Advances in Science and Technology Research Journal
Volume 9, No. 28, Dec. 2015, pages 1–9
DOI: 10.12913/22998624/60776

Research Article

DISTRIBUTED CERTIFICATE AUTHORITY IN CLUSTER-BASED MANET USING MULTI SECRET SHARING SCHEME

Mohammed Azza[1], **Sofiane Boukli Hacene**[1]

[1] EEDIS Laboratory, Computer Science Department, Djillali Liabes University of Sidi Bel Abbes, Algeria, e-mail: azza.mohammed.amine@gmail.com

Received: 2015.10.15
Accepted: 2015.11.14
Published: 2015.12.04

ABSTRACT

Providing secure communications in mobile ad hoc networks (MANET) is an important and difficult problem, due to a lack of a key management infrastructure. The authentication is an important security service in (MANETs). To provide a node authentication service we use a fully distributed certificate authorities (FDCA) based on the threshold cryptography. In this paper we propose an efficient and verifiable multi secret sharing scheme in cluster-based MANET with a low computation system. Our scheme is based on the overdetermined linear system equation in Galois fields $GF(2^r)$. We have analyzed our scheme based on security and performance criteria, and compared with existing approaches. The efficiency of our proposed schemes was verified and evaluated by simulation. Simulation results show that this approach is scalable.

Keywords: secret sharing scheme, clustering, mobile ad hoc network, overdetermined systems of linear equations, Galois fields, certificate authority.

INTRODUCTION

The dynamic aspect of the ad hoc network and lack of central controlling authority and infrastructure make this network vulnerable to some attack [1]. Authentication system is necessary to prevent a behavior of a malicious node. The cryptography is further classified into two categories symmetric key cryptography and asymmetric key cryptography, and it is one of the techniques used to provide communication in ad hoc networks.

The symmetric key cryptography uses one shared key and requires less computation and energy against the second type that uses two keys one private and the other is public. For efficient management of certificates systems we need a public key infrastructure (PKI) [2]. The certificate authority (CA) is the most important unit in PKI.

A PKI can be grouped with the kind of CA employed: Offline CA-based PKI and online partially distributed CA-based PKI and Online fully distributed CA-based PKI [3, 4]. The success of PKI is dependent on the availability of CA. The certificate authority is responsible for signing public key certificates and certificate revocation lists. The presence of a centralized CA is vulnerable that can be exploited by malicious behavior such as denial of service (DOS) [5].

The distribution of functionality of CA node in the network increases their availability, this technique is inspired from the threshold cryptography [6]. The key sharing scheme is introduced to avoid one to one encryption and added more security to symmetric cryptography and reduces the computing in asymmetric encryption. Many secret sharing applications, in particular those associated with the key management; require the protection of more than one secret. In [7] authors proposed a secure and efficient key management (SEKM) framework for mobile ad hoc networks. The secret key of CA is shared with m shareholders (server node). These nodes are capable to generate a partial certificate. In 2012 Xianyoun et al. [8] proposed a dynamic threshold key management system based on bilinear pairing. The threshold value can be changeable and there is no trusted third party TTP. Each node can verify the correctness of the share and the threshold polynomial based on elliptic curve cryptography (ECC). A key management infrastructure to secure distributed routing protocol SAODV proposed in [9]. It uses a CA certificate authority to sign the certificates of the new node through a key sharing scheme. The CA private key is divided into n nodes. When a source wants to send data, it reconstructs the secret key and encrypts de packet. The destination needs *t* share to decrypt the packet. In 2009 Dahshan et al. [10] proposed a self organized key based on trust management infrastructure between the nodes, when a new node joins the networks it must collaborate with *t* node for signing their certificate to become node trust.

The idea of secret sharing scheme (*t, N*) allows the dealers to divide the secret *K* over *N* share and distribution over a group of node *N* and only a *t* or more shares can reconstruct the secret *K*. The first (*t, N*) threshold secret sharing schemes (SSS) are proposed by Shamir [11] and Blakley [12] in 1979, independently. Shamir's scheme is based on the Lagrange interpolating polynomial, while Blakely's scheme is based on the geometry of hyperplanes in finite fields. The secret is a point of a t-dimensional space and it is represented by a coordinate, and the n shares are the affine hyperplanes passing through this point. A hyperplane in a t-dimensional affine space coordinated over a field *F*. Asmuth and Bloom [13] proposed using the Chinese remainder theorem (CRT) to build a secret sharing scheme in 1983. The secret *S* is reduced modulo a set of relatively primes integers $m_1, m_2, ..., m_n$ to produce the different shares, while the construction is performed by solving the system of *t* congruencies using the CRT. In [14] presents an enhanced and secured SSS based on CRT.

