International Journal of Computer Network and Security (IJCNS)
Vol 6. No.1 – Jan-March 2014 Pp. 51-58
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0975-8283

MULTIPLE SENSOR APPLICATIONS USING SECURE DATA AGGREGATION

Mr.S.M.Bharathi, Mr.D. Durai kumar,
Department of Information Technology,
Ganadipathy Tulsi's Jain Engineering College,
Kaniyambadi – 632 102.
bharathism31@gmail.com

ABSTRACT

A wireless sensor network, data aggregation scheme that reduces a large amount of transmission is the most practical technique. In earlier studies, Homomorphic encryptions have been applied to conceal communication during aggregation such that enciphered data can be aggregated algebraically without decryption. Since aggregators collect data without decryption, adversaries are not able to forge aggregated results by compromising them. However, these schemes are not satisfy multi-application environments. Second, these schemes become insecure in case some sensor nodes are compromised. Third, these schemes do not provide secure counting; thus, they may suffer unauthorized aggregation attacks. Therefore, I use a new concealed data aggregation scheme extended from Boneh et al.'s Homomorphic public encryption system. The proposed scheme has three contributions. First, it is designed for a multi-application environment. The base station extracts application-specific data from aggregated cipher texts. Next, it mitigates the impact of compromising attacks in single application environments. Finally, it degrades the damage from unauthorized aggregations. To prove the proposed scheme's robustness and efficiency, I also include the timer which is used to give alert message to base station prevent from attackers.

Index Terms— Base stations, Concealed data aggregation, homomorphic encryption, wireless sensor networks

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of thousands of sensor nodes (SN) that gather data from deployed environments. There are plenty of rich applications proposed for WSNs, such as environment monitoring, accident reporting, and military investigation [1]. Depending on the purpose of each application, SN are customized to read different kinds of data (e.g., temperature, light, or smoke). Typically, SN are restricted by the resources due to limited computational power and low battery supply; thus, energy saving technologies must be considered when we design the protocols. For better energy utilization, cluster-based WSNs have been proposed. In cluster-based WSNs [2], SN resident in nearby area would form a cluster and select one among them to be their cluster head (CH). The CH organizes data pieces received from SN into an aggregated result, and then forwards the result to the base station based on regular routing paths. Generally, aggregative operations are algebraic or statistical operation. Although data aggregation could significantly reduce transmission, it is vulnerable to some attacks. For instance, compromising a CH will allow adversaries to forge aggregated results [4] as similar as compromising all its cluster members. To solve this problem, several studies, such as the delay aggregation have been proposed.

In this proposed scheme, called CDAMA, provides CDA between multiple groups. Basically, CDAMA is a modification from Boneh et al.'s [13] PH scheme Here, I also suppose three practical application scenarios for CDAMA, all of which can be realized by only CDAMA. The first scenario is designed for multi-application WSNs. In practice, SN having different purposes, e.g., smoke alarms and thermometer sensors may be deployed in the same environment. The second scenario is designed for single application WSNs. Compared with conventional schemes [9], [10], [11], [12]; CDAMA mitigates the impact of compromising SN through the construction of multiple groups. The last scenario is designed for secure counting capability. In previous schemes, the base station does not know how many messages are aggregated from the decrypted aggregated result. In CDAMA, the base station exactly knows the number of messages aggregated to avoid above attacks.

2 SYSTEM MODEL

Here, I state two models for further uses, aggregation model and attack model. The aggregation model defines how aggregation works; the attack model defines what kinds of attacks a secure data aggregation scheme should protect from.

WSN Setup

In this phase I have to set up the WSN environment by designing base station, Aggregators and multiple sensors. The communications between them also have to establish. Group Public and Private Key established that keys are known by Application sensors and Base station.

Key Distribution

It briefly explains how to deliver the group public keys to SNs securely. There are two main approaches.

Key pre-distribution

If I know the locations of deployed Sensor nodes (SN), I can preload necessary keys and functions into SNs and Aggregators (AG) so that they can work correctly after being spread out over a geographical region.

