© 2015 IJSRSET | Volume 1 | Issue 1 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Certificateless Public Key cryptosystems For Mobile Ad hoc Networks

Shabnam Kasra-Kermanshahi*[1], Mazleena Salleh[2]
*[1] Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia

ABSTRACT

Due to importance of security in many critical applications in Mobile Adhoc Networks (MANETs) and the limitation of the resources in mobile devices, it is important to have secure lightweight cryptosystem. The easier key management and less overhead of transmitting processes make Public Key Cryptosystems (PKC) suitable for MANETs. Obviously, the main issue regarding to the use of PKC is to ensure about the authenticity of users' public key. However, complex management of Public Key Infrastructure in Traditional PKC and Key Escrow problem of Identity Based ones led to emphasize on the use of Certificateless PKC. In this research, beside of a Certificateless Public Key management scheme, a public key authentication schemes named IDRSA and two improved version of that named ClessRSA and EIDRSA have been investigated. In order to compare mentioned works, a standard format is given to investigate mentioned schemes based on the same notations and assumptions. Beside of mathematical comparison, the growth rate of computational expense for the particular part of mentioned schemes as a function of the number of requests is visualized. The results indicate that EIDRSA schemes has lower computational expense in compare with other existing ones because of eliminating Bilinear Pairing operation.

Keywords: Public Key Cryptosystems, Certificateless, MANET, Efficiency

I. INTRODUCTION

The widely usage of mobile applications recently led to developing a large variety of security mechanisms in mobile networks especially in those ones without any fixed infrastructure. The reason is that traditional networks based on pre-existing fixed infrastructure cannot support many modern applications. Beside of these attributes of such networks, it is worth noting that the use of networks without any fixed infrastructure sometimes leads to installations without the interference of administrators or managers. Mobile Ad hoc Network (MANET) is one of the instances of this category of networks, which can support mentioned requirements.

MANETs are wireless networks consisted of mobile free nodes that can move anywhere at any time without the need to any fixed infrastructure or any centralized administration to manage or organize mentioned nodes. More accurately, existing nodes cooperate with each other to carry the responsibility of managing the network requirements. Scalability is one of the most significant attributes of mobile ad hoc networks. This term refers to the ability of managing all participating nodes which are going to join and leave the network quickly. This attribute is the basis of this fact that existing nodes must rely on each other to play the role of routers or switches instead of using central ones.

There are many reasons that convinced a large variety of network developers to use MANETs in many modern applications especially crucial ones such as battlefield missions, rescue operations, etc. in these category of applications, a large group of communicating mobile nodes move to wherever at any time without supporting by any central administration. Since, the use of any fixed infrastructure is impractical in such a situation, the use of mobile ad hoc networks is the best option in such environments [1]. Beside of what mentioned above, MANETs can be appeared in the form of other applications such as search and rescue operations [2-4], transportation vehicular applications to avoid accidents and traffic jams [5,6], etc.

In continue to what noted above, it is necessary to mention that the self-organized nature of such

environments made MANETs vulnerable against many security threats. As a result, providing security requirements in MANETs is one of the most interesting challenges in such a network. Moreover, there are many security issues in mobile ad hoc networks [7]. It can be claimed that a subset of these issues are trying to prevent network nodes from external attacks by the use of some mechanisms such as authentication of existing network nodes [8-10] or trough proposing secure routing protocols [11-14]. Although it seems that such preventing mechanisms are useful to securing MANETs, they are not perfect enough. More precisely, there must be other classes of security mechanisms to detect occurred attacks or resist against possible security problems. To reach this goal, on one hand a subset of researches has tried to propose a prevention mechanism by the use of Intrusion Detection Systems (IDS) [15-17]. On the other hand, the use of appropriate cryptographic schemes in MANETs attracted other researchers recently. It is necessary to note that the focus of this research is based on the second group. In this way, eliminating the need to Public Key Infrastructure (PKI) made the use of Identity Based cryptographic schemes one of the most popular cryptosystems in such networks. However, in an Identity-based cryptosystem each entity must collect its private key from PKG hence PKG can eavesdrop the messages or impersonate entities. This inherent problem in Identity-based cryptosystems called "key escrow". This problem limits the use of Identity-based cryptosystems to closed organizations [18]. Early solutions focus on utilizing more key pairs, using threshold, and considering expiry date for the master key. However, they have some drawbacks that make them unsuitable for MANETs such as too much overhead to the network, more computation /communication for nodes which are resource constrained devices [19].

