1

Secure Data Aggregation in Wireless Sensor Networks. Homomorphism versus Watermarking Approach.

Jacques M. Bahi, Christophe Guyeux and Abdallah Makhoul
{jacques.bahi, christophe.guyeux, abdallah.makhoul}@univ-fcomte.fr

Computer Science Laboratory, University of Franche-Comté (LIFC)
Rue Engel-Gros, BP 527, 90016 Belfort Cedex, France

Summary. Wireless sensor networks are now in widespread use to monitor regions, detect events and acquire information. Since the deployed nodes are separated, they need to cooperatively communicate sensed data to the base station. Hence, transmissions are a very energy-consuming operation. To reduce the amount of sending data, an aggregation approach can be applied along the path from sensors to the sink. However, usually the carried information contains confidential data. Therefore, an end-to-end secure aggregation approach is required to ensure a healthy data reception. End-to-end encryption schemes that support operations over cypher-text have been proved important for private party sensor network implementations. These schemes offer two main advantages: end-to-end concealment of data and ability to operate on cipher text, then no more decryption is required for aggregation. Unfortunately, nowadays these methods are very complex and not suitable for sensor nodes having limited resources. In this paper, we propose a secure end-to-end encrypted-data aggregation scheme. It is based on elliptic curve cryptography that exploits a smaller key size. Additionally, it allows the use of higher number of operations on cypher-texts and prevents the distinction between two identical texts from their cryptograms. These properties permit to our approach to achieve higher security levels than existing cryptosystems in sensor networks. Our experiments show that our proposed secure aggregation method significantly reduces computation and communication overhead and can be practically implemented in on-the-shelf sensor platforms. By using homomorphic encryption on elliptic curves, we thus have realized an efficient and secure data aggregation in sensor networks. Lastly, to enlarge the aggregation functions that can be used in a secure wireless sensor network, a watermarking-based authentication scheme is finally proposed.

Key words: Wireless Sensor Networks; Secure Data Aggregation; Authentication; Homomorphic Encryption; Elliptic Curves; Watermarking.

1.1 Introduction

Wireless sensor networks have received enormous attention over past few years, due to a wide range of potential applications (environmental, ecological, military, *etc.*). A typical sensor network is expected to consist of a large number of sensor nodes deployed randomly in a large scale. Usually, these nodes have limited power, storage, communication, and processing capabilities, making energy consumption an issue.

A major functionality of a sensor node is to measure environmental values using embedded sensors, and transmit it to a base station called “sink”. The sensed data needs to be analyzed, which eventually serves to initiate some action. Almost this analysis presumes computation of the maximum, minimum, average, *etc*. It can be either done at the base station or by the nodes themselves, in a hierarchical scenario. In order to reduce the amount of data to be transmitted to the sink, it is beneficial that this analysis can be done over the network itself. To save the overall energy resources of the network, it is agreed that the sensed data needs to be aggregated on the way to its final destination. Sensor nodes send their values to certain special nodes, i.e., aggregators. Each aggregator then condenses the data prior to sending it on. In terms of bandwidth and energy consumption, aggregation is beneficial as long as the aggregation process is not too CPU-intensive. The aggregators can either be special (more powerful) nodes or regular sensors nodes.

At the same time, sensor networks are often deployed in public or otherwise untrusted and even hostile environments, which prompts a number of security issues (e.g., key management, privacy, access control, authentication, *etc.*). Then, if security is a necessary in other (e.g., wired or MANET) types of networks, it is much more so in sensor networks. Actually, it is one of the more popular research topic and many advances have been reported on in recent years.

From the above observations, we can notice the importance of a cooperative secure data aggregation in sensor networks. In other terms, after the data gathering and during transmissions to the base station, each node along the routing path cooperatively integrates and secures the fragments messages. In this paper, we focus on security data aggregation and we propose a simple secure homomorphic cyphersystem that allows efficient aggregation of encrypted data.

Data encryption becomes necessary in sensor networks when this type of sensors can be subject of many types of attacks [1]. Without encryption, adversaries can monitor and inject false data into the network. In a general manner the encryption process is done as follows: sensor nodes must encrypt data on a hop-by-hop basis. An intermediate node (*i.e.*, aggregator) possessing the keys of all sending nodes, decrypts the received encrypted value, aggregates all received values, and encrypts the result for transmission to the base station. Though viable, this approach is fairly expensive and complicated, due to the fact of decrypting each received value before aggregation, which generates an overhead imposed by key management. Encryption can solve the security problem, but how can we aggregate over encrypted data [1]?