Key post-distribution

Before SNs are deployed to their geographical region, they are capable of nothing about CDAMA keys. These SNs only load the key shared with the BS prior to their deployment, such as the individual key and the master secret key. Once these SNs are deployed, they can run the LEACH protocol to elect the AGs and construct clusters. After that, the BS sends the corresponding CDAMA keys, encrypted by the pre-shared key, to SNs and AGs.

Aggregation Model

In WSNs, SN collect information from deployed environments and forward the information back to base station (BS) via multihop transmission based on a tree or a cluster topology.

The accumulated transmission carries large energy cost for intermediate nodes. To increase the lifetime, tree-based or cluster networks force the intermediate nodes (a sub tree node or a cluster head) to perform aggregation, i.e., to be aggregators (AG). After aggregation done, AGs would forward the results to the next hop. In general, the data can be aggregated via algebraic operations (e.g., addition or multiplication) or statistical operations (e.g., median, minimum, maximum, or mean). For example, an AG can simply forward the sum of numerical data received instead of forwarding all data to the next hop.

Attack Model

First of all, I categorize the adversary's abilities as follows:
1. Adversaries can eavesdrop on transmission data in a WSN.
2. Adversaries can send forged data to any entities in a WSN (e.g., SN, AG, or BS).
3. Adversaries can compromise secrets in SN s or AGs through capturing them.

Second, I define the following attacks to qualify the security strength of a CDA scheme. Part of these attacks refers to Peter et al.'s analysis [15]. Based on adversary's abilities and purposes, I further classify these attacks into three categories.
In the first category A, an adversary wants to deduce the secret key (i.e., decrypting arbitrary ciphertexts). Category A consists of four attacks that are commonly used in qualifying an encryption scheme. In practice, the first two attacks are feasible in WSNs [15]. Here, I use them to qualify the underlying homomorphic encryption schemes. In category B, an adversary wants to send the forged messages to cheat the BS even though she does not know the secret key. This category consists of two attacking scenarios based on specific features deriving from PH schemes. The last category C consists of three attacks and considers the impact of node compromising attacks. The first attack is the case of compromising an AG, and the last two attacks are cases of compromising an SN. I discuss them separately because they store different secrets in the PH schemes.

A1.Ciphertext only attack. An adversary can deduce the key from only the encrypted messages.
A2.Known plaintext attack. Given some samples of plaintext and their ciphertext, an adversary can deduce the key or decrypt any ciphertext.
A3. Chosen plaintext attack. Given some samples of chosen plaintext and their ciphertext, an adversary can deduce the key or decrypt any ciphertext.
A4.Chosen cipher text attack. Given some samples of chosen ciphertext and their plaintext, an adversary can deduce the key or decrypt any ciphertext she has

International Journal of Computer Network and Security (IJCNS)
Vol 6. No.1 – Jan-March 2014 Pp. 51-58
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0975-8283

not chosen before. The model is CCA1, also called lunchtime attacks.

B1. Unauthorized aggregation. An adversary can aggregate sniffed ciphertexts into forged but format-valid ciphertexts.

B2. Malleability. An adversary can alter the content of a ciphertext.

C1. B1/B2 under compromised AG. When an adversary captures an AG and compromises its secret, she can use it to launch B2/B3 attacks with higher probability of success.

C2.Unauthorized decryption under compromised SN. When an adversary captures an SN and compromises its secret, she can decrypt not only the ciphertexts from that SN but also the ciphertexts from the other remaining SN s. Asymmetric schemes can defend against unauthorized decryption under compromised secrets because knowing the public key is useless for decryption.

C3.Unauthorized encryption under compromised SN. When an adversary captures an SN and compromises its secret, she can impersonate not only that SN but also the other remaining SN s to generate legal ciphertexts.