In 2003, the suggested public key cryptosystem by Al-Riyami and Paterson in [20] named "Certificateless public key" could overcome the problems of Traditional and Identity-Based cryptosystems. This new cryptosystem utilizes a trusted third party known as called Key Generator Center (KGC) who generates partial keys for the involving entities. Each entity can generate its own private key by the use of received partial value from KGC and a confidential value chosen by the entity therefore there is no key escrow problem [18]. In this paper, several Certificateless public key schemes have been investigated based on the same notations and assumptions [21-24]. Moreover we made a comprehensive comparison over the computational cost of the considered schemes.

The rest of this paper is organized as followed. Next section consists of required notation and assumption for the rest of this paper. In the third section, several Certificateless Public Key cryptosystems have been investigated in detail. Section 4 is dedicated to the comparison of computational costs of considered schemes. Finally the last section draws the conclusion of this paper.

II. METHODS AND MATERIAL

Notation and Assumptions

The suggested notation and assumption for the rest of this paper are as followed.

- G_1: an additive algebraic group
- G_2: an additive algebraic group
- G_T: a multiplicative algebraic group
- q: the order of mentioned groups
- P: an element of G_2
- $\hat{e}$: a bilinear pairing over mentioned algebraic groups $\hat{e}: G_1 \times G_2 \rightarrow G_T$
- n: a positive integer number that determines the number of bits of the two components of the RSA public key (e and N)
- H_1 : one-way collision-free hash functions $H_1: \{0,1\}^* \rightarrow G^*$
- H_2: one-way collision-free hash functions $H_2: G \rightarrow \{0,1\}^n$
- H_3: one-way collision-free hash functions $H_3: \{0,1\}^* \rightarrow \{0,1\}^n$

Certificateless Public Key Cryptosystems in MANETS

The purpose of this section is to probe into four existing Certificateless public key cryptosystem in MANETs from cryptographic functionality viewpoint named IDRSA, C_{less}RSA, EIDRSA, and CLPKKM [21-24].

A. A review of IDRSA scheme

The main objective of this subsection is to investigate the IDRSA protocol [21]. IDRSA tries to guarantee that the public keys are just accessible by the trusted entities to make the protocol protected against RSA cryptanalysis attacks. To reach this goal, it is assumed that any user is a member of a logical group of users named coalition. To obtain the public key of other side party, existing users must ask the required public key

from the coalition that the considered user if a member of. Based on these assumptions, the rest of this subsection investigates the phases of IDRSA and the correctness of this protocol logically.
It can be claimed that the core part of IDRSA scheme consists of three main phases that we named them Setup, Node Initialization, and Public-key Obtaining Process. In continue, theses main phases are reviewed briefly.

Setup: In this phase, a trusted third party generates public parameters of the cryptosystem $Params: < G_1, G_2, G_T, q, P, \hat{e}, n, H_1, H_2, H_3 >$ after taking the required security parameters. The elements of Params have been explained in the second section.

Node Initialization: The basis of this phase is to generate a subset of public and private parameters for existing users and coalitions, beside of publishing a subset of public ones. The public parameters of mentioned entities are named Identity-key, General-key and public-key. Here, Identity-key of any user is computable by all other existing ones, while General-key and Public-key must be generated by the owner of them. To support freshness, Identity-key of the user or coalition "*i*" (which possess ID_i) would be created as below:

$$Q_i = H_1(ID_i \parallel time)$$

Here, the entity who possess ID_i randomly chooses the prime number "e_i"as a randomly chosen element of $\mathbb{Z}_q^*$ or $e_i \in_r \mathbb{Z}_q^*$. Then, each node such as node "*i*" runs the RSA key generation algorithm to generate the parameterse_i, d_i, N_i . Such as traditional RSA scheme, d_i and $< e_i, N_i >$ are the private-key and public-key of mentioned entity, respectively. After that, mentioned user or coalition publishes the value $P_i = (d_i . P)$ as the General-key.

Public key obtaining process: In the last predicted phase of *IDRSA*, each user can refer to the considered coalition that the other party is a member of, to take the required Public-key securely. In the sake of simplicity, assume that node *A*needs the Public-key of node B, and sends the request to the desirable coalition named $IDRSA_i$. Then, the "Public key obtaining process" will be done by performing followed three steps:

Step1: $A \rightarrow IDRSA_i: P_A, ID_B$

In this step, the node *A* introduces himself by sending P_A, then requests to obtain the Public-key of the node $B(e_B \text{and} N_B)$ by sending ID_B to the $IDRSA_i$ coalition.

