10

Linearly Time Efficiency in Unattended Wireless Sensor Networks

Faezeh Sadat Babamir and Fattaneh Bayat Babolghani
Shahid Beheshti University of Tehran
Iran

1. Introduction

In the past decades, wireless Sensor Networks (WSNs) attracted many researchers. A lot of them considered important issues such as: routing, security, power awareness and data abstraction, But security is prior common assumption in the most of works. On the other hand, WSNs should collect small size and especially secure data in real-time manner. This problem is considered because sensor nodes are small, low power with low storage. Therefore, classical algorithms maybe inapplicable, i.e. considering constrained sensor, these algorithms cannot guarantee the security of data. The aforementioned problem is very critical in the new generation of WSNs referred to as Unattended or disconnected wireless sensor networks.

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 leaved unattended and will be periodically visited. This property provides some threats such as discovering and compromising sensor nodes by adversary without detection. Moreover, adversary invisibly performs to be intractable and unpredictable. Also, some adversary is curious and aims just to disclose data, while some aims search data to replace them with forged. The third kind of network adversary whiles to inject invalid data to corrupt network called DoS attack or mislead sink. In such setting, the main challenge is assurance about data survival for long time.

In this research, we propose scheme that firstly shares generated data and encodes them to provide confidentiality and integrity. Moreover, utilizing efficient mathematical solution, every sensor with unique identification encodes shares, in which encoding process is one-way with initial boundary conditions. Then a linear signing algorithm applies to provide authentication and prevent DoS attack. In addition, in order to defend curious adversary, 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 decrease size of total received shares.

Organization: Section 2 reviews the related work of UWSNs. Section 3 sketches our proposed algorithm including applied network coding, homomorphic and mathematical solution. In section 4 we have demonstrated our scheme efficiency implemented by Maple. We have ended this chapter with conclusion section.

2. Related works

In this setting, the adversary may have different goals. Reactive adversary is the adversary who starts compromising sensors after he identifies the target. More exactly, such an adversary is inactive until it gets a signal that certain data must be erased, then it wakes up and starts compromising up to l sensors per round 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 delete and prevent.

Di Pietro et al. in [1] investigated the data survival for the first time. They proposed a straight-forward non-cryptographic technique to hide the sensed data from the adversary. In [1], the adversary was actively hunting data and was not afraid to delete/erase any data he found. They claimed that they could achieve surprising degree of data survival with respect to the time between successive sink visits but they considered small number of compromised sensors including k=2, 3, 5, 10 which make it non-realistic. So when l increases, the benefits of replication attack are magnified. Observing that the simple technique has certain basic limitations, they proposed a more advanced approach based on standard cryptographic tools. They discussed the effects of encryption and claimed that regardless of the encryption type, the adversary has equally diminished capacity to detect and erase target data as it inspects the memory of compromised nodes.

To defend 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 but 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.

Mateus et al. in [2] evaluated proposed cryptographic based schemes on a real sensor platform. They measured some basic operation usage and presented results for encryption, super-encryption and key evolution which are feasible for protecting UWSNs against mobile adversary. Encryption is the central tool in the design of any symmetric scheme and is usually implemented by means of a block-cipher. Therefore, it becomes necessary to choose a suitable block-cipher for the development of secure and efficient schemes for super encryption.

Finally they calculated that if super-encryption is applied many rounds by different nodes, an adversary would have to make a great effort in order to find and destroy the targeted data. However the number of rounds and the payload size in super-encryption have significant impact over the performance of this technique. These disadvantages presented in figure 1 and table 1 in terms of time and energy consumption.

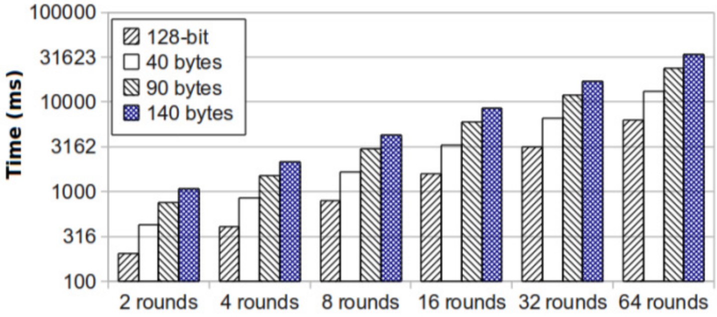

Fig. 1. Time costs for super-encryption (100 executions)

	Energy (mJ)
2 rounds	1.09 ± 0.04
4 rounds	2.21 ± 0.01
8rounds	4.45 ± 0.02

Table 1. Super-encryption energy consumption (100 executions)

[2] In order to implement and evaluate some key operations for re-encryption process, the code of the MIRACL library [3] is adapted. They measure inversion and exponentiation operations through the polynomial arithmetic which depend on the field chosen. The algorithm used for inversion is a polynomial version of the Extended Euclidian algorithm from Lim and Hwang [4]. They have chosen a general algorithm for exponentiation. Although the symmetric algorithms are not expensive, the re-encryption strategy is still the main alternative against proactive adversaries. Moreover, according to [2], the Elliptic Curve Cryptography (ECC) schemes show an important drawback of the re-encryption solutions,since the exponentiation is not as suitable as polynomial operations. Hence Public Key Cryptography (PKC) should be considered.

