Arab J Sci Eng (2012) 37:2179–2192
DOI 10.1007/s13369-012-0318-y

RESEARCH ARTICLE – COMPUTER ENGINEERING AND COMPUTER SCIENCE

Tameem Eissa · Shukor Abd Razak · Mohd Asri Ngadi

A Novel Lightweight Authentication Scheme for Mobile Ad Hoc Networks

Received: 4 September 2010 / Accepted: 13 March 2011 / Published online: 21 June 2012

Abstract The lack of infrastructure and central authority (CA) makes MANET security a very challenging mission. Fast and lightweight security solutions are required because of the mobility feature and resources limitation of such network. Most recent work uses the identity-based encryption as a basic solution for MANET security. However, the continuing usage of the bilinear pairing operations is costly and not suitable for such environment. In this paper, we present DIDRSA, a new decentralized identity-based RSA authentication scheme for MANET. The number of bilinear pairing operations is reduced to $(1 + t)$ operations. The public keys are secured to provide a safe method for using RSA cryptography speeding techniques. We prove that our scheme is secure against RSA attacks involved when using such speeding techniques. The scheme performance has been tested using simulation scenarios under different routing protocols. We also highlight the usage of this scheme for AODV routing protocol security as a future work.

Keywords Mobile ad hoc network · Security · Trust feature

الخلاصة

إن غياب البنية التحتية والسلطة المركزية (CA) جعل أمن MANET مهمة صعبة للغاية، حيث إن الحلول الأمنية سريعة وخفيفة الوزن هي مطلوبة بسبب ميزة التنقل ومحدودية الموارد لشبكة من هذا القبيل. وقد استخدمت معظم الأعمال الأخيرة التشفير المستند إلى الهوية كحل أساسي لأمن MANET. ومع ذلك فإن الاستخدام المستمر لعمليات الاقتران شبه الخطية هو مكلف وغير مناسب لبيئة من هذا القبيل.

وسوف نقدم - في هذه الورقة العلمية – DIDRSA ، وهو مخطط مصادقة لامركزي جديد على أساس الهوية RSA لنظام MANET. يتم تقليل عدد عمليات التزاوج شبه خطية للعمليات، كما يتم تأمين المفاتيح العمومية لتوفير وسيلة آمنة لاستخدام تقنيات تشفير RSA المسرعة. لقد أثبتنا أن مخططنا هو آمن ضد هجمات RSA المعنية عند استخدام تقنيات مسرعة من هذا القبيل. وقد تم اختبار أداء النظام باستخدام سيناريوهات محاكاة تحت بروتوكولات توجيه مختلفة. وسوف نسلط الضوء أيضا على استخدام هذا النظام لتحقيق أمن بروتوكول توجيه AODV بوصفه عملاً في المستقبل.

T. Eissa (✉)
Division of CSE, Chonbuk University, 664-14 Duckjin-Dong, Duckjin-Gu, Jeonju, 561-756 Jeonbuk, South Korea
E-mail: Tamnet83@gmail.com

S. A. Razak · M. A. Ngadi
Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
E-mail: shukorar@utm.my

1 Introduction

Nowadays, mobile ad hoc network (MANET) has attracted many researchers due to its capabilities to be installed on fly and without infrastructure requirements. These features make such networks suitable to be used in many fields (military applications, business indoor applications, civilian outdoor applications, emergency and wireless imaging applications [1]).

Implementing security in such networks is not as easy as in the networks with infrastructure because of the mobility nature and the lack of infrastructure. The distributed authentication scheme has been deployed in [2–5] where the authority is distributed on many nodes (servers) using threshold cryptography (TC). Other works implemented RSA in MANET using threshold cryptography [6]. However, it is proved that using RSA-based TC is unsuitable for the limited resources of the devices in MANET due to the high storage and computational requirements [7].

The cluster-based authentication scheme has been proposed in [8] where a manager node (cluster head) controls each group of nodes (cluster). However, this scheme is exposed to the single point of failure since all the cluster nodes depend on one CH node. To increase the availability of the service, George et al. [9] proposed a hierarchical certificate authority scheme. However, this scheme requires a lot of control messages and thus, it is not suitable for MANET. Weimerskirch and Thonet [10] proposed a trust-based authentication scheme to evaluate the nodes using friends' recommendations and references. However, this scheme does not provide data privacy as long as messages go through the shared wireless medium and can be seen by all the nodes in the range. An Ad Hoc Trust (ATF) framework has been proposed to support Ad Hoc Distributed OCSP for Trust (ADOPT) scheme and improve its performance and efficiency [11]. To create a trust chains between nodes, a trust reference chain has been proposed in [12] where each element of the chain is the trust value of a route hop.

Datta et al. [13] proposed an autonomous gossiping algorithm to propagate selective information. Selective data are efficiently disseminated in the network using an epidemic mechanism and without the need for routing information.

Most of the works mentioned above use the public key cryptography (PKC) as cryptography scheme. However, keys and certificates management is one of the difficulties faced when using the PKC [14]. Identity-based encryptions (IBE) had been proposed as an alternative to the PKC. The idea is to use the identity information (such as email address) as public key. As a result, there is no need for certificates implementation and then this provides lightweight implementation for MANET.[1]

The different MANET IBE-based schemes can be listed briefly as following:

- Polynomial interpolation-based IBE [16]: in this scheme, the master private key is generated cooperatively by the PKC nodes using threshold cryptography. Each PKG node participates in the private key generation of the other nodes using its master private key share.
- Bilinear pairing-based IBE [17]: this scheme uses the bilinear pairing techniques. The service master key is used to generate the nodes private keys where the Bilinear Diffie–Hellman Problem (BDHP) problem is intractable.
- Identity-based threshold decryption scheme (IDTHDBM) [18]: this scheme uses bilinear pairings and threshold cryptography concepts. It enables a group of nodes to cooperate with each other to decrypt an encrypted message. At least t nodes from this group should be available to decrypt the message. To decrease the number of pairing calculations, Kiltz and Galindo [19] proposed an identity-based key encapsulation mechanism (IB-KEM) based on data-encapsulation mechanism (DEM).
- Trivariate polynomial-based IBE scheme [20]: this scheme is fully self-managed by the nodes. The decryption threshold (t'') is chosen by the sender at the time of encrypting.

