TLPKA: Pairwise Key Agreement Based on the Geometrical Property of the Tangent Line in Ad Hoc Network Systems

Chung-Wei Chen
Institute of Communications Engineering National Tsing Hua University, Hsinchu, Taiwan 300

Shiuh-Jeng Wang*
Department of Information Management Central Police University, Taoyuan County, Taiwan 333

Yuh-Ren Tsai
Institute of Communications Engineering National Tsing Hua University, Hsinchu, Taiwan 300

Abstract

In this paper, we propose a new pairwise key agreement, TLPKA (Tangent Line Pairwise Key Agreement), for establishing a secure link between members in an ad hoc network. With pairwise keys, data transmitted in ad hoc networks can be protected from the eavesdropping of outsiders. These pairwise keys can also be used to authenticate members in ad hoc networks. In addition, due to the property of uniqueness of pairwise keys, it also provides non-repudiation for data transmitting. In our scheme, we propose a new idea for the design of pairwise key agreements in an ad hoc network. Our scheme is based on the geometrical property of the tangent line and the mechanisms in our scheme are inspired by the Shamir secret sharing scheme. Our schemes also provide reciprocal authentication among node communicants. The robustness of our scheme is based on the DH problem and the indefinite solutions in coordinate geometry. In addition, we compare our scheme with other schemes. Although there is a higher overhead associated with our scheme, it maintains a higher security level than the other schemes.

1. Introduction

The ad hoc network is a kind of network that operates over a distributed architecture. Each node in the ad hoc network acts as a server and a client as well. When a node wants to send some information to another node, it needs to go through other nodes between it and the receiver. Therefore, each node needs to trust each other. However, this hypothetical situation causes some security flaws. In addition, dynamic topologies, bandwidth-constraint, variable capacity links, energy-constrained operation, and limited physical security also become the challenges of security in ad hoc networks [5] because the security system may not be able to figure out whether an attack is really happening or not [15].

A conventional key agreement with centralized architecture like the one developed by Kerberos [11], is not suitable due to the hostile environments involved. The key distribution center (KDC) in such architecture plays an important role arranging and managing the pairwise key or session key for members in an ad hoc network. Therefore, it is vulnerable to any compromise of the KDC. In addition, the security of such a system would not work correctly if there was any malfunction in the KDC, like having a non line-of-sight path and

* Whom correspondence: **Shiuh-Jeng Wang**, Email: sjwang@mail.cpu.edu.tw

noise in the wireless channel. Therefore, a new key agreement in an ad hoc network is necessary.

In order to solve the problems associated with the environmental and security flaws of ad hoc networks, engineers have proposed many mechanisms for providing secure communications in ad hoc networks since the early 1990s [9][16]. Early days, master key predistribution is proposed for ad hoc networks. In master key predistribution, the central authority predistributes a master key into each node. Each node will set up a secure link so as to communicate with each other. However, the whole system will crash if just one node is compromised. Because each node holds the same master key, an intruder can get the master key even if he or she just captures one node and draws the master key from the memory or hard disk. In order to improve the master key predistribution scheme, the pairwise key predistribution scheme is proposed. However, it will require a lot of memory to store the pairwise key if the central authority predistributes pairwise keys that are shared with other nodes into a node. In 2005, Du *et al*. [7] proposed reducing the storage overhead of the pairwise key pre-distribution. However, Du *et al*.'s security flaw is that the secret key pre-distributed into each node is chosen randomly. In other words, the secret pre-distributed into each node is not unique. The intruder can impersonate a node if he or she compromises the same secret that was pre-distributed into any other node before.

In this paper, we propose a new pairwise key agreement, TLPKA (Tangent Line Pairwise Key Agreement), inspired by the geometrical property of the tangent line and we combine this property with Shamir's secret sharing scheme. This is a new take on designing a pairwise key agreement in ad hoc networks. In our key agreement, each node is capable of authenticating other nodes and establishes a pairwise key. In addition, pairwise keys would refresh each time in order to prevent the chosen-cipher text attack and the human force attack.

The remainder of this paper is organized as follows: Section 2 introduces related work that has been proposed for the security of an ad hoc network. In Section 3 we put forward the details of TLPKA. In Section 4, we analyze the security properties of TLPKA and use modified Buttyán *et al*.'s logic method [2] to prove the validity of TLPKA. In Section 5, we present the comparison between TLPKA and other schemes. Finally, in Section 6, we conclude our work and present future work to be done on our scheme.

2. Related work

A variety of key management mechanisms for the ad hoc network have been proposed over the last several years. We introduce these previous studies briefly for their resilience against node compromise.

2.1. Master key pre-distribution schemes

In this scheme, the central authority only distributes a master key to each node. With this master key, each node can recognize one another if the other node is also a member of the same ad hoc network. However, the intruder only needs to compromise one node to crash the whole system.

To mitigate these vulnerabilities, Zhu *et al.* [19] proposed transitory master key schemes to generate pairwise keys between each node. However, the entire system can still be corrupted if the master key is compromised.

2.2. Public key cryptosystem

The central authority pre-distributes to each node a particular public/secret key pair. This public/secret key pair can be used to authenticate other nodes that want to set up a secure link. However, public key systems are inefficient because the computational loads of public key systems are too heavy for the nodes in an ad hoc network. The speed of en/decryption within a symmetric key system is obviously faster than within a public key system. For example, in [18], it is known that DES is 100 times faster than RSA in software and 1000 times faster in hardware.

