I. J. Computer Network and Information Security, 2012, 9, 63-76
Published Online August 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2012.09.08

Security Analysis and Performance Evaluation of Enhanced Threshold Proxy Signature Scheme Based on RSA for Known Signers

Raman Kumar[1], Harsh Kumar Verma[2] , Renu Dhir[3]
[1,2,3] Department of Computer Science and Engineering,
[1,2,3] Dr B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India.
[1,2,3] {er.ramankumar}@aol.in

Abstract **—** An efficient threshold signature scheme solves the difficulties of the receiver to proof the approval of the document from the sender as well as detecting if the file has been altered by illegitimate parties. In these days there are plenty of signature schemes such as (t,n) threshold proxy signature scheme. The network is a shared medium so that the weakness security attacks such as eavesdropping, replay attack and modification attack. Thus, we have to establish a common key for encrypting/decrypting our communications over an insecure network. In this scheme, a (t,n) threshold proxy signature scheme based on RSA, any t or more proxy signers can cooperatively generate a proxy signature while t-1 or fewer of them can't do it. The threshold proxy signature scheme uses the RSA cryptosystem to generate the private and the public key of the signers. Comparison is done on the basis of time complexity, space complexity and communication overhead. We compare the performance of four schemes: Hwang et al., Wen et al., Geng et al. and Fengying et al. with the performance of a scheme that has been proposed by the authors of this article earlier. In the proposed scheme, both the combiner and the secret share holder can verify the correctness of the information that they are receiving from each other. Therefore, the enhanced threshold proxy signature scheme is secure and efficient against notorious conspiracy attacks.

Index Terms **—** Unforgeability, Secret Sharing, Non repudiation, Time constraint, RSA cryptosystem for known signers

I. Introduction

Today Internet is inseparable part of our life and millions of people will be using the Internet. Reading the news, chatting with friends, purchasing a new product, researching for a paper the number of uses of the Internet is endless. One of the attractions of the Internet is that one can do almost anything from the comfort of his/her own home and with a relative sense of anonymity.

Unfortunately, the data going across the Internet may not be as secure as we would like to think. It is not especially difficult for a person with the right technical skills to intercept the data going from one computer to another. Usually this is not a problem; people don't really care if someone knows that they went to google.com and started researching Number Theory. However, if the intercepted data contains a credit card number, password, social security number, or some other private information – it becomes a whole different story.

Online banking and a host of other services rely heavily upon the security of credit card numbers, PINs, and other private information as it goes across the network. But if it is easy to intercept these numbers, how do these services work? The answer: Cryptography.

In today's commercial environment, establishing a framework for the authentication of computer based information requires a familiarity with concepts and professional skills from both the legal and computer security fields. Combining these two disciplines is not an easy task concepts from the information security field often correspond only loosely to concepts from the legal fields, even in situations where the terminology is similar. For example, from the information security point of view, "digital signature" means the result of applying to specific technical processes. The historical legal concept of "signature" is broader. It recognizes any mark made with the intention of authenticating the marked document.

In this research paper, we discuss threshold proxy signature schemes. In a (t,n) threshold proxy signature schemes, an original signer delegates a group of n proxy signers to sign message on behalf of him or her. When the proxy signature is created, t or more proxy signers cooperate to generate valid proxy signatures and less than t proxy signers can't cooperatively produce valid proxy signatures. In essence, we have tested our enhanced threshold proxy signature scheme by undergone some fruitful attacks. In section II we have reviewed the various threshold proxy signature schemes. In section III we have discussed our enhanced threshold proxy signature scheme. In section IV we have discussed security analysis of the enhanced threshold proxy signature scheme. In section V we have performance analysis of various threshold proxy signature schemes.

II. Review of Threshold Proxy Signature Schemes

A. History of threshold proxy signature schemes

In the history of proxy signature technological development, the (1,n) threshold proxy signature technique was the first to come. In (1,n) proxy signature schemes a legal proxy signature can be generated by a designated proxy signer by using a proxy signing key. The proxy signing key is computed from the original signer's private key, but the private key should not be computed from the proxy signing key in any way. In the eye of a modern user, such schemes are simple but not flexible. In order to extend proxy signature schemes to fit various practical situations, many (t,n) threshold proxy signature schemes have been proposed. For example, we have (t,n) threshold proxy signature schemes that allow any t or more proxy signers from a designated group of n members to cooperatively sign messages while t-1 or fewer members cannot generate the legal proxy signature. In practice, the original signer can flexibly choose the threshold t. The approach agrees with (1,n), (t,n) and (n,n) threshold delegations.

Shamir [8] and Blakley firstly proposed the (t,n) threshold secret sharing scheme based upon Lagrange interpolating polynomial and linear projective geometry respectively in 1979. In a (t,n) threshold secret sharing scheme, secret holder delivers the distinct secret values (calls shares or shadows) to n participants. At least t or more participants can combine their shares and reconstructs the secret, but only t-1 or fewer members cannot. Based on these properties, secret sharing is an important part of modern cryptography and has been use in many fields of modern cryptography. In 1996, Mambo et al. [7] proposed the concept of proxy signature. In their schemes, original signer can delegate his/her right to the proxy signers who can sign the message instead of the original signer.

Recently, many threshold proxy signature schemes were proposed. The history of threshold prosy signature schemes is made up in Table 1.

Table 1 – History of threshold proxy signature schemes

Sr. No.	Scheme	Method
1.	Shamir and Blakley [1979]	Lagrange interpolating polynomial and linear projective geometry
2.	Elgamal [1985]	Discrete Logarithms
3.	Denmedt and Frankel [1991]	RSA and Lagrange Coefficient
4.	Zhang [1997]	Discrete Logarithms
5.	Kim [1997]	Discrete Logarithms
6.	Sun [1999]	Discrete Logarithms
7.	Lee [2001]	Discrete Logarithms
8.	Hwang [2003]	RSA and Lagrange Coefficient
9.	Wang [2004]	RSA and Lagrange Coefficient
10.	Kuo and Chen [2005]	RSA and Lagrange Coefficient
11.	H. Jiang [2007]	RSA and Lagrange Coefficient
12.	Fanyu [2007]	RSA and Lagrange Coefficient
13.	Fengying [2007]	RSA and Lagrange Coefficient
14.	Geng [2007]	RSA and Lagrange Coefficient

The concept of threshold cryptosystems was also brought up by Denmedt and Frankel in 1991. They adapted the ElGamal public key cryptosystem and used Lagrange interpolation or geometry to produce shadows.

To make proxy signature be applicable to group oriented situations, K. Zhang [15] and Kim et al [15] proposed (t,n) threshold proxy signature in 1997, which is variant of proxy signature by using the ideas of secret sharing and threshold cryptosystems. The basic strategy used in Kim et al.'s scheme is random number generation.

B. Review of Kim et al. scheme

1) The random number generation phase

This scheme requires a protocol to generate a random number among the group without the dealer. Let P_0 be the original signer and P_1, P_2, ….. , P_n is the n proxy signers of the proxy group.

1. Each proxy signer P_i selects secret polynomial of degree t-1 such that

$$f_i(x)=r_i+a_{i,1}x+a_{i,2}x^2+ \ldots\ldots + a_{i,t-1}x^{t-1} \bmod q \quad (1)$$

where r_i, $a_{i,1}$, $a_{i,2}$ and $a_{i,t-1}$ are random numbers.