Step2: $IDRSA_i \rightarrow A: < U, C, W, Y >$

In this step, the coalition $IDRSA_i$ first of all checks if the node *B* is in the list or not. Then it will send mentioned parameters to the node *A*. here, the mentioned four parameters are as below:
$U = P_i$, $C = e_B \oplus H_2(g_i)$ that g_i is equal to $\hat{e}(Q_A, P_A)^{d_i} \times \hat{e}(d_i Q_i, P_A)$, $W = e_B . P$ and$Y = N_B \oplus H_3(e_B)$.

Step3: Public key extraction by A

In this step, the node *A* tries to extract the requested Public-key of the node *B* (e_B and N_B) and verify its authenticity by performing followed computations:

First of all, *A* computes$g_A = \hat{e}(d_A Q_A, P_i) \times \hat{e}(Q_i, P_i)^{d_A}$, then computes$e_B = C \oplus H_2(g_A)$. Obviously, the result of g_A must be equal to g_i. After that, the node *A* computes $N_B = Y \oplus H_3(e_B)$. Finally, to verify the authenticity of the Public-key of the node *B* ($e_B \text{and} N_B$), the node *A* checks if ($W = e_B . P$) to decide whether accept or reject the calculated Public-key pair of the node *B*.

1) Investigating the correctness of IDRSA

To investigate *IDRSA* logically, it must be proved that the user *A* and the coalition $IDRSA_i$ will achieve the same value by computing $g_A \text{and} g_i$, respectively. The calculations below, can show that the result of both computations is the same value

$$\begin{gathered} \hat{e}(Q_A + Q_i, P)^{d_i d_A}. \\ g_A = \hat{e}(d_A Q_A, P_i) \times \hat{e}(Q_i, P_i)^{d_A} \\ = \hat{e}(Q_A, P)^{d_i d_A} \times \hat{e}(Q_i, P)^{d_i d_A} \\ = \hat{e}(Q_A + Q_i, P)^{d_i d_A} \end{gathered}$$

$$\begin{gathered} g_i = \hat{e}(Q_A, P_A)^{d_i} \times \hat{e}(d_i Q_i, P_A) \\ = \hat{e}(Q_A, P)^{d_i d_A} \times \hat{e}(Q_i, P)^{d_i d_A} \\ = \hat{e}(Q_A + Q_i, P)^{d_i d_A} \end{gathered}$$

As a result, it can be concluded that *IDRSA* is logically a correct scheme.

B. Review of C_{less}RSA Scheme

The $C_{less}RSA$ scheme is an improved version of IDRSA, from computational efficiency perspective. The outline of current subsection is to investigate this scheme in detail [22]. Since the Setup and Node Initialization phases of $C_{less}RSA$ scheme is similar to IDRSA, only the Public-key Obtaining Process is described in this subsection. Note that in the $C_{less}RSA$ scheme computational expenses are more lightweight than IDRSA. In more detail, the Public-key Obtaining Process is as followed:

Public key obtaining process in C_{less}RSA: Similar to IDRSA, in the last phase each user refers to the considered coalition and requests for the Public-key of the other side party. If roughly speaking, we assume that node A needs the public key of node B, and sends the request to the desirable coalition named $C_{less}RSA_i$. Then, the "Public key obtaining process" will be done by performing three steps below:

Step1; $A \rightarrow C_{less}RSA_i: P_A, ID_B$

In this step, the public parameter $P_A$introduces the node A as the one who issued his request. Moreover, the public identity ID_Bdetermines the other party who his public key ($e_B$and N_B) is requested by A.

Step2;$C_{less}RSA_i \rightarrow A: < U, C, W, Y >$

In this step, the coalition $C_{less}RSA_i$ will send back the tuple $< U, C, W, Y >$ to node A. here, the mentioned four parameters are as below:
$U = P_i$, $C = e_B \oplus H_2(g_i)$ that g_i is equal to $g_i = \hat{e}(d_i Q_A, P_A)$, $W = e_B.P$ and$Y = N_B \oplus H_3(e_B)$.

