This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

Lightweight and Compromise-Resilient Message Authentication in Sensor Networks

Wensheng Zhang and Nalin Subramanian
Department of Computer Science
Iowa State University
Ames, IA 50011
Email: {wzhang,nvsubram}@cs.iastate.edu

Guiling Wang
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ 07102
Email: gwang@njit.edu

***Abstract*—Numerous authentication schemes have been proposed in the past for protecting communication authenticity and integrity in wireless sensor networks. Most of them however have following limitations: high computation or communication overhead, no resilience to a large number of node compromises, delayed authentication, lack of scalability, etc. To address these issues, we propose in this paper a novel message authentication approach which adopts a perturbed polynomial-based technique to simultaneously accomplish the goals of *lightweight*, *resilience to a large number of node compromises*, *immediate authentication*, *scalability*, and *non-repudiation*. Extensive analysis and experiments have also been conducted to evaluate the scheme in terms of security properties and system overhead.**

I. Introduction

When a sensor network [1] is deployed in an unattended or hostile environment, the adversary may capture and reprogram sensor nodes, or inject their own sensor nodes into the network and induce the network to accept them as legitimate nodes [2]. Once in control of a few sensor nodes, the adversary can mount various attacks from inside the network.

One common type of attack is targeted at message authenticity and integrity. For example, if the sender and the receiver are not within the transmission range of each other, an intruder on the path connecting them can modify pass-by messages or inject false messages. It appears to be a solution that the sender and the receiver share a secret key, and the shared key is used by the sender to generate message authentication code (MAC) for any outgoing message, and by the receivers to verify the authenticity and integrity of any incoming message. If a message is tampered en route, it will be detected by the receiver. This method however is not effective due to the following reasons: First of all, it cannot authenticate messages that are multicast because, if one of the receivers is compromised, the intruder can use the secret key held by the compromised receiver to fake MACs for messages modified or injected by it itself to cheat other receivers. Secondly, the method only allows end-to-end message authentication while en-route forwarding nodes cannot authenticate pass-by messages; as a result, the intruder may launch denial-of-service attacks by repeatedly modifying messages or injecting false messages to deplete the communication resources of intermediate forwarding nodes.

To thwart the above attacks, each message should be verifiable by both its final receivers and its intermediate forwarders. This may be simply implemented on top of the public key infrastructure [3]; specifically, each message is sent along with a digital signature generated by the sender using its private key, and every intermediate forwarder or final receiver can authenticate the message using the public key of the sender. However, this approach may incur high overhead in terms of computational cost and network bandwidth consumption. To mitigate the overhead, researchers have proposed low-cost schemes [4], [5] that use symmetric keys and hash functions. In these schemes, however, each symmetric authentication key is shared by a set of sensor nodes, and the keys can be captured by the intruder as sensor nodes are compromised. Therefore, these schemes are not resilient to large number of node compromises. Utilizing an one-way key chain and delayed disclosure of keys, the TESLA schemes [6] and its variants can achieve message authenticity in the presence of a large number of node compromises. However, these schemes require synchronization among nodes. They introduce delay in message authentication and the delay increases as the network scales up. Moreover, they repel the adoption of asynchronous communication [7].

In this paper, we propose a new message authentication approach to address the aforementioned limitations. Our approach has following features: *lightweight* in terms of computation, communication and storage overhead; *resilience* to a large number of sensor node compromises; *immediate authentication* (therefore supporting both synchronous and asynchronous communication); *scalability*; and *non-repudiation*. These features are attained by applying a number of novel techniques: Firstly, we adopt polynomials for message authentication, which provides higher adaptability than existing authentication techniques based on multiple MACs [4], [5], and at the same time, keeps the advantage of immediate authentication held by those techniques. Secondly, messages are authenticated and verified via evaluating polynomials, which incurs lower overhead than existing asymmetric cryptography-based authentication techniques such as digital signature. Thirdly, independent and random factors are employed to perturb polynomial shares (of a system-wide secret polynomial) that preloaded to individual nodes, which significantly increases

978-1-4244-2026-1/08/$25.00 © 2008 IEEE

the complexity for the intruder to break the secret polynomial, and therefore renders the proposed approach to be resilient to node compromises. To the best of our knowledge, the proposed approach is the first one that applies the aforementioned techniques in message authentication for sensor networks, and also the first one that can achieve simultaneously the features of compromise-resiliency, flexible-time authentication, efficiency and non-repudiation without employing public key cryptography.

The rest of this paper is organized as follows. Section II defines the problem. Section III presents, analyzes and evaluates our proposed schemes. Section IV compares our schemes with related work. Finally, Section V concludes the paper.

II. Problem Definition

A. Network Assumptions

We consider a sensor network that consists of a base station and a certain number of sensor nodes, where each sensor node can be a data source or a data sink. The network support the following communication patterns: (i) the base station broadcasts/multicasts messages to all or a certain set of sensor nodes; (ii) a sensor node broadcasts/multicasts messages to all or a certain set of other sensor nodes; (iii) the base station unicasts messages to a certain sensor node; and (iv) a sensor node unicasts messages to the base station or a certain sensor node. The above communication patterns may be either synchronous (i.e., the receivers are available to receive messages when messages are disseminated) or asynchronous (i.e., when a sender disseminates messages, some desired receivers may not be available; after becoming available, the receivers may obtain the messages from other receivers that have received and cached the messages [7]).

B. Security Assumptions and Attack Model

We assume there is a security server which will never be compromised. Before a sensor node joins the network, it is preloaded by the security server with a unique ID and some security-related information. Similarly, the base station is also preloaded with a unique ID and some security-related information. Sensor nodes are innocent before deployment. However, after deployment, they can be captured and compromised by attackers due to the unattended deployment environments and the lack of tamper resistance. Once being compromised, all information stored in the sensor node can be read out by the attackers. Furthermore, the compromised nodes can be reprogrammed and thus fully controlled by the attackers.

Based on the above assumptions, this paper considers the following types of attacks.

- Outsider attacks: launched by attackers that have not compromised any sensor nodes and therefore do not know any secret of the network. In particular, the attackers may
  - *[type-I attacks]* modify messages or inject their own messages, and attempt to induce en-route nodes and receivers to accept these messages;
  - *[type-II attacks]* eavesdrop and collect messages, and attempt to derive some secrets from the messages.
- Insider attacks: launched by attackers that have compromised some sensor nodes and therefore know some secrets preloaded to these compromised nodes. Certainly, insider attackers can also launch the above type-I and type-II attacks. In addition, the attackers may
  - *[type-III attacks]* collect the secrets owned by compromised nodes, and attempt to derive the secrets held by innocent nodes (and therefore can cheat these innocent nodes or impersonate as them).

In this paper, we call both outsider attackers and insider attackers (e.g., compromised sensor nodes) *intruders*.