3 PRELIMNARIES

Privacy Homomorphic Cryptosystem

Privacy homomorphic encryption (PH) is an encryption scheme with homomorphic property. The homomorphic property implies that algebraic operations on plaintexts can be executed by manipulating the corresponding ciphertexts; for instance, $D_K (E_K (m_1) \odot E_K (m_2)) = m1 \oplus m2$, where E_K (.) is the encryption with key K, D_K (.) is the decryption with key K, and $\odot$ and $\oplus$ denote operations on ciphertexts and plaintexts, respectively. In general, operations and can be addition, multiplication, and so on.Similar to conventional encryption schemes, PH schemes are classified to symmetric cryptosystem when the encryption and decryption keys are identical, or asymmetric cryptosystem when the two keys are different. Symmetric PH schemes, Castelluccia et al.'s scheme, usually are more competitive in terms of efficiency than asymmetric schemes. The most notable asymmetric PH schemes are based on elliptic curve cryptography (ECC). Compared with RSA cryptosystems, ECC provides the same security with a shorter key size and shorter ciphertexts. A 160-bit ECC cryptosystem provides the same security as a 1,024-bit RSA cryptosystem. In energy- constraint WSNs, constructing PH via ECC is more efficient

CDA Based on PH

Conventional hop-by-hop aggregation schemes are insecure because an adversary is able to forge aggregated results such as compromising all the AG's child nodes when he compromises the secret of an AG. To diminish this impact, PH schemes have been applied to WSNs [9], [10], [11], [12], [14]. By PH schemes, SN s encrypts their sensed readings and allows AGs to homomorphically aggregate their ciphertexts without decryption. Therefore, compromising AGs earns no advantage of forging aggregated results.

Westhoff et al. [9] and Girao et al. [10] proposed CDA based on symmetric PH to facilitate the aggregation of encrypted data. In contrast to symmetric PH construction, Mykletun et al. [11] Adopted public-key-based PH to construct their systems, and Girao et al. [12] extended the ElGamal PH encryption to construct an aggregation scheme. In these schemes, because all SN in a network only share a common key for encryption [9], [10], [11], [12], an adversary can forge the aggregated results by simply compromising one SN . To solve this problem, Castelluccia et al. [14] proposed an encryption scheme similar to one- time pad. In each transmission, individual SN generates temporary key from a pseudo random number generator (PRNG) and adds its messages with the key under modulation. The AG aggregates those ciphertexts through modular addition. And the BS decrypts the ciphertext received by modular subtraction with all the temporal keys. If an adversary tries to forge aggregated results, he must compromise all SN s. However, their scheme cannot prevent the adversary from injecting forged data packets into the legitimate data flow. In addition, key synchronization must be guaranteed because each SN must rekey after each encryption.

BGN Scheme

In 2006, Boneh et al. [13] proposed a public-key PH scheme, which integrates the Paillier with the Okamoto- Uchiyama encryption schemes.I call it, BGN for simplicity. BGN provides additive and multiplicative homomorphism. Since the multiplicative property, based on the bilinear pairing [13], is much expensive and inefficient for SN s, we only utilize the additive homomorphism of BGN. It provides a possible application for BGN, data aggregation. Furthermore, I modify BGN to fit multigroup construction or stronger security and better applicability.

International Journal of Computer Network and Security (IJCNS)
Vol 6. No.1 – Jan-March 2014 Pp. 51-58
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0975-8283