Step3; Public key extraction by A

In this step, the node A extracts the public key of the node B (e_B and N_B) and verifies its authenticity by performing followed computations:

At first, A computes$g_A = \hat{e}(d_A Q_A, P_i)$, then computes $e_B = C \oplus H_2(g_A)$. Clearly, the result of g_A must be the same asg_i. In continue, node A computes $N_B = Y \oplus H_3(e_B)$. Finally, to verify the authenticity of the public key of the node B ($e_B$andN_B), the node A checks if ($W = e_B.P$) to decide whether accept or reject the calculated public key pair of the node B.

1) Investigating the correctness of C_{less}RSA

To investigate logical functionality of $C_{less}RSA$, we show that the user A and the coalition $C_{less}RSA_i$ will achieve the same value by computing g_A and g_i , respectively. The two calculations below, prove that the result of both computations is the same value $\hat{e}(Q_A, P)^{d_i d_A}$

$$g_A = \hat{e}(d_A Q_A, P_i) = \hat{e}(Q_A, P)^{d_i d_A}$$

$$g_i = \hat{e}(d_i Q_A, P_A) = \hat{e}(Q_A, P)^{d_i d_A}$$

As a result, the functionality of $C_{less}RSA$ is logically correct.

C. A review of EIDRSA Scheme

This section presents a brief review of Certificateless authenticating public key protocol named EIDRSA [23]. This scheme is designed on the basis of IDRSA scheme, however the EIDRSA uses Elliptic Curve based Algebraic Groups instead of multiplicative ones over Finite Fields as the output of Bilinear Pairings that leads to lower computational cost in Public-key Obtaining Process phase. EIDRSA scheme is constructed based on three phases named Setup, Node Initialization, and Public-key Obtaining Process as followed.

Setup: The public output of our scheme, Params, includes following items:

$$Params: < G, q, P, n, H_1, H_2, H_3 >$$

Here, G is a cyclic Elliptic Curve group with order q and the generator P. The integer number n is the same as n in IDRSA protocol. In addition, $H_1: \{0,1\}^* \rightarrow G^*$, $H_2: G \rightarrow \{0,1\}^n$ and $H_3: \{0,1\}^* \rightarrow \{0,1\}^n$ are three one-way collision-free hash functions.

Node Initialization: Similar to the ID-RSA scheme, the entities can be user or coalition logically. Here, the public and private parameters of mentioned entities are the same as what introduced in ID-RSA. In addition, each entity such as the entity who possesses ID_i generates the parameter $E_i = e_i P_i$.

Public key obtaining process: In this phase, the scenario is similar to what proposed in ID-RSA except

that the details of the steps. Here, these steps are as followed:

Step1: $A \rightarrow Coalition_i: E_A, P_A, ID_B$
In this step, the second input introduces the node *A* as the entity who sent the request and the third one refers to the identity of the entity who his public key is requested.

Step2:$Coalition_i \rightarrow A: < E_i, U, C, W, Y >$
In this step, the inputs U, C, W and Y are as followed. $U = P_i$, $C = e_B \oplus H_2(g_i)$ that g_i is equal to $g_i = d_i(E_A + e_i P_A)$ $W = e_B . P$ and$Y = N_B \oplus H_3(e_B)$.

Step3: In this step, the node *A* must be able to extract the public key of *B* (which are e_B and N_B) and verify its authenticity as followed:

First of all, A computes $g_A = d_A(E_i + e_A P_i)$, then computes $e_B = C \oplus H_2(g_A)$. Clearly, the result of g_A and g_i must be the same. In addition, the entity A computes $N_B = Y \oplus H_3(e_B)$. To verify authenticity of the obtained public key, the entity *A* investigates the equality of $(W = e_B . P)$ to decide whether accept or reject the obtained public key of *B*.

1) Investigating the correctness of EIDRSA

Beside of what mentioned above, it is necessary to prove that the computed values of g_A and g_i are the same. The two equalities below will lead to this result:

$$\begin{aligned} g_A &= d_A(E_i + e_A P_i) \\ &= d_A e_i P_i + d_A e_A P_i \\ &= (d_A e_i d_i)P + (d_A e_A d_i)P \end{aligned}$$

$$\begin{aligned} g_i &= d_i(E_A + e_i P_A) \\ &= d_i e_A P_A + d_i e_i P_A \\ &= (d_i e_A d_A)P + (d_i e_i d_A)P \end{aligned}$$

As a result, the functionality of our proposed protocol is logically correct.

