Secure Network Coding in Unattended Wireless Sensor Networks

Faezeh Sadat Babamir
Department of Computer Science
Shahid Beheshti University
Tehran, Iran
f.babamir@mail.sbu.ac.ir

Fattaneh Bayat Babolghani
Department of Computer Science
Shahid Beheshti University
Tehran, Iran
f.bayat@mail.sbu.ac.ir

Kayhan Moharreri
Department of Computer Science and Engineering
Ohio State University
Columbus, USA
moharrer@cse.osu.edu

Abstract—Security issues in wireless sensor networks have been focused by extensive researches in recent years. Security concerns are particularly critical in disconnected or Unattended Wireless Sensors Networks (UWSNs). In this setting, the sink periodically collects sensed data and therefore the network will be left unattended most of the time. An adversary can take advantage of this behavior to modify or erase data. Thus, cryptographic techniques must be employed to ensure privacy and integrity of the information. In this paper, we consider network coding along with data sharing to provide confidentiality and integrity simultaneously. Moreover, every shared message will be signed and encrypted in efficient time to make the communication secure.

Keywords-component: unattended wireless sensor networks, authenticity, confidentiality, integrity, time complexity, modifies generalized laguerre functions, collocation method.

I. Introduction

In the past decades, wireless Sensor Networks (WSNs) attracted many researchers. A lot of them considered as important issues such as: routing, security, power awareness and data abstraction, but security is prior common assumption in most of these works. On the other hand, WSNs should collect small size and especially secure data in real-time manner. This is a crucial property as sensor nodes are small, low power with limited storage capacity. Therefore, classical algorithms may not be applicable, i.e. considering resource constrained sensors. These algorithms cannot guarantee the security of data. The aforementioned problem is even more critical in the new generation of WSNs referred to as Unattended or disconnected Wireless Sensor Networks (UWSNs) as sink periodically leaves and returns to the network.

The disconnected networks are established in critical or military environments. Hence, sink or collector is unable to gather data in real-time manner. Moreover, the network will be left unattended and will be periodically visited by the itinerant sink. This property provides some threats such as discovering and compromising sensor nodes by the adversary without detection of communication. The adversary, also, invisibly can perform to be intractable and unpredictable. A UWSN adversary may have different goals; some are curious and aim just to disclose data, while others aim to search data to replace them with forged message. The third type of adversary, known as the polluter, aims to inject invalid data to corrupt network called DoS attack or mislead the sink. In such setting, the main challenge is assurance about data integrity for long time.

In this research, we propose a scheme that encrypts shares and signs the generated data to provide confidentiality and integrity. We also leverage an efficient numerical solution for encryption; every sensor with unique identification encrypts shares, in which encryption is one-way without the knowledge of initial boundary conditions. Then a linear signing algorithm is applied to provide authentication and prevention of DoS attack. The signed generated data will be broadcasted to the neighbour sensors. Every neighbour uses network-encoding for received shares and homomorphic signs to remove previous signature and generate unique signature. This process decreases the size of total received shares.

Organization: Section II reviews the related work of UWSNs. Section III sketches our proposed algorithm including applied network coding, homomorphic and numerical encryption process. In section IV we have demonstrated our scheme is efficient. We have ended this paper with conclusion section.

II. Related Work

In UWSN setting, the adversary may have different goals. Reactive adversary is the adversary who starts compromising sensors after he identifies the target. To be more precise, such an adversary is inactive until it gets a signal that certain data must be erased, and then it wakes up and starts compromising up to $l$ sensors per round. This is unlike the proactive adversary who can compromise sensors before identifying the target i.e. he essentially starts compromising sensors at round 1, before receiving any information about the target sensor and the target data collection round. He would choose and compromise different sensors in a geographic area even
before such signal is received. This powerful adversary who usually referred to as mobile adversary can even roam around the network and change from one set of compromised nodes to another, making such attacks more difficult to detect and prevent.

To defend against reactive adversary, many papers have been proposed encryption based schemes. Encryption can be employed to hide the collected information as well as the identity of the sensors that collect it. If the key of compromised node is not available, the reactive adversary is unable to distinguish the specific piece of collected data. However, proactive adversary can restore the keys of the other earlier compromised nodes to memorize encrypted data. These keys help adversary to encrypt some forged data and place them with the target data. Therefore, encryption is not enough to defend proactive adversary.