2.3. Pairwise key pre-distribution scheme

The central authority gives each node pairwise keys that are shared with other nodes. With the pairwise key, a sender and a receiver can authenticate each other and set up a secure link. However, the amount of data associated with the pairwise key that would need to be stored in the node would use a lot of memory. To mitigate this vulnerability, [3] [8] [13] proposed the random key pre-distribution technique to reduce the memory used to store the pairwise key. However, there is a tradeoff between the security and the connection. If a higher security level is required, fewer pairwise keys are predistributed into a node. However, the connection will reduce dramatically. Therefore, Du *et al*. [7] upgrades the connection rate by modifying the random key pre-distribution and Blom scheme. However, an intruder can compromise some nodes so as to impersonate other normally functioning nodes. This security flaw may actually increase the probability of an intruder being able to successfully attack the ad hoc network.

3. Our scheme

In this section, we illustrate the procedure of TLPKA. First, we demonstrate the threat model of TLPKA, and then we present the agreement in TLPKA. In TLPKA, our scheme has two parts. One part is the **Source Node**; another part is the **Destination Node**. In the **Source Node** part**,** we describe the procedure of source nodes who requests to set up a secure link. In the **Destination Node** part, there are the procedures what the node requested to set up a secure link to do.

3.1. The Threat Model

For the environment of TLPKA, we assume that the attack action as follows occurs in order to try to compromise TLPKA.

1). If the intruder compromises the nodes in TLPKA; all of the secrets stored in the compromised node would be derived by the intruder.
2). The nodes in an ad hoc network will not act in collusion to induce the secret generated by the central authority.
3). By compromising a node in an ad hoc network, the intruder will clone nodes, and introduce them in several different areas, known as the Sybil attack [6].
4). The intruder has the ability to eavesdrop on messages exchanged between the sender and receiver.

5). No attacks occur inside the central authority, e.g., the central authority wouldn't be compromised and the predistributed secret will not be eavesdropped within the central authority.

3.2. Our Proposed Scheme: TLPKA

TLPKA is based on a geometric property of the tangent line: a tangent line is tangent to the curve of certain function at a unique point. And utilizing the concept of the existence of a unique point of intersection between two tangent lines, a unique pairwise key can be established between the sender and receiver.

In TLPKA, there are two separated parts: the **Source Node** part and the **Destination Node** part:

Firstly, we define the factors and parameters in the notation as follows:

NOTATIONS

Symbol	Meaning
N	The number of members.
P	The prime number for the characteristic of the finite field, Z_p.
ID_i	The unique identification of member i, $i=\{1,2,3,...,n\}$, which is public.
F	A polynomial generated by the central authority, of degree t, which is kept secret.
T	The degree of f, the critical value of the system, which is public in this system.
a_j	The coefficient of x^j in f, where $j=\{1,2,3,...,t\}$, which is kept secret.
y_i	The secret used to authenticate member i, $f(ID_i)$, which is always kept secret.
G	The primitive root, which is publicly known in this system.
s_j	$s_j \equiv g^{a_j} \pmod p$, which is public.
l_i	The tangent line of f passes through (ID_i, y_i), $i=\{1,2,3,...,n\}$, which is kept secret.
$E_{k_{ij}}\{m\}$	The cipher text of the plaintext *m* encrypted using the pairwise key of node *i* and node *j*.
$D_{k_{ij}}\{m\}$	The decryption of the cipher text *m*, using the pairwise key of node *i* and node *j*.

Before deployment, the central authority generates a polynomial, $f(x)$ in the beginning, where the degree of $f(x)$ is t. For each node, the central authority individually assigns each a unique identification. According to each one's identification, the system will assign tangent lines of $f(x)$. With these tangent lines, members can authenticate each other and establish pairwise keys amongst themselves. Taking member *i* as an example, one may observe he/she has a unique identification, ID_i. For this member *i*, the central authority assigns line $l_i=a_ix+b_i \pmod p$ to this member *i*, where Line l_i is tangent to $(ID_i, f(ID_i))$. After pre-distributing l_i for each node, the central authority computes $s_i \equiv g^{f(ID_i)} \pmod p$, where $i=\{1,2,3,...,n\}$ and broadcasts s_i over the entire ad hoc network.

After finishing the predistribution of the tangent line to a member, that member is then deployed into the ad hoc network. After all members of the ad hoc network are deployed, members use the tangent line pre-distributed before deployment to set up a secure link with others. The source node and the destination node of this secure link will process the phase of TLPKA as follows. We now assume that node *m* is the source node and node *q* is the destination node for setting up a secure link. Node *m* first chooses a nonce d_m randomly, and

then node m carries out the exponentiation operation of g to the power of d_m. Node m then multiplies g^{d_m} by s_q. The procedure for this computation is outlined as follows:

$$g^{d_m} * s_q \equiv \rho_m (\mathrm{mod}\, p) \quad (1)$$

Node m then sends ρ_m to node q. When node q receives ρ_m, it then uses the tangent line l_q pre-distributed by the central authority to derive $f(ID_q)$ and multiplies ρ_m with $g^{-f(ID_q)}$. We outline the procedure of this computation as follows in (2).

$$\rho_m * g^{-(a_q ID_q + b_q)} \equiv \tau_m (\mathrm{mod}\, p) \quad (2)$$

After node q derives $\tau_m \bmod p$, it chooses a nonce d_k randomly and does a computation similar to that of (1). We demonstrate this procedure in (3).

$$g^{d_q} * s_m \equiv \rho_q (\mathrm{mod}\, p) \quad (3)$$

After node m receives ρ_q, node m computes $f(ID_m)$ by way of the tangent line l_m pre-distributed by the central authority. Afterwards, node m multiplies ρ_q times $g^{-f(ID_m)}$. We show this procedure as follows in (4).

