Design Efficient Distributed Key Generation For Secure Network Applications over Cloud

Ms. Pratima Popat Gutal
M. E. Student
Department of Computer Engineering,
JSPM's Rajarshi Shahu College of Engineering,Tathawade,Pune,India

***Abstract* - Distributed key generation cryptosystems has a major content called distributed key generation (DKG) protocol. This protocol allows number of participants of group to generate a set of keys such as private key and public key without determining some trusted participants. Only authorized subgroups of participants are capable to regenerate or implement the private key and the public key is an output in clear. Previous analysis says that the DKG protocols determine similar authority for participants. In our analysis, we take in account the problem of the DKG protocols in groups with different hierarchical structure where the authorized subsets will be classified through a HTAS (hierarchical threshold access structure) and for the begging propose we use a VHTSS (verifiable hierarchical threshold secret sharing) protocol. By analyzing references we understand that they presented new DKG protocol with hierarchical threshold access structure for discrete-logarithm-based cryptosystems. Also analysis of HTDKG protocol says that this protocol fulfills the needed security requirements. Furthermore, we have tendency to develop our HTDKG protocol for secure in front of adaptive attackers through presenting the attacker model as well as communication model. This system additionally proposed for the decrease the processing time and enhances the memory utilization by utilizing of KDC.**

***Index Terms* - Public key extraction, asynchronous, cryptosystems, threshold, discrete-logarithm-based cryptosystems.**

I. INTRODUCTION

Distributed cryptosystems are entirely based over the distributed key generation (DKG) protocols. DKG protocol works like a group of participants In DKG protocols, a group of participants commonly generates a set of private as well as public keys on the basis of the partially separated and illustrated by the fundamental cryptosystem, without determining any trusted party (dealer). This procedure accomplished without once having to calculate, rebuild or store the private key in any single location. Like this protocol, the public key is output in clear and the private key is retained as virtual secret shared via a secret sharing method [1]. This typical private key will be after implemented by a distributed cryptosystem, such as to calculate the distributed signatures or execute distributed decoding. The Pedersen utilized Feldman's verifiable threshold secret sharing (VTSS) protocol [3] is to propose the initial DKG protocol that is a (t, n)-threshold DKG (TDKG). In a (t, n) - TDKG protocol, subsets of participants who are capable to utilize the private key inside a distributed manner are those with minimum t peoples. The [4] various set of Pedersen's (t, n) - TDKG protocol are proposed for threshold cryptography. In 2007, Gennaro [1] shows that Pedersen's protocol [2] and thus each it sets, don't ensure uniform distribution of the created public key and proposed a novel (t, n) - TDKG protocol on the basis of Pedersen's VTSS protocol. Let's take an example of banking system, in which multiple user access the same data by using single key which will increase the time and also requires large memory size. But in our system we are making use of KDC to prevent the load on the system which takes responsibility of creation and distribution of the keys. Due to which the processing time of the system as well as the memory consumption is reduced. [1] In our contribution, we presented the initial practical protocol model for utilization on the Internet and password authentication. We officially show its security and livener's properties as well as demonstrate practical methodology to obtain cryptographically important property of uniform randomness of the shared secret. We also utilize as well as confirm the practical sense of our DKG protocol on the group of computer systems platform. Initially, we presented a practical system model on the Password authentication over Internet. Also, we included the standard Byzantine opponent with crash recovery as well as network failures in an asynchronous setting [7]. In this study we additionally studied the asynchronous versus partially synchronous dichotomy for the Internet and to legitimate the selection of conduct crashes as well as network failures divided. We explore a VSS technique that included in our system model. [3] Analyzing the requirement of a protocol for compliance over a set for asynchronous DKG, we utilize as well as show a practical DKG protocol (HTDKG) for utilization on the Internet. [7] We implement a leader depended compliance technique inside our DKG, as we analyze some pragmatic problems with the commonly preferred randomized compliance technique. We also determine the uniform randomness of the shared secret in DKG as well as change our HTDKG to complete uniform randomness in the random oracle model. At the end, we can utilize HTDKG protocol as well as test its execution [4-6] on the PlanetLab platform. For the best analysis, this is the initial time that a distributed cryptographic protocol within the asynchronous setting has been tested with until number nodes (replicas) spread across number of institutes/organization. We have expectation that the HTDKG protocol extends and far as the execution time as well as the system load and it will be implemented to comprehend threshold cryptography and MPC on the Internet [8]. We also demonstrate few system-level optimizations for HTDKG and determine the system's flexibility in front of attacks.