C. Design Goals

Our message authentication schemes are designed with the following goals. (i) *Message authenticity*: Intruders shall not be able to impersonate any innocent node to send out a message without being detected. (ii) *Message integrity*: Intruders shall not be able to modify a message sent by any innocent nodes without being detected. (iii) *Non-repudiation*: A node shall not be able to deny the sending of a message. Note that this property is important in some scenarios such as intrusion detection and intruder identification. After a node sends out an accusation report, it should not be able to deny that. This way, malicious accusations can be detected. (iv) *Resilience to a large number of node compromises*: Even if a large number of nodes have been compromised, these nodes shall have very low probability to break our proposed message authentication protocols. (v) *Efficiency*: Our proposed protocols shall have low system overhead in terms of computation, communication and storage. (vi) *Immediate Message Authentication*: A receiver/forwarder of a message shall be able to verify the authenticity and integrity of the message immediately after reception.

III. Proposed Schemes

In this section, we propose a series of message authentication schemes. Our study is conducted evolutionarily through several steps: Firstly, to illustrate the basic idea of polynomial-based message authentication, we present a bivariate polynomial-based scheme (*Scheme-I*) for authenticating message sent from a trustworthy base station to ordinary sensor nodes. Secondly, Scheme-I is enhanced to a perturbed bivariate polynomial-based scheme (*Scheme-II*) such that it can tolerate a large number of sensor node compromises. Thirdly, to authenticate not only the messages sent by the base station but also those sent by any ordinary sensor node, *Scheme-III* is developed by replacing the bivariate polynomial in *Scheme-II* with a three-variable polynomial. Finally, to address a subtle flaw in the Scheme-III, we develop our final scheme (*Scheme-IV*). Before presenting the details, we introduced several common notations as follows.

- q, F_q, l, r, γ: q denotes a prime number, F_q is a prime finite field of order q, l is the integer such that $2^l > q > 2^{l-1}$.

- r, γ: r and γ are integers such that $\gamma < r < l$.

A. Scheme-I: A Basic Bivariate Polynomial-Based Scheme for Authenticating Messages Sent by the Base Station

In this scheme, we only consider the authentication of messages sent from a trusted base station to ordinary sensor nodes.

1) Scheme Specification:

- *Initialization of the Security Server and the Base Station*. Over finite field F_q, the security server randomly picks a secret bivariate polynomial:
$$f(x,y) = \sum_{0 \le i \le d_x, 0 \le j \le d_y} A_{i,j} x^i y^j, \qquad (1)$$
where each coefficient $A_{i,j}$ is an element of F_q, and system parameters d_x, d_y are degrees of x and y, respectively. Then, the security server preloads the base station with $f(x,y)$ and a secure one-way hash function $h(.)$, which could be MD5, SHA, etc.
- *Initialization of Sensor Nodes*. Before a sensor node is deployed, it is preloaded by the security server with:
  - a unique ID u, which is an element of F_q
  - polynomial $verf_u(y) = f(u,y)$, which is called the *verification polynomial* of node u; and
  - the secure one-way hash function $h(.)$.
- *Message Sending at the Base Station*. Assuming the base station wants to send out a message, denoted as m, it executes the following steps to sign m:
  - Hash function $h(.)$ is applied on m to get $h(m)$.
  - Polynomial $f(x,y)$ is evaluated at $y = h(m)$ to get a univariate d_x-degree polynomial $MAF_m(x) = f(x,h(m))$, which is called the *message authentication function* for m.
  - Message $\langle m, MAF_m(x) \rangle$ is sent out, where $MAF_m(x)$ is represented by its $d_x + 1$ coefficients.
- *Message Verification at Sensor Nodes*. When a sensor node with ID u (called node u thereafter) receives message $\langle m, MAF_m(x) \rangle$, it executes the following steps to verify the authenticity and integrity of the message:
  - $h(.)$ is applied on m to get $h(m)$.
  - $verf_u(y)$ is evaluated at $y = h(m)$ to get $verf_v(u,h(m))$.
  - Received $MAF_m(x)$ is evaluated at $x = u$ to get $MAF_m(u)$.
  - If and only if $verf_v(u,h(m)) = MAF_m(v)$, the received message is regarded as authentic and intact.

2) An Example: Fig. 1 shows an example of how Scheme-I works. Here, $q = 31$; that is, all arithmetic operations are over finite field F_{31}. The base station is preloaded with $f(x,y) = x^2y^2 + 2x^2y + 3x^2 + 4xy^2 + 5xy + 6x + 7y^2 + 8y + 9$. Node 3 has $verf_3(y) = 28y^2 + 10y + 23$ and node 9 has $verf_9(y) = 29y + 27$. Suppose the base station wants to broadcast message m, where $h(m) = 29$. It computes $MAF_m(x) = f(x,h(m)) = 3x^2 + 12x + 21$ and sends out m along with $MAF_m(x)$. On receiving the message, node 3 evaluates both $MAF_m(3)$ and $verf_3(h(m))$. Since $MAF_m(3) = verf_3(h(m)) = 22$, the message is verified as authentic and integral. On the other hand, assume that node 6 is an intruder (compromised node). It modifies message m to m' where $h(m') = 1$. Node 6 does not know $f(x,y)$ for computing correct $MAF_{m'}(x)$. Assume that it arbitrarily fakes a $MAF'_{m'}(x) = 2x^2 + 3x + 4$ and forwards m' together with $MAF'_{m'}(x)$. On receiving this faked message, node 9 evaluates $MAF'_{m'}(9) = 7$ and $verf_9(h(m')) = 25$. Since $MAF'_{m'}(9) \neq verf_9(h(m'))$, the message fails in verification and is dropped.

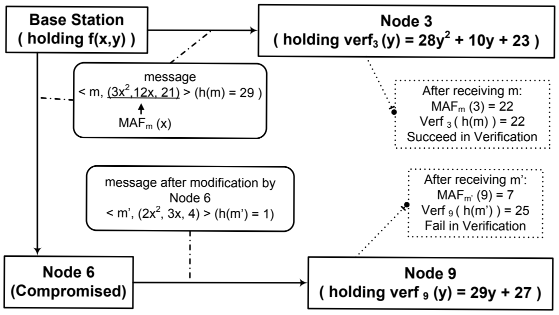

Fig. 1. An Example of Scheme-I

3) Discussion: While Scheme-I provides an efficient mechanism for message authentication, it has following limitations.

- If the intruder has received and recorded $d_y + 1$ or more message authentication functions (e.g., $MAF_{m_0}(x) = f(x,h(m_0)), \cdots, MAF_{m_{d_y}}(x) = f(x,h(m_{d_y}))$) that are sent by the base station, the intruder can derive $f(x,y)$. The reason is as follows. The recorded functions are $d_y + 1$ shares of $f(x,y)$. Since the degree of y in $f(x,y)$ is $d_y + 1$, $f(x,y)$ can be derived via interpolation based on these shares.
- If $d_x + 1$ or more sensor nodes (e.g., as $v_0, \cdots, v_{d_x}$) have been compromised, they can collude by sharing their verification functions (i.e., $verf_{v_0}(y) = f(v_0,y)$, $\cdots$, $verf_{v_{d_x}}(y) = f(v_{d_x},y)$). Since the degree of x is d_x in $f(x,y)$, polynomial $f(x,y)$ can be derived through interpolation based on these verification functions.