Some privacy homomorphism based works have been proposed recently [2, 3, 4] that, without participating in checking, the aggregators can directly aggregate the encrypted data. However, such schemes, for the moment, need high and complex computations to encrypt data and aggregate it, which leads to large cypher-texts. Sensor nodes cannot provide sufficient CPU, memory and bandwidth to address such complex operations. For instance, Rivest Shamir Adleman (RSA) cryptosystems [5, 6] are used, which requires high CPU and memory capabilities to perform exponential operations. Therefore, in our study we adopt an elliptic curve encryption [7] that allows nodes to generate a smaller key size while providing the same security level of existing complex schemes. The cypher-system we exploit permits N additions and one product, thus it is not limited to a single basic function. A major advantage of our method is the fact that it has been proved safe, and until now it has not been cryptanalized. To assess the practicality of our technique, we evaluate it and compare it to existing cypher-system. The obtained results show that we significantly reduce computation and communication overhead and that our secure aggregation method can be practically implemented in on-the-shelf sensor platforms.

The rest of this paper is organized as follows: in the next section we present a review of some previous related work. Section 1.3 presents our security model. Sections 1.4 and 1.5 discuss the details of the proposed aggregation scheme for sensor networks. In Section 1.6, we describe simulation and results of simulation experiments. In Section 1.7 is proposed a new authentication scheme based on a watermarking approach, to improve the variety of aggregation functions through the secure wireless sensor network. Finally, we end the paper by a conclusion.

1.2 Related Work

The benefit and vulnerability, as well as the need to secure in-network aggregation, have been identified by a number of schemes in the literature. One approach [8] proposed a secure information aggregation protocol to answer queries over the data acquired by the sensors. Even though their method provided data authentication to provide secrecy, the data still sent in plain text format, which removes the privacy during transmission. Another one [9] proposed a secure energy efficient data aggregation (ESPDA) to prevent redundant data transmission in data aggregation. Unlike conventional techniques, their scheme prevents the redundant transmission from sensor motes to the aggregator. Before transmitting sensed data, each sensor transmits a secure pattern to the aggregator. Only sensors with different data are allowed to transmit their data to the cluster-head. However, since each sensor at least needs to transmit a packet containing a pattern once, power cannot be significantly saved. In addition, each sensor mote uses a fixed encryption key to encrypt data; data privacy cannot be maintained in their scheme. In [10], the authors presented a secure encrypted-data aggregation scheme for wireless sensor networks. The idea is based on eliminating redundant sensor readings without using encryption and maintains data secrecy and privacy during transmission. This scheme saves energy on sensor nodes but still do not guarantee the privacy of sent data.

The problem of aggregating encrypted data in sensor networks was introduced in [3] and further refined in [2]. The authors propose to use homomorphic encryption schemes to enable arithmetic operations over cypher-texts that need to be transmitted in a multi-hop manner. However, these approaches provide a higher level of system security, since nodes would not be equipped with private keys, which would limit the advantage gained by an attacker compromising some of the nodes. Unfortunately, existing privacy homomorphisms used for data aggregation in sensor networks have exponential bound in computation. It is too computationally expensive to implement in sensor nodes. Moreover, the expansion in bit size during the transformation of plain text to cypher-text introduces costly communication overhead, which directly translates to a faster depletion of the sensors energy. On the other hand and from security viewpoint, the cryptosystems [11] used in these approaches were cryptanalized [12, 13], which means they can't guarantee anymore high security levels.

In this paper we try to relax the statements above by investigating elliptic curve cryptography that allows feasible and suitable data aggregation in sensor networks beside the security of homomorphisms schemes. First, our proposed scheme for secure data aggregation in sensor networks is based on a cryptosystem, which has been proved safe and has not been cryptanalyzed. Another property that enforces the security level of such approach is coming from the fact that, as it is the case in ElGamal cryptosystem, for two identical messages it generates two different cryptograms. This property suggested fundamental for security in sensor networks [7, 10, 14], to the best of our knowledge, was not addressed in previous homomorphism-based security data aggregation works. Beside all these properties and due to the use of elliptic curves, our approach saves energy by allowing nodes to encrypt and aggregate data without the need of high computations. Lastly, the scheme we use allows more aggregations types over cypher data than the homomorphic cryptosystem used until now.

1.3 Security Model

In this work, we are primarily concerned with data privacy in sensor networks. Our goal is to prevent attackers from gaining any information about sensor data. Therefore, ensuring an end-to-end privacy between sensor nodes and the sink becomes problematic. This is largely because popular and existing cyphers are not additively homomorphic. In other words, the summation of encrypted data does not allow for the retrieval of the sum of the plain text values. Moreover, privacy existing homomorphisms have usually exponential bound in computation. To overcome this problem, in our model we propose a security scheme for sensor networks using elliptic curve based cryptosystem. We show that our model permits many operations on crypted data and does not demand high sensor capabilities and computation.

1.3.1 Operations over elliptic curves

In this section, we give a brief introduction to elliptic curve cryptography. The reader is referred to [15] for more details.

