Privacy Preserving Secure Data Aggregation for Wireless Sensor Networks

Vivaksha J. Jariwala*
Department of Information Technology, Sarvajanik College of Engineering and Technology, Dr. R. K. Desai Marg, Athwalines, Surat, Gujarat, India.

* Corresponding author. Tel.: +919898382008; email: vivaksha.jariwala@scet.ac.in
Manuscript submitted June 30, 2017; accepted August 12, 2017.
doi: 10.17706/jcp.13.6.655-677

Abstract: The Wireless Sensor Networks (WSNs) protocols commonly use in-network processing to optimize the communication costs. In-network processing involves processing of the sensed data on-the-fly during the course of the communication to the base station. However, due to the fusion of data items sourced at different nodes into a single one, the security of the aggregated data as well as that of the aggregating node, demands critical investigation. There has been numerous proof-of-concept attempts published in the literature, that devise secure data aggregation protocols offering either end-to-end or hop-by-hop secure aggregation. However, as per our modest observations, an integrated framework that offers all the necessary security attributes viz. confidentiality, privacy and integrity for secure data aggregation in WSNs is required. In this paper, we propose a framework for privacy preserving secure data aggregation in WSNs that provides support for confidentiality, privacy and integrity collectively. To support the confidentiality and privacy attributes in the framework, we evaluate the classical homomorphic encryption algorithms by comparative evaluation. Subsequently, based on this elaborate evaluation, we integrate the optimal algorithm amongst these, into tree topology in WSNs. To support the data integrity attribute, we analyse Elliptic Curve Digital Signature Algorithm (EC-DSA) and all of its published variants. However, in order to enhance the security strength of the basic EC-DSA, we propose our own variant of EC-DSA. In our humble observations, ours is a unique attempt that integrates the support for confidentiality, privacy and integrity in defined tree architecture for secure data aggregation protocol. We substantiate the proposal with elaborate experimental evaluation, too.

Key words: Elliptic curve cryptography, in-network processing, privacy homomorphism, secure data aggregation, wireless sensor networks (WSNs).

1. Introduction

Wireless sensor nodes that collaborate to form a Wireless Sensor Networks (WSNs), suffer from severe constraints in power, computational resources, memory and bandwidth [1]. The typical applications, for which the WSNs are deployed, require data to be sensed from different locations and be communicated to the base station. The eventual processing and analysis of the data is used to control a physical parameter of interest in the application concerned. However, an important design consideration herein is - *where* to process the data, i.e. to process the data *centrally* at the base station or to process the data in a *distributed* manner, *nearer* to the location where the data has originated. Naturally, if all the sensor nodes communicate the data directly to the base station, the resulting communication costs would increase, reducing the overall lifetime of the sensor nodes. Instead, an attractive option is to process the data in the network itself, aggregating different sensed values into a single value, before further communication. The

sensor nodes designated as the aggregator nodes, carry out such aggregation and the technique of processing the data on the fly in the network itself, is known as *in-network* processing.

However, the data aggregation operational paradigm, demands greater security concerns. This is so, because by default, the aggregator nodes are privy to the values sensed by the different sensor nodes jeopardizing their confidentiality. In addition, a single alteration to the aggregated data affects the sensor readings taken by many different sensor nodes. Hence, it is important to ensure the security of the aggregator node as well as that of the aggregation operations. Note that, with or without data aggregation, ensuring secure communications in WSNs is anyway a challenge; due to the conventional, two-party oriented, resource intensive security protocols and severe resource limitations of the WSNs nodes. However, ensuring secure data aggregation adds another dimension to this already highly researched problem.

1.1. Motivation

There are several attempts in the literature that focus on devising secure data aggregation (SDA) protocols. We classify them into a) SDA protocols based on link layer security architecture and b) SDA protocols using homomorphic encryption. SDA based on typical link layer security architecture [2]-[5] involve each intermediate node to decrypt the data and apply aggregation functions to it. Thus, exposes the private sensed data values to the aggregator nodes and compromises the privacy. In addition, due to the multiple security operations to be invoked, it increases the overall overhead in the network. Hence, devising an SDA protocol using homomorphic encryption (privacy homomorphism), is an attractive option [6].

Privacy homomorphism is a mechanism that allows direct computation on encrypted data [6]. This approach not only supports privacy, but also reduces vulnerabilities to attacks due to the infrequent cipher operations and thus enables reduced overhead. The SDA algorithms employing privacy homomorphism, broadly use two primitives viz. Symmetric key cryptography [6], [7] and asymmetric key cryptography [8], [9]. The symmetric key based approaches, as is common, inherently suffer from underlying uncertainty in key distribution [9], whereas the asymmetric key based approaches require higher computational power and storage resources [10], [11]. This has motivated the researchers, to devise approaches using Elliptic Curve Cryptography (ECC) (*with smaller key-sizes, shorter ciphertexts and better security-per-bit ratio*) based privacy homomorphism for SDA in WSNs [9]. ECC based SDA approaches are proposed in [9], [10], [12]-[15].

With respect to the attribute offered by a secure data aggregation approach, we observe that there are a few approaches that offer only confidentiality and privacy [9], [12]-[14], [16]-[27]. These approaches are suitable for various applications that demand obviously confidentiality and privacy e.g. health care monitoring system or vehicular traffic analysis system/vehicular location tracking system. In the former, the patient's blood pressure and sugar level need to be transferred confidentially; whereas in the latter, vehicle id or the location of specific vehicle needs to be transferred confidentially. However, without offering robustness of the data - through any data integrity protecting mechanism – all these approaches are susceptible to *data modification* attacks by an adversary. More simply, it is apparent that any security protocol that offers confidentiality without offering data integrity is practically unusable. Though, these approaches are notable attempts to showcase confidentiality and privacy in secure data aggregation, their practical implementation remains an issue.

Obviously motivated by this fact, there are approaches that provide only integrity [28]-[37]. These approaches can be used in environmental monitoring applications, wherein some environmental parameter is required to be monitored and controlled e.g. for temperature monitoring in a large building.

The important security attributes desired in such applications are ensuring the integrity of the aggregated data – the confidentiality/secrecy of the data is of no importance. The above-referred approaches cannot be used in applications like traffic analysis or battlefield monitoring where confidentiality is also required in addition to the data integrity. However, these approaches are notable attempts to showcase the integrity in secure data aggregation, their practical implementation remains an issue.

Thus, ideally when designing a *practically implementable* secure data aggregation system, one would expect the attribute of confidentiality as well as integrity to be offered by that protocol. One indeed finds approaches in the literature that provide confidentiality as well as integrity [15], [38]. The latter approach is an ECC based SDA scheme that provides confidentiality and integrity, but uses EC-EG for encryption. These approaches are suitable in the applications like *battlefield monitoring* wherein confidentiality, privacy and integrity of the data should be maintained. In battlefield surveillance, ensuring data confidentiality will mean not letting the adversary get hold of the data from the network. Whereas, ensuring integrity will mean, blocking off any attempts by the adversary, to inject false data into the network.

All of the approaches referenced until now, largely remain a *proof-of-concept* kind of approaches – the focus of these approaches is not to propose a complete framework for secure data aggregation. Motivated by the same, we attempt here to propose a complete security framework offering a suite of security algorithms to choose from, using a defined topology and selecting the desired security attributes. Our approach can be used in applications like environmental parameter monitoring, battlefield monitoring, identity tracking, industrial applications, etc. - where confidentiality, privacy and integrity of the data should be preserved.

The first issue that we pick up in the design of the framework is: which homomorphic encryption algorithm to use? For the purpose, we carry out an elaborate evaluation of the ECC based privacy homomorphic encryption algorithm for WSNs. Based on our literature review, we select the classical algorithms based on ECC, viz. Elliptic Curve Okamoto Uchiyama (EC-OU) [39], Elliptic Curve Paillier (EC-P) [39], Elliptic Curve Naccache-Stern (EC-NS) [39], Elliptic Curve Integrated Encryption Scheme (EC-IES) [12] and Elliptic Curve ElGamal (EC-EG) [8] for evaluation on the TinyOS platform using the TinyECC library. TinyECC provides support for three algorithms, namely Elliptic Curve Digital Signature Algorithm (EC-DSA) used for integrity, Elliptic Curve Diffie Hellman (EC-DH) used for key exchange and homomorphic encryption algorithm (EC-IES). TinyECC provides support for only one homomorphic encryption algorithm EC-IES in the WSNs. Thus, we evaluate the remaining algorithms viz. Elliptic Curve Okamoto Uchiyama (EC-OU) [39], Elliptic Curve Paillier (EC-P) [39], Elliptic Curve Naccache-Stern (ECNS) [39] and Elliptic Curve ElGamal (EC-EG) [12] in the WSNs and in the process of evaluation, we augment the TinyECC library. Subsequently, we integrate EC-OU into a tree topology and then further evaluate the implementation using the defined metrics. We find that EC-OU performs satisfactorily for secure data aggregation based on tree topology, in WSNs.

The underlying mechanism for data integrity could be either a Message Authentication Code (MAC) based algorithm or a digital signature based algorithm. As digital signature provides non-repudiation property, our focus in this paper is on the digital signature. We use Elliptic Curve Digital Signature Algorithm (EC-DSA) as the candidate algorithm. However, we observe several of its variations already proposed in the literature [40], [41]; but not evaluated on the TinyOS or other suitable platform for WSNs. We, therefore, implement and evaluate these variants on the TinyOS platform. In order to estimate the feasibility of these approaches, we also integrate the EC-DSA with our earlier benchmarked EC-OU setup to provide confidentiality, privacy and integrity.