2. Then, each Pi computes $f_i(j) \bmod q$ sends it to P_j for all $1<=j<=n$ and $j \neq n$. Furthermore, P_i computes

$$g^{ri}, g^{ai,1}, g^{ai,2}, \ldots\ldots, g^{ai,t-1} \pmod p \quad (2)$$

and broadcasts them.

3. After receiving $f_j(i)$ (for $j=1,2, \ldots\ldots, n$ and $j \neq i$), P_i confirms that the validity of $f_j(i)$ by checking whether or not $g^{fj(i)}$ satisfies following equation:

$$g^{fj(i)} = g^{rj} \times ((g^{aj,1})^{i})^{1} \times \ldots\ldots \times ((g^{aj,t-1})^{i})^{t-1} \bmod p \quad (3)$$

4. If the verifications in step 3 hold, each P_i computes the secret share

$$S_i = \sum_{j=1}^{j=n} f_j(i) \quad (4)$$

and computes public outputs

$$r = \prod_{j=1}^{j=n} r_j \bmod p$$

$$g^{a1} = \prod_{j=1}^{j=n} g^{j,1} \bmod p$$

$$g^{a2} = \prod_{j=1}^{j=n} g^{j,2} \bmod p$$

.
.
.
.
.

$$g^{at-1} = \prod_{j=1}^{j=n} g^{j,t-1} \bmod p$$

2) The proxy sharing phase

1. *Group Key Generation:* First, the proxy group must execute te above protocol to obtain the share s_i and the public outputs $y_G=g^{a0} \bmod p$, $A_j = g^{aj} \bmod p$, where $j=1,2 \ldots\ldots t-1$ (5)

2. *Proxy Generation:* The original signer computes $K=g^k \bmod p$ and $e=h(m_w,K)$, where k is a random number, m_w is a warrant, and h() is one way hash function. After this, P_0 computes $\sigma = e \times x_0 + k \bmod q$, where x_0 is a private key of the original signer. (6)

3. *Proxy Sharing:* P_0 randomly chooses a polynomial such that

$$f'(x) = \sigma + b_1 x + b_2 x^2 + \ldots\ldots + b_{t-1} x^{t-1} \quad (7)$$

where b_1, b_2, b_{t-1} are random numbers. Then, P_0 computes f'(i) and sends it to each Pi in a secret manner, P_0 also computes

$$B_1 = g^{b1}, B_2 = g^{b2}, \ldots\ldots, B_{t-1} = g^{t-1} \pmod p \quad (8)$$

and publishes m_w, K, B_1, B_2, $\ldots\ldots$, $B_{t-1} \pmod p$.

4. *Proxy Share Generation:* After receiving f' (i) , each P_i has to validate f' (i) using following equation:

$$g^{f_r(i)} = (y_0)^{h(mw,K)} K \prod_{j=1}^{t-1} ((B_j)^{i})^{j} \bmod p \quad (9)$$

where y_0 is the original signer's public key. If it holds, each proxy signer P_i, computes the proxy sharing

$$\sigma'_i = f'(i) + s_i \times e \bmod q.$$

3) The proxy issuing and verification phase

1. The t or more actual signers have to execute the random number generation phase to obtain the secret output $s_i'$and public outputs

$$y= g^{c0}, C_1 = g^{c1}, C_2 = g^{c2}, \ldots\ldots, C_{t-1} = g^{ct-1} \pmod p$$

$$\text{where } s_i' = f'(i) = c_0 + c_1^{i} + c_2^{i2} + \ldots\ldots + c_{t-1} i^{t-1} \quad (10)$$

2. Then each actual signer uses his proxy signature key to issue a partial proxy signature such that $e' =h(y,m)$ and $\gamma_i = s_i' + \sigma'_i \times e \bmod q$, where m is message. Then, each actual signer reveals γ_i.

3. Everyone can verify the validity of γ_i by following equation:

$$g^{\gamma i} = (y \prod_{i=1}^{t-1} (C_i)^{j})^{i} \times (y_0)^{h(mw,K)} K \prod_{i=1}^{t-1} ((B_i)^{j})^{i} \bmod p \times$$

$$y_G \prod_{i=1}^{t-1} (((A_i)^{j})^{i\, h(mw,K)})^{h(y,m)} \bmod p) \quad (11)$$

4. If the previous verifications holds, the signature, the signature on m is (m, T, e', ,k, m_w), where $T= c_0 + \sigma \times e' = f' (0) + f (0) \times e'$ can be computed by applying the Lagrange formula. (12)

5. To verify the validity of the signature, anyone can examine the following equation:

$$y' = g^{T}x ((y_0)^{h(mw,K)} K)^{-ei)} \bmod p \text{ and } e' = h(y',m) \quad (13)$$

C. Security analysis of Kim et. al. and related schemes

The Kim et al's [15] scheme has been shown insecure by Sun et al. [6] using the public key updating attack. Kim et al. proposed two types of threshold proxy signature schemes, which were the proxy-protected scheme and the proxy-unprotected scheme. In the proxy protected scheme, the original cannot impersonate a proxy signer to issue a valid proxy signature. The proxy signing key combines the original signer's secret sharing key and a secret value among the t proxy signers. Therefore, the original signer cannot obtain the proxy signing keys. This property is called proxy-protected. One major drawback in Kim et al.'s scheme is that the actual signers cannot be identified. This can be very inconvenient for internal auditing. Kim et al.'s scheme does not satisfy the known signer's requirement, proxy protection requirement and the time constrain requirement. It does not satisfy the known signer's requirement as the actual signer cannot be identified. Also, It is necessary for a verifier to use the public information to check validity of proxy signature. If the pubic information is not authenticated, the original signer is able to execute the (t,n) threshold proxy signature scheme to generate a valid proxy signature key by himself, i.e. he plays the roles of the original signer and the proxy signers simultaneously. This is because a verifier is unable to distinguish whether the public information is created by the legal proxy group or by others (a dishonest original signer or unauthorized group). Hence, it doest not satisfy the proxy protection requirement. This scheme does not have the ability to put time constraints on the threshold delegation.

In order to remedy the problem of unknown signers, Sun et al. [6] revised Kim et al.'s [15] proxy protected type threshold proxy signature scheme and made the actual signers able to be identified. Sun et al.'s scheme is also insecure since any n-1 proxy signers in the group can conspire o obtain secret key needed by the remainder of the group. Also, the computational and communicational overhead of Sun's scheme is high. With t or more proxy signers needed to cooperatively issue a proxy signature that they have to generate and share a random number among them, and that requires several expansion modular exponential computations and communications.

Unfortunately, the Zhang's scheme [15] has also shown to be insecure by Lee, Hwang and Wang. They have shown a shown a dishonest proxy signer can cheat to get a signature which is generated by the original signer on any message with the condition that a conventional digital signature scheme is a variation of ElGamal type signature.

In 1991, Desmedt and Frankel proposed a threshold RSA signature scheme. This technique allows t out of n individuals to generate a signature for a message. The signature is on the behalf of group of n members; hence, we also call it group signature, Hwang et al. [8] extended the concepts and principles from Desmedt and Frankel's threshold RSA signature to develop a threshold RSA proxy signature scheme.

In 1999, Sun et al. [9] also suggested an enhanced proxy signature scheme based on both the Mambo-Usuda-Okamoto and Kim-Park-Won schemes. Later on, Sun, Lee and Hwang examined the security of the Sun-Hsieh scheme based on Kim-Park-Won scheme and proved that the scheme is not non-repudiable. And also, a slightly modified version was suggested by them.

Hwang et al. [1] have shown that Sun's scheme has a security weakness. An adversary can impersonate a legal proxy signer to generate a proxy signature and the real proxy signer cannot deny having signed the proxy signature.