Addition and multiplication

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields [15]. Elliptic curves used in cryptography are typically defined over two types of finite fields: prime fields $\mathbb{F}_p$, where p is a large prime number, and binary extension fields $\mathbb{F}_{2^m}$ [16]. In our paper, we focus on elliptic curves over $\mathbb{F}_p$. Let $p > 3$, then an elliptic curve over $\mathbb{F}_p$ is defined by a cubic equation $y^2 = x^3 + ax + b$ as the set

$$\mathcal{E} = \left\{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p, y^2 \equiv x^3 + ax + b \pmod p\right\}$$

where $a, b \in \mathbb{F}_p$ are constants such that $4a^3 + 27b^2 \not\equiv 0 \pmod p$. An elliptic curve over $\mathbb{F}_p$ consists of the set of all pairs of affine coordinates (x, y) for $x, y \in \mathbb{F}_p$ that satisfy an equation of the above form and an infinity point $\mathcal{O}$.

The point addition and its special case, point doubling over $\mathcal{E}$ is defined as follows (the arithmetic operations are defined in $\mathbb{F}_p$) [15] :

Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points of $\mathcal{E}$. Then:

$$P + Q = \begin{cases} \mathcal{O} & \text{if} \quad x_2 = x_1 \text{ and } y_2 = -y_1, \\ (x_3, y_3) & \text{otherwise.} \end{cases}$$

where:

- $x_3 = \lambda^2 - x_1 - x_2$,
- $y_3 = \lambda \times (x_1 - x_3) - y_1$,

$$\lambda = \begin{cases} (y_2 - y_1) \times (x_2 - x_1)^{-1} & \text{if} \quad P \neq Q\,, \\ (3x_1^2 + a) \times (2y_1)^{-1} & \text{if} \quad P = Q. \end{cases}$$

Finally, we define $P + \mathcal{O} = \mathcal{O} + P = P, \forall P \in \mathcal{E}$, which leads to an abelian group $(\mathcal{E}, +)$. On the other hand the multiplication $n \times P$ means $P + P + ... + P$ n times and $-P$ is the symmetric of P for the group law + defined above for all $P \in \mathcal{E}$.

Public/Private keys generation with ECC

In this section we show how we can generate the public and private keys for encryption, following the cryptosystem proposed by Boneh *et al.* [7]. The analysis of the complexity will be treated in a later section.

Let $\tau > 0$ be an integer called “security parameter”. To generate public and private keys, first of all, two τ-bits prime numbers must be computed. Therefore, a cryptographic pseudo-random generator can be used to obtain two vectors of τ bits, q_1 and q_2. Then, a Miller-Rabin test can be applied for testing the primality or not of q_1 and q_2. We denote by n the product of q_1 and q_2, $n = q_1 q_2$, and by l the smallest positive integer such that $p = l \times n - 1$. l is a prime number while $p = 2 \pmod 3$.

In order to find the private and public keys, we define a group H, which presents the points of the super-singular elliptic curve $y^2 = x^3 + 1$ defined over $\mathbb{F}_p$. It consists of $p + 1 = n \times l$ points, and thus has a subgroup of order n, we call it G. In another step, we compute g and u as two generators of G and $h = q_2 \times u$. Then, following [7], the public key will be presented by (n, G, g, h) and the private key by q_1.

Encryption and Decryption

After the private/public keys generation, we proceed now to the two encryption and decryption phases:

- **Encryption :** Assuming that our messages space consists of integers in the set $\{0, 1, ..., T\}$, where $T < q_2$, and m the (integer) message to encrypt. First, a random positive integer is picked from the interval $[0, n-1]$. Then, the cypher-text is defined by
$$C = m \times g + r \times h \in G,$$
in which + and × refer to the additive and multiplication laws defined previously.
- **Decryption :** Once the message C arrived to destination, to decrypt it, we use the private key q_1 and the discrete logarithm of $(q_1 \times C)$ base $q_1 \times g$ as follows:
$$m = \log_{q_1 \times g} q_1 \times C.$$
This takes expected time $\sqrt{T}$ using Pollard's lambda method. Moreover, this decryption can be speed-up by precomputing a table of powers of $q_1 \times g$.

1.3.2 Homomorphic properties

As we mentioned before, our approach ensures easy encryption/decryption without any need of extra resources. This will be proved in the next section. Moreover, our approach supports homomorphic properties, which gives us the ability to execute operations on values even though they have been encrypted. Indeed, it allows N additions and one multiplication directly on cryptograms, which prevents the decryption phase at the aggregators level and saves nodes energy, which is crucial in sensor networks.

Additions over cypher-texts are done as follows: let m_1 and m_2 be two messages and C_1, C_2 their cypher-texts respectively. Then the sum of C_1 and C_2, let call it C, is represented by $C = C_1 + C_2 + r \times h$ where r is an integer randomly chosen in $[0, n-1]$ and $h = q_2 \times u$ as presented in the previous section. This sum operation guarantees that the decryption value of C is the sum $m_1 + m_2$. The addition operation can be done several times, which means we can do sums of encrypted sums.