As per our observations, the principal limitation of EC-DSA is that if an EC-DSA cryptosystem encounters

the same random number (as was used in a previous run of the EC-DSA), to generate another signature, the adversary can know the private key and generates false signature. The obvious solution to this issue, is to use two different numbers viz. k1 and k2 for signature generation [40]. Hung-Zih Liao et al. [40] prove formally that the use of two different numbers k1 and k2, indeed increases the security strength of the EC-DSA algorithm as compared to the same using only a single value of k. However, the issue that crops up, then why not to use three different numbers k1, k2, k3 especially if using two numbers k1 and k2 instead of a single random number k increases the security strength, then we may use one more random number. However, we note that using more k'i s would increase the overhead also. We also emphasize that the use of multiple k'is is driven by the motivation to ensure that the values of k'is are not repeated and thereby ensure the security of EC-DSA.

Driven by the same argument, another attractive alternative is to employ a random number generator, to generate the number k used in EC-DSA - that assures non-repetition of its output. However, we argue that, when being dependent on a random number generator, the strength of EC-DSA algorithm is not intrinsic, i.e. the security strength is not built into the algorithm, but is dependent on the implementation.

Hence, we try to explore alternate solutions that avoid the use of random numbers. Motivated by the properties of a Bloom filter [42] as an efficient set membership test data structure, we propose our own Bloom-filter based variant of EC-DSA that uses set membership test methods and light-weight hash functions [43] to generate a unique secret every time. As is evident from various research attempts at using the Bloom filter data structure in WSNs for different purposes [3], [4], [44]-[49].

The only argument, against the usage of the proposed variant could be "if at all we have a strong random number generator that ensures non-repetition, is it necessary to employ this variant?" However, our aim in dispelling with a random number is to make the EC-DSA algorithm inherently secure and not through its implementation. In addition to that, there is a likelihood that if one uses a poor pseudo-random number generator of the kind mentioned in [50], [51]; security of the algorithm is broken down. Moreover, all provably secure random number generator are public key based (a public key based approach does not always indicate that it is private/public key pair, but it implies that the approach employs typical public key crypto system operations like exponentiation over a big group) that involves heavy computations [46]. Thus, our proposal is justified that it enhances the intrinsic security strength of the EC-DSA algorithm, without assuming any guarantees from the underlying implementation.

We demonstrate with our experimentation, a Bloom filter based approach indeed works well within the resource bounds in WSNs. As per our evaluation, while offering added security strength, the proposed variant works reasonably well within only 7% additional overhead over the basic EC-DSA. To the best of our knowledge, our proposal is a simple and yet a unique attempt that applies the set membership test operation data structure viz. Bloom filter to an advantage to the EC-DSA in the resource constrained environment of WSNs while enhancing the security of the basic EC-DSA.

Thus, the proposed framework for Secure Data Aggregation in WSNs provides privacy and confidentiality using privacy homomorphic property and our own bloom filter based EC-DSA algorithm for data integrity. To the best of our knowledge, ours is a unique attempt that does so.

The rest of the paper is organized as follows: in Section 2, we present the related work in the area and summary of privacy homomorphic encryption algorithms. In Section 3, we briefly describe our approach to provide confidentiality and privacy. In Section 4, we evaluate various variants of EC-DSA and also extend our approach to provide integrity support for our framework. In Section 5, we proposed new variant of EC-DSA based on bloom filter and integrate it in our proposed framework. In Section 6, we describe our methodology of evaluation and the experimental setup used. In Section 7, we present the discussion of the results obtained followed by conclusion in Section 8.

2. Theoretical Background

As we know that, the privacy homomorphic schemes are broadly divided into two types: symmetric key cryptography [52], [7], [53] and asymmetric key cryptography [54]-[59]. The main advantage of symmetric key cryptography is efficiency [60], but it has a number of significant drawbacks viz. key distribution and key management. Though public key cryptography provides an elegant solution to the problems inherent in symmetric key cryptography, there are reasons [8], for less deployment of it in the resource-constrained environment. Vivaksha *et al.* [61], evaluated symmetric and asymmetric key homomorphic encryption algorithms in WSNs. RSA [54], DSA [60] and ECC [62] are the three most widely adopted public key cryptosystems. In Table 1, we show the key length ratio between these three crypto systems in same security property [63].

Table 1. The Key Length Ratio of ECC, RSA and DSA under the Same Security Property [63]

Key length of RSA/DSA	Key length of ECC	Ratio of RSA/ECC
512	106	5:1
768	132	6:1
1024	160	7:1
2048	210	10:1

From the table, we observe that though public key cryptosystems are expensive, ECC is an emerging type of public key cryptography that presents advantages compared to other public key algorithms. Einar et al. [8], investigated various suitable public-key schemes. They have also demonstrated the feasibility of efficiently computing elliptic curve operations on the microcontroller of the Mica2 sensor nodes [63]. ECC is characterized by small key, small system parameter, small public key, saving bandwidth, fast implementation, low power, and low hardware requirements.

For this reason, in this paper, we attempt to investigate the prevalent approaches in the literature for ECC based homomorphic encryption that ensures *Privacy* property in WSNs. In view of the fact that, the resource constrained are a prime concern in the environment, we attempt to explore the suitability of the ECC based privacy homomorphic encryption algorithms, by experimentally evaluating and analyzing ECC based homomorphic encryption algorithms. Eventually, we aim to prescribe the appropriate set of efficient algorithms offering the optimum level of security that can be used further to achieve our goal of Privacy Preserving Secure Data Aggregation.

To the best of our knowledge, ours is the first attempt in implementing and benchmarking the *storage requirements, energy consumption and execution time for encryption and decryption function* for the ECC based privacy homomorphic encryption algorithms in WSNs.

2.1. ECC Based Privacy Homomorphic Encryption Algorithms

From the literature survey, we observe that the basic additively privacy homomorphic EC-based schemes are EC-P [39], EC-NS [39], EC-OU [39] and EC-EG [8]. There are other ECC based homomorphic encryption schemes proposed in the literature [25], [14], [15] but these are sporadic attempts that lack any standardization. Hence, we focus only on classical and standard ECC based homomorphic encryption schemes whose security are proven in the literature. We describe these *four* specific schemes in this section.

Algorithm: Elliptic Curve Paillier (EC-P) [39]

Public Key: $n = pq$, G, E_{n^2}
Private Key: $\mu = lcm(p + 2, q + 2)$ or equivalently (p, q)
Encryption: plaintext $m \in Z_m$, $r \in_R Z_n$, cipher text $C = (m + nr)$
Decryption: compute $m = \frac{\Psi n(\mu C)}{\Psi n(\mu G)} \pmod n$

EC-P [39] is an asymmetric key homomorphic encryption algorithm. It is probabilistic schemes, i.e., if the same plaintext is encrypted more than once, it results in randomly distributed cipher texts. In this, the secret key is hidden. There is no way to obtain information out of encrypted data. Therefore, it is secure against *cipher text analysis attack*. In this scheme, depending on the applied system parameters, a set of plaintexts with corresponding cipher texts is not sufficient to deduct the secret key. Hence, it is not vulnerable to *known plain text attack*. This scheme, do not provide any inner protection against *replayed packets*. The additional integration of timestamps, sensor IDs, or a challenge/response system may help cope with the problem. This is vulnerable to *malleability* because in this cipher text can easily be altered. EC-P [39], do not have any protection against *unauthorized aggregation*. An attacker can take any two cipher texts and aggregate them without leaving marks. The security of this scheme is based upon the problem of computing residuosity classes over E_n.

Algorithm: Elliptic Curve Naccache-Stern (EC-NS) [39]

Public Key: $n = pq$, b, σ, G, $E_n(0, b)$
Private Key: (p, q) or $\mu = lcm\ (p + 1, q + 1)$
Encryption: plaintext $m \in Z\sigma$, $r \in_R Z_n$, cipher text $C = (m+ \sigma r)G$
Decryption: compute $u = (\mu/\sigma)$ $C = mG'$

EC-NS [39] is an asymmetric key homomorphic encryption algorithm. It is probabilistic schemes, i.e., if the same plaintext is encrypted more than once it results in randomly distributed cipher texts. In this, the secret key is hidden. There is no way to obtain information out of encrypted data. Therefore, it is secure against *cipher text analysis attack*. In this scheme, depending on the applied system parameters, a set of plaintexts with corresponding cipher texts is not sufficient to deduct the secret key. Hence, it is not vulnerable to *known plain text attack*. This scheme, do not provide any inner protection against *replayed packets*. The additional integration of timestamps, sensor IDs, or a challenge/response system may help cope with the problem. This is vulnerable to *malleability* because in this cipher text can easily be altered. EC-NS [39], do not have any protection against *unauthorized aggregation*. An attacker can take any two ciphertexts and aggregate them without leaving marks. The security of this scheme is based upon the problem of computing residuosity classes on elliptic curves.