Therefore, d_x and d_y have to be as large as possible in order to achieve a high level of security. On the other hand, each sensor node needs to store a verification function, that is, $d_y + 1$ coefficients of the function; each message sent by the based station must include a message authentication function, that is, $d_x + 1$ coefficients of the polynomial. For storage and communication efficiency, both d_x and d_y should therefore be as small as possible. To resolve the *dilemma*, we propose Scheme-II in the following.

B. Scheme-II: A Perturbed Bivariate Polynomial-Based Scheme for Authenticating Messages Sent by the Base Station

We propose Scheme-II to overcome the aforementioned limitations of Scheme-I. Based on the idea of perturbing polynomial shares with randomly-picked numbers, borrowed from [8], this scheme differs from Scheme-I mainly in the following aspects:

- Firstly, when the base station sends out a message m, its authentication function $MAF_m(x)$ will not be $f(x, h(m))$; instead, it shall be a perturbed version of $f(x, h(m))$, i.e., $MAF_m(x) = f(x, h(m)) + s_m$, where s_m is a number picked from F_q. After the perturbation, even an intruder has collected $d_y + 1$ or more messages (and hence $d_y + 1$ or more perturbed authentication functions), it cannot obtain any *exact* shares of $f(x, y)$. Consequently, as we will show later, the complexity for deriving $f(x, y)$ will be very high.
- Secondly, when the security server distributes verification functions to sensor nodes, these functions shall not be exact shares of $f(x, y)$, but perturbed ones. Specifically, the verification function for node u shall be $verf_u(y) = f(u, y) + r_u$, where r_u is a number picked from F_q. This way, even the adversary has compromised $d_x + 1$ or more sensor nodes (and hence $d_x + 1$ or more perturbed verification functions), they cannot obtain any exact share of $f(x, y)$. As we will show later, the complexity for deriving $f(x, y)$ from the captured verification functions will also be very high.

After using the above perturbation mechanisms, the process for verifying a message will certainly not be as straightforward as in Scheme-I, and this issue is addressed in Scheme-II.

1) Scheme Specification:

- *Initialization of the Security Server and the Base Station*. The security server constructs a secret bivariate polynomial $f(x, y)$, the same as Eq. (1), over finite field F_q. As introduced at the beginning of Section III, we let l be an integer such that $2^{l-1} < q < 2^l$ and system parameter r be an integer smaller than l, the security server preloads the base station with $f(x, y)$, a secure one-way hash function $h(.)$ and system parameter r.
- *Initialization of Sensor Nodes*. Before a sensor node is deployed, the security server preloads it with
  - a unique ID u, which is an element of F_q;
  - a perturbed verification polynomial $verf_u(y) = f(u, y) + r_u$, where r_u is a number randomly picked from $\{0, \cdots, 2^r - 1\}$;
  - the secure one-way hash function $h(.)$; and
  - system parameter r.
- *Message Sending at the Base Station*. Suppose the base station wants to send out a message m. The following steps shall be executed:
  - Hash function $h(.)$ is applied on m to get $h(m)$.
  - Polynomial $f(x, y)$ is evaluated at $y = h(m)$ to get a univariate d_x-degree polynomial $f(x, h(m))$.
  - The message authentication function for m, i.e., $MAF_m(x)$, is computed as $f(x, h(m)) + s_m$, where s_m is a number randomly picked from $\{0, \cdots, 2^r - 1\}$. Note that the choice of s_m is independent to the message.
  - Message $\langle m, MAF_m(x) \rangle$ is sent out.
- *Message Verification at Sensor Nodes*. When message $\langle m, MAF_m(x) \rangle$ is received at sensor node u, the message is verified by testing if $verf_u(h(m)) - MAF_m(u)$ belongs to $\{0, \cdots, 2^r - 1, q - (2^r - 1), \cdots, q - 1\}$. The principle behind this step is explained as follows: Because $MAF_m(x) = f(x, h(m)) + s_m$ and $verf_u(y) = f(u, y) + r_u$, we have

$$\begin{aligned} & verf_u(h(m)) - MAF_m(u) \\ = \; & f(u, h(m)) + r_v - [f(u, h(m)) + s_m] \\ = \; & r_u - s_m. \end{aligned}$$

Also due to $s_m \in \{0, \cdots, 2^r - 1\}$ and $r_u \in \{0, \cdots, 2^r - 1\}$, we have

$$r_v - s_m \in \{0, \cdots, 2^r - 1, q - (2^r - 1), \cdots, q - 1\}.$$

Therefore, $verf_u(h(m)) - MAF_m(u)$ must also belong to this set.

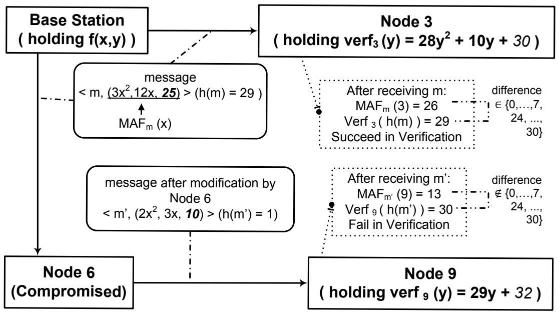

Fig. 2. Example of Scheme-II

2) An Example: Fig. 2 shows an example on how Scheme-II works. Here, we set q to 31, l to 5 and r to 3. Each sensor node is preloaded with perturbed shares of $f(x, y)$. In particular, node 3 has $verf_3(y) = f(3, y) + 7 = 28y^2 + 10y + 30$ (parameter r_3 is set to 7), and node 9 has $verf_9(y) = f(9, y) + 5 = 29y + 32$ (parameter r_9 is set to 5). When the base station disseminates message m, $MAF_m(x) = f(x, h(m)) + 4$ (parameter $s_m = 4$) is also sent out. On receiving the message, node 3 evaluates $MAF_m(3) = 26$ and $verf_3(h(m)) = 29$. Since their difference is within $\{0, \cdots, 2^r - 1 = 7, q - (2^r - 1) = 24, \cdots, 30\}$, the message is verified as authentic and integral. On the other hand, node 9 receives message m faked by node 6, an intruder. It evaluates $MAF_m(9) = 13$ and $verf_9(h(m)) = 30$. Since the difference is not within $\{0, \cdots, 7, 24, \cdots, 30\}$, the message is not accepted.

3) Security Analysis and Evaluation:

a) Capability against type-I attacks: When forwarding a message m sent by the base station, an intruder (either insider or outsider) may modify the message to m', and arbitrarily change the message authentication function associated with the message in order to hide the modification from receivers. The following Theorem 3.1 shows the capability of Scheme-II against such attacks.