However, these schemes require a lot of bilinear pairing operations. Such operations are considered costly for the limited resources of MANET devices. Furthermore, such schemes require a lot of control messages which cause overhead in the network.

In this paper, we present DIDRSA, a new lightweight authentication scheme for MANET. The public keys are secured using pairing techniques. The purpose of securing the public keys is to apply the RSA speeding techniques without making the system vulnerable to RSA attacks [21]. The bilinear pairing operations are reduced to $(t+1)$ operations. This improves the computation cost. The enhanced RSA cryptography operations are used for data authentication. This makes the scheme more lightweight. This paper is organized as follows.

[1] Refer to our paper [15] for a survey of authentication schemes in MANET.

Table 1 Notation

$G, G_1, G_2, G_\mathrm{T}$	Cyclic groups of order q
$\hat{e}$	Bilinear pairing, i.e.: $G_l \times G_l \to G_2$
P	The generator of G_1
t, n	Threshold cryptography parameters
e	RSA public exponent
d	RSA private exponent
N	The modulus of RSA
p,q	Two large primes
C	The ciphertext
M	The plaintext
$H_1. H_2, H_3$	Hashing functions
1	The message length
P_{d_A}	Public parameter for node $A.P_{d_A} = d_A \cdot P$
Q_{ID_A}	Identity key of node A
MIPS	One million instructions per second

Section 2 describes some preliminaries and notations. Section 3 presents our new scheme and evaluates it via simulation. Section 4 discusses the expected future work of the proposed scheme. Finally, this paper is concluded in Sect. 5.

2 Preliminaries

In this section, we first define the notation to be used in the rest of this paper. We then define some preliminaries about bilinear pairings.

2.1 Notation

Table 1 lists some important notations to be used in this paper. The meaning of these notations will be further mentioned where they appear for the first time.

2.2 Preliminaries Definitions

2.2.1 Bilinear Pairings

Let G_1 be an additive group of order q, G_2 a multiplicative group of the same order.

The map $e : G_1 \times G_1 \to G_2$ is called a bilinear pairing [22], if (and only if) it satisfies the following properties:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1, a, b \in Z_q$.
- Non-degeneratity. Each element of G_1 is appended to an element S from G_2 such that: $s \neq ID_{G_2}$ (the identity element in G_2).
- Computability: $\forall P, Q \in G_1, e(P, Q)$ can be computed efficiently.

2.2.2 Bilinear Diffie–Hellman Problem (BDHP)

Let G_1 be an additive group of prime order q, G_2 a multiplicative group of the same order, P the generator of G_1, and $e : G_1 \times G_1 \to G_2$ a bilinear pairing on (G_1, G_2). Bilinear Diffie–Hellman Problem is defined as following: given P, aP, bP, cP, for $a, b, c \in Z_a^*$ compute $e(P, P)^{abc} \in G_2$

2.2.3 Discrete Logarithm Problem (DLP)

Let G be a group of order q, P the generator of G, $y \in G$, find an integer $x \in Z_q$ such that $P^x = y$.

2.2.4 Computational Diffie–Hellman Problem (CDH)

Let G be a group of order q, P the generator of G. Given P^{x_A}, P^{x_B} (where $x_A, x_B \in Z_q$), compute $P^{xA \cdot xB}$.

3 Decentralized Identity-Based RSA Authentication Scheme

This section presents DIDRSA, a new decentralized identity-based RSA authentication scheme that combines the PKC and IBE features in one scheme. We adopt using RSA since it is one of the strongest PKC schemes. We also adopt using small public key exponent e and small secret CRT exponent d (such that $q > N^{0.468}$) in order to speed up the cryptography operations.

CRT represents the equation $m = C^d$ as following:

$$m = (m1, m2)$$
$$m1 = C_p^d mod(p-1)$$
$$m2 = C_q^d mod\left(\frac{q-1}{2}\right)$$

This is called the modular representation of m and it is computed much faster than computing I as: $m = C^d \bmod n$ [23].

Shorter exponents make the cryptography operations much faster and more lightweight and suitable to be implemented by MANET limited resource nodes. However, it makes the scheme vulnerable to many RSA attacks. Fortunately, such attacks require the knowledge of the public key information. As a result, we propose to secure the public keys in order to prevent such attacks. The identity key can be used by any node directly to secure the public keys without the need for certificates. We assume that the Computational Diffie–Helman problem (CDH) is hard. The proposed protocol is described in the following sections:

3.1 Nodes Initialization

In this phase, node keys are generated as follows:

1. The identity key can be generated by any node using the hash function as: $Q_{\downarrow}ID = H_{\downarrow}1(ID\|time-expire)$
2. Each node chooses a prime number e from $(0.1)^l$.
3. Each node runs RSA key generation algorithm to generate a private key using small CRT secret exponent d modulus N.

