computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

DISTRIBUTED TRUST IN *e*PAYMENTS SYSTEM

K. P. Vidya

Department of Mathematics, Madras Christian College(Autonomous),
Affiliated to University of Madras, Chennai, India.
kpvidya@hotmail.com

Abstract: *In this paper, a secret sharing scheme that is based on the Parallel Pollard rho Attack of the Elliptic Curve Discrete Logarithm Problem (ECDLP) is proposed for hierarchical access structures that can be activated dynamically. The shares of the scheme are distributed across two levels of participants but the reconstruction of the secret takes place at level zero which is the central processor or the trusted party of the scheme. The scheme finds its application in the Payments System of Banks and facilitates the replacement of paper cheques with eCheques. It also provides an efficient method of processing the payments at the Clearing House of Banks.*

Keywords: *Payment System, Secret Sharing Scheme, Elliptic Curve, ECDLP, Pollard's rho attack on ECDLP.*

1. PAPER SIZE

Large volumes of paper cheques are processed each day by the Clearing House of Banks. The initial effort at automating the payments system was the computerisation of preparing reports at the Clearing House which was followed by the introduction of magnetic ink character recognition(MICR) cheques and reader-sorter machines to count these cheques. The settlements statements stored in magnetic diskettes are delivered by physical courier to the clearing house and respective banks. This has lead to reduction in time required to process the payments and also a lesser percentage of mismatches at the time of tallying the results at the Clearing House at the end of transaction for each day. But the processing of outstation cheques drawn on branch offices of banks where there are no such high speed clearance systems takes time and allows withdrawal of the funds credited to the customers account only on the second day or in some cases after a week or two. A possible solution to overcome this drawback is the use of *e*Cheques or other *e*Payment Instruments together with secure methods of processing these instruments presented each day at the Clearing House.

Precisely, the problem is defined as a secure multiparty protocol among entities in a hierarchically structured *e*Payments network of Banks that ensures circulation of money. A transaction involves five entities: Alice, Bob, Alice's Bankers, Bob's Bankers and the Central Bank together with the Clearing Members of the respective Banks at the Clearing House.

In section 1 we give an introduction to the motivation of this paper while section 2 deals with a concise description of secret sharing schemes. Section 3 and 4 describes the mathematical background of the scheme and Section 5 describes the proposed secret sharing scheme, the communication model, the protocol, and mechanism for hierarchical access structures that can be activated dynamically. In Section 6, the advantages of the scheme are discussed along with the necessary precautionary methods to be considered for additional security in the payments system.

2. SECRET SHARING

A secret sharing scheme is that in which a secret α is divided into n shares which are distributed among the n participants so that a coalition of authorized participants can combine to reconstruct the secret. Shamir's[9] results based on Lagrange's interpolation of polynomials simultaneously with Blakley's[1] contribution were the first ever known secret sharing schemes that were later classified as threshold schemes. If only a coalition of $t \leq n$ participants can reconstruct the secret while t–1 or fewer participants cannot, then the scheme is called a threshold secret sharing scheme with a threshold value of t.

If Φ denotes the group of participants and Γ and Δ respectively denote the set of authorized and unauthorized participants where Γ and Δ are assumed to be mutually disjoint then the collection

(Γ, Δ) is called the *access structure* of the secret sharing scheme. The access structure is called a *monotone* access structure if a set P containing Γ is also a set of authorized participants. A *hierarchical* threshold access structure [12] defines sets of participants distributed in different levels with different or same threshold values for each level. Shamir's scheme is a weighted threshold scheme that uses Lagrange's Interpolation. Tassa's scheme is that which uses Birkhoff's interpolation to solve the threshold secret sharing problem over the hierarchical access structure where there is at least one participant who belongs to each level in the secret sharing scheme. If different access structures in a family of access structures are to be activated at different instances of time then we say that the secret sharing scheme is dynamic. A *fully dynamic* secret sharing scheme as defined in [2] is the sharing of a set of secrets among a group of participants such that any subset of participants has no information about the new secret before knowing the new broadcast message but there exists a perfect secret sharing scheme after seeing the new broadcast message.

A *perfect* secret sharing scheme is one in which the shares corresponding to each unauthorized subset provides absolutely no information about the shared secret. In fact, they have a monotone access structure. The efficiency of any secret sharing scheme is measured by its information rate = (Size of the shared secret) / (Size of that participant's share). Since in any perfect secret sharing scheme the size of a share is greater than or equal to the size of the shared secret for all shares of the participants of the scheme, it follows that all perfect secret sharing schemes have information rate ≤ 1. Secret sharing schemes of rate 1 are called *ideal*. The *Shamir's scheme* is an example of a perfect and ideal threshold scheme.