4 SYSTEM ARCHITECTURE

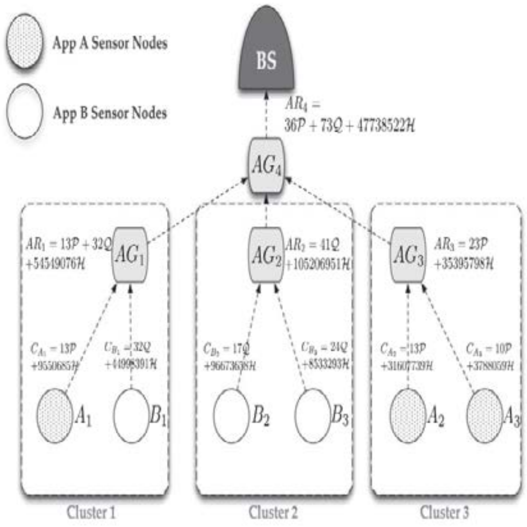

App A Sensor Nodes
App B Sensor Nodes
BS
$AR_4 =$ $36P + 73Q + 47738522H$
AG_4
$AR_1 = 13P + 32Q$ $+54549076H$
AG_1
AG_2
$AR_2 = 41Q$ $+1052069951H$
AG_3
$AR_3 = 23P$ $+35395798H$
$C_{A_1} = 13P$ $+95506853H$
$C_{B_1} = 32Q$ $+41998391H$
$C_{B_2} = 17Q$ $+966736398H$
$C_{B_3} = 24Q$ $+8533293H$
$C_{A_2} = 13P$ $+316077339H$
$C_{A_3} = 10P$ $+3788059H$
A_1
B_1
B_2
B_3
A_2
A_3
Cluster 1
Cluster 2
Cluster 3

Figure-1. System Architecture

Bs -> Base Station
AG1, AG2, AG3 -> Sub Aggregator or Cluster Head (CH)
AG4 -> Main Aggregator

When the scenario of multiple applications working concurrently is more realistic in most cases. Study indicates that deploying multiple applications in a shared WSN can reduce the system cost and improve system flexibility. The reason is because an SN supports multiple applications and can be assigned to different applications dynamically. For instance, three different kinds of SNs, smoke detectors, temperature collectors, and light detectors, are deployed in the same building. Each room contains an AG and some SNs. A big challenge for the AGs, $AG_1$to AG_4, is to aggregate the sensed readings from the different applications to a mixed aggregated result. Unfortunately, two limitations make the aggregation more difficult:

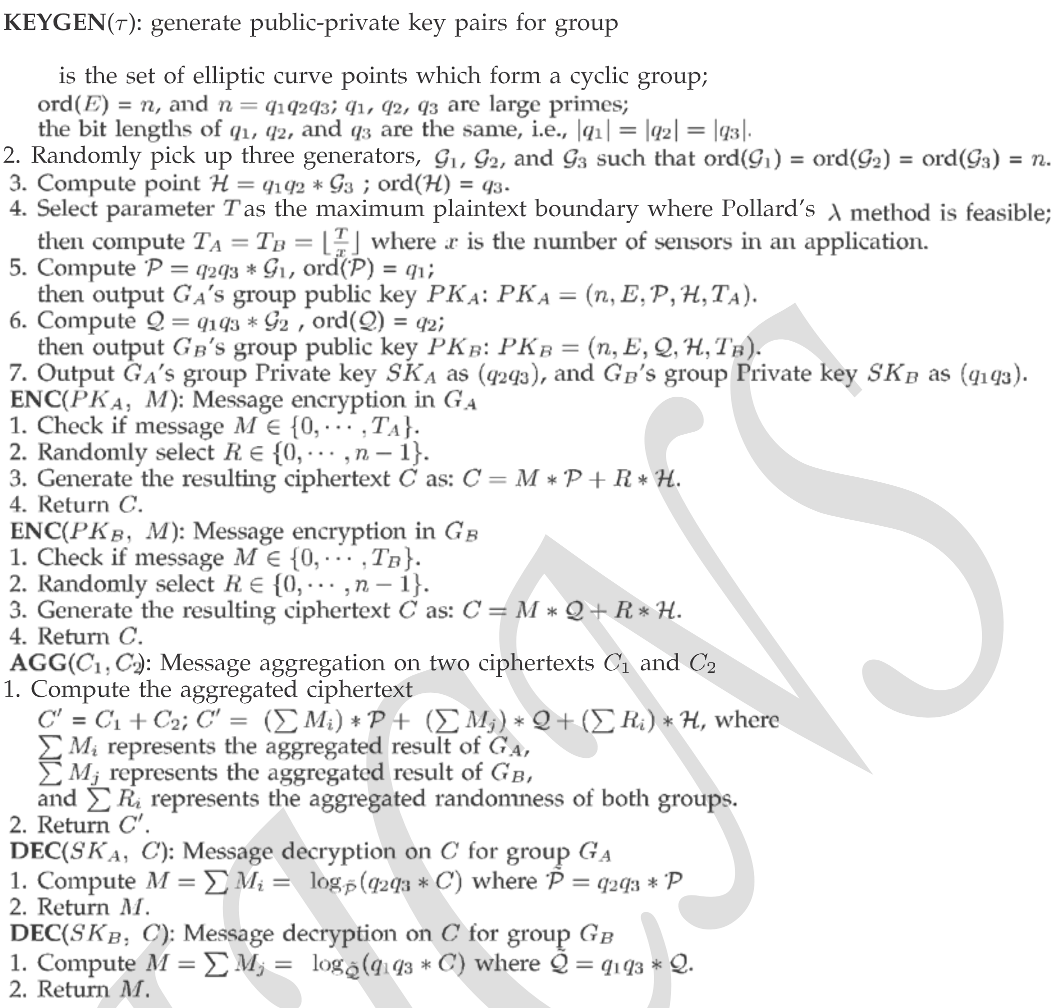
1. To maintain data privacy and reduce the communication overhead, sensed reading should be encrypted by SNs and the corresponding cipher- texts must be aggregated. The solution satisfying this requirement has already been proposed, called CDA.
2. Even if aggregation on cipher texts is possible, aggregation of multi-application is still hard because the decryption cannot extract application-specific aggregated result from a mixed cipher text.