3. Proposed scheme

Ren et al. [5] prove that in order to achieve perfect security, data sharing between neighbours is suitable way. Therefore, in our scheme, sensor node collects data *data* and breaks it to equal shares d_1, d_2..., d_n. Using following process, the sensor sends signed encoded Y_i to the randomly selected neighbours.

3.1 Share generation, encoding, signing and broadcasting processes

After sensor v_i collects data *data*, it proceeds following steps to achieve data integrity, confidentiality and also authenticity.

1. Shares *data* into equal $d_1,d_2,\ldots,d_n$.
2. Using our mathematical encoding solution (refer to section 3.5), the sensor encodes every d_i to Y_i.
3. Every Y_i will be signed by sensor $v_i(\delta_i)$.
4. Lastly, sensor v_i broadcasts every δ_i to the each neighbour.

Below we describe mathematically this algorithm. *Set$_i$* is the set of all neighbours of sensor *i*.

```
Alg. 1: Collecting and sending data(data, set_i)
        {
                Shares data into $d_1,d_2,..,d_n$.
        Encode $d_i$ by $Y_i=f(d_i)$.
                Sign $e_i$ by $\delta_i=Sig(Y_i)$
                Obtain $pk_i=\{Y_i||\delta_i||t||TS||CNT\}$
        Broadcast every $pk_i$ to every neighbour sensor belong to $set_i$.
                }
```

3.2 β-bounded moving(adapted [5])

Every signedY_i should disperse enough to defend against mobile adversary. To determine β value, *DLE* variable is defined to determine the entropy of data d_i location entropy. This concept makes trade-off between hops steps and energy communication. Moreover, more β 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 v_i to another neighbour, e.g. v_j in which Y_i, δ_i, t, CNT are encoding vector of data share, signature of Y_i, sequence order of d_t and $CNT=\beta$ respectively. Also, *TS* is time stamp of producing time. We define tuple$UID=\{TS||t\}$, that can uniquely identify a share.

3.3 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 them, while forwarder sensor receives the data packets from other sensors and then transforms theses data packets into one packet; Moreover, since communications consume more energy than computation, forwarding nodes probability encode received packets into one 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 3.4).

3.3.1 Basic setting

In this setting, we show the network with $G=(V,E)$. Source nodes and forwarding nodes are$S=\{s_1,s_2,\dots,s_N\}\subseteq V$ and $f=\{f_1,f_2,\dots,f_L\}\subseteq V$ respectively. The inputs of forwarding nodes are $Y_i, i\epsilon[1,p]$ of pk_i and output packets are Z_j, $j\epsilon[1,q]$. Source nodes (s_i) propagate packets pk_i to the forwarding nodes. Each forwarding sensor, after receiving Y_i of pk_i from p incoming channels, computes following linear combination Y_j to transmit it to the j-the channel. The linear combination formula is:

$$Z_j = \sum_{i=1}^{p}(\alpha_i)(Y_i) \tag{1}$$

In formula (1), $\alpha = (\alpha_1, \alpha_2,\dots,\alpha_p)$ is encoding vector. The node randomly generatesα or α 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.

3.3.2 Random linear network coding algorithm

In proposed scheme, every forwarding node receives some Y_is $i\epsilon[1,p]$and encodes them via network coding with probability p_{nc}. Finally, it sends one packet contained p encoded vectors. For simplicity, we let pre-deployed encoding vector (α). Consider, Alg. 1, for encoding p packets. The final outputs are encoded vectors Z_j and the same inputs.

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1p} \\ \vdots & \cdots & \cdots & \vdots \\ \alpha_{i1} & \alpha_{i2} & \cdots & \alpha_{ip} \\ \vdots & \cdots & \cdots & \vdots \\ \alpha_{j1} & \alpha_{j2} & \cdots & \alpha_{jp} \end{pmatrix} \begin{pmatrix} Y_1 \\ \vdots \\ Y_i \\ \vdots \\ Y_p \end{pmatrix} = \begin{pmatrix} Z_1 \\ Y_2 \\ \vdots \\ Y_{p-1} \\ Z_j \end{pmatrix}$$

Fig. 2. Encoder Matrix

Our scheme is able to reconstruct thoroughly the primary data from all received packets. Moreover, by using aforementioned equation $data_i$ will be recovered in polynomial time (adapted [5]). In section 3.5, we propose a new algebraic algorithm to easily encoding shares with *time efficiency*. This innovation solution is considerable either for sink or forwarding nodes, i.e. our scheme either in node side or in sink side is efficiently ran.