3. ELLIPTIC CURVES AND ECDLP

An elliptic curve E defined over a finite field F_q, of characteristic greater than three is given by the set of points that satisfy the equation $y^2 = x^3 + ax + b$, a, $b \varepsilon F_q$ where, discriminant $\Delta = -16(4a^3 + 27b^2) \neq 0$ together with the point at infinity O. It forms an abelian group over a special type of addition, where, O serves as the identity element of the group and the inverse of a point $R = (x_1, y_1)$ on the curve is given by $-R = (x_1, -y_1)$. The Group law for addition of two points $R = (x_1, y_1)$ and $S = (x_2, y_2)$ for $R \neq S$ and $S \neq -R$, is given by the co-ordinates $(x_3, y_3) \varepsilon E(F_q)$ where, $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda (x_1 - x_3) - y_1$ and the slope λ is given by $(y_2 - y_1)/(x_2 - x_1)$ if $R \neq S$ and $S \neq -R$ and $(3x_1^2 + a)/2y_1$ if $R = S$. The order p of the elliptic curve over F_q, i.e., the number of elements in the abelian group is determined by the bounds stated in Hasse's Theorem $q + 1 - 2\sqrt{q} < p < q + 1 + 2\sqrt{q}$ while the order of a point $R \varepsilon E(F_q)$ is the smallest positive integer α for which $\alpha R = O$. Further, if the group is of prime order it implies that the group is cyclic and every element of the group other than O is a generator of the group.

To define the elliptic curve discrete logarithm problem (ECDLP): Given an elliptic curve E defined over a finite field F_q, a point $P \varepsilon E(F_q)$ of order p, and a point $Q \varepsilon \langle P \rangle$, find the integer $l \varepsilon [0, p - 1]$ such that $Q = lP$. The integer l is called the discrete logarithm of Q to the base P, denoted $l = \log_P Q$.

4. POLLARD'S RHO ATTACK ON ECDLP

The Pollard rho attack on the *ECDLP* [8] finds two distinct pairs (c', d'), (c'', d'') of integers modulo p such that the points $X' = c'P + d'Q$ and $X'' = c''P + d''Q$ collide. That is, a suitable iteration function f: $\langle P \rangle \rightarrow \langle P \rangle$ is defined so that any point X_0 in $\langle P \rangle$ determines a sequence $\{X_i\}_{i \geq 0}$ of points where $X_i = f(X_{i-1})$ for $i \geq 1$. Now, since $\langle P \rangle$ is finite, the sequence will collide at some i^{th} iteration and then cycle for the remaining iterations forming a ρ-like shape. Then value of l can be obtained by computing $l = (c' - c'')(d'' - d')^{-1} \bmod p$. The improvisations suggested by Teske [13, 14], has a runtime complexity of $\sqrt{(\pi n/2)}$.

In the Parallel Pollard rho attack[7], the single processor is replaced by M processors to speed up the process of computation. For the same iteration function f each processor determines points in $\langle P \rangle$ having an identical pre-defined distinguishing property and sends these points to the Central Processor. The Central processor computes the value of l from the triples associated with these distinct values received from the M processors as $l = (c' - c'')(d'' - d')^{-1} \bmod p$. This algorithm is known to have a runtime complexity that is linear in the number of processors.

In the proposed secret sharing scheme, l is set as the secret α. To generate the shares or shadows, $\langle P \rangle$ is partitioned into sets of roughly the same size. These form the shares that are distributed to all the participants of the scheme. The threshold values at each level of the scheme are set depending on the size of the secret, the processing speed and the minimum number of machines used for the purpose, and the least number of partitions that will be required to compute the secret within the stipulated time as per the requirements of the chosen application. Thus the computational feasibility of the secret defined by the boundaries of the security conditions of the application plays an important role in determining the threshold value at each level of

the scheme.

5. PARALLEL POLLARD SECRET SHARING SCHEME

5.1. COMMUNICATION MODEL

The scheme is defined for a hierarchical structure with two levels. At level zero is the Trusted Party *T* in this case the Central Processor (Entity), who deals the shares and reconstructs the secret. At level 1, just below level zero, is the set of *M* Processors (Entities) which compute the distinguished points from the set of shares received from the participants in level 2. In relation to the application of this scheme to Payments System, the Central Processor denotes the Central Bank's node in the Clearing House. Each of the *M* Processors located in the Clearing House represent the clearing members of the individual Banks. The participants in level 2 are the authorized officials at the branch offices of the banks.

