Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 9, August 2021: 7755-7762

Research Article

Some Noteworthy Cryptosystem and Its Characteristics Are Analyzed by Fuzzy Logic

Dr. Ashutosh Pandey

Assistant Professor, Department of Mathematics, D.P. VIPRA P.G.College Bilaspur (C.G.) Pin- 495001
Email Id- ashutoshpandey28jd@gmail.com

Abstract

In this research ,we propose the "Fuzzy-Crypto"and this associates to "Fuzzy Cryptography" which reflects the idea of using the uniqueness of Fuzzy in the arbitrary and limited sense as the image or impression of parts of body or any another type of biological related terms directly or indirectly as recorded or live and exclusive performance based cryptography.

Introduction

Cryptanalysis means Cryptosystem and its analysis . The security and the efficiency are the two main parameters which lies in the cryptanalysis. The hard mathematical problems interacts with the security of any cryptosystem .Basically, the security of any cryptosystem proportionate to the hard problems of the mathematics.

Although efficiency corresponds to the computer science .Thus ,in cryptanalysis as the function of security and efficiency, mathematics and the computer science are corresponding bivariate.In addition fuzzy logic and fuzzy set are the another tool to analyze the cryptosystem .Here ,cryptanalysis is given with its historical journey and then the domain of cryptanalysis is presented.

Review of liteture :-

Menzes et al [13]'s book is the unique and complete source of knowledge in real and practical cryptography, In 2004, Miller[14] presented the weil pairing based cryptosystems. Miyaji et al[15] proposed new security scheme on elliptic curves in 2001. Okamato etal[16] put the gap problems in 2001.

In 2004,page eta 1[17] compared the cures. Press et al [18] drafted the numerical orientation of the security systems. In 2001, Rivest et al [19] focused on inverse aspect of the security. In 2002, Rubin et al [20] launched new cryptosystems. Scott [22,23,24] presented some advance results on curves, shamir[25] introduced identity based cryptosystems in 1984.

In 1985, silverman[26] originated the arithmetic concept of elliptic curves. In 2006,stange[27]shown the Tate-pairing relationship in security, stinson[27]inveted the new algorithms on the baby-ste[giant -step algorithm in 2002.

In 2004, verheul[29] put the evidence of STRsecurity. Weimeskrich[30] designed the karatsoba algorithm for polynomial multiplication in 2006.

Let,

$F = GF(p) = Z_p$: Finite Field,

$\{x_1,...,x_n\}$: Elements of Finite field,

μ : Grade or Membership Rule,

(F,μ): Fuzzy Set,

S: Statement,

$\mu: F \to S$: Fuzzy Logic.

The RSA Algorithm [10]

The RSA cryptosystem is a public-key cryptosystem that offers both encryption and digital signatures/authentication. The RSA system is used in a wide variety of products, and industries throughout the world. RSA is found in many commercial software products. The RSA algorithm is built into operating systems by Microsoft, Apple, Sun, and Novell. The RSA algorithm can also be found in secure telephones, Ethernet network cards, and smart cards.

RSA is used by many people; however, in order to look at why people choose this encryption method over any other, we need to see what RSA does. A general idea of how this works is as follows:

Take two large primes, *p* and *q*, compute their product. Let *n* be the modulus, or remainder, from this division. Choose a number, *e*, to be less than $n \in (F,\mu)$, and relatively prime to $(p-1)(q-1) \in (F,\mu)$. In other words, *e* and $(p-1)(q-1) \in (F,\mu)$ have no common factors except 1. Find another number *d* such that $(ed-1) \in (F,\mu)$ is divisible by $(p-1)(q-1)$. Now we have *e* and *d* which are the public and private exponents. The public key is the pair (n,e); the private key is (n,d). The factors *p* and *q* may be kept with the private key, or destroyed.

It is currently a difficult problem to find the private key *d* from the public key (n,e). If one could easily factor *n* into *p* and *q*, then one could find the private key *d*. Because of this, we can see that the security of the RSA system is based on the assumption that factoring is difficult. A way to break RSA would be to find an easy method of factoring large numbers. However, until this method is created, the usage of RSA will continue. Part of the reason for the use of RSA is that it has a relatively easy operation for encryption.