D. Review of Hwang's et. al. Scheme

In the HLL scheme, Hwang et al. [1] proposed a practical and efficient (t,n) threshold proxy signature scheme based on the RSA cryptosystem. This scheme uses only RSA digital signature scheme and a simple Lagrange formula to share the proxy signature key.

There are three types of participants in the scheme: the original signer, the n proxy signer and combiner. The original signer allows a group of n proxy signers to sign a message. The combiner can be the secretary of the original signer. The proposed threshold proxy signature scheme can be divided into three phases:

1. The proxy sharing phase
2. The proxy issuing phase
3. The verification phase.

In the proxy generation phase, the original signer computes the partial proxy signing keys from his private key and sends them to each designated proxy signer. In the proxy signature issuing phase, the proxy signers cooperatively create a valid signature on a message M. In the verification phase, the verifier can identify not only the original signer, but also the actual signers. P_0 stands for the original signer and P_1, P_2, ….. , P_n stands for the n proxy signers. N_i is a public RSA modulus for P_i such the Ni is a public RSA modulus for P_i such that $N_i = p_i \times q_i$, where p_i and q_i are two secret large primes. Where d_i is the private key for P_i and its corresponding public key is e_i, such that $d_i \times e_i = 1 \bmod \phi(N_i)$, where $\phi(N_i) = (p_i - 1) \times (q_i - 1)$. The parameters e_i and N_i can be published. The parameters d_i and $\phi(N_i)$ are kept secret by the holder. $[M]^{di} \bmod N_i$ represents M signed with P_i's private key d_i, and $[M]^{ei} \bmod N_i$ represents M encrypted with P_i's public key e_i using the ordinary RSA cryptosystem. The message m_w stands for a warrant that is minted by the original signer and it contains important information such as the validity period of the proxy key, the identities of the proxy signers, and the original signer, etc. In the proposed scheme, let $N_0 < N_i$ (i = 1, 2, …, n).

1 The proxy sharing phase

Assume that an original signer P_0 delegates the power to sign messages to n members during s stipulated period. The steps to generate the proxy key are as follows:

1.1 Proxy generation

P_0 produces the group proxy signing key D and proxy verification key E, where

$$D = d_0^{mw} \bmod \phi(N_0) \quad (14)$$

$$E = e_0^{mw} \bmod \phi(N_0) \quad (15)$$

where, $m_w = (P + T + r) \bmod \phi(N_0)$

P is the validity period of proxy signatures, T is the sum of identities of P_0, $P_1, \ldots, P_n$ and r is a random number.

Then P_0 publishes $\{m_w, E, [m_w, E]^{d0} \bmod N_0\}$.

1.2 Proxy sharing

P_0 selects a t-1 degree polynomial,

$$f(x) = D + a_1 x + \ldots + a_{t-1} x^{t-1} \bmod \phi(N_0) \quad (16)$$

where $a_1, a_2, \ldots a_{t-1}$ are random numbers. Meanwhile, P_0 calculates proxy signer P_i's partial proxy signing key $k_i = f(i)$ and sends $[[k_i]^{d0} \bmod N_0, k_i]^{ei} \bmod N_i$ to the proxy signer P_i.

2 Proxy share generation

When proxy signer P_i receives $[[k_i]^{d0} \bmod N_0, k_i]^{ei} \bmod N_i$, he or she can get $\{[k_i]^{d0} \bmod N_0, k_i\}$ by his or her secret key d_i. And then P_i confirms the validity of k_i and keeps it secret.

2.1 The proxy signature issuing phase

2.2 Let T denote the group members including any t or more proxy signers who want to generate a proxy signature on message M on behalf of P_0 cooperatively. Each proxy signer P_i uses the partial proxy signing key k_i to generate the partial signature

$$s_i = M^{ki} \bmod N_0 \quad (17)$$

Then P_i sends $\{[s_i,i]^{di} \bmod N_i, s_i\}$ to the combiner.

2.3 Upon the combiner receives all partial signature s_i from P_i, firstly, he or she verifies the validity of the partial proxy signature by checking if $[s_i,i]^{di.ei} \bmod N_i = (s_i,i)$ or not. If all partial signatures are valid, the combiner computes the value of

$$v = \prod_{\substack{ID_a, ID_b \in T \\ a>b}} (ID_a - ID_b)$$

$$vLi = \prod_{\substack{ID_a, ID_b \in T \\ a>b}} (ID_a - ID_b) \prod_{j=1, j\neq i}^{t} (-ID_j / (ID_i - ID_j)) \quad (18)$$

Here, $\prod_{j=1, j\neq i}^{t} (ID_i - ID_j)$ is a factor of $\prod_{\substack{ID_a, ID_b \in T \\ a>b}} (ID_a - ID_b)$

So vLi is an integer and the combiner needn't compute the inverse of

$$\prod_{j=1, j\neq i}^{t} (ID_i - ID_j)$$

Finally, the combiner generates the signature S as follows:

$$S = \prod_{i \in T} s_i^{vLi} \bmod N_0 \quad (19)$$

The result of proxy signature is $\{v, S\}$.

3 The proxy signature verification phase

3.1 The verifier can verify the signature signed on behalf of the original signer by following equation:

$$S^E = M^v \bmod N_0 \quad (20)$$

3.2 The original signer can differentiate the actual signer from the signature $s_i^{diei} \bmod N_i = s_i$. Then the original signer can trace the actual signers by e_i.

E. Conclusions from the threshold proxy signature schemes

All analysis indicated that the scheme fails to satisfy all the requirements except the one or two. So, an enhanced threshold proxy signature scheme must satisfy all of the following basic requirements which can be called as proxy requirements [1], [2], [3] and [4]:

1. Secrecy: - The original signer's private key is very important. It must be kept secret. If it is discovered, the security of the system is ruined. Therefore, the system must ensure that the private key never gets derived from any information such as the sharing of the proxy signing key or the original signer's public key. Furthermore, no proxy signers should be able to cooperatively derive the original signer's private key.

2. Proxy Protected: - Only a delegated proxy signer can generate his partial proxy signature. Even the original signer cannot masquerade as a proxy signer to generate a partial proxy signature. This property protects the authority of the proxy signer.

3. Unforgeability: - A valid proxy signature can only be cooperatively generated by t or more proxy signers. Nondelegated signers have no capability to generate a valid proxy signature. Also, (t-1) or less proxy signers have no capability of forging a valid proxy signature.

4. Nonrepudiation: - Any valid proxy signature must be generated by t or more proxy signers. The verifier can make sure that the signed message is a correct one by using the proxy signing keys. The original signer cannot deny having delegated the power of signing messages to the proxy signers. Furthermore, the proxy signers cannot deny that they have signed the message.

5. Time Constraint: - The proxy signing keys can be used only during a stipulated period. Once expired, proxy signing keys become invalid; as a result, the signing capability of the proxy signers disappears. However, the original signer's private key can be repeatedly used. This is more suitable for use in the real world.

6. Known Signers: - For internal auditing purposes, the system is able to identify the actual signers in the original signer's private key. The proxy signer has the capability to sign on behalf of the original signer, but from the

proxy signing key the proxy signer cannot recover the original signer's private key.