5 CDAMA

BGN is implemented by using two points of different orders so that the effect of one point can be removed by multiplying the aggregated ciphertext with the order of the point, and then the scalar of the other point can be obtained. Based on the same logic of BGN, CDAMA is designed by using multiple points, each of which has different order. I can obtain one scalar of the specific point through removing the effects of remaining points (i.e., multiplying the aggregated ciphertext with the product of the orders of the remaining points). The security of CDAMA and BGN are based on the hardness assumption of subgroup decision problem, whereas CDAMA requires more precise secure analysis for parameter selections, discussing. I use CDAMA (k = 2) to explain how it works in multiple groups.

CDAMA (k=2) Construction

Assume that all SN s are divided into two groups, G_A and G_B. CDAMA contains four procedures: Key generation, encryption, aggregation, and decryption, listing in Fig. 2. As we can see, CDAMA (k = 2) is implemented by using three points P, Q, and H whose orders are q_1 , q_2 , and q_3 , respectively. The scalars of the first two points carry the aggregated messages in G_A and G_B , respectively, and the scalar of the third point carries randomness for security. As shown in the DEC functions, by multiplying the aggregated ciphertext with q_2 q_3 (i.e., SK in G_A), the scalar of the point P carrying the aggregated message in G_A can be obtained. Similarly, by multiplying the aggregated cipher- text with q1 q_3 (i.e., the SK in G_B), the scalar of the point Q carrying the aggregated message in G_B can be obtained. In this way, the encryptions of messages of two groups can be aggregated to a single ciphertext, but the aggregated message of each group can be obtained by decrypting the ciphertext with the corresponding SK. Considering deployment, the private keys should be kept secret and only known by the BS. SN s in the same group shares the same public key and no other entities outside the group know the group public key. Another major change is the decryption procedure. By performing individual decryption, the BS extracts individual aggregated results of different groups from an aggregated ciphertext.



```
KEYGEN($\tau$): generate public-private key pairs for group

    is the set of elliptic curve points which form a cyclic group;
  ord($E$) = $n$, and $n = q_1q_2q_3$; $q_1$, $q_2$, $q_3$ are large primes;
  the bit lengths of $q_1$, $q_2$, and $q_3$ are the same, i.e., $|q_1| = |q_2| = |q_3|$.
2. Randomly pick up three generators, $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$ such that ord($\mathcal{G}_1$) = ord($\mathcal{G}_2$) = ord($\mathcal{G}_3$) = $n$.
3. Compute point $\mathcal{H} = q_1q_2 * \mathcal{G}_3$ ; ord($\mathcal{H}$) = $q_3$.