For understanding the working process of The RSA, we must first start with a corollary to the Totient theorem. This is as follows:

$$\gcd(a,n)=1 \text{ and } a<n \text{ then } a^{(k\phi(n))} = 1(\bmod n) \in (F,\mu):\ \forall k \in Z$$

To come across the RSA algorithm we manipulate this corollary further.

$$\gcd(a,n)=1 \text{ and } a<n \text{ then } a^{s} = 1(\bmod n) \text{ where } s = 0(\bmod \phi n) \in (F,\mu)$$

$$a^{\phi n} = 1(\bmod n) \qquad \in (F,\mu)$$

$$a^{\phi n} \cdot a^{\phi n} = 1 \cdot 1(\bmod n) \qquad \in (F,\mu)$$

$$a^{\phi n+\phi n} = 1 \cdot 1(\bmod n) \qquad \in (F,\mu)$$

$$a^{2\phi n} = 1(\bmod n) \qquad \in (F,\mu)$$

$$a^{s} \cdot a = 1 \cdot a(\bmod n) \qquad \in (F,\mu)$$

$$a^{s+1} = a(\bmod n)$$

$$a^{s+2} = a^2 (\mathrm{mod}\, n) \qquad \in (F, \mu)$$

$$\vdots$$

$$a^e = a^f (\mathrm{mod}\, n) \text{ whenever } e = f (\mathrm{mod}\, \phi n) \qquad \in (F, \mu)$$

So, with the above work, we have created a very useful phenomenon. We now have a function that returns its own input. I.e. $a^{s+1} = a (\mathrm{mod}\, n) \in (F, \mu)$ is a function that has *a* on both sides. This function is the basis of what RSA is founded upon.

Now we demonstrate the encryption and decryption algorithm as follows;

Let $p \cdot q = s + 1 \in (F, \mu)$ or to put it another way $p \cdot q = 1 (\mathrm{mod}\, \phi n) \in (F, \mu)$. (Note: *p* and *q* cannot have a factor in common with $\phi(n)$ otherwise $(p \cdot q) \mid \phi(n)$ will not have a remainder of 1.) With this we can continue manipulating this function.

$$a^{pq} = a (\mathrm{mod}\, n) \qquad \in (F, \mu)$$

$$(a^p)^q = a (\mathrm{mod}\, n) \qquad \in (F, \mu)$$

This function can now be split into two separate parts:

$$a^p = X (\mathrm{mod}\, n) \in (F, \mu) \text{ and } X^q = a (\mathrm{mod}\, n) \in (F, \mu)$$

Almost every possible mapping of *a* to *X* exists. This cipher is hard to break because the modulus arithmetic erases too much information. If *a*, *p*, and *X* are known *q* still cannot be found easily.

Let $E(T) \in (F, \mu)$ be the encryption process that changes plaintext (*T*) to ciphertext(*X*). In other words $E(T) = T^p \bmod n = X$. The decryption process would then be $D(X) = X^q \bmod n = T$. The simple explanation for why this works is $D(E(T)) = (T^p \bmod n)^q \bmod n = (X)^q \bmod n = T$. So the equation gives back its original value. The next question that is raised now that we know how RSA works is why RSA works.

For the mathematical explanation as to why RSA works I will refer to the above functions and theorems. To start with this explanation I will prove that *D(E(T)) =T* mod *n*.

$$D(E(T)) = (T^p \bmod n)^q \bmod n \qquad \in (F, \mu)$$

$$D(E(T)) \equiv T^{pq} \bmod n \qquad \in (F, \mu)$$

$$< p \cdot q = k\phi(n) + 1 \text{ from earlier } > \in (F, \mu)$$

$$D(E(T)) \equiv T^{k\phi(n)+1} \bmod n \qquad \in (F, \mu)$$

$$D(E(T)) \equiv (T^{\phi(n)})^k \cdot T \,(\mathrm{mod}\, n)$$

But $T^{\phi(n)} = 1 \pmod{n} \in (F, \mu)$ with Euler's theorem

$$D(E(T)) \equiv 1^k \cdot T \pmod{n} \in (F, \mu)$$

$$D(E(T)) \equiv 1 \cdot T \pmod{n} \in (F, \mu)$$

$$D(E(T)) \equiv T \pmod{n} \in (F, \mu)$$

This proof of the RSA algorithm only works if *T* and *n* are co prime. SO this proof is not quite correct. Another way to approach this algorithm is with Fermat's Little Theorem.