III. Our Scheme

The concept of threshold cryptosystems was first proposed up by Desmedt and Frankel [12]. They adapted the ElGamal [13] public key cryptosystem and used Lagrange interpolation or geometry to produce the shadows. In the history of proxy signature technological development, the (1,n) threshold proxy signature technique was the first to come [1]. In (1,n) proxy signature schemes [7], [9], [10], a legal proxy signature can be generated by a designated proxy signer by using a proxy signing key. However, in a (t,n) threshold proxy signature scheme, the original signer delegates the power of signing messages to a designated proxy group of n members. Any t or more proxy signers of the group can cooperatively issue a proxy signature on behalf of the original signer, but (t-1) or fewer proxy signers cannot. Previously, all of the proposed threshold proxy signature schemes, for instance Lee et al. [5], Sun et al. [14], Zhang et al. [15] and Mambo et al. [7], have been based on the discrete logarithm problem. However, the recently proposed threshold proxy signature schemes are based on RSA cryptosystem [8] and Lagrange coefficient. In 2003, Hwang et al. [1] proposed a practical and efficient (t,n) threshold proxy signature scheme based on the RSA cryptosystem. This scheme uses only RSA digit signature scheme and a simple Lagrange formula to share the proxy signature key. In 2004, Wang et al. [11] pointed out a problem on the correctness of the HLL scheme. In 2005, Wen et al. [2] also indicated two security weaknesses in HLL scheme and proposed a new scheme to overcome these weaknesses.

We compare the performance of four schemes: Hwang et al. [1], Wen et al.[2], Geng et al.[3] and Fengying et al[4] with the performance of a scheme that has been proposed by the authors of this article earlier and proposed an enhanced secure threshold proxy signature scheme. In the proposed scheme, both the combiner and the secret share holder can verify the correctness of the information that they are receiving from each other. Therefore, the enhanced threshold proxy signature scheme is secure and efficient against notorious conspiracy attacks. Table 2 gives the comparison of threshold proxy signature schemes based on proxy requirements each scheme.

Table 2 - A Comparison of Threshold Proxy Signature Schemes Based on Proxy Requirements.

Sr. No.	Proxy Signature Scheme/ Requirements	Kim et. al.	Sun et. al.	HLL et. al.	Wen et.al.	Enhanced Scheme
1.	Secrecy	Yes	Yes	No	No	Yes
2.	Proxy Protection	No	No	No	No	Yes
3.	Unforgeability	Yes	No	No	No	Yes
4.	Non-repudiation	Yes	Yes	No	No	Yes
5.	Time-Constraint	No	No	Yes	Yes	Yes
6.	Known Signers	No	Yes	No	No	Yes

IV. Security Analysis of the Proposed Scheme

1. *Factorization of RSA Module*

Factoring n: The fastest known factoring algorithm developed by Pollard is the General Number Field Sieve [8], which has running time for factoring a large number of size n, of order

$$O\left(\exp\left((\frac{64}{9}\log n)^{\frac{1}{3}}(\log\log n)^{\frac{2}{3}}\right)\right)$$

The method relies upon the observation that if integers x and y are such that $x \neq y \pmod n$ and $x^2 = y^2 \pmod n$ then gcd (x − y, n) and gcd(x+y, n) are non-trivial factors of n.

The following Table 3(a), 3(b) gives the number of operations needed to factor n with GNFS method, and the time required if each operation uses one microsecond, for various lengths of the number n (in decimal digits).

Table 3(a) The number of operations needed to factor n with GNFS method

Digits	Number of operations	Time
100	9.6×10^{8}	16 minutes
200	3.3×10^{12}	38 days
300	1.3×10^{15}	41 years
400	1.7×10^{17}	5313 years
500	1.1×10^{19}	3.5×10^{5} years
1024	1.3×10^{26}	4.2×10^{12} years
2048	1.5×10^{35}	4.9×10^{21} years

Computing $\phi(n)$ without Factoring "n":
Assume that $n=p\times q$, $p<q$
Since $(q+p)^2-(q-p)^2=4pq=4n$,
then $(q+p)^2=4n+(q-p)^2$ so $q+p=\sqrt{4n+(q-p)^2}$;
guess q-p and then find q+p, so $\varphi(n)=n-(p+q)+1$.
Example:
Suppose $n=221\ (4n=884)$

Table 3(b) Computing $\phi(n)$ without Factoring "n":

$q-p$	$(q+p)^2=4n+(q-p)^2$	$q+p=\sqrt{4n+(q-p)^2}$
1	885	29.7489 · · ·
2	888	29.7993 · · ·
3	893	29.8831 · · ·
4	900	30

So, $q-p=4$ and $q+p=30$ then $\varphi(n)=221-30+1=192$
and $p=13, q=17, n=13\times17$

2. Lattices and Lattice reduction of RSA Module

Lattice Based Attacks on RSA

The following attacks has been tested for RSA modules:

- Hastad's Attack
- Franklin-Reiter Attack
- Extension to Wiener's Attack

Lattices and Lattice reduction

Given a set of m linearly independent vectors, $\{b_1,\ldots,b_m\}$in R^n. The set of all real linear combinations of these vectors $V=\left\{\sum_{i=1}^{m}a_ib_i : a_i\in R\right\}$ is a vector subspace.

Gram-Schmidt process[12]: takes one basis $\{b_1,\ldots,b_m\}$ and produces a basis $\{b_1^*,\ldots,b_m^*\}$ which is pairwise orthogonal.

$b_1^*=b_1$

$$\mu_{i,j}=\frac{\langle b_i,b_j^*\rangle}{\langle b_j^*,b_j^*\rangle},\ \text{for } 1\le j<i\le n$$

$$b_i^*=b_i-\sum_{j=1}^{i-1}\mu_{i,j}b_j^*$$

Example:

$$b_1=\begin{pmatrix}2\\0\end{pmatrix}\text{ and }b_2=\begin{pmatrix}1\\1\end{pmatrix}$$

$$b_1^*=b_1=\begin{pmatrix}2\\0\end{pmatrix}$$

$$\mu_{2,1}=\frac{\langle b_2,b_1^*\rangle}{\langle b_1^*,b_1^*\rangle}=\frac{1}{2}$$

$$b_2^*=b_2-\mu_{2,1}=\begin{pmatrix}0\\1\end{pmatrix}$$

Given a set of basis vectors $\{b_1,\ldots,b_m\}$ in R^n, and m<=n.A lattice $L=\left\{\sum_{i=1}^{m}a_ib_i : a_i\in Z\right\}$ is a set of all integer linear combinations of the b_i.

Definition 1:

A basis $\{b_1,\ldots,b_m\}$ is called LLL reduced if the associated Gram-Schmidt basis $\{b_1^*,\ldots,b_m^*\}$ satisfies

$$|\mu_{i,j}|\le\frac{1}{2}\ \text{ for }\ 1\le j<i\le m$$

$$\|b_i^*\|^2\ge\left(\frac{3}{4}-\mu_{i,i-1}^2\right)\|b_{i-1}^*\|^2\ \text{ for }\ 1<i\le m$$

For all non-zero , $\|b_1\|\le 2^{(m-1)/2}\|x\|$ we have

$$\|b_1\|\le 2^{m/4}\Delta^{1/m},\quad \Delta=\left|\det(B^TB)\right|^{1/2}$$

3. Security Levels of RSA Module on different platforms

The following are the creation of key in seconds for different security levels which can be used for encryption and decryption:

Figure 4(a) – Security Levels of RSA Module on 90MHz Pentium Platform

Security Level	Encrypt (blks/sec)	Decrypt (blks/sec)	Create Key (sec)
512 bit	370	42	0.45
768 bit	189	15	1.5
1024 bit	116	7	3.8

Figure 4(b) – Security Levels of RSA Module on 255 MHz Digital AlphaStation

Security Level	Encrypt (blks/sec)	Decrypt (blks/sec)	Create Key (sec)
512 bit	1020	125	0.26
768 bit	588	42	0.59
1024 bit	385	23	1.28

The fields in Table 4(a) and 4(b) have been generated by varying the values of security levels for both the Pentium and AlphaStation respectively. It shows the various parameters generated for different security levels.

