2014 International Symposium on Biometrics and Security Technologies (ISBAST)

An Enhanced Certificateless Cryptosystem for Mobile Ad Hoc Networks

Shabnam Kasra Kermanshahi, Mazleena Salleh
Faculty of Computing
Universiti Teknologi Malaysia
Johor, Malaysia
shabnam.kasra@gmail.com, mazleena@fsksm.utm.my

***Abstract*—Due to the importance of security in many critical applications in MANETs and the limitation of the resources in mobile devices, it is important to have lightweight cryptosystems. Although some lightweight certificateless cryptosystems for MANETs have been proposed, it is possible to improve them in the term of reducing the complexity of the computations. In this paper, we have tried to propose a lightweight certificateless public key cryptographic scheme based on bilinear pairings. In addition, we compared our proposed scheme with other existing certificateless pairing based and result shows that the proposed scheme is more efficient based on computational cost and the rate of growth of computational expense viewpoints.**

Keywords—Certificateless PKC, Bilinear Pairings, Lightweight, MANETs.

I. INTRODUCTION

These days popularity of distributed applications has led to the emphasizing of security issues in wide categories of mobile networks. Variety of mobile applications required security services and this causes many researchers to focus on the security issues over this kind of networks especially mobile ad hoc that is to overcome a subset of vulnerabilities in MANETs [1, 2].

MANETs, which are in the category of fixed-infrastructure-free wireless networks consist of movable nodes without the need to any centralized management. More precisely, involving nodes are responsible to perform the network functions, such as routing, cooperatively. The especial features of MANETs made these networks appropriate for a wide range of crucial applications such as military and battlefield ones such as rescue operations. However, the lake of a fixed infrastructure made it challenging in terms of security issues [3].

Previously, suggested security solutions by the researchers were mostly based on the attack-oriented approaches [4-8]. In fact, after identifying possible threats, they enhanced available schemes or designed a new one. Moreover, proposed schemes based on these approaches just cover limited attacks and could be once again vulnerable to some other attacks or a combination of them [1].

Cryptography has been widely used to provide the security of mobile ad hoc networks in general design framework [1]. However, the nature of resource constrained nodes is one of the significant problems that enforced developers to propose lightweight and less resource consuming cryptosystems in MANETs. The above mentioned reasons were sufficient to persuade some researchers that traditional public key cryptosystems such as RSA or DSA are not acceptable to use in such networks [8, 9]. Therefore, the use of symmetric cryptosystems became the basis for cryptographic schemes such as RC5 [10] and Skip-Jack [11] in resource constrained platforms for many years [12]. However the use of symmetric cryptosystems suffers from a subset of disadvantages especially key management problem. This problem can be overcome with the use of public key cryptographic schemes that could make key management security services easier and reduced the overhead of transmitting processes [13, 14]. It is due to this fact that many researchers have proposed public key cryptosystems that are lightweight enough in order to make them implementable in resource constrained nodes.

To make the use of public key cryptosystems feasible in mobile ad hoc networks, elimination of Public Key Infrastructure (PKI) has been one of the interesting challenging issues. To reach this goal, Adi Shamir proposed the idea of Identity-based cryptosystems in order to eliminate the need to certificates and PKI management [15]. This cryptographic idea remained an open problem for seventeen years, until Boneh and Franklin [16] proposed a practicable solution. From there onwards many researchers have been pursuing the fully functional solution of Boneh and Franklin in a large variety of Identity-based cryptographic primitives, such cryptosystems which suffer from a significant problem namely the Key Escrow. More accurately, in these cryptosystems there is a trusted third party, PKG, who possesses the private-key of all existing entities. To solve this problem, Al-Riyami and Paterson proposed a novel cryptosystem to capture the advantages of both Identity-based and traditional, named Certificateless PKC [17]. In this category of cryptosystems, the main idea is the use of

978-1-4799-6444-4/14/$31.00 ©2014 IEEE

identifier not as a public key, but to eliminate the need to public key certificates beside of solving Key Escrow problem.