$$p \cdot q = k\phi(n) + 1 \quad \in (F, \mu)$$

$$p \cdot q = 1 + k(p-1)(q-1) \text{ for some } k > 0 \in (F, \mu)$$

$$(T^p \bmod n)^q \pmod{n} \equiv T(T^{(p-1)})^{k(q-1)} \pmod{n} \in (F, \mu)$$

$$< T^{(p-1)} \equiv 1 (\bmod\ p) > \in (F, \mu)$$

$$T^{p(\bmod n) q(\bmod n)} \equiv T (\bmod\ p) \in (F, \mu)$$

If *p* divides *T*, then *T* and $T^{p(\bmod n) q(\bmod n)} \in (F, \mu)$ are congruent 0 modulo *p*. So the congruence holds and $D(E(T)) \equiv T \pmod{n} \in (F, \mu)$.

The XTR Algorithm [5]

The XTR public-key algorithm was introduced in 2000 by Arjen Lenstra and EricVerheul. It was designed specifically with efficiency in mind and much of the underlying theory is based on the earlier work of Andries Brouwer, Ruud Pellikaan and Eric Verheul, as described in their paper, "Doing More with Fewer Bits". In this paper, the authors describe a variant of the classical Diffie-Hellman key exchange scheme that allows the total number of bits exchanged to be reduced to one third of the number in the original scheme. Unfortunately, the potential increase in performance that this should offer was not fully realised. The problem lay in the particular way that the algorithm was implemented. The computational inefficiencies inherent in the proposed method led to a system that was both impractical and not especially fast, despite its communication advantage. XTR significantly improves upon this algorithm and achieves the same communication efficiency at a greatly reduced computational cost. As with its predecessor, XTR is *not* a new cryptosystem; rather it is an efficient and compact implementation of the classical Diffie-Hellman scheme. Thus, we can be confident of its security. The real area of interest lies in its performance

XTR is the first method we are aware of that uses GF(p^2) $\in (F, \mu)$ arithmetic to achieve GF(p^6) $\in (F, \mu)$ security, without requiring explicit construction of GF(p^6) $\in (F, \mu)$. Let g be an element of order q > 6 $\in (F, \mu)$ dividing $p^2-p+1 \in (F, \mu)$. Because p^2-p+1 divides the order p^6-1 of GF(p^6)_ this g generates an order q subgroup of GF(p^6)_. Since q does not divide any $p^s - 1$ for s = 1, 2, 3, the subgroup generated by g cannot be embedded in the multiplicative group of any true subfield of GF(p^6). We show, however, that arbitrary powers of g can be represented using a single element of the subfield GF(p^2), and that such powers can be computed efficiently using arithmetic operations in GF(p^2) while avoiding arithmetic in GF(p^6).

Representation of an Element of GF(p^2) $\in (F, \mu)$***:***

Let *p* be a large prime such that *p* = 2 (mod 3).Let α and α*p* be zeros of $Y^2 + Y + 1$ that form an optimal normal basis for GF(p^2) over GF(*p*).

Since $\alpha^i = \alpha^i \pmod 3$ an element *x* ε GF(p^2) Using this representation results in a large reduction in the cost of many arithmetic operations in GF(*p*2). The following table shows how several of these operations are carried out and specifies the cost of each (as is customary, the cost of additions in GF(*p*) are not counted):can be represented as:

$x_1\alpha + x_2\alpha^p = \mathbf{x_1}\alpha + \mathbf{x_2}\alpha^2$ where x1, x2 ε GF(p).
Let x, y, z belongs to GF(p^6) with p = 2 mod 3, and let a, b belongs to Z with 0 < a, b < p. Assume that a squaring in GF(p) takes 80% of the time of a multiplication in GF(p) we can state some of the postulates like

i. Computing x2 takes 14.4 multiplications in GF(p).

ii. Computing x _ y takes 18 multiplications in GF(p).

iii. Computing xa takes an expected 23.4 log2(a) multiplications in GF(p).

iv. Computing xa _ yb takes an expected 27.9 log2(max(a, b)) multiplications in

GF(p).