3.2 The Bootstrapping Phase

The bootstrapper chooses two additive groups G_1, G_2 and a multiplicative group G_{T} of the same prime order q, defines asymmetric pairing of type 2 [24]: $e : G_2 \times G_1 \rightarrow G_{\mathrm{T}}$ and chooses a random generator P from G_1. The message space $M = (0.1)^l$, where l is the message length, the cipher space is $G_l^{\cdot} \times (0.1)^l$. It chooses three cryptography hash functions:

$$H_1 : (0,1)^* \rightarrow G_1^*$$
$$H_2 : G_T^* \rightarrow (0,1)^l$$
$$H_3 : (0,1)^* \rightarrow (0,1)^l$$

and defines two mapping functions $F_{1,}$ F_2

$$F_1 : Z_p \rightarrow (0,1)^*$$
$$F_2 : (0,1)^* \rightarrow Z_p$$

Then it publics the system parameters:

$$\langle P, G_1, G_2, G_{\mathrm{T}}, q, e, l, H_1, H_2, H_3 F_1 F_2 \rangle$$

Finally, it chooses n nodes as DIDRSA$_{\mathrm{s}}$ servers according to some features determined in the pre-configuration phase.

3.3 The Authentication Process

Suppose that a node A needs to communicate with a node B. If it already has its public key (if they communicated before) then it uses it directly to verify B messages. Otherwise, it should ask DIDRSA$_\text{s}$ nodes for the public key of node A. The algorithm works as following:

1. A sends a request Req-pub to any coalition of t DIDRSA$_s$ nodes, the request includes the identity of the requested node ID_B, and the value $P_{d_A} = d_A \cdot P$. Where d_A is the private exponent of A.
2. When DIDRSA$_i$ receives the request, it checks if ID_B is registered in its trusted list. If yes, it retrieves A public key and signcrypts it as following:

   $$c = F_1(e_B) \oplus H_2(g_{ID_i})$$

   where

   $$g_{ID_i} = \hat{e}(Q_{ID_A}, P_{d_A} \cdot Q_{ID_i})^{d_i}$$

   Then DIDRSA$_\text{s}$ sends $\langle U, C, W, Y\rangle$ to node A. Where $U = P_{d_i}$, $W = e_B \cdot P$, $Y = N_B \oplus H_3(e_B)$, N_B is the modulus of RSA cryptography for B.
3. Node A gets the public key of B by decrypting the ciphertext as follows:

   $$e_B = F_2(C \oplus H_2(\hat{e}(d_A \cdot Q_{ID_A}, P_{d_i} \cdot Q_{ID_i})))$$

   Then it computes $N_B = Y \oplus H_3\{e_B\}$. Finally, it checks if $W = e_B \cdot P$, if not, it rejects DIDRSA$_i$ response.
4. Upon receiving t valid responses, node A registers node B public key in its trusted list.

3.4 Correctness

We provide the correctness of the proposed scheme using bilinear pairing properties as follows:

$$\begin{aligned}
&\hat{e}(Q_{ID_A}, P_{d_A} \cdot Q_{ID_i})^{d_i}\\
&\quad = \hat{e}(Q_{ID_A}, d_A \cdot P \cdot Q_{ID_i})^{d_i}\\
&\quad = \hat{e}(d_A \cdot Q_{ID_A}, P \cdot Q_{ID_i})^{d_i}\\
&\quad = \hat{e}(d_A \cdot Q_{ID_A}, d_i \cdot P \cdot Q_{ID_i})\\
&\quad = \hat{e}(d_A \cdot Q_{ID_A}, P_{d_i} \cdot Q_{ID_i})
\end{aligned}$$

3.5 Security Analysis

In this section, we focus on RSA attacks involved when using short public and private exponents. We differentiate between two types of such attacks:

- Type 1 attacks: such attacks aim to get the private exponent of any node. They include Brute-force attack [25], common modulus attacks [26], Wiener attack [27], Boneh attack [28], May attack [29], timing attack [30], random faults attack [31] and partial key exposure attack [32].

All type 1 attacks require the knowledge of the public key in addition to other parameters. For example, the partial key exposure attack requires the calculation of d' which represents an approximation (almost the half) of d and computed as $d^{tr} = [k(N+1)/e$. As a result, our scheme is secure against type 1 attacks.

- Type 2 attacks: such attacks aim to recover a message M encrypted by any node. They include low public exponent attack [33], Hastad's broadcast attack [34], Franklin–Reiter-related message attack [35] and Coppersmith's short pad attack [36].

Most type 2 attacks require the knowledge of the public key except for the timing attacks and Hastad's broadcast attack. Timing attacks can be defended by adding delay so that the signing process will take fixed amount of time [30]. While Hastad's broadcast attack can be defended using different public exponents for each node in the network.

Table 2 Approximate time for factoring the modulus without the knowledge of *e*

Number of digits	Number of bits	Time (MIPS)	Modulus value
34	112	1.398 (min)	2000336999557143748666175116362331
50	166	3.818 (h)	67446226657731121816160970852484034396059245676261
55	182	15.52 (h)	7754714625452413658101666462631056234613335613636956159
60	199	2.364 (days)	549651848716851476171519309520512180240631684479399905831127
71	253	1.12 (months)	19944243511003710645688041908750310432034150054070052409154544735870689
106	352	258.77 (years)	8987519406550215703753149415623952312887748246322638454671128134621192823223702645452672771994322611107223

3.5.1 Defense Against Factoring Attacks

Let us say that the attacker could get the modulus N of one of the nodes somehow. Can he use this information to recover the message without the knowledge of the public exponent e?