Theorem 3.1: If a message m sent by the base station is modified to $m' \neq m$, where $h(m') \neq h(m)$, the probability that the message is be verified by node u as valid is less than $\frac{1}{2^{l-r-1}}$.

Proof: (sketch) When message m', along with an arbitrary $MAF'_m(x)$, arrives at sensor node u, u will compute $verf_u(h(m')) - MAF'_m(u)$. Due to the arbitrariness, $verf_u(h(m'))-MAF'_m(u)$ could be any element in $\{0, \cdots q-1\}$. Therefore, the probability that it is in $\{0, \cdots, 2^r - 1, q - (2^r - 1), \cdots, q-1\}$ is $\frac{2^{r+1}}{q} < \frac{1}{2^{l-r-1}}$. ■

Note that it is possible $h(m') = h(m)$ when $m' \neq m$. However, we do not consider this in this paper; we assume the hash function is a secure one-way function and hence the probability of collision should be very low.

b) Capability against type-II attacks: An intruder (either insider or outsider) may collect the message authentication functions sent by the base station for $d_y + 1$ or more different messages, and attempts to derive $f(x, y)$ based on the collected information. Note that once $f(x, y)$ is compromised, Scheme-II is broken. The following Theorem 3.2 shows the capability of Scheme-II against such attacks.

Theorem 3.2: If an intruder has obtained $n \geq d_y + 1$ message authentication functions, denoted as $MAF_{m_i}(x) = f(x, h(m_i)) + s_i$ $(i = 0, \cdots, n-1)$, the complexity for the adversary to break $f(x, y)$ based on the captured functions is $\Omega(2^{r*(d_y+1)})$.

Proof: (sketch) Assume v is an arbitrary element in F_q. Let us consider the complexity for breaking $f(v, y)$. Let $f(v, y) = \sum_{j=0}^{d_y} C_j y^j$. Based on the captured message authentication functions, the adversary obtains the following system of linear equations:

$$\sum_{j=0}^{d_y} C_j [h(m_i)]^j + s_i = MAF_{m_i}(v), i = 0, \cdots, n-1.$$

Here, each s_i represents an arbitrary element in $\{0, \cdots, 2^r - 1\}$. All C_j $(j = 0, \cdots, d_y)$ and s_i $(i = 0, \cdots, n-1)$ are unknowns. Therefore the total number of unknowns is $d_y + n + 1$, which is greater than the number of equations (i.e., n). To solve the linear system, at least $d_y + 1$ unknowns should be eliminated; that is, the values of $d_y + 1$ unknowns should be found out correctly. Since each s_i is of r bits, which is shorter than any coefficient C_i, the adversary must choose to eliminate $d_y + 1$ of s_i's. Because s_i is randomly picked from a set of 2^r elements, and the solution for $f(v, y)$ is unique, the expected time complexity to find out the right values for these s_i's is $\Omega(2^{r*(d_y+1)})$. ■

Note that we only consider the case that $n \geq d_y + 1$. This is because, if $n \leq d_y$, it is impossible for the adversary to derive $f(x, y)$, in which the degree of y is d_y.

c) Capability against type-III attacks: Compromised sensor nodes may collude by sharing their preloaded verification functions to find out the secret polynomial $f(x, y)$, and thus the authentication mechanisms can be broken. The following Theorem 3.3 shows the capability of Scheme-II against such attacks.

Theorem 3.3: If the adversary has captured $n \geq d_x + 1$ sensor nodes and thus has obtained n message verification functions, denoted as $very_{u_i}(y) = f(u_i, y) + r_i$ $(i = 0, \cdots, n-1)$, the complexity for the adversary to break $f(x, y)$ based on the captured functions is $\Omega(2^{r*(d_x+1)})$.

Proof: (similar to the proof of Theorem 3.2) ■

C. Scheme-III: A Three Variable Polynomial-Based Scheme for Authenticating Messages Sent by Any Node

Scheme-II can only authenticate messages sent from the trustworthy base station to ordinary sensor nodes. However, as discussed in Section II, any ordinary sensor node may also send out messages and the authenticity and integrity of these messages can also be attacked. To address this issue, we extend Scheme-II, which is based on bivariate polynomials, to this scheme, which is instead based on three variable polynomials.

1) Scheme Specification:

- *System Initialization*. The security server constructs a secret polynomial $f(x, y, z)$ over finite field F_q, where the degree of x, y and z are d_x, d_y and d_z, respectively. Similar to Scheme-II, we let l be the integer such that $2^{l-1} < q < 2^l$, and system parameter r be an integer smaller than l.
- *Node initialization*. The security server preloads each node (either a sensor node or base station) with
  - a unique ID u;
  - polynomial $auth_u(y, z) = f(u, y, z) + r_{u,0}$ ($r_{u,0}$ is randomly picked from $\{0, \cdots, 2^{r-1} - 1\}$), which is used for generating message authentication functions for outgoing messages;
  - polynomial $verf_u(x, z) = f(x, u, z) + r_{u,1}$ ($r_{u,1}$ is randomly picked from $\{0, \cdots, 2^r - 1\}$), which is used for verifying incoming or passby messages;
  - secure one-way hash function $h(.)$; and
  - system parameter r.
- *Message Sending at Senders*. When a node u wants to send out a message m, the following steps are performed:
  - Polynomial $auth_u(y, z)$ is evaluated at $z = h(m)$ to get a univariate d_y-degree polynomial $auth_u(y, h(m))$, and message authentication function $MAF_{u,m}(y)$ is set to $auth_u(y, h(m)) + s_{u,m}$, where $s_{u,m}$ is a number randomly picked from $\{0, \cdots, 2^{r-1}\}$.
  - Message $\langle u, m, MAF_{u,m}(y) \rangle$ is sent out.
- *Message Verification at Receivers*. When message $\langle u, m, MAF_{u,m}(y) \rangle$ is received at node v, the message is verified by testing if $verf_v(u, h(m)) - MAF_{u,m}(v)$ belongs to $\{0, \cdots, 2^r - 1, q - (2^r - 1), \cdots, q-1\}$.