In order to draw appropriate cryptosystems for mobile ad hoc networks, a number of certificateless public key cryptosystems have been proposed [18, 19]. However, none of them seems to be reliable enough to be used in some crucial applications. In addition, the efficiency of the proposed schemes needs to be further improved so that it is appropriate for resource constrained problem devices of mobile ad hoc networks. The aim of this research is to propose a certificateless cryptosystem in the context of MANETs, named C_{less}RSA, which is an improvement of IDRSA [18] in terms of time and computational efficiency perspective.

II. TECHNICAL BACKGROUND

Bilinear pairings is the preliminary requirement of certificateless cryptosystem. The followed subsection briefly introduces bilinear pairings which is defined on algebraic groups over Elliptic Curves.

Bilinear pairings

Bilinear maps are based on Miller algorithm [20]. A bilinear map, a cryptographic building block in designing pairing-based cryptosystems, is in fact a deterministic function such as $\hat{e}$ which is defined over three algebraic groups. In more detail, assume that there are three algebraic groups G_1, G_2 and $G_T$of a prime order q. Then, a map such as $\hat{e}: G_1 \times G_2 \rightarrow G_T$ is a bilinear pairing if it can support three of the following characteristics as stated below:

i. Bilinearity, means that $\forall P \epsilon G_1$, $\forall Q \epsilon G_2$, $\forall a, b \epsilon Z_q$: $\hat{e}(P^a, Q^b) = \hat{e}(P, Q)^{ab}$
ii. Non-degeneracy, means that if I_1, I_2 and I_T are identity elements of G_1, $G_2$and G_T, respectively, then *"e"* do not map any pair of $(G_1 \times G_2)$ to I_T unless (I_1, I_2).
iii. Computability, means that for any$P \epsilon G_1$ and$Q \epsilon G_2$, there must be an efficient algorithm to compute $\hat{e}(P, Q)$.

It is worth noting that because of the widely usage of pairing-based applications, many researchers have propose several efficient bilinear pairings schemes. Weil pairing and Tate pairing are examples of the most recently used bilinear pairings in cryptographic schemes [21, 22]. Implementing bilinear pairings is mathematically complex and this research excludes the details of this category of cryptographic maps. However, to implement bilinear pairings, the mentioned three groups are defined over algebraic elliptic curves. There exist various documents that have tried to investigate algebraic groups, which are defined over different Elliptic Curves.

Many reasons could persuade cryptologists to use elliptic curves based cryptosystems. One of the most significant advantages of using ECC based algebraic groups in compare with RSA based on the need of a smaller key size in the same security level [23, 24].

Different types of bilinear pairings

It is possible to define three different pairing types based on the proposed pattern in [25]. To introduce different types of this kind of cryptographic maps, assume that the considered bilinear pairing appears in the form of $\hat{e}: G_1 \times G_2 \rightarrow G_T$. Then, different types of bilinear pairings can be defined as followed:

Type 1 bilinear pairings

In this category of pairings there exists two efficiently computable homomorphism between G_1 and G_2 in both directions (but it is usually supposed that $G_1 = G_2$). This type of bilinear pairing is efficient, but it is not easily possible to obtain more than 80 bits security level [25].

Type 2bilinear pairings

In this category of pairings, there exists an efficiently computable homomorphism from G_2 to G_1, but it is not efficient to find a homomorphism from G_1 to G_2. It is necessary to note that it is possible to obtain more than 80 bits security level for this category of bilinear pairings, but the main problem of them is that computations over G_2 group operations are expensive [25].

Type 3 bilinear pairings

In this category of pairings, there is not any efficiently computable homomorphism between G_2 and G_1. It is worth mentioning that it is possible to obtain more than 80 bits security level for this category of bilinear pairings and computations over G2 group operations are not expensive. However, this type of pairing is not as efficient as Type 1 bilinear pairings [25].

It is worth pointing out that knowing the features of mentioned types and supposing valid assumptions to build a cryptographic scheme, is one of the most important parts of developing pairing-based cryptosystems. The reason is that although building cryptographic schemes through assuming black-box pairings is not necessarily a bad approach, it is sometimes easy for designers to make invalid or impractical assumptions [25].