4. A general coalition attack against threshold signature schemes

Though our modification can withstand the forgery attack suffered by the said [1], [2], [3] and [4] threshold group signature scheme, there is a general coalition attack against threshold signature schemes. In the ordinary threshold signature scheme, the group's secret key is $f(0)$, and each member U_i has the secret share $f(x_i)$. If t or more malicious members pool their secret shares together, they can recover f(0) by applying Lagrange interpolating polynomial. Then each one of them can alone compute valid signatures for new messages on

behalf of the group afterwards without the cooperation of other signers and without being detected by verifiers. Obviously, this violates the group's signing policy. Otherwise, if such coalition is permissive, other signers would follow this kind of dishonesty. Thus, each user can also alone compute valid group signatures after one coalition. It's terrible for threshold signature schemes. This coalition attack is inherent in many threshold signature schemes using threshold secret share scheme, as long as the secret key can be recovered from secret shares.

5. The probability of catching a user

The probability of catching a user in enhanced threshold proxy signature scheme depends on the number of identity pairs used in it. The more pairs used, the greater the chance of catching the anonymous user. The probability of catching the anonymous user is:

$$1-½^{n}$$

Where n is the number of pairs used.

Example, if n=5 then the chance of catching a user is 0.97.

V. Performance Analysis of the Proposed Scheme

The analysis reports of the proposed hypothesis for enhanced threshold proxy signature scheme are given below:

A. *Entropy*

In this case, the value of entropy is the measure of the tendency of a process, to be entropic ally *favored*, or to proceed in a particular direction. Moreover, entropy provides an indication for a specific encryption method. We have analyzed our hypothesis on the basis of entropy generated.

The Fig. 1 shows that Entropy for enhanced threshold proxy signature scheme. The Fig. 2 shows that compression ratio required in each scheme. Table 5 lists the name and compression ratio required in each scheme.

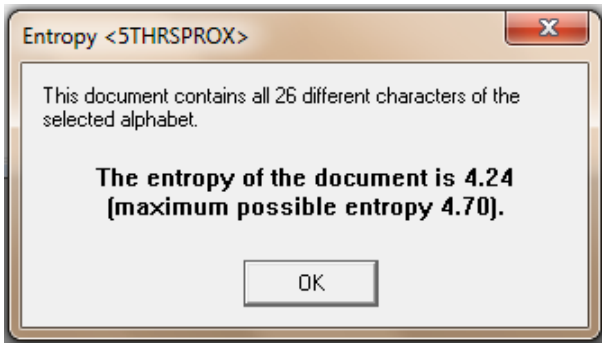

Figure 1 –Entropy for enhanced threshold proxy signature scheme

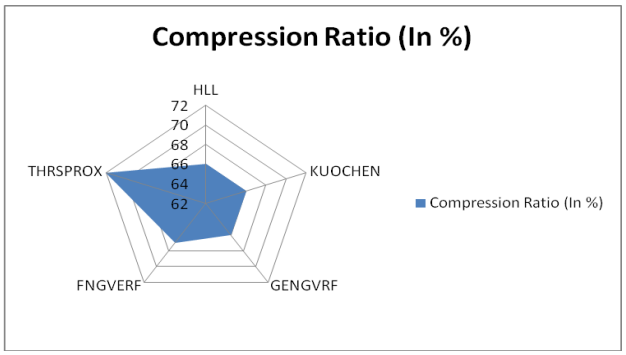

Figure 2 –Radar Chart showing compression ratio required in each schemes

Table 5 – Compression Ratio (In %) for threshold proxy signature schemes

Threshold Proxy Signature Scheme	Compression Ratio (In %)
HLL	66
KUOCHEN	66
GENGVRF	66
FNGVERF	67
THRSPROX	72

B. *Floating Frequencies/Intuitive Synthesis*

Floating Frequencies/Intuitive Synthesis in its completed three part entirely which takes full advantage of the time complexity, space complexity and communication overhead provided by the digital medium. We have calculated floating frequency of threshold proxy signature scheme. The Fig. 3 shows that Floating Frequencies/Intuitive Synthesis for enhanced threshold proxy signature scheme.

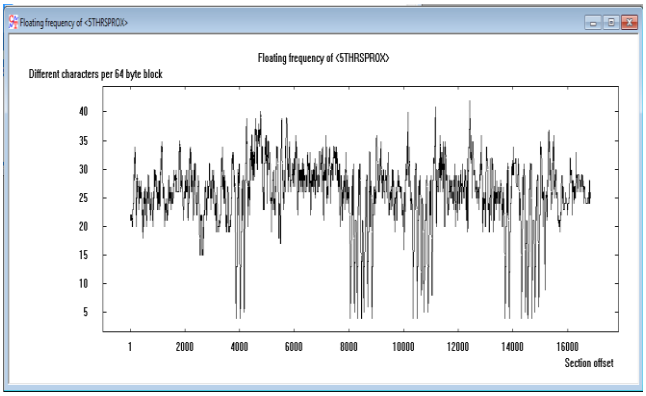

Figure 3 – Floating Frequencies/Intuitive Synthesis for enhanced threshold proxy signature scheme

C. *ASCII Histogram*

The ASCII Histogram proved to be very useful since it helped enormously in debugging code involving probability calculations with simple print statements. Probabilistic simulations are extremely hard to test because the results of a given operation are never strictly the same. However, they should have the same probability distribution, so by looking at the rough shape of the histogram, you tell you if your calculations are

going in the right direction. In this context, we have calculated ASCII histogram for our threshold proxy signature scheme. The Fig. 4 shows that ASCII Histogram for enhanced threshold proxy signature.

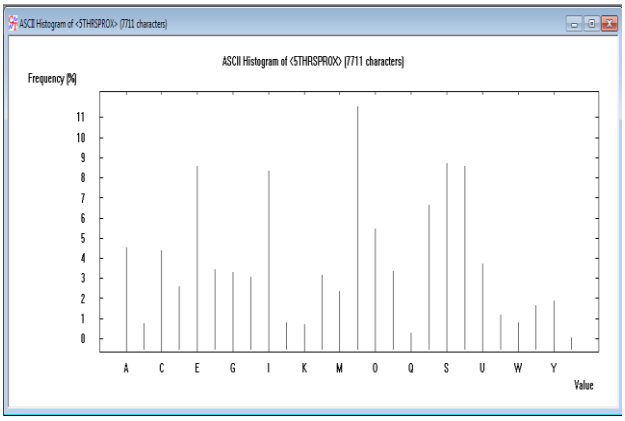

Figure 4 – *ASCII Histogram* for enhanced threshold proxy signature scheme

D. *Autocorrelation*

A mathematical representation of the degree of similarity between a given time series and a lagged version of itself over successive time intervals, called correlation. It is the same as calculating the correlation between two different time series, except that the same time series is used twice - once in its original form and once lagged one or more time periods. The term can also be referred to as "lagged correlation" or "serial correlation". In this, we have calculated autocorrelation for threshold proxy signature scheme. The Fig. 5 shows that Autocorrelation for enhanced threshold proxy signature scheme