Algorithm: Elliptic Curve ElGamal (EC-EG) [8]

Public Key: E, p, G, $Y = xG$, where $G, Y \in Fp$
Private Key: $x \in F_p$
Encryption: plaintext $M = map(m)$, $r \in_R F_p$,
Cipher text $C = (R, S)$, where $R = kG$, $S = M + kY$
Decryption: $M = -xR + S = -xkG + M + xkG, m = rmap\ (M)$

EC-EG [8], is an asymmetric key homomorphic encryption algorithm. It is probabilistic schemes, i.e., if the same plaintext is encrypted more than once, it results in randomly distributed cipher texts. In this, the secret key is hidden. There is no way to obtain information out of encrypted data. Therefore, it is secure against *cipher text analysis attack*. In this scheme, depending on the applied system parameters, a set of plaintexts with corresponding cipher texts is not sufficient to deduct the secret key. Hence, it is not vulnerable to *known plain text attack*. This scheme, do not provide any inner protection against *replayed packets*. The additional integration of timestamps, sensor IDs, or a challenge/response system may help cope with the problem. This is vulnerable to *malleability* because in this cipher text can easily be altered. In EC-EG [8], the aggregated random parts (k_1+ k_2) could somehow be noticed by the receiver, so that the interference may be detectable. Hence, provide protection against *unauthorized aggregation*.

Algorithm: Elliptic Curve Okamoto Uchiyama (EC-OU) [39]

Public Key: $n = p^2q$, G, H, E_n

Private Key: p

Encryption: plaintext $m < 2^{k-1}$, $r \in_R 2^{2k}$, cipher text $C = mG + rH$

Decryption: compute $m = \frac{\Psi_p((p+2)C)}{\Psi_p((p+2)G)} \pmod p$

Where $\Psi_p(x, y) = -\frac{x}{y} \pmod{p^2}$ and has the homomorphic property that if $P = mG$ for arbitrary points P, G, then $m = \frac{\Psi_p(P)}{\Psi_p(G)} \pmod p$ Provided that $G \neq O_p{}^2$

EC-OU [39], is an asymmetric key homomorphic encryption algorithm. It is probabilistic schemes, i.e., if the same plaintext is encrypted more than once it results in randomly distributed cipher texts. In this, the secret key is hidden. There is no way to obtain information out of encrypted data. Therefore, it is secure against *cipher text analysis attack*. Its encryption process relies on random numbers; the resulting cipher text is probabilistic and hence robust to *known plaintexts attack*. This scheme, do not provide any inner protection against *replayed packets*. In this scheme, the content m of the cipher text c = mG + rH can be modified by multiplying or dividing g that is part of the public key. Thus, this scheme is not secure against *malleability*. In order to modify the content of a cipher text, an adversary needs a part of the secret key. EC-OU [39], do not have any protection against *unauthorized aggregation*. An attacker can take any two cipher texts and aggregate them without leaving marks. In this algorithm, the chinese reminder theorem is used to combine two elliptic curves. Its security is based on factoring $n=p^2q$.

The security of EC-OU is based on factorization problem. If we select large prime numbers as factors in EC-OU, then the time required to break EC-OU is higher than the overall lifetime of WSNs. Hence, the security provided by the EC-OU is sufficient to achieve SDA in WSNs. In addition, EC-OU has an added advantage that it does not entail a higher number of computations in decryption function. In Table 2, we show a summary of security attributes of the *ECC* homomorphic encryption algorithms (discussed in the previous section).

Table 2. Security Characteristics of ECC Privacy Homomorphic Algorithms

Name of Algorithm	Acronym	Ciphertext analysis	Known plaintext attack	Replay attack	Malleability	Unauthorized aggregation	Key Features
Elliptic Curve Okamoto-Uchiyama	EC-OU	X	√	X	X	X	Security is equivalent to factoring
Elliptic Curve Paillier	EC-P	√	√	X	X	X	Security is based upon computing residuosity classes oven En
Elliptic Curve Naccache-Stern	EC-NS	√	√	X	X	X	Security is based upon computing residuosity classes on Elliptic Curve
Elliptic Curve ElGamal	EC-EG	√	√	X	X	X	Security is based upon the elliptic curve discrete log problem (EC-DLP)

2.2. Related Work

In this section, we describe attempts prevalent in the literature, to implement one or the other ECC based privacy homomorphic algorithms in the resource-constrained environment. An Liu *et al.* [12] implement TinyECC as a configurable library for ECC operations in WSNs. However, in this attempt, the authors implement EC-DSA (integrity preservation), EC-DH (key exchange) and EC-IES (homomorphic encryption) and apparently highlight EC-IES, as the best suited homomorphic encryption algorithm. Osman Ugus *et al.* [13] implement EC-EG, but without using vital metrics of evaluation for WSNs viz. *Storage utilization, energy consumption, encryption-decryption time*. Osman Ugus *et al.* [13] again use only EC-EG privacy homomorphic algorithm. S. Peter *et al*. [9] theoretically evaluated four ECC based privacy homomorphic encryption algorithms, without any underlying implementation. Vivaksha Jariwala *et al.* [61] attempt comparative empirical evaluation, but using *non-ECC* based privacy homomorphic encryption algorithms. Xiaoyan Wang *et al.* [16], propose secure and efficient scheme for data aggregation in WSNs. The proposed approach uses Castelluccia-Mykletun-Tsudik [7] scheme for encryption. Castelluccia-Mykletun-Tsudik [7] is a symmetric key cryptography based scheme and suffers from the key management issues. Poornima et al. [17], follow a similar approach based on symmetric key cryptography where the same key is shared between the leaf node and the base station. Jacques M. *et al.* [14], propose a scheme for SDA using elliptic curve cryptography that achieves only confidentiality without integrity.

None of the approaches discussed so far, support confidentiality as well as message integrity that the authors in [15], attempt to do. The proposal in [15] provides confidentiality as well as integrity; although without using ECC for data integrity. Vimal Kumar et al. [38], propose ECC based SDA scheme that provides confidentiality and integrity, but that uses EC-EG for encryption. However, as per evaluation [9], decryption time is more for this algorithm.

Thus, as mentioned before, neither do we observe any research attempt to evaluate comparatively, the ECC-based homomorphic encryption algorithms using defined metrics in literatures; nor do we see any attempt in using such a benchmarked algorithm in a tree topology, to showcase *truly secure* data aggregation. Thus, we believe ours is the first and unique attempt in proclaiming EC-OU as the efficient algorithm for WSNs and integrating it in the tree topology for providing privacy and confidentiality, as well as the use of novel bloom filter based variant of EC-DSA to provide integrity and achieving secure data aggregation that supports all the required security attributes.

3. The Proposed Approach

The main objective of the theoretical and empirical evaluation of the ECC based algorithms is to find a suitable candidate that can be plugged into the tree topology. Further, that algorithm can be used for secure data aggregation in resource constrained environments of WSNs.

3.1. Network Architecture

In wireless sensor networks, the sensor nodes are randomly distributed in an open area. Hence, it must require a specific network architecture that receives the sensor readings and forward the aggregated results to the base station. The network architecture further can be classified into two types: tree based network architecture and cluster based network architecture. In tree based architecture, the topology changes in only two conditions: when a sensor node is dying or when a new node is added to the network. Hence, in tree based architecture, topology changes not as frequently as of a cluster based network architecture, where cluster-heads are re-elected periodically that lead to change in topology from time to time [16]. Hence, we employ the tree architecture to examine the secure data aggregation for ECC homomorphic encryption schemes. However, we emphasize that our proposed framework can as well

extended to any network architecture including cluster based architecture.

3.2. The Proposed Algorithms

In our scheme, we propose two algorithms, first for the sensor node and the other one for the base station. Each sensor node would execute *SensorNodeAlgo ()* that takes a plain text message and private key as input to the algorithm. Next each sensor node, would compute its public key by multiplying its private key to the base point of the elliptic curve E. Subsequently, each sensor node comprises of two large prime number p & q and calculates n as public key and calculate H=nG with point multiplication. Next, sensor node would compute the cipher text and transmit it to its parent node.

Cipher text C = mG + rH

The parent node receives cipher text from the children, performs a summation of all the received cipher texts and finally transmits the aggregated cipher text to the base station.

Algorithm 1: SensorNodeAlgo ()

```
// Maps its reading m_i on the elliptic curve E
// Elliptic Curve Parameters E = (q, a, b, G, p, h)
// Each sensor node will computes following
1. Public Key:  n = p^2q, G, H, Q=pG
2. Encryption:  plaintext m < 2^(k-1), r ∈_R 2^(2k),
                Cipher text C = mG + rH
3. If sensor is a parent (Aggregator Node)
     c = ∑c_i // combines all cipher texts into one cipher text
   End if
```

The base station algorithm takes cipher text as input to the algorithm. The algorithm then decrypts the incoming packets and calculates plain text that is the summation of all the messages from all the sensor nodes.

Algorithm 2: BaseStationAlgo ()

```
// Maps its reading c_i from the elliptic curve E
// Elliptic Curve Parameters E = (q, a, b, G, p, h)
// Base station will computes following
1. Public Key: n = p^2q, G, H, Q=pG
2. Private Key:    p
3. Compute m = Ψp((p+2)C) / Ψp((p+2)G)  (mod p)

              Ψp(x, y) = - x/y (mod p^2)

4. The base station can get m=∑m_i
```

4. Support of Data Integrity

Any security algorithm for WSNs, must minimally provide the attributes viz. Confidentiality, data integrity and entity authentication. However, a protocol that provides confidentiality without support for the data integrity is meaningless. Hence, we focus on investigating techniques for supporting the data integrity. The approaches to provide data integrity can be either *cryptography-based* or *non-cryptography-based*. Our focus here is, only on cryptographic approaches. As per our literature survey, we categorize the techniques for supporting data integrity in WSNs into three classes viz. *Signature based, hash function based* and *Message Authentication Code (MAC) based*. Observing the fact that the digital

signature based approach yields non-repudiation property, we focus on digital signature based solution for supporting data integrity. There are two types of ECC based signatures available in literature viz. Elliptic Curve Digital Signature Algorithm (EC-DSA) [64] and Elliptic Curve Pintsov Vanstone Signature (EC-PVS) [65]. EC-DSA in TinyECC [12], is an example of a standard signature based algorithm for data integrity that is our primary focus of this research. The EC-DSA has a small key size that leads to faster computations time and reduction in processing power, storage space and bandwidth. This makes the EC-DSA implementation, suitable for the resource-constrained environments such as the Wireless Sensor Networks (WSNs) [1].