III. RELATED WORKS OVER CERTIFICATELESS PUBLIC KEY CRYPTOSYSTEMS IN MANETS

As mentioned, in an identity-based cryptosystem each entity must collect its private key from PKG. Hence PKG can eavesdrops 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 [26]. 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 [1].

In 2003, a novel public key cryptosystem was introduced by Al-Riyami and Paterson [17], named certificateless public key, that could overcome the key escrow problem while public keys are authenticated without need to the PKI. In this cryptosystem, a trusted third party called Key Generator Center (KGC) is responsible for generating partial private keys for the users. This key is driven by master key (only known by KGC), and the users' identity. Then, each user can

produce his private key. The user's private key can be generated by the use of partial private key received from KGC and a secret value chosen by the user hence there is no problem regarding to the key escrow [26]. Since, in the CL-PKC the partial secret key generated by the KGC 'implicitly' certified the public key, the certificates are not needed [26]. Therefore, for the public key authentication process, the public key of the KGC is required.

The purpose of this section is to probe into IDRSA from cryptographic functionality viewpoint.

An overview of IDRSA scheme

The main objective of this subsection is to investigate the *IDRSA* protocol. *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.

Main phases of IDRSA

It can be claimed that the core part of *IDRSA* scheme is consisted of three main phases that we named them Setup, Node Initialization, and Public-key Obtaining Process. To have better understanding of the proposed scheme, the *IDRSA* three main phases will be reviewed here briefly.

Setup: In this phase, a trusted third party generates public parameters of the cryptosystem (Params) after taking the security parameters as below:

$$Params: < G_1, G_2, G_T, q, P, \hat{e}, n, H_1, H_2, H_3 >$$

Here, $< G_1, + >$, $< G_2, + >$ and $< G_T, \times >$ are three algebraic groups of the same prime order q. In addition, $P \in G_2$ and $\hat{e}: G_1 \times G_2 \to G_T$ is a bilinear pairing over mentioned algebraic groups. Moreover, n is a positive integer number that determines the number of bits of the two components of the RSA public key (e and N) for existing users. Beside of these, $H_1: \{0,1\}^* \to {G_1}^*$, $H_2: G_T \to \{0,1\}^n$ and $H_3: \{0,1\}^* \to \{0,1\}^n$ are three one-way collision-free hash functions

Node Initialization: The basis of this phase is to generate a subset of public and private parameters for existing users and coalitions, besides 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 followed:

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

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 parameters e_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: $A \to IDRSA_i: P_A, ID_B$

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 and N_B) by sending ID_B to the $IDRSA_i$ coalition.

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

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 follow:

$$U = P_i \text{ , } C = e_B \oplus H_2(g_i) \text{ that } g_i \text{ is equal to } \hat{e}(Q_A, P_A)^{d_i} \times \hat{e}(d_i Q_i, P_A),$$
$$W = e_B . P \text{ and } Y = N_B \oplus H_3(e_B)$$

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

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 and g_i, respectively. The calculations below, can show that the result of both computations is the same value $\hat{e}(Q_A + Q_i, P)^{d_i d_A}$.

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

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

IV. Proposed Scheme: $C_{less}RSA$

This section discusses the proposed certificateless PKC scheme named $C_{less}RSA$, which is an improved version of *IDRSA* from computational efficiency viewpoint. In more detail, we demonstrated that our proposed scheme can perform less computation in the initialization phase in

compare with *IDRSA*. It is worth to note that in this part, our proposed scheme, $C_{less}RSA$, is compared with *IDRSA* from computational efficiency perspective.