3.4 Applied linear homomorphic signature over $\mathcal{F}_2$

In this paper, we utilize Boneh et al. scheme which is inspired by Gentry, Peikert and Vaikuntanathan [6] defined linearly over binary field [7]. This signature is a short vector $\delta\epsilon Z^m$in $\Lambda_{2q}^{q.v}(\Delta)$, i.e. $\delta$is in both $\Lambda_q^{\perp}(\Delta)$ and $\Lambda_2^{v}(\Delta)$ simoltaneusly. Mod 2 relates the signature to the message while mod q is designed to prove unforgeability of the scheme. This Δ is different for signing every packet.

The source sensor signs every Y_i using its identity based private key and then sends (Y_i, δ_i) to the forwarding neighbour node. Forwarding node receives Y_is 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 re-encodes them to the e' and generates a homomorphic signature from share signatures without knowing the original messages (d_i) or the private key of source nodes. The detail of scheme is as follow:

3.4.1 Parameter setup phase

Following, we define parameters that used in [7] to describe applied signature. Λ is an m-dimensional lattice whose points are defined on $\mathbb{Z}^m$. Also, Λ is a full-rank discrete subgroup of $\mathbb{R}^m$ and consist of vectors either generated by or orthogonal to a certain "parity check matrix" $\Delta\ \epsilon\ \mathbb{Z}_q^{m\times n}$ modular integer q. The utilized lattices are defined:

$$\Lambda_q^{\perp}(\Delta) = \{e\epsilon\mathbb{Z}^m\colon \Delta.e = 0\ mod\ q\} \tag{2}$$

$$\Lambda_q^{u}(\Delta) = \{e\epsilon\mathbb{Z}^m\colon \Delta.e = u\ mod\ q\} \tag{3}$$

$$\Lambda_q(\Delta) = \{e\epsilon\mathbb{Z}^m\ \colon\ \exists s\epsilon\mathbb{Z}_q^n\ with\ \Delta^t.s = e\ mod\ q\}$$

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

3.4.2 Signature scheme

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

TrapGen(*q, n*): this algorithm receives an integer q and n holds in $m = [6nlgq]$. Also this algorithm outputs $(\Delta\epsilon\mathbb{Z}_q^{n\times m}, S\epsilon\mathbb{Z}^{m\times m})$, where Δ is statistically close to a uniform matrix in $\mathbb{Z}_q^{n\times m}$ and S is a basis for $\Lambda_q^\perp$.

ExtBasis(*S, B*): let m' be an arbitrary dimension. This algorithm gets $(S, B = \Delta\|\ \Delta')$ where $\Delta'\epsilon\mathbb{Z}_q^{n\times m'}$and $S\epsilon\mathbb{Z}^{m\times m}$be an arbitrary basis of $\Lambda^\perp(\Delta)$ for a rank n matrix $\Delta\epsilon\mathbb{Z}_q^{n\times m}$that outputs a basis T of $\Lambda^\perp(B) \subset \mathbb{Z}^{(m+m')\times(m+m')}$.

SamplePre(*Δ, T, u, δ*): this algorithm inputs matrix $\Delta\epsilon\mathbb{Z}_q^{n\times m}$, a basis T of $\Lambda_q^\perp(\Delta)$, a parameter δ and a vector $u\epsilon\mathbb{Z}^n$. Then outputs a sample which is statistically close to the distribution of $\mathfrak{D}_{\Lambda_q^u,\delta}$.

- Signing algorithm

1- Choose a $id \overset{R}{\leftarrow} \{0,1\}^n$ randomly. If *id* has already been queried to the hash function *H*, then abort. (The simulation has failed).

a. Setup(*1^n; k*)

On input of a security parameter *n* and a maximum data set size *k*, do the following:

1. Choose two primes p, q = poly(n)with q ≥ $(nkp)^2$. Define l :=[n/6 log q].
2. Set $\Lambda_1 := pZ^n$.
3. Use TrapGen(*q;l; n*) to generate a matrix $\Delta\epsilon F_q^{l\times n}$along with a short basis T_q of $\Lambda_\mathrm{q}^\perp(\Delta)$. Define$\Lambda_2 = \Lambda_\mathrm{q}^\perp(\Delta)$and $T := p.T_q$. Note that T is a basis of $\Lambda_1 \cap \Lambda_2 = p\Lambda_2$.
4. Set $v := p.\sqrt{n.logq}.logn$.
5. Let $H: \{0,1\}^* \to F_q^l$be a hash function (modeled as a random oracle).
6. Output the public key $pkey = (\Lambda_1, \Lambda_2, v, k, H)$and the secret key *skey*= *T*.