In fact, the attacker can factor the modulus N into two prime factors (p, q). However, the requirement of such attack is different from RSA standard attacks requirement where the public exponent is available for the attacker. The most famous factoring algorithm known for factoring large integers is the quadratic sieve (QS) [37]. The QS was the fastest known factoring algorithm until the discovery of the number field sieve algorithm [38]. However, for factoring a number less than 110 digits, QS is still faster than the number field sieve. QS requires asymptotic running time

$$O(e^{1=o(1)\mathrm{sqrt}(\log(n)\cdot\log(\log(n)))}).$$

We calculated the approximate running time required to factor the modulus with different sizes. This time is measured in MIPS which means one million instructions per second. MIPS year represents the number of instructions processed for 1 year using MIPS scale. We also use MIPS minute, hour and day to represent the number of instructions processed in 1 m, h and day, respectively, using MIPS scale. The prime numbers have been generated in MAPLE 13 (Table 2). It can be seen that using modulus of size 182 bits for RSA is safe in our scheme by assuming that MANET application life is not longer than 15 continuous hours. However, in some cases when MANET application life is longer than this time, RSA keys should be revocated and updated each 15 h to ensure that the attacker will not be able to factor the modulus during MANET running life. As a result, using such RSA modulus size provides fast and secure cryptography operations for MANET.

3.6 New DIDRSA Nodes Participation

In MANET, there is always possibility for some $\mathrm{DIDRSA_s}$ nodes to be unavailable for some reasons (battery life expired, getting out of range,…). To increase the fault tolerance, DIDRSA scheme gives the possibility for the other nodes to join the DIDRSA service. When any node joins the network, it requests to be one of the DIDRSA nodes. It sends the request to t $\mathrm{DIDRSA_s}$ nodes. If the new node is qualified to act as DIDRSA using side channel (such as phone call, physical contact,…, etc): the following conditions are required for each $\mathrm{DIDRSA_i}$ to check:

- The requester node should be registered at least in t DIDRSA nodes as a trusted node.
- It should have an extra memory to enable the other nodes to be registered by this node.
- The battery life of this node should be long-lasting enough to serve the other nodes.

If the previous conditions are fulfilled, the following steps are taken:

1. The new $\mathrm{DIDRSA_w}$ node broadcasts a status message $\mathrm{DIDRSA_{ST}}$ indicating that it is now an DIDRSA node.
2. Each node X receiving the message sends a service check request $\mathrm{DIDRSA_{CHK}}$ to the existing $\mathrm{DIDRSA_s}$. The request includes the identity of $\mathrm{DIDRSA_w}$ and the value P_{d_x}

3. When the request is received by $\mathrm{DIDRSA_i}$, it checks if $\mathrm{DIDRSA_w}$ is registered as DIDRSA node. If yes, it signcrypts $\mathrm{DIDRSA_w}$ public key as

$$c = F_1(e_w) \oplus H_2(g_{ID_i})$$

where

$$g_{ID_i} = \hat{e}(Q_{ID_x}, P_{d_x} \cdot Q_{ID_i})^{d_i}$$

Then $\mathrm{DIDRSA_i}$ sends ⟨U, C, W, Y⟩ to X. Where $U = P_{d_i}$, $W = e_w \cdot P$, $Y = N_w \oplus H_3(e_w)$. Then it sends the signcrypted message to the requester node.
4. Node x gets and verifies the public key as follows:

$$e_w = F_2(C \oplus H_2(\hat{e}(d_X \cdot Q_{ID_X}, P_{d_i} \cdot Q_{ID_i})))$$

Then it computes $N_w = Y \oplus H_3(e_w)$ and tests that $W = e_w \cdot P$. If not, it rejects the DIDRSA_i response.
5. Upon receiving t valid responses, node X registers $\mathrm{DIDRSA_w}$ as a trusted DIDRSA node.

4 Efficiency and Performance Evaluation

4.1 Efficiency

4.1.1 Computation Efficiency

The pairing operations are expensive comparing with the modular exponentiation and scalar multiplication operations. However, our scheme requires pairing operations only for transferring the public keys. After that, RSA modular exponentiation operations with short exponents are used for data authentication. The other pairing-based IBE schemes [18–20] require pairing operations for every data message authentication. The comparison between the existing schemes and DIDRSA scheme according to the number of pairing, modular exponentiation and scalar multiplication operations required to authenticate 20 messages in one-to one node scenario is shown in Table 3. It shows that our scheme requires less pairing operations than the other schemes.

4.1.2 Memory Efficiency

MANET nodes are usually some kinds of mobile phone, wireless handset or PDA,…, etc. Memory in such devices is not only limited but also requires energy to store or retrieve data. As a result, an efficient usage of memory is required for such devices.

1. RSA memory requirements: Memory specifications are determined by the type of node. DIDRSA servers should have enough memory to store the public keys of the trusted nodes. According to our scheme RSA key sizes, 1 kB free memory is enough to store 3,000 nodes public keys. The non-DIDRSA server node store only the public keys of nodes that it communicates often with them. Fifty byte free memory is enough to store the public keys of 100 nodes. The largest amount of memory RSA consumes is that required to perform the modular exponentiation operations. We adopt using Montgomery Reduction to reduce this memory consumption. Montgomery Reduction allows calculating the modular arithmetic with only two modular reductions instead of repeatedly multiplying the base by itself b times (where b is the exponent), each time reducing the result modulo the modulus N. The detail of this technique can be found in [39].