The faults of these solutions are discussed by Di Pietro et al. in [1]. They proposed super-encryption and re-encryption techniques to defend against mobile adversary. But they did not take into account the cost of time, memory and energy consumption overhead. In addition, the proposed solution have limitations because of dependency on symmetric (shared key) encryption. Symmetric setting prevents sensors to use data aggregation techniques. Another solution is asymmetric based scheme. Although it is more resource consuming than symmetric solution, the sensors can decrypt the ciphertext and perform data aggregation, eliminating redundancy to minimize memory and communication overhead. Therefore, in this scheme, extra efficiency through data aggregation will be obtained by more energy and memory consuming. In general, data aggregation is more considered than energy and memory consumption, since 1 bit transmitted may require the power equivalent to execute 800-1000 instructions [2].

D. Ma et al. in [3] proposed 2 approaches: First, FssAgg-BLS (a kind of signature) as ideal cryptographic tools for achieving data integrity and authentication for UWSNs in presence of active adversary. However, this public key encryption setting imposes extreme computational overheads on the network entities, which are intolerable for UWSNs applications. Second approach, FssAgg-Mac is based on symmetric key encryption, hash chains and Message Authentication Codes (MACs) which requires full symmetric key distribution and does not allow to be public verifiable. This makes it impractical for large distributed UWSNs. Later D. Ma et al. developed FssAgg-AR and FssAgg-BM in [4, 5] that more computational and storage efficient than FssAgg-BLS. However, all these schemes are still not efficient enough for UWSNs and are effective only against the reactive adversary that is relatively weak and easily to overcome.

III. Proposed Scheme

Ren et al. [6] prove that in order to achieve perfect secrecy, data sharing between neighbours is a suitable way. Therefore, in our scheme, sensor node collects data $D$ and breaks it to equal shares $d_1, d_2, ..., d_n$. Using following process, the sensor sends signed encrypted $d_s$ to the neighbours:

**A. Share Generation, encoding, signing and broadcasting process**

After sensor $s_i$ collects data $D$, it proceeds following steps to achieve data integrity, confidentiality and also authenticity.

1) Shares $D$ into equal $d_1, d_2, ..., d_n$

2) Using our numerical encrypting method (refer to section E), the sensor encrypts every $d_i$ to $Y_i$.

3) Every $Y_i$ will be signed by sensor $s_j$ (known as $\delta$).

4) Lastly, sensor $s_i$ broadcasts every $\delta_i$ to the each neighbour.

Below we describe mathematically this algorithm. $Set_j$ is the set of all neighbours of sensor $j$.

**Alg. 1:** Collecting and sending data($D$, $set_j$)

\[
\begin{align*}
&\text{Shares } D \text{ into } d_1, d_2, ..., d_n \\
&\text{Encodes } d_i \text{ by } Y_i = f(d_i) \\
&\text{Sign } Y_i \text{ by } \delta_i = \text{Sig}(Y_i) \\
&\text{Obtain } pk_i = \{Y_i || \delta_i || t || TS || CNT\} \\
&\text{Broadcast every } pk_i \text{ to every neighbour sensor belonging to } Set_j
\end{align*}
\]

$t, TS, CNT$ will be defined in section B

B. $\beta$-bounded moving (adapted from [7])

Every signed $Y_i$ should disperse enough to defend against mobile adversary, $\beta$ is number of traversed hops up to now while CNT is number of hops which should be traversed. To determine $\beta$ value, DLE variable is defined to determine the location entropy of data $d_i$. This concept makes trade-off between number of hops and energy communication. Moreover, more $\beta$ consumes much energy communication but makes higher security against mobile adversary. DLE helps us to determine suitable value.

Finally, $pk_i = \{Y_i || \delta_i || t || TS || CNT\}$ is output of sensor $s_j$ to another neighbour. More precisely, $s_j$, $\delta_i$, $t$, CNT are encoding vector of data share, signature of $Y_i$, sequence order of $d_i$ and $CNT = \beta$ respectively. Also, $TS$ is the time stamp of producing time. We define a tuple $UID = [TS || t]$, that can uniquely identify a share.