Having established how the arithmetic in GF(p^2) is performed, we are now in a position to examine the algorithm used for XTR exponentiation. To aid in this process, I first state the following definitions:

Definition 1: For c ε GF(p2) let $F(c, X)$ be the polynomial $X^3 - cX^2 + c_pX - 1$

ε GF(p2)[X] with roots h_0, h_1, h_2 in GF(p^6) and let $c_n = h_0^n + h_1^n + h_2^n$ for n ε **Z**.

Definition 2: Let $S_n(c) = (c_{n-1}, c_n, c_{n+1})$ ε GF(p^2)^3.

Computation of Sn(c) given c If n < 0, apply this algorithm to −n and If n = 0, then S0(c) = (cp, 3, c) If n = 1, then S1(c) = (3, c, $c^2 - 2c^p$) If n = 2, use Corollary 2.3.5.ii and S1(c) to compute c3 and thereby S2(n).

B. To compute Sn(c) for n > 2

Computation of Sn(c) given c (where n > 2):

1) Compute $S3(c) = (c2, c3, c4)$

where $c2 = c^2 - 2c$

$c3 = cc_2 - cc^p + 3$

$c4 = cc_3 - c2c^p + c$

2) If n is odd let $m = (n - 1) / 2$ else let $m = (n -$

2) / 2.

3) Let L be the bit-length of m.

4) Let $x = c2$, $y = c3$, $z = c4$.

5) For $i = L - 2, L - 3, L - 4, \ldots, 0$ in succession do the following:

Let mi be the ith bit of m (the least significant bit is the 0th bit).

If $mi = 0$:
$x' = x^2 - 2x^p$

$y' = xy - c^py^p + z^p$

$z' = y^2 - 2y^p$

Else $mi = 1$:

$x' = y^2 - 2y^p$

$y' = yz - cy^p + x^p$

$z' = z^2 - 2z^p$

$x = x'$, $y = y'$, $z = z'$

6) If n is even:

$x' = y$

$y' = z$

$z' = cz - c^p y + x$

$x = x', y = y', z = z'$

7) $Sn(c) = (x, y, z)$

C. Computation of trace group Tr(g):

Computation of $Tr(ga * gbk)$ given $Tr(g)$, $Sk(Tr(g))$ for unknown k and $a, b \varepsilon Z$ with $0 < a, b < q$:

1) Compute $e = a / b \pmod q$.

2) Compute $S_e(Tr(g))$ using Algorithm 1.

3) Compute the matrix M where:

$$M = \frac{1}{D} * \begin{pmatrix} 2c^2 - 6c^p & 2c^{2p} + 3c - c^{p+2} & c^{p+1} - 9 \\ 2c^{2P} + 3c & (c^2 - 2c^p)^{p+1} - 9 & (2c^{2p} + 3c - c^{p+2})^p \\ c^{p+1} - 9 & (2c^{2p} + 3c - c^{p+2})^p & (2c^2 - 6c^p)^p \end{pmatrix}$$

and

$$D = 2c^{2p+2} + 18c^{p+1} - 4(c^{3p} + c^3) - 27$$

4) Compute $C(A(Tr(g))^e) = M * (S_e(Tr(g)))^T$.

5) Compute $Tr(g^{e+k}) = S_k(Tr(g)) * C(A(Tr(g))^e)$.

6) Compute $S_b(Tr(g^{e+k}))$ using above algorithm.

7) Return $Tr(g^{(e+k)b}) = Tr(g^a * g^{bk})$.

It is worth noting that the XTR algorithm itself is not capable of encryption, as such. As mentioned previously, it is merely an efficient and compact implementation of the classical Diffie-Hellman key-exchange algorithm. However, XTR *can* be used in anycryptosystem that relies on the discrete logarithm problem for its security. Lenstra and Verheul suggest combining XTR with the ElGamal algorithm [2] to produce a fully functional encryption scheme. I now describe how the key generation, encryption and decryption stages of this XTR cryptosystem are performed.

D. XTR Key Generation:

Let Q be the bit-length of q. To achieve security equivalent to 1024-bit

RSA, set $6Q \approx 1024$, i.e. $Q \approx 170$.

1) Find random $r \varepsilon Z$ such that $q = r^2 - r + 1$ is a Q-bit prime.

2) Find $k \varepsilon Z$ such that $p = r + kq$ and $p = 2 \pmod$

3).

3) Choose random $c \varepsilon GF(p^2) \setminus GF(p)$ and compute $cp+1$ using above Algorithm 1.