4. Select parameter $T$ as the maximum plaintext boundary where Pollard's $\lambda$ method is feasible;
   then compute $T_A = T_B = \lfloor \frac{T}{x} \rfloor$ where $x$ is the number of sensors in an application.
5. Compute $\mathcal{P} = q_2q_3 * \mathcal{G}_1$, ord($\mathcal{P}$) = $q_1$;
   then output $G_A$'s group public key $PK_A$: $PK_A = (n, E, \mathcal{P}, \mathcal{H}, T_A)$.
6. Compute $\mathcal{Q} = q_1q_3 * \mathcal{G}_2$ , ord($\mathcal{Q}$) = $q_2$;
   then output $G_B$'s group public key $PK_B$: $PK_B = (n, E, \mathcal{Q}, \mathcal{H}, T_B)$.
7. Output $G_A$'s group Private key $SK_A$ as $(q_2q_3)$, and $G_B$'s group Private key $SK_B$ as $(q_1q_3)$.
ENC($PK_A$, $M$): Message encryption in $G_A$
1. Check if message $M \in \{0, \cdots, T_A\}$.
2. Randomly select $R \in \{0, \cdots, n-1\}$.
3. Generate the resulting ciphertext $C$ as: $C = M * \mathcal{P} + R * \mathcal{H}$.
4. Return $C$.
ENC($PK_B$, $M$): Message encryption in $G_B$
1. Check if message $M \in \{0, \cdots, T_B\}$.
2. Randomly select $R \in \{0, \cdots, n-1\}$.
3. Generate the resulting ciphertext $C$ as: $C = M * \mathcal{Q} + R * \mathcal{H}$.
4. Return $C$.
AGG($C_1, C_2$): Message aggregation on two ciphertexts $C_1$ and $C_2$
1. Compute the aggregated ciphertext
   $C' = C_1 + C_2$; $C' = (\sum M_i) * \mathcal{P} + (\sum M_j) * \mathcal{Q} + (\sum R_i) * \mathcal{H}$, where
   $\sum M_i$ represents the aggregated result of $G_A$,
   $\sum M_j$ represents the aggregated result of $G_B$,
   and $\sum R_i$ represents the aggregated randomness of both groups.
2. Return $C'$.
DEC($SK_A$, $C$): Message decryption on $C$ for group $G_A$
1. Compute $M = \sum M_i = \log_{\hat{\mathcal{P}}}(q_2q_3 * C)$ where $\hat{\mathcal{P}} = q_2q_3 * \mathcal{P}$
2. Return $M$.
DEC($SK_B$, $C$): Message decryption on $C$ for group $G_B$
1. Compute $M = \sum M_j = \log_{\hat{\mathcal{Q}}}(q_1q_3 * C)$ where $\hat{\mathcal{Q}} = q_1q_3 * \mathcal{Q}$.
2. Return $M$.
```

Figure-2. Procedures of CDAMA (k =2).

A Concrete Example

Now, I use an instance to describe how CDAMA (k = 2) works. In WSN consists of six SN s and four AGs. After deployments, they form three clusters. Each SN belongs to either application A or B. Without loss of generality, sensors A_1, A_2, and A3 perform application A and keep the public key PK_A = (n, E, P, H, and T_A). The others, B_1, B_2 and B3 keep PKB = (n, E, Q, H, T_B). Four aggregators, AG1 to AG4 are deployed to gather messages from their child nodes. To simplify the example, we set the order of P, Q, and H to small numbers.We assume that $|q_1| = |q_2| = |q_3| = 10$, e.g., ord(P) = q_1= 521, ord(Q)=q_2 =523,ord (H)= q_3= 541, and n= q_1 q_2 q_3 =147; 413; 303,where $|q_i|$ is the bit size of q_i . Moreover, we assume T=128 and x=3 Such that the maximal sensed value in both applications is at most 42 (i.e., $T_A = T_B = 42$).