$$\rho_q * g^{-(a_m ID_m + b_m)} \equiv \tau_q (\mathrm{mod}\, p) \quad (4)$$

After node m derives $\tau_q \bmod p$, it generates a parallel line, l_m', having a distance d_m to its tangent line l_m. Node m next sends l_m' to node q. After node q receives l_m', node q generates a parallel line l_q', which is parallel to its tangent line l_q and computes the intersection point of l_q and l_m'. Node q then computes the intersection point of l_q' and l_m' as well. Node q next computes the length L_q between these two section points. Finally, Node q computes the exponential operation, τ_m to the power of L_q. Using this result as the symmetric key of the symmetric key cryptosystem, k_{qm}, to encrypt challenge c_q, node q then sends l_q' and the cipher text $E_{k_{qm}}\{c_q\}$ to node m. After node m receives l_q' and $E_{k_{qm}}\{c_q\}$, it computes the intersection points of l_q' and l_m'. It also computes the intersection point of l_q' and l_m. Node m next utilizes the length of line L_m between these two intersection points to compute the exponential operation of τ_q to the power of L_m. Using the result as the symmetric key, k_{mq}, to carry out the encryption/decryption procedure in the symmetric key cryptosystem, node m decrypts $E_{k_{mq}}\{c_q\}$ to derive c_q'. Sequentially, node m chooses its challenge c_m randomly and encrypts c_m using k_{mq}. Node m then sends cipher text $E_{k_{mq}}\{c_m, c_q'+1\}$ to node q. After node q receives this cipher text, it decrypts it and gets c_m' and $c_q'+1$. Node q checks to see if $c_q'+1$ equals c_q+1. If the validity of $c_q'+1$ is verified, node q identifies node m as a certified member. Node q next encrypts $c_m'+1$ utilizing k_{qm} and sends this cipher text back to node m. Node m then check the validity of $c_m'+1$ as well. If $c_m'+1$ equals c_m+1, node m identifies node q as a verified member. Finally, the symmetric keys, k_{mq} and k_{qm} which node m and node q have computed individually are confirmed as the pairwise keys that they use for encrypting/decrypting data transmissions. We show the data flow between the source node and destination node in Fig. 1.

We describe our scheme for the two separate parts, the **Source Node** and **Destination Node** parts, briefly shown as follows(We assume that the source node is node i and destination node is node j):

Source Node

Step1. Source node i chooses a random number d_i.

Step2. According to the destination node j's ID, ID_j, node i computes ρ_i.

Step3. Node i sends ρ_i to node j.

Step4. After ρ_j is received from node j, node i generates a parallel line l_i' to the tangent line l_i predistributed by the central authority.

Step5. Node *i* sends l_i' to the destination node *j*.
Step6. After receiving $E_{k_{ji}}\{c_j\}$ and l_j' from node *j*, node *i* computes k_{ij} by using l_i , l_i', and l_j'.
Step7. Node *i* decrypts $E_{k_{ji}}\{c_j\}$ using k_{ij} and derives c_j'.
Step8. Node *i* generates a nonce, c_i.
Step9. Node *i* encrypts c_i and $c_j'+1$.
Step10. Node *i* sends $E_{k_{ij}}\{c_i, c_j'+1\}$ to node *j*.
Step11. After receiving $E_{k_{ji}}\{c_i'+1\}$, node *i* decrypts this cipher text and checks to see if $c_i'+1$ equals c_i+1.

Destination node

Step1. After receiving source node *i*'s ID, ID_i, destination node *j* generates a random number d_j.
Step2. Node *j* computes ρ_j.
Step3. Node *j* sends ρ_j to node *i*.
Step4. After receiving line l_i' from node *i*, node *j* computes k_{ji} using line l_i', l_j, and l_j'.
Step5. Node *j* generates a nonce c_j and encrypts it using k_{ji}.
Step6. Node *j* sends the cipher text, $E_{k_{ji}}\{c_j\}$, to node *i*.
Step7. After receiving $E_{k_{ij}}\{c_i, c_j'+1\}$, node *j* decrypts it to get c_i' and $c_j'+1$. Afterwards, node *j* checks to see if $c_j'+1$ equals c_j+1.
Step8. Node *j* adds 1 to c_i' derived in Step 7 and encrypts it with k_{ji}.
Step9. Node *j* sends cipher text, $E_{k_{ji}}\{c_i'+1\}$, back to node *i*.

4. Security Analysis

In this section, we indicate the details of the security analysis of TLPKA. Informally, if each user can get his or her session key verified, and this session key reveals no information to outsiders, TLPKA is safe. This TLPKA scheme, has two security properties: simultaneous mutual authentication and pairwise key establishment. TLPKA can also maintain its strength under the well-known impersonation attack, the dictionary attack, the known key attack, the replay attack, the man in the middle attack, and the insider attack. In section 4.1, we demonstrate the security mechanism of our proposed scheme. In section 4.2, we use the Buttyán *et al*. logic analysis method to prove the correctness of TLPKA. And we show TLPKA can resist the replay attack, the man in the middle attack, and the insider attack in sections 4.2-4.5.

4.1. Security Mechanism of TLPKA

In TLPKA, a secret polynomial is chosen and utilized by the central authority. By manipulating this secret polynomial, each member can authenticate other members. According to the Shamir secret sharing scheme, the critical value of the secret sharing system is the degree of the polynomial used to distribute and share. In TLPKA, the security mechanism is also based on the Shamir secret sharing scheme. Therefore, the security level of TLPKA is also the same as the Shamir secret sharing scheme.

The authentication mechanism of our scheme is inspired by certain geometric principles and Diffie-Hellman key agreement protocols. In Euclidean geometry, a tangent *l* of a

polynomial $f(x)$ is only tangent to the curve of $f(x)$ at a unique point on $f(x)$. Therefore, another point at which the tangent line is tangent to the curve of $f(x)$ doesn't exist. We use Fig. 2 to indicate this theorem. We then use this property to construct the authentication mechanism in our protocol. First, the system generates a polynomial $f(x)$. With this polynomial and the unique ID of each node, each node is assigned a unique tangent line to $f(x)$. This unique tangent line is used as a unique certificate.