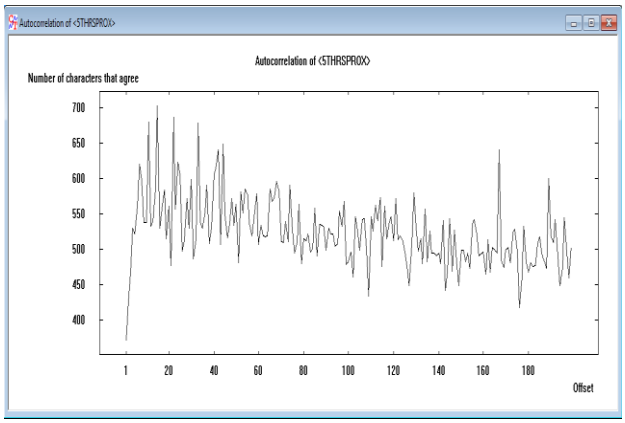

Figure 5 – Autocorrelation for enhanced threshold proxy signature scheme

E. *Histogram Analysis*

A histogram is a graphical representation showing a visual impression of the distribution of data. We have analyzed histogram of for all threshold proxy signature schemes.

Detailed View

The detailed view of the histogram analysis of all schemes can be represented as follows:

Experiment 1:

Histogram Analysis of <1HLL>. File size 12581 bytes. Descending sorted on frequency.

Table 6 – Histogram Analysis for HLL threshold proxy signature scheme

No.	Substring	Frequency (in %)	Frequency
1	N	11.0343	654
2	I	9.1277	541
3	T	8.824	523
4	E	8.6216	511
5	S	7.4405	441
6	R	7.1368	423
7	A	5.0785	301
8	O	4.6567	276
9	C	4.1842	248
10	D	3.6612	217
11	U	3.5937	213
12	F	3.2732	194
13	P	3.1213	185
14	G	3.0707	182
15	L	3.0201	179
16	H	2.8682	170
17	Y	2.6489	157
18	M	2.4296	144
19	X	1.4341	85
20	V	1.0798	64
21	W	0.9954	59
22	J	0.8267	49
23	B	0.7761	46
24	K	0.6917	41
25	Q	0.3374	20
26	Z	0.0675	4

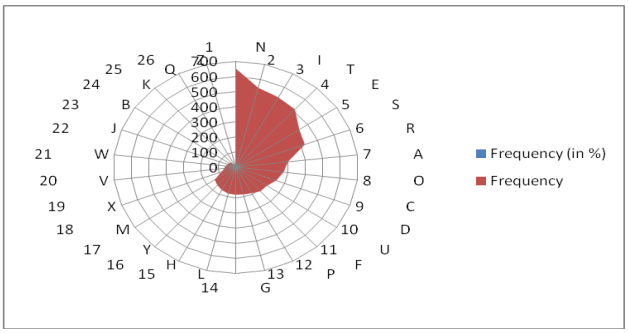

Figure 6 – Radar Chart showing Histogram Analysis for HLL threshold proxy signature scheme

The Fig. 6 shows that Radar Chart showing Histogram Analysis for HLL threshold proxy signature scheme.

Table 6 lists the Histogram Analysis for HLL threshold proxy signature scheme.

Experiment 2:

Histogram Analysis of <2KUOCHEN>. File size 11733 bytes. Descending sorted on frequency.

Table 7 – Histogram Analysis for KUOCHEN threshold proxy signature schemes

No.	Substring	Frequency (in %)	Frequency
1	N	11.3387	631
2	I	8.841	492
3	E	8.6253	480
4	T	8.4636	471
5	S	7.8886	439
6	R	7.044	392
7	O	4.8697	271
8	A	4.6361	258
9	C	4.4205	246
10	U	3.8455	214
11	G	3.2884	183
12	P	3.2165	179
13	L	3.1626	176
14	F	3.0189	168
15	H	2.9111	162
16	D	2.8392	158
17	M	2.6954	150
18	Y	1.8509	103
19	X	1.4196	79
20	W	1.2579	70
21	J	1.15	64
22	V	1.0422	58
23	B	0.9164	51
24	K	0.7907	44
25	Q	0.3953	22
26	Z	0.0719	4

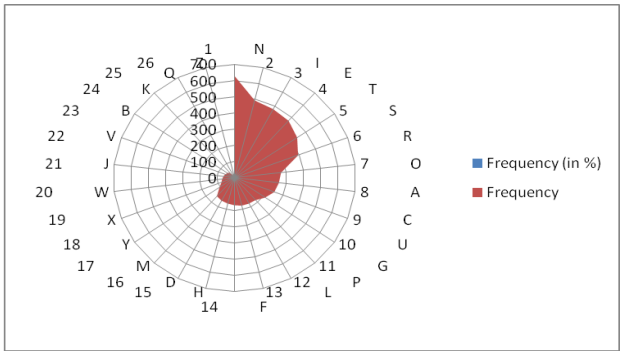

Figure 7 – Radar Chart showing Histogram Analysis for KUOCHEN threshold proxy signature scheme

The Fig. 7 shows that Radar Chart showing Histogram Analysis for KUOCHEN threshold proxy signature scheme. Table 7 lists the Histogram Analysis for KUOCHEN threshold proxy signature scheme.

Experiment 3:

Histogram Analysis of <3GENGVRF>. File size 11259 bytes. Descending sorted on frequency.

Table 8 – Histogram Analysis for GENGVRF threshold proxy signature schemes

No.	Substring	Frequency (in %)	Frequency
1	N	10.9658	587
2	I	9.4153	504
3	T	9.079	486
4	S	8.3878	449
5	E	7.9208	424
6	R	7.1175	381
7	O	4.7076	252
8	A	4.5769	245
9	C	3.9978	214
10	U	3.6802	197
11	F	3.5681	191
12	P	3.5494	190
13	G	3.4747	186
14	L	2.989	160
15	D	2.8956	155
16	H	2.7835	149
17	M	2.3538	126
18	Y	1.8121	97
19	X	1.4945	80
20	V	1.3824	74
21	J	0.9714	52
22	B	0.8967	48
23	W	0.7659	41
24	K	0.7286	39
25	Q	0.411	22
26	Z	0.0747	4

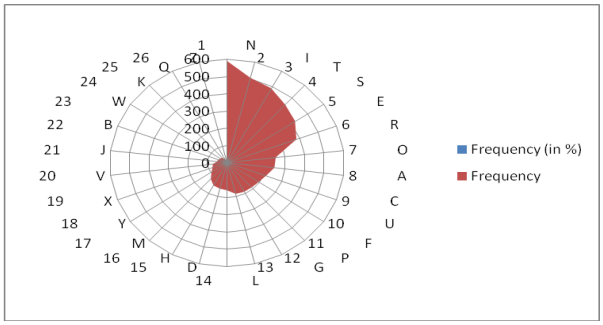

Figure 8 – Radar Chart showing Histogram Analysis for GENGVRF threshold proxy signature scheme

The Fig. 8 shows that Radar Chart showing Histogram Analysis for GENGVRF threshold proxy signature scheme. Table 8 lists the Histogram Analysis for GENGVRF threshold proxy signature scheme.