We assume the messages of these sensors are MA1 =13, MA_2 = 21, MA_3 = 10, MB_1 =32, MB2 = 17, and MB_3 =24. They are encrypted to the corresponding ciphertexts. After the aggregation by the AGs, the BS receives the final aggregated result AR_4 whose value is 36P+ 73Q+ 195, 121, 825H=36P+73Q+ 477, 385, 22H. The aggregated result in application A, M_A =M_1 +M_2 +M_3 =36 can be obtained by decrypting AR_4 using SK_A in the following steps:

1. Compute q_2 q_3 *AR_4 =282943*(36P + 73Q+ 477, 385, 22H) = 101, 859, 48P = 398P, where 521P=523Q = 541H =∞.
2.MA=$\log_p$(q2q3*AR_4)=log398p,wherep=q2q3*p=40p(mod521)and 521p=∞.Since M_A=$\log_p$398p,we infer that M_A*(40p)=398p(mod521)
3. Finally, through Pollard's _ method, MA =36 can be obtained by the BS.
Similarly, the BS can extract the aggregated result MB in application B by computing the discrete logarithm of q1q3 *AR4 to the base point ~Q = q1q3 *Q.

6 APPLICATIONS

In this section, I propose three applications that are realized by only CDAMA multi group construction.

Multi-Application WSNs

Compared with the multi-application WSNs, the scenario of a single application is more commonly discussed in WSNs. However, the scenario of multiple applications working concurrently is more realistic in most cases. It indicates that deploying multiple applications in a shared WSN can reduce the system cost and improve system flexibility. The reason is because an SN supports multiple applications and can be assigned to different applications dynamically.

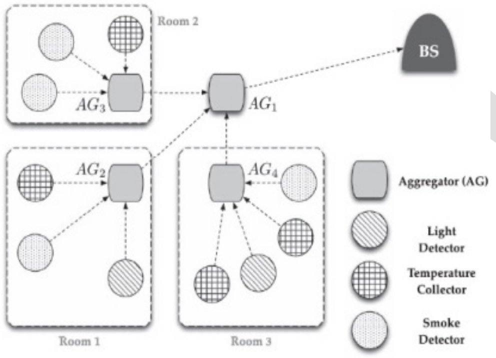


Figure-3. A multi-application WSN example

For instance, three different kinds of SN s, smoke detectors, temperature collectors, and light detectors, are deployed in the same building. Fig. 3 shows this typical case. Each room contains an AG and some SNs. A big challenge for the AGs, AG_1 to AG_4 , is to aggregate the sensed readings from the different applications to a mixed aggregated result.

Conventional Aggregation Model with Multiple Groups

Interestingly, applying CDAMA to the conventional aggregation model can mitigate the impact from compromising attacks. In Fig. 4, all SN s are in the same application, e.g., fire alarm, but they can be arranged into two groups through CDAMA construction. Each group could be assigned a distinct group public key. Once an adversary compromised a SN in group A; it only reveals P K_A , not P K_B . Since the adversary can only forge messages in group A, not group B, the SN s in group B can still communicate safely.

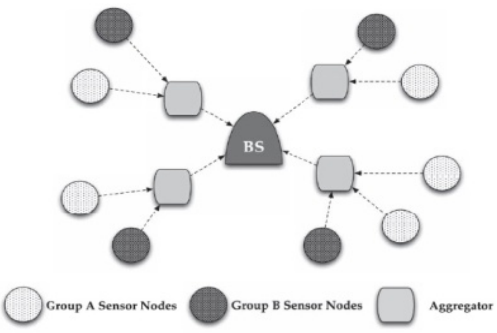


Figure-4. Two groups for a single application

The ideal case is that CDAMA assigns every node for its own group, resulting in the strongest security CDAMA ever offered. However, this is impractical because the size of ciphertext becomes extremely large when we construct groups with a huge group number. Thus, assigning a reasonable number of groups for a single application not only keeps the overhead acceptable but also mitigates the impact of compromising attacks.