II. Related Work

Wherever Gennaro, R.,Jarecki, S.,Krawczyk, H [1] introduces the methods that ensures the security. At start, presents the normal needs to solve the issue of DKG problems. Further, introduce the suitable DLG method as well as carefully demonstrate that it satisfy the privacy requirements. Additionally, it proves the outcome achieved from the implementation of private as well as public key is as needed. Other than that it ensures the security needs of the algorithm on the basis of simulation argument.

Pedersen,T [2] unrecognized group are calculated. Commonly, this technique works over the enhancement of the threshold cryptosystem structured by Desmedt and Frankel. At the start, it demonstrate the methods for preventing the trusted party as well as who can choose and distribute the secret key. Further, it demonstrates that how to share the secret key in authenticated member group.

Kate, A., Goldberg, I [7] presented the technique as well as characterizing the resourceful showable secret sharing method. They analyzed the needs of Byzantine agreement for asynchronous DKG and additionally analyses the difficulty of implementation of a randomized method for it. Through implementing this showable sharing method designed a leader based agreement protocol.

III. Motivation

Up to the date, the problem of DKG is just determined on the basis of similar authority for all of the participants. So, the determinations not possible come true and but there is different cases in that participants have various hierarchy of authority and holding the similar integrity of the specifications.

IV. Problem Definition

In this paper, we concentrate over the issue of generating a suitable key within a group of participants with determination of different hierarchy of authority. As a solution, DKG protocol with HTAS is presented for discrete-logarithm-based cryptosystems. On the basis of this issue of discrete logarithm problem (DLP), the guarantee of the proposed protocol is presented. The introduced protocol is implemented such as a portion of distributed cryptosystems to generate hierarchical threshold signature/ encryption methods.

V. Implementation Details

System Overview

VSS with HTAS(VHTSS)[12]:
Let U = {P1,…, Pn} be the group of participants and D be the dealer.
Then, a VSS protocol is a pair phases as follows.

1. Sharing verify

In first step, over input the secret s, D developed the share relating to every participant $Pi \in U$ and sends it through a secure medium to Pi. The dealer additionally generates several public data to analyze the validity of shares. At the end of this analysis, every participant $Pi \in U$ is instructed to output value verification $\in \{accept, reject\}$..

2. Reconstruction

The input of this step is the shares relating to a subset of participants. Initially, the validity of all shares is confirmed by other participants. Then, if the set of participants with valid shares is an authorized set, the secret can be calculated by implementing a reconstruction function on the given shares [4].

A Verifiable Secret Share protocol is called ensure if it accomplishes the following security need [3]:

- Acceptance: If a honest participant outputs 'reject', then each honest participants also output 'reject' at the end of sharing verify phase. However, in the attacker model the dealer is not corrupted and then all honest participants output 'accept'.
- Verifiability/reconstruct ability: All subsets of participants determining one authorized subset of participants with valid shares retain the same unique secret. With the determination of honesty of the dealer, we should have $\sigma = s$, where s is the original shared secret.
- Privacy: Comparatively the dealer had not corrupted when no unauthorized subset of contributors is capable to secure information about the secret [9].

Existing TDKG protocols are not suitable when participants have various levels of authority .We propose new DKG protocol with HTAS (HTDKG).[1]

DKG with HTAS (HTDKG):

Let $U = \{P1,..., P2\}$ be the group of participants and p and q be two huge primes such that $q/(p-1)$ .A DKG protocol is a pair of phases.

1. Secret-key generation

In this initial step, every participant distributes random value. Finally, a secret x (mod q) is distributed among set of participants those access his shares.