4) If $c_{p+1} \varepsilon GF(p)$ then return to step 3.

5) Compute $c_{(p2 - p + 1)} / q$ using Algorithm 1.

6) If $c_{(p2 - p + 1)} / q = 3$ then return to step 3.

7) Let $Tr(g) = c_{(p2-p+1)} / q$.

8) The XTR public key data consists of (p, q, $Tr(g)$).

E . XTR-ElGamal Encryption:

Let (p, q, $Tr(g)$) be XTR public key data either owned by Alice or shared by all parties. Moreover, let $Tr(g^k)$ be a value computed and made public by Alice for some secret integer k, selected by and known only to Alice. Bob may encrypt a message M for Alice as follows:

1) Bob selects a random b ε Z such that $1 < b < q$

– 2 and uses Algorithm 1 to compute

$S_b(Tr(g)) = (Tr(g^{b-1}), Tr(g^b), Tr(g^{b+1}))$.

2) Again using Algorithm 1, Bob computes $S_b(Tr(g^k)) = (Tr(g^{(b-1)k}), Tr(gb^k), Tr(g^{(b+1)k}))$.

3) Bob determines the symmetric encryption key K based on $Tr(g^{bk})$.

4) Bob uses an agreed symmetric encryption method with key K to encrypt M, resulting in the encryption E.

5) Bob sends ($Tr(g^b)$, E) to Alice.

F. XTR-ElGamal Decryption:

Upon receipt of ($Tr(g^b)$, E) from Bob, Alice decrypts E as follows:

1) Using Algorithm 1, Alice computes

$S_k(Tr(g^b)) = (Tr(g^{b(k-1)}, Tr(g^{bk}), Tr(g^{b(k+1)}))$.

2) Alice determines the symmetric encryption key K based on $Tr(g^{bk})$.

3) Alice uses the agreed symmetric encryption method with key K to

decrypt E, resulting in the message M.

The XTR-ElGamal cryptosystem reduces the communication and computational overhead by a factor of three at no cost to security Hence, despite the complexity of its mathematical basis, XTR does appear to have the potential for integration into low-cost and mobile devices.

CONCLUSION:

Although cryptography started for the security of the message only I ancient but now it has generalized the digital society.message transformation has become a science today. Recently biological aspect of the cryptography has launched. The natural secret also lies with this domain.

Hence the current work will be directed the future plan towards the security. This study is presented as the application of the fuzzy set and cryptography both. Zadeh's contribution was the landmark in the development of real mathematical application but today this has been applied in almost every field of the society.

REFERENCES:

1. W. Diffie, M.E. Hellman, New Direcitons in Public key cryptography, IEEE Transcations on Information Theory, Volume IT-22, (1976), 109-112.

2. T. Elgamal, A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory, 31, 1985, 469-472.

3. G.Gong, L.harn, Public key cryptosystems based on cubic finite field extensions, IEEE Transaction, On Information Theory 45(1999), 2601-2605.

4. D.E.Knuth, The art of computer programming, Volume 2, Seminumerical Algorithms, Second Edition, Addison-Wesley, 1981.

5. A.K.Lenstra, E.R.Verheul, The XTR Public key cryptosystems, Proceedings of Crypto (2000,), 1-19.

6. A.K.Lenstra, E.R.Verheul, Key Improvements to XTR proceedings of Asia crypt 2000, LNCS 1976, Springer-Verlag (2000), 220-233.

7. P.J. Smith, J.J.J.Lennon, LUC : A new public key system, In proceedings of the IFIPTC 11 Ninth International Conference on Intimation Security IFIP/Sec93 (ed. E.G.Dorgall), Narth-Holland, Amerterdarm (193), 103-117.

8. G.Gong, L.harn, Public key cryptosystems based on cubic finite field extensions, IEEE Transaction, On Information Theory 45(1999), 2601-2605.

9. D.E.Knuth, The art of computer programming, Volume 2, Seminumerical Algorithms, Second Edition, Addison-Wesley, 1981

10. R. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Communication of the ACM, 21(2), 1978, 120-126.

11. Bruce Schneier , Applied Cryptography, Second edition, John Wiley and Sons, 1996, Cryptography Book.

12. Douglas R. Stinson, Cryptography, Theory and Practice, CRC Press, 1995.