EC-DSA uses a random number k to generate the signature. Therefore, if the same number is used for generating another signature, then an adversary can find the value of private key x. The adversary can then use private key x to generate another signature. The obvious solution is to employ a random number generator that assures non-repetition of its output. However, in that case, the strength of the EC - DSA algorithm is not *intrinsic, i.e.* not built into the algorithm, but dependent on the implementation. One of the panaceas to this issue is to use two different numbers viz. k_1 and k_2 for signature generation [40]. Hung-Zih Liao *et al.* [40], prove formally in their paper, this has indeed increased the security strength. However, the logical issue is why *not* to use three different numbers – especially if using two numbers k_1 and k_2 instead of a single random number k increases the security strength, then using one more shall surely further increases the same. We indeed propose one and experiment with it as explained further in the section.

However, two vital issues that crop up here are as follows:

- If increasing the number of random numbers used to generate the signature increases the security strength, then why not to employ still higher numbers of such random numbers?
- Whether the overhead incur in generating increased number of random numbers is tolerable?

Such issues provoked us to explore any other alternative to argue that eventually the goal in using multiple k_i's is only to prevent the probability of their repeated occurrences; so that the signature cannot be forged. Hence, would not be worthwhile to explore any alternate mechanism to ensure the same, i.e. increases the security strength of EC-DSA while at the same time preventing multiple invocation of k_i's? With this aspect in focus, and with the resource constrained in WSNs at the backdrop, we observe that the space-efficient probabilistic set membership test data structure can be employed for the purpose here [42]. Thus, we propose a bloom filter based EC-DSA that uses set membership test methods and light-weight hash functions [43], to generate unique secret every time. Thus, because whether a k_i generated is tested a priori for its use in earlier runs, before actually using it, it is not possible to obtain the private key for the adversary and generate a false signature.

As our evaluation also shows, the overhead in our approach is tolerable, even in the resource-constrained environments while imparting the necessary security strength to the EC-DSA. There is no doubt that our approach being simple, can easily be extended to other environments also – beyond WSNs. To the best of our knowledge, our proposal is a simple and yet a unique attempt that applies the set membership test operation data structure viz. Bloom filter, to an advantage to the EC-DSA in the resource constrained environment of WSNs while enhancing the security of EC-DSA.

4.1. A Possible Attack on EC-DSA

In the original EC-DSA, the integer k selected in the signature generation step should be unique to sign distinct messages. This means, every time a message is to be signed; a unique, distinct secret *k* should be used to sign messages. If it is not so, the private key *x* can be disclosed and it makes the scheme vulnerable to the attacks. This is shown in [40]. Assume that the same secret *k* is used to generate two EC-DSA signature (r, s_1) and (r, s_2), for two different messages m_1 and m_2. Thus,

$s_1 = k^{-1}(H(m_1)+xr) \pmod n$

$s_2 = k^{-1}(H(m_2)+xr) \pmod n$

Here $H(m_1)$ is Hash of message m_1 and $H(m_2)$ is Hash of message m_2.

$$ks_1 = H(m_1) + xr \pmod n \quad (1)$$

$$ks_2 = H(m_2) + xr \pmod n \quad (2)$$

Now, if we subtract equation 1 from equation 2

$k(s_2-s_1)=H(m_2) - H(m_1) \pmod n$

If, $s_2 \neq s_1 \pmod n$, that occurs with high probability, then,

$k=(s_2-s_1)^{-1}(H(m_1) - H(m_2)) \pmod n$

Hence, an adversary can determine k and then use k to reveal the secret x. Thus, if the same secret k is used to sign two different messages, an adversary can easily reveal secret x. There are numerous attempts made in the literature to deal with this issue [40], [66], [41], [67], [68]. We critically analyse these efforts and use the same analysis to justify our motivation in proposing a new variant of EC-DSA.

4.2. Variant#1 — Limited Computation Capacity Signer

Hu Junru *et al.* [41] propose a variant for limited computation capacity signer. The scheme in [41], is suitable for limited computation capability like a signer; using his smart card that stores secret key and signs a message on a terminal. Here, the advantage is that there is no need of calculating the inverse of x in each individual signing operation. The private key of the signer is x that will remain stable for a period, it can be pre computed and stored in the key generation phase itself. Here, in this scheme [41], attack possible on basic EC-DSA is possible. Hence, it is also not secure.

4.3. Variant#2 — Limited Computation Capacity Verifier

Hu Junru *et al.* [41], propose a variant for limited computation capacity verifier. The scheme proposed in [41], is suitable for the verifier who has limited compute apparatus. In this scheme, the complexity of verification operation is lesser as compared to that of the previous schemes. In this scheme, k^{-1} is no longer be calculated, but we must calculate $(h+rx)^{-1}$ in the signing phase. However, there is no need of calculating inverse in verification phase that is one of the most expensive operations in modular arithmetic. Therefore, the complexity of the verification process is less in this scheme. Here in this scheme [41], attack possible on basic EC-DSA is possible. Hence, it is also not secure.

4.4. Variant#3 — Using Two Different Secrets

In the variant proposed in [40], the digital signatures are generated using two secrets k_1 and k_2, instead of relying upon a single secret k. Here, x cannot be determined even if the same secret k_1 and k_2 is repeated. Hung-Zih Liao *et al.* formally proved this, in their paper [40]. If a signature (r_1, s) on a message m was indeed generated by A, then $s = k^{-1}{}_1(H(m)k_2 + x(r_1 + r_2)) \pmod n$. If the same secret k_1, k_2 was used to generate EC-DSA signatures (r_1, s_1) and (r_1, s_2) on two different messages m_1 and m_2, then [40],

$s_1 = k{-}11(H(m_1)k_2 + x(r_1 + r_2)) \pmod n$

$s_2 = k{-}11(H(m_2)k_2 + x(r_1 + r_2)) \pmod n$

Where $H(m_1)$ = SHA-1(m_1) and $H(m_2)$ = SHA-1(m_2). Then,

$$k_1s_1 = H(m_1)k_2+x(r_1+r_2) \pmod n \quad (3)$$

$$k_1s_2 = H(m_2)k_2+x(r_1+r_2) \pmod n \quad (4)$$

Subtraction gives $k_1 (s_1 - s_2) = (H(m_1)-H(m_2))k_2 \pmod n$. We cannot determine k by this equation and

then use this to recover *x.* Hence, this scheme is more secure. However, the processes are more complex than the original EC-DSA.

5. Proposed Variant (Variant #4) of EC-DSA Based on Bloom Filter

As already discussed in the previous section; if the same secret *k* is used to sign two different messages, an adversary can easily reveal secret *x.* Therefore, after revealing secret *x* any adversary can easily generate a malicious signature on the message. To surmount that, we can use additional secrets to generate signatures as proposed in [40]. We have also proposed a variant of EC-DSA that uses multiple secrets (three) to generate the signature. Our proposed variant of EC-DSA using multiple secrets is discussed further.

Signature Generation ()

1. Select k_1, k_2, k_3 $1 \leq k_1, k_2, k_3 < n$.
2. $k_1G = (x_1, y_1)$, $r_1 = x_1 \pmod n$
3. $k_2G = (x_2, y_2)$, $r_2 = x_2 \pmod n$
4. $k_3G = (x_3, y_3)$, $r_3 = x_3 \pmod n$
5. $s = k_1^{-1}(H(m)k_2k_3 + x(r_1 + r_2 + r_3)) \pmod n$
6. (r_1, s) is the signature of m.

In this scheme, we cannot determine k_1, k_2, k_3 and then use this to recover *x*. Hence, this scheme is more secure than basic EC-DSA. However, when using additional random numbers (secrets) to enhance the security strength, a vital issue that crops up is viz. How many such secrets invocations to use? To answer the same, we also propose a solution that uses the set membership data structure viz. Bloom filter, to avoid repetition of a random number (secrets) used in EC-DSA. Our proposed approach uses bloom filter to generate unique *k* that can be used to sign different messages. Hence, our approach is using different secrets every time to generate different signatures without multiple secrets and consequently there are no chances of generation of false signatures.

Signature Verification ()

1. $w = s^{-1} \pmod n$
2. $u_1 = H(m)wk_2k_3 \pmod n$
3. $u_2 = (r_1 + r_2 + r_3)\ w \pmod n$
4. $u_1G + u_2Q = (x_4, y_4)$,
5. $v = x_4 \pmod n$
6. $V = r_1$ ➔ accept the signature

5.1. Proposed Algorithm

Select Ep(a,b), x, and $1 \leq x < n$. Select $G \in$ Ep(a,b) with order n and compute $Q = xG$. Public key (Ep(a,b), p,G,n,Q). Private key: x.