2. Public-key extraction

In this second step, set of participants help to produce public key corresponding to secrets in previous phase(x),

$$g^x (\mathrm{mod} q)$$

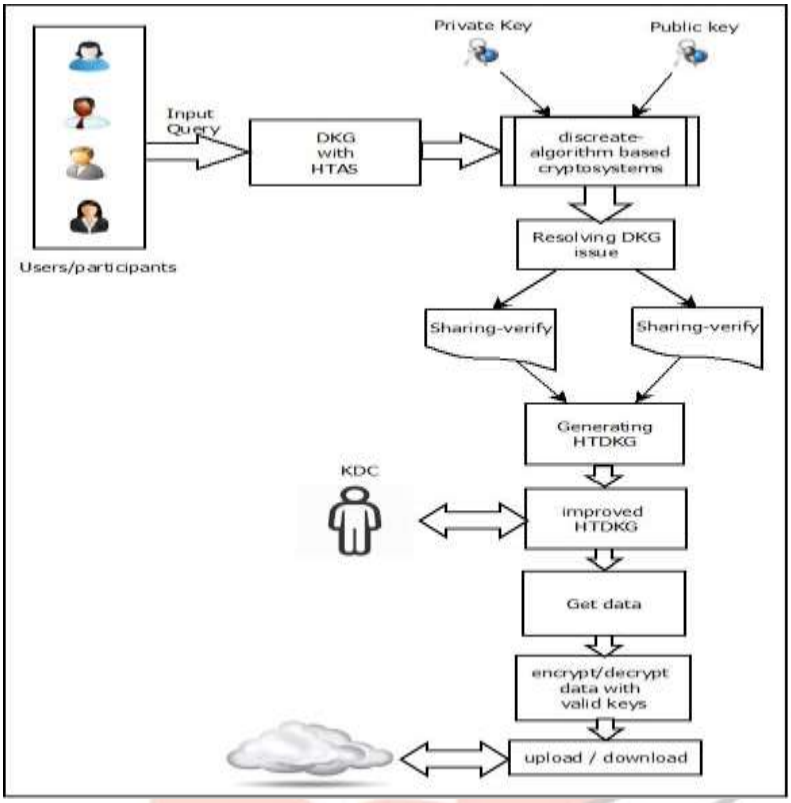

Fig. 1 System Architecture

A Distributed Key Generation protocol is called assure if it fulfills following security requirements:

- *Correctness*

(C1) The set of shares corresponding to every subset of participants, including one authorized subset of participants with valid shares define the similar unique private key x.
(C2) All truthful participants have the same value of public key y = gx (mod p), where x is the unique secret assured by (C1).
(C3) x is uniformly distributed in Zq (and hence y is uniformly distributed in the subgroup generated by g).

- *Secrecy*

Except for what is implied by the value $y = g^x (\mathrm{mod}\, p)$, no data on x can be obtained by an opponent who is not able to corrupt all of the participants in an authorized subset.

KDC (Key Distribution Center):

The Key Distribution center works on the participants request for key generation, as it takes participants request as input and check the authentication of the participants with registered participants, if user is valid then KDC runs Key Generation Algorithm and send the Key Pairs to the requested user.

Mathematical Model

Logarithms can be computed easily.
Mathematical representation for paillier algorithm is as follow,

$$(1+n)^x = \sum_{k=0}^{x} \binom{x}{k} n^k = 1 + nx + \binom{x}{2} n^2 + higher\ power\ of\ n$$

This indicates that:

$$(1+x)^n = 1 + nx\ (mod\ n^2)$$

Therefore, if:

$$y = (1 + x)^2 \, mod \, n^2$$

Then,

$$x = \frac{y - 1}{n} \, (mod \, n)$$

Thus,

$$L((1 + n)^x mod \, n^2) = x \, (mod \, n)$$

Where function L is defined as $L(u) = \frac{u-1}{n}$ (quotient of integer division) and, $x \in \mathbb{Z}_n$.

Let W is the whole system which consists,

$$W = \{I, U, D, R, O, KDC, CS\}$$

Where I be the input to the system,
U be the set of Participant or registered users in system,
R is the Reconstruction phase.

O be the output of proposed system
CS is cloud server,
KDC is key distribution center,

1. Input (I)

Let $U = \{P1, P2...Pn\}$ be a set of n participants which is partitioned into m + 1 levels $U_0, U_1, ..., U_m$. Suppose that the participants in $U_0 are P1, ..., P_{u0}$ the participants in $U_1 are P_{|u0|=1}, ..., P_{|u0|+|u1|}$ and so on, where —u1— denotes the size of the i[th] level for i = 0, , m. Suppose that the sequence of threshold requirements $t_0, t_1, ..., t_m (= t)$ determines the HTAS.