The public key *pkey* defines the following system parameters:

- The message space is $F_p^n$and signatures are short vectors in Z^n.
- The set of admissible functions *F* is all F_p-linear functions on *k*-tuples of messages inF_p^n.
- For a function $f\epsilon F$defined by $f(m_1, \ldots, m_k) = \sum_{i=1}^k c_i m_i$, we encode *f* by interpreting the c_i as integers in (-p/2, p/2].

b. Sign($\boldsymbol{skey}, \boldsymbol{\tau}, \boldsymbol{m}, \boldsymbol{i}$)

On input of a secret key *skey*, a tag$\tau\epsilon\{0,1\}^n$, a packet$pkey\epsilon F_p^n$and an index *i*, do:

1. Compute $\alpha_i = H(\tau||i) \in F_q^l$.
2. Compute $t \in Z^n$such that *t*mod *p* = *pkey*and $\Delta. t$ mod q $= \alpha_i$.
3. Output $\sigma \leftarrow \mathrm{SamplePre}(\Lambda_1\Lambda_2, \mathrm{T}, \mathrm{t}, \mathrm{v}) \in (\Lambda_1\Lambda_2) + \mathrm{t}$.

c. Verify($\boldsymbol{pk}, \boldsymbol{\tau}, \boldsymbol{pk}, \boldsymbol{\sigma}, \boldsymbol{f}$)

On input of a public key *pkey*, a tag$\tau \in \{0,1\}^*$,a message $m \in F_p^n$, a signature$\sigma \in Z^n$and a function$f \in F$, do:

1. If all of the following conditions hold, output 1 (accept); otherwise output 0 (reject):

a. $||\sigma|| \leq k.\frac{p}{2}.v\sqrt{n}$

b. σ mod p = pkey.

Evaluate($pkey, \tau, f, \vec{\sigma}$). On input a public key *pkey*, a tag $\tau \in \{0,1\}^*$, a function $f \in F$ encoded as$< \mathrm{f} >= (\mathrm{c}_1, \ldots, \mathrm{c}_\mathrm{k}) \in \mathrm{Z}^\mathrm{k}$and a tuple of signatures$(\sigma_1, \ldots, \sigma_\mathrm{k}) \in \mathrm{Z}^\mathrm{k}$, output $\sigma = \sum_{i=1}^{k} c_i \, \sigma_i$.

After sink receives all signed encoded shares, it verifies the homographic signature and decodes them to reconstruct *data*.

In this signing algorithm, we apply linear signing and efficient encoding algorithms. More exactly, we firstly encode d_i into Y_i included in pk_i= $\{Y_i || \delta_i || t || TS || CNT \}$by proposed mathematic function. This encoding solution prevents adversary to read data because our mathematical encoding solution (equation 4) is a differential equation and insolvable without knowing boundary conditions. Boundary conditions are initial values of the equation 4 which is available for either sender or receiver. We discuss about our mathematical technique in following section.

3.5 Mathematical encoding solution

In this section, we used the Ordinary Differential Equation(denoted as ODE) for encoding the data shares. This ODE is solvable (or received data is decodable) just with presence of boundry conditions. Moreover, we solve this equation by modified generalized Laguerre which is orthogonal function. The utilization of colocation method reduces the solution of our problem to the solution of algebratic equation. Applying our technique, we show that the encoding process is time efficient, more accurate and converges faster.

We basicallywork on an equation of flow and diffusion of chemical reactive species over a nonlinearly stretching sheet problem. this non-linear ordinary differential equation is [8-23, 35]:

$$f'''(d_i) + f(d_i)f''(d_i) - \left(f'(d_i)\right)^2 - idf'(d_i) = 0 \tag{4}$$

Subject to boundary conditions,

$$f(0) = 0, f'(0) = 1, f'(\infty) = 0, \tag{5}$$

Where id is identification of every sensor and $d_1, d_2, .., d_n$ are data shares. Considering unique *id* for every sensor, the equation 4 releases new equation which is unique for every sensor. Moreover, all of equation replaced in sensors, are different as well as the whole equations are hard to invert. These issues gurantee the security against curious adversary.

Different techniques have been used to obtain analytical and numerical solutions for this problem. Raptis and Perdikis [13] used the shooting method for this problem. Kechil and

Hashim [15] obtained approximate analytical solution via Adomian decomposition method. Recently, in [16] and [14] the homotopy analysis method was also applied for solving the above equation [14].