Signature Generation ()

To sign a message m, an entity A with domain parameters (p,Ep(a,b),G,n) and associated key pair (x,Q) does the following:

1. Select k, $1 \leq k < n$.
2. Create Bloom Filter
3. Call Membership Test for k
4. If returns yes, go to step 1 Else go to the next step
5. kG= (x_1, y_1), r = $x_1 \pmod n$
6. s = $k^{-1}(H(m)+xr) \pmod n$

   (r, s) is the signature of m.

Signature Verification ()

The receiver can verify the authenticity of the sender's signature (r, s) for message m by performing following operations:

1. $w = s^{-1} \pmod n$
2. $u_1 = H(m)w \pmod n$
3. $u_2 = rw \pmod n$
4. $u_1G+u_2Q = (x_2, y_2)$,
5. $v = x_2 \pmod n$
7. $v = r$ accept the signature

5.2. Proof of the Scheme

Our proposed approach uses bloom filter to generate separate k that can be used to sign different messages. Therefore, attack possible on basic EC-DSA is not possible on our variant of bloom filter based EC-DSA. In our bloom filter based approach, we first check set membership test. If the number generated belongs to the set, then we generate another number, if it doesn't, then we use that secret for the generation of the signature. Therefore, every time different secrets are used to generate different signatures. So, here secret $bloom_1$ and $bloom_2$ are used to generate two EC-DSA signature (r, s_1) and (r, s_2) for two different messages m_1 and m_2. Thus,

$s_1 = bloom_1^{-1}(H(m_1)+xr)(mod\ n)$

$s_2 = bloom_2^{-1}(H(m_2)+xr)(mod\ n)$

Here $H(m_1)$ is Hash of message m_1 and $H(m_2)$ is Hash of message m_2.

$$bloom_1 s_1 = H(m_1) + xr \pmod n \tag{5}$$

$$bloom_2 s_2 = H(m_2) + xr \pmod n \tag{6}$$

Now, if we subtract equation 5 from equation 6

$(bloom_2 s_2 - bloom_1 s_1) = H(m_2) - H(m_1) \pmod n$

If, $s_2 \neq s_1 \pmod n$, that occurs with high probability, then, we cannot determine $bloom_1$ and $bloom_2$ by this equation and then use this to recover *x*. Hence, this scheme is secure.

5.3. Integration of Integrity of Novel Bloom Filter Based Approach with Secure Data Aggregation

In our proposed approach of secure data aggregation, algorithm 1 is to be implemented on the sensor nodes. In this, each sensor node computes signature s_i on the outgoing message m_i.

$s_i = sign(m_i)$

Each sensor node will generate cipher text c_i of the message.

$c_i = Enc(m_i)$

If a sensor node is aggregator node, then it combines all the signatures received from the child node into one signature and all the cipher texts received from the child node into one cipher text.

Algorithm 2 is to be implemented on the base station. The base station receives aggregated cipher texts and aggregated signatures from sensor nodes. The base station applies decryption function on aggregated cipher texts and gets the original message that is the summation of all the plain text messages. Similarly, the base station accepts the message only if the signature is verified. Thus, our approach ensures confidentiality and privacy through privacy homomorphic encryption and in addition to that, it ensures integrity through digital signature.

Algorithm: SensorNodeAlgo ()

```
// Maps its reading m_i on the elliptic curve E
// Elliptic Curve Parameters E = (q, a, b, G, p, h)
// Each sensor node will computes following
1.  Public Key:  n = p^2q, G, H, Q=pG
2.  Private Key:  p
3.  Signature Generation:
       a.select bloom, 1<=bloom<n
       b.Create Bloom Filter
       c.Call MembershipTest for bloom
       d.If returns yes, go to step a
         Else go to the next step
       e.bloomG=(x_1,y_1), r=x_1(mod n)
       f.Each sensor computes signature
         s_i=bloom^-1(H(m)+pr)(mod n)
        (r,s_i) is the signature
4.  Encryption:
    Plaintext m_i< 2^(k-1),r∈_R 2^2k,
    Cipher text C_i = m_iG + rH
5.  If the sensor is a parent (Aggregator Node)
       S = ∑ s_i //combines all signatures into one signature
       C = ∑ c_i //combines all cipher texts into one cipher text
   End if
```

Algorithm: BaseStationAlgo ()

```
// Maps its reading c_i from the elliptic curve E
// Elliptic Curve Parameters E = (q, a, b, G, p, h)
// Base station will computes following
1. Public Key: n = p^2q, G, H, Q=pG
2. Private Key: p
```

3. Compute $m = \frac{\Psi p((p+2)C)}{\Psi p((p+2)G)} \pmod p$

$$\Psi p(x, y) = -\frac{x}{y} \pmod{p^2}$$

```
4. Base station can get m = ∑ m_i
5. Compute w=s^-1(mod n)
           u_1=H(m)w(mod n)
           u_2=rw(mod n)
           u_1G + u_2Q = (x_2,y_2)
          v=x_2(mod n)
6. If v==r then
           A signature is verified
```

8. End if

6. Experimental Setup and Methodology of Evaluation

In this section, we describe our experimental setup including the tools, the test application and the metrics that we used for evaluation.

6.1. Platforms and Tools Used

As it is evident by now, we carry out our experimentations and evaluation for WSNs environment, using mica2 motes as the target devices for the deployment. In accordance with the same, we use TinyOS [69] as the operating environment, nesC [70] as the implementation language and TOSSIM [71] as the simulation environment. TOSSIM does not model the power consumption because it captures TinyOS behaviour at

very low level and cannot provide exact information of CPU energy consumption. Hence, for energy analysis, we use Avrora [72], an instruction level event simulator. Using the results obtained from TOSSIM and Avrora, we evaluate performance of all the ECC based privacy homomorphic encryption algorithms.

Thus, our evaluation is based on a two-step approach:

We use TOSSIM as the WSNs simulator, using that we also obtain the estimates of the storage requirements of the respective implementations.

We determine the energy consumption in Joules using the Avrora [72].

We determine the encryption and decryption time in micro seconds using a microsecond resolution timer for the algorithms.

6.2. Test Application

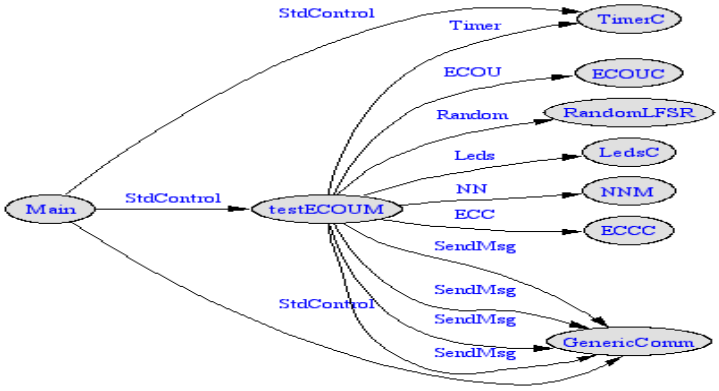

Fig. 1. Flow graph of our application.

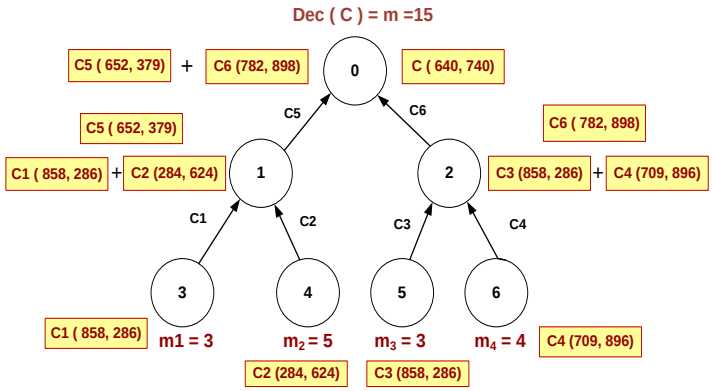

Fig. 2. Our approach for confidentiality.

In Fig. 1, we show the part of a component graph of our application that we create for each ECC based privacy homomorphic encryption algorithms as mentioned before. As shown in the figure, we implement the nesC module testECOUM that in turn performs homomorphic encryption and decryption of the algorithms discussed before. In Fig. 2, we show our sample approach of secure data aggregation supporting confidentiality and privacy using tree topology. In the figure, node 3, 4, 5 and 6 are leaf nodes. Node 1 and 2 are aggregator nodes and node 0 is the base station. In Fig. 3, we show an example of our approach of confidentiality, privacy and integrity preserving secure data aggregation. Node 3, 4, 5 and 6 generate encrypted data and signature on outgoing messages. Then, leaf node 3, 4, 5 and 6 transmit encrypted data, signature and public key to aggregator nodes. Aggregator nodes receive data from child nodes, perform

aggregation on encrypted data and apply summation on public key and signatures. Aggregator node, then transmits these data to the base station. Then, the base station verifies the integrity of the data with the public key received by the aggregator nodes. If it is verified, the base station applies decryption function on the encrypted data and accepts the message.

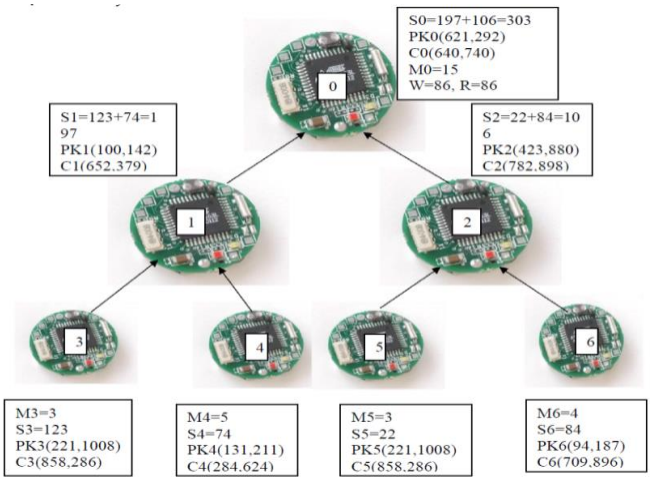

Fig. 3. Our approach for confidentiality, privacy and integrity.