Later, several secret sharing schemes were proposed. Authors in [15] proposed a hierarchical secret sharing scheme based on interpolation of Hermite-Brikhoff that allows to derived the polynomial $p(x)$. The elements of the matrix are constructed with factorial function. In 2008, Zhang et al. [16] enhanced and proposed a hierarchical secret sharing based on Brikhoff interpolation without derived polynomial *p* and factorial operation.

In 2013, authors [17] proposed a threshold secret sharing scheme based on the cube of n-dimensions. In their scheme, the secret $S(x_i, y_i, z_i)$ is the geometrical center of a cube and all actions are spread over the surfaces of the cube. In 2009, Bai et al. [18] used the matrix projection to define the shares.

Using other methods, in 2009 Tassa et al. [19] proposed an approach based on binary linear codes. Li et al. developed and built four verifiable secret sharing schemes (VSSS):

1) class of binary irreducible cyclic codes;
2) class of BCH codes;
3) double-error correcting BCH codes Secret sharing and
4) Melas codes; where it is based on BCH (Bose, Ray-Chaudhuri and Hocquenghem) which is a class of cyclic errors-corrector code.

The concept of verifiability is used to ensuring the honesty of the participant. In verifiable secret sharing schemes (VSS), the validity of the shares is checked before the reconstruction process, many works are introduced in this concept. Authors in [20] create a public matrix M ($n \times t$) where each element M_{ij} is the hash information shared for each node. The shared information is verifiable with the discrete logarithm problem through the elliptic curve. In 2012 Hu et al. [21] proposed two verifiable multi secret sharing schemes (VMSS); the first uses Lagrange polynomial for distributing and reconstructing the secret and LFSR based on public key cryptosystem to check the validity of the data. The second scheme uses homogeneous LFSR sequence and the verification is based with a public key cryptosystem. Authors in [22] use the One-way function with two variables $f(r, x)$, these are enhanced in [23] with Combining one-way functions and Lagrange interpolation. In 2014 Tentu and al. [24] Combined the hash function and RSA, they used a hash function and the identity-based signature (IBS) schemes to verify the shares, moreover, use RSA to secure the distribution.

In this paper we propose an ideal and verifiable key sharing system based on overdetermined system of linear equation. The security in our scheme is unconditionally based on the discrete logarithm problem. It ensures the proactive and the dynamicity of shares.

OUR APPROACH

Our method allows sharing *m* secret $K = (K_1, K_2, ..., K_m)$ over a group of *n* nodes $N = (N_1, N_2, ..., K_n)$ and a threshold *t* that defines the node that can reconstruct the m secrets.

Initialization step:

In this phase the dealer has a set of key $K = (K_1, K_2, ..., K_m)$ who wants to share with all nodes, it chooses in the Galois field $GF(2^r)$:

1. Two random vectors $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ and $\beta = (\beta_1, \beta_2, ..., \beta_t)$ such as $\alpha_i + \beta_j \neq 0 \ \forall i, j$;
2. A public generator g.

Construction step:

1. Compute the Cauchy matrix C using equation 1

$$C_{ij} = \left(\frac{1}{\alpha_i + \beta_j}\right)_{i=1\ldots n, j=1\ldots t}, C = \begin{pmatrix} \frac{1}{\alpha_1+\beta_1} & \frac{1}{\alpha_1+\beta_2} & \cdots & \frac{1}{\alpha_1+\beta_t} \\ \frac{1}{\alpha_2+\beta_1} & \frac{1}{\alpha_2+\beta_2} & \cdots & \frac{1}{\alpha_2+\beta_t} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_n+\beta_1} & \frac{1}{\alpha_n+\beta_2} & \cdots & \frac{1}{\alpha_n+\beta_t} \end{pmatrix}, \alpha_i + \beta_j \neq 0 \ \forall i, j \quad (1)$$

2. Publish C for all nodes.
3. Compute the set of shares $P = \{p_i\}_{i=1\ldots n}$ as the following:

$$P = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix} = C.K^T = \begin{pmatrix} \frac{1}{\alpha_1+\beta_1} & \frac{1}{\alpha_1+\beta_2} & \cdots & \frac{1}{\alpha_1+\beta_t} \\ \frac{1}{\alpha_2+\beta_1} & \frac{1}{\alpha_2+\beta_2} & \cdots & \frac{1}{\alpha_2+\beta_t} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_n+\beta_1} & \frac{1}{\alpha_n+\beta_2} & \cdots & \frac{1}{\alpha_n+\beta_t} \end{pmatrix} . \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_{t-1} \\ k_t \end{pmatrix} \quad (2)$$

4. Compute $V_i = g^{k_i}$ for $i = 1 \ldots n$ (3)
5. Distribute for each node N_i its share P_i with secure channel.
6. Publish V_i.