Let p and q be two large primes such that $q/(p1)$ and $G = \langle g \rangle$ be a subgroup of elements of order q in Z_p^*. Where Z_p^* is cyclic group of prime order p. To ensure well posedness of the interpolation problem needed in the reconstruction phase of the proposed protocol, the prime q also has to satisfy q > 2t + 2(t1) (t1)/2(t1)!n(t1)(t2)/2.

Let h be an element of G such that its relative discrete logarithm with g is unknown to any entity in the system.

2. Process

The whole process is divided into parts:
a) Sharing verify:
To share the secret $s \in Zq$ among participants P1, , Pn,

Step1: The dealer (D):

- Constructs polynomials f1() and f2() as follows

$$f1(x) = a0 + a1x + a2x^2 + ...at - 1x^{t-1},$$

$$f2(x) = b0 + b1x + b2x^2 + ...bt - 1x^{t-1},$$

Where a0 = s and a1 at1, b0 bt1 are random values in Zq.

- Broadcasts $C_i = g^{ai} h^{bi} (\text{mod} \, p) \, for i = 0t1$
- Computes the shares corresponding to P_j (j = 1,…, n) as follows

$$(sh_j \, f1^{(t_{k-1})}(j)(\text{mod} q), sh'_j = f_2^{t_{k-1}}(j)(\text{mod} q))$$

Where k is such that $P_j \in Uk$ and $f_b^{t_{k-1}}(x)$ is the $(t_k 1)$ th derivative of $f_b(x)$ for $b \in 0,1.$

- Sends $(sh_j, sh_j^{'})$ to Pj for j = 1 n via a secure channel. Step2: To verify the validity of his shares, each Pj U verifies the following relation

$$g_j^{sh} h_j^{sh'} = \prod_{i=0}^{t-1} C g_j^{tk-1}(j)(\text{mod} \, p).....(1)$$

Where $g_j^{tk-1}(j)$ is the value of (t_{k-1}) th dervative of $g_i(x) = x^i$ at point x = j and k is such that $P_j \in Uk$. Broadcasts a complaint against the dealer if does not hold.

Step2: If the set of complaining participants is an authorized subset, then all participants output reject and stop.

Step3: The dealer reveals the shares $(sh_j, sh_j^{'})$ corresponding to each complaining participant Pi.

Step4: Each participant checks the validity of the revealed shares by the dealer through and outputs reject if any of the revealed shares fails this equation. Otherwise, outputs accept. [11]

b) Reconstruction(R):

Suppose that t participants $P_{\alpha 1},..P_{\alpha t}$ .fulfill all threshold requirements, pool their shares $((sh_{\alpha 1}, sh_{\alpha 1}^{'}),...,(sh_{\alpha t}, sh_{\alpha t}^{'}))$ to recover the secret. At first, the validity of the shares is checked. Then, if all of the shares are valid, the polynomials f1(.) and f2(.) can be reconstructed by employing Birkhoff interpolation on pairs $(\alpha_t, sh_{\alpha}^{'}t),,1 \le i \le t,$ respectively. The secret can be retrieved as the constant term of f1(.).

c) Key Distribution Centre (KDC):

Receiving the key request from participant {P1,P2,P3,Pn} according to their level (U_0, U_1, U_m), KDC generate Keys and distribute to the requesting participant, if keys are already generated then it distribute according to their level of accessing the system KDC. During this processes the authentication of Participant is checked, if participant is authenticated person then he gets the required keys, which further uses those keys for encryption and decryption.

d) Cloud server (CS):

The Cloud server CS stores the data or files uploaded by participant (P1, P2, …, Pn).

3. Output (O)

Generating hierarchical threshold distributed keys ((PK1, SK1), (PK2, SK2),...,(PKm, SKm)). PK: Private Key SK: Secret Key

Algorithm

Paillier Algorithm

1. Key generation

- Choose two large prime numbers p and q randomly and independently of each other such that gcd (pq, (p-1) (q-1)) = 1. This property is assured if both primes are of equal length.