7. Performance Results

In this, we attempt to extend existing TinyECC [12] library of TinyOS [69] and proposes a new framework of secure data aggregation that provides confidentiality, privacy and integrity. In order to do this, we implement *four standard* ECC based privacy homomorphic encryption algorithm, plugged them into TinyECC and try to evaluate these based on the different metrics viz. *Storage requirements* (RAM and ROM), *energy* in joule and *encryption-decryption time* in microseconds. From the results obtained, we integrate best-suited algorithm into tree topology, to execute actual secure data aggregation scenario. Moreover, we know that without the support of data integrity, data aggregation is of no use. Therefore, in addition to support of privacy and confidentiality, we integrate support for data integrity for secure data aggregation with the help of our novel proposed bloom filter based variant of EC-DSA. As we know that sensor nodes of WSNs are working in the resource constrained environment, the algorithm employed in it must be carefully designed to save its energy and increase the lifetime of WSNs. Hence, we use previously mentioned metrics that are directly affecting the lifetime of the sensor nodes to evaluate the performance of privacy homomorphic encryption algorithms. In the next subsection, we show our experimental results for ECC based homomorphic encryption algorithms based on the previously mentioned metrics. After that, we discuss results of various variants of EC-DSA as well as our proposed variant based on three secrets and based on bloom filter. Subsequently, we show experimental results of our framework that integrates bloom filter based EC-DSA and provides confidentiality, privacy and integrity preserving secure data aggregation in the WSNs.

7.1. Evaluation of ECC Based Algorithms

In Table 3, we discuss the results of an empirical evaluation of ECC based homomorphic encryption algorithms compared to EC-IES, default encryption algorithm of TinyECC in TinyOS and thus, shows the support of all ECC based homomorphic encryption algorithms (Extension of TinyECC). From the table, we observe that, EC-EG requires more decryption time and energy consumption and hence it is not suitable for large network. As per our results, we observe that EC-NS requires more storage and more decryption time

compared to other algorithms. S. Peter et al. [9] mentioned EC-P is not as efficient and requires many computations compared to other ECC algorithms. Security of EC-OU is based on the factorization problem; if we select large prime number as a factor in EC-OU then time requires to break EC-OU is much larger than the overall lifetime of WSNs. Therefore, in the WSNs, security provided by the EC-OU is sufficient to achieve SDA. Moreover, as per the results, though EC-OU requires more encryption time, considering other factor like storage and energy consumption, we observe that resource consumption wise EC-OU is the best suited in the resource constraint environment of WSNs compared to other ECC based algorithms. Therefore, by considering all the above points, theoretical analysis, our results and security analysis, we observe that EC-OU can be best suited for the secure data aggregation in WSNs.

Table 3. Results of ECC Based Homormophic Encryption Algorithms Compared to EC-IES

Name of Algorithm	%Decrease in ROM	%Decrease in RAM	%Decrease in Energy	%Increase in Encryption Time	%Increase in Decryption Time
EC-OU	24.306	12.097	4.179	100	0
EC-P	24.889	8.529	3.495	0	100
EC-NS	21.446	-2.175	-80.940	0	400
EC-EG	21.843	14.273	-67.273	0	213500

7.2. Evaluation of Various Variants of EC-DSA Including Our Novel Bloom Filter Based Variant

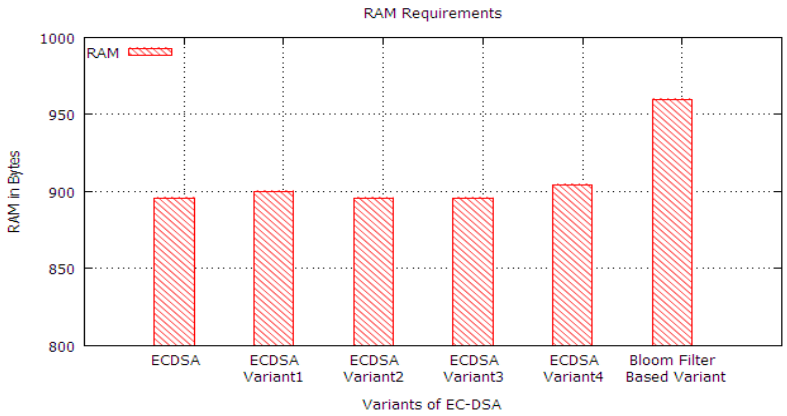

Fig. 4. RAM requirements of various variant of EC-DSA.

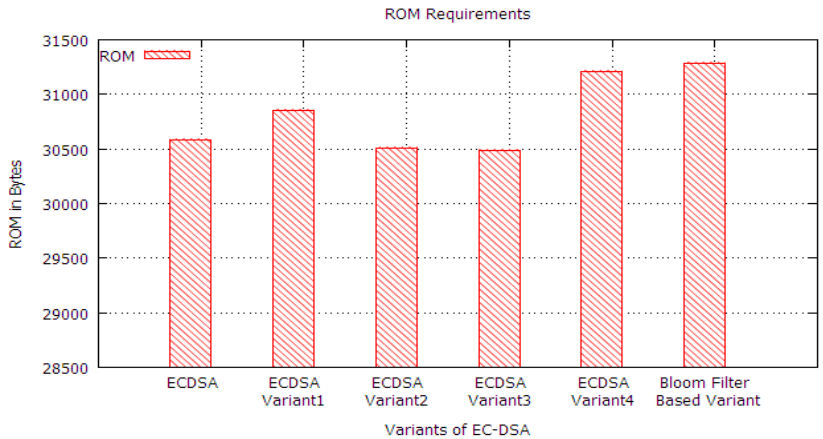

Fig. 5. ROM requirements of various variant of EC-DSA.

In Fig. 4, we show RAM requirements for various variants of EC-DSA. In the figure, ECDSA is the basic EC-DSA, ECDSA1 is limited computation capacity signer, ECDSA2 is limited computation capacity verifier, ECDSA3 is using two different secrets, ECDSA4 is using three different secrets and last is our proposed bloom filter based variant. We observe that our approach using bloom filter requires only 6% more RAM than that of another variant of EC-DSA but that is for the benefit of additional security.

In Fig. 5, we show ROM requirements for various variants of EC-DSA. We observe that our approach using bloom filter requires only 2% more ROM than that of another variant of EC-DSA but that is for the benefit of additional security.

In Fig. 6, we show energy consumption for various variants of EC-DSA. Energy consumption is the average energy consumed by the nodes of WSNs. We observe that our approach using bloom filter requires approximately the same energy compare to the other variant of EC-DSA and that is for the benefit of additional security. In the next subsection, we discuss the results of actual secure data aggregation with various options.

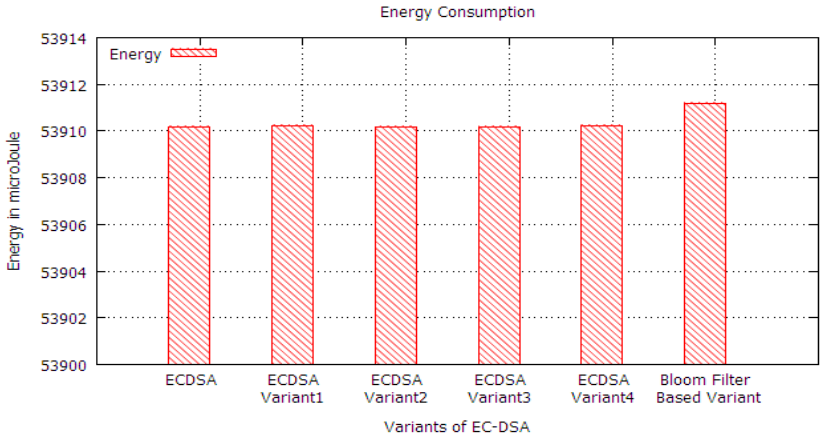

Fig. 6. Energy consumption of various variant of EC-DSA.

7.3. Evaluation of Actual Secure Data Aggregation

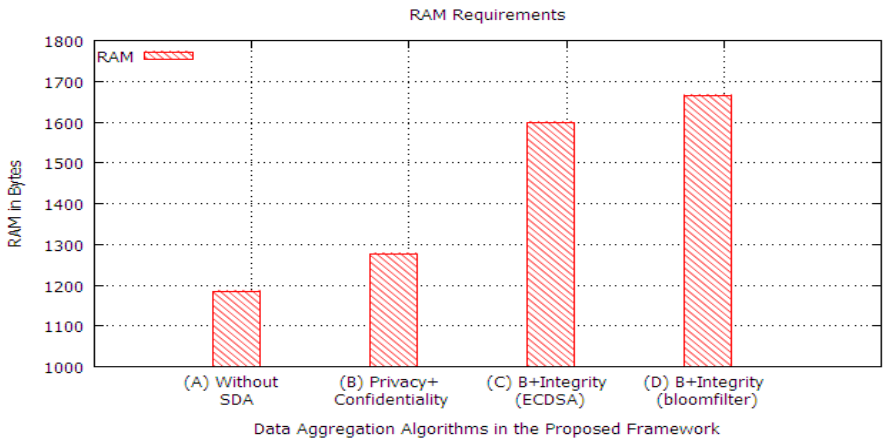

Fig. 7. RAM requirements of various frameworks.