Table 3 Comparison of die pairing-based IBE schemes

Operation	DIDRSA	BAEK [18]	Daza [20]	IB-KEM [19]
Pairing operations	$1 + t$	$20(n + 13)$	$20(n + 13 + t' - t'')$	44
Modular exponentiation	21	$20(2 + t)$	$20(2 + t + t'.t(t + l)/2)$	0
Scalar multiplication	$3(l + t)$	$20((t - l)n + 3)$	$20(t - l)n + 3)$	$20(4n + 3t + 7)$

Table 4 Simulation parameters

Parameter	Value
Binning mode	SPATIAL-HIER
Binning degree	5
Start time	30 s
Resolution time	30 s
Random seed	0
Bitrate for CBR traffic	2,048 bit
Cbr packet size	1,024 bit
Send rate	1.0
mac protocol	MAC-802-11
Frequency	2.4 GHz
Band width	11Mb/s
Transmission strength	15.0 dBm
Antenna gain	1.0 dB
Radio reception sensitivity	−91 dBm
Radio reception threshold	−81 dBm
Ambient noise	0.0
Threshold signal-to-noise ratio	10.0
Number of channels	6
Default channel	1
Radio-mode channel switch	2

2. Pairing cryptography memory requirement: The pairing cryptography operations are more expensive than RSA operations in terms of memory consumptions. However, these operations are not used as often as RSA operations in our scheme. Ate pairing has been chosen instead of Tate pairing since it reduces the loop in Miller algorithm to the half [24]. Four kilo byte free memory is enough to run pairing cryptography operations [40]. Since the DIDRSA nodes are going to run such operations more than the normal nodes, we allocate more free memory for such nodes (at least 4 kB RAM extra for each).

4.2 Performance Evaluation

In this section, we implement the proposed DIDRSA scheme using simulation approach. Our main goal is to make the authentication processes faster and to reduce the communication overhead. Therefore, we focus on measuring the time taken by the primitive cryptography operations, the packet overhead and the success ratio. We randomly choose different network sizes with different parameters (n, t).

4.2.1 Simulation Setup

We have implemented DIDRSA using JIST/SWANS, a java-based MANET simulation [41] on an Intel Pentium Dual Desktop (1.86 GHz processor, 1.99 GHz RAM). All the paring primitives are built using Secure SMS library [42]. The RSA primitive operations are built using RSA-CRT library. RSA key sizes are chosen such that $(q = 50, N = 200)$. $q > N^{0.486}$.

The Elliptic curves used here are

$$\text{MNT4E} : y^2 = x^3 - 3x + \text{b}$$

The quadratic twist is

$$\text{MNT4}'(F_P^2) : y'^2 = x'^3 - 3.V^2.X' - \text{b}.V^3$$

The pairing fields’ size is 176 bit. The x coordinator is fixed to -1, while y coordinator is passed to the pairing functions. The master secret key is used randomly with 176 bit length. The nodes are moving in a 700 * 600 m^2 square.

The initial placements of nodes are set randomly. We compare the scheme performance under three routing protocols : AODV, GPSR, and DSR. The other simulation parameters are shown in Table 4.

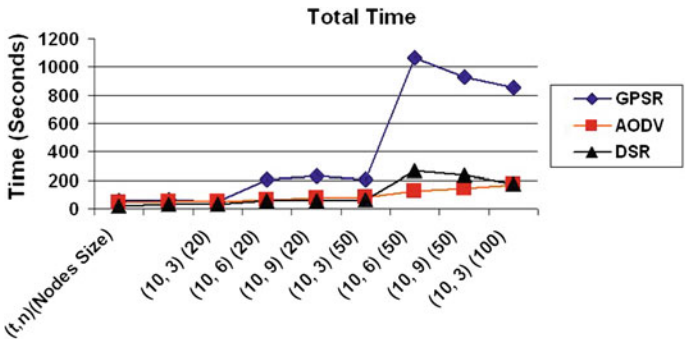

Fig. 1 The total time of running the simulation

Table 5 Time of primitive operator is for DIDRSA

Operation	Time (ms)
Public key encryption time	2.901486
Public key decryption time	50.99112
Scalar multiplication time in G_1	17
Pairing time	36

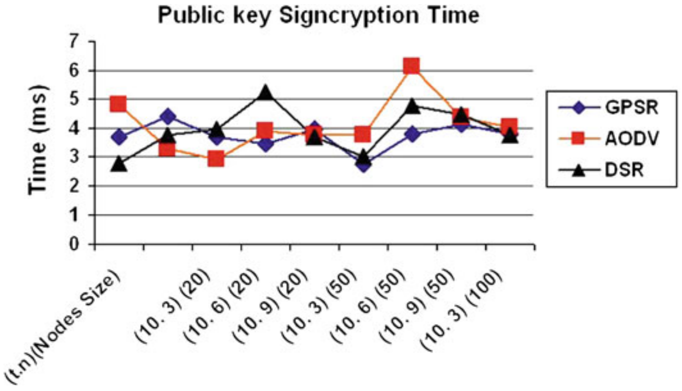

Fig. 2 The time of public key signcryption operations

4.2.2 Time Cost

We calculated the time taken by sending 400 messages using random transmitters and receivers with different parameters (n, t) (network size), Fig. 1. For fixed network size, the larger the threshold t, the more is the time required to finish the process because more neighbors are required to participate in the security process. Furthermore, the larger the network size, the more is the time needed since more number of hops between the sources and destinations is required (Table 5).

We also calculated the time taken by the public key signcryption/verification operations (Figs. 2, 3) and the data encryption/decryption operations (Figs. 4, 5).

Furthermore, Table 6 shows that our scheme operations are faster than the existing RSA operations.