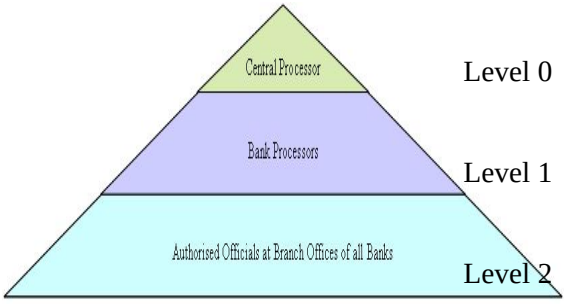

Fig. 1 – The hierarchy defined in our model for the Payments System in Banks.

5.2. PROTOCOL

The protocol in the proposed *e*Payments System consists mainly of five entities: Alice, Bob, Alice's Bankers, Bob's Bankers, and the Central Bank and the Clearing Members of the respective Banks at the Clearing House.

The *e*Cheque is encrypted at any bank using the public key published by the Central Bank at the Clearing House. The Central Bank distributes the shares of the secret key to all participants of the banking network. The *e*Cheques can be decrypted only at the Clearing House using the private key that is reconstructed with the shares provided by the members of the banking network.

The *e*Payment instruments presented at the branch offices of banks for transfer of funds are digital images of the *e*Cheques created by the respective bank with appropriate software applications. The cipher text of these *e*Payment Instruments are embedded in them. The branch offices sort these *e*Cheques based on the banks and branch and prepare a consolidated statement of the sum totals of the amounts to be received from each bank. Each branch office then sends the *e*Cheques along with the statements to the respective nodal office at the Clearing House. The authorized officials at each branch also affix their shares on the statements that are sent to the Clearing House.

To decrypt the *e*Cheques, the secret key has to be reconstructed from the shares received from the participants, for which purpose, the Central Processor may choose any access structure. When the secret key is reconstructed the *e*Cheques are decrypted and processed for payment to respective banks. The final tally is carried out and the statements are prepared for the *e*Cheques that are accepted and rejected. The consolidated amounts that are due to each bank is settled via *e*Payment instruments drawn on the Central Bank.

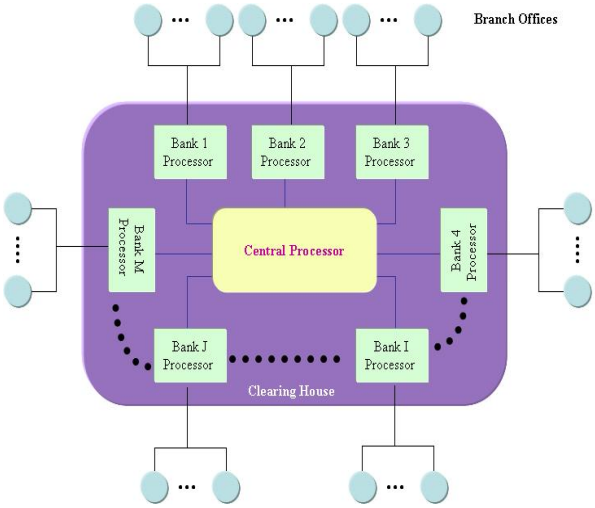

Fig. 2 – *e*Payments System Communication Model.

Let us suppose that the public key pair of the cryptosystem used by the Clearing House is (h, g), where g is the generating element of the chosen finite field F_p^* of order $p-1$ in which the Discrete Logarithm Problem is intractable. Let α be the private key that is shared among all the participants of the banking network.

Now let us assume that the bankers of Alice are branch office A_{11} of Bank B_1 and they provide her with *e*Cheque facility for withdrawal of funds from her personal account. Let us also suppose that branch office A_{21} of Bank B_2 is the banker for Bob. Now, if Alice wants to write out an *e*Cheque in the name of Bob for the goods that she has purchased from him, she would access her account with A_{11} of Bank B_1 and fill up the *e*Cheque fill-out form. We now assume that, the *e*Cheque is converted into an image. The details on the *e*Cheque consists of the string "Date : Time : Payee : Amount (words and Figures) : Account Type : Account Number : Drawee Bank (Name and Address) : Drawer (Name

and Signature) : *e*Cheque Number : Bank Code". These details are imbedded as a number m in F_p^* and encrypted using the public key $h = g^{\alpha}$, published by the Central Bank Server. The cipher text c given by $c = mh = mg^{\alpha}$ is embedded in this image and the *e*Cheque is posted to Alice's email account who then sends it to Bob. Now, if Bob wishes to encash the *e*Cheque, he would fill-out the *e*Cheque deposit form and post it to his bankers. The *e*Cheque would undergo the process as per the banking rules. When the Central Bank Server triggers off the reconstruction of the secret, the secret key α is reconstructed and the *e*Cheques are decrypted as $m = cg^{-\alpha}$. All computations are with respect to modulo p.