An overview of $C_{less}RSA$

$C_{less}RSA$ scheme has three main phases named Setup, Node Initialization, and Public-key Obtaining Process. However, the computational expenses are more lightweight than *IDRSA*. In more detail, these phases are described as follow

Setup: In this phase, the KGC takes the required security parameter to issue the public parameters of the cryptosystem (Params) as followed:

$$Params: < G_1, G_2, G_T, q, P, \hat{e}, n, H_1, H_2, H_3 >$$

The elements of the tuple params are as follow:

q is a large prime integer, $< G_1, + >$, $< G_2, + >$ and $< G_T, \times >$ are three algebraic groups with the same order q. $P$is an element of the group G_2. The map $\hat{e}: G_1 \times G_2 \rightarrow G_T$is a determined bilinear pairing over mentioned three groups. In addition, the element n refers to an integer number that determines the number of bits for the public key of the RSA cryptosystem (e and N) for existing entities. Moreover, $H_1: \{0,1\}^* \rightarrow G_1{}^*$, $H_2: G_T \rightarrow \{0,1\}^n$ and $H_3: \{0,1\}^* \rightarrow \{0,1\}^n$ are three one-way collision-free hash functions

Node Initialization: Similar to *IDRSA* scheme, $C_{less}RSA$ consisted of some entities that can be users or coalitions. In the Node Initialization phase, the public and private parameters for mentioned entities will be generated and a subset of public ones will be published to all other entities. Such as *IDRSA*, the public parameters of $C_{less}RSA$ scheme are identity-key, general-key and public-key. All entities are able to compute the identity-key of existing users, but to generate general-key and public-key, the owner of them must do that. Similar to the *IDRSA* scheme, the identity-key of the entity who possessID_i identifier would be generated as follow:

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

Then, mentioned entity randomly chooses the prime number e_i that $e_i \in_r \mathbb{Z}_q^*$. After this, mentioned entity runs the RSA key generation algorithm to generate the parameterse_i, d_i, N_i . The same as traditional RSA scheme, d_i and $< e_i, N_i >$ are the private-key and public-key of mentioned entity, respectively. Finally, mentioned entity publishes the value $P_i = (d_i . P)$ as the general-key.

Public key obtaining process in $C_{less}RSA$:

Similar to *IDRSA*, in the last phase of $C_{less}RSA$ each user refers to the considered coalition and requests for the public-key of the other side party. 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:

$$U = P_i, C = e_B \oplus H_2(g_i) \text{ that } g_i \text{ is equal to } g_i = \hat{e}(d_i Q_A, P_A),$$
$$W = e_B . P \text{ and} Y = N_B \oplus H_3(e_B).$$

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:

Firstly, 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 as g_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 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.

B. 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 $g_A$andg_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.

V. Efficiency Comparison Between $C_{less}RSA$ And *IDRSA*

Because of the high expense of bilinear pairings in compare with other group operations [27] the comparison of computational expense emphasizes on computing g_A and g_i parts of *IDRSA* and $C_{less}RSA$ schemes. This comparison is based on assuming that $g_A$and $g_i$parts of $C_{less}RSA$ and *IDRSA* schemes are constructed by Type2 or Type3 bilinear pairings. 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.

Computational expenses of IDRSA and $C_{less}RSA$ schemes

In order to make the *IDRSA* more efficient, in the proposed $C_{less}RSA$ scheme, we have tried to decrease the number of utilized bilinear pairings. The main reason is that the pairing operations are more expensive than modular exponentiation and scalar multiplication operations [27]. The Table I illustrate the expenses of operations (pairings,

modular exponentiation and scalar multiplication) in Type2 and Type3 bilinear pairings based on the assumptions of the [27]. 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 [27].

TABLE I. COMPUTATIONAL EXPENSE OF GROUP OPERATIONS INTYPE2 AND TYPE3 BILINEAR PAIRINGS [27]

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

To compute the efficiency of *IDRSA* and $C_{less}RSA$ schemes we focused on the computational expense of $g_A$or$g_i$parts of " Public key obtaining process," which is the core of the difference between mentioned schemes. Based on the Table I, computational expense of $g_A$or g_i part in *IDRSA* is equal to "E_T+M_1+2P".

Table II illustrates the expense of these parts in *IDRSA* scheme followed by the values of the Table I.

TABLE II. COMPUTATIONAL EXPENSE OF g_AORg_i PARTS IN IDRSA SCHEME

Pairings type	Type2	Type3
Computational expense	46	44

Furthermore, based on the Table I, computational expense of $g_A$or $g_i$parts of public key obtaining process in $C_{less}RSA$ is equal to M_1+P. Table III demonstrates the expense of these parts in $C_{less}RSA$scheme.