4.2.3 Success Ratio

The success ratio refers to the number of messages that have been secured and received successfully. It is calculated as:

$$\text{Success ratio} = \text{MsgRvc}/\text{MsgSent}$$

where MsgSent is the number of messages that have been encrypted/signed and sent successfully. MsgRcv is the number of messages that have been received and decrypted/verified successfully. Figure 6 shows that the

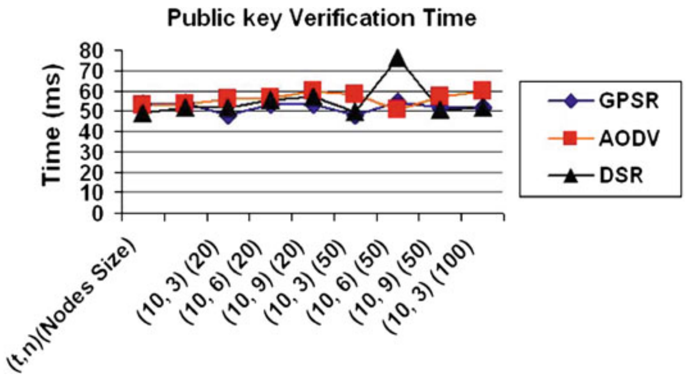

Fig. 3 The time of public key verification operations

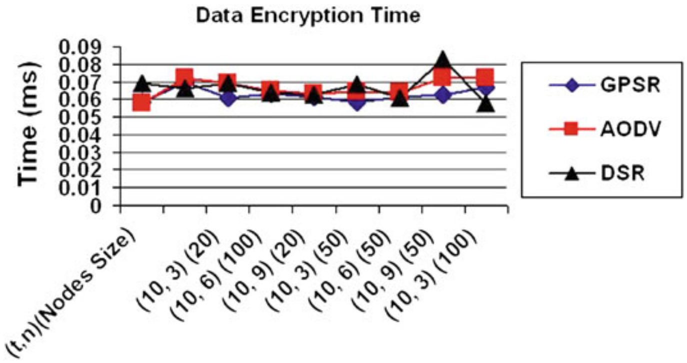

Fig. 4 The time of data encryption operations

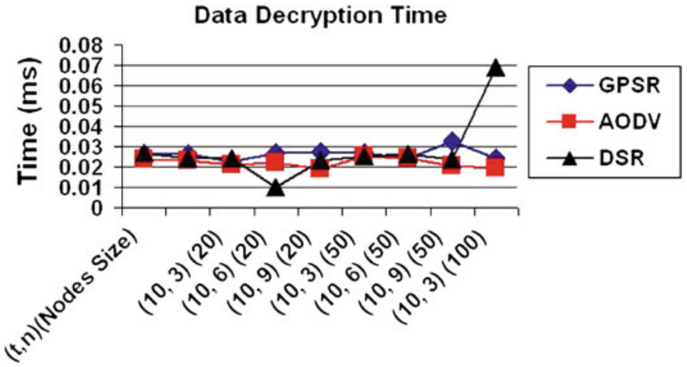

Fig. 5 The time of data decryption operations

Table 6 Comparision between DIDRSA and normal RSA primitive operations

Operation	DIDRSA Time (ms)	Normal RSA Time (ms)
RSA key generation	2.437181	1589.450335
RSA encryption/verification	0.065371	5.341461
RSA decryption/signing	0.029054	0.080736

success ratio is much better when using AODV protocol. That makes our scheme more compatible with this protocol than the other routing protocols (GPSR and DSR) (Fig. 6).

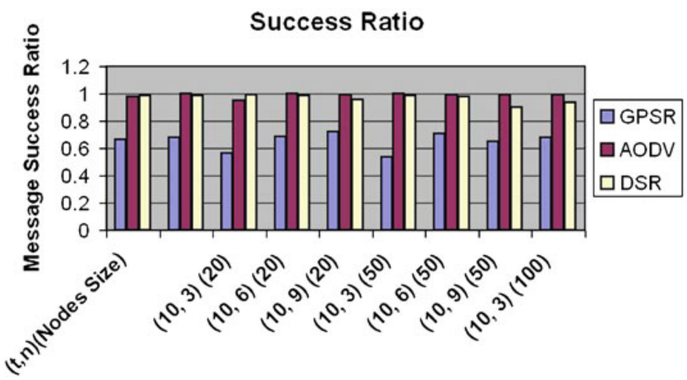

Fig. 6 The success ratio for 400 secured data messages

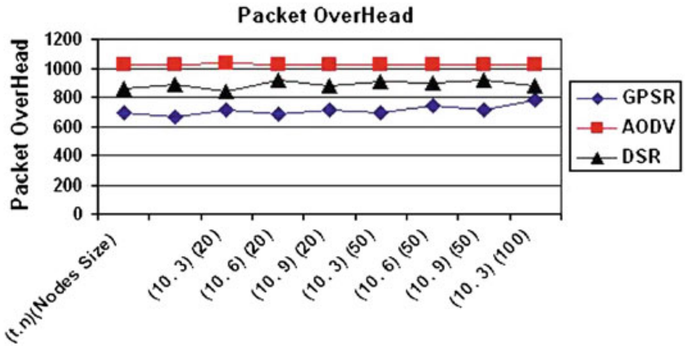

Fig. 7 The overhead caused by the security messages

4.2.4 Packet Overhead

In this paper, we are not measuring the overhead caused by the routing protocol messages. We focus on the overhead caused by the security messages which includes:

- Signcryption messages.
- The secured data messages.

The overhead is more in the starting time of running the simulations. In this duration, the nodes run signcryption operations to determine the trusted and untrusted nodes. Then, nodes send signcryption messages just when a new node joins the network or in the process of revoking some public keys from the network. On other hand, the overhead caused by the signcryption messages results when a transmitter wants to communicate with a destination for the first time. The total overhead caused by the signcryption messages can be calculated as:

$$\text{pksMsg} = \text{numTrans} \times \text{numRec} \times \text{DIDRSA}_\text{s}$$

where numTrans is the number of transmitters, numRec is the number of receivers, and DIDRSA_s is the number of DIDRSA servers.