Now, the bit stream of the cipher text that was embedded in the image of the *e*Cheque is decrypted and a search algorithm finds the corresponding match to the *e*Cheque in the fill-out-forms submitted by Alice's bankers. The *e*Cheque is thus processed and funds transferred from Alice's account to Bob's account. In case of discrepancies such as an outdated *e*Cheque or insufficient funds the instrument is returned to Bob by his bankers.

For example, let F_{37}^* be the underlying finite field whose generator g is 2. Then let ($h = 2^{30}$, $g = 2$) be the public key and $\alpha = 30$ be the private key. Then for $m = 10$, the binary equivalent of cipher text $c = m * h = 10 * 2^{30} = 36 \pmod{37}$ is embedded in the image of the *e*Cheque. On reconstruction of the secret at the clearing house, the value of m is obtained from c as $m = c * g^{-\alpha} = 36 * 2^{-30} \pmod{37} = 10$.

5.3. THE SCHEME FOR A SINGLE PROCESSOR

We first describe the scheme for the case of a Single Processor. In this scheme, α is set as the secret of the threshold scheme. $<P>$ is partitioned into n number of sets of roughly the same size and these form the shares or shadows that are distributed to all the participants A_i, $i = 1,2, \ldots, n$. The threshold value is denoted by the value of k. The scheme has three main phases, the Setup, Share Generation, and the Reconstruction of the Secret. An entity T plays the role of the trusted authority who generates the public and private key pair and the shares to the decryption key. The mechanism for a scheme with a threshold value of two is given below. The function H is a hash function that determines a point in a partition and the function f is an iteration function that determines the sequence of points in the elliptic curve that collide at some stage. In our scheme we define simple functions for H and f.

5.3.1.MECHANISM: The Pollard Threshold Scheme

SUMMARY A secret α that controls a critical action is distributed among n participants A_i $i = 1, \ldots, n$, of the secret sharing scheme.

RESULT For $k = 2$ or more number of participants pool in their shares to trigger the critical action where $k \le n$.

A. Setup

A trusted entity T

1. Selects an elliptic curve E over a finite field F_q generated by $<P>$ of prime order p.
2. Sets the secret α that controls the critical action as a random integer l and determines the point $Q = lP$ on E.
3. Selects a partition function H: $<P> \rightarrow L = \{1, 2, \ldots, m\}$ where m indicates the number of partitions that are used during recovery of secret. Here, we choose a simple partition function such that, for $X' \varepsilon <P>$, $h = H(X') = H(x, y) = x \bmod m + 1$ for $k \le m \le n$.

B. Share Generation

1. Selects random integers a_i, $b_i \varepsilon [0, p-1]$ and computes $R_i = a_iP + b_iQ$, $i = 1 \ldots n$
2. Distributes the shares $S_i = (a_i, b_i, R_i)$ to the participants A_i, $i = 1, \ldots, n$
3. Keeps the verification parameters (P, Q) confidential along with the list of R_i.

C. Reconstruction of the Secret (in the case of 3>k inputs)

(a) Verification of the Shares

1. T receives the shares from any three participants, say, A_j, $j = 1,2,3$.
2. Uses verification parameters P and Q to compute $a_jP + b_jQ = V_j$, $j = 1,2,3$.
3. Verifies if V_j equals R_j and if they are found to be equal, for all $j = 1,2,3$, respectively, the steps in Recovery of Secret is carried out.

(b) Recovery of the Secret

1. Set $L = \{1, 2, 3\}$.
2. Set $c' = \Sigma a_j \pmod p$, $d' = \Sigma b_j \pmod p$ and $X' = \Sigma R_j = c'P + d'Q \pmod p$ for $j = 1,2,3$.
3. Repeat
   a) Compute $h = H(X')$ where $h = 1, 2$ or 3 corresponds respectively to $j = 1, 2$ or 3.
   b) Set $X' = X' + R_h \pmod p$, $c' = c' + a_h \bmod p$, $d' = d' + b_h \bmod p$.
   c) For r from 1 to 2 do
      (i) Compute $h = H(X'')$. where $h = 1, 2$ or 3 corresponds respectively to $j = 1, 2$ or 3.
      (ii) Set $X'' = X'' + R_h \pmod p$, $c'' = c'' + a_h \pmod p$, $d'' = d'' + b_h \pmod p$.

