithms of Pell's , with existing RSA algorithm)

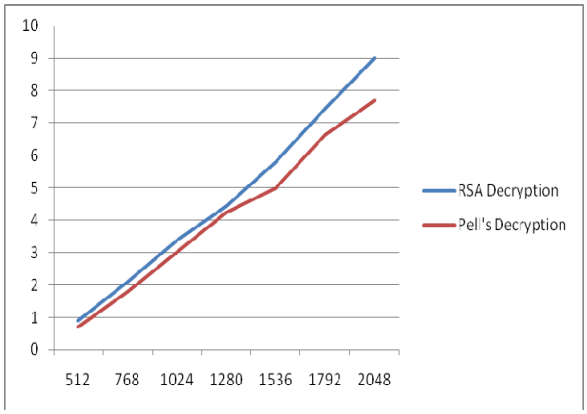

Fig 2

(Comparison of the Decryption algorithms of Pell's , with existing RSA algorithm)

7. Efficiency

We first compare our scheme with the standard RSA [26] scheme. Here, we focus on the decryption method to calculate the average number of modular multiplications. Generally, M ≡ Cd (mod n) needs 1.5log d multiplications modulo n on average. Besides, the cost of isomorphism mapping requires two modular inverses and one modular multiplication. In addition, to compute Mx , one modulo multiplication and one inverse is required during the decryption process. Since, one modulo inverse needs six modulo multiplications [7, 8], the decryption of proposed scheme requires 1.5log d + 20 modular multiplication on average. Neglecting the cost of isomorphic mapping, the proposed scheme has almost the same decryption time as RSA scheme. 2 − log n bit message is encrypted at a time by our proposed scheme, so block size is two times larger than the standard RSA crypto system[29]. In the standard RSA scheme, to decrypt for 2 − log n-bit message requires 3log d multiplication modulo n on average. Thus, the decryption efficiency of our new crypto system would be about two times faster than that of the RSA scheme for a k bit long message if k/log n is even[29].

8. Conclusion

In this paper an efficient implementation of public key cryptosystem based on Pell's equation is shown by using different functions of the GMP library. Analysis is done by comparing the time taken for encryption and decryption algorithms of Pell's equation with existing RSA algorithm. In this context Pell's equation and RSA having their own limitations. The values of operands are to be taken in such a way that the overhead is kept at minimum. And finally it shows that when we increase the number of bits of information to be processed together, the total time including encryption and decryption decreases comparatively with RSA.

9. References

[1] Bleichenbacher D., On the security of KMOV public key cryptosystem. LNCS Crypto'97v.1294, 235-348(1997).
[2] Chen C.Y., Chang C.C. Yang W.P. Fast RSA type cryptosystem based on Pell equation. Proceeding of International Conf. On Cryptology and Information Security Taiwan, Dec.1-5, 1996.
[3] Coppersmith D., M. Franklin, J. Patarin and M Reiter., Low exponent RSA with related messages. Eurocrypt'96 LNCS 1070, pp 1-9, Springer Verlag (1996).
[4] Catalano D., R. Gennaro, N. Howgraw-Crahan and P. Nguyen,Paillier's cryptosystem revisited. ACM Conference on Computer and Communication Security (2001).
[5] Coppersmith D. , Finding a small root of a bivariate integer equation; factoring with high bits known. Advances in Cryptology-Eurocrypt'96, LNCS vol.1070, Springer-Verlag, 1996, pp.178-189.
[6] O'Connor, J.J and Robertson,E. F. ,February 2002.Pell's Equations.[Online], Available: http/www-history.mcs.st-andrews.ac.uk/HistTopics/Pell.html.
[7] Chiou, C.W. and Yang T.C., Iterative modular multiplication algorithm without magnitude comparison. Electronic Letters, v.130, no. 24, 1994 pp.2017-1018.
[8] Demytko N., A new elliptic curve based analogue to RSA. LNCS EUROCRYPT'93, 40-49(1993).
[9] Diffie W.and Hellmann Martin, New direction in cryptography.IEEE Transaction on Information Theory, v.22, 1976, 644-654.
[10] Hasted J., On the using RSA with low exponent in a public key network. LNCS Crytpo'85, V.218 pp. 403-408(1985).
[11] Kaliski Jr.B.S., The Montgomery inverse and its applications. IEEE Transaction on Computers, vo.8, Aug.1995 pp.1064-1065.
[12] Koyama K., H. Kawakado A new RSA type scheme based on singular cubic curve $(y - ax)(y - bx) = x3 \bmod N$ IEICE Trans. Fund E79-A, pp49-539(1996).
[13] Kuwadado H, Koyama K., Y.Tsuraoka, A new RSA type scheme based on singular cubic curve $y\ 2 = x3 + bx2 \bmod N$, IEICE Trans.Fund E78-A, 27-33(1995).
[14] Koyama K., U.Maurer, T. Okamoto, S.A.Vanstone, New public key schemes based on elliptic curves over the rung Zn, Crypto'91 252-266 (1991).
[15] Koblitz Neal, Elliptic curve cryptosystem, Math.Comput.48.203-209(1985).
[16] Koyama Kenji, Fast RSA type scheme based on singular cubic curve $y\ 2 + axy = x3 \bmod N$, Eurocrypt'95 329-339 (1995).
[17] Kouichi S. and Tsuyoshi Takagi. New semantically secure public key cryptosystems from RSA-Primitive. LNCS PKC'02 Vol.2274,pp.1-16(2002).
[18] Lenstra H.W Jr. Solving the Pell equation. Notice of AMS v.49 no.2 186-192 (2002).
[19] Lenstra H.W.Jr. Factoring integers with elliptic curve. Annals of Mathematics 126,pp 649-673(1987).
[20] Menezes A., Elliptic curve pubic key cryptosystem . Kluwer Acad. Pub. 1993.
[21] Miller V., Uses of elliptic curve in cryptography, LNCS CRYPTO'85pp 417-426(1985).
[22] Marc Gysin, Jennifer Sebery, How to use Pell's equation in cryptography.Preprint.
[23] O'Connor, J.J and Robertson,E. F. ,February 2002. Pell's Equations.[Online], Available:http/www-history.mcs.st. Andrews. ac.uk/HistTopics/Pell.html.
[24] Sahadeo Padhye, Partial known plaintext attack on Koyama scheme. Information Processing Letters, 96/3 pp.96-100(2005).
[25] David Pointcheval, New public key cryptosystem based on the dependent-RSA problem. Eurocrypt'99 LNCS Springer-Verlag , vol.1592, pp.239-254, (1999).
[26] Rivest R.L. , Shamir A. and Adllemann L., A method for obtaining digital signature and public key cryptosystems. Comm. Of the ACM 21, 2 pp. 120-126 (1978).
[27] Simmons G.J, editor .Contemporary Cryptology-The Science of Information Integrity. IEEE Press, 1992
[28] Well A., Number theory, an approach through history. Birkhausiur Boston 1984. Demytko N., A new elliptic curve based analogue to RSA. LNCS EUROCRYPT'93,40-49(1993).
[29] Sahadeo Padhye, A public key cryptosystem based on pell equation
[30] K V S S R S S Sarma, P.S. Avadhani; Pell's Equation based Cryptosystem for providing Confidentiality and Authentication.
[31] Rajorshi Biswas, Shibdas Bandyopadhyay, Anirban Banerjee; A fast implementation of the RSA algorithm using the GNU MP library