TABLE III. COMPUTATIONAL EXPENSE OF g_AORg_i PARTS IN $C_{less}RSA$SCHEME

Pairings type	Type2	Type3
Computational expense	22	21

To conclude the results of the tables Table II, Table III, it is clear to realize that $C_{less}RSA$ scheme requires less computational cost for the used operations than the *IDRSA* scheme. To see this improvement more accurately we can refer to the Table IV. This table demonstrates the percentage of computational efficiency improvement of $C_{less}RSA$ scheme in compare with IDRSA.

TABLE IV. IMPROVEMENT PERCENTAGE OF COMPUTATIONAL EXPENSE OF $C_{less}RSA$ SCHEME IN COMPARE WITH *IDRSA*

Pairings type	Type2	Type3
Improvement percentage over *IDRSA* scheme	%52	%52

This issue would be more drastic during the growth of the number of requests for public key obtaining process in IDRSA and $C_{less}RSA$ schemes. Figure 1 and Figure 2 depict the rate of growth of computational expense for$g_A$or$g_i$parts of public key obtaining process of the schemes IDRSA and $C_{less}RSA$ scheme as a function of the number of requests.

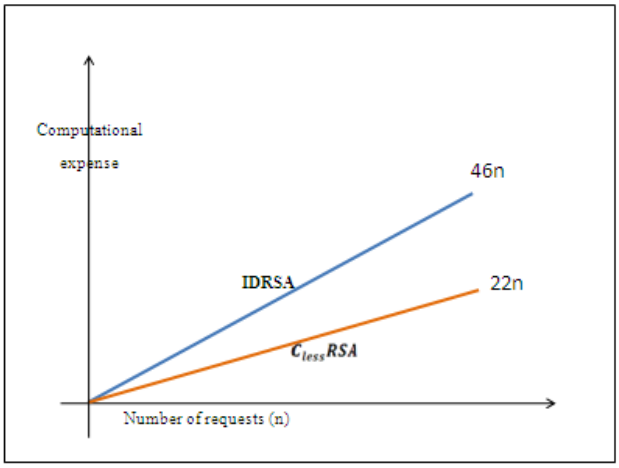

Figure 1. Growth rate of computational cost based on Type2 pairings

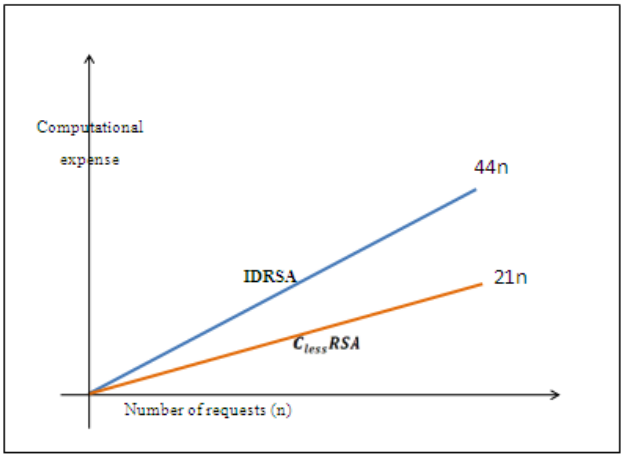

Figure 2. Growth rate of computational cost based on Type3 pairings

Figure1 assumes that the used bilinear pairings of all *IDRSA* and $C_{less}RSA$ schemes are Type2, whereas Figure2 assumes that mentioned bilinear pairings are Type3 ones.

VI. SUMMARY

In this paper, the functionality of the improved version of $IDRSA$ named $C_{less}RSA$ is introduced in detail. Finally, a separate section compared the computational expense of the schemes $IDRSA$ and $C_{less}RSA$. The result of this study proves that $C_{less}RSA$ scheme is more efficient than $IDRSA$ from both computational expense and the rate of growth of computational expense viewpoints.

REFERENCE

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

[2] L. Abusalah, A. Khokhar, and M. Guizani. "A survey of secure mobile ad hoc routing protocols," IEEE Commun. Surveys & Tutorials, IEEE, vol. 10, no. 4, pp. 78–93, 2008.