Aggregation with Secure Counting

The main weakness of asymmetric CDA schemes is that an AG can manipulate aggregated results without encryption capability. An AG is able to increase the value of aggregated result by aggregating the same ciphertext of sensed reading repeatedly, or decrease the value by selective aggregation. Since the BS does not know the exact number of ciphertexts aggregated, repeated or selective aggregation may happen .To avoid this problem, we adopt CDAMA (k = 2) scheme to provide secure counting for single application case, i.e., the BS exactly knows how many sensed readings are aggregated while it receives the final result.

Security Analysis and Comparison

In this section, I analyze the security of CDAMA and other conventional schemes. More specifically, we compare CDAMA with four well-known CDA schemes: CDA [9], [10], Castelluccia et al.'s scheme [14], Mykletun et al.'s scheme [11], and TinyPEDS [12]. In Mykletun et al.'s scheme, the authors applied several well-known public key PH schemes to WSNs. They recommended two schemes which are suitable for WSNs, EC-OU and EC EG. Since TinyPEDS [12] is the same as the EC-EG scheme[11], we chose TinyPEDS as a candidate. In addition to these four schemes, BGN—from which our proposed CDAMA is extended—is also analyzed. Consequently, we analyze CDA, Castelluccia et al.'s scheme, TinyPEDS,EC-OU, BGN, and CDAMA based on the attack model.

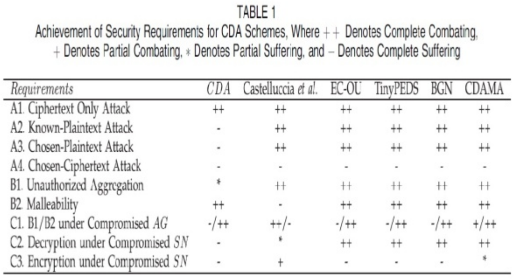


TABLE 1
Achievement of Security Requirements for CDA Schemes, Where ++ Denotes Complete Combating, + Denotes Partial Combating, * Denotes Partial Suffering, and – Denctes Complete Suffering

Requirements	*CDA*	Castelluccia *et al.*	EC-OU	TinyPEDS	BGN	CDAMA
A1. Ciphertext Only Attack	++	++	++	++	++	++
A2. Known-Plaintext Attack	-	++	++	++	++	++
A3. Chosen-Plaintext Attack	-	++	++	++	++	++
A4. Chosen-Ciphertext Attack	-	-	-	-	-	-
B1. Unauthorized Aggregation	*	++	++	++	++	++
B2. Malleability	++	-	++	++	++	++
C1. B1/B2 under Compromised *AG*	-/++	++/-	-/++	-/++	-/++	+/++
C2. Decryption under Compromised *SN*	-	*	++	++	++	++
C3. Encryption under Compromised *SN*	-	+	-	-	-	*

7 CONCLUSIONS

In this section, I designed multi-application environment, CDAMA is the first CDA scheme. Through CDAMA, the cipher texts from distinct applications can be aggregated, but not mixed. For a single-application environment, CDAMA is still more secure than other CDA schemes. When compromising attacks occur in WSNs, CDAMA mitigates the impact and reduces the damage to an acceptable condition. In above applications, CDAMA is the first CDA scheme that supports secure counting. The base station would know the exact number of messages aggregated, making selective or repeated aggregation attacks infeasible. Finally, the performance evaluation shows that CDAMA is applicable on WSNs while the number of groups or applications is not large. In future, we wish to apply CDAMA to realize aggregation query in Database-As-a-Service (DAS) model. In DAS model, a client stores her database on an untrusted service provider. Therefore, the client has to secure their database through PH schemes because PH schemes keep utilizable properties than standard ciphers. Based on PH schemes, the provider can conduct aggregation queries without decryption.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Comm. Magazine, vol. 40, no. 8, pp. 102-114, Aug. 2002.

[2] R. Min and A. Chandrakasan, "