D. A review of CLPKKM scheme

This subsection reviews the *CLPKKM* protocol briefly [24]. Although *CLPKKM* is fundamentally in the category of public key cryptosystems, tries to guarantee the security of the scheme by the use of the idea of hybrid cryptosystems. In more detail, communicating parties try to share a secret whenever the system changes one of the broadcasted public parameters named Salt. To reach this goal, it is assumed that each user possesses a changeable private key in any stage that the cryptosystem changes the Salt public parameter beside of the fixed public and private key pairs. It can be claimed that *CLPKKM* scheme includes three main phases that we named them Setup, Node Initialization, and Shared-Secret computation. In continue, these phases are introduced in more detail.

Setup: In this phase, a trusted third party named KGC[1] takes the security parameters to generate followed confidential Master-key and publicly known parameters named Params.

$$Master - key: s \in_r Z_q^*$$
$$Params: < G_1, G_2, G_T, q, P, P_{pub}, \hat{e}, H, Salt_1 >$$

In the mentioned parameters above, $< G_1, + >$, $< G_2, + >$ and $< G_T, \times >$ are three groups and q is the order of them. Moreover, $P \in G_2$ and $P_{pub} = sP$. In addition, $\hat{e}: G_1 \times G_2 \rightarrow G_T$ is a bilinear pairing over mentioned algebraic groups and $Salt_1$is an integer pre-deployed public value that will be changed in any stage based on a simple formula that we will see later. Beside of these, $H: \{0,1\}^* \rightarrow {G_1}^*$is a random one-way collision-free hash function.

Node Initialization: The basis of Node Initialization phase is to generate public and private keys for existing users. Before introducing the details and structures of these keys, it is necessary to mention the assumptions of this phase. As it is noted before, each user possesses a changeable private key in any stage that the cryptosystem changes the Salt public parameter beside of the fixed one. In the rest of this research, it is assumed that the name of the fixed public key and private key of the user who possesses the identity ID_i is named PK_i and SK_i , respectively. Moreover, in the stage "j" the changeable private key of the user who possesses the identity $ID_i$is named $SK_{i,j}$.The value of this changeable private key is related to the value of the *Salt* value on that stage. As it is explained before, the first value of *Salt*, $Salt_1$, is predetermined as one of the parameters of Params. After that, in the other stages such as stage *j*, this value is publicly computable as followed:

$$Salt_j = \begin{cases} Salt_1 & j = 1 \\ Salt_{j-1} + 1 & j > 1 \end{cases}$$

[1] Key Generator Center

It is necessary to point out that in the stage j, the *KGC* publicly publishes the value $H_j = sH(Salt_j)$ to all existing users.

In continue to the assumptions above, the rest of this subsection introduces the mentioned public and private keys of an assumptive user such as the user who possesses the identity ID_i.First of all, this user refers to the *KGC* to take his partial private key, d_i.This partial private key is computable by *KGC* as below:

$$d_i = sQ_i$$

Here, $Q_i = H(ID_i)$ Then, mentioned user randomly chooses the value $x_i \in_r Z_q^*$. After that he can compute the valuesSK_i, PK_i and $SK_{i,j}$as follow:

$$SK_i = x_i sQ_i$$
$$PK_i = < u_i, v_i > = < x_i P_{pub}, x_i P >$$
$$SK_{i,j} = x_i H_j$$

In continue, the last phase of *CLPKKM* is investigated.

Shared-Secret computation: Since, *CLPKKM* tries to rely on hybrid cryptosystems to provide required security services, communicating entities can share a secret based on the number of stage that they are involving with. Assume that two users *A* and *B* who possess the identities$ID_A$and ID_B, respectively want to share a secret in the j stage. In this case, A and B must compute the same values $K_{AB,j}$and $K_{BA,j}$, respectively. These values are computable as following equations:

$$K_{AB,j} = \hat{e}(Q_B, u_B) \times \hat{e}(SK_A, P) \times \hat{e}(SK_{Aj}, v_B) K_{BA,j} = \hat{e}(Q_A, u_A) \times \hat{e}(SK_B, P) \times \hat{e}(SK_{Bj}, v_A)$$

Next section, logically investigates the correctness of the *CLPKKM* scheme.