   Until $X'' = X'$.
4. Compute $l = (c' - c'')(d'' - d')^{-1} \bmod p$ which

is the secret α and trigger the critical action.

5. Exit.

5.3.2 ILLUSTRATION

Suppose that, A_i, i = 1, …, 5, are the participants of a secret sharing scheme and that a subset of two or more participants are to combine to reconstruct the secret key that controls a critical action.

Setup: The trusted entity T selects at random the elliptic curve $E(F_{29})$ given by $y^2 = x^3 + 4x + 20$ where the discriminant $\Delta = -176896 \not\equiv 0 \pmod{29}$. The number of elements in the elliptic curve group is 37 a prime, and so, $E(F_{29})$ is a cyclic group. All elements of $E(F_{29})$ for $P = (1, 5)$ as the generator are listed in the Table 1 below. Now assume that, the secret S that triggers this critical action is set as equal to 30. If the point P in our algorithm is chosen to be the pair (1, 5) then $Q = 30P = (24, 7)$.

Share Generation: The trusted entity T computes the shares of the secret as $S_i = (a_i, b_i, R_i)$ for i equal to 1 to 5 and distributes them to each A_i. The corresponding shares for each A_i are S_1 = (28 , 34 , (19, 13)), S_2 = (17 , 27 , (16, 27)), S_3 = (20 , 14 , (15, 2)), S_4 = (14 , 23 , (1, 5)), S_5 = (12 , 3 , (14, 6)). These shares are distributed to each participant A_i, i = 1, …, 5.

Verification: Now, suppose that, A_1, A_3 and A_5 wish to carry out the critical action. The shares S_1 = (28 , 34 , (19, 13)), S_3 = (20 , 14 , (15, 2)) and S_5 = (12 , 3 , (14, 6)) are input to the system and the shares are verified for their authenticity.

Reconstruction: The initial values of the iterating function are given by (c', d', X') = (23, 14, (1, 24)) where c', $d' \in [0, 36]$ and $X' = c'P + d'Q = 23P + 14(30P) = 36P$ modulo 37 = (1, 24). The tabulations of c', d', X', c'', d'', X'' for the iterations are shown in Table 2. The process terminates in the 4th iteration when $X' = X'' = 31$P. The corresponding values of c', d', c'', d'', are 16, 19, 7, 23 respectively.

Now, $l = (16 - 7)(23 - 19)^{-1} \pmod{37} = 30$ gives the value of the secret α. On the reconstruction of the secret α the critical action is carried out by the participants of the authorized set A_1, A_3, and A_5.

5.4 THE SCHEME FOR THE PARALLEL PROCESSOR

The proposed scheme is an extension of the scheme described above and relies on the intractability of the ECDLP. This scheme is defined for a hierarchical access structure of two levels with additional capabilities of verification of shares and dynamic activation of an access structure.

5.4.1. SYSTEM SETUP

Suppose that, α is the secret that triggers the critical action. A trusted entity T, that is the Central Bank Server, divides the secret α such that these shares may be distributed to all the participants of the network. The secret α is chosen to be an integer $l \in [1, p-1]$ where P and $Q = lP$ are points on a

Table 1.

0P=O	7P=(24,22)	14P=(5,22)	21P=(0,7)	28P=(14,6)	35P=(4,10)
1P= (1,5)	8P=(8,10)	15P=(3,1)	22P=(3,28)	29P=(8,19)	36P=(1, 24)
2P= (4,19)	9P=(14,23)	16P=(0,22)	23P=(5,7)	30P=(24,7)	
3P= (20,3)	10P=(13,23)	17P=(27,2)	24P=(16,2)	31P=(17,10)	
4P=(15,27)	11P=(10,25)	18P=(2,23)	25P=(19,16)	32P=(6,17)	
5P= (6,12)	12P=(19,13)	19P=(2,6)	26P=(10,4)	33P=(15,2)	
6P=(17,19)	13P=(16,27)	20P=(27,27)	27P=(13,6)	34P=(20,26)	

Table 2.