The multiplication of two encrypted values and its decryption are done as follows: let e be the modified Weil pairing on the curve and g, h the points of G as defined previously. Let us recall that this modified Weil pairing e is obtained from the Weil pairing E [7], [17] by the formula: $e(P, Q) = E(x \times P, Q)$, where x is a root of $X^3 - 1$ on $\mathbb{F}_{p^2}$. Then, the result of the multiplication of two encrypted messages C_1, C_2 is given by $[C_m = e(C_1, C_2) + r \times h_1]$, where $h_1 = e(g, h)$ and r is a random integer pick in $[1, n]$.

The decryption of C_m is equal to the discrete logarithm of $q_1 \times C_m$ to the base $q_1 \times g_1$:
$$m_1 m_2 = \log_{q_1 * g_1} (q_1 \times C_m.)$$
where $g_1 = e(g, g)$.

1.4 Our Secure Data Aggregation for Sensor Networks

1.4.1 Presentation

Data aggregation schemes aim to combine and summarize data packets of several sensor nodes so that amount of data transmission is reduced. An example data aggregation scheme is presented in Figure 1.1 where sensor nodes collect information from a region of interest. When the user (sink) queries the network, instead of sending each sensor node's data to the base station, aggregators collect the information from its neighboring nodes, aggregates them, and send the aggregated data to the base station over a multihop path.

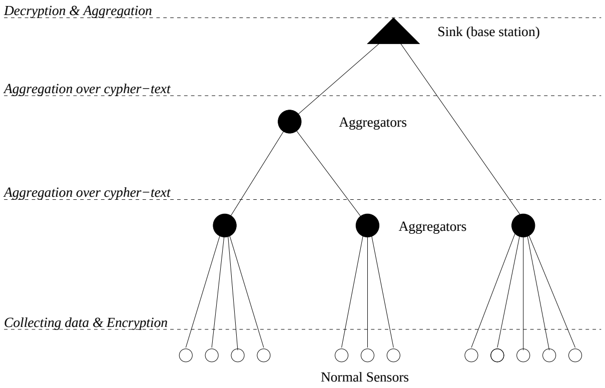
Decryption & Aggregation
Sink (base station)
Aggregation over cypher−text
Aggregators
Aggregation over cypher−text
Aggregators
Collecting data & Encryption
Normal Sensors

Fig. 1.1. Secure data aggregation in sensor networks

As the majority of wireless sensor network applications require a certain level of security, encryption of the sensed data before its transmission becomes necessary and it is preferable to decrypt the data only at the base station level (*c.f.* previous sections). In our work, we adopt the following scenario as shown in Figure 1.1: after collecting information, each sensor node encrypts its data according to elliptic curve encryption (*c.f.* Section 1.3.1) and sends it to the nearest aggregator. Then, aggregators aggregate the received encrypted data (without decryption) and send it to the base station, which in his turn decrypts the data and aggregates it. We notice that all aggregators can done N additions and the final layer of aggregators can done one multiplication on encrypted data.

1.4.2 Example of use

Computing the Arithmetic Mean

The arithmetic mean is the "standard" average, often simply called the "mean", defined for n values $x_1, \dots, x_n$ by

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i.$$

To compute the average of nodes measurements, aggregators can calculate the sum of the encrypted measurements and the number of nodes took these measurements and send it to the base station. More precisely, when using our scheme, each sensor encrypts its data x_i to obtain cx_i. The sensor then forwards cx_i to its parent, who aggregates all the cx_j's of its k children by simply adding them up. The resulting value is then forwarded. The sink ends up with value $Cx = \sum_{i=1}^{n} cx_i$. It can then decrypt Cx, and divide the result by n to derive the average.

Computing the Variance

Another common aggregation is to estimate the variance of the sensed values. Let us recall that the variance of n values $x_1, ..., x_n$ is defined by:

$$s_n^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2 = \left(\frac{1}{n} \sum_{i=1}^{n} x_i^2 \right) - \overline{x}^2.$$

Our scheme can also be used to derive the variance of the measured and encrypted data, by the same method as in [18]. In this case, each sensor i must compute $y_i = x_i^2$, where x_i is the measured sample, and encrypts y_i to obtain cy_i. x_i must also be encrypted, as explained in the previous section. The sensor forwards cy_i, together with cx_i, to its parent. The parent aggregates all the cy_j of its k children by simply adding them up. It also aggregates, separately, the cx_j, as explained in the previous section. The two resulting values are then forwarded. The sink ends up with values $Cx = \sum_{i=1}^{n} cx_i$ and $Cy = \sum_{i=1}^{n} cy_i$. Cx is used to compute the average Av, when Cy is used to compute the variance as follows: $Var = \frac{Vy}{n} - Av^2$, where Vy is the decryption of Cy.