3.5.1 Modified generalized Laguerre functions

This section is devoted to the introduction of the basic notions and working tools concerning orthogonal modified generalized Laguerre. It has been widely used for numerical solutions of differential equations on infinite intervals. $L_n^\alpha(x)$

$L_n^\alpha(x)$ (generalized Laguerre polynomial) is the n-th eigenfunction of the Sturm-Liouville problem [24-27]:

$$x\frac{d^2}{dx^2}L_n^\alpha(x) + (\alpha + 1 - x)\frac{d}{dx}L_n^\alpha(x) + nL_n^\alpha(x) = 0,$$

$$x \in I = [0, \infty), n = 0, 1, 2, ... \tag{6}$$

The generalized Laguerre in polynomial manner are defined with the following recurrence formula:

$$L_0^\alpha(x) = 1, \tag{7}$$

$$L_1^\alpha(x) = 1 + \alpha - x,$$

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

These are orthogonal polynomials for the weight function $w_\alpha = x^\alpha e^{-x}$. We define Modified generalized Laguerre functions (which we denote MGLF) ϕ_j as follows [24]:

$$\phi_j(x) = \exp\left(\frac{-x}{2L}\right)L_j^1\left(\frac{x}{L}\right), L > 0. \tag{8}$$

This system is an orthogonal basis [35, 36] with weight function $w(x) = \frac{x}{L}$and orthogonality property [24]:

$$\langle\phi_m, \phi_n\rangle_{w_L} = \left(\frac{\Gamma(n+2)}{L^2 n!}\right)\delta_{nm}, \tag{9}$$

where δ_{nm} is the Kronecker function.

3.5.2 Function approximation with Laguerre functions

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

$$f(x) = \sum_{i=0}^{+\infty} a_i\phi_i(x), \tag{10}$$

Where

$$a_i = \frac{\langle f, \phi_i\rangle_w}{\langle\phi_i, \phi_i\rangle_w}. \tag{11}$$

If the infinite series in Eq. (10) is truncated with N terms, then it can be written as [24].

$$f(x) \simeq \sum_{i=0}^{N-1} a_i \phi_i(x) = A^T \phi(x), \tag{12}$$

with

$$A = [a_0, a_1, a_2, \dots, a_{N-1}]^T, \tag{13}$$

$$\phi(x) = [\phi_0(x), \phi_1(x), \dots, \phi_{N-1}(x)]^T. \tag{14}$$

3.5.3 Modified generalized Laguerre functions collocation method

Laguerre-Gauss-Radau points and generalized Laguerre-Gauss-type interpolation were introduced by [24, 28-30].

Let:

$$\Re_N = Span\{1, x, \dots\dots, x^{2N-1}\} \tag{15}$$

we choose the collocation points relative to the zeroes of the functions [24].

$$p_j(x) = \phi_j(x) - (\tfrac{j+1}{j})\phi_{j-1}(x). \tag{16}$$

Let$w(x) = \frac{x}{L}$ and $x_j, j = 0, 1, \dots, N-1$, be the N MGLF-Radau points. The relation between MGLF orthogonal systems and MGLF integrations is as follows [24, 31]:

$$\int_0^{+\infty} f(x)w(x)dx = \sum_{j=0}^{N-1} f_j(x)w_j + \left(\frac{\Gamma(N+2)}{(N)!(2N)!}\right) f^{2N}(\xi)e^{\xi}, \tag{17}$$

where $0 < \xi < \infty$ and

$$w_j = x_j \frac{\Gamma(N+2)}{(L(N+1)!\,[(N+1)\phi_{N+1}(x_j)]^2)}, j = 0, 1, 2, \dots, N-1.$$

In particular, the second term on the right-hand side vanishes when $f(x)$ is a polynomial of degree at most $2N-1$ [24]. We define:

$$I_N u(x) = \sum_{j=0}^{N-1} a_j \phi_j(x), \tag{18}$$

Such that:

$$I_N u(x_j) = u(x_j), j = 0, 1, 2, \dots, N-1.$$

$I_N u$ is the orthogonal projection of u upon $\Re_N$ with respect to the discrete inner product and discrete norm as [24]:

$$< u, v >_{w,N} = \sum_{j=0}^{N-1} u(x_j)v(x_j)w_j, \tag{19}$$

$$||u||_{w,N} =< u, v >_{w,N}^{1/2} \tag{20}$$

thus for the MGLF Gauss-Radau interpolation we have:

$$< I_N u, v >_{w,N} =< u, v >_{w,N} \ \forall u. v\epsilon\Re_N$$

$$< I_N u, v >_{w,N} = < u, v > \tag{21}$$

3.5.4 Solving the problem with modified generalized Laguerre functions

To apply modified generalized Laguerre collocation method to Eq. (24) with boundary conditions Eq. (5), at first we expand$f(d_i)$ as follows:

$$I_N f(d_i) = \sum_{j=0}^{N-1} a_j \phi_j, \tag{22}$$

To find the unknown coefficients a_j's, we substitute the truncated series $f(d_i)$ into Eq. (24) and boundary conditions in Eq. (5). Also, we define Residual function of the form:

$$Res(d_i) = \sum_{j=0}^{N-1} a_j \phi_j'''(d_i) + \sum_{j=0}^{N-1} a_j \phi_j(d_i) \sum_{j=0}^{N-1} a_j \phi_j''(d_i) - \left(\sum_{j=0}^{N-1} a_j \phi_j'(d_i)\right)^2 - id \sum_{j=0}^{N-1} a_j \phi_j'(d_i) \tag{23}$$

$$\sum_{j=0}^{N-1} a_j \phi_j(0) = 0, \tag{24}$$

$$\sum_{j=0}^{N-1} a_j \phi_j'(0) = 1, \tag{25}$$

$$\sum_{j=0}^{N-1} a_j \phi_j(\infty) = 0. \tag{26}$$

Applying d_i in Eq. (23) with the N collocation points which are roots of functions L_n^a , we have N equations that generate a set of N non-linear equations; also, we have one boundary equation in Eq. (24-25). Now, all of these equations can be solved by Newton method for the unknown coefficients. We must mention Eq. (26) is always true; therefore, we do not need to apply this boundary condition.

Here we note that the Eq.(24) subject to boundary conditions Eq.(5) has an exact solution [35] as:

$$f(d_i) = \frac{1}{\sqrt{1+id}}\left(1 - e^{-\sqrt{1+(id)d_i}}\right) \tag{27}$$

While in the absence of the magnetic field where $id = 0$, the exact solution first obtained by Crane [16] is

$$f(d_i) = 1 - e^{-d_i}. \tag{28}$$

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.

4. Performance analysis

This approach is based on the modified generalized Laguerre which is an orthogonal function that solves the non-linear differential equation governing the problem on the semi-infinite domain without truncating it to a finite domain. Modified generalized Laguerre function was proposed to provide simple way to improve the convergence of the solution through collocation method by $N = 20$, $\alpha = 1$and $L = 0.99$. 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 Figure shows more accurate manner and convergencesfaster.

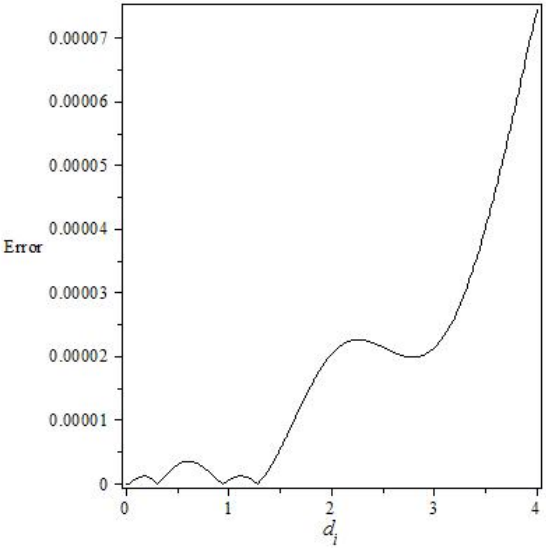

Fig. 2. Graph of Error by MGLFMs solution for id=0.6

In addition, The codes of MAPLE software of this implementation are mentioned in Appendix A. This implementation was executing in a computer whose information was:

- Windows Seven
- Proccesor: Intel(R) Core(TM) i3 CPU 2.53 GHz
- RAM: 4.00 GB
- System type: 32-bit Operating System

In this condition, the time of solution was reported 3.96s which is more efficient than other solutions. Also, solving this problem is not possible without boundary conditions; therefore, only the owner of these boundary conditions can solve this in efficient time.

5. Conclusion

In this paper, we proposed an efficient scheme including special technique to defend adversary against curious, search-replace and injection attacks. Actually, we shared data (defence against curious attack) and code them using a mathematical function (defence against search and replace), and efficiently sign every unit of data to prevent injection attack. Mathematical function is designed for initializing with the sensor properties such as id. Therefore, we use one-to-one function that hold in equation 4.

Moreover, based on this equation and boundary conditions, a new function for every sensor is released. This equation is general as well as the adversary knows this equation but the calculated function is hard to obtain without knowing boundary condition. Hence, variable encoded packet of every function detects no information about the original data. This technique is firm against injection attack which is the most rampant attack in general unattended wireless sensor network. Totally, we can claim that, our work is applicable and secure against various attacks.