Iter	c'	d'	X'	c''	d''	X''
---	23	14	36P = (1, 24)	23	14	36P=(1, 24)
1	6	28	32P = (6, 17)	25	22	19P=(2, 6)
2	34	25	7P = (24, 22)	19	10	23P=(5, 7)
3	25	22	19P = (2, 6)	13	35	27P=(13, 6)
4	16	19	31P = (17, 10)	7	23	31P=(17, 10)

randomly chosen elliptic curve E. The point P generates the curve E of prime order p defined over a finite field F_q of characteristic greater than three. The trusted entity T also selects a partition function H: $<P> \rightarrow L = \{1, 2, \ldots, m\}$ where L is the set of indices used to index the partitions used in the reconstruction of the secret key. The cardinality m of L is the number of partitions that are used to recover the secret. For simplicity the partition function considered here is such that for any point $X' \varepsilon <P>$, $h = H(X') = x \bmod m + 1$, spans all the elements of the set L. The minimum permissible cardinality of L is set as the threshold value k of the level two of participants of the secret sharing scheme. The value of k is dependent on factors such as hardware configuration of the machine used, the security strength of the secret, the minimum number of partitions required for computing the distinguishing point within the time limit specified for the application. $D = \{d_1, d_2, \ldots, d_{r'}\}$ is defined as the set of distinguishing properties for the points in $<P>$. The elements in D determine the choice of the access structure of the secret sharing scheme that is to be activated at any point of time. T sets the threshold value t for level one of the hierarchy as the minimum number of Bank Processors required to compute the value d_i from the broadcast message b.

5.4.2 SHARE GENERATION

The participants include entities (Bank Processors), B_i, $i = 1, \ldots, n$, at level one and branch offices A_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, m_i$ associated with each B_i at level two. T distributes the shares (S_i, r_i) to the processors B_i, $i = 1, \ldots, n$ in the level one of the scheme, where $S_i = (a_i, b_i, R_i)$, and $r_i \varepsilon [1, p-1]$ are chosen randomly. T then distributes the shares $S_{ij} = (a_{ij}, b_{ij}, R_{ij})$ to participants A_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, m_i$ in the level two of the scheme. T maintains the list of (r_i, R_i) of level one and (R_{ij}) of level two.The trusted party T also keeps the parameters P and Q confidential.

5.4.3 RECONSTRUCTION OF THE SECRET

The Central Bank Processor may choose any access structure to reconstruct the secret. It first chooses a distinguished property d_i from the set D. Let us suppose that the property of the chosen d_i, is that the leading u bits of the elliptic curve points are zeroes. T then uses the dynamic secret sharing scheme with single message broadcast as proposed in [2], described in mechanism 5.4.4, to activate the access structure.

Each of the r Bank Processors B_i pool in their respective r_j and determine d_i by subtracting Σr_j from b. The B_i verify the shares $S_{ij} = (a_{ij}, b_{ij}, R_{ij})$ of participants A_{ij} associated with them using (P, Q). If, out of the total number m_i of participants associated with each B_i, m number of them send their shares S_{ij} to the respective B_i , then L is set as $L = \{1, 2, \ldots, m\}$ where $m \geq k$.

Each of the processors in the access structure first compute the initial values of the iterative function f. They are set as c', d', X' where, $c' = \Sigma a_i$, $d' = \Sigma b_i$ and $X' = \Sigma R_i$. The operations are performed modulo p. If X' is a distinguished point having property d_i, the triples (c', d', X') associated with it are sent to the Central Bank Processor.

The Central Bank Processor also uses the verification parameters P and Q to verify if the shares S_i are authentic by computing $V_i = a_iP + b_iQ$, and comparing them with R_i in the respective S_i. If the Central Processor comes across duplicate distinguished points, the triples associated with it are used to compute the secret key. Suppose that the two triples are (c', d', X') and (c'', d'', X'') of integers modulo p. These are used to compute the secret α given by $(c'-c'')(d''-d')^{-1} \bmod p$. The reconstruction of the secret key α triggers the critical action of decrypting the cipher text embedded in the *e*Payment Instruments and the instruments are processed for payment.

5.4.4 Mechanism: Parallel Pollard Secret Sharing Scheme

SUMMARY A secret α that controls a critical action is distributed among the participants of the secret sharing scheme.

RESULT The set of participants belonging to the specified hierarchical access structure pool in their shares to reconstruct the secret that triggers the critical action.

I *Setup*: A trusted entity T

1. Selects an elliptic curve E over a finite field F_q generated by $<P>$ of prime order p.
2. Sets the secret α that controls the critical action as a random integer l and determines the point $Q = lP$ on E.
3. Defines a set $D = \{d_1, d_2, \ldots, d_{r'}\}$ of distinguishing properties for the points in $<P>$.
4. Sets the threshold value t for level one of the hierarchy as the minimum number of Bank Processors required to compute the value d_i from the broadcast message b.
5. Sets the threshold value k of level two based on the computational feasibility of the distinguishing point d_i, within the time limit specified for the application.