1) Investigating the correctness of CLPKKM

To prove the correctness of CLPKKM scheme logically, the values $K_{AB,j}$ and $K_{BA,j}$ must be the same. Followed calculations can reach to this result:

$$K_{AB,j} = \hat{e}(Q_B, u_B) \times \hat{e}(SK_A, P) \times \hat{e}\left(SK_{Aj}, v_B\right) = \hat{e}\left(Q_B, x_B P_{pub}\right) \times \hat{e}(x_A sQ_A, P) \times \hat{e}\left(x_A H_j, x_B P\right) = \hat{e}(Q_B, P)^{Sx_B} \times \hat{e}(Q_A, P)^{Sx_A} \times \hat{e}(H_j, P)^{x_A x_B}$$

$$K_{BA,j} = \hat{e}(Q_A, u_A) \times \hat{e}(SK_B, P) \times \hat{e}\left(SK_{Bj}, v_A\right) = \hat{e}\left(Q_A, x_A P_{pub}\right) \times \hat{e}(x_B sQ_B, P) \times \hat{e}\left(x_B H_j, x_A P\right) = \hat{e}(Q_A, P)^{Sx_A} \times \hat{e}(Q_B, P)^{Sx_B} \times \hat{e}(H_j, P)^{x_A x_B}$$

As a result, it is proved that *CLPKKM* is logically a correct scheme.

III. RESULTS AND DISCUSSION

Efficiency Comparison

This section emphasizes on comparing the computational expense of the reviewed schemes in the previous section. We focused on the computational expense of g_A and g_i parts of "Public key obtaining process," in IDRSA, $C_{less}RSA$, *and EIDRSA* schemes and $K_{AB,j}$ and $K_{BA,j}$ parts of *CLPKKM* scheme which is the core of the difference between mentioned schemes. Then, computational expense of these parts are calculated and compared together. Moreover, the rate of growth of computational expense for mentioned parts are depicted in two separate diagrams. This comparison is based on assuming that $g_A$and g_i parts of $C_{less}RSA$ and *IDRSA* schemes and $K_{AB,j}$ and $K_{BA,j}$ parts of *CLPKKM* scheme are constructed by Type2 or Type3 Bilinear Pairings. It is worth to remind that EIDRSA do not require any pairing operation. The main reason is that the pairing operations are more expensive than modular exponentiation and scalar multiplication operations [18]. The TABLE I illustrates the expenses of operations (pairings, modular exponentiation and scalar multiplication) in Type2 and Type3 Bilinear Pairings based on the assumptions of the [18].The reason that we just emphasized on Type2 and Type3 Bilinear Pairings is that Type1 Bilinear Pairing is limited to obtain less than 80 bits security level, while the use of Type2 and Type3 Bilinear Pairings can lead to obtaining 128 bits or 256 bits security level [18].

TABLE I. Computational expense of group operations inType2 and Type3 Bilinear Pairings [18]

Group operation	Computational expense	
	Type2	Type3
Multiplication in G_1 (M_1)	1	1
Multiplication in G_2 (M_2)	45	3
Exponent in G_T (E_T)	3	3
Pairing (P)	21	20

Based on the Table 1, computational expense of g_A or g_i part in *IDRSA* is equal to "E_T+M_1+2P". Hence, the total computational cost for the considered parts in

this scheme is equal to 46 and 44 for Type2 and Type3 of Bilinear Pairings, respectively.

Followed by what mentioned in Table 1, computational expense of considered parts of "Public key obtaining process" in $C_{less}RSA$ is equal to "M_1+P" which means the expense of these parts in$C_{less}RSA$ scheme are 22 and 21 for Type2 and Type3 of Bilinear Pairings, respectively.

Furthermore, based on the Table 1, computational expense of g_A or g_i part in EIDRSA is equal to "2M_1" with one point addition. Since the computational cost of point addition is negligible, the total computational cost for this scheme is equal to 2 which is quite efficient in compare with all mentioned works.

Beside of the mentioned computations above, based on the Table 1, computational expense of $K_{AB,j}$ or $K_{BA,j}$ parts of *CLPKKM* scheme is equal to "3P" which means the expense of these parts in this scheme are 63 and 60 for Type2 and Type3 of Bilinear Pairings, respectively.

In order to have better understanding about the overall computational cost of the considered schemes in Type2 and Type3 of Bilinear Pairings, Figure 1 and Figure 2 depict the growth rate of computational expense for $g_A$or g_i parts of "Public key obtaining process" of the schemes IDRSA, $C_{less}RSA$, EIDRSA and the growth rate of computational expense for $K_{AB,j}$ or $K_{BA,j}$ parts of CLPKKM scheme as a function of the number of requests.