6. Appendix A

```
>restart :
>with(orthopoly) : with(LinearAlgebra) :
>with(student) :
>with(plots) :
>n := scanf(`%d`)[1] :
>id := 0.6 :  l := 0.99 :
>y := [evalf(solve(L(n, 1, x/l), x))] :
>
  E := 0 :
   for i from 0 to n - 1do
      E := E + a[i]*(exp(-x/(2·l))·L(i, 1, x/l)) :
  end do:
  E :
>dE := diff(E, x) :
>ddE := diff(E, x, x) :
>dddE := diff(E, x, x, x) :
>Eq := dddE + E·ddE - (dE)^2 - id·(dE) :
>for i from 1 to n do A[i] := eval(Eq, x = y[i]) = 0;  end do:
>A[n - 1] := eval(E, x = 0) = 0 :
>A[n] := eval(dE, x = 0) = 1 :
>X := fsolve({seq(A[m], m = 1 .. n)}, {seq(a[i] = 0, i = 0 .. n - 1)}) :
>c1 := array(0 ..n - 1) :
>for i from 0 to n - 1 do c1[i] := subs(X, a[i]); od:
>fu := sum(c1[u]*exp(-x/(2·l))·L(u, 1, x/l), u = 0 .. n - 1) :
>dfu := diff(fu, x) :
>ddfu := diff(fu, x, x) :
>subs(x = 0, ddfu); dddfu := diff(fu, x, x, x) :
>plot(fu, x = 0 .. 5)
>plot(ddfu, x = 0 .. 5)  :
>evalf(eval(ddfu, x = 0)) :
>m := abs(fu - 1/sqrt(1 + id)·(1 - e^(-sqrt(1 + id)·x))) :
>plot(m, x = 0 ..4) ;
>fu :
>Eq1 := (dddfu) + (fu)·(ddfu) - (dfu)^2 - id·(dfu) :
> evalf(int((Eq1)^2, x = 0 ..10)) :
>∫_0^10 (Eq1)^2 dx :
```

7. References

[1] Pietro, R.D. Mancini, L.V. Spognardi, A. Soriente, C. Tsudik, G.: Catch me (if you can): Data survival in unattended sensor networks. IEEE international conference on pervasive computing and communications (PerCom). China. 185-194 (2008)

[2] Mateus, A. S. S. Margi, C. B. Simplicio, M. A. Geovandro, C. C. F. P. de Oliveira, B. T. : Implementation of data survival in unattended wireless sensor network using cryptography. IEEE conference on Local Computer Networks (LCN). USA. 961-967 (2010)

[3] MIRACL Big Integer Library, http://www.shamus.ie/(2009)

[4] Lim, C. H. Hwang, H. S.: Fast implementation of elliptic curve algorithm in GF(p^n). in public key cryptography series, lecture notes in computer science. springer. 1751. 405-421 (2000)

[5] Ren, W., Zhao, J., Ren, Y.: network coding based dependable and efficient data survival in unattended wireless sensor networks. Journal of Communications. 4, NO. 11,894-901 (2009)

[6] Gentry, C., Peikert, C., Vaikuntanathan, V.: trapdoors for hard lattices and new cryptography constructions: In STOC, ed. R. E. Ladner and C. Dwork, ACM, 197-206(2008)

[7] Boneh, D., Freeman, D. M.: linearly homomorphic signature over binary fields and new tools for lattice-based signatures. In Proceeding of PKC'11, LNCS 6571. 1-16

[8] Sakiadis, B. C.: Boundary-layer behaviour on continuous solid surfaces: I. boundary-layer equations for two-dimensional and axisymmetric Flow. AIChE J. 7, 26-28 (1961)

[9] Sakiadis, B. C.: Boundary-layer behaviour on continuous solid surfaces: II. boundary-layer equations for two-dimensional and axisymmetric flow. AIChE J. 7, 221-225 (1961)

[10] Crane, L. J.: Flow past a stretching plate. Z. Angew. Math. Phys. 21, 645-647 (1970)

[11] Andersson, H. I., Hansen, O. R., Holmedal, B.: Diffusion of a chemically reactive species from a stretching sheet. Int. J. Heat Mass Trans. 37, 659-664 (1994)

[12] Thakar, H. S., Chamkha, A. J., Nath, G.: Flow and mass transfer on a stretching sheet with a magnetic filed and chemically reactive species. Int. J. Eng. Sci. 38, 1303-1314 (2000)

[13] Raptis, A., Perdikis, C.: Viscous flow over a non-linearly stretching sheet in the presence of a chemical reaction and magnetic field. Int. J. Nonlinear. Mech. 41, 527-529 (2006)