In Fig. 7, we show RAM requirements for various frameworks. A) Without secure data aggregation, B) with privacy and confidentiality preserving secure data aggregation, C) with confidentiality, privacy and an EC-DSA based integrity and D) with confidentiality, privacy and bloom filter based integrity support. We

observe that, our approach using bloom filter requires only 1% more RAM than that of the framework providing integrity with EC-DSA, but that is for the benefit of additional security.

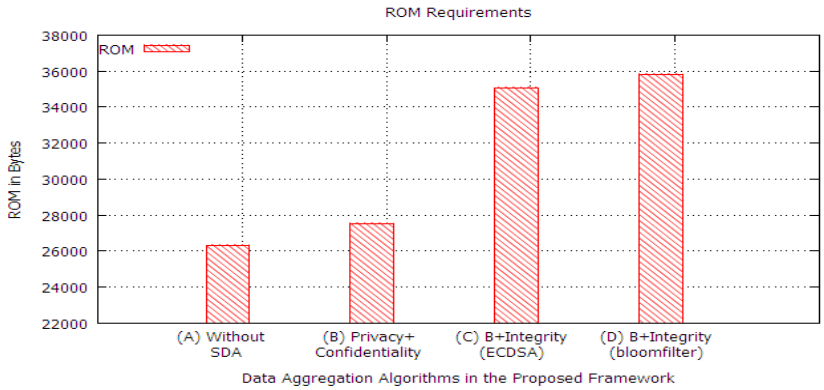

Fig. 8. ROM requirements of various frameworks.

In Fig. 8, we show ROM requirements for various frameworks. We observe that our approach using bloom filter requires only 1% more ROM than that of the framework providing integrity with EC-DSA but that is for the benefit of additional security.

In Fig. 9, we show Energy consumption for various frameworks. Energy consumption is the average energy consumed by the nodes of WSNs for secure data aggregation. We observe that our approach using bloom filter requires only 1% more energy than that of the framework providing integrity with EC-DSA and that is for the benefit of additional security.

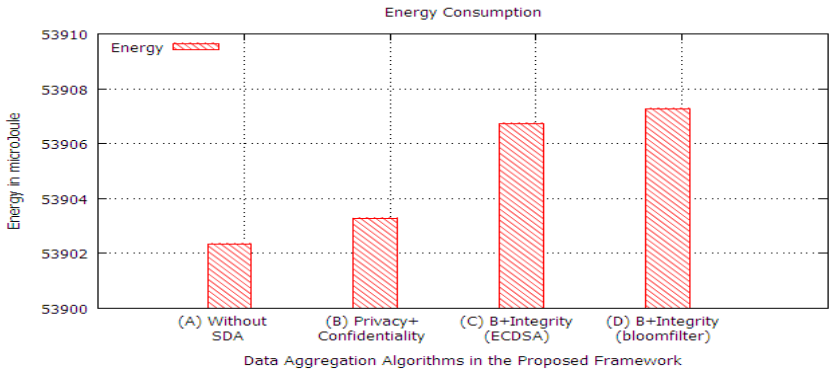

Fig. 9. Energy consumption of various frameworks.

8. Conclusion

In this research exercise, we discuss our attempts at proposing a framework for secure data aggregation that preserves confidentiality, privacy, integrity and authentication in WSNs. In the process, we expand the existing TinyECC library as a first step to provide support for *four* classical homomorphic encryption algorithms. We comparatively evaluate and justify the use of EC-OU for privacy and confidentiality in our framework. To provide the support for integrity, we empirically evaluate existing variants of EC-DSA and show that none of the current variants of EC-DSA are suitable to be used. Hence, we also propose our own variant of EC-DSA that is based on probabilistic set membership data structure viz. Bloom filter. Our results show that our variant of EC-DSA is suitable for any application demanding integrity support in resource constrained environments of WSNs. We also incorporate novel variant based on bloom filter with our

Proposed framework of SDA. Thus, the proposed framework of secure data aggregation provides support for confidentiality, privacy and novel bloom filter based end-to-end integrity.

References

[1] Akyildiz, F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks, 38(4),* 393–422.

[2] Chris, K., Naveen, S., & David, W. (2004). TinySec: A link layer security architecture for wireless sensor networks. *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems* (pp. 162-175).

[3] Mark, L., Ghita, M., Adrian, P., & Virgil, G. (2007). MiniSec: A secure sensor network communication architecture. *Proceedings of the ACM International Conference on Information Processing in Sensor Networks* (pp. 479-488). New York: ACM.

[4] Devesh, J., Dhiren, P., & Dasgupta, K. S. (2009). FlexiSec: A configurable link layer security architecture for wireless sensor networks. *Journal of Information Assurance & Security: Special Issue on Information Assurance and Data Security, 4(6),* 582-603.

[5] Devesh, J., Dhiren, P., & Dasgupta, K. S. (2008). Optimizing the block cipher & modes of operations overhead at the link layer security framework in the wireless sensor networks. *Proceedings of the 4th International Conference on Information Systems Security* (pp. 258-272).

[6] Josep, F. (1996). A new privacy homomorphism and applications. *Inf. Process Lett., 60(5),* 277–282.

[7] Claude, C., Einar, M., & Gene, T. (2005). Efficient aggregation of encrypted data in wireless sensor networks. *Proceedings of the MobiQuitous on IEEE Computer Society* (pp. 109-117).

[8] Einar, M., Joao, G., & Dirk, W. (2006). Public key based cryptoschemes for data concealment in wireless sensor networks. *Proceeding of IEEE International Conference on Communications* (pp. 2288-2295).

[9] Peter, S., Westhoff, D., & Castelluccia, C. (2010). A survey on the encryption of convergecast-traffic with in-network processing. *IEEE Transactions on Dependable and Secure Computing, 7(1),* 20-34.

[10] Ajay, M. (2004). SecureDAV: A secure data aggregation and verification protocol for sensor networks. *Proceedings of the IEEE Global Telecommunications Conference* (pp. 2175–2179).

[11] Yi, Y., Xinran, W., Sencun, Z., & Guohong, C. (2008). Sdap: A secure hop-by-hop data aggregation protocol for sensor networks. *Inf. Syst. Secur., 11(4),* 1–43.

[12] Liu, A., Kampanakis, P., & Ning. P. (2008). TinyECC: Elliptic curve cryptography for sensor networks. *Proceedings of International Conference on Information Processing in Sensor Networks* (pp. 245-256).

[13] Osman, U., Alban, H., & Dirk, W. (2007). Performance of additive homomorphic EC-ElGamal encryption for TinyPEDS. *GI/ITG KuVSFachgespräch, DrahtloseSensornetze, RWTH Aachen.*

[14] Bahi, J. M., Christophe, G., & Abdallah, M. (2010). Efficient and robust secure aggregation of encrypted data in sensor networks. *Proceedings of the 4th International Conference on Sensor Technologies and Applications* (pp. 472-477). Washington: IEEE Computer Society.

[15] Suat, O., & Yang, X. (2011). Integrity protecting hierarchical concealed data aggregation for wireless sensor networks. *Computer Network, 55(8),* 1735-1746.

[16] Wang, X., Li, J., Peng, X., & Zou, B. (2010). Secure and efficient data aggregation for wireless sensor networks. *Proceeding of the 72nd Vehicular Technology Conference Fall (VTC 2010-Fall)* (pp. 1-5).

[17] Poornima, A. S., & Amberker, B. (2010). SEEDA: Secure end-to-end data aggregation in wireless sensor networks. *Proceeding of the 7th International Conference of Wireless and Optical Communications Networks* (WOCN) (pp. 1-5).

[18] Rabindra, B., Kyoung, J., & Chang, J. (2009). A new approach to secure aggregation of private data in wireless sensor networks. *Proceedings of the 8th IEEE International Conference on Dependable*

Autonomic and Secure Computing (pp. 394-399).

[19] Oenen, M., & Molva, R. (2007). Secure data aggregation with multiple encryption. *Proceedings of 4th European Conference on Wireless Sensor Networks.*

[20] He, W., Liu, X., Hoang, N., Klara, N., & Tarek, A. (2007). PDA: Privacy-preserving data aggregation in wireless sensor networks. *Proceedings of the IEEE Infocom* (pp. 2045-2053).

[21] Girao, J., Westhoff, D., & Schneider, M. (2005). CDA: Concealed data aggregation for reverse multicast traffic in wireless sensor networks. *Proceedings of IEEE International Conference on Communications: Vol. 5* (pp. 3044–3049).

[22] Li, H., Lin, K., & Li, K. (2011). Energy-efficient and high-accuracy secure data aggregation in wireless sensor networks. *Journal of Comput. Commun*, *34(4),* 591-597.

[23] Huang, X., Yang, M., & Tong, Y. (2007). An efficient and secure aggregation of encrypted data for wireless sensor network based on dynamic cluster. *Proceedings of the Spring Simulation Multi International Conference Society for Computer Simulation*: *Vol. 1* (pp. 51-57).

[24] Hasan, Ç., Suat, Ö., Prashant, N., Devasenapathy, M., & OzgurSanlia, H. (2006). Energy-efficient secure pattern based data aggregation for wireless sensor networks. *Journal Computer Communications Archive, 29(4).*

[25] Suat, O., & Yang, X. (2009). Hierarchical concealed data aggregation for wireless sensor networks. *Proceedings of Embedded Systems and Communications Security Workshop in conjunction with IEEE SRDS* (pp. 27-29).