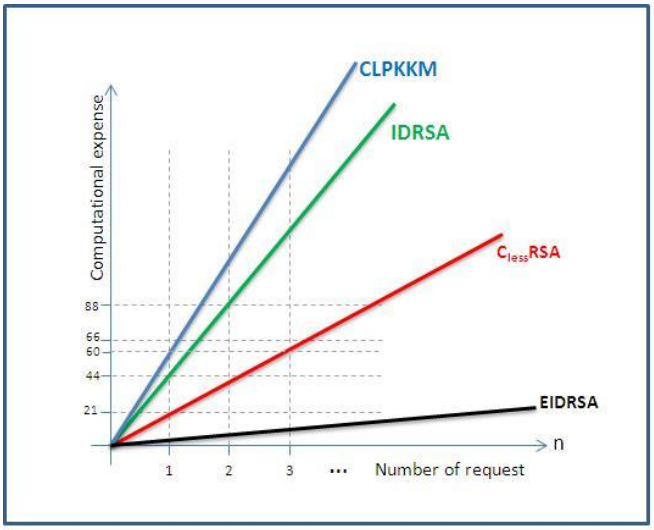

Fig.1 Growth rate of computational cost based on Type2 pairings

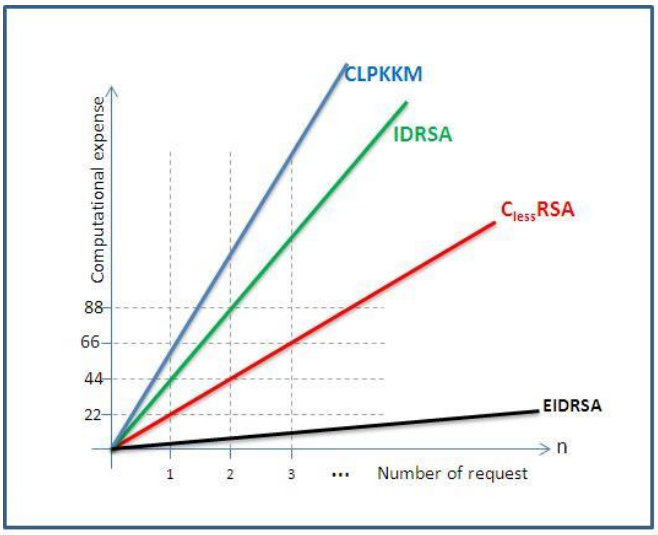

Fig.2 Growth rate of computational cost based on Type3 pairings

In the Figure 1 it is assumed that the utilized Bilinear Pairings in *CLPKKM*,*IDRSA* and $C_{less}RSA$ schemes are Type2, whereas in Figure 2 utilized Bilinear Pairings are Type3.

IV. CONCLUSION

In this paper, several certificateless public key cryptosystems for Mobile ad Hoc Networks have been reviewed. The functionality of each scheme is introduced in detail. Finally, a separate section compared the computational expense of the considered schemes. The result of this paper indicates that due to the elimination of Bilinear Pairing operation EIDRSA scheme is more efficient than the other ones from both computational expense and the rate of growth of computational expense viewpoints.

V. REFERENCES

[1]. Genik, L., Salmanian, M., Mason, P., Schotanus, H.A., Verkoelen, C.A.A., Hansson, E., (2004). MobileAd Hoc Network Security from a Military Perspective, DRDC Ottawa TM 2004-252, DefenceR&D Canada- Ottawa.

[2]. Hegland, A. M., Winjum, E., Spilling, P., Rong, C. and Kure, O. (2006). Analysis of IBS for MANET security in emergency and rescue operations. In Proceedings International Conference on Advanced Information Networking and Applications, AINA, vol. 2. Piscataway, NJ 08855-1331, United States. ISSN 1550-445X, 155 – 159.

[3]. Puzar, M., Andersson, J., Plagemann, T. and Roudier, Y. (2005). SKiMPy: A simple key management protocol for

MANETs in emergency and rescue operations. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3813 NCS. Heidelberg, D-69121, Germany. ISSN 0302-9743, 14 – 26.

[4]. Ru, L., Rui-Lin, Y. and Da-Wei, H. (2008). A designing of mobility management mechanism in MANET in disaster-rescue situations. In International Conference on Communication Technology Proceedings, ICCT. Piscataway, NJ 08855-1331,United States, 596 – 599.