2) Discussion: This scheme appears to work, but it has a subtle security flaw which can be exploited by the intruder. We elaborate the attack, which we call *reflection attack*, as follows. Suppose the intruder has compromised $d_y + 2$ nodes and captured the secret functions $auth_{u_i}(y, z)$ and $verf_{u_i}(x, z)$ $(i = 0, \cdots, d_y + 1)$ stored in these nodes; u and w be two arbitrary elements of F_q. Let a_i denote $verf_{u_i}(u_0, w) - auth_{u_0}(u_i, w)$ $(i = 0, \cdots, d_y + 1)$. Then, we have

$$\begin{aligned} a_i &= verf_{u_i}(u_0, w) - auth_{u_0}(u_i, w) \\ &= f(u_0, u_i, w) + r_{u_i,1} - (f(u_0, u_i, w) + r_{u_0,0}) \\ &= r_{u_i,1} - r_{u_0,0}. \end{aligned} \tag{2}$$

This is equivalent to

$$r_{u_i,1} = r_{u_0,0} + a_i \tag{3}$$

Let $f(u, y, w) = \sum_{j=0}^{d_y} C_j y^j$. Then, we have

$$\sum_{j=0}^{d_y} (u_i)^j C_j = verf_{u_i}(u, w) - r_{u_0,0} - a_i, \; i = 0, \cdots, d_y + 1. \tag{4}$$

Since for every $i, j = 0, \cdots, d_y + 1$, the adversary knows u_i, $verf_{u_i}(u, w)$ and a_i. There are $d_y + 2$ unknowns including $r_{u_0,0}$ and C_j ($j = 0, \cdots, d_y$), and $d_y + 2$ linear equations. So, $f(u, y, w)$ can be solved. Since u and w can be arbitrary elements of F_q, the adversary can modify the message sent by any arbitrary node u, or inject an arbitrary message in the name of node u, without being detected. To address the above security flaw, we propose a new scheme in the following.

D. Scheme-IV: The Final Scheme

To address the above flaw in Scheme-III, as well as enable any node to authenticate any outgoing messages any verify any pass-by/incoming messages, we propose Scheme-IV. Different from Scheme-III, we borrow the idea of randomly-constructed polynomials (called perturbation polynomials) [9] to perturb shares of $f(x, y, z)$, so as further increase the difficulty to derive $f(x, y, z)$ from its perturbed shares.

1) Scheme Specification:

- *System Initialization*. Similar to Scheme-III, the security server randomly constructs polynomial $f(x, y, z)$ over F_q, where the degrees of x, y and z are d_x, d_y and d_z, respectively.
- *Constructing a perturbation polynomial for message verification purpose and an ID space for senders*. The security server randomly constructs a univariate d_x-degree polynomial $\alpha(x)$ over F_q. Based on $\alpha(x)$, all elements in F_q can be divided into $2^{l-(r-\gamma-1)}$ sets

  $$S_i = \{x | x \in F_q, \alpha(x) - i * 2^{r-\gamma-1} \in \{0, \cdots, 2^{r-\gamma-1} - 1\}\}$$

  for $i = 0, \cdots, 2^{l-(r-\gamma-1)} - 1$.
  Let S_k be the largest set among all these S_i's. Thus, the size of S_k (denoted as $\|S_k\|$) must be at least $\frac{q}{2^{l-(r-\gamma-1)}} \geq \frac{2^{l-1}}{2^{l-(r-\gamma-1)}} = 2^{r-\gamma-2}$.
  Let us use $\mathcal{I}_s$ (called the ID space for senders) to denote S_k, and use $\overline{\alpha}(x)$ (called the perturbation polynomials for verification) to denote $\alpha(x) - k * 2^{r-\gamma-1}$. Hence, for any $u \in \mathcal{I}_s$ and $r_{u,0} \in \{0, \cdots, 2^\gamma\}$, we have $r_{u,0} * \overline{\alpha}(u)$ must be in $\{0, \cdots, 2^{r-1} - 1\}$. As to be described later, $\mathcal{I}_s$ will be used in message authentication while $\overline{\alpha}(x)$ will be used in message verification.
- *Constructing a perturbation polynomial for authentication purpose and an ID space for receivers/forwarders*. Similar to the previous step, the security server randomly constructs a univariate d_y-degree polynomials $\beta(y)$ over F_q. Based on $\beta(y)$, all elements in F_q can be divided into $2^{l-(r-\gamma-1)}$ sets

  $$R_i = \{y | y \in F_q, \beta(y) - i * 2^{r-\gamma-1} \in \{0, \cdots, 2^{r-\gamma-1} - 1\}\}$$

  for $i = 0, \cdots, 2^{l-(r-1)} - 1$.
  Let us use $\mathcal{I}_r$ (called the ID space for receivers/forwarders) to denote $R_{k'}$, and use $\overline{\beta}(y)$ (called the perturbation polynomial for authentication) to denote $\beta(y) - k' * 2^{r-\gamma-1}$. Hence, for any $v \in \mathcal{I}_r$ and any $r'_{v,0} \in \{0, \cdots, 2^\gamma\}$, $r' * \overline{\beta}(v)$ must be in $\{0, \cdots, 2^{r-1} - 1\}$. As to be described later, $\mathcal{I}_r$ will be used in message verification, while $\overline{\beta}(y)$ will be used in message authentication.
- *Node Initialization*. The security server preloads each node u (either an ordinary node or base station) with
  - a unique sender ID denoted as u_s, where $u_s \in \mathcal{I}_s$;
  - a unique receiver ID denoted as u_r, where $u_r \in \mathcal{I}_r$;
  - polynomial $auth_u(y, z)$ for authenticating outgoing messages, where

    $$auth_u(y, z) = f(u_s, y, z) + r_{u,0} * \overline{\beta}(y) + r_{u,1},$$

    $r_{u,0}$ is randomly picked from $\{0, \cdots, 2^\gamma\}$ and $r_{u,1}$ is randomly picked from $\{0, \cdots, 2^{r-2}\}$;
  - verification polynomial $verf_u(x, z)$ for verifying passby/incoming messages, where

    $$verf_u(x, z) = f(x, u_r, z) + r'_{u,0} * \overline{\alpha}(x) + r'_{u,1},$$

    $r'_{u,0}$ is randomly picked from $\{0, \cdots, 2^\gamma\}$ and $r'_{u,1}$ is randomly picked from $\{0, \cdots, 2^{r-1}\}$;
  - secure one-way hash function $h(.)$; and
  - system parameter r.
- *Message Sending at Senders*. When a sender u wants to send message m, it constructs a message authentication function $MAF_{u,m}(y) = auth_u(y, h(m)) + s_{u,m}$, where $s_{u,m}$ is randomly picked from $\{0, \cdots, 2^{r-2}\}$. Then, message $\langle u, m, MAF_{u,m}(y) \rangle$ is sent out.
- *Message Verification at Receivers/Forwarder* When the message is received at receiver/forwarder v, the message is verified by testing if $verf_v(u, h(m)) - MAF_{u,m}(v) \in \{0, \cdots, 2^r - 1, q - (2^r - 1), \cdots, q - 1\}$. The principle behind this step is explained as follows:
  According to the above algorithm, we have

  $$\begin{aligned} & verf_v(u_s, h(m)) - MAF_{u,m}(h(m)) \\ = \; & [f(u_s, v_r, h(m)) + r'_{v,0} * \overline{\alpha}(u_s) + r'_{v,1}] \\ & -[f(u_s, v_r, h(m)) + r_{u,0} * \overline{\beta}(v_r) + r_{u,1} + s_{u,m}] \\ = \; & [r'_{v,0} * \overline{\alpha}(u_s) + r'_{v,1}] - [r_{u,0} * \overline{\beta}(v_r) + r_{u,1} + s_{u,m}] \end{aligned}$$