1. Compute n=pq and λ=lcm (p-1, q-1).

2. Select random integer g where $g \in \mathbb{Z}_{n^2}^*$

3. Ensure 'n' divides the order of 'g' by checking the existence of the following modular multiplicative inverse:

$$\mu = (\mathrm{L}(g^{\lambda} \bmod n^2))^{-1} \bmod n$$

where function L is defined as .

$$L(u) = \frac{u-1}{n}$$

Note that the notation $\frac{a}{b}$ does not denote the modular multiplication of 'a' times the modular multiplicative inverse of 'b' but rather the quotient of 'a' divided by 'b', i.e., the largest integer value v ≥ 0 to satisfy the relation a ≥ vb.

- The public (encryption) key is (n, g).
- The private (decryption) key is (λ, μ)

If using p,q of equivalent length, a simpler variant of the above key generation steps would be to set $g = n + 1$, and $\lambda = \varphi(n)^{-1} mod\ n$, where $\varphi(n) = (p-1)(q-1)$.

2. Encryption

1. Let 'm' be a message to be encrypted where $m \in \mathbb{Z}_n$
2. Select random 'r' where $r \in z_n^*$
3. Compute ciphertext as: $c = g^m . r^n mod\ n^2$

3. Decryption

1. Let be the ciphertext to decrypt, where $c \in z_{n^2}^*$
2. Compute the plaintext message as: $m = \mathrm{L}(c^{\lambda} \bmod n^2 .\ \mu \bmod n)$

V. Result and Discussion

We can contrast the Pedersen's [2] TDKG (t, n) and Gennaro [1] TDKG (t, n) that defined as TDKGP and TDKGG. At the time of increased number of interaction rounds increases the difficulty and gets more secured as compared with previous synchronized model. On the basis of experimentation Pedersens implementing TAS (Threshold Access Structure) but no entire security kept but Gennaro also utilizes TAS access stucture that is secured. In existing Pakniat [12] implemented a HTAS hierarchical TAS [10] that

have also entire security. So, result is that when the interaction rounds are increases by implementing the hierarchical access structure the security always increases. The required number of multiplication as well as exponentiation operations in TDKGP is about half of that in TDKGG. Hence TDKGP is more effective as compared with TDKGG as well as Pakinat HTDKG model. Figure 2 demonstrates the time comparison graph. Previous system runs without KDC. Thus it implements number of processes over client side and increments in execution time over client side. Proposed system implements KDC to decrease the execution time. In Figure 2X-axis shows systems while Y-axis shows Time in milliseconds.

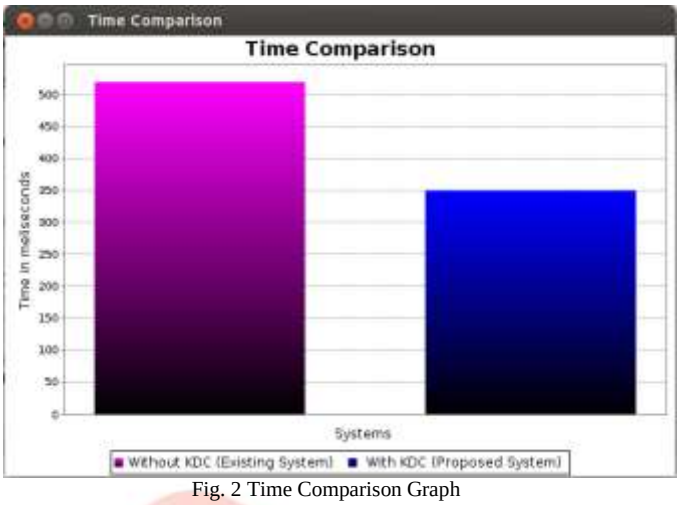

Fig. 2 Time Comparison Graph

Figure 3 demonstrates the memory comparison graph. Previous system runs without KDC. Thus it implements number of processes over client side and increments memory overhead over client side. Proposed system implements KDC to decrease the memory overhead. In Figure 3 X-axis shows systems while Y-axis shows memory in bytes.