C. Network Coding

In this paper, we use two kinds of sensors that were called source sensor and forwarder sensor; source sensor should collect data and broadcast, while forwarder sensor receives the data from other sensors and then transforms...
theores data packets into one packet and then forwards the resulting packet to the next hop. Furthermore, since communications consume more energy than computations, forwarding nodes encode received packets into one packet by using network coding solution. Clearly, network coding technique increases overall computation energy instead it significantly decreases communication consumption. Finally, the forwarding sensor signs the packet through homomorphic signature (refer to section D).

1) Basic setting

In this setting, we show the network with \(G=(V,E)\). Source nodes and forwarding nodes are \(\{s_1,s_2,\ldots,s_p\} \subset V\) and \(f=\{f_1,f_2,\ldots,f_l\} \subset V\) respectively. The inputs of forwarding nodes are \(Y_i,e \in \{1, n\}\) of \(pk\) and output packets are \(Z_1, j \in \{1, p\}\). Source nodes \(S, G \in \{1, p\}\) propagate packets \(pk\) to the forwarding nodes. Each forwarding sensor, after receiving \(Y_i\) of \(pk\) from \(n \) incoming channels, computes following linear combination \(Z_i\) to transmit it to the \(j\)-the channel. The linear combination formula is:

\[
Z_j = \sum_{i=1}^{n} (\alpha_i)Y_i
\]

In formula (1), \(\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)\) is encoding vector. The node randomly generates \(\alpha\) or \(\alpha\) is pre-deployed, (depend on static network topology). It is proven that random coefficient optimises network performance with high probability because of independency of network topology.

2) Random linear network coding algorithm

In proposed scheme, every forwarding node receives some \(Y_i \in \{1, n\}\) and encodes them via network coding as explained in equation (1). Finally, it sends one packet containing an encoded vector of size \(n\). For simplicity, we let pre-deployed encoding vector \((\alpha)\). Consider, Alg. 1, for encoding \(n\) packets. The final output is encoded vector of \(Z\).

\[
\begin{pmatrix}
\alpha_{a_1} & \alpha_{a_2} & \ldots & \alpha_{a_p} \\
\vdots & \vdots & \ldots & \vdots \\
\alpha_{a_1} & \alpha_{a_2} & \ldots & \alpha_{a_p} \\
\vdots & \vdots & \ldots & \vdots \\
\alpha_{a_1} & \alpha_{a_2} & \ldots & \alpha_{a_p}
\end{pmatrix}
\begin{pmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_p
\end{pmatrix}
= \\
\begin{pmatrix}
Z_1 \\
Z_2 \\
\vdots \\
Z_p
\end{pmatrix}
\]

Figure1. Encoder Matrix

Our scheme is able to reconstruct thoroughly the primary data from all received packets. Moreover, by using aforementioned equation \(Y_i\) will be recovered in polynomial time (adapted [7]). In section E, we will propose a new numerical method to easily encrypt shares with time efficiency. This innovative solution is considerable either for sink or forwarding nodes, i.e. our scheme either on the node side or origin sink side runs efficiently.

D. Applied linear Homomorphic signature over \(\mathbb{F}_2\)

In this paper, we utilize Boneh et al. scheme which is inspired by Gentry, Peikert and Vaikuntanathan [7] defined linearly over binary field [8]. This signature is a short vector \(s \in \mathbb{Z}_q^{m}\) in \(\mathbb{Z}_q^{m}\), i.e. \(s\) is in both \(\mathbb{Z}_q\) and \(\mathbb{Z}_q^{m}\). Mod 2 relates the signature to the message while mod \(q\) is designed to prove unforgeability of the scheme. This \(\Delta\) is different for signing every packet.

The source sensor signs every \(Y \) using its identity based private key and then sends \((Y, \delta)\) to the forwarding neighbour node. Forwarding node receives \(Y_s\) along with their signatures.

Firstly, it checks the validity of signature. If it is not valid, forwarding sensor removes it as bogus data. Receiving enough valid data, forwarding sensor encodes and generates a homomorphic signature from share signatures without knowing the original messages \((d)\) or the private key of source nodes. The detailed scheme is as follow:

1) Parameter setup phase

Following, we define parameters that used in [5] to describe applied signature. It is an \(m\)-dimensional lattice whose points are defined on \(\mathbb{Z}^m\). Also, \(\mathbb{Z}^m\) is a full-rank discrete subgroup of \(\mathbb{R}^m\) and consist of vectors are generated by orthogonal to a certain “parity check matrix” \(\Delta\in\mathcal{F}_2^{m \times n}\) modular integer \(q\). The utilized lattices are defined:

\[
\Lambda_q^\perp(\Delta) = \{eeZ^m : \Delta, e = 0 \mod q\}
\]

\[
\Lambda_q^u(\Delta) = \{eeZ^m : \Delta, e = u \mod q\}
\]

\[
\Lambda_q^\perp(\Delta) = \{eeZ^m : \exists s \in \mathbb{Z}_q^n \text{ with } \Lambda^\perp s = e \mod q\}
\]

In formula (3), \(\Lambda^\perp_q(\Delta)\) is a coset of lattice \(\Lambda_q(\Delta)\) of formula (2) such that \(\Lambda^\perp_q(\Delta) = \Lambda^\perp_q(\Delta) + t\) in which \(t\) holds in \(\Delta \cdot t = u \mod q\).

2) Signature scheme

Firstly, we describe following functions that used in the Boneh et al. scheme:

- \(\text{TrapGen}(q, n)\): this algorithm receives an integer \(q\) and \(n\) holds in \(m = \lceil 6n \log q \rceil\). Also this algorithm outputs \((\Delta \in \mathbb{Z}_q^{m \times m}, S \in \mathbb{Z}_q^{m \times m})\), where \(\Delta\) is statistically close to a uniform matrix in \(\mathbb{Z}_q^{m \times m}\) and \(S\) is a basis for \(\Lambda_q^\perp\).

- \(\text{ExtBasis}(S, B)\): let \(m'\) be an arbitrary dimension. This algorithm gets \((S, B = \Delta \| \Delta')\) where
\[ \Delta^t \mathbb{E}_{\mathbb{Z}_q^{n \times m}} \] and \[ S \mathbb{E}_{\mathbb{Z}^{m \times m}} \] be an arbitrary basis of \( \Lambda^t_1(\Delta) \) for a rank \( n \) matrix \( \Delta \mathbb{E}_{\mathbb{Z}_q^{n \times m}} \) that outputs a basis \( T \) of \( \Lambda^t_1(B) \subset \mathbb{Z}^{(m+m) \times (m+m)} \).

- SamplePre(\( \Delta, T, u, \delta \)): this algorithm inputs matrix \( \Delta \mathbb{E}_{\mathbb{Z}_q^{n \times m}} \), a basis \( T \) of \( \Lambda^t_1(\Delta) \), a parameter \( \delta \) and a vector \( u \in \mathbb{Z}^n \). Then outputs a sample which is statistically close to the distribution of \( \mathcal{D}_{\Lambda^u} \).