The packet overhead is less when using our scheme with GPSR comparing to the other two routing protocols (DSR and AODV), Fig. 7.

4.3 Different Scenarios Evaluation

We have also evaluated our scheme in three different scenarios: Faculty, University and city scenarios. In all these scenarios, 5 nodes send 400 messages to other 5 nodes. The senders and receivers are chosen randomly.

Table 7 Faculty scenario: 200 nodes, $l \times l$ km

Parameter	Value
Data encryption/verification (ms)	0.060902
Data decryption/signing time (ms)	0.023187
Success ratio	0.975845
Total time (s)	480
Packet overhead	1027
Public key signcryption time (ms)	3.719188
public key verification time (ms)	50.08011

Table 8 University scenario: 500 nodes, 2 × 2 km

Parameter	Value
Data encryption/verification time (ms)	0.058666
Data decryption/signing time (ms)	0.021511
Success ratio	0.9846626
Total time (s)	1594
Packet overhead	965
Public key sigiicryption time (ms)	4.087391
public key verification time (ms)	51.753784

Table 9 City scenario: 1,000 nodes, 5 × 5 km

Parameter	Value
Data encryption/verification time fins)	0.07487
Data decryption/signing time (ms)	0.02179
Success ratio	0.8C290324
Total time (s)	2651
Packet overhead	754
Public key signcryption time (ms)	2.643G321
Public key verification time (ms)	51.564655

4.3.1 Case Study, Faculty Scenario

Two hundred persons each carrying a PDA are moving in 1 × 1 km in faculty building. The nodes are moving in a speed ranges from 2 to 5 km/h. The initial placement of nodes is random. The threshold cryptography is set to 6. The results are shown in Table 7.

4.3.2 Case Study, University Scenario

Five hundred persons each carrying a PDA are moving in 2 × 2 km in university area. The nodes are moving in a speed ranging from 4 to 8 (min speed, max speed) km/h. The nodes are initially placed in a grid form. The threshold cryptography is set to 6, Table 8.

4.3.3 Case Study, City scenario

Thousand nodes are moving in 5 × 5 km in city area in a speed ranging from 5 to 20 km/h. The initial placement of nodes is random. The threshold cryptography used here is 3. The results show that the success ratio is less in the city scenario than the faculty and university scenario, because of the loss in some security messages when the nodes are far from each others. Furthermore, the packet overhead is less in the city scenario because of using less threshold cryptography ($t = 3$). On the other hand, the simulation requires more time when the number of nodes is more (Table 9).

5 Conclusion and Future Work

Authentication and data privacy are challenging issues in MANET. This paper presented DIDRSA, a secure and lightweight authentication and encryption scheme for MANET. Public keys are secured for better performance.

Identity keys are used to secure the public keys. DIDRSA provides a safe way to use RSA speeding techniques effectively. Our scheme requires less computational power and memory comparing with the existing schemes. In addition, we give guidelines for choosing the RSA parameters in such a way that makes it secure and fast in the same time. The performance of our scheme is proved using simulation approach.