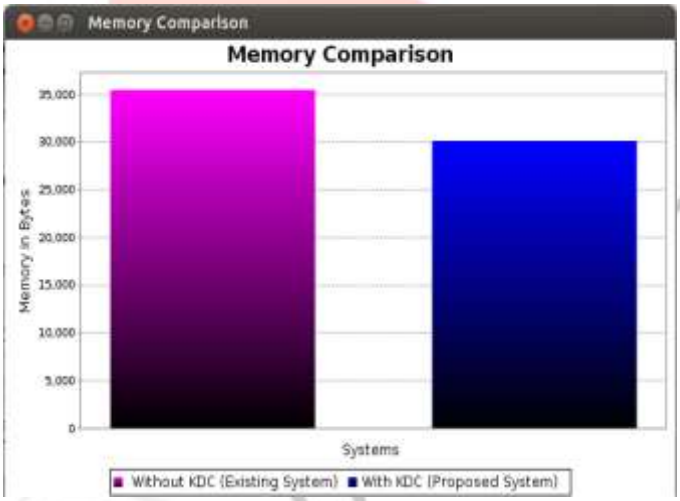

Figure 3: Memory Comparison Graph

VI. Conclusion and Future Scope

By the merge of DKG technique with pairing-based cryptography technique like CBE is helpful to practically whole distributed DCA as well as encryption/decryption methodology and able to provide certificate for participant recognizing and that will be implemented as keys for encryption as well as decryption. In this methodology there is no requirement of trusted entities as well as it is simple to add and eliminate the participant and also CBE certificate revocation technique. It is additionally proven that on the basis of techniques performance characteristics that until the participant number is included inside an appropriate bound (that will be matched through most present distributed network applications such as Ad Hoc networks or MANETs); a robust as well as effective security foundation will be obtained by providing flexibility, scalability and robustness. Furthermore, we has tendency to add dishonorable participant security in proposed system as well as offer a testing platform security infrastructure for distributed network applications. In future, we have strategy; in that we create our HTDKG protocol secure in front of adaptive attackers through presenting reverse attack same to the one in.

Moreover, we also implement the random oracle assumption to obtain uniform randomness of the shared secret in HTDKG. Additionally, we will try to enhance the key generation method as the proposed system introduced. This system implements hierarchical structure that is role based (roles are determined like manager, employee, etc). So, we will implement identity based or attribute based system to enhance the system.

VII. ACKNOWLEDGMENT

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

REFERENCES

[1] Gennaro, R., Jarecki, S., Krawczyk H. et al., "Secure distributed key generation for discrete-log based cryptosystems," J. Cryptol., 2007, 20(1), pp. 51-83.

[2] Pedersen T, "A threshold cryptosystem without a trusted party," Proc. Advances in Cryptology (Eurocrypt91), Brighton, UK, April 1991, pp. 552-526.

[3] Feldman, P., "A practical scheme for non-interactive verifiable secret sharing," Proc. 28th Annual Symp. On Foundations of Computer Science, (FOCS 87), CA, USA, October 1987, pp. 427438.

[4] Herzberg, A., Jakobsson,M. Jarecki, S.,et al., "Proactive public key and signature systems," Proc. Fourth ACM Conf. on Computer and Communications security(CCS 97), Zurich, Switzerland, April 1997,pp.100-110.

[5] Wang, F., Chang, C.-C., Harn, L., "Simulatable and secure certificate-based threshold signature without pairings," Security and Communication Networks, 2013, 7, (11), pp. 20942103.

[6] Fournaris, A.P., "A distributed approach of a threshold certificate-based encryption scheme with no trusted entities," Inf. Secure. J. Glob. Perspect., 2013, 22, (3), pp. 126139

[7] Kate, A., Goldberg, I., "Distributed key generation for the internet," Proc. 29th IEEE Int. Conf. on Distributed Computing Systems (ICDCS 90), Montreal, Quebec, Canada, June 2009, pp.119-128.

[8] Boneh, D., Franklin, M., "Efficient generation of shared RSA keys," Proc. Advances in Cryptology (Crypto 97), CA, USA, August 1997, pp. 425439

[9] Tassa, T., "Hierarchical threshold secret sharing," J. Cryptol., 2007, 20, (2), pp. 237264.

[10] Simmons, G.J., "How to (really) share a secret", Proc. Advances in Cryptology

[11] Pakniat, N., Noroozi, M., Eslami, Z., "Secret image sharing scheme with hierarchical threshold access structure," J. Vis. Commun. Image Represent., 2014, 25, (5), pp. 10931101

[12] Nasrollah Pakniat, Mahnaz Noroozi, Ziba Eslami., "Distributed key generation protocol with hierarchical threshold access structure," 2014, ISSN 1751-8709