Experiment 4:

Histogram Analysis of <4FNGVERF>. File size 12067 bytes. Descending sorted on frequency

Table 9 – Histogram Analysis for FNGVERF threshold proxy signature schemes

No.	Substring	Frequency (in %)	Frequency
1	N	10.947	630
2	I	9.1573	527
3	T	8.7576	504
4	S	8.3927	483
5	E	8.2711	476
6	R	6.9505	400
7	O	4.7089	271
8	A	4.6568	268
9	C	4.066	234
10	U	3.8401	221
11	F	3.5274	203
12	G	3.5274	203
13	P	3.4231	197
14	L	3.3189	191
15	D	2.7454	158
16	H	2.6933	155
17	M	2.6759	154
18	Y	1.7724	102
19	X	1.4248	82
20	V	1.1816	68
21	W	1.0252	59
22	B	0.8862	51
23	J	0.8688	50
24	K	0.7298	42
25	Q	0.3823	22
26	Z	0.0695	4

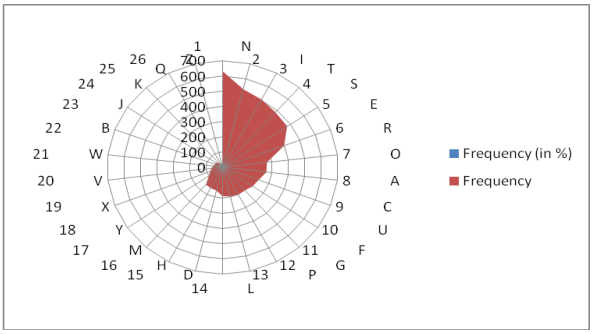

Figure 9 – Radar Chart showing Histogram Analysis for FNGVRF threshold proxy signature scheme

The Fig. 9 shows that Radar Chart showing Histogram Analysis for FNGVRF threshold proxy signature scheme. Table 9 lists the Histogram Analysis for FNGVRF threshold proxy signature scheme.

Experiment 5:

Histogram Analysis of <5THRSPROX>. File size 16897 bytes. Descending sorted on frequency.

Table 10 – Histogram Analysis for enhanced threshold proxy signature schemes

No.	Substring	Frequency (in %)	Frequency
1	N	11.5549	891
2	S	8.7278	673
3	E	8.5981	663
4	T	8.5722	661
5	I	8.3258	642
6	R	6.6399	512
7	O	5.4597	421
8	A	4.539	350
9	C	4.3963	339
10	U	3.696	285
11	F	3.4626	267
12	P	3.3329	257
13	G	3.307	255
14	L	3.1384	242
15	H	3.0865	238
16	D	2.5937	200
17	M	2.3603	182
18	Y	1.8934	146
19	X	1.634	126
20	V	1.2061	93
21	J	0.83	64
22	W	0.817	63
23	B	0.7522	58
24	K	0.7133	55
25	Q	0.3112	24
26	Z	0.0519	4

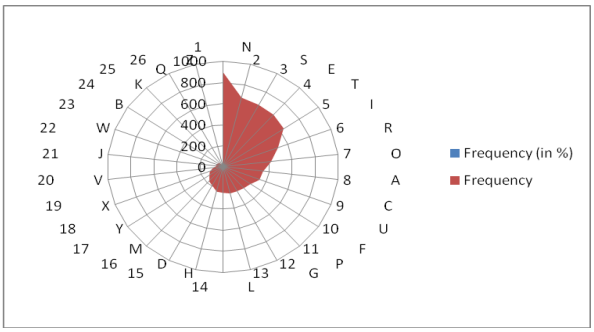

Figure 10 – Radar Chart showing Histogram Analysis for enhanced threshold proxy signature scheme

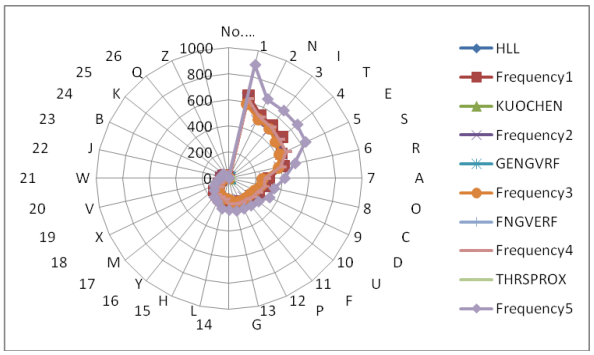

Figure 11 – Radar Chart showing Overall Analysis for all threshold proxy signature schemes

The Fig. 10 shows that Radar Chart showing Histogram Analysis for enhanced threshold proxy signature scheme. Table 10 lists the Histogram Analysis for enhanced threshold proxy signature scheme.

The Fig. 11 shows that Radar Chart showing Overall Histogram Analysis for all threshold proxy signature schemes. Annexure - I lists the Histogram Analysis for overall threshold proxy signature schemes.

VI. Conclusion

As the proxy signature is the solution to the delegation of signing capabilities in any electronic environment. In this paper, various threshold proxy signature schemes have been compared based on whether they fulfill the proxy signature requirements or not and proposed an enhanced secure threshold proxy signature scheme. Some of these schemes are based on RSA cryptosystem for known signers, as RSA cryptosystem is a popular security technique. In this, we also propose a new scheme which includes the features and benefits of the two schemes: Fengying et al. and Geng et al. The implementation of the encryption and decryption function justify the real life applicability of the propose scheme. In this, we have analyzed our enhanced threshold proxy signature scheme for various parameters**.** In essence, the results shows that the enhanced threshold proxy signature scheme is an efficient one and it can provide great capabilities for various applications. Future work may extend these studies to analyze the impact of other parameters on enhanced secure threshold proxy signature scheme to optimize these parameters to make scheme better, secure, efficient and more adaptable in commercial applications.

Acknowledgment

The authors also wish to thank many anonymous referees for their suggestions to improve this paper.

References

[1] Min-Shiang Hwang, Member, IEEE, Eric Jui-Lin Lu, and Iuon-Chang Lin (2003). A Practical (t,n) Threshold Proxy Signature Scheme Based on the RSA Cryptosystem, IEEE Transactions on knowledge and data Engineering, 15(6), 1552-1560.

[2] Wen-Chung Kuo, Ming-Yang Chen (2005). A Modified (t, n) Threshold Proxy Signature Scheme Based on the RSA Cryptosystem, In Proceedings of the Third International Conference on Information Technology and Applications, ICITA'05, 576-579.

[3] Geng Yong-Jun, Tian Hui, Hong Fan (2007). A Modified and Practical Threshold Proxy Signature Scheme Based on RSA, In Proceedings of the ICACT,(ICACT '07, 1958-1960.

[4] Fengying Li, Qingshui Xue and Zhenfu Cao (2007). Crypanalysis of Kuo and Chen's Threshold Proxy Signature Scheme Based on the RSA, In the proceedings of International Conference on Information Technology, ITNG'07, 815-818.

[5] Lee N. Y., Hwang T., and Wang C. H., Zhang O. (1998). Nonrepudiable Proxy Signature Schemes, Proceedings of Australasian Conference on Information Security and Privacy, ACISP '98, 415-422.

[6] Sun H.-M., Lee N.-Y.and Hwang T.(1999). Threshold Proxy Signatures, IEEE Proceedings of Computers and Digital Techniques, 146(5), 259-263.

[7] Mambo M., Usuda K. and Okamoto E. (1996). Proxy Signature Delegation of the Power to Sign Message, IEICE Transactions on Fundamentals, E-79A(9), 1338-1353.

[8] Rivest R.L., Shamir A., and Adleman L. M. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems Communications, ACM, 21(2), 120-126.

[9] Okamoto T., Mitsuru T., Okamoto E (1999). Extended Proxy Signature for Smart Cards, LNCS, Springer, 247-258.