Validation step:

Each node verifies its share and the consistency of the system through the following procedure:

1. It computes

$$Y_i = \prod_{j=1}^{t} \left(V_j\right)^{C_{ij}} = \prod_{j=1}^{t} \left(V_j\right)^{\frac{1}{\alpha_i+\beta_j}} \quad (4)$$

2. IF $Y = g^{p_i}$ so the share is valid and it accepts it, otherwise it rejects and asks another share.

Exception:

- IF $m < t$ t-m random secret is generated from $GF(2^r)$ to complete the vector $K = (K_1, K_2, ..., K_t) = (K_1, K_2, ..., K_m, R_m, R_{m+1}, ..., R_{t-m})$;
- IF $m > t$ the dealer compute a Cauchy matrix of $(n + m - t)$ line. The rest of share $(p_{n+1}, p_{n+2}, ..., p_{n+m-t})$ can attribute if the old is hacked or can give each node many shares

$$P = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \\ \vdots \\ p_{n+m-t-1} \\ p_{n+m-t} \end{pmatrix} = C.K^T = \begin{pmatrix} \frac{1}{\alpha_1+\beta_1} & \frac{1}{\alpha_1+\beta_2} & \cdots & \frac{1}{\alpha_1+\beta_m} \\ \frac{1}{\alpha_2+\beta_1} & \frac{1}{\alpha_2+\beta_2} & \cdots & \frac{1}{\alpha_2+\beta_m} \\ \frac{1}{\alpha_n+\beta_1} & \frac{1}{\alpha_n+\beta_2} & \cdots & \frac{1}{\alpha_n+\beta_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{n+m-t}+\beta_1} & \frac{1}{\alpha_{n+m-t}+\beta_2} & \cdots & \frac{1}{\alpha_{n+m-t}+\beta_m} \end{pmatrix} . \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_{m-1} \\ k_m \end{pmatrix} \quad (5)$$

Reconstruction step:

If *t* nodes want to reconstruct the m secret through a rebuilder, they send their share $\{P_i\} i = i_1..i_t$ with a secure channel. The rebuilder checks the validity of the shares:

1. It computes $Y_i = \prod_{j=1}^{t}\left(V_j\right)^{C_{ij}} = \prod_{j=1}^{t}\left(V_j\right)^{\frac{1}{\alpha_i+\beta_j}}$ for $i = i_1..i_t$ (6)
2. For each share $\{P_i\} i = i_1..i_t$ verifies if $Y_i = g^{p_i}$ so the share is valid, if it does not cancel the reconstruction.
3. The rebuilder constructs ($t \times t$) sub-matrix C_s where the lines corresponding to *t* participating nodes in the public matrix of the construction shares *C* (Cramer system).

$$C_S = \begin{pmatrix} C_{i_1 1} & C_{i_1 2} & \cdots & C_{i_1 t} \\ C_{i_2 1} & C_{i_2 2} & \cdots & C_{i_2 t} \\ \vdots & \vdots & \ddots & \vdots \\ C_{i_t 1} & C_{i_t 2} & \cdots & C_{i_t t} \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha_{i1}+\beta_1} & \frac{1}{\alpha_{i1}+\beta_2} & \cdots & \frac{1}{\alpha_{i1}+\beta_t} \\ \frac{1}{\alpha_{i2}+\beta_1} & \frac{1}{\alpha_{i2}+\beta_2} & \cdots & \frac{1}{\alpha_{i2}+\beta_t} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{it}+\beta_1} & \frac{1}{\alpha_{it}+\beta_2} & \cdots & \frac{1}{\alpha_{it}+\beta_t} \end{pmatrix} \tag{7}$$

To find the secrets, the rebuilder solves the system of linear equations $Cs \cdot X = P$.

4. He computes the inverse of sub-matrix C_s^{-1} in $GF(2^r)$ and multiplies it by the vector of corresponding shares $P = \left(p_{i1}, p_{i2} \ldots\ldots, p_{it}\right)$

$$K = \begin{pmatrix} \mathbf{k_1} \\ \mathbf{k_2} \\ \vdots \\ \mathbf{k_{t-1}} \\ \mathbf{k_t} \end{pmatrix} = \boldsymbol{C_s^{-1}} \times \mathbf{P} = \boldsymbol{C_s^{-1}} \times \begin{pmatrix} \mathbf{p_{i_1}} \\ \mathbf{p_{i_2}} \\ \vdots \\ \mathbf{p_{i_{t-1}}} \\ \mathbf{p_{i_t}} \end{pmatrix} \tag{8}$$