  Due to

  $$\begin{aligned} & r'_{v,0} * \overline{\alpha}(u_s) \in \{0, \cdots, 2^{r-1} - 1\} \\ & \wedge r'_{v,1} \in \{0, \cdots, 2^{r-1}\} \\ & \Longrightarrow r'_{v,0} * \overline{\alpha}(u_s) + r'_{v,1} \in \{0, \cdots, 2^r - 1\} \end{aligned}$$

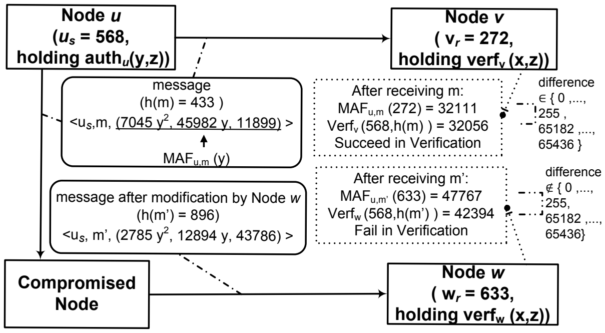

Fig. 3. An Example of Scheme-IV

and

$$
\begin{aligned}
& r_{u,0} * \overline{\beta}(v_r) \in \{0, \cdots, 2^{r-1} - 1\} \\
& \wedge r_{u,1} \in \{0, \cdots, 2^{r-2}\} \\
& \wedge s_{u,m} \in \{0, \cdots, 2^{r-2}\} \\
& \Longrightarrow r_{u,0} * \overline{\beta}(v_r) + r_{u,1} + s_{u,m} \in \{0, \cdots, 2^r - 1\},
\end{aligned}
$$

$verf_v(u_s, h(m)) - MAF_{u,m}(v_r)$ must be in

$$\{0, \cdots, 2^r - 1, q - (2^r - 1), \cdots, q - 1\}.$$

2) An Example: We present an example in the following to illustrate the working of Scheme-IV.

- Assume the security sever sets system parameters q to 65437, l to 16, r to 8, γ to 2, and $f(x, y, z)$ to $6x^2y^2z^2 + 9x^2y^2z + 2x^2yz^2 + 3x^2yz + 12xy^2z^2 + 18xy^2z + 4xyz^2 + 6xyz + 4x^2z^2 + 6x^2z + 3x^2y^2 + x^2y + 6xy^2 + 2xy + 8xz^2 + 12xz + 18y^2z + 27y^2z + 6yz^2 + 9yz + 2x^2 + 4x + 9y^2 + 3y + 12z^2 + 18z + 6$.
- The security server randomly constructs
$$\alpha(x) = 2x^2 + 4,$$
Based on $\alpha(x)$, set $\{0, \cdots, q - 1\}$ is divided into $2^{l-(r-\gamma-1)} = 2048$ subsets: $S_0, \cdots, S_{2047}$. Among these, the largest one is S_{1759} where $|S_{1759}| = 50$. Therefore, the security server gets the ID set for senders
$$I_s = S_{1759} = \{568, 1142, 1501, \cdots\},$$
and the perturbation polynomial for verification
$$\overline{\alpha}(x) = \alpha(x) - 1759 * 2^{r-\gamma-1} = 2x^2 - 56284$$
- Similarly, the security sever randomly constructs
$$\beta(y) = 3y^2 + 4y + 5,$$
Based on them, the security sever gets the ID set for forwarders/receivers
$$I_r = R_{835} = \{272, 633, 1388, \cdots\}$$
as well as the perturbation polynomial for authentication
$$\overline{\beta}(y) = 3y^2 + 4y - 26715,$$

Next, we consider a scenario shown in Fig. 3. Let us assume node u wants to send a message to nodes v and w. Node u was preloaded $u_s = 568$ (from $\mathcal{I}_s$) and authentication polynomial $auth_u(y, z) = f(u_s, y, z) + 3 \times \overline{\beta}(y) + 48$. Node v was preloaded $v_r = 272$ (from $\mathcal{I}_r$) and verification polynomial $verf_v(x, z) = f(x, v_r, z) + 2 \times \overline{\alpha}(x) + 29$. Node w was preloaded $w_r = 633$ (from $\mathcal{I}_r$) and verification polynomial $verf_w(x, z) = f(x, w_r, z) + 3 \times \overline{\alpha}_1(x) + 61$.

For a message m, let $h(m) = 433$. Node u computes $MAF_{u,m}(y) = auth_u(y, h(m)) = 7045y^2 + 45982y + 11899$ and send out u_s, m along with $MAF_{u,m}(y)$. On receiving the message, node v evaluates both $MAF_{u,m}(v_r = 272) = 32111$ and $verf_v(u_s = 568, h(m) = 433) = 32056$. Since their difference is within

$$\{0, \cdots, 2^r - 1 = 255, q - (2^r - 1) = 65182, \cdots, 65436\},$$

the message is verified as authentic and integral. On the other hand, node w receives message m faked as m' and $MAF_{u,m'}$ by a compromised intermediate node (an intruder). It evaluates $MAF_{u,m'}(w_r = 633) = 47767$ and $verf_w(u_s = 568, h(m') = 896) = 42394$. Since the difference is not within $\{0, \cdots, 255, 65182, \cdots, 65436\}$, the message is not accepted.

3) Security Analysis and Evaluation:

a) Capability against type-I attacks:

Theorem 3.4: If a message m sent by the base station is modified to $m' \neq m$, where $h(m') \neq h(m)$, before it reaches an innocent node u, the probability that the message is be verified by node u as valid is $\frac{1}{2^{l-r-1}}$.

Proof: (similar to the proof of Theorem 3.1) ■

b) Capability against type-II attacks:

Theorem 3.5: If the intruder has obtained $n \geq d_y + 1$ message authentication functions, denoted as $MAF_{u,m_i}(y)$ ($i = 0, \cdots, n - 1$), the complexity for the adversary to break $f(u, y, z)$ based on the captured functions is $\Omega(2^{r*(d_y+1)})$.

Proof: (similar to the proof of Theorem 3.2) ■

c) Capability against type-III attacks: This is formally stated in the following Theorem 3.6.

Theorem 3.6: (i) If the intruder has compromised $n \leq min\{d_x, d_y\}$ nodes, it cannot break $f(x, y, z)$; (ii) if the intruder has compromised $n > min\{d_x, d_y\}$ nodes, the complexity for it to break $f(x, y, z)$ is $\Omega(2^{min\{r,2\gamma\}*min\{d_x+1,d_y+1\}})$.

Proof: (see Appendix A) ■

4) Implementation and Performance Evaluation: We implement Scheme-IV on top of the Mica2 Mote/TinyOS platform, and simulate it with TOSSIM [10]. Based on the implementation, we measured (i) computational overhead in terms of the delay for message authentication and verification; (ii) memory/storage overhead in terms of the required ROM and RAM consumption; and (iii) communication overhead in terms of the size of a message authentication function (MAF).