[5]. Cano, J.-C., Calafate, C., Manzoni, P. and Toh, C.-K. (2007). Modeling of mobility and groups in inter-vehicular MANET-based networks. In 2007 2nd International Symposium on Wireless Pervasive Computing. Piscataway, NJ 08855-1331, United States, 333 – 337.

[6]. Tsukada, M. and Ernst, T. (2007). Vehicle communication experiment environment with MANET and NEMO. In SAINT - 2007 International Symposium on Applications and the Internet - Workshops, SAINT-W. Piscataway, NJ 08855-1331, United States, 4090112 –.

[7]. Van Der Merwe J., Dawoud D., and McDonald S.. (2007). A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Comput. Surv.* 39, 1, Article 1.

[8]. Capkun, S., Buttyan, L. and Hubaux, J.-P. (2003). Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on Mobile Computing. 2(1), 52 – 64. ISSN 1536-1233.

[9]. Douceur, J. R. (2002). The Sybil Attack. In IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems. London, UK: Springer-Verlag. ISBN 3540441794, 251–260.

[10]. Yi, S. and Kravets, R. (2003). MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks. In 2nd Annual PKI Research Workshop Program. 65–79.

[11]. Kim, J. and Tsudik, G. (2009). SRDP: Secure route discovery for dynamic source routing in MANETs. Ad Hoc Networks. 7(6), 1097 – 1109. ISSN 15708705.

[12]. Mao, L.-Q., Ma, J.-F. and Li, X.-H. (2009). Analysis of provably secure on-demand source routing in MANET. Tongxin Xuebao/Journal on Communication. 30(1), 38 – 44. ISSN 1000436X.

[13]. Xu, Y. and Xie, X. (2008). Security analysis of routing protocol for MANET based on extended Rubin logic. Sanya, China, 1326 – 1331.

[14]. Yu, M., Zhou, M. and Su, W. (2009). A secure routing protocol against byzantine attacks for MANETs in adversarial environments. IEEE Transactions on Vehicular Technology. 58(1), 449 – 460. ISSN 00189545.

[15]. Anjum, F. and Talpade, R. (2004). LiPaD: lightweight packet drop detection for ad hoc networks. Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th. 2, 1233–1237. ISSN 1090-3038.

[16]. Subhadrabandhu, D., Sarkar, S. and Anjum, F. (2004). Efficacy of misuse detection in ad hoc networks. Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on, 97–107. doi:10.1109/SAHCN.2004.1381907.

[17]. Vigna, G., Gwalani, S., Srinivasan, K., Belding-Royer, E. M. and Kemmerer, R. A.(2004). An intrusion detection tool for AODV-based ad hoc wireless networks. In Proceedings - Annual Computer Security Applications Conference, ACSAC. Los Alamitos, CA 90720-1314, United States. ISSN 1063-9527, 16 – 27.

[18]. Chen L., Cheng Z., Smart N.P. (2007). Identity-Based Key Agreement Protocols from Pairings .International Journal Of Information Security– Springer.

[19]. Zhao S., Akshai A., Frost R., Bai X.. (2011). A survey of applications of identity-based cryptography in mobile ad-hoc networks. IEEE Commun. Surv. Tutorials Early Access.

[20]. Al-Riyami S.S., Paterson K.G. (2003). Certificateless public key cryptography. Advances in Cryptology C Asiacrypt 2003, Lecture Notes in Computer Science, pp 452-473.

[21]. Eissa T., Razak S. A., Ngadi M.A. (2012). A Novel Lightweight Authentication Scheme for Mobile Ad Hoc Networks. AJSE **37**. pp 2179–2192.

[22]. Shabnam Kasra-kermanshahi, Mazleena Salleh, "An Enhanced Certificateless Cryptosystem for Mobile Ad Hoc Networks," in International Symposium on Biometrics and Security Technologies (ISBAST) (Kuala Lumpur, Malaysia: IEEE, 2014), pp. 176–181.

[23]. Shabnam Kasra-kermanshahi, Mazleena Salleh, "An Improved Certificateless Public Key Authentication Scheme For Mobile Ad Hoc Networks Over Elliptic Curves," in 4th World Congress on Information and Communication Technologies (Malacca, Malaysia: Springer, 2014), pp. 289–296.

[24]. Li L., Wang Z., Liu W. , Wang Y. (2011). A Certificate less Key Management Scheme in Moblie Ad Hoc Networks. 7th International Conf. on Wireless Communications, Networking and Mobile Computing, pp 1-4.