Exception:

- IF $m < t$ only the first m secrets are considered, the rest of keys are randomly generated $K = \left(K_1, K_2, ..., K_m\right)$
- IF $m > t$ the rebuilder construct a sub-matrix $C_s(m \times m)$, where the line corresponding to t node participating in public matrix of construction of shares C and $(m-t)$ line corresponding to public participating.

$$C_S = \begin{pmatrix} C_{i_1 1} & C_{i_1 2} & \cdots & C_{i_1 t} \\ C_{i_2 1} & C_{i_2 2} & \cdots & C_{i_2 t} \\ \vdots & \vdots & \ddots & \vdots \\ C_{i_t 1} & C_{i_t 2} & \cdots & C_{i_t t} \\ C_{n+1,1} & C_{n+1,2} & \cdots & C_{n+1,m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n+m-t,1} & C_{n+m-t,2} & \cdots & C_{n+m-t,m} \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha_{i1}+\beta_1} & \frac{1}{\alpha_{i1}+\beta_2} & \cdots & \frac{1}{\alpha_{i1}+\beta_t} \\ \frac{1}{\alpha_{i2}+\beta_1} & \frac{1}{\alpha_{i2}+\beta_2} & \cdots & \frac{1}{\alpha_{i2}+\beta_t} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{it}+\beta_1} & \frac{1}{\alpha_{it}+\beta_2} & \cdots & \frac{1}{\alpha_{it}+\beta_t} \\ \frac{1}{\alpha_{n+1}+\beta_1} & \frac{1}{\alpha_{n+1}+\beta_2} & \cdots & \frac{1}{\alpha_{n+1}+\beta_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_{n+m-t}+\beta_1} & \frac{1}{\alpha_{n+m-t}+\beta_2} & \cdots & \frac{1}{\alpha_{n+m-t}+\beta_m} \end{pmatrix} \tag{9}$$

Such as $\left(p_{n+1}, p_{n+2} \ldots\ldots, p_{n+m-t}\right)$ is the public share.

Verifiability:

The proposed scheme allows verifying the correctness of each share; the rebuilder has the possibility to check that the node put a valid share as follows:

$$Y_i = \prod_{j=1}^{t}\left(V_j\right)^{C_{ij}} \tag{10}$$

According to public value V_i the equation 10 will be:

$$Y_i = \prod_{j=1}^{t}\left(V_j\right)^{C_{ij}} = \prod_{j=1}^{t}\left(g^{k_j}\right)^{C_{ij}} = \prod_{j=1}^{t}\left(g^{K_j.C_{ij}}\right)$$

$$= \left(g^{K_1.C_{i1}}\right).\left(g^{K_2.C_{i2}}\right).\left(g^{K_3.C_{i3}}\right)\ldots\left(g^{K_t.C_{it}}\right)$$

$$= g^{S_1.C_{i1}+S_2.C_{i2}+....+S_t.C_{it}} = g^{\sum_{j=1}^{t} C_{ij}S_j} = g^{p_i}$$

Since the discrete exponentiation function is bijective over $GF(2^r)$, if g is a generator.

Example: we have 3 key to share K = (10, 20, 30) over $GF(2^8)$ so that $t=3$. In order to define an overdetermined system admitting S as a unique solution, we first define the number of equations n (that can be any desired value): let's choose $n=7$. Further, we generate two random sequences $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7)$ and $\beta = (\beta_1, \beta_2, \beta_3)$ over $GF(2^8)$ in order to construct a Cauchy matrix. Let us choose α = (15, 3, 85, 117, 200, 187, 98) and β = (31, 18, 179). According to equation (3), the corresponding Cauchy matrix is defined by:

$$C = \begin{pmatrix} 216 & 131 & 149 \\ 160 & 114 & 135 \\ 43 & 4 & 226 \\ 239 & 77 & 163 \\ 214 & 118 & 187 \\ 52 & 59 & 173 \\ 205 & 40 & 113 \end{pmatrix}$$

The coefficient's vector B is computed according to equation (4) as the following:

$$B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = C.S^T = \begin{pmatrix} 216 & 131 & 149 \\ 160 & 114 & 135 \\ 43 & 4 & 226 \\ 239 & 77 & 163 \\ 214 & 118 & 187 \\ 52 & 59 & 173 \\ 205 & 40 & 113 \end{pmatrix} . \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix} = \begin{pmatrix} 237 \\ 115 \\ 240 \\ 27 \\ 202 \\ 100 \\ 108 \end{pmatrix}$$