We vary system parameters l, d_x, d_y, d_z, r and γ (Note that all the settings ensure that the complexity to breaking the secret polynomial $f(x, y, z)$ is at least 2^{64} according to Theorems 3.5 and 3.6, and the measured results are shown in Table I, which also shows the probability that a message is fabricated without being detected (estimated based on Theorem 3.4). Note that, the probability (e.g., 0.015 when

$l = 24$ and $r = 17$) is higher than using public key techniques. However, we argue that it will not lead to the scenario that bogus messages are flooded over the network, because each message is checked independently by nodes that it passes and will be filtered within a small number of hops (e.g., if $P = 0.015$, a bogus message is detected and filtered within 2 hops with probability $1 - 2^{-18}$ and within 10 hops with probability $1 - 2^{60}$).

As shown in the table, the *computation overhead* for message authentication and verification ranges from $6ms$ to $85ms$. This is much smaller than applying public key cryptographic techniques [11]–[13] to sensor motes. The evaluation results demonstrate the range of ROM consumption (ranging from 15 KB to 23 KB) and RAM consumption (ranging from 2 KB to 3.4 KB), which are acceptable storage overhead for current generation of sensor nodes. The table also shows the size of MAF, which ranges from 12 to 24 bytes. The size is larger than the size of a single MAC in TESLA-based authentication schemes and in some cases may be larger than the size of multiple MACs in Multi-MAC-based schemes. However, the overhead can be mitigated as the packet size increases. Note that, although the default packet is 29 bytes in TinyOS, the actually packet size of wireless sensor network could be larger (IEEE 802.15.4 allows packet size of up to 128 bytes), and recent research [14] has verified the efficiency and reliability when the packet size is more than 100 bytes. The overhead can also be mitigated by sending one authentication function for every a certain number of consecutive messages instead of having one authentication function for each message.

TABLE I

PERFORMANCE RESULTS OF SCHEME III [$d_x = 80, \gamma = 8$, MAF: THE SIZE OF MESSAGE AUTHENTICATION FUNCTION (MAF), P: THE SUCCESS PROBABILITY OF TYPE-I ATTACKS (NOTE: THE COMPLEXITY FOR TYPE-II AND TYPE-III ATTACKS ARE BOTH HIGHER THAN 2^{64}), N: THE SIZE OF SENDER/RECEIVER ID SPACE (I.E., THE MAXIMUM NUMBER OF NODES CAN BE SUPPORT)]

l	r	d_y (d_z)	ROM (KB)	RAM (B)	MAF (B)	Sign (ms)	Verf (ms)	P	N
24	17	3	14.766	1938	12	5.8	57.89	0.015	2^7
24	19	3	14.766	1938	12	6.44	57.85	0.06	2^9
24	17	4	15.036	2211	15	7.59	70.8	0.015	2^7
24	19	4	15.036	2211	15	7.72	70.5	0.06	2^9
24	17	5	15.328	2490	18	9.31	84.22	0.015	2^7
24	19	5	15.328	2490	18	8.54	83.49	0.06	2^9
32	25	3	21.832	2622	16	6.31	57.13	0.015	2^{15}
32	27	3	21.832	2622	16	5.58	58	0.06	2^{17}
32	25	4	22.200	2986	20	6.85	70.64	0.015	2^{15}
32	27	4	22.200	2986	20	6.72	70.74	0.06	2^{17}
32	25	5	22.976	3359	24	9.1	83.8	0.015	2^{15}
32	27	5	22.976	3359	24	9.44	84.35	0.06	2^{17}

5) Summary of Analysis and Evaluation Results: The above analysis and evaluation verify that our proposed scheme has the following features.

- Lightweight and scalability: As shown in Section III-D 4), our proposed scheme can support a large-scale sensor network (for example, having up to 2^{17} sensor nodes). And the overhead for computation (i.e., signing and verifying messages), communication and storage is low or moderate.
- Resilience to node compromise: Sections III-D 3) and 4) demonstrate that, if system parameters l, d_x, d_y, d_z, r and γ are appropriately chosen, our proposed scheme is resilient to a large number of node compromise because breaking the scheme will require prohibitively high computational complexity.
- Immediate and effective authentication: The low verification delay shown in Section III-D 4) and Theorem 3.4 in Section III-D 3) demonstrate that our proposed scheme enables immediate authentication, and the authentication is effective if system parameters l and r are appropriately chosen.
- Non-repudiation: Every sensor node is preloaded with unique authentication functions, and the probability for breaking these functions is low when system parameters are appropriately chosen. Therefore, the source of a message can be determined the message authentication function used.

IV. RELATED WORK

Digital Signature-Based Approaches. As the most natural approach, the public key cryptography may be applied to generate digital signatures [3] for message authentication. However, adopting this approach to resource constrained wireless sensor networks may either (i) results in high computational cost or (ii) requires special hardware supports [11]–[13], [15]. *Our proposed approach only involves simple arithmetic operations (i.e., polynomial evaluations over a finite field) and low-cost hash functions; hence, it has much lower overhead (i.e., a few milliseconds in authentication and tens of milliseconds in verification).*

Multiple Message Authentication Code-Based Approaches. Some researchers [4], [5] proposed to use hash functions to produce multiple message authentication codes (MACs) to authenticate messages. These schemes are more efficient than the approach based on public key cryptography. However, because each secret key is shared by multiple nodes, these schemes become ineffective or even useless if a large number of nodes are compromised. Moreover, these schemes cannot achieve non-repudiation. *Our proposed approach is also computationally efficient. Different from these schemes, our proposed approach can tolerate a large number of node compromises and can achieve non-repudiation.*

TESLA and Its Variants. Assuming time synchronization among nodes as well as the sharing of initial secrets between authenticators and verifiers, the TESLA scheme and its variants [6] can perform delayed authentication in the presence of a large number of colluding malicious nodes. In some scenarios, especially when the network size is large, it is hard to determine the bound of normal delay, and this can be exploited by the adversary to launch denial of service attacks. Moreover, these schemes repel asynchronous interaction between the source and the destination/verifier [7]. The scheme cannot achieve non-repudiation, either. *However, our approach does not require time synchronization between* *nodes, allows immediate authentication, and is applicable in asynchronous communication scenarios.*

Perturbation-based Schemes for Key Establishment. Zhang and Subramanian et al. [8], [9] proposed perturbation number and perturbation polynomial based techniques for compromise-resilient key management in sensor networks. The techniques are extended and applied in this paper, but they are used for a different purpose of message authentication. Furthermore, in the message authentication scenario that we study, every sensor node can be both sender and receiver, which pose new challenges (for example, the possibility of reflection attack). This also makes the perturbation techniques devised in this paper distinguished from those in [8], [9].

V. Conclusion

We proposed a novel message authentication approach which adopts a perturbed polynomial-based technique to simultaneously accomplish the goals of lightweight, resilience to node compromises, immediate authentication, scalability, and non-repudiation.