6. Selects a partition function H: $<P> \rightarrow L=\{1, 2,\ldots, m\}$ where m indicates the number of partitions that are used by each B_i to recover the secret key. Here again, a simple partition function is chosen, *i.e.*, for $X' \varepsilon <P>$, $h = H(X') = H(x, y) = x \bmod m + 1$.

II *Share Generation*

1. T selects random integers a_i, b_i, a_{ij}, $b_{ij} \varepsilon [0, p-1]$ and computes $R_i = a_iP + b_iQ$, $i = 1\ldots n$ and $R_{ij} = a_{ij}P + b_{ij}Q$, $i = 1\ldots n$, $j = 1,\ldots,m_i$.
2. T distributes the shares $S_{ij} = (a_{ij}, b_{ij}, R_{ij})$ to the participants A_{ij}, $i = 1,\ldots, n$, $j = 1,\ldots,m_i$ in level two of the scheme.
3. T distributes the tuple (S_i, r_i) to the participants B_i of level one, where $r_i \varepsilon [1, p-1]$ are chosen randomly.
4. T keeps the verification parameters (P, Q) confidential and also maintains the list of (r_i, R_i) of level one and (R_{ij}) of level two.

III *Reconstruction of the Secret*

A. *Central Bank Server activates an authorized subset in the access structure*:

Let us suppose that the set A is a family of access structures and $\Gamma_1 \varepsilon$ A is an authorised subset. Let t be the threshold value for the level one participant set and let r, $n \geq r \geq t$, be the number of B_i participating in the reconstruction of secret α.

1. If T wishes to activate the access structure $\{B_1B_3B_5\} \varepsilon \Gamma_1$ then T selects at random a $d_i \varepsilon D$. Let us suppose that the property of the chosen d_i, is that its leading u bits are zeroes.
2. T then computes the broadcast message as $b = d_i + r_1 + r_3 + r_5$. In general, for any access structure $\Gamma_1 \varepsilon$ A, the broadcast message $b = d_i + \Sigma r_j$, where r_j belongs to the share of B_i in Γ_1.
3. T then sends b to all the participants B_i in level one of the hierarchy. However only the authorized subset of participants can compute d_i from b.

B. *Each Bank Processor B_i in the chosen access structure does the following*:

1. Suppose that r number of B_i determine the d_i by pooling in their respective r_j and subtracting Σr_j from b.
2. Verifies the shares $S_{ij} = (a_{ij}, b_{ij}, R_{ij})$ of participants A_{ij} associated with them using (P, Q). Suppose that out of the total number m_i of participants associated with each B_i, m number of them send their shares S_{ij} to the respective B_i.
3. Sets $L = \{1, 2, \ldots, m\}$ where m is the total number of shares available with each B_i and $m_i \geq m \geq k$, and where k is the threshold value of the level two set of participants associated with each B_i.
4. Sets $c' = \Sigma a_j$, $d' = \Sigma b_j$ and $X' = \Sigma R_j = c'P + d'Q$, $j = 1,2,\ldots,m$.
5. Repeat
   a) If X' is the point with distinguishing property d_i then send the triple (c', d', X') to the Central Bank Processor together with their share S_i.
   b) Compute $h = H(X')$ where h corresponds to an element in L.
   c) Set $X' = X' + R_h$, $c' = c' + a_h \bmod p$, $d' = d' + b_h \bmod p$.

Until the Central Bank Processor receives another point X'' with the same distinguishing property from another processor involved in the computation.

C. Central Bank Server does the following:

a) Verification of the Shares

1. T receives the shares S_i along with the triple (c', d', X') associated with the distinguishing point d_i from the r processors B_i, $i = 1, 2,\ldots, r$, in the chosen access structure.
2. T then uses verification parameters P and Q to compute $a_iP + b_iQ = V_i$.
3. If $V_i = R_i$ for $i = 1\ldots, r$, the steps for recovery of secret are carried out.

b) Recovery of Secret

1. Computes $l = (c' - c'')(d'' - d')^{-1} \bmod p$ using the corresponding triples associated with any two sets of distinct points having the same distinguishing property d_i. The secret α is the value of l.
2. The reconstruction of α triggers the critical action of decrypting the *e*Cheques and processing them for payments.
3. The statements are prepared indicating the consolidated amounts due to each bank that are transferred via *e*Payment instruments drawn in favour of the central bank.