IF $X = (x_1, x_2, x_3)$ is a vector of variables over $GF(2^8)$, then the targeted solvable over-determined system is defined by $C \cdot X^T = B$, and can be presented as the following:

$$\begin{cases} 216x_1 & + & 131x_2 & + & 149x_3 & = & 237 \\ 160x_1 & + & 114x_2 & + & 135x_3 & = & 115 \\ 43x_1 & + & 4x_2 & + & 226x_3 & = & 240 \\ 239x_1 & + & 77x_2 & + & 163x_3 & = & 27 \\ 214x_1 & + & 118x_2 & + & 187x_3 & = & 202 \\ 52x_1 & + & 59x_2 & + & 173x_3 & = & 100 \\ 205x_1 & + & 40x_2 & + & 113x_3 & = & 108 \end{cases}$$

The solution S of the system in equation (8) can easily be recovered by inverting any 3×3 sub-matrix of C. The inverse is computed over $GF(2^8)$ and the result is multiplied by B to get S. For example, the following sub-matrices:

$$C_1 = \begin{pmatrix} 216 & 131 & 149 \\ 160 & 114 & 135 \\ 43 & 4 & 226 \end{pmatrix} C_2 = \begin{pmatrix} 216 & 131 & 149 \\ 43 & 4 & 226 \\ 205 & 40 & 113 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} 160 & 114 & 135 \\ 214 & 118 & 187 \\ 52 & 59 & 173 \end{pmatrix} C_4 = \begin{pmatrix} 43 & 4 & 226 \\ 239 & 77 & 163 \\ 205 & 40 & 113 \end{pmatrix}$$

Admit the following inverses over $GF(2^8)$:

$$C_1^{-1} = \begin{pmatrix} 167 & 135 & 91 \\ 9 & 169 & 146 \\ 228 & 250 & 176 \end{pmatrix}, C_2^{-1} = \begin{pmatrix} 42 & 229 & 32 \\ 86 & 78 & 204 \\ 18 & 100 & 203 \end{pmatrix}$$

$$C_3^{-1} = \begin{pmatrix} 229 & 52 & 181 \\ 117 & 164 & 161 \\ 97 & 119 & 204 \end{pmatrix}, C_4^{-1} = \begin{pmatrix} 86 & 79 & 193 \\ 85 & 242 & 243 \\ 35 & 43 & 120 \end{pmatrix}$$

Solution S is recovered by multiplying any inverse C_i^{-1} with the corresponding coefficient's sub-vector B_i from B. Each sub-vector B_i has a size of $1xt$, and contains values from B corresponding to selected rows defining the matrix C_i. Hence, solution of the above system can be obtained using any of following computations:

$$S = C_1^{-1}.B_1 = \begin{pmatrix} 167 & 135 & 91 \\ 9 & 169 & 146 \\ 228 & 250 & 176 \end{pmatrix} . \begin{pmatrix} 237 \\ 115 \\ 240 \end{pmatrix} = \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix}$$

$$S = C_2^{-1}.B_2 = \begin{pmatrix} 42 & 229 & 32 \\ 86 & 78 & 204 \\ 18 & 100 & 203 \end{pmatrix} . \begin{pmatrix} 237 \\ 240 \\ 108 \end{pmatrix} = \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix}$$

$$S = C_3^{-1}.B_3 = \begin{pmatrix} 229 & 52 & 181 \\ 117 & 164 & 161 \\ 97 & 119 & 204 \end{pmatrix} . \begin{pmatrix} 115 \\ 202 \\ 100 \end{pmatrix} = \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix}$$

$$S = C_4^{-1}.B_4 = \begin{pmatrix} 86 & 79 & 193 \\ 85 & 242 & 243 \\ 35 & 43 & 120 \end{pmatrix} . \begin{pmatrix} 240 \\ 27 \\ 108 \end{pmatrix} = \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix}$$

SECURITY AND PERFORMANCE ANALYSIS

Security analysis

We conduct security analysis of the proposed scheme by proving the following theorems.

Theorem 1. Any t or more participants can reconstruct m secrets $K = (K_1, K_2, ..., K_m)$.
Proof: (see reconstruction phase).

Theorem 2. Any collaboration of t–1 shares $P = (p_1, p_2, ..., p_{t-1})$ cannot allow reconstructing the set of key K.

Proof: The obtained system is underdetermined. Hence, it admits an infinite set of solutions each with same probability (uniformly distributed).

- $m < t$: They construct an underdetermined system of equations C_s of *t-1* equation and *t* variable that has infinity of solutions.
- $m > t$: They construct an underdetermined system of equations C_s of *m-1* equation and *t* variable that has infinity of solutions.