As the source node intends to set up a pairwise key with the destination node, we demonstrate the pairwise key establishment mechanism by using the example of node m and node q that we used before. We use Fig. 3 to indicate the situation of the pairwise key establishment mechanism procedure. As node m gets the two intersection points P and S, node q gets intersection points, P and Q. From the graph below, we can see parallelogram $YPQRS$, where R is the intersection point between the tangent line, l_m, of node m and tangent line, l_q, of node q. For node m, line segment $\overline{PS}$ is derived by the two intersection points, P and S using l_m, l_m', and l_q'. In the same way, node q realizes line segment $\overline{PQ}$. After node m receives $g^{d_q} \bmod p$ from node q, it calculates $(g^{d_q})^{PS} \bmod p$, where the value of $(g^{d_q})^{PS} \bmod p$ is equal to that of $g^{YPQRS} \bmod p$. In the case of node q, it will also realize $g^{YPQRS} \bmod p$ after it calculates a similar value, $(g^{d_m})^{PQ} \bmod p$. Then node m and node q use $g^{YPQRS} \bmod p$ as their pairwise key to encrypt the data that they want to send. The reason why node m sends $g^{d_m} \bmod p$ is that node q can't deduce the tangent l_m in node m. If node q deduces l_m in node m, it may impersonate node m in order to deceive other nodes.

4.2. Logic Proof

There are some logic analysis methods proposed to prove the correctness of protocols [1][2][17]. In 1990, Ban logic [1] was proposed to analyze authentication protocol. Then Buttyán *et al*. [2] proposed their logic method in 1998. With the Buttyán *et al*. logic method, we can analyze authentication protocol in a simple manner. In this section, we use a modified Buttyán *et al*. method to analyze the correctness of the authentication property in TLPKA. Channels are the main abstractions in Buttyán *et al*. method. For a channel C, the set of readers and the set of writers of channel C are denoted by $r(C)$ and $w(C)$, respectively. And we present the notation and synthetic rules in the modified Buttyán *et al*. method as follows:

Notation and synthetic rules

The notations for the logic used are defined as follows:

- U_i, U_j: principal of source node and destination node.
- C: channel.
- X: message, which can be data or formulae or both.
- ϕ : a formula.
- $C(X)$: message X transmitted on channel C.
- $U_i < C(x)$: U_i sees $C(x)$.
- $U_i < X \mid C$: U_i sees X via C.
- $U_i < X$: U_i sees X .
- $U_i \mid\equiv X$: U_i believes X .
- $\phi_1 \rightarrow \phi_2$: ϕ_1 implies ϕ_2.
- k_{ij}: the pairwise key between U_i and U_j.

- $E_{k_{ij}}()$: the encryption using key k_{ij}.

Then we present the synthetic rule, which is used to prove the proposed protocol, and is detailed as follows.

- Synthetic rule 1: As a message, X arrives via channel C, a principal, U_i has to see $C(x)$ and be able to read C.

$$U_i \mid\equiv (U_i < X \mid C) \rightarrow U_i < C(X) \rightarrow U_i \in r(C)$$

- Synthetic rule 2: In order to believe formula ϕ, a principal U_i needs to believe a formula ϕ' and the implication that $\phi' \rightarrow \phi$.

$$U_i \mid\equiv \phi \rightarrow U_{\mathrm{i}} \mid\equiv \phi' \rightarrow U_{\mathrm{i}} \mid\equiv (\phi' \rightarrow \phi)$$

Proof for TLPKA

We assume that source node U_i wants to set up a secure link with destination node U_j. The message transmitted between them is as follows:

1) $U_i \rightarrow U_j$: ID_i, ρ_i
2) $U_j \rightarrow U_i$: ID_j, ρ_j
3) $U_i \rightarrow U_j$: l_i'
4) $U_j \rightarrow U_i$: $E_{k_{ji}}\{c_j\}, l_j'$
5) $U_i \rightarrow U_j$: $E_{k_{ij}}\{c_i, c_j'+1\}$
6) $U_j \rightarrow U_i$: $E_{k_{ji}}\{c_i'+1\}$

Because of the pairwise key establishment mechanism, we can determine $(g^{d_i})^{|L_j|} = (g^{d_j})^{|L_i|} = k_{ji} = k_{ij}$, where $|L_j|$ is the length of line segment of two intersection points of line l_i, line l_i', and line l_j', while $|L_i|$ is the length of line segment of two intersection points of line l_j, line l_j', and line l_i'. And then these messages are transmitted to their channels. We transfer these messages to the descriptions by Buttyán *et al*. method in the following manner.

1) $U_j < C_{s_j g^{-(a_j ID_j + b_j)}}(g^{d_i})$
2) $U_i < C_{s_i g^{-(a_i ID_i + b_i)}}(g^{d_j})$
3) $U_j < C(l_i')$
4) $U_i < C_{k_{ij}}(c_j)$
5) $U_j < C_{k_{ij}}(c_i, c_j)$
6) $U_i < C_{k_{ij}}(c_i)$

We also define the following assumptions:

Assumption 1: $U_j \in r(C_{s_j g^{-(a_j ID_j + b_j)}})$

U_j can read channel $C_{s_j g^{-(a_j ID_j + b_j)}}$.

Assumption 2: $U_i \in r(C_{s_i g^{-(a_i ID_i + b_i)}})$

U_i can read channel $C_{s_i g^{-(a_i ID_i + b_i)}}$.