Computing the Weighted Mean

The weighted mean of a non-empty set of data $x_1, x_2, \ldots, x_n$ with non-negative weights $w_1, w_2, \ldots, w_n$, is the quantity

$$\bar{x} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}{w_1 + w_2 + \cdots + w_n}.$$

We suppose now that each aggregator i of the first aggregation layer has computed the mean x_i of the encrypted values received from its sensor node. Additionally, we suppose that these aggregators are weighted, depending on their importance. For security reasons, this weight is also encrypted and the cypher value is denoted by w_i. This w_i can be proportional to the number of aggregated sensors. This weight can also illustrate the fact that two given regions have not the same relevance. To achieve weighted mean, each aggregator multiplies its encrypted mean x_i with encrypted weight w_i as it has been explained previously. The resulting value is then forwarded to the sink, which can decrypt $w_i \times x_i$ and sum all these decrypted values, to obtain the weighted mean defined above.

1.5 Security study

Due to hostile environments and unique characteristics of sensor networks, it is a challenging task to protect sensitive information transmitted by nodes to the end user. In addition, this type of networks has security problems that traditional networks do not face. In this section, we present a security study dedicated to wireless sensor networks. First we introduce the principal attacks that sensor networks can face and how our approach can support them, then we present some practical issues that improve the network security.

1.5.1 Related attacks and results

In a sensor network environment adversaries can commonly use the following attacks:

Known-plain text attack: They can use common key encryption to see when two readings are identical. By using nearby sensors under control, attackers can conduct a known-plain text attack.

Chosen-plain text attack: Attackers can tamper with sensors to force them to predeterminated values.

Man-in-the-middle: They can inject false readings or resend logged readings from legitimate sensor motes to manipulate the data aggregation process.

In Tables 1.1, 1.2 and similar to [16], we present a comparison between different encryption policies and possible attacks. In our method, as data are encrypted by public keys, and these public keys are sent by the sink to the sole authenticated motes, the wireless sensor network is then not vulnerable to a Man-in-the-middle attacks. On the other hand, our approach guarantees that for two similar texts gives two different cryptograms, which prevents the Chosen-plain text attacks and the Man-in-the-middle attacks. Finally, as the proposed scheme possesses the homomorphic property, data aggregation is done without decryption, and no private key is used in the network.

1.5.2 Practical issues

In this section we present some practical issues to our security model. First we study the sizes of the encryption keys and we compare it to existing approaches. Then, we present how we can optimize the sizes of cryptograms in order to save more sensors energy.

Table 1.1. Encryption polices and vulnerabilities

Encryption Policy	Possible attacks
Sensors transmit readings without encryption	Man-in-the-middle
Sensors transmit encrypted readings with permanent keys	Known-plain text attack Chosen-plain text attack Man-in-the-middle
Sensors transmit encrypted readings with dynamic keys	None of above
Our scheme	None of above

Table 1.2. Encryption polices and aggregation

Encryption Policy	Data aggregation
Sensors transmit readings without encryption	Generating wrong aggregated results
Sensors transmit encrypted readings with permanent keys	Data aggregation is impossible, unless the aggregator has encryption keys
Sensors transmit encrypted readings with dynamic keys	Data aggregation cannot be achieved unless the aggregator has encryption keys
Our scheme	Data aggregation can be achieved

Sizes of the keys

Cryptograms are points of the elliptic curve $\mathcal{E}$. They are constituted by couples of integer coordinates lesser than or equal to $p = lq_1q_2 - 1$.

It is commonly accepted [19], [20] that for being secure until 2020, a cryptosystem:

- must have $p \approx 2^{161}$, for EC systems over $\mathbb{F}_p$,
- must satisfy $p \approx 2^{1881}$ for classical asymmetric systems, such as RSA or ElGamal on $\mathbb{F}_p$.

Thus, for the same level of security, using elliptic curve cryptography does not demand high keys sizes, contrary to the case of RSA or ElGamal on F_p. The use of small keys leads to small cryptograms and fast operations for encryption.

1.5.3 Reducing the size of cryptograms

In this section we show how we can reduce the size of cryptograms while using ECC. This is benefit for sensor nodes in terms of reducing energy consumption by sending data with smaller size. The messages are encrypted with q_2 bits, which leads to cryptograms with a mean of 160 bits long.

Let us suppose that $p \equiv 3 \pmod 4$. As the cryptogram is an element (x, y) of $\mathcal{E}$, which is defined by $y^2 = x^3 + 1$, we can compress this cryptogram (x, y) to $(x, y$ mod

2)) before sending it to the aggregator (as the value of y^2 is known). In this situation, we obtain cryptograms with a mean of 81 bits long for messages between 20 and 40 bits long.