Theorem 3. The dealer's secret information *K* cannot be obtained from public information *C*.
Proof: An attacker may try to recover a secret k_i form the public corresponding value V_i. Since V_i= g^{ki}, recovering *Ki* requires the resolution of the Discrete Logarithm Problem over $GF(2^r)$. It is known that for high value of *r*, the problem is difficult and cannot be solved in reasonable time. Consequently, the secrets cannot be recovered from the public information.

Theorem 4. The shares provided by participants during the reconstruction phase can be verified so that cheaters are identified.
Proof: A participant P_i can give his secret share to a cheater P_i* in order to construct a group of *t* valid shares with a group of (*t*–2) other valid participant. This will end up with a set of *t* valid shares $P = (p_1, p_2 \ldots, p_i, p_i \ldots, p_t)$, while the share P_i is repeated. Such attack fails to pass the verification step since the participant P_i* is rejected when computing V_i* using wrong coefficients from *C*. The only row's index that can be used by participant is the one used initially (during sharing phase) to compute his share, and this index cannot be used twice during recovery phase.

Performances analysis

1. The secret space has the same size as the share space so our scheme is ideal.
2. In each secret sharing session several secrets can be shared, so our scheme is a multi-secret sharing scheme.
3. In our scheme each participant can verify the validity of shares of the other participants and her/himself in the verification phase. Hence, the cheating can be detected. Therefore, it is a verifiable secret sharing scheme and the participants can verify that they can recover unique secrets.
4. Sharing and reconstruction are performed within linear computational complexity, no need for modular exponentiation, primes generation or modular reduction. Since shares generation and secrets reconstruction use linear matrices operations, the cost of both phases is linear with respect to *m*, *t* and *n*. In addition, since operations are performed over a Galois field $GF(2^r)$, addition uses a simple fast *Xor* operator, while multiplication is implemented using optimized and extremely fast algorithms, especially if using pre-computed look-up tables providing extremely optimized performances. With respect to existing schemes, no modular exponentiation or polynomial computation are needed. Table 1 gives performance comparison between the proposed scheme and some popular existing ones in terms of computational complexity and security.

Table 1. Security and performances comparison

Property	CRT-based scheme [14]	Matrices projection [18]	LFSR-based scheme [21]	YCH scheme [23]	Proposed scheme
Mathematical basics	CRT Theorem	Matrices projection	Polynomial interpolation	Polynomial interpolation	overdetermined eq. systems
Sharing's complexity	$O(n.\log_2(n))$	$O(n^2.k^2/m^2)$	$O(n^2)$	$O(n^2)$	$O(n)$
Reconstruction's complexity	$O(m.\log_2(m))$	$O(m^2)$	$O(m^2)$	$O(m^2)$	$O(m)$
Secrecy Assumption	Unconditional	Unconditional	Unconditional	Unconditional	Unconditional
Verifiability Assumption	–	–	RSA	–	DLP
Robustness and ideality	Yes	Yes	Yes	Yes	Yes
Scalability for large secret	No	No	No	No	Yes
Additional crypto. tools	–	–	LFSR PKE	One-way fun.	–
Dynamic sharing	No	No	Yes	No	Yes

EVALUATION

Networks model

We use a cluster-based network model (Fig. 1). It is represented by a set of nodes. Each group is constituted by a cluster head (CH) and gateway nodes (GW) that manages the communication with neighboring groups and his member nodes.

The private key of the CA is distributed about all nodes using our threshold secret sharing scheme (*t*, *n*). The certificate Issuance and renewal is summarized in the following steps, when the node Ni wants to authenticate a certificate

- Step 1: N_i contacts CH and sent his certificate. CH verifies credential of N_i.
- Step 2: CH broadcast a request certificate message over *t* other nodes (shareholders).
- Step 3: *t* shareholders issue CH (partial key of CA).
- Step 4: CH reconstructs the CA secret key SCA.
- Step 5: CH signs and validates the certificate of N_i.

Performance analysis

To see the pertinence of our approach and to measure the effect that will cause the implementation of our MSSS for distribute de CA in cluster-based network, we performed several simulations with a variable number of nodes. We used CBRP routing protocol and NS2 as network simulator with the following parameters.

Table 2. Parameter of simulation

Parameter	Value
Simulation area	800×800
Protocol	CBRP
Number of nodes	10...50
Radio range	250
Mobility model	Random way point
Simulation time	200

We defined two performance metrics to evaluate the effectiveness of our proposed scheme. These parameters are: the average delivery delay of a certificate representing the time elapsed since the request to delivery of a certificate, and the CDF (Certificate Delivery Fraction) which represents the percentage of certificates issued. We also measure the influence of the threshold parameters (k and n) in order to observe the performance of our scheme.