References

1. Guarnera, M.; Villari, M.; Zaia, A.; Puliafito, A.: Manet: Possible applications with pda in wireless imaging environment. In: 13th IEEE international symposium on personal, indoor and mobile radio communications, pp 2394–2398 (2002)
2. Zhou, L.; Haas, Z.: Securing ad hoc networks. Netw. IEEE. **13**(6), 24–30 (1999)
3. Kong, J.; Zerfos, P.; Luo, H.; Lu, S.; Zhang, L.: Providing robust and ubiquitous security support for mobile ad-hoc networks. In: International conference on network protocols, Department of Computer Science, California University, Los Angeles, pp. 251–260 (2001)
4. Capkun, S.; Buttyan, L.; Hubaux, J.-P.: Self-organized public-key management for mobile ad hoc networks. IEEE Trans. Mobile Comput. **2**(1), 52–64 (2003)
5. Yi, S.; Kravets, R.: Moca: mobile certificate authority for wireless ad hoc networks. In: 2nd Annual PKI Research Workshop Program (PKI 03), pp. 65–79 (2003)
6. Zhou, L.; Schneider, F.; Van Renesse, R.: Coca: a secure distributed online certification authority. In: Foundations of Intrusion Tolerant Systems, 2003 (Organically Assured and Survivable Information Systems), pp. 152–191 (2003)
7. Levent, E.; Chavan, N.J.: Elliptic curve cryptography based threshold cryptography (ecc-tc) implementation for manets. IJCSNS Int. J. Comput. Sci. Netw. Secur. **7**(4), 48–61 (2007)
8. Bechler, M.; Hof, H.-J.; Kraft, D.; Pahlke, F.; Wolf, L.: A cluster-based security architecture for ad hoc networks, INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 4, pp. 2393–2403, March 2004
9. Hadjichristofi, G.C.; Adams, W.J.; Davis, N.J., IV.: A framework for key management in mobile ad hoc networks. In: International Conference on Information Technology: Coding and Computing, ITCC, vol. 2, pp. 568–573. New York 10016-5997 (2005)
10. Weimerskirch, A.; Thonet, G.: A distributed light-weight authentication model for ad-hoc networks. In: The 4th International Conference on Information Security and Cryptology (ICISC 2001), pp. 341–354. London, Springer (2001)
11. Marias, G.F.; Papapanagiotou, K.; Tsetsos, V.; Sekkas, O.; Georgiadis, P.: Integrating a trust framework with a distributed certificate validation scheme for manets. EURASIP J. Wirel. Commun. Netw. **2006**(2), 77–77 (2006)
12. Wang, G.; Wang, Q.; Cao, J.; Guo, M.: An effective trust establishment scheme for authentication in mobile ad-hoc networks. In: CIT 2007, 7th IEEE International Conference on Computer and Information Technology, pp. 749–754. Piscataway (2007)
13. Datta, A.; Quarteroni, S.; Aberer, K.: Autonomous gossiping: a self organizing epidemic algorithm for selective information dissemination in wireless mobile Ad-Hoc networks. In: ICSNW (2004)
14. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Proceedings of CRYPTO 84 on Advances in cryptology, pp. 47–53. Springer, Inc., New York (1985)
15. Eissa, T.; Razak, S.; Ngadi, M.: Authentication techniques in manet. In: Student Conference on Research and Development (SCOReD 2008), pp. 130–134. March 2008
16. Deng, H.; Mukherjee, A.; Agrawal, D.P.: Threshold and identity-based key management and authentication for wireless ad hoc networks. In: International Conference on Information Technology: Coding Computing, ITCC, vol. 1, pp. 107–111. Piscataway (2004)
17. Boneh, D.; Franklin, M.: Identity-based encryption from the weil pairing. SIAM J. Comput. **32**(3), 586–615 (2003)
18. Baek, J.; Zheng, Y.: Identity-based threshold decryption. In: Proceedings of PKC04, LNCS 2947, pp. 262–276. Springer, London (2004)
19. Kiltz, E.; Galindo, D.: Direct chosen-ciphertext secure identity-based key encapsulation without random oracles. Theor. Comput. Sci. **410**(47–49), 5093–5111 (2009)
20. Daza, V.; Herranz, J.; Morillo, P.; Rifols, C.: Cryptographic techniques for mobile ad-hoc networks. Comput. Networks. **51**(18), 4938–4950 (2007)
21. Eissa, T.; Razak, S.; Ngadi, M.: Enhancing manet security using secret public keys. In: 2009 International Conference on Future Networks, pp. 130–134. March 2009
22. Barreto, P.S.L.M.; Kim, H.Y.; Lynn, B.; Scott, M.: Efficient algorithms for pairing-based cryptosystems. In: CRYPTO '02 Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology, pp. 354–368. London, Springer (2002)
23. Thomas, C.E.L.R.L.R.; Cormen, H.; Stein, C.: Introduction to Algorithms, vol. second edn. MIT Press and McGraw-Hill (2001)
24. Galbraith, S.D.; Paterson, K.G.; Smart, N.P.: Pairings for cryptographers. Discret. Appl. Math. **156**(16), 3113–3121 (2008)
25. Menezes, A.J.; Vanstone, S.A.; Oorschot, P.C.V.: Handbook of Applied Cryptography. CRC Press, Inc., Boca Raton (1996)
26. Boneh, D.: Twenty years of attacks on the rsa cryptosystem. Notices AMS. **46**, 203–213 (1999)
27. Wiener, M.: Cryptanalysis of short rsa secret exponents. IEEE Trans. Inf. Theory **36**(3), 553–558 (1990)
28. Boneh, D.; Durfee, G.: Cryptanalysis of rsa with private key d less than n 0.292. IEEE Trans. Inf. Theory. **46**(4), 1339–1349 (2000)
29. May, A.: Cryptanalysis of unbalanced rsa with small crt-exponent. In: CRYPTO'02 Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology, pp. 242–256. Springer, London (2002)

30. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: CRYPTO ’96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, pp. 104–113. Springer, London (1996)
31. Lenstra, A.K.; Lenstra, H.W. Jr.: Algorithms in number theory, pp. 673–715 (1990)
32. Boneh, Y.F.D.; Durfee, G.: An attack on rsa given a fraction of the private key bits. In: Advances in Cryptology, Asiacrypt98, LNCS, vol. 1514, pp. 25–34 (1998)
33. Antonov, P.; Antonova, V.: Development of the attack against rsa with low public exponent and related messages. In: CompSysTech ’07 Proceedings of the 2007 international conference on computer systems and technologies, pp. 1–8. ACM, New York (2007)
34. Hastad, J.: Solving simultaneous modular equations of low degree. SIAM J. Comput. Arch. **17**(2), 336–341 (1988)
35. Coppersmith, D.; Franklin, M.; Patarin, J.; Reiter, M.: Low-exponent rsa with related messages, pp. 1–9. Springer, Berlin (1996)
36. Coppersmith, D.: Small solutions to polynomial equations, and low exponent rsa vulnerabilities. J. Cryptol. **10**(4), 233–260 (1997)
37. Pomerance, C.: The quadratic sieve factoring algorithm. In: Proceedings of the EURO-CRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques, pp. 169–182. Springer, Inc., New York (1985)
38. Gordon, D.M.: Discrete logarithms in gf(p) using the number field sieve. SIAM J. Discret. Math. **6**(1), 124–138 (1993)
39. Montgomery, P.L.: Modular multiplication without trial division. Am. Math. Soc. **44**(170) (1985)
40. Zhou, Z.; Huang, D.: Computing cryptographic pairing in sensors. SIGBED Rev. **5**(1), 1–2 (2008)
41. Barr, R.: Jist java in simulation time user guide (2004). http://www.isi.edu/nsnam/ns/
42. Rodrigopitanga, z.: Geovandro. Pereira, secure-sms (2008). http://code.google.com/p/secure-sms/