To decompress the cryptogram (x, i), the aggregator must compute $z = x^3 + 1 \bmod p$ and $y = \sqrt{z} \bmod p$, which can be written as $y = z^{(p+1)/4} \bmod p$, then :

- if $y \equiv i(\bmod 2)$, then the decompression of (x, i) is (x, y).
- else the decompression is $(x, p - y)$.

1.6 Experimental Results

To show the effectiveness of our approach we conducted a series of simulations comparing our method to another existing one based on RSA cryptosystem. We considered a network formed of 500 sensor nodes, each one is equipped by a battery of 100 units capacity. We consider that the energy consumption $''E''$ of a node is proportional to the computational time t, *i.e.*, $E = kt$. The same coefficient of proportionality k is taken while comparing the two encryption scenarii. Sensor nodes are then connected to 50 aggregators chosen randomly. Each sensor node choose the nearest aggregator. The running of each simulation is as follows: each sensor node takes a random value, encrypts it using one of the encryption methods then sends it to its aggregator. Aggregators compute the sum of the encrypted received data and send it to the sink. We compared our approach to the known RSA public-key cryptographic algorithms, and we evaluated the energy consumption of the network while varying the sizes of the keys and obviously the security levels. The energy consumption is the units of the battery used to do the encryption.

Tables 1.3 and 1.4 show the energy consumption of sensor nodes to do the encryption operations using our encryption method and the RSA one respectively. We varied the keys sizes and obviously the security levels. We notice that for the same level of security in our approach we used small keys while saving more energy. For instance, for high security levels (4 for example) a node using our approach needs to use a key of 167 bits instead of 1891 in the case of RSA and consumes 0.1 % of the battery power instead of 3.63 %.

Security level	Size p of the key	E (battery units)
1	46	0.02
2	85	0.05%
3	125	0.07
4	167	0.10

Table 1.3. Our approach

Tables 1.5 and 1.6 give the energy consumption E at the aggregation stage. The same hypothesis that above have been made, the sole difference is that aggregator nodes have a battery of 1000 units of energy. It can be seen that the energy needed by

Security level	Size of the key	E (battery units)
1	472	0.08
2	945	0.53
3	1416	1.63
4	1891	3.63

Table 1.4. RSA encryption

aggregators are between 50 and 500 times more important in the RSA-based scheme, for the same level of security.

Security level	Size p of the key	E (battery units)
1	46	0.02
2	85	0.04
3	125	0.07
4	167	0.10

Table 1.5. Our approach

Security level	Size of the key	E (battery units)
1	472	1.13
2	945	8.09
3	1416	24.74
4	1891	56.27

Table 1.6. RSA encryption

Figure 1.2 gives the comparison between RSA and elliptic curve based encryption, concerning the average energy consumption of an aggregating wireless sensor network. We can notice that our approach saves the energy largely greater than the case of RSA, where its depletion is so fast. Finally let us notice that, in addition of reducing the amount of energy units needed for encryption and aggregation, the sink receives many more values per second in EC-based networks than in RSA-based one.

1.7 Enlarging the number of allowing authentication functions

In the previous sections, we have proposed to use a homomorphism encryption scheme to support in-network processing while preserving privacy. Compared to existing secure aggregation schemes based on homomorphism encryption, our method has not been cryptanalysed. Moreover, due to the possibility to realize n additions and one product over the cypher values, this scheme enlarges the variety of allowing

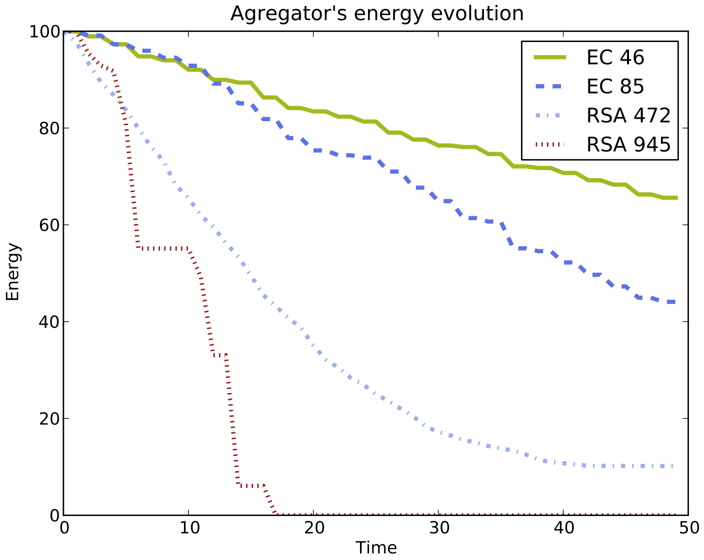

Fig. 1.2. Comparison of energy consumption

aggregation operations through cyphertexts. However, all of the homomorphism encryption schemes only allow some specific query-based aggregation functions, *e.g.*, sum, average, *etc*.