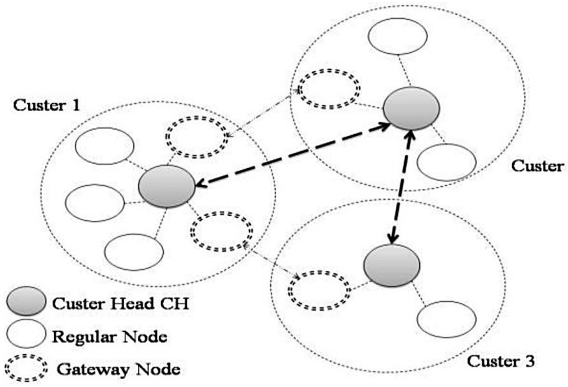

Fig. 1. The networks model

Results

Figure 2 shows the average certificates delivery fraction for varying the nodes from 10 to 50, which represents the percentage of issued certificates by emitted certificates. We note that the parameter k has an influence on the CDF, where *t* = 1 this means that the CA node needs only their private key to sign the certificate of issuer node. When *t* increases the CDF decreases because the CA node waits *t* partial response to build the private key of CA, and if he receive less then *t* response the certificate issuance cannot establish and the node cannot authenticate.

Figure 3 shows the average delay between the request and reception of a certificate. We note that the parameter *k* has influence on the time of delivery of a certificate. We show when *t* increases the delay increase. Because the node CA waits more time to receive the partial response. In the ideal case where *t* = 1 the delay of issuance of certificate is constant because there is no partial request, the CA node uses only his private key to a signed certificate.

CONCLUSIONS

In this work, we propose a novel ideal multi secrets threshold sharing scheme. Using overdetermined systems of linear equations over finite Galois fields $GF(2^r)$, the scheme provides linear sharing and reconstruction complexities with an unconditional security. We integrate our method in cluster-based architecture for MANET to distribute the certificate authority role. The number of shares can increase or decrease dynamically during the lifetime of the system. With the process of verifiability we detect the cheater so it makes

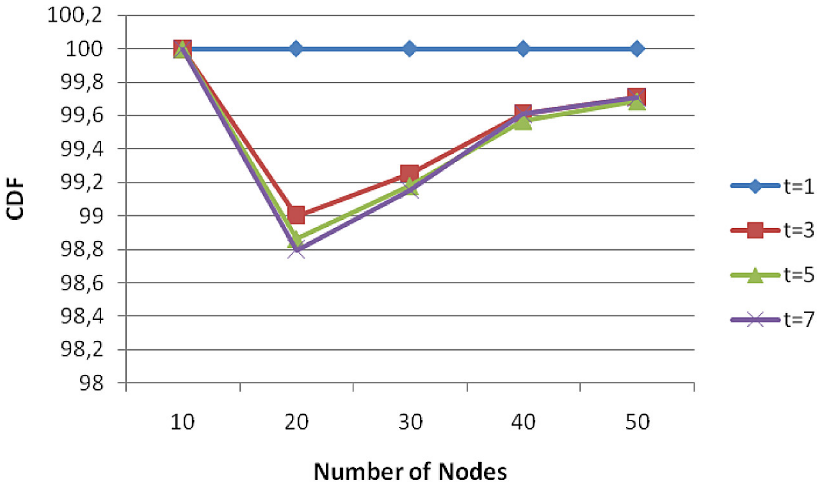

Fig. 2. The certificate delivery fraction in term of node

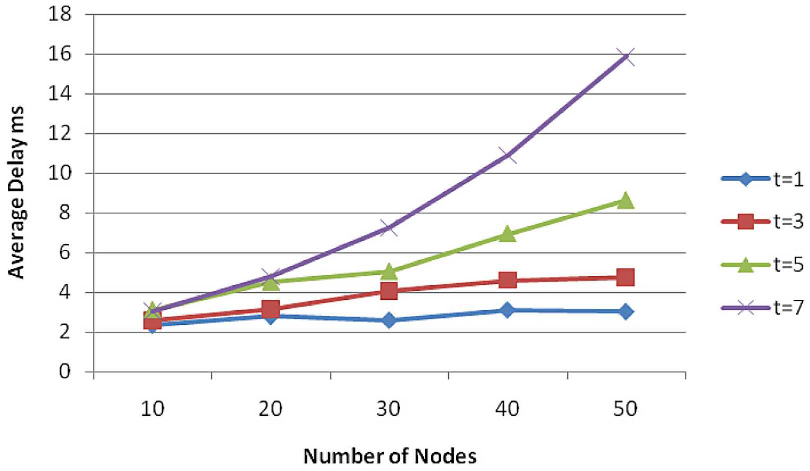

Fig. 3. Average delay of issued certificate in term of node

our system more robust. Furthermore, the security is unconditionally with discrete logarithm problem. Finally, the correctness and security of proposed scheme are proved and evaluated with some parameter in a network simulator.