\[ \begin{align*}
\text{a) Signing algorithm} & \\
1. \text{Choose a } id \sim \{0,1\}^* \text{ randomly. If } id \text{ has already been queried to the hash function } H, \text{ then abort. (The simulation has failed).} \\
\text{Setup}(\pi; k): \text{On input of a security parameter } n \text{ and a maximum data set size } k, \text{ do the following:} & \\
1. \text{Choose two primes } p, q = \text{poly}(n) \text{ with } q \geq (kp)^2. \text{ Define } l := \lfloor n/\delta \log q \rfloor. & \\
2. \text{Set } A := pZ^n. & \\
3. \text{Use TrapGen}(a; l) \text{ to generate a matrix } A \in \mathbb{E}_{\mathbb{Z}_q^{n \times m}} \text{ along with a short basis } T_0 \text{ of } \Lambda^t_1(\Delta). \text{ Define } \Delta = \Lambda^t_1(\Delta) \text{ and } T := p.T_0. \text{ Note that } T \text{ is a basis of } A \cap A = p.A_2. & \\
4. \text{Set } v := p. \sqrt{n}. \log q. \log n. & \\
5. \text{Let } H: \{0,1\}^* \rightarrow F^l \text{ be a hash function (modeled as a random oracle).} & \\
6. \text{Output the public key } pkey = (A, A_2, v, k, H) \text{ and the secret key } skey = T. & \\
\text{The public key } pkey \text{ defines the following system parameters:} & \\
- \text{The message space is } F^n_p \text{ and signatures are short vectors in } Z^l. & \\
- \text{The set of admissible functions } F \text{ is all } F^l_\rho \text{ linear functions on } k \text{-tuples of messages in } F^n_p. & \\
- \text{For a function } f \in F \text{ defined by } f(m_1, \ldots, m_k) = \sum_{i=1}^k c_i m_i, \text{ we encode } f \text{ by interpreting the } c_i \text{ as integers in } (-p/2, p/2). & \\
\text{Sign}(skey, t, m, i): \text{On input of a secret key } skey, \text{ a tag } \tau \in \{0,1\}^*, \text{ a packet key } pkey \mathbb{E}_{\mathbb{Z}_q^{n \times m}} \text{ and an index } i, \text{ do:} & \\
1. \text{Compute } a_1 = H(\tau||i) \in F^l. & \\
2. \text{Compute } t \in Z^n \text{ such that } t \text{ mod } p = pkey \text{ and } \Delta \text{ mod } q = a_1. & \\
3. \text{Output } \sigma \leftarrow \text{SamplePre}(A_1, A_2, T, t, v) \in (A_1, A_2) + t. & \\
Verify(pkey, \tau, pk, \sigma, f): \text{On input of a public key } pkey, \text{ a tag } \tau \in \{0,1\}^*, \text{ a message } m \in F^n_p, \text{ a signature } \sigma \in Z^n \text{ and a function } f \in F, \text{ do:} & \\
1. \text{If all of the following conditions hold, output 1 (accept); otherwise output 0 (reject):} & \\
(a) ||\sigma|| \leq k.2^{-q}.2^{\sqrt{n}} & \\
(b) \sigma \text{ mod } p = pkey. & \\
2. \text{Evaluate } (pkey, \tau, f, \sigma). \text{ On input a public key } pkey, \text{ a tag } \tau \in \{0,1\}^*, \text{ a function } f \in F \text{ encoded as }<f> := (c_1, \ldots, c_k) \in Z^k \text{ and a tuple of signatures } (\sigma_1, \ldots, \sigma_k) \in Z^k, \text{ output } \sigma = \sum_{i=1}^k c_i \sigma_i. & \\
\text{After sink receives all signed encrypted shares, it verifies the homographic signature and decrypts them to reconstruct } D. & \\
\text{In this signing algorithm, we apply linear signing and efficient encoding algorithms. More exactly, we firstly encrypt } d_i \text{ into } Y, \text{ included in } pkey = f_Y||\delta||\tau||TS||CNT / \text{ by proposed numerical method. This encrypting solution prevents adversary to read data because our numerical encrypting solution Eq. (1) is a differential equation and insolvable without knowing boundary conditions. Boundary conditions are initial values of the Eq. (2) which is available for receivers. We discuss about our numerical technique in following section.} & \\
\text{E. Numerical encryption process} & \\
\text{In this section, we have an improvement in our last scheme that was mentioned in [38]. In fact, we used an Ordinary Differential Equation (which we denote as ODE) that has an exact solution for encrypting the data. This ODE can only be solved with its boundary conditions and we use that as the decryption part on the receivers’ side; meanwhile senders have the exact solution that would be mentioned in Eq. (3) for encryption. The ODE would be well known, and the boundary conditions would be the secret key of receivers for decryption.} & \\
\text{The receivers decrypt and solve this equation by modified generalized Laguerre functions which are orthogonal functions. Collocation method is used in this approach that reduces the solution of this problem to the solution of an algebraic equation. Moreover, in the graph of the } ||Res||^2, \text{ we show that the present solution is more accurate and faster in convergence for this problem.} & \\
\text{Firstly, we introduce the well known equation from the problem of flow and diffusion of chemically reactive species over a nonlinearly stretching sheet. This equation is well known by the adversary and receivers. This non-linear ODE is [9-24, 35]:} & \\
\frac{d^2f}{dd_i} + f \frac{d^2f}{dd_i^2} + (\frac{df}{dd_i})^2 - id \frac{df}{dd_i} = 0 \quad (1) & \\
\text{Subject to boundary conditions,} & \\
f(0) = 0, \quad f’(0) = 1, \quad f’(d_i) = 0 \text{ as } d_i \rightarrow \infty, \quad (2) & \\
\text{Thses boundary conditions are kept secret on receivers.} & \\
\text{In the Eq. (1), } id \text{ is identification of every sender sensor, and in Alg. 1, data shares into } d_1, d_2, \ldots, d_n \text{ that every } d_i \text{ is entry of Eq. (1).} & \\
\text{Here we note that the Eq.(1) subject to boundary conditions Eq.(2) has an exact solution in [11,17] that is used for encrypting the data:}
\[
f(d_i) = \frac{1}{\sqrt{1 + d_i}} \left(1 - e^{-\sqrt{1 + d_i}}\right)
\]  