[14] Rajagopal, K., Veena, P. H., Pravin, V. K.: Nonsimilar solutions for heat and mass transfer flow in an electrically conducting viscoelastic fluid over a stretching sheet saturated in a porous medium with suction/blowing. J. Porous Media. 11, 219-230 (2008)

[15] Bejan, A.: Convection heat transfer. Wiley-Interscience, New York, USA (1984)

[16] Akyildiz, F. T., Bellout, H., Vajravelu, K.: Diffusion of chemically reactive species in a porous medium over a stretching sheet. J. Math. Anal. Appl. 320, 322-339 (2006)

[17] Cortell, R.: MHD flow and mass transfer of an electrically conducting fluid of second grade in a porous medium over a stretching sheet with chemically reactive species. Chem. Eng. Process. 46, 721-728 (2007)

[18] Cortell, R.: Toward an understanding of the motion and mass transfer with chemically reactive species for two classes of viscoelastic fluid over a porous stretching sheet. Chim. Eng. Process. 46, 982-989 (2007)

[19] Prasad, K. V., Abel, M. S., Khan, S. K., Datti, P. S.: Non-darcy forced convective heat transfer in a viscoelastic fluid flow over a non-isothermal stretching sheet. J. Porous Media. 5, 41-47 (2002)

[20] Prasad, K. V., Abel, M. S., Datti, P. S.: Diffusion of chemically reactive species of a non-newtonian fluid immersed in a porous medium over a stretching sheet. Int. J. Non-Linear Mech. 38, 651-657 (2003)

[21] Ziabakhsh, Z., Domairry, G., Bararnia, H., Babazadeh, H.: Analytical solution of flow and diffusion of chemically reactive species over a nonlinearly stretching sheet immersed in a porous medium. J. Taiwan Inst. Chem. Eng. 41, 22-28 (2010)

[22] Kechil, S. A., Hashim, I.: Series solution of flow over nonlinearly stretching sheet with chemical reaction and magnetic fleld. Phy. Lett. A 372, 2258-2263 (2008)

[23] Dinarvand, S.: A reliable treatment of the homotopy analysis method for viscous flow over a non-linearly stretching sheet in presence of a chemical reaction and under influence of a magnetic fleld. Cent. Eur. J. Phys. 7, 114-122 (2009)

[24] Parand, K., Taghavi, A.: Rational scaled generalized Laguerre function collocation method for solving the Blasius equation. J. Comput. Appl. Math. 233, 980-989 (2009)

[25] Parand, K., Taghavi, A., Shahini, M.: Comparison between rational Chebyshev and modified generalized Laguerre functions Pseudospectral methods for solving Lane-emden and unsteady gas equentions. Acta Physica Polonica B. 40, 1749-1763 (2009)

[26] Coulaud, O., Funaro, D., Kavian, O.: Laguerre spectral approximation of elliptic problems in exterior domains. Comput. Method. Appl. Mech. Eng. 80, 451-458 (1990)

[27] Guo, B. Y., Shen, J., Xu, C. L.: Generalized Laguerre approximation and its applications to exterior problems. J. Comput. Math. 23, 113-130 (2005)

[28] Zhang, R., Wang, Z. Q., Guo, B. Y.: Mixed Fourier-Laguerre spectral and Pseudospectral methods for exterior problems using generalized Laguerre functions. J. Sci. Comput. 36, 263-283 (2008)

[29] Wang, Z. Q., Guo, B. Y., Wu, Y. N.: Pseudospectral method using generalized Laguerre functions for singular problems on unbounded domains. discret. contin. dyn. s. 11, 1019-1038 (2009)

[30] Iranzo, V., Falqus, A.: Some spectral approximations for differential equations in unbounded domains. Comput. Methods Appl. Mech. Engrg. 98, 105-126 (1992)

[31] Szeg, G.: Orthogonal polynomils. AMS, New York, (1939)

[32] Parand, K., Dehghan, M., Taghavi, A.: Modified generalized Laguerre function Tau method for solving laminar viscous flow: The Blasius equation. Int. J. Numer. Meth. Heat Fluid Flow. 20, 728-743 (2010)

[33] Parand, K., Shahini, M., Dehghan, M.: Rational Legendre Pseudospectral approach for solving nonlinear difierential equations of Lane-Emden type. J. Comput. Phys. 228, 8830-8840 (2009)

[34] Rajagopal, K., Tao, L.: Mechanics of mixture. World Scientific, Singapore, (1995)

[35] Gasper, G. Stempak, K., Trembels, W.: Fractional integration for Laguerre expansions. J. Math. Appl. Anal. 67, 67-75 (1995)

[36] Taseli, H.: On the exact solution of the Schrodinger equation with quartic anharmonicity. Int. J. Quantom. Chem. 63, 63-71 (1996)