IV End program.

6 CONCLUSION

The novelty of our scheme is its application to hierarchical access structures having the ability to activate the access structure dynamically, all within the same framework.

As our scheme is based on elliptic curves, it

ensures higher levels of security for shorter key sizes. The shares function as digital signatures of the corresponding officials in the network for the life-time of the pair of keys. The shares of the secret can be generated easily and their authenticity can be verified at the time of reconstruction of the secret. Also, the generation of additional shares for new participants of the scheme does not compromise on the secrecy of existing shares. This makes it relatively easy to include new participants to the scheme without having to change the secret itself. Moreover, the fact that the Pollard rho attack on *ECDLP* uses negligible memory space during the iterative process contributes to the efficiency of our scheme. All these factors enable an easy implementation of our scheme for a very practical application in modernising the Payments System in Banks. However, care should be taken to set the life time period of the public and private key pairs to be a specified period of hours and the validity period of the *e*Payment Instruments encrypted using that public key should lie within this stipulated number of hours. Also, a better choice of the hash function and iteration function pair can result in more efficient computations of the secret key.

Thus our scheme eliminates the possibility of the adversary pre-determining the choice of the servers he/she may have to corrupt at any instant of time. It is difficult for an adversary to know the distinguishing point and the resources required to compute the secret key within the specified validity period of the *e*Cheque.

7. REFERENCES

[1] Blakley, G.R., Safeguarding cryptographic keys. In Proc. Nat. Computer Conf. AFIPS Conf. Proc., pp. 313-317, 1979.vol48.

[2] Blundo, C., Cresti, A., De Santis, A., Vaccaro, U., Fully Dynamic Secret Sharing Schemes, Theoretical Computer Science **155** (1996), 407-410.

[3] Desmedt, Y., Society and group oriented cryptography: a new concept. In C. Pomerance, editor, Advances in Cryptology, Proc. Of Crypto '87 (Lecture Notes in Computer Science 293), pp.120-127. Springer-Verlag. 1988. Santa Barbara, California, U.S.A., August 16-20.

[4] Desmedt, Y., Threshold cryptography. In W. Wolfowicz, editor, Proceedings of the 3rd Symposium on: State and Progress of Research in Cryptography, pp. 110-122, February 15-16, 1993. Rome, Italy, invited paper.

[5] Koblitz, N., Elliptic curve cryptosystems, Math. Comp., 48(177) : 203-209, January 1987.

[6] Kuhn, K., Struik, R., Random walks revisited: Extensions of Pollard's rho algorithm for computing multiple discrete logarithms, Selected Areas in Cryptography-SAC 2001 (Lecture Notes in Computer Science 2259) [468], 212-229, 2001.

[7] Oorschot van, P., Weiner, M., Parallel collision search with cryptanalytic applications, Journal of Cryptology, 12: 1-28, 1999.

[8] Pollard, J.M., Monte Carlo methods for index computation mod p, Math. Comp., 32(143): 918-924, July 1978.

[9] Shamir, A., How to share a secret, Communications of ACM, 22, pp. 612-613, November 1979.

[10] Silverman, J.H., The Arithmetic of elliptic curves, volume 106 of Graduate Texts in Mathematics, Springer-Verlag, 1986.

[10]Simmons, G.J., How to (really) share a secret, Santa Barbara, California, U.S.A., Advances in cryptology, Proc. of Crypto '88 (Lecture Notes in Computer Science) Springer-Verlag, August 1988.

[11]Smart, N., The discrete logarithm problem on elliptic curves of trace one, Journal of Cryptology, 12:193-196, 1999.

[12]Tassa, T., Hierarchical threshold secret sharing, M.Naor, editor, Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Volume 2951 of Lecture Notes in Computer Science, p.473-490, Springer-Verlag, 2004.

[13]Teske, E., Speeding up Pollard's rho method for computing discrete logarithms, Algorithmic Number Theory-ANTS-III (Lecture Notes in Computer Science 1423) [82], 541-554, 1998.

[14]Teske, E., On random walks for Pollard's rho method. Mathematics of Computation, 70: 809-825, 2001.

K.P.Vidya, *is a Research Scholar at the Department of Mathematics, Madras Christian College, Chennai, India since Jan 2003 till date.*

She has an MSc. in Mathematics and a Masters in Computer Applications. Her employment history includes experience at a Nationalized Bank and a Software Company in India.

Her research interests are Algebraic Cryptography and Group Oriented Cryptographic Protocols and Applications.