[10] Mambo M., Usuda K., and Okamoto E. (1996). Proxy Signatures for Delegating Signing Operation, Proceeding of Third ACM Conference of Computer and Communications Security, 48-57.

[11] Guilin Wang, Feng Bao, Jianying Zhou, and Robert H. Deng. (2004). Comments on A Practical (t,n) Threshold Proxy Signature Scheme Based on the RSA Cryptosystem, IEEE Transactions on Knowledge and Data Engineering, 16(10), 1309-1311.

[12] Stefan Katzenbeisser (2001). Recent Advances In RSA Cryptography, Springer, 85-90.

[13] Desmedt Y. and Frankel Y. (1989). Threshold Cryptosystems, In the Proceedings of Advances in Cryptology, Crypto '89, 307-315.

[14] ElGamal T.(1985). A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Transactions of Information Theory, .31(4), 469-472.

[15] Sun H.-M. (1999). An Efficient Nonrepudiable Threshold Proxy Signature Scheme with Known Signers, Computer Communications, 22(8), 717-722.

[16] Kim S., Park S. and Won D.(1997). Proxy signatures, revisited, In the prceedings of ICICS'97, LNCS, 1334, 223-232.
[17] Zhang K. (1997). Threshold Proxy Signature Schemes, Proceedings of Information Security Workshop, 191-197.
[18] Lee N. Y., Hwang T. and Wang C. H. (1998). On Zang's nonrepudiable proxy signature schemes, In the prceedings of ACISP'98, LNCS, 415-422.
[19] Denning D. E. R. (1982). Cryptography and Data Seurity, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©1982, 115-265.
[20] Hsu C. L., Wu T. S, and Wu T. C. (2001). New Nonrepudiable Threshold Proxy Signature Scheme with Known Signers, The Journal of Systems and Software, 58(5), 119-124.
[21] Agrawal M., Kayal N., Saxena N. (2004). PRIMES in P, Ann. Math,*160, 781-793.*
[22] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein (2001). Section 31.8: *Primality testing, Introduction to Algorithms (Second Edition ed.),* MIT Press, McGraw-Hill, 889–890. ISBN 0-262-03293-7.
[23] Chang-Tsun Li (2008). *Multimedia Foresics and Security (First ed.),* IGI Global, 73-74. ISBN 978-1-59904-869-7.
[24] Friedman, Milton (December 1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association (American Statistical Association) 32 (200): 675–701.
[25] Harsh Kumar Verma, Kamalpreet Kaur and Raman Kumar, (2008). Comparison of Threshold Proxy Signature Schemes, International Conference on Security and Management, SAM'08, USA, 227-231.
[26] Raman Kumar and Harsh Kumar Verma (2010). An Advanced Secure (t,n) Threshold Proxy Signature Schemes Based on RSA Cryptosystem for Known Signers, IEEE 2nd International Advance Computing Conference, IACC'10, INDIA, 293-298.
[27] Raman Kumar and Harsh Kumar Verma (2010). Secure Threshold Proxy Signature Scheme Based on RSA for Known Signers, Journal of Information Assurance and Security, USA, 5(4), 319-326.

Author(s) Biographies

Mr. Raman Kumar *(er.ramankumar@aol.in)* joined National Institute of Technology, Jalandhar *(Deemed University)*, where he is working as a Research Scholar with the Department of Computer Science and Engineering.

Before joining National Institute of Technology, Jalandhar, He did his Bachelor of Technology *with honours* in Computer Science and Engineering from Guru Nanak Dev University; Amritsar *(A 5 Star NAAC University).* He did his Master of Technology *with honours* Computer Science and Engineering from Guru Nanak Dev University; Amritsar *(A 5 Star NAAC University)*. His major area of research is Cryptography, Security Engineering and Information security. He has published many papers in refereed journals and conference proceedings on his research areas.

Dr. Harsh Kumar Verma is working as an Associate Professor and Head in the Department of Computer Science and Engineering at National Institute of Technology, Jalandhar *(Deemed University)* since 1996.

His major areas of research are Scientific Computing, Information security and Software Systems. He has a number of publications in National as well as International Journals. He is a life member of Computer Society of India, ISTE. He is also a member of IEEE Computer Section.

Dr. Renu Dhir joined National Institute of Technology, Jalandhar *(Deemed University)* on 9th April 1997, where she is working as an Associate Professor with the Department of Computer Science and Engineering.

Before joining National Institute of Technology, Jalandhar, She did her B.Sc Electrical Engineering from Panjab University, Chandigarh during year 1983. She did her Master of Technology from TIET, Patiala during year 1997. She did her Ph. D from Punjabi University, Patiala in the year 2008. She has various publications in National as well as International Conferences and Journals on her research areas.

Annexure - I

*	Schemes	HLL	Frequency1	KUOCHEN	Frequency2	GENGVRF	Frequency3	FNGVERF	Frequency4	THRSPROX	Frequency5
No.	Substring	Frequency (in %)	Frequency	Frequency (in %)	Frequency	Frequency (in %)	Frequency	Frequency (in %)	Frequency	Frequency (in %)	Frequency
1	N	11.0343	654	11.3387	631	10.9658	587	10.947	630	11.5549	891
2	I	9.1277	541	8.841	492	9.4153	504	9.1573	527	8.7278	673
3	T	8.824	523	8.6253	480	9.079	486	8.7576	504	8.5981	663
4	E	8.6216	511	8.4636	471	8.3878	449	8.3927	483	8.5722	661
5	S	7.4405	441	7.8886	439	7.9208	424	8.2711	476	8.3258	642
6	R	7.1368	423	7.044	392	7.1175	381	6.9505	400	6.6399	512
7	A	5.0785	301	4.8697	271	4.7076	252	4.7089	271	5.4597	421
8	O	4.6567	276	4.6361	258	4.5769	245	4.6568	268	4.539	350
9	C	4.1842	248	4.4205	246	3.9978	214	4.066	234	4.3963	339
10	D	3.6612	217	3.8455	214	3.6802	197	3.8401	221	3.696	285
11	U	3.5937	213	3.2884	183	3.5681	191	3.5274	203	3.4626	267
12	F	3.2732	194	3.2165	179	3.5494	190	3.5274	203	3.3329	257
13	P	3.1213	185	3.1626	176	3.4747	186	3.4231	197	3.307	255
14	G	3.0707	182	3.0189	168	2.989	160	3.3189	191	3.1384	242
15	L	3.0201	179	2.9111	162	2.8956	155	2.7454	158	3.0865	238
16	H	2.8682	170	2.8392	158	2.7835	149	2.6933	155	2.5937	200
17	Y	2.6489	157	2.6954	150	2.3538	126	2.6759	154	2.3603	182
18	M	2.4296	144	1.8509	103	1.8121	97	1.7724	102	1.8934	146
19	X	1.4341	85	1.4196	79	1.4945	80	1.4248	82	1.634	126
20	V	1.0798	64	1.2579	70	1.3824	74	1.1816	68	1.2061	93
21	W	0.9954	59	1.15	64	0.9714	52	1.0252	59	0.83	64
22	J	0.8267	49	1.0422	58	0.8967	48	0.8862	51	0.817	63
23	B	0.7761	46	0.9164	51	0.7659	41	0.8688	50	0.7522	58
24	K	0.6917	41	0.7907	44	0.7286	39	0.7298	42	0.7133	55
25	Q	0.3374	20	0.3953	22	0.411	22	0.3823	22	0.3112	24
26	Z	0.0675	4	0.0719	4	0.0747	4	0.0695	4	0.0519	4