[26] Ajay, J. P. (2013). Cb-SDA: Cluster-based secure data aggregation for private data in wsn. *Wireless and Mobile Technologies, 1(1),* 37-41.

[27] Wang, T., Qin, X., & Liu, L. (2013). An energy-efficient and scalable secure data aggregation for wireless sensor networks. *International Journal of Distributed Sensor Networks*.

[28] Julia, A., & Sanjay, M. (2009). Secure hierarchical data aggregation in wireless sensor networks. *Proceeding of the 2009 IEEE Conference on Wireless Communication and Networking* (pp. 2420-2425).

[29] Suat, O. (2008). Functional reputation based reliable data aggregation and transmission for wireless sensor networks. *Journal of Computer Commun, 31(17),* 3941-3953.

[30] Alzaid, H., Foo, E., Nieto, J., & Park, D. (2008). A taxonomy of secure data aggregation in wireless sensor networks. *International Journal of Communication Networks and Distributed Systems, 8(1),* 101–148.

[31] Zhang, W., Sajal, K. D., & Liu, Y. (2006). A trust based framework for secure data aggregation in wireless sensor networks. *Proceeding of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*: *Vol. 1* (pp. 60-69).

[32] Bartosz, P., Dawn, S., & Adrian, P. (2007). SIA: Secure information aggregation in sensor networks. *Journal of Computer Security – Special Issue on Security of Ad-hoc Networks*, *15(1),* 69-102.

[33] Hu, L., & David, E. (2003). Secure aggregation for wireless networks. *Proceedings of the 2003 Symposium on Applications and the Internet Workshops*. IEEE Computer Society, Washington: USA.

[34] Kui, W., Dennis, D., Bo, S., & Yang, X. (2006). Secure data aggregation without persistent cryptographic operations in wireless sensor networks. *Proceedings of the 25th IEEE International Performance Computing and Communications Conference, IPCCC 2006*.

[35] Du, W., Deng, J., Han, Y., & Varshney, P. K. (2003). A witness-based approach for data fusion assurance in wireless sensor networks. *Proceedings of the IEEE Global Telecommunications Conference* (pp. 1435-1439).

[36] Bagaa, M., Challal, Y., Ouadjaout, A., Lasla, N., & Badache, N. (2012). Efficient data aggregation with in network integrity control for wsn. *Journal of Parallel and Distributed Computing, 72(10),* 1157–1170.

[37] Shaik, M., & Subbarao, K. (2014). Secure data aggregation in wireless networks. *International Journal of*

Research in Computer and Communication Technology, 3(1), 87-93.

[38] Vimal, K., & Sanjay, M. (2010). Performance analysis of secure hierarchical data aggregation in wireless sensor networks. *Proceedings of the Mobile Data Management* (pp. 299-300).

[39] Paillier, P. (2000). Trapdooring discrete logarithms on elliptic curves over rings. *Proceedings of Annual International Conference on Theory and Application of Cryptology and Information Security* (pp. 573-584). London: Springer-Verlag.

[40] Liao, H., & Shen, Y. (2006). On the elliptic curve digital signature algorithm. *Tunghai Science, 8*, 109–126.

[41] Hu, J. (2011). The improved elliptic curve digital signature algorithm. *Proceedings of International Conference on Electronic & Mechanical Engineering and Information Technology* (pp. 257-259).

[42] Bloom, B. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM, 13(7)*, 422-426.

[43] Lawrence, C. J., & Mark, W. (1979). Universal classes of hash functions. *Journal of Computer and System Sciences, 18,* 143-154.

[44] Zhang, M., Chan, M. C., & Akkihebbal, L. A. (2008). Connectivity monitoring in wireless sensor networks. *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems* (pp. 69–79).

[45] Peter, H., & Adrian, R. P. (2006) Data-centric routing using bloom filters in wireless sensor networks. *Proceedings of 4th International Conference on Intelligent Sensing and Information Processing, ICISIP* (pp. 72-77).

[46] Li, Z., & Gong, G. On data aggregation with secure bloom filter in wireless sensor networks. *Technical Report Citeseer.*

[47] Chen, T., Guo, D., He, Y., Chen, H., Liu, X., & Luo, X. (2013) A bloom filters based dissemination protocol in wireless sensor networks. *Ad Hoc Networks, 11(4)*, 1359-1371.

[48] Nidal, Q., & Ramiro, L. (2013) Bloom filter supporting distributed policy-based management in wireless sensor networks. *Procedia Computer Science, 19*, 248-255.

[49] Tong, E., Niu, W., Li, G., Tang, D., Chang, L., Shi, Z., & Song, C. (2013) Bloom filterbased workflow management to enable qos guarantee in wireless sensor networks. *Journal of Network and Computer Applications.*

[50] Peter, Y. (2013). *Randu: A Random Number Generator.*

[51] George, M. Random number generation. *Encyclopedia of Computer Science*, 1499-1503. UK: John Wiley and Sons Ltd., Chichester.

[52] Domingo, F., & Joseph. (2002). A provably secure additive and multiplicative privacy homomorphism. *Proceedings of the 5th International Conference on Information Security* (pp. 471–483). London: Springer-Verlag.

[53] Peter, S., Langendo¨ rfer, P., & Piotrowski, K. (2007). On concealed data aggregation for wireless sensor networks. *Proceedings of the 4th IEEE Consumer Comm. and Networking Conf. (CCNC).*

[54] Rivest, R., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21(2),* 120–126.

[55] Goldwasser, S., & Micali, S. (1984). Probabilistic encryption. *Journal of Computer and System Sciences, 28(2),* 270–299.

[56] Benaloh, J. (1994). Dense probabilistic encryption. *Proceedings of the Workshop on Selected Areas of Cryptography* (pp. 120–128).

[57] Okamoto, T., & Uchiyama, S. (1998). A new public-key cryptosystem as secure as factoring. *EUROCRYPT,* 308–318.

[58] ElGamal, T. (1984). A public key cryptosystem and a signature scheme based on discrete lograithms. *Proceedings of CRYPTO 84, Advances in Cryptology, Lecture Notes in Computer Science* (pp. 10–18).

California: Springer-Verlag.

[59] Pascal, P. (1999). Public-Key cryptosystems based on composite degree residuosity classes. *EUROCRYPT,* 223-238.

[60] Stallings, W. (2006). *Cryptography and Network Security: Principles and Practice* (4th ed.). New Jersey: Prentice Hall Inc.

[61] Vivaksha, J., & Devesh, J. (2011). Evaluating homomorphic encryption algorithms for privacy in wireless sensor networks. *International Journal of Advancements in Computing Technology, 3(6)*, 215-223.

[62] Hankerson, D., Menezes, A., & Vanstone, S. (2004). *Guide to Elliptic Curve Cryptography*. Springer.

[63] Wang, W., Lin, Y., & Chen, T. (2008). The study and application of elliptic curve cryptography library on wireless sensor network. *Proceedings of the 11th IEEE International Conference on Communication Technology ICCT 2008* (pp. 785-788).

[64] Don, J., Alfred, M., & Scott, V. (2001). The elliptic curve digital signature algorithm (ECDSA). *International Journal of Information Security, 1(1)*.

[65] SEC 3, Standards for efficient cryptography. *Elliptic Curve Signatures Giving Partial Message Recovery.* Retrieved from http://www.secg.org

[66] Aqeel, K., Kuldip, S., & Sandeep, S. (2010). Implementation of elliptic curve digital signature algorithm. *International Journal of Computer Applications, 2(2),* 21–27.

[67] *Technical Guideline TR-03111Elliptic Curve Cryptography Version 2.0.Bundesamt fur Sicherheit in der Informationstechnik.*

[68] Zhang, Q., Li, Z., & Song, C. (2011). The improvement of digital signature algorithm based on elliptic curve cryptography. *Proceedings of Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)* (pp. 1689-1691).

[69] Hill, J., *et al.* (2000). System architecture directions for networked sensors. *Proceedings of the 9thIntl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000)* (pp. 93-104). New York: ACM Press.

[70] David, G., Phil, L., Rob von, B., Matt, W., Eric, B., & David, C. (2003). The nesC language: A holistic approach to network embedded systems. *Proceedings of Programming Language Design and Implementation (PLDI)* (pp. 1-11). New York: ACM.

[71] Philip, L., & Lee, N. (2003). *TOSSIM: A simulator for TinyOS Networks Version 1.0.* Berkeley.

[72] Ben L. T., Lee, D., & Jens, P. (2005). Avrora: Scalable sensor network simulation with precise timing. *Proceedings of the 4th Intl. Conf. on Information Processing in Sensor Networks (IPSN)* (pp. 477-482).

Vivaksha Jariwala is an associate professor in Information Technology Deparment at Sarvajanik College of Engineering and Technology, Surat. She completed her Ph.D. in comuter engineering from Sardar Vallabhbhai National Institute of Technology (SVNIT) Surat in September, 2014. She completed her M.Tech. from Sardar Vallabhbhai National Institute of Technology (SVNIT) Surat in May 2009. She completed her B.E. in computer engineering in 2002. Her major research area includes information security, computer netwoks, wireless sensor networks and software engineering. She has more than 15 international and national publications. Vivaksha is a member of professional societies like CRSI, ISTE, CSI, ACM and IEI.