Another way to achieve secure data aggregation in wireless sensor networks is to authenticate sensing values. In-network processing presents a critical challenge for data authentication in wireless sensor networks. Current schemes relying on Message Authentication Code (MAC) cannot provide natural support for this operation, because a MAC computation is a very energy-consuming operation. Additionally, even a slight modification to the data invalidates the MAC.

In [21] a new way to achieve authentication through wireless sensor networks is introduced. It is based on digital watermarking and proposes an end-to-end, statistical approach for data authentication that provides inherent support for in-network processing. In this scheme, authentication information is modulated as watermark and superposed on the sensory data at the sensor nodes. The key idea formerly presented in [21] is to visualize the sensory data at a certain time snapshot as an image. Each sensor node is viewed as a pixel and its value corresponds to the gray level of the pixel. Due to this equivalency, information hiding techniques can be used to authenticate a wireless sensor network. The watermarked data can be aggregated by the intermediate nodes without incurring any en route checking. Upon reception of the sensory data, the sink is able to authenticate the data by validating the watermark, thereby detecting whether the data has been illegitimately altered. In this way, the aggregation-survivable authentication information is only added at the sources and checked by the data sink, without any involvement of intermediate nodes.

In [21] the authors proposes to use a data hiding scheme based on spread spectrum techniques to achieve authentication. In their proposal, “each sensor node embeds part of the whole watermark into its sensory data, while leaving the heavy computational load of watermark detection at the sink”. Moreover, as stated before, their scheme supports in-network aggregation. However spread spectrum is known to be not robust: even if their scheme survives to a certain degree of distortion, spread-spectrum cannot face to elementary blind attack. Furthermore, spread-spectrum data hiding techniques are only stego-secure in the “Natural Watermarking” situation [22]. The spread-spectrum subclass used in [21] is related to classical SS, *i.e.* with BPSK modulation [22]. This subclass is neither stego-secure [22], nor chaos-secure [23]. Among the consequences of these lack of security is the fact that an attacker how observes the network can access to the secret embedding key in all of the following situations:

- Watermarked Only Attack (WOA): the attacker has access only to watermarked contents.
- Known Message Attack (KMA): the attacker has access to pairs of watermarked contents and corresponding hidden messages.
- Known Original Attack (KOA): occurs when an attacker has access to several pairs of watermarked contents and their corresponding original versions.
- Constant-Message Attack (CMA): the attacker observes several watermarked contents and only knows that the unknown hidden message is the same in all contents.

To improve the security of the network in WOA setup, the use of Natural Watermarking instead of BPSK modulation is required [22]. Indeed, this subclass of spread-spectrum techniques, recalled in [22], is stego-secure and so can face WOA attacks. However, Natural Watermarking is less chaos-secure than the data hiding algorithm presented in [24]. This algorithm, based on chaotic iterations, is able to withstand attacks in KMA, KOA and CMA setups [25]. Moreover, this technique is more robust than spread-spectrum, as it is stated in [26]. To sum up, the use of the scheme proposed in [24] improves the security and robustness of the scheme presented in [21].

Finally, an hybrid approach of secure data aggregation in wireless sensor networks can be obtained by combining homomorphic encryption and watermarking-based authentication, as it is summed up in Figure 1.3.

1.8 Conclusion

In this paper, we presented an elliptic curve based approach for secure data aggregation in sensor networks. It is based on data encryption with homomorphic properties that provide the possibility to operate on cypher-text. It prevents the decryption phase at the aggregators layers and saves nodes energy. Existing works have exponential bound in computation and are not suitable for sensor networks, which we tried to

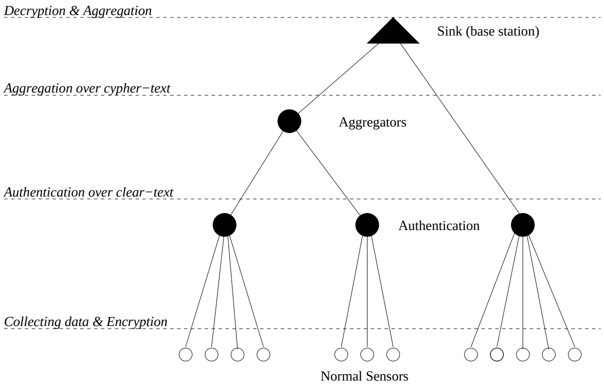
Decryption & Aggregation
Sink (base station)
Aggregation over cypher−text
Aggregators
Authentication over clear−text
Authentication
Collecting data & Encryption
Normal Sensors

Fig. 1.3. Secure data authentication and aggregation in sensor networks

relax in our approach. The proposed scheme permits the generation of shorter encryption asymmetric keys, which is so important in the case of sensor networks. The experimental results show that our method significantly reduces computation and communication overhead compared to other works, and can be practically implemented in on-the-shelf sensor platforms.