Assumption 3: $U_i \mid\equiv (w(C_{s_j g^{-(a_j ID_j + b_j)}}) = U_j)$

U_i believes that only U_j can write channel $C_{s_j g^{-(a_j ID_j + b_j)}}$.

Assumption 4: $U_j \mid\equiv (w(C_{s_i g^{-(a_i ID_i + b_i)}}) = U_i)$

U_j believes that only U_i can write channel $C_{s_i g^{-(a_i ID_i + b_i)}}$.

Assumption 5: $U_i \in r(C_{k_{ij}})$

U_i can read channel $C_{k_{ij}}$.

Assumption 6: $U_j \in r(C_{k_{ij}})$

U_j can read channel $C_{k_{ij}}$.

Assumption 7: $U_i \mid\equiv (w(C_{k_{ij}}) = \{U_i, U_j\})$

U_i believes only U_i and U_j can write channel $C_{k_{ij}}$.

Assumption 8: $U_j \mid\equiv (w(C_{k_{ij}}) = \{U_i, U_j\})$

U_j believes only U_i and U_j can write channel $C_{k_{ij}}$.

Assumption 9:

$$U_i \mid\equiv ((U_i < c_i + 1 \mid C_{k_{ij}}) \rightarrow (U_j \mid\equiv U_i \overset{k_{ij}}{\longleftrightarrow} U_j))$$

U_i believes that U_i sees c_i+1 via channel $C_{k_{ij}}$, which implies that U_j believes U_i and U_j share $k_{i,j}$.

Assumption 10:

$$U_j \mid\equiv ((U_j < c_j + 1 \mid C_{k_{ij}}) \rightarrow (U_i \mid\equiv U_i \overset{k_{ij}}{\longleftrightarrow} U_j))$$

U_j believes that U_j sees c_j+1 via channel $C_{k_{ij}}$, which implies U_i believes that U_i and U_j share k_{ij}.

Assumption 11:

$$U_i \mid\equiv ((U_i < g^{s_j} \mid C_{s_j g^{-(a_j ID_j + b_j)}}) \rightarrow U_i \overset{k_{ij}}{\longleftrightarrow} U_j)$$

U_i believes that U_i sees g^{d_j} via channel $C_{s_j g^{-(a_j ID_j + b_j)}}$, which implies U_i believes that U_i and U_j share k_{ij}.

Assumption 12:

$$U_j \mid\equiv ((U_j < g^{s_i} \mid C_{s_i g^{-(a_i ID_i + b_i)}}) \rightarrow U_i \overset{k_{ij}}{\longleftrightarrow} U_j)$$

U_j believes that U_j sees g^{d_i} via channel $C_{s_i g^{-(a_i ID_i + b_i)}}$, which implies U_j believes that U_i and U_j share k_{ij}.

In order to prove the authentication properties of the proposed protocol, these four goals must be achieved:

Goal 1: $U_i \mid\equiv U_i \overset{k_{ij}}{\longleftrightarrow} U_j$.

Goal 2: $U_j \mid\equiv U_i \overset{k_{ij}}{\longleftrightarrow} U_j$.

Goal 3: $U_i \mid\equiv (U_j \mid\equiv U_i \overset{k_{ij}}{\longleftrightarrow} U_j)$.

Goal 4: $U_j \mid\equiv (U_i \mid\equiv U_i \overset{k_{ij}}{\longleftrightarrow} U_j)$.

[Hint:] The achievements of Goal 1-4 means that the authentication status in source and destination nodes, the principles are shown below. The pairwise key is believed by **Source Node** under Goal 1 achievement and by **Destination Node** under Goal 2 achievement, respectively. **Source Node** believes that the **Destination Node** believes

the pairwise key whenever Goal 3 is done. Conversely, **Destination Node** believes that the **Source Node** believes the pairwise key under Goal 4 derivation.

We then use the previously defined synthetic rules and assumptions to prove these four goals are achieved in TLPKA.

Proof of the achievement of Goal 1:

$U_i \mid\equiv U_i \xleftrightarrow{k_{ij}} U_j$.

By Synthetic rule 2, we know that

$$U_i \mid\equiv U_i \xleftrightarrow{k_{ij}} U_j$$
$$\rightarrow U_i \mid\equiv (U_i < g^{d_j} \mid C_{s_j g^{-(a_j ID_j + b_j)}})$$
$$\rightarrow U_i \mid\equiv (U_i < g^{d_j} \mid C_{s_j g^{-(a_j ID_j + b_j)}} \rightarrow U_i \xleftrightarrow{k_{ij}} U_j)$$

The second new goal is reached by Assumption 11. We then continue with the first goal. Using Synthetic rule 1:

$$U_i \mid\equiv (U_i < g^{d_j} \mid C_{s_j g^{-(a_j ID_j + b_j)}})$$
$$\rightarrow U_i < C_{s_j g^{-(a_j ID_j + b_j)}}(g^{s_j})$$
$$\rightarrow U_i \in r(C_{s_j g^{-(a_j ID_j + b_j)}})$$

The first new goal is message 2. And the second new goal is reached by Assumption 2.

Proof of the achievement of Goal 2:

Similarly we can prove Goal 2 with Assumption 12, message 3, and Assumption 1.

Proof of the achievement of Goal 3: $U_i \mid\equiv (U_j \mid\equiv U_i \xleftrightarrow{k_{ij}} U_j)$.

By Synthetic rule 2, we know that

$$U_i \mid\equiv (U_j \mid\equiv U_i \xleftrightarrow{k_{ij}} U_j)$$
$$\rightarrow U_i \mid\equiv (U_i < c_i + 1 \mid C_{k_{ij}})$$
$$\rightarrow U_i \mid\equiv (U_i < c_i + 1 \mid C_{k_{ij}} \rightarrow (U_j \mid\equiv U_i \xleftrightarrow{k_{ij}} U_j))$$

The second new goal is reached by Assumption 9. Now we continue with the first new goal with Synthetic rule 1:

$$U_i \mid\equiv (U_i < c_i + 1 \mid C_{k_{ij}})$$
$$\rightarrow U_i < C_{k_{ij}}(c_i + 1)$$
$$\rightarrow U_i \in r(C_{k_{ij}})$$

For the first new goal here, we now know that it is message 4. The second new goal is then achieved by Assumption 5.

Proof of the achievement of Goal 4:

Similarly, we can prove the achievement of Goal 4 with Assumption 10, message 5, and Assumption 6.

By way of the above discussion, we show that the proposed protocol provides mutual authentication and explicit key authentication properties.

4.3. Replay Attack

If the intruder can impersonate the source node or the destination node by replaying information what he or she collected as the source node and the destination node established a pairwise key, we say that the protocol used for security can not prevent the replay attack [10].

In TLPKA, the source node and the destination node record the random number d_i and nonce c_i for a period of time. Even if the intruder can collect messages g^{d_i}, $E_{k_{ij}}\{c_j\}$, $E_{k_{ij}}\{c_i, c_j'+1\}$, and $E_{k_{ij}}\{c_i'+1\}$, he or she can not impersonate the source node or the destination node and try to establish a pairwise key with a normal source node or destination node. If the intruder tries to impersonate the destination node, the normal source node will check to see whether it can receive the correct challenge c_i+1 and different c_i in Step 11 of the **Source Node** part. The message of Step 11 changes each time. Only the source node can decrypt the correct challenge c_i'+1 and check its validity. Consequently, the old message can not be used for a replay attack.

4.4. Man in the Middle Attack

The man in the middle attack is where an intruder intercepts the message and tampers with a vulnerable message. This kind of attack happens between the source node and the destination node. During this type of attack, both the source node and the destination node are not aware of this attack.

In TLPKA, the vulnerable message can not affect the procedure because the source node derives a unique pairwise key which is shared with the correct destination node. Therefore, even if the adversary intercepts the message, he or she can not reveal the information used for authentication. Only the correct destination node having the correct pairwise key can reply to the message correctly. Consequently, the Man in the middle Attack is prevented successfully.

4.5. Insider Attack

If an insider within the group of valid members can obtain the password of a victim member and impersonate this victim node in order to communicate with another member, then we declare that this security protocol cannot prevent against an insider attack. In TLPKA, members can establish a pairwise key without revealing secret information through a wireless channel. An insider can only know the pre-distributed secret that belongs to him or her. He/She knows nothing about any other member's secret information. Therefore, we can see that TLPKA can prevent against an insider attack.

5. Comparison

This section makes a comparison amongst TLPKA, Du *et al*., Liu *et al*., and Chang *et al*.'s. The comparison is divided into three parts: security property, computation overhead, and communication overhead.

5.1. Security Properties

We discuss the security properties provided by Du *et al*. [7], Liu *et al*. [13], Chang *et al*. [4], and TLPKA. These security properties include mutual authentication, explicit key authentication, resistance to the replay attack, resistance to the man in the middle attack, and resistance to the insider attack. The results of these security properties comparisons are shown in Table 1. From Table 1, we can see that TLPKA achieves all of these security properties while Du *et al.*'s scheme and Liu *et al.*'s scheme can not realize the security property of explicit key authentication. Furthermore, Chang *et al.*'s scheme does not have most of these security properties.

5.2. Computational Overhead

In this section, we present the evaluation of the computational overhead. We define the notation for computational overhead as follows:

exp: a round of the function , $z(g, x)$, from which the output is g^x.
add: a round of the function, $z(a, b)$, from which the output is $a+b$.
mul: a round of the function, $z(a, b)$, from which the output is $a*b$.
comp: a round of the function that compares indices of the corresponding predistributed secrets. With the comparison function, each node can find the overlap secret that is shared between the other nodes.
n: the number of nodes in the ad hoc network.
s: the number of the pre-distributed secret in a node for Liu *et al*. and Du *et al*..
d: the column number and row number of the secret matrix generated by the system in Du *et al*., $d \le n$.
t: the degree of the bi-variate polynomial used in Liu *et al*.
Ec\Dc: the symmetric encryption\decryption in Chang *et al*.
XOR: a round of the function, $g(a, b)$, from which the output is $a \oplus b$.
$H()$: a round of a one-way hash function computation.

In this comparison, we only consider the computational overhead of the pairwise key.

From Table 2, we can see that our scheme has only two exponential operations when it is compared with Du *et al*.'s and Liu *et al.*'s scheme. Furthermore, our scheme only has one extra exponential operation when it is compared with Chang *et al*.'s scheme. However, we can provide explicit key authentication in our scheme. And we can provide more security properties than Chang *et al*.'s scheme.

5.3. Communication Overhead

To measure the overhead of the communication, we use the size of the package exchange between the source node and the destination node as the criterion. If the source node needs to transit too large a package, it will consume the frequency bandwidth excessively. Therefore, the efficiency of the bandwidth allocation would not be very high. And the power cost of message exchanges would also reduce the life expectancy of the network.

In Du *et al*.'s scheme, the node only needs to compare the indices of the pre-distributed secrets. With these indices, the node can check whether there is any overlap secret in the other node. If a node is pre-distributed into s secret, the size of these indices is $s \log_2 n$ and the communication overhead of the Liu *et al*. is the same.

Therefore, the communication overhead of Du *et al*.'s scheme and Liu *et al*.'s scheme depends on the number of the pre-distributed secrets in a node and the number of nodes in the ad hoc network. For the Chang *et al*. scheme, six package exchanges are needed to set up a pairwise key. For TLPKA, six package exchanges are needed to set up and authenticate a pairwise key.

6. Conclusions

We have proposed TLPKA which provides a new idea for the design of a pairwise key agreement in ad hoc networks. In TLPKA, the pairwise keys between nodes refresh each time a link is set up. When compared with other key agreements in ad hoc networks, TLPKA requires higher computational overhead, however it provides much more of a secure link between nodes. In addition, TLPKA can resist the replay attack, the man in the middle attack, and the insider attack. We have also proven the correctness of TLPKA by using Buttyán *et al*. logic analysis. In the future, we will devote ourselves to the study of how to reduce the computational overhead in TLPKA.

Acknowledgments

This research was partially supported by the National Science Council of the Republic of China under the Grants NSC 97-2221-E-015 -001-.

Table 1. The comparison of the security property

	TLPKA	Du *et al*.	Liu *et al*.	Chang *et al*.
Mutual authentication	Y	Y	Y	Y
Explicit key authentication	Y	N	N	N
Resistance to the replay attack	Y	Y	Y	Y
Resistance to the man in middle attack	Y	Y	Y	N
Resistance to the insider attack	Y	Y	Y	N

Table 2. The comparison of the communication overhead

	Comp	*Add*	*mul*	*exp*	*Ec\Dc*	*XOR*	*H*()
Du *et al*.	1	*d*-1	*d*	0	0	0	0
Liu *et al*.	1	*t*	*t*	0	0	0	0
Chang *et al*.	0	1	0	1	2	2	2
TLPKA	0	1	1	2	2	0	0

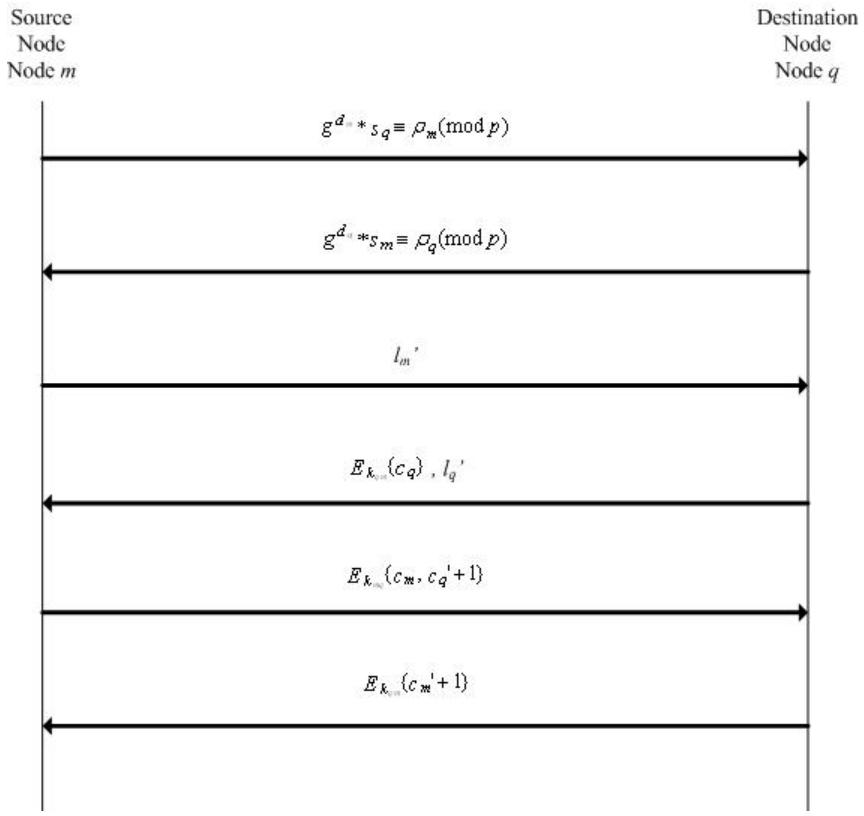


Fig. 1. The dataflow between source node *i* and destination node *j*.

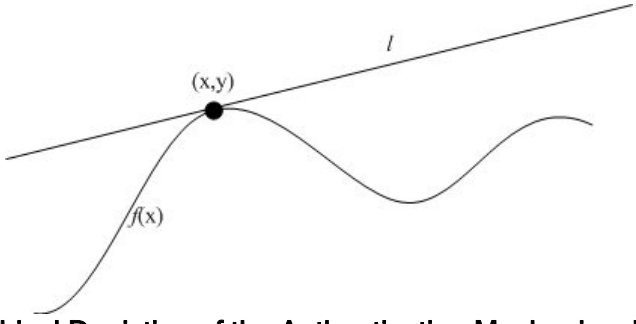


Fig. 2. A Graphical Depiction of the Authentication Mechanism in our Scheme

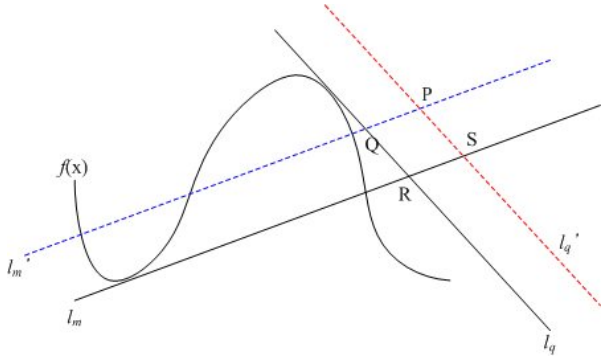


Figure 3. A graphical depiction of the Pairwise key Set Up Phase

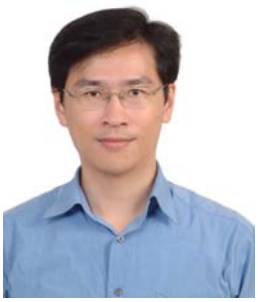
References

[1] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," *ACM Transactions on Computer Systems*, Vol. 8, No. 1, February. 1990.
[2] L. Buttya´n, S. Staamann, and U. Wilhelm, "A Simple Logic for Authentication Protocol Design," In *Proceedings of the 11th IEEE Computer Security Foundations Worksho*p*, Rockport, MA, USA*, June 1998, pp. 153–162.
[3] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," In *Proceedings of IEEE Symposium on Research in Security and Privacy. Oakland: IEEE Computer Society*, 2003, pp. 197-213.
[4] C. Chen, Chang, K. C. Lin, and J. S. Lee, "DH-Based Communication Method for Cluster-Based Ad Hoc Networks," in *Proceedings of the Second Asia Pacific Conference on Mobile Technology, Applications and Systems*: 1-8, November 2005, pp. 2-2A-1.
[5] S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *RFC 2501*, January 1999.
[6] J. Douceur, "The Sybil Attack," In *Proceedings of the IPTPS02 Workshop*, Cambridge, MA(USA), March 2002.
[7] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks," *ACM Transactions Information System Security*, vol. 8, 2005, pp. 228-258.
[8] L. Eschenauer and V. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002.
[9] M. Gerla and J.T.C. Tsai, "Multicluster, Mobile, Multimedia Radio Network," *Wireless Networks*, vol. 1, no. 3, 1995, pp. 255-265.
[10] M. S. Hwang, C. C. Lee, and Y. L. Tang, "A simple remote user authentication scheme," *Mathematical Computer Modelling*, vol. 36, 2002, pp. 103–107.
[11] J.T. Kohl, "The Evolution of the Kerberos Authentication Service," *Europen Conference Proceedings*, 1991, pp.295-313.
[12] W. Liao and M. Y. Jiang, "Family ACK Tree (FAT): Supporting Reliable Multicast in Mobile Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 52, Issue: 6, Nov 2003, pp.1675-1685.
[13] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, 2003, pp.52-61.
[14] C. de Morais Cordeiro, H. Gossain, and D. P. Agrawal, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions," *IEEE Network*, vol. 17, Issue: 1, Jan.-Feb 2003, pp.52-59.
[15] A. Patwardhan, J. Parker, A. Joshi, A. Karygiannis, and M. Iorga, "Secure Routing and Intrusion Detection in Ad Hoc Networks," In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications* 2005.
[16] M.B. Pursley and H.B. Russell, "Routing in Frequency Hop Packet Radio Networks with Partial-Band Jamming," *IEEE Transactions on Communications*, 1993, pp. 1117-1124.
[17] N. Smart, (2002) Cryptography. McGraw-Hill Education, UK.
[18] P. Sutherland, "Applied Cryptography, Protocols, Algorithm, and Source Code in C Bruce Schneier," *John Wiley & Sons Inc., 2nd Edition*, USA., 1996, pp.15.
[19] S. Zhu, S. Setia, and S. Jajodia, "Leap: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, Washington D.C, USA, October 2003.

Authors

Chung-Wei Chen received the MS degree in communication engineering from National Tsing–Hua University, Taiwan and BS degree in electric engineering from National Central University, Taiwan. He is now working toward PhD program in Institute of Communication Engineering of National Tsing-Hua University, Taiwan. His research interest includes security issues in mobile ad hoc network and wireless sensor network.

Shiuh-Jeng Wang was born in Taiwan, 1967. He received the M.S. degree in Applied Mathematics from National Chung-Hsing University, Taichung, Taiwan, in 1991. He received his PhD degree in Electrical Engineering at National Taiwan University, Taipei, Taiwan in 1996. He is currently with Dept. of Information Management at Central Police University, Taoyuan, Taiwan, where he directs the Information Cryptology and Construction Laboratory (ICCL, http://hera.im.cpu.edu.tw). He was a recipient of the 5th Acer Long-Tung Master Thesis Award and the 10th Acer Long-Tung Ph.D Dissertation Award in 1991 and 1996, respectively. Dr. Wang was a visiting scholar of Computer Science Dept. at Florida State University (FSU), USA in 2002 and 2004. He also was a visiting scholar of Dept. of Computer and Information Science and Engineering at University of Florida (UF) from Aug. 2004 to Feb. 2005. He served the editor-in-chief of the *Communications of the CCISA* in Taiwan from 2000-2006. He has been elected as the Panel Director of Chinese Cryptology and Information Security Association (CCISA) since Sept. 2006. Dr. Wang academically toured the CyLab with School of Computer Science in Carnegie Mellon University, USA, in Jan. 2007 for international project collaboration inspection. He is also the author/co-author of six books (in Chinese versions): *Information Security*, *Cryptography and Network Security*, *State of the Art on Internet Security and Digital Forensics, Eyes of Privacy –Information Security and Computer Forensics, Information Multimedia Security,* and *Computer Forensics and Digital Evidence* published in 2003, 2004, 2006, and 2007, respectively. He is a full professor and a member of the IEEE, ACM. His current interests include information security, digital investigation and computer forensics, steganography, cryptography, data construction and engineering.



Yuh-Ren Tsai received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1989, and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1994. From 1994 to 2001, he was a Researcher in Telecommunication Laboratories of Chunghwa Telecom Co., Ltd., Taiwan. Since 2001, he has been with the Department of Electrical Engineering and the Institute of Communications Engineering at National Tsing Hua University, Hsinchu, Taiwan, where he is currently an Associate Professor. His research interests include wireless sensor networks, mobile cellular systems, CDMA technology and cryptography.