2) Function approximation with Laguerre functions

A function \( f(x) \) defined over the interval \( I = [0, \infty) \) can be expanded as:

\[
f(x) = \sum_{n=0}^{\infty} a_n \phi_n(x),
\]

where

\[
a_i = \frac{\langle f(x) \phi_i(x) \rangle}{\langle \phi_i(x) \phi_i(x) \rangle}
\]

If the infinite series in Eq. (7) is truncated with \( N \) terms, then it can be written as [25]:

\[
f(x) \approx \sum_{n=0}^{N-1} a_n \phi_n(x) = A^T \phi(x),
\]

with

\[
A = [a_0, a_1, a_2, \ldots, a_{N-1}]^T,
\]

\[
\phi(x) = [\phi_0(x), \phi_1(x), \ldots, \phi_{N-1}(x)]^T.
\]

3) Modified generalized Laguerre functions collocation method

Let \( w(x) = \frac{x}{l} \) and \( x_j, j = 0, 1, \ldots, N - 1 \), be the \( N \) MGLF-Radau points that are the collocation points. The relation between MGLF orthogonal systems and MGLF integrations is as follows [25, 29-32]:

\[
f_0^{+\infty} f(x)w(x)dx = \sum_{j=0}^{N-1} f_j(x)w_j + \frac{\Gamma(N+2)}{(N+2)!} f^{2N}(\xi)e^\xi,
\]

where \( 0 < \xi < \infty \) and

\[
w_j = x_j \left( \frac{L(N+1)!}{(N+1)!} \right)^{1/2} \phi_{N+1}(x_j)^2,
\]

\[j = 0, 1, 2, \ldots, N - 1.\]

In particular, the second term on the right-hand side vanishes when \( f(x) \) is a polynomial of degree at most \( 2N - 1 \) [25]. We define:

\[
u(x) = \sum_{j=0}^{N-1} a_j \phi_j(x),
\]

4) Solving the problem with modified generalized Laguerre functions

To apply modified generalized Laguerre collocation method to Eq. (1) with boundary conditions Eq. (2), at first we expand \( f(d_i) \) as follows:

\[
f(d_i) = \sum_{j=0}^{N-1} a_j \phi_j(d_i)
\]

To find the unknown coefficients \( a_j \)'s, we substitute the truncated series \( f(d_i) \) into Eq. (1) and boundary conditions in Eq. (2). Also, we define Residual function of the form:

\[
\text{Res}(d_i) = \sum_{j=0}^{N-1} a_j \phi_j''(d_i) + \sum_{j=0}^{N-1} a_j \phi_j(d_i) \sum_{j=0}^{N-1} a_j \phi_j'(d_i) - \left( \sum_{j=0}^{N-1} a_j \phi_j(d_i) \right)^2 - \mu d_i \sum_{j=0}^{N-1} a_j \phi_j(d_i)
\]

\[
\sum_{j=0}^{N-1} a_j \phi_j(0) = 0,
\]

\[
\sum_{j=0}^{N-1} a_j \phi_j(l) = 0.
\]

Encryption:

Assuming that every \( d_i \) is \( l \) bits (also known as share length), then senders use a pseudorandom generator with the expansion factor of \( l \) to mask the predictability of \( d_i \). For every \( d_i \) we will generate a random number of the length \( s \), where \( s << l \) for the preformance efficiency. Technically, that is defined \( G : \{0, 1\}^s \rightarrow \{0, 1\}^l \); also, \( r \in \{0, 1\}^s \) is chosen randomly.

For every \( d_i \), the sensors computes ciphertext as follows:

\[
Y_i = f(G(r) \oplus d_i) \| r
\]

Where \( f \) is only known to sender sensors, and it will be unpredictable for the adversary to guess \( d_i \)'s from the ciphertexts as they are randomly being masked by our pseudorandom generator. As a result of this encryption we can also apply Cipher Block Chaining (CBC) mode to this scheme where \( f(G(r) \oplus d_j) \) will feed the next mask for \( d_i \)'s instead of generating a new \( G(r) \). (assuming that the range of the function \( f \) can be adapted to size \( s \)). However, in practice, we found that CBC is biased towards the output of the function as \( f \) being used here is clearly not a pseudorandom function.

Decryption:

We introduce the method encrypting and solving this equation as follows. Also, different techniques have been used to obtain analytical and numerical solutions for this problem in [14, 22-24].

1) Modified Generalized Laguerre functions

The generalized Laguerre in polynomial manner are defined with the following recurrence formula [25-28]:

\[
L_n^\alpha(x) = 1, \quad L_n^\alpha(1) = 1 + \alpha - x,
\]

\[
nL_n^\alpha(x) = (2n - 1 + \alpha - x)L_n^\alpha(x) - (n + \alpha - 1)L_{n-2}^\alpha(x),
\]

These are orthogonal polynomials for the weight function \( w_\alpha(x) = e^{-x}x^\alpha \). We define Modified Generalized Laguerre Functions (which we denote MGLF) \( \phi_j \) as follows [25]:

\[
\phi_j(x) = \exp \left( \frac{x^2}{2l} \right) L_j^\alpha \left( \frac{x}{l} \right), \quad L > 0.
\]

This system is an orthogonal basis [36, 37] with weight function \( w(x) = \frac{x}{l} \) and orthogonality property [25]:

\[
(\phi_m, \phi_n / w_\alpha) = \frac{\Gamma(n+2)}{\Gamma(n)} \delta_{mn},
\]

where \( \delta_{nm} \) is the Kronecker function.
\[ \sum_{j=0}^{N-1} a_j \phi_j(0) = 1, \quad (17) \]
\[ \sum_{j=0}^{N-1} a_j \phi_j(\alpha) = 0. \quad (18) \]

By applying \( d_i \) in Eq. (15) with the \( N \) collocation points which are roots of functions \( L^n_k \), we have \( N \) equations that generate a set of \( N \) non-linear equations; also, we have two boundary equations in Eq. (16,17). Now, all of these equations can be solved by Newton method for the unknown coefficients. We must mention Eq. (18) is always true; therefore, we do not need to apply this boundary condition.

All in all, we can find the polynomial approximation function of \( f(d_i) \); therefore, every \( Y_i \) can be decrypted on the recivers.

The absolute error between MGLFMs solution and exact solution of the velocity profile \( f(d_i) \) for \( id = 0.6 \) is shown in Figure 2. This graph shows the error is negligible; Also, we know that every \( d_i \) is corresponding to an integer value; therefore, the result of decrypting must be rounded to the nearest integer value. In addition, the graph of the \( \| \text{Res}\|_2^2 \) for MGLF at \( id = 0.6 \) \( b_2 = 0.1 \) is shown in Figure 3. This graph illustrates the convergence rate of the method.

**IV. Performance Analysis**

This approach proposed above is secure against chosen-plaintext attacks (CPA-secure) because it uses random masking mechanism. So the same message encrypts to different values each time making it hard for the adversary to guess the real data flow underneath the masking. Also, we sign every encrypted shares that makes this scheme secure against chosen-ciphertext attacks (CCA-secure).

Besides, decryption is based on the modified generalized Laguerre. Modified generalized Laguerre functions are orthogonal functions that solved the system of non-linear differential equations governing the problem on the semi-infinite domain without truncating it to a finite domain. Modified generalized Laguerre functions were proposed to provide simple way to improve the convergence of the solution through collocation method by \( N = 20 \), \( \alpha = 1 \) and \( L = 0.99 \). Figure 2 and Figure 3 show this approach is more accurate and faster convergence in this problem.

**V. Conclusion**

In this paper, we proposed an efficient scheme including special technique to defend against curious, and search-replace adversary as well as injection capable attacker. In fact, we shared and encrypted data using a numerical method (defence against curious, and search and replace adversary), and efficiently signed every unit of data to prevent injection attacks.

Moreover, based on an ODE and its boundary conditions, a new function for every sensor is released because every sensor has its special \( id \). This equation is publically known but the calculated function is infeasible to obtain without knowing boundary condition. Hence, encrypted packet of any function provides no information about the original data. This technique is also firm against injection attack which is the most rampant attack in general unattended wireless sensor network. Furthermore, we can claim that, our model is applicable and scalable to real world applications and it is secure against statistical traffic analysis attacks since it blocks chosen-plaintext attacks and chosen-ciphertext attacks.

**References**


Szeg, G.: Orthogonal polynomials. AMS, New York, (1939)