[3] C. Sen, M. Salmanian, M. Kellett., "A Mobile Ad Hoc Networking Test Bed", DRDC Ottawa TM 2005-158, DefenceR&D Canada-Ottawa, August 2005.

[4] Y. Hu, A. Perrig, and D. Johnson, Packet Leashes: A Defense Against Wormhole Attacks in Wireless Ad Hoc Networks. Proc. of IEEE INFORCOM, 2002.

[5] S. Capkun, L. Buttyan, and J. Hubaux, Sector: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. Proc. of the ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003.

[6] S. Yi, P. Naldurg, and R. Kravets, Security-Aware Ad-hoc Routing for Wireless Networks. Report No.UIUCDCS-R-2002-2290, UIUC, 2002.

[7] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, A Secure Routing Protocol for Ad Hoc Networks. Proc. of IEEE International Conference on Network Protocols (ICNP), pp. 78-87, 2002

[8] Y. Hu, D. Johnson, and A. Perrig, SEAD: Secure Efficient Distance Vector Routing in MobileWireless Ad-Hoc Networks. Proc. of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02), pp. 3-13, 2002.

[9] P. Papadimitratos and Z. Haas. Secure Routing for Mobile Ad Hoc Networks, Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), 2002.

[10] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," in Proc. Conf. Wireless Networks, pp. 521-534, 2002.

[11] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: A link layer security architecture for wireless sensor networks," in 2nd ACM Sens Sys, pp.162-175, Nov. 2004.

[12] L.B. Oliveira and R. Dahab, "Pairing-based cryptography for sensor networks," presented at IEEE International Symposium on Network Computing and Applications, Cambridge, MA, July 2006.

[13] G. Gaubatz, J.-P. Kaps, E. Oztruk, and B. Sunar, "State of the art in ultra-low power public key cryptography for wireless sensor networks," in Proc. Per Sec '05, IEEE, pp. 146-150, 2005.

[14] J. K. Liu, J. Baek, J. Zhou, Y. Yang, and J. W. Wong, " Efficient online/offline identity-based signature for WSN, " in Proc. IJIS, pp. 287-296, 2010.

[15] A. Shamir, "Identity-Based Cryptosystems And Signature Scheme"S, In Advances In Cryptology—Crypto 1984, Lecture Notes In Comput.Sci. 196, Springer-Verlag, Berlin, 1984.

[16] Boneh, D., Franklin, M., "Identity Based Encryption From The Weil Pairing". Advances In Cryptology—Crypto, 2001.

[17] S.S.Al-Riyami K.G.Paterson. Certificateless public key cryptography. page 452C473. C.S. Laih (ed.) Advances in Cryptology C Asiacrypt 2003, Lecture Notes in Computer Science, 2003.

[18] T. Eissa, S. A. Razak, M.A, Ngadi.. "A novel lightweight authentication scheme for mobile ad hoc networks". AJSE 37, 2179–2192, 2012.

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

[20] V. Miller, "Short Programs For Functions On Curves", Unpublished Manuscript, 1986.

[21] J. Tate, "Duality Theorems In Galois Cohomology Over Number Fields", Proceedings Of The International Congress Of Mathematicians (Stockholm, 1962), Djursholm: Inst. Mittag-Leffler, 1963.

[22] J. Capco, " Weil Pairings On Elliptic Curves", 2003.

[23] NIST Recommendation For Key Management Part 1: General, NIST Special publication 800-57. August, 2005.

[24] ECRYPT Yearly Report On Algorithms And Keysizes , 2004.

[25] Galbraith, S., Paterson, K., Smart, N.P.:. "Pairings For Cryptographers", 2006. Cryptology Eprint Archive, Report 2006/165.

[26] Zhenfei Zhang, Willy Susilo, and Raad Raad," Mobile Ad-hoc Network Key Management with Certificateless Cryptography," IEEE, 978-1-4244-4242, Aug. 3, 2008.

[27] L. Chen, Z. Cheng, Np Smart – "Identity-Based Key Agreement Protocols From Pairings" .International Journal Of Information Security– Springer, 2007.