REFERENCES

1. Khurana S., Gupta N. and Aneja N. Minimum exposed path to the attack (MEPA) in mobile ad hoc network (MANET). In: Networking, ICN '07. Sixth International Conference on, 2007.
2. Hongmei D., Mukherjee A. and Agrawal D.P. Threshold and identity-based key management and authentication for wireless ad hoc networks. In: Information Technology: Coding and Computing. Proceedings ITCC 2004. International Conference on, 2004.
3. Dong Y. et al. Providing distributed certificate authority service in cluster-based mobile ad hoc networks. Comput. Communic., 30(11–12), 2007, 2442–2452.
4. Lung-Chung L. and Ru-Sheng L. Securing cluster-based ad hoc networks with distributed authorities. Wireless Communications, IEEE Transactions on, 9(10), 2010, 3072–3081.
5. Zhou L., Schneider F.B. and Renesse R.V. Coca: A secure distributed online certification authority. ACM Trans. Comput. Syst., 20(4), 2002, 329–368.
6. Seung Y. and Robin K. Moca: Mobile certificate authority for wireless ad hoc networks. In: The second anunual PKI research workshop (PKI '03). Gaithersburg 2003.
7. Wu B. et al. Secure and efficient key management in mobile ad hoc networks. Journal of Network and Computer Applications, 30(3), 2007, 937–954.
8. Xianyong M. and Yangmin L. A verifiable dynamic threshold key management scheme based on bilinear pairing without a trusted party in mobile ad hoc network. In: Automation and Logistics (ICAL), IEEE International Conference on. 2012.
9. Neha G. and Manish S. Securing routing protocol by distributed key management and threshold cryptography in mobile ad hoc network. Interna-

tional Journal of Advanced Computer Research, 3(9), 2013.

10. Dahshan H. and Irvine J. A trust based threshold cryptography key management for mobile ad hoc networks. In: Vehicular Technology Conference Fall (VTC 2009-Fall), IEEE 70th, 2009.
11. Shamir A. How to share a secret. Commun. ACM, 22(11), 1979, 612–613.
12. Blakley G.R. Safeguarding cryptographic keys. in Proceedings of the National Computer Conference. 1979.
13. Asmuth C. and Bloom J. A modular approach to key safeguarding. IEEE Transactions on Information Theory, 29(2), 1983, 208–210.
14. V, S.R.Y. and Bhagvati C. CRT based threshold multi secret sharing scheme. International Journal of Network Security, 16(4), 2014, 249–255.
15. Schoenberg I.J. On Hermite-Birkhoff interpolation. Journal of Mathematical Analysis and Applications, 16(3), 1966, 538–543.
16. Zhang Y., Liu Z. and Huang G. Sure interpolation and its application to hierarchical threshold secret sharing scheme. In: Computer Science and Computational Technology. ISCSCT '08. International Symposium on, 2008.
17. Yang G. et al. (k, n) threshold secret sharing scheme based on n-dimensional cube. In: Z. Zhong (Ed.) Proceedings of the International Conference on Information Engineering and Applications (IEA 2012). Springer, London 2013, 611–618.
18. Bai L. and Zou X. A proactive secret sharing scheme in matrix projection method. Int. J. Secur. Netw., 4(4), 2009, 201–209.
19. Tassa T. and Villar J. On proper secrets, (t, k)-bases and linear codes. Designs, Codes and Cryptography, 52(2), 2009, 129–154.
20. Wang S.-J., Tsai Y.-R. and Shen C.-C. Verifiable threshold scheme in multi-secret sharing distributions upon extensions of ECC. Wireless Personal Communications, 56(1), 2011, 173–182.
21. Hu C., Liao X., and Cheng X. Verifiable multi-secret sharing based on LFSR sequences. Theor. Comput. Sci., 445, 2012, 52–62.
22. Hung-Yu C., Jinn-Ke J.A.N. and Yuh-Min T., A practical (t, n) multi-secret sharing scheme. Communic. and Comput. Sci., 83(12), 2000, 2762–2765.
23. Yang C.-C., Chang T.-Y. and Hwang M.-S. A (t, n) multi-secret sharing scheme. Applied Mathematics and Computation, 151(2), 2004, 483–490.
24. Tentu A. and Rao A. Efficient verifiable multi-secret sharing based on Y.C.H scheme. In: Z. Kotulski, B. Księżopolski and K. Mazur (Eds.) Cryptography and Security Systems. Springer, Berlin–Heidelberg 2014, 100–109.