References

1. R. Chandramouli, S. Bapatla, and K.P. Subbalakshmi. Battery power-aware encryption. *ACM transactions on information and system security*, pages 162–180, 2006.
2. C. Castelluccia, E. Mykletun, and G. Tsudik. Efficient aggregation of encrypted data in wireless sensor networks. *Proc. of the 2nd Annual MobiQuitous*, pages 119–117, 2005.
3. J. Girao, M. Schneider, and D. Westhoff. Cda: Concealed data aggregation in wireless sensor networks. *Proceedings of the ACM Workshop on Wireless Security*, 2004.
4. M. Acharya, J.Girao, and D. Westhoff. Secure comparison of encrypted data in wireless sensor networks. *Third international symposium WiOpt'05*, pages 47–53, 2005.
5. W. Haodong, S. Bo, and L. Qun. Elliptic curve cryptography-based access control in sensor networks. *International Journal of Security and Networks*, 1(3-4):127–137, 2006.
6. A. Liu and P. Ning. Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. *Proceedings of IPSN'08*, pages 245–256, 2008.
7. D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. *Theory of Cryptography, LNCS*, pages 325–341, 2005.
8. B. Przydatek, D. Song, and A. Perrig. Sia: Secure information aggregation in sensor networks. *In proceedings of ACM SenSys conference*, pages 255–265, 2003.
9. H. Cam, S. Ozdemir, P. Nair, D. Muthuavinashinappan, and H. O. Sanli. Espda: Energy-efficient secure pattern based data aggregation for wireless sensor networks. *Computer Communication journal (29)*, pages 446–455, 2006.
10. S.-I. Huang, S. Shieh, and J. D. Tygar. Secure encrypted-data aggregation for wireless sensor networks. *Wireless Networks journal, Springer*, pages 1022–0038, 2009.
11. J. Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. *6th ISC conference*, pages 471–483, 2003.

12. J. Cheon, W.-H. Kim, and H. Nam. Known-plaintext cryptanalysis of the domingo ferrer algebraic privacy homomorphism scheme. *Inf. Processing Letters*, 97(3):118–123, 2006.
13. D. Wagner. Cryptanalysis of an algebraic privacy homomorphism. *6th ISC conference*, 2851, 2003.
14. H.-Y. Lin and T.-C. Chiang. Cooperative secure data aggregation in sensor networks using elliptic curve based cryptosystems. In Yuhua Luo, editor, *CDVE*, volume 5738 of *Lecture Notes in Computer Science*, pages 384–387. Springer, 2009.
15. D. Hankerson, A. Menezes, and S. Vanstone. Guide to elliptic curve cryptography. *Springer*, 2004.
16. R.C.C. Cheung, N.J. Telle, W. Luk, and P.Y.K. Cheung. Secure encrypted-data aggregation for wireless sensor networks. *IEEE Trans. on Very Large Scale Integration Systems*, 13(9):1048–1059, 2005.
17. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. *Crypto'2001, LNCS*, 2139:213–229, 2001.
18. C. Castelluccia, A. Chan, E. Mykletun, and G. Tsudik. Efficient and provably secure aggregation of encrypted data in wireless sensor networks. *ACM Trans. Sen. Netw.*, 5(3):1–36, 2009.
19. E. Barker and A. Roginsky. Draft nist special publication 800-131 recommendation for the transitioning of cryptographic algorithms and key sizes. 2010.
20. A.K. Lenstra and E.R. Verheul. Selecting cryptographic key sizes. *Jour. of the International Association for Cryptologic Research*, 14(4):255–293, 2001.
21. W. Zhang, Y. Liu, S.K. Das, and P. De. Secure data aggregation in wireless sensor networks: A watermark based authentication supportive approach. *Pervasive and Mobile Computing*, 4(5):658 – 680, 2008.
22. F. Cayre and P. Bas. Kerckhoffs-based embedding security classes for woa data hiding. *IEEE Transactions on Information Forensics and Security*, 3(1):1–15, 2008.
23. J.M. Bahi and C. Guyeux. A chaos-based approach for information hiding security. *ArXiv e-prints*, May 2010.
24. J.M. Bahi and C. Guyeux. Hash functions using chaotic iterations. *Journal of Algorithms & Computational Technology*, 4(2):167–181, 2010. Accepted manuscript. To appear.
25. C. Guyeux, N. Friot, and J. M. Bahi. Chaotic iterations versus Spread-spectrum: chaos and stego security. *ArXiv e-prints*, May 2010.
26. J.M. Bahi and C. Guyeux. A new chaos-based watermarking algorithm. In *SECRYPT 2010, International conference on security and cryptography*, pages ***–***, Athens, Greece, 2010. To appear.