Acknowledgements

This work is partially supported by NSF CNS-0716744 and CNS-0627354.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, March 2002.

[2] H.Chan and A. Perrig, "Security and Privacy in Sensor Networks," *IEEE Computer*, October 2003.

[3] C. K. Won and S. S. Lam, "Digital signatures for flows and multicasts," *IEEE ICNP*, 1999.

[4] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical En-route Filtering of Injected False Data in Sensor Networks," *IEEE Infocom'04*, March 2004.

[5] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An Interleaved Hop-by-Hop Authentication Scheme for Filtering False Data in Sensor Networks," *IEEE Symposium on Security and Privacy*, 2004.

[6] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," *IEEE Symposium on Security and Privacy*, May 2000.

[7] S. Bhattacharya, H. Kim, S. Prabh, and T. Abdelzaher, "Energy-Conserving Data Placement and Asynchronous Multicast in Wireless Sensor Networks," *MobiSys*, 2003.

[8] N. Subramanian, C. Yang, and W. Zhang, "Securing distributed data storage and retrieval in sensor networks," *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2007.

[9] W. Zhang, M. Tran, S. Zhu, and G. Cao, "A Random Perturbation-Based Pairwise Key Establishment Scheme for Sensor Networks," *ACM MobiHoc*, September 2007.

[10] Philip Levis, Nelson Lee, Matt Welsh, and David Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," in *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, 2003.

[11] D. Malan, M. Welsh, and M. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," *First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, October 2004.

[12] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "Tinypk: Securing sensor networks with public key technology," *ACM SASN'04*, October 2004.

[13] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," *Third IEEE International Conference on Pervasive Computing and Communication (PerCom 2005)*, March 2005.

[14] B. Gougard, F. Catthoor, D. Daly, A. Chandrakasan, and W. Dehaene, "Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: modeling and improvement perspectives," *IEEE DATE*, 2005.

[15] G. Gaubatz, J. Kaps, E. Ozturk and B. Sunar, "State of the art in ultra-low power public key cryptography for wireless sensor networks," 2005, pp. 146–150.

[16] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences," *Lecture Notes in Computer Science*, vol. 740, pp. 471–486, 1993.

Appendix A Proof of Theorem 3.6 (sketch): Without loss of generality, we assume $d_x \geq d_y$. We also assume the compromised nodes have sender IDs: $u_1, \cdots, u_n$ and receiver IDs: $v_1, \cdots, v_n$; that is, they know $auth_{u_i}(y,z)$ and $verf_{v_i}(x,z)$ for $i = 1, \cdots, n$.

For case (i), the conclusion is obvious according to [16].

For case (ii), the intruder may attack in two ways:

The first attack is only based on $verf_{v_i}(x,z) = f(x, v_i, z) + \overline{\alpha}(x) + r'_{v_i}$, where $i = 1, \cdots, n$, $\overline{\alpha}(x)$ could be either $\overline{\alpha}_0(x)$ or $\overline{\alpha}_1(x)$. Therefore, for an arbitrary pair of x_0 and z_0, similar to the attack shown in the proof of Theorem 3.2, the intruder may attempt to infer $f(x_0, y, z_0)$, and the complexity is $\Omega(2^{r*(d_y+1)})$.

The second attack is based on both compromised $verf_{v_i}(x,z)$'s and compromised $auth_{u_i}(y,z)$'s: Again let x_0 and z_0 be arbitrary elements of F_q, and the intruder wants to find out $f(x_0, y, z_0)$. Also, let $a_i = verf_{v_i}(u_1, w) - auth_{u_1}(v_i, w)$. Then, we have

$$\begin{aligned} a_i &= verf_{v_i}(u_1, z_0) - auth_{u_1}(v_i, z_0) \\ &= f(u_1, v_i, z_0) + r'_{v_i,0} * \overline{\alpha}(u_1) + r'_{v_i,1} \\ & \quad -[(f(u_1, v_i, z_0) + r_{u_1,0} * \overline{\beta}(v_i) + r_{u_1,1} \\ &= r'_{v_i,0} * \overline{\alpha}(u_1) + r'_{v_i,1} - [r_{u_1,0} * \overline{\beta}(v_i) + r_{u_1,1}]. \end{aligned}$$

This is equivalent to

$$r'_{v_i,1} = a_i + r_{u_1,0} * \overline{\beta}(v_i) + r_{u_1,1} - r'_{v_i,0} * \overline{\alpha}(u_1).$$

Let $f(x_0, y, z_0) = \sum_{j=0}^{d_y} C_j y^j$. Then, we have

$$\begin{aligned} & \sum_{j=0}^{d_y} C_j (v_i)^j = verf_{v_i}(x_0, z_0) - [r'_{v_i,0} * \overline{\alpha}(x_0) + r'_{v_i,1}] \\ \Rightarrow \quad & \sum_{j=0}^{d_y} C_j (v_i)^j = verf_{v_i}(x_0, z_0) - a_i \\ & +r'_{v_i,0} * [\overline{\alpha}(u_1) - \overline{\alpha}(x_0)] - r_{u_1,0} * \overline{\beta}(v_i) - r_{u_1,1} \end{aligned}$$

In this system of linear equations, unknowns include C_j ($j = 0, \cdots, d_y$), $r'_{v_i,0}$ ($i = 1, \cdots, n$), $\overline{\alpha}(u_1) - \overline{\alpha}(x_0)$, $\overline{\beta}_{k_i}(u_1)$, $r_{u_1,0} * \overline{\beta}(v_i)$ ($i = 1, \cdots, n$), and $r_{u_1,1}$. Each $r_{u_1,0} * \overline{\beta}(v_i)$ is an instance of the $r_{u_1,0} * \overline{\beta}(y)$. Hence, it can be expressed as $\sum_{j=0}^{d_y} B_j y^k$ with $(d_y + 1)$ unknowns. Therefore, the total number of unknowns becomes $2*d_y+n+4$. On the other hand, the total number of equations is n. Hence $2*d_y+4$ unknowns should be eliminated. Also note that the scope of each $r'_{v_i,0}$ is 2^γ, which is smaller than the scope of any other unknowns. Therefore, eliminating these variables is more efficient than eliminating others. The complexity to figure out correctly the values of these n independent unknowns is $\Omega(2^{n*\gamma})$. After that, the number of unknown can be reduced to $2*d_y+4$, and can be solved when $n \geq 2*d_y+4 > 2*(d_y+1)$. Therefore, the complexity of this attack is $\Omega(2^{2*\gamma*(d_y+1)})$.

Overall, the complexity is $\Omega(min\{2^{r*(d_y+1)}, 2^{2*\gamma*(d_y+1)}\})$. Further, dropping the assumption of $d_y \leq d_x$, we get the complexity of $\Omega(2^{min\{r,2*\gamma\}*min\{d_x+1,d_y+1\}})$.