Secure Routing for WSN-Based Tactical-Level Intelligent Transportation Systems

Der-Chen Huang[1], Ying-Yi Chu[1], Yuan-Kwei Tzeng[1], Yu-Yi Chen[2], Wei-Ming Chen[3]

[1] Department of Computer Science and Engineering, National Chung Hsing University, Taiwan
[2] Department of Management Information Systems, National Chung Hsing University, Taiwan
[3] Department of Computer Science and Information Engineering, National Ilan University, Taiwan
huangdc@nchu.edu.tw, {chuyy282, tzeng.yk}@gmail.com, chenyuyi@nchu.edu.tw, wmchen@niu.edu.tw

Abstract

Wireless sensor network (WSN) is gaining popularity in recent years due to the advantages of the WSN such as mobility, flexibility and low power consumption. Therefore, the usage of WSN in tactical-level Intelligent Transportation Systems (ITS) is expected to be able to overcome restrictions that the conventional ITS can only detect the vehicle in fixed position and high cost of construction and maintenance. Thanks to highly dynamic nodes to forward the network packets to the destinations, WSN is particularly vulnerable to attacks of interception and flooding, forging and tampering packets. Accordingly, reliable communication between nodes is dependent on the mechanism to verify the network traffic authenticity and communicating peers identity. In this paper, we propose a Zero Knowledge Key Exchange (ZKKE) scheme to setup a lightweight and adaptive hop to hop zero knowledge authentication chain (ZKAC). We adopt the GNY cryptographic protocol to prove the correctness of ZKKE and ZKAC. Based on computational cheap hash function and public-key scheme without trust third party (TTP), ZKAC enables hop-to-hop as well as end-to-end integrity protection for both routing and transformation informations in the WSN-based tactical-level ITS.

Keywords: Intelligent Transportation System (ITS), Integrity protection, Hash chain

1 Introduction

Recently, many of the electronic techniques required have been used in the modern battlefield [1]. A real-time demand has become an important issue for making a tactical decision since military operations are usuarlly dominated by electronic devices [2]. To realize this goal, there are various communication skills proposed in the applications of battlefield. Tactical decisions on the battlefield in intelligent vehicles have to meet the real-time requirement; otherwise, delayed decisions might cause unimaginable damages [3-6]. Generally, intelligent wireless sensors have been popularly applied to battlefield surveillance to collect information and send back to the base station in a mesh sensor network for military applications [7].

The idea of wireless sensor networks (WSNs) development derives from military battlefield monitor. An Ad-hoc On-Demand Distance Vector (AODV)-based routing protocol proposal is suitable for the WSN environments. AODV routing protocol [10] starts a route discovery process only when the service of an origination node needed. When an origination node has data packets to send but there has no route in its routing table, it broadcasts a Route Request (RREQ) message to its neighbors. Then, its neighbors rebroadcast the RREQ message to their neighbors if they have no fresh route to the destination node. This process continues until the RREQ message finds the final destination node or an intermediate node has a fresh route to the destination.

Mobile network topology in Intelligent Transportation Systems (ITS) is highly dynamic since the sensor is attached to the vehicle. These characteristics result in problems that do not exist in traditional WSNs. Mobile devices are often used in adverse or un-trusted environments with different malicious attacks on packet forwarding [8]. As a result, the assurance of integrity and authenticity is critical for each network layer [9]. To guarantee the secure communication of the WSN-based tactical-level ITS, it is essential to build security mechanisms that can endure malicious attacks from an insider who has access to the key data or whole control of several nodes.

For multi-hop networks, end-hosts can communicate with each other and possibly contain lots of forwarding nodes. It could cause resource eclipse attacks, i.e. CPU resource and target bandwidth, on communication paths. In order to restrict these attacks, the message authenticity and sender identity become vital to detect unauthorized, tampered, or forged messages in advance. Between forwarding nodes and end hosts, data authenticity is also allowed to control and signal data when location updates via mobile applications.

*Corresponding Author: Der-Chen Huang; E-mail: huangdc@nchu.edu.tw

DOI: 10.3966/160792642019072004002

Based on symmetric ciphers and shared keys, a lightweight end-to-end [11-12] data integrity protection and encryption is presented to provide communication security. Since forwarding nodes cannot have access to the shared secrets, the integrity check is not activated in hop-by-hop basis. For Ad-Hoc Networks, the validation of data authenticity and node identity is not able to function completely. Besides, the sharing of these symmetric keys between forwarding nodes becomes impossible since malicious activity can use the keys to manipulate packet. So unauthorized transmission and data manipulation can only identified via destination node. However, public-key approach is more complicated related to symmetric cryptography. For the packet verification in multi-hop WSN, energy consumption and communication latency are prohibited. In contrast, hash chain is computationally practical and useful for various protocols. TESLA [13] is dependent on a receiver capability to decide which key have already published from a sender, and CSA [14] applies an N-Party protocol to exchange the authenticated data. The Guy Fawkes Protocol [15] exploits a one-way hash chain for each message authentication, and Torvinen et al. [19] proposed the Weak Identifier Multi-homing Protocol (WIMP) to build and retain a context between an initiator and a responder. Nevertheless, these solutions are lack of on-route data verification in wireless multi-hop network environments and inefficiently used in the broader scope.

In this paper, we present zero knowledge authentication chain (ZKAC), which is an adaptive and lightweight secure authentication protocol for authentication and integrity protection. It is not about entities identities but relies on re-recognizing on-route communication based on hash chain method. ZKAC does not only provide end-to-end, but also hop-by-hop authentication and integrity protection for pervasive wireless networks, i.e. mobile ad hoc networks (MANETs) and WSNs. Hence, it can replace typical end-to-end encryption, which cannot be authenticated without pre-established secret or common security infrastructure. The goal is to adopt new authentication schemes by a secure and coherent system, which provides an efficient end-to-end and hop-by-hop integrity verification in highly dynamic pervasive transportation networks of the tactical environment.

The rest of the paper is organized as follows. Section 2 described related works and the motivations of this research. Section 3 describes the Zero Knowledge Key Exchange (ZKKE) key exchange scheme. Correctness of the scheme is proofed by GNY cryptographic protocol. Section 4 explains our ZKAC scheme to secure the AODV routing protocol and prove its correctness. In Section 5, we introduce seven scenarios that can be misused in the WSN-based tactical-level intelligent transportation systems. Finally, Section 6 gives the conclusions.

2 Related Works

An advantage of ID-based cryptography is less traffic due to no keys distributed, but it contains a third party, named as Private Key Generator, to create keys for packets decrypted. However, this induces Ad Hoc networks lose flexibility and scalability. A hierarchical ID-based architecture [16] assigns the workloads via releasing identity authentication and private key generation to lower-layer PKGs, but the security risks are simultaneously increased. The main disadvantage is a single point of failure. When the root PKG server on PKGs hierarchical head is compromised, then the hierarchy may no longer be able to function in the communications.

Hash chain has been used in the authentication and integrity protection in mobile Ad Hoc networks. Cheung introduced an effective message authentication scheme for link state routing based on time signatures [17]. Later, Perrig et al. proposed an efficient secure authentication protocol, named as TESLA [13], for multicast messages. TESLA divides time into several fixed-length time slots and each period is linked to various hash chain elements. It uses element of the present epoch in a one-way chain to calculate a Hash Message Authentication Code (HMAC) to give a protection for the data traffic. μTESLA and multi-level μTESLA [18] improve the TESLA approach to broadcast authentication by utilizing symmetric cryptography and limiting the number of senders for sensor networks. By means of substantial bandwidth and storage at nodes, μTESLA decreases the resource demands to store hash chain. Although time-based μTESLA divides the time period into many intervals for broadcasting, it restricted adaptability for on-path authentication. Besides, jitter could cause packet delivery disclosed hash-chain link. So the verifier drops the packet, and the minimum time frame induces the application latency in multi-hop wireless networks. TESLA is extended to various network latencies [18], and this method is devoted to latency discrepancy between receivers and cannot be employed to unicast communication. In addition, sender uses self-authenticating values to generate a one-way chain, and allocates hash values into uniform time interval. TESLA presents hash elements of the chain periodically even when no payload is transmitted and leads to computational overhead. Consequently, time-based hash chain is not suitable for the on-path problem, because it does not consider on-path integrity protection.

For interactive hash chain-based signatures (IHCS), a signer sends a message and HMAC are established with a hash chain to the receiver. The interaction of a signer and a receiver guarantees temporal segmentation between the creation of a signature and the disclosure of the hash-chain value. Unlike the time-based hash chain, IHCS does not need loose time synchronization among the peer nodes, which is hard to reach in a large network. Actually, each design is generally limited to a specific use-case. Anderson et al. proposed the Guy Fawkes protocol [15] to establish digital signatures via a small number hash function computation for integrity protection and unicast stream authentication. Torvinen et al. [19] have revealed a weak context establishment protocol for hash chain signatures and can be utilized for mobility and multi-homing signaling in the IPv6 network. Weimerskirch et al. [20] apply an interactive approach in the Zero Common Knowledge protocol for communication partner recognition in pervasive networks. Yao et al. [21] exploit an interactive protocol to construct reliable broadcast messages in wireless sensor networks where a single server transfers the message integrity codes to all nodes before delivering it. Although the protocol reaches on-path message authentication, it cannot offer point-to-point authentication among random wireless nodes. IHCS has no need to time synchronization and a fixed delay until the receiver verifies the packet. Hence, IHCS is especially well adapted in the widely wireless network environments than the TESLA.

LHAP [23] and HEAP [22, 24] were network level solution, which specifically designed to offer hop-by-hop authentication and integrity verification without any kind of security association with senders in MANETs. LHAP adopts TESLA key to achieve trust maintenance by authenticating periodically, and sends message to ensure the current released key validation. Thus, the malicious nodes will not be able to utilize an obsolete key to forge or tamper a packet. Lu et al. [23] propose a refined version of LHAP, which applies a TESLA-like protocol to securely transmit data packet between two adjacent nodes. HEAP employs pair-wise symmetric keys and a modified HMAC function to authenticate packets through hop-by-hop manner. Akbani et al. [22, 24] present a modified HMAC-algorithm based on authentication scheme, where each node keeps several pair-wise secret keys for its neighbors and a neighborhood secret key. For the modified HMAC-algorithm, both keys are utilized to create the MAC messages, which are attached with index number to avoid the replay attack. All of the protocols mentioned above are designed to block external attacks. However, these protocols make the procedure more vulnerable to against attacks. These protocols are also susceptible to against the internal attacks, which are initiated by the compromised nodes located inside the network. Protection against these insider attacks needs end-to-end integrity protection, and has be conformed on each hop. Zhu et al. [25] and Ye et al. [26] proposed Lightweight Hop-by-hop Authentication Protocol for Ad Hoc Networks to resolve the issue of efficient en-route validation with probabilistically methods. Nevertheless, these skills are closely related to sensor network environments with many cooperation sensors, detecting and transmitting the identical data to certain stations. Furthermore, they are not appropriate to point-to-point communication among unitary hosts for all sizes of networks. Zhang et al. [27] adopt polynomial-based message authentication for packet authentication in the presence of networks. This technique uses a central security server, which offers key materials to entire nodes before arrangements. Additionally, it is feasible in numerous WSN environments, but is not suitable for lots of dynamic and distributed arrangements.

Unlike traditional WSNs, MANETs have no requirement of infrastructure, and allow multi-hop connectivity among nodes. It is therefore able to support military and commercial applications due to characteristic of dynamic and random topologies, and real-time communication. As proposed in [29], a Security Using Pre-Existing Routing for Mobile Ad hoc Networks (SUPERMAN) protocol gives a full security to protect routes and communication for MANETs. As a way to increase the security in MANETs, a unified trust management scheme [30] utilizes uncertain reasoning theory to derive accurate trust values based on direct and indirect observation of nodes. Accordingly, a more secure path can be created to transmit data packets between nodes. For the next generation mobile network of 4G and 5G technologies, vehicular ad hoc networks (VANETs) [31-32] becomes critical in the development of highly dynamic and mobile vehicles. In order to protect messages between sender and receiver, an improved dual-protected ring signature (DPRS) [33] is used to guarantee both communications security in VANETs.

3 The ZKKE Scheme

3.1 Basic ZKKE Protocol

In this section, we propose a new ZKKE schema based on the WSN for Intelligent Transportation System (ITS) architecture. For better understanding, we first give an overview of the basic key exchange process before discussing extensions that enable the adaptation of ZKAC. The goal of the key exchange is very lightweight but still provably secure scheme. It is not only aiming at the inferring involved entities' identities but also re-recognizing foreign communication partners whenever necessary. We follow a fully infrastructure-less approach of establishing trust relationships in highly dynamic pervasive transportation networks of the tactical environment. A network is pure if there exist neither trust third party (TTP) that provides central services nor does a fixed infrastructure exist. If there is no pre-established knowledge between intermediate entities in a general manner, it might be a wireless multi-hop network that intruders may launch misused attacks such as route disruption, route invasion, node isolation, and resource consumption. ZKKE can be described as: A is able to authenticate B in a zero common-knowledge fashion if A is able to identify the authority again that is generated from B. That is, B is able to convince A that both had some relationship in the past. In other words, we also say that A recognizes B, or that B authenticates to A. For example, if B is forwarding a data-packet for A in the beginning phase, then later on A is able to recognize B to forward another packet in following phases. ZKKE authentication is specifically in a highly dynamic pervasive network consisting of weak devices, where there are no pre-established secrets. It can replace common security infrastructure, and be used without any pre-established secret. Accordingly, the ZKKE authentication provides a fully infrastructure-less approach of establishing trust relationships in highly dynamic pervasive transportation networks of the tactical environment. That is, we do not use any tamper resistant devices to perform expensive public-key operations, special access structures for deploying a distributed public key infrastructure (PKI), and so on. Such approach is more practical and flexible, yet still sufficient in a high dynamic pervasive network. Without knowing user's identity in advance, our solution is suited for cooperation based schemes as well as secures routing methods and routing authentication based on re-recognition mutually. The ZKKE protocol provides security features as follows:

(1) Each target node can authenticate its origination node;

(2) Each receive node can authenticate its sender from which message sending;

(3) Each middle node can authenticate its previous node for routing table updating;

(4) By using hash chain, a hop count is maintained or increased.

Here, we describe the generate ZKKE scheme in WSN routing process. Consider the case where an entity A wants to be able to recognize an entity B in response its request after an initial contact. In this scenario, B might be any node that willing to transferring request from A. First, the step A broadcasts its public-key with some data that needed to cover on. Let x be the nonce generated by A for this session, A is able to prove that B is willing to cover the data for her. Further, B is able to use the public key $f(x)$ corresponding to the nonce x in such a way that A can verify that a message is original from B, i.e., B is able to perform a message authentication by using the public key $f(x)$. Let $\{m\}_{+f(x)}$ be an authentication of a message m, then A can verify the origin by checking $\{\{m\}_{+f(x)}\}_{-f(x)} = m$.

A simplified version of ZKKE is illustrated in Figure 1 and as follows:

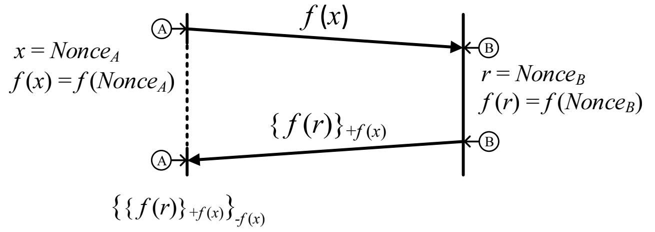


Figure 1. The ZKKE scheme

(1) A generates a nonce x and public key $f(x)$ generated by x

(2) A sends $f(x)$ to B

(3) B generates nonce r randomly, and public key $f(r)$ created by r

(4) B sends $\{f(r)\}_{+f(x)}$ to A

(5) A opens $\{f(r)\}_{+f(x)}$ with $-f(x)$

If 'succeed', A accepts $f(r)$, A gets B's public-key, otherwise she rejects

Remarks:

Step 1. only needs to be performed once for each request entity.

Step 2. needs to be performed once for each communication pair A and B.

Step 3.-5. have to be performed for each authentication process.

Step 4. can combine with a message block r', $(r', f(r))_{+f(x)}$ with random r' to avoid chosen-text attacks.

We consider this scheme the capability to ensure that entities are able to recognize another entity in order to receive a service response to (routing request) their request. Hence, the public key $f(x)$ always has to be sent together with the offered service (say *RREP*), i.e., service and public-key have to be bound together to avoid that a malicious entity can misuse the whole service. It follows that the authentication scheme we are envisioning here usually is connected to the service, i.e., to the requesting messages that are exchanged.

Repeat Steps 4 and 5 for each reply message to authenticate, we send m_A together with $f(x)$ to B. Then, the steps 4 and 5 of ZKKE can be modified as:

(4') B sends $(m_B, f(r))_{+f(x)}$ to A as reply

(5') A opens $(m_B, f(r))_{+f(x)}$ with $-f(r)$

If 'succeed', A accepts $f(r)$ and m_B, A gets B's public-key and the reply, otherwise she rejects.

Message authentication keep the message integrity, the data is recent and not replayed. In contrast to PKI, without a logical central certificate directory, A gets B's public key (together with B's ID string) to be able to recognize the reply from B. The above protocol can easily be extended for this case.

3.2 ZKKE Scheme Correctness Proof

We present a formal analysis of the ZKKE scheme and verify that the state key exchange goal is achieved. Protocol is abstracted as the exchange of secret between two nodes. The GNY [28] cryptographic protocol gives a systematic way for the analysis of security of cryptographic protocol. In GNY, symbol X and Y refer to formulas, P and Q refer to principals, S and K are a shared secret and encryption key respectively, and C is a statement. The symbol representations and explanations can be shown as follows:

- (X,Y): conjunction of two formulas; it is treated as a set with properties such as associativity and commutativity.
- $\{X\}_K$ and $\{X\}_K^{-1}$: conventional encryption and decryption, they satisfy $\{\{X\}_K\}_K^{-1} = X$, but not necessarily $\{\{X\}_K\}_K^{-1} = X$.
- $\{X\}_{+K}$ and $\{X\}_{-K}$:public-key encryption and decryption, they satisfy $\{\{X\}_{+K}\}_{-K} = X$.
- $H(X)$: a one-way function of x. It is needed that given x it is computationally feasible to compute $H(X)$; given $H(X)$ it is infeasible to compute x.
- $*X$: Not-originated-here formula property. If P is told X (see below), it can distinguish it did not previously convey X in the current run.

Basic statements:

- $P \triangleleft X$: P is told with formula X . P receives X , possibly after performing some computation such as decryption.
- $P \ni X$: P possesses, or is capable of possessing, formula X .
- $P \mid\sim X$: P once conveyed formula X . That is a formula can be conveyed implicitly.
- $P \mid\equiv \sharp(X)$: P believes, or is entitled to believe that formula X is fresh. That is, X has not been used for the same purpose at any time before the current run of the protocol.
- $P \mid\equiv \phi(X)$: P believes, or is entitled to believe that formula X is recognizable. That is, P would recognize X if P has certain expectations about contents of X before actually receiving it.
- $P \mid\equiv P \xleftrightarrow{s} Q$: P believes, or is entitled to believe that S is a suitable secret for P and Q . S will never be discovered by any principal except P, Q, or a principal trusted by either P and Q .
- C_1 , C_2 : conjunction of two statements, with properties such as associativity and commutativity.
- $P \mid\equiv C$: P believes or is entitled to believe that statement C holds.
- $\dfrac{P \triangleleft (X,Y)}{P \triangleleft X}$: P being told a formula implies P being told each of the formula's concatenated components.

Analysis and evaluation of the proposed ZKKE protocol is provided for correctness and security using the GNY cryptographic protocol. Now, we give some assumptions as shown in Figure 1. The ZKKE scheme is immediately derived from the follows:

(1) $A \ni x$, $A \mid\sim f(x)$

A generates a nonce x and public key $f(x)$ created by x, then A sends $f(x)$ to B

(2) $B \triangleleft f(x)$

B receives $f(x)$

(3) $B \ni x$, $\{f(r)\}_{+f(x)}$

B generates nonce r randomly, and public key $f(r)$ created by r

(4) $B \mid\sim \{f(r)\}_{-}(+f(x))$

B sends $f(r)$ encrypted by $f(x)$

(5) $A \triangleleft \{f(r)\}_{+f(x)}$, $\{\{f(r)\}_{+f(x)}\}_{-f(x)}$

A receives $\{f(r)\}_{+f(x)}$ and decrypted it with private key it process

(6) $A \mid\equiv \phi(f(r))$

if 'succeed', A accepts $f(r)$, A gets $B's$ public-key, otherwise she rejects

We can consider the traditional security claims, such as authentication, confidentiality, integrity, and non-repudiation. The ZKKE scheme cannot establish confidentiality to sure the identity of each entity. By authenticating above messages, it is possible to build integrity of messages. Obviously, the scheme does not emphasis on the non-repudiation. Nevertheless, for some scenarios of AODV routing protocol non-repudiation may also be appropriate. We define ZKKE non-repudiation to be the service, which avoids an entity to decline a commitment or actions on routing chain. In the AODV routing case, this implies that an entity A is capable of conforming to a third party that a number of actions or commitments were made by the same (perhaps unknown) entity B. Clearly, a scheme that offers signatures meets the ZKKE non-repudiation objective.

To break the ZKKE protocol, an intruder has to construct an authenticated message $(m)_x$ for given message m and public key $f(x)$. Considering man-in-the middle attack as follows, there is an entity B that offers a service, an entity A that seeks this service, and a malicious entity M. The entity B sends $f(x_B)$ to A. M interrupts this message, and sends $f(x_M)$ to A. Then, M satisfies the needs of A by offering its services. All that A is concerned the service only. She does not care who provided the service. Man-in-the-middle attack has no harm on our protocol.

Based on traditional signature schemes, authentication is done by the proof of knowledge of the secret key in a challenge and response manner. In ZKKE scheme, the verifier sends a challenge r, and the authenticator computes a message authentication code (MAC) of r. Using digital signatures the scheme ensures ZKKE authentication and ZKKE non-repudiation, and also message authentication.

In the case of symmetric cipher, a secret key ‘s’ has to be shared a priori, i.e., the protocol requires a secret channel to exchange the shared secret. It is suited for applications where devices are relationship tightened, for example, military devices attached on ITS, which can exchange the keys by only a single trust authority.

4 General ZKAC Scheme

In this section, we introduce a new protocol that is based on ZKKE and it is more efficient and faster than any public-key scheme. Our scheme provides a secure AODV route discovery operation for WSN. The ZKAC scheme combats attacks that disrupt the route discovery process. It incorporates mechanisms that safeguard the network functionality from attacks exploiting the protocol itself to degrade network performance and possibly lead to denial of service.

4.1 ZKAC Route Request

When a source node X_0 initiates a route discovery to the destination X_T, the request packet from X_0 constructing a unique identified by the packet itself: $RREQ_{X_0 \leftrightarrow X_T} =< RREQ\ ID, Dest\_IP, Dest\_seq, Orig\_IP, Orig\_seq >$. Where $RREQ_{X_0 \leftrightarrow X_T}$ denotes the fingerprint of the source and destination and the unique (with respect to the pair of end nodes) query identifiers which is the input for the calculation of the MAC, along with the shared secret *x* between X_0 and X_T. In addition, the hop-count of the traversed intermediate nodes is accumulated in the route request packet.

We define a hash chain provides a novel approach to the secure route discovery operation for WSN routing protocol, the relaying nodes on a path can verify the integrity and origin of a message if they have forwarded all previous signatures between the signer and the verifier, which can use a key-value of the chain to generate an authenticated message by a MAC (keyed hash function). Let $H(m, f(x)) = HMAC(m, f(x))$ be the hash MAC of a message *m* by the public-key of *x* says $f(x)$. The main idea of our protocol is as follows: First, exchange a value $f(x)$, which the intermediate receiver will tie together with some experience (the hop account). Then, prove knowledge of the pre-image of $f(x)$ to authenticate by establishing a relationship to $f(x)$ and the past experience. Since we want to be able to repeat the authentication step arbitrary many times, we propose a method of using key-chain based on a one-way hash function. Figure 2 illustrates the ZKAC_RREQ scheme. The protocol works as follows:

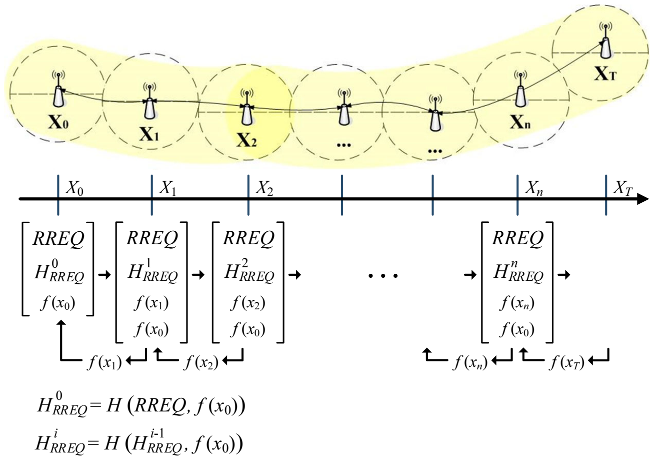

X₀
X₁
X₂
...
...
Xₙ
Xт
X_0
X_1
X_2
X_n
X_T
RREQ H^0_{RREQ} $f(x_0)$
RREQ H^1_{RREQ} $f(x_1)$ $f(x_0)$
RREQ H^2_{RREQ} $f(x_2)$ $f(x_0)$
...
RREQ H^n_{RREQ} $f(x_n)$ $f(x_0)$
$f(x_1)$
$f(x_2)$
$f(x_n)$
$f(x_T)$
$H^0_{RREQ} = H(RREQ, f(x_0))$
$H^i_{RREQ} = H(H^{i-1}_{RREQ}, f(x_0))$

Figure 2. ZKAC_RREQ scheme

(1) X_0 broadcasts her public key $f(x_0)$ with message *RREQ* and $H^0_{RREQ} = H(RREQ, f(x_0))$ says $(RREQ\ H^0_{RREQ}, f(x_0))$.

(2) Intermediate node X_1 gets the packet and willing to forward this request for X_0. X_i stores $(RREQ\ H^0_{RREQ}, f(x_0))$.

(3) X_i generates random number x_1 and the corresponding public key $f(x_1)$.

(4) X_i sends his public key encrypted by public key encryption $\{f(x_1)\}_{+f(x_0)}$ to the node where he get the message from (say X_0), X_0 decrypts $\{f(x_1)\}_{+f(x_0)}$ as $\{\{f(x_1)\}_{+f(x_0)}\}_{-f(x_0)}$, X_0 accepts $f(x_1)$, X_0 gets X_1 ‘s public-key.

(5) X_i broadcasts its public key $f(x_1)$ with *RREQ* and $H^1_{RREQ} = H(RREQ, f(x_0))$ says $(RREQ\ H^1_{RREQ}, f(x_0))$.

Repeat steps 6 to 10 for each authenticate intermediate node until X_T get the message

(6) X_i broadcasts its public key $f(x_i)$ with message *RREQ* and $H^i_{RREQ} = H(H^{i-1}_{RREQ}, f(x_0))$ says $(RREQ, H^i_{RREQ}, f(x_i), f(x_0))$.

(7) Intermediate node X_{i+1} gets the packet and willing to forward this request for X_i. X_{i+1} stores $(RREQ, H^i_{RREQ}, f(x_i), f(x_0))$.

(8) X_{i+1} generates random number x_{i+1} and the corresponding public key $f(x_{i+1})$.

(9) X_{i+1} sends his public key encrypted by public key encryption $\{f(x_{i+1})\}_{+f(x_i)}$ to the node where he get the message from (say X_i), X_i decrypts $\{f(x_{i+1})\}_{+f(x_i)}$

as $\{\{f(x_{i+1})\}_{+f(x_i)}\}_{-f(x_i)}$, X_i accepts $f(x_{i+1})$, X_i gets X_{i+1}'s public-key.

(10) X_{i+1} broadcasts its public key $f(x_{i+1})$ with message $RREQ$ and $H_{RREQ}^{i+1} = H(H_{RREQ}^{i+1}, f(x_0))$ says $(RREQ, H_{RREQ}^{i+1} f(x_{i+1}), f(x_0))$.

(11) X_T gets $(RREQ, H_{RREQ}^{n} f(x_n), f(x_0))$ from X_n, X_T verifies $f(x_0)$ with the secret x_0.

If it is true, then check if $H_{RREQ}^{n} = H(RREQ, f(x_0))^n$ is satisfied.

If it is true, then accepts the request.

Remarks:

ZKAC scheme considers the case where an entity X_0 wants to be able to generate a routing path to the entity X_T. Obviously, the communication channel they use constructed by X_i is insecure. In the first step, X_0 provides X_T with some data that allows the later one to recognize X_0. Let S be a set of secrets, and $x \in S$ be a secret.

The ZKAC used in route request and reply packets is described individually. However, it is possible for ZKAC to operate in a more general setting, for example, a route reply is appended to a data packet. Figure 3 is an extension of a routing protocol and the ZKAC header is appended to the routing protocol packet.

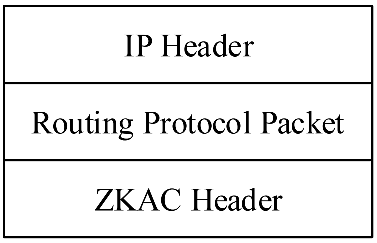


Figure 3. ZKAC for a reactive routing protocol extension

A source node X_0 maintains a public key $f(x_0)$ that is generated by the secret x_0 and it is shared with the target node. $f(x_0)$ is placed in the *ZKAC* header as illustrated in Figure 4, along with the Request HMAC. The one-way function input is the whole *IP* header, which is the basic protocol route request packet and most significantly, where the public key is generated by shared key x_0. The *Route Request* fields are updated as the packet propagates toward the destination, that is to say, the accumulated hop number of the intermediate nodes, and the IP-header mutable fields are removed.

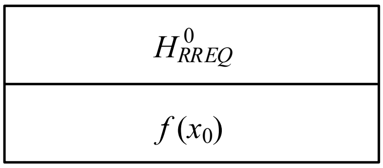


Figure 4. The ZKAC header

4.2 ZKAC Route Query/Forwarding

Intermediate nodes analyze the received Route Requests for the purpose of decide whether a *ZKAC* header is presented. If it is not, process the packet as represented in the basic protocol standard. Besides, the intermediate nodes accumulate the H_{RREQ}^{0} as $H_{RREQ}^{i} = H(H_{RREQ}^{i-1}, f(x_0))$. After that, the intermediate nodes broadcast the route request directly. Meanwhile, intermediate nodes can measure the frequency of queries received from their neighbors to regulate the query propagation process. On one hand, all nodes self-regulate the generation of new route requests to make the control traffic overhead low. On the other hand, malicious nodes could act selfishly and prevent from backing off before a new route query is generated, or generate queries at the highest possible rate to consumes network resources and degrades the routing protocol performance resulting in denial of service.

In order to ensure the responsiveness of the routing protocol, each benign node keeps a priority ranking of its neighbors based on the corresponding observed rate of queries. The lowest priority is allocated to the neighbors producing queries constantly, and the highest priority is allocated to the nodes producing or relaying requests with the lowest rate. Accordingly, quanta are assigned in proportion to the priorities and each class query is served in a round-robin fashion.

When immediate neighbor of a malicious node notice a high rate of incoming queries, which update the corresponding priority. Besides, no service low priority queries are eventually excluded. By doing this, non-malicious queries are just affected for a period of time that is no longer than the time of detecting and updating the priority for misbehaving neighbor. Meanwhile, the round-robin operation offers extra insurance to ensure benign requests can propagate as well. Therefore, the suspected requests near the potential source of misbehavior can be filtered, and benign nodes farther away from the adversary nodes will not be affected.

4.3 ZKAC Route Reply

X_T validates the received route request packet first by verifying that it was original from a node with which it has a security binding. Then, X_T checks if the *RREQ* packet is first received. Otherwise, X_T calculates the keyed hash of the request fields by hop count times. If the output matches the ZKAC header

MAC, the integrity of this request is verified, which is along with the authenticity of its origin. The destination generates a number of replies to valid requests, where the number is as many as the number of its neighbors. In order to disallow a possibly malicious neighbor to control multiple replies, Figure 5 illustrates each valid request. X_T is used to placing $H_{REEP} = H(REEP, f(x_0))$ in the route reply packet in the corresponding ZKAC header fields so that X_0 can verify that the reply is really replied from the destination node. The ZKAC header covers the basic protocol route reply and the MAC_{REEP}, protects the integrity of the reply on its way to the source and offers an evidence to X_0 that the request has indeed reached the destination.

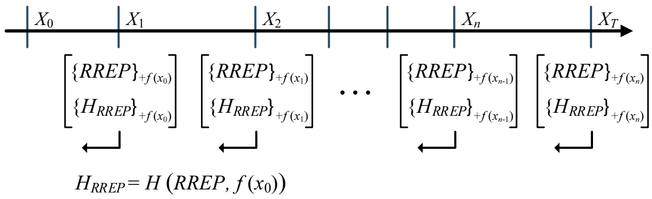


Figure 5. ZKAC_RREP scheme

4.4 ZKAC Route Reply Validation

On reception of the Route Reply, the intermediate node X_i decrypts the ZKAC packet with it's private key x_i. If it cannot, X_i rejects the reply. After that, X_i sends the ZKAC packet that uses the PKC to encrypt the *RREQ* and H_{REEP} with the public key $f(x_{i-1})$ via the reverse path from where it got the *RREQ* packet.

On reception of the Route Reply, X_0 checks the source and destination addresses, and excludes the *Route Reply* since it does not correspond to the currently pending query. Besides, it compares the reply *IP* source-route with the reverse of the route carried in the reply payload. If the two routes match, X_0 computes the MAC utilizing the replied route, the ZKAC header fields and the key *x* shared with X_T. Based on successful verification, X_0 is used to verify that the request, indeed, reached X_T and that the reply was not failed on its way from X_T to X_0. Moreover, since the reply packet has been routed and successfully received along the reverse route it carried, the route information has not been compromised during the request propagation before arriving at X_T. Hence, the connectivity information is genuine.

4.5 ZKAC Intermediate Node Replies

The caching of overheard routes is a severe vulnerability because false topology information is simply disseminated while propagating through a big portion of the network.

The attacker fabricates the packets or routing replies transferred between nodes and resends, which are cached by nodes operating in promiscuous mode. When such fabricated routes are utilized or offered as replies, more unsuspecting nodes cache such failed routes and may utilize them in future. To realize the robustness, route caching is not welcomed generally and intermediate nodes are not necessary to give route replies. Nevertheless, route caching can be implemented to enhance the efficiency of the route discovery process. In such a case, if an intermediate node X_M has an active route to X_T and a ZKAC had built between X_0 and X_M, then a reply could be provided to X_0. The ZKAC protocol is to enabling the intermediate node to reply the request that has been validated previously on the route to the X_T, which is based on ZKAC header to generate the reply.

4.6 ZKAC Routing Maintenance

This function is an integral part of most WSN routing protocols. When preventing false or fabricated notifications, topology changes must to be detected and the sources of the affected routes must to be informed. This task is facilitated by the fact that intermediate node caching is disabled, but route error messages must be retained. The ZKAC allows for enhanced detection of any type of transmission failures. However, this end-to-end approach does not allow to distinguishing benign (due to topology changes) from malicious route failures. Thus, route error messages generated by intermediate nodes are retained in ZKAC to provide fast detection of path breakages. The route error packets are source-routed along the prefix of the route reported as broken, and X_0 compare the route traversed by the error message to the prefix of the corresponding route. The ZKAC can back to the location of generating routing error in terms of verifying the routing error response.

With ZKAC an intruder node lying on a $X_0 \rightarrow X_T$ route cannot invalidate the route, mislead X_0 by corrupting error messages generated by another node, or mistaking a dropped packet as a link failure.

4.7 ZKAC Scheme Correctness Proof

In the following, we analyze and evaluate the proposed ZKAC protocol is abstracted as the exchange of two messages, a route request and a route reply. The messages are transmitted over pervasive pubic network. The idealized form is shown in Figure 6 and Figure 7. Similarly, the symbol representations and explanations in the following can be referred to Section 3.2. For generalize, as shown in Figure 6, we assume the process between two intermediate nodes, which are immediately derived from the follows:

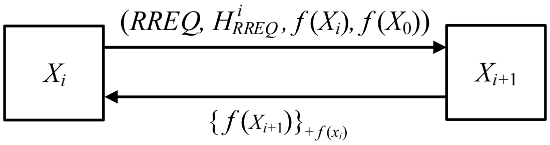

X_i
X_{i+1}
$(RREQ, H^i_{RREQ}, f(X_i), f(X_0))$
$\{f(X_{i+1})\}_{+f(x_i)}$

Figure 6. ZKAC intermediate node communication scheme

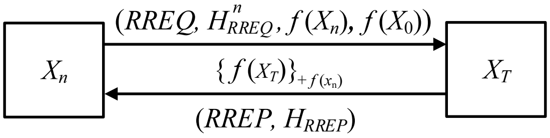


Figure 7. ZKAC destination node communication scheme

(1) $X_0 \ni f(X_0), X_0 \mid\equiv X_0 \xleftrightarrow{f(X_0)} X_T, ,$

The sender X_0 possesses the share secret $f(X_0)$ and it believes it is used for mutual proofs of secret between X_0 and X_T.

(2) $X_T \ni f(X_0), X_T \mid\equiv X_0 \xleftrightarrow{f(X_0)} X_T, ,$

The receiver also trusts the shared secret $f(X_0)$ and it believes it is used for mutual proofs of secret between X_0 and X_T.

(3) $\dfrac{X_{i+1} \triangleleft^* (RREQ, H^i_{RREQ}, f(X_i), f(X_0))}{X_{i+1} \triangleleft (RREQ, H^i_{RREQ}, f(X_i), f(X_0))}$ and

$$\frac{X_{i+1} \triangleleft^* (RREQ, H^i_{RREQ}, f(X_i), f(X_0))}{X_{i+1} \ni (RREQ, H^i_{RREQ}, f(X_i), f(X_0))}$$

The intermediate node X_{i+1} gets the route request form the previous node, after doing the key exchange; X_{i+1} forwards the request.

(4) After n steps we obtain $\dfrac{X_{i+1} \triangleleft (RREQ, H^n_{RREQ}, f(X_n), f(X_0))}{X_{i+1} \ni (RREQ, H^n_{RREQ}, f(X_n), f(X_0))}$

Finally (Figure 7), the destination node X_T gets the route request and confirm the request is from node X_0 by share secret $f(X_0)$.

(5) $\dfrac{X_T \ni (RREQ, H^n_{RREQ}, f(X_n), f(X_0))}{X_T \ni RREQ, X_T \ni H^n_{RREQ}}$, $X_T \ni RREQ,$ $X_T \mid\sim RREP$

After confirming the route request, X_T send the reply back via the reverse route with hashed message.

(6) After tracing back n hops to X_0, $\dfrac{X_0 \equiv (REEP, H_{RREP})}{X_0 \mid\equiv X_T \sim (REEP, H_{RREP}), X_0 \mid\equiv X_T \mid\sim H_{REEP}}$, $X_T \mid\equiv$ $X_0 \xleftrightarrow{f(x_0)} X_T, X_0 \equiv \phi(REEP)$

RREP travels across the secured opposite route back to X_0, X_0 confirmed the reply is really send back from the destination node X_T, then the secure route between X_0 and X_T created.

5 ZKAC in WSN-Based Tactical-Level ITS Environments

As WSN prove usefulness in both military and civil applications, ITS is one of the interesting areas, where WSN could enhance the performance significantly. ITS will include both stationary sensors (roadside sensors) and mobile sensors (Improved Mobile Subscriber Equipment: IMSE).

Considering the typical scenario for strategic and operational-level communications depicted in Figure 8, our IMSE vehicle (X_0) is in the right lane of a divided highway, trying to send tactical decision to X_T. Unfortunately, an intruder (X_M) breaks into our communication by misusing the route discovery protocol, and preventing us from constructing communication network to the Command and Control center (C2C) X_T. We represent the query request as a list $\{REEQ, X_0, X_1, X_2, \ldots, X_n, X_T\}$, with *RREQ* denoting the ZKAC header for the request query searching for X_T which is initiated by X_0. Similarly, the route reply is denoted as $\{REEP, X_0, X_1, X_2, \ldots, X_n, X_T\}$. We now describe a number of security attack scenarios that initiated by the intrude node which can also happen on WSN-based tactical-level ITS environment. To demonstrate our proposed works, we have illustrated seven scenarios as follows:

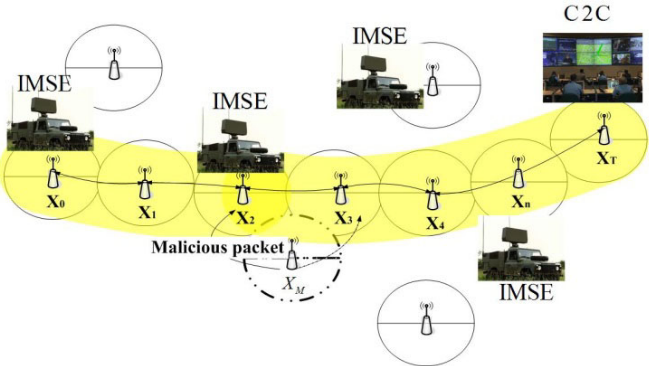


Figure 8. IMSE dynamically established a secure route to C2C by ZKAC scheme

Scenario 1: intruder modified *RREQ* → forward ⇒ route disruption

Assume node X_0 broadcasts a *RREP* message to create a route to node X_T. After accepting the *RREP* message, the attack nodes could include the following modifications to the *RREP* message:

Change the *RREQ* ID of node X_0 with the *RREQ* ID of node X_T, and increases it via a small number;

- Exchange the originator IP address (X_0 node) with the destination IP address (X_T node) in the *RREQ* message;

Increase the destination sequence number via not less than one, and then exchanges the sequence number beween originator and the destination;

- Write the source IP address in header using a non-existent IP address.

For these revisions, the attack nodes feign to forward a *RREQ* message initiated from the X_T node to the X_0 node, which the original *RREQ* message is initiated from the X_0 node to the X_T node. Attack neighbor node will receive the forged *RREQ* message, since they have not accepted a *RREQ* message with a *RREQ* ID from the X_T node in advance. Since the forged *RREQ* message has a bigger originator sequence number, these neighbors will renew their next hop to the X_T node as a non-existent node, which is pointed via the source IP address in header. These neighbors will rebroadcast the forged *RREQ* message to their neighbors. If the X_T node accepts the forged *RREQ* message, it merely throws the message because this message is originated from itself. If the X_0 node receives the forged *RREQ* message, it will renews its opposite routing table because the originator sequence number (X_T node) in the forged *RREQ* message is bigger related to its routing table. Then, the X_0 node renews the next hop to the X_T node from receiving the forged *RREQ* message and unicasts a *RREP* message to this neighbor. If the *RREP* message is unicasted along the opposite route, it will be missed owing to the non-existent node in the opposite route.

Owing to the broadcast of the legal *RREQ* message, the *S* node could receive normal *RREP* messages. Nevertheless, the route created via the forged *RREQ* message will reduce the routes created via these regular *RREP* messages, because sequence number of the X_T node in the forged *RREQ* message is bigger related to the normal *RREP* messages. When the *S* node transmits data packets through the route created via the forged *RREQ* message, all packets will be missed if they are transmitted to the non-existent node. If the previous attack neighbor node finds the link fail, it will either transmits a *RREP* message to the X_0 node or begins local repair, which broadcasts a *RREQ* message to find a route from itself to the destination node when the destination is not far related to the maximal number of repair hops.

Scenario 2: intruder modified *RREQ* → forward ⇒ route invasion

A scenario in an inside attacker is in the transfer range of an originator node, which initiates route discovery with a *RREQ* message. After accepting the *RREQ* message from the source node, the attack nodes could alter the *RREQ* message:

- Increment the *RREQ* ID of originator node via not less than one;
- Increment the originator sequence number via not less than one;
- Increment the destination sequence number via not less than one.

After creating this forged *RREQ* message, the attack nodes broadcast it to its neighbors. These neighbors will receive this forged *RREQ* message owing to the new *RREQ* ID. Then, they renew their next hop to the originator node as attack node, since the forged *RREQ* message has a bigger originator sequence number related to those in their routing tables. They also rebroadcast the forged *RREQ* message to their neighbors. If the originator node receives the forged *RREQ* message, it throw the message because this message seems to originate itself. If the destination node accepts the forged *RREQ* message, it renews its next hop to the originator node as the neighbor from the forged *RREQ* message. Then, it renews its sequence number to the destination sequence number in the *RREQ* message, which is bigger related to its sequence number. Subsequently, it write the renewed sequence number into the destination sequence number in the *RREP* message. Then, the destination node unicast the *RREP* message to the originator node through the opposite route, which contains the attack nodes. Because this *RREP* message includes a bigger destination sequence number related to the routing table of originator node, which could have been renewed via other legal *RREP* messages, the originator node renew the destination sequence number to the *RREP* message, and arrane the attacker as the next hop to the destination node. Consequently, the attacker achieves intruding the route from the originator to the destination node.

Scenario 3: intruder forges *RREQ* → broadcast ⇒ route disruption

When there hass a route from an originator node to a destination node, an inside attacker X_M can break the route down via broadcasting a forged *RREQ* message. In this forged *RREQ* message, the attacker feigns to rebroadcast a *RREQ* message started from the destination node to the originator node with a non-existent node as the source IP address in header. Owing to the identical reason presented from scenario 1, the originator node will renew its route to move through the node of non-existent to the destination. Consequently, the route is crushed.

Scenario 4: intruder active forges *RREQ* → broadcast ⇒ route isolation

An attacker could invade a route via generating a forged *RREQ* message actively. Intruder can generate a forged *RREQ* message as described in scenario 2. That is, the forged *RREQ* message should contain: first of all, a *RREQ* ID bigger related to recently *RREQ* ID in the *RREQ* message transmitted via the originator node; secondly, an originator sequence number bigger related to recently originator sequence number; and finally, a destination sequence number bigger related to recently destination sequence number. Then, the attack nodes broadcast this message and feign to send a *RREQ* message from the originator to the destination node.

If the destination node receives the message, it will transmit back a *RREP* message based on the AODV protocol. The *RREP* message will arrive the attack nodes via the opposite route. Then, the attack nodes send it to the originator node. After accepting the *RREP* message, the originator node renews the attack nodes as the next hop to the destination.

Scenario 5: intruder modified *RREP* → broadcast ⇒ route disruption

In a route discovery operation, when the *RREP* message merely goes through an attacker, the attacker will avoid the route from being created via using one of the improvements as follows:

- Modify the message form;
- Change the destination IP address with another IP address;
- Reduce the destination sequence number to a smaller number;
- Change the originator IP address with another IP address;
- Reduce the lifetime area to zero;
- Change the source IP address in header with a non-existent IP address.

Due to the *RREP* message improvements, the originator node can accept a fail *RREP* message or just without *RREP* message. Consequently, the originator will not create a route to the destination node in this route discovery.

Scenario 6: intruder forge reply *RREP* → broadcast ⇒ route disruption

After accepting a *RREQ* message, the attacker could tamper a *RREP* message when it has a fresh enough route to the destination. To maintain other legal *RREP* messages that the originator node could accept from the other nodes, the attacker could tamper a forged *RREP* message as follows:

- Arrange the destination IP address to the IP address of destination node;
- Arrange the originator IP address to the IP address of originator node;
- Arrange the source IP address in header to the non-existent IP address;
- Arrange the destination IP address in header to the attack node receiving the *RREQ* message;
- Increment the destination sequence number via not less than one and reduce the hop count to one.

The attacker uni-casts the forged *RREP* message to the originator node through the opposite route, which was created via the *RREQ* message. After accepting the forged *RREP* message, the attack neighbor node renews the next hop to the destination to the non-existent IP address in header. Before the forged *RREP* message arrives the originator node, the originator had already accepting other legal *RREP* messages.

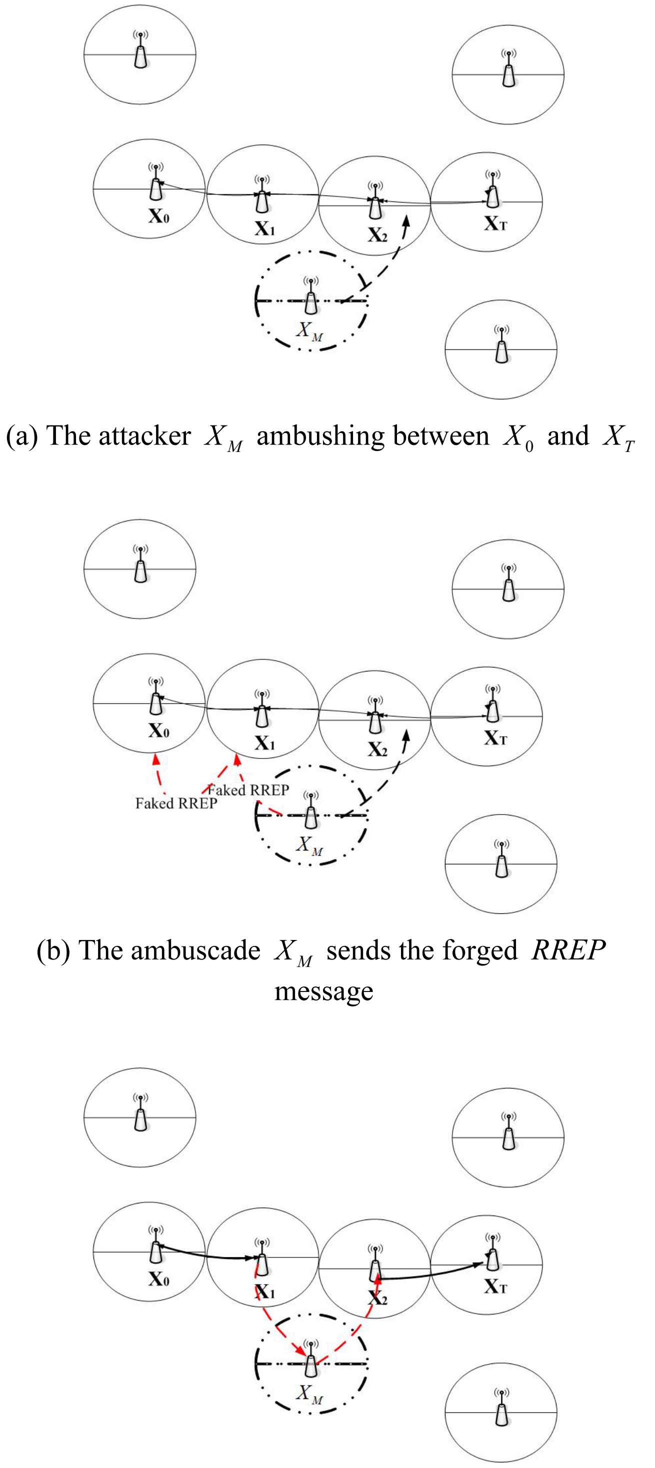
In this scenario, the originator node renews its next hop to the destination node as the neighbor from receiving the forged *RREP* message, because the forged *RREP* has a bigger destination sequence number and a smaller hop count related to the routing table of the originator. Consequently, the subsequently packets from the originator to the destination node can be missed, because they will finally be transmitted to a non-existent node.

Scenario 7: intruder active forge *RREP* → broadcast ⇒ route isolation

Whan an attacker has routes between the originator and the destination nodes of an existent route as presented in Figure 9(a), which intrudes the route via transmittng a forged *RREP* message to the originator. According to Figure 8, suppose X_M is the attack node, which had a route to respective nodes of 0 and 3. The X_M will tamper a *RREP* message:

- Arrange the originator IP address to the originator node X_0;
- Arrange the destination IP address to destination node X_T;
- Arrange the destination sequence number to the sequence number of destination node X_T and increase not less than one one;
- Arrange the source IP address in header to the attack node X_0;
- Arrange the destination IP address in header to one intermediate node X_1.

Then, the X_M node transmits the forged *RREP* message to X_1 node, which sends the forged *RREP* message to X_0 node as presented in Figure 9(b). If both X_0 and X_1 nodes accept the forged *RREP* message, they renew the X_T sequence number in routing tables to the destination sequence number by the forged *RREP* message. The X_0 node still utilizes X_1 node as the next hop to the X_T, but the X_1 renews the X_0 as the next hop to X_T. Besides, the X_M node had a route to the X_T node. Consequently, the X_M node becomes a part of the route from the X_0 node to the X_T as presented in Figure 9(c).



(a) The attacker X_M ambushing between X_0 and X_T

(b) The ambuscade X_M sends the forged $RREP$ message

(c) The ambuscade X_M successfully being a part of the rout

Figure 9.

6 Conclusions

In this paper, an efficient secure routing protocol is proposed to be effectively immune to IP spoofing for WSN-based tactical-level ITS Environments and guarantees the discovery of correct connectivity information over an unknown network. The protocol presents several features, the basic key exchange process, the route request and reply procedures, and the routing maintenance management. The ZKAC protocol is able to operate without the complete knowledge of keys of network nodes and the existence of an on-line certification authority. The only need is that any two nodes that want to securely communicate with each other, and then they can easily build a priori shared secret. Furthermore, the correctness and validation of the protocol is maintained regardless of resident joining node of IP address, a significant importance of security, dynamic, and random tactical-level environments.

References

[1] T.-H. Lin, H. Sanchez, W. J. Kaiser, H. O. Marcy, Wireless Integrated Network Sensors (WINS) for Tactical Information Systems, *Government Microcircuit Applications Conference*, Thousand Oaks, CA, 1998, pp. 1-6.

[2] W. Chen, L. Chen, Z. Chen, S. Tu, A Real time Dynamic Traffic Control System Based on Wireless Sensor Network, *International Conference on Parallel Processing Workshops*, Oslo, Norway, 2005, pp. 258-264.

[3] M. Khanafer, M. Guennoun, H. T. Mouftah, WSN Architectures for Intelligent Transportation Systems, *3rd International Conference on New Technologies, Mobility and Security*, Cairo, Egypt, 2009, pp. 1-8.

[4] Y. Zhou, Y. Fang, Y. Zhang, Securing Wireless Sensor Networks: A Survey, *IEEE Communications Surveys & Tutorials*, Vol. 10, No. 3, pp. 6-28, September, 2008.

[5] T. Kavitha, D. Sridharan, Security Vulnerabilities in Wireless Sensor Networks: A Survey, *Journal of Information Assurance and Security*, Vol. 5, No. 1, pp. 31-44, February, 2010.

[6] F. Anjum, P. Mouchtaris, *Security for Wireless Ad Hoc Networks*, Wiley InterScience, 2007.

[7] T. Haenselmann, Sensor Networks, http://www.informatik.unimannheim.de/~haensel/sn_book

[8] W. Xiao, M. Xu, Y. Chen, A Self-adaptive Fault-Tolerant Mechanism in Wireless Sensor Networks, *International Conference on Scalable Information Systems*, Hong Kong, China, 2009, pp. 228-240.

[9] Y. C. Hu, D. Johnson, A. Perrig, SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks, *4th IEEE Workshop on Mobile Computing Systems and Applications*, Callicoon, NY, 2002, pp. 3-13.

[10] C. E. Perkins, E. M. Royer, Ad-hoc On-Demand Distance Vector Routing, *2nd IEEE Workshop on Mobile computing Systems and Applications*, New Orleans, LA, 1999, pp. 90-100.

[11] P. Papadimitratos, Z. J. Haas, Secure Routing for Mobile Ad hoc Networks, *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, San Antonio, TX, 2002, pp.193-204.

[12] P. Papadimitratos, Z. J. Haas, P. Samar, *The Secure Routing Protocol (SRP) for Ad Hoc Networks*, draft-papadimitratos-

secure-routing-protocol-00.txt, December, 2002.

[13] A. Perrig, R. Canetti, D. Song, D. Tygar, Efficient and Secure Source Authentication for Multicast, *Network and Distributed System Security Symposium*, San Diego, CA, 2001, pp. 35-46.

[14] F. Bergadano, D. Cavagnino, B. Crispo, Chained Stream Authentication, *International Workshop on Selected Areas in Cryptography*, Ontario, Canada, 2000, pp. 144-157.

[15] R. Anderson, F. Bergadano, B. Crispo, J. Lee, C. Manifavas, R. Needham, A New Family of Authentication Protocols, *ACM Operating Systems Review*, Vol. 32, No. 4, pp. 9-20, October, 1998.

[16] C. Gentry, A. Silverberg, Hierarchical ID-based Cryptography, *8th International Conference on the Theory and Application of Cryptology and Information Security*, Queenstown, New Zealand, 2002, pp. 548-566.

[17] S. Cheung, An Efficient Message Authentication Scheme for Link State Routing, *13th Annual Computer Security Applications Conference*, San Diego, CA, 1997, pp. 90-98.

[18] A. Perrig, R. Szewczyk, V. Wen, D. Culler, J. D. Tygar, SPINS: Security Protocols for Sensor Networks, *Seventh Annual International Conference on Mobile Computing and Networks*, Rome, Italy, 2001, pp. 189-199.

[19] V. Torvinen, J. Ylitalo, Weak Context Establishment Procedure for Mobility Management and Multi-Homing, *IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, Windermere, UK, 2004, pp. 111-123.

[20] A. Weimerskirch, D. Westhoff, Zero Common-Knowledge Authentication for Pervasive Networks, *International Workshop on Selected Areas in Cryptography*, Ottawa, Canada, 2003, pp. 73-87.

[21] T. Yao, S. Fukunaga, T. Nakai, Reliable Broadcast Message Authentication in Wireless Sensor Networks, *International Conference on Embedded and Ubiquitous Computing*, Seoul, South Korea, 2006, pp. 271-280.

[22] R. Akbani, T. Korkmaz, G. Raju, HEAP: Hop-by-hop Efficient Authentication Protocol for Mobile Ad-Hoc Networks, *Spring Simulation MultiConference*, San Diego, CA, 2007, pp. 157-165.

[23] B. Lu, U. W. Pooch, A Lightweight Authentication Protocol for Mobile Ad Hoc Networks, *International Journal of Information Technology*, Vol. 11, No. 2, pp. 119-135, 2005.

[24] R. Akbani, T. Korkmaz, G. V. S. Raju, HEAP: A Packet Authentication Scheme for Mobile Ad Hoc networks, *Ad Hoc Networks*, Vol. 6, No. 7, pp. 1134-1150, September, 2008.

[25] S. Zhu, S. Xu, S. Setia, S. Jajodia, LHAP: A Lightweight Hop-by-Hop Authentication Protocol for Ad-Hoc Networks, *23rd International Conference on Distributed Computing Systems Workshops*, Providence, Rhode Island, 2003, pp. 749-755.

[26] F. Ye, H. Luo, S. Lu, L. Zhang, Statistical En-Route Filtering of Injected False Data in Sensor Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 4, pp. 839-50, April, 2005.

[27] W. Zhang, N. Subramanian, G. Wang, Lightweight and Compromise-Resilient Message Authentication in Sensor Networks, *27th Conference on Computer Communications*, Phoenix, AZ, 2008, pp. 2092-2100.

[28] L. Gong, R. Needham, R. Yahalom, Reasoning about Belief in Cryptographic Protocols, *IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, CA, 1990, pp. 234-248.

[29] D. Hurley-Smith, J. Wetherall, A. Adekunle, SUPERMAN: Security Using Pre-Existing Routing for Mobile Ad hoc Networks, *IEEE Transactions on Mobile Computing*, Vol. 16, No. 10, pp. 2927-2940, January, 2017.

[30] Z. Wei, H. Tang, F. R. Yu, M. Wang, P. Mason, Security Enhancements for Mobile Ad Hoc Networks With Trust Management Using Uncertain Reasoning, *IEEE Transactions on Vehicular Technology*, Vol. 63, No. 9, pp. 4647-4658, April, 2014.

[31] D.-J. Deng, S.-Y. Lien, C.-C. Lin, S.-C. Hung, W.-B. Chen, Latency Control in Software-Defined Mobile-Edge Vehicular Networking, *IEEE Communications Magazine*, Vol. 55, No. 8, pp. 87-93, August, 2017.

[32] D.-J. Deng, S.-Y. Lien, J. Lee, K.-C. Chen, On Quality-of-Service Provisioning in IEEE 802.11ax WLANs, *IEEE Access*, Vol. 4, pp. 6086-6104, August, 2016.

[33] Y. Han, N.-N. Xue, B.-Y. Wang, Q. Zhang, C.-L. Liu, W.-S. Zhang, Improved Dual-Protected Ring Signature for Security and Privacy of Vehicular Communications in Vehicular Ad-Hoc Networks, *IEEE Access*, Vol. 6, pp. 20209-20220, April, 2018.

Biographies

Der-Chen Huang received the B.S. degree in electronic engineering from Fung Chia University, Taiwan, in 1983, the MS degree in computer engineering from Florida Institute of Technology, U.S.A., in 1991, and the Ph.D. degree in computer engineering from the Department of Computer Science and Information Engineering, Chung- Cheng University, Chiayi, Taiwan, R.O.C. in 2000. From 1983 to 1989, he worked as a design engineer with the Computer Communication Lab. (CCL)/Industrial Technology Research Institute (ITRI) and Chung-Shan Institute and Science of Technology (CSIST) when he was assigned to a partnership project at General Dynamics, Fort Worth, Texas, U.S.A. He was an associate professor with the Department of Electronic Engineering, National Chinyi Institute of Technology, Taichung, Taiwan, R.O.C. from 1991 to 2004. In 2004, he joined the Department of Computer Science and Engineering, National Chung Hsing University, Taichung, Taiwan, R.O.C. He was a director of Computer and Information Center of Chung Hsing University from 2007 to 2011. Currently, he is a professor of Chung Hsing University. Dr. Huang served as a reviewer for various technical journal and conferences and a member of editorial board of *Journal of Internet Technology*. He received the Best Paper Award from the 5th International Conference on

Future Information Technology, Korea, in 2010. His research interests include VLSI design for testability and diagnosis, VLSI Digital Signal Process, Communication and Medical Image.

Ying-Yi Chu received the M.S. degree in Electronic Engineering from National Chin-Yi University of Technology, Taichung, Taiwan, in 2008. She is currently pursuing the Ph.D. degree in Computer Science and Engineering at National Chung Hsing University, Taichung, Taiwan. Her research interests include VLSI design and network.

Yuan-Kwei Tzeng received the B.S. degree and the M.S. degree in the Department of Applied Mathematics from National Tsing Hua University, in 1983 and 1985, respectively, and the Ph.D. degree in the Department of Computer Science and Engineering from the National Chung Hsing University, in 2012. Mr. Tzeng has been working at Chung-Shan Institute of Science & Technology in the area of intelligent data integration technologies since 1985. His current research aims at the creation and study of Semantic Web for modeling and simulation network security mining applications and information modeling.

Yu-Yi Chen was born in Kaohsiung, Taiwan, in 1969. He received the B.S., M.S., and Ph.D. in Applied Mathematics from the National Chung Hsing University in 1991, 1993, and 1998, respectively. He is presently a professor of the Department of Management Information Systems, National Chung Hsing University, Taiwan. His research interests include computer cryptography, network security, and e-commerce.

Wei-Ming Chen is currently a professor in the Department of Information Management at National Dong Hwa University. He was a professor in the Department of Computer Science and Information Engineering and a Chairman of Computer Science and Information Engineering at National Ilan University. He received his ME, and Ph.D. degrees in Computer Science and Information Engineering from National Chung Cheng University. His current research interests include image processing, computer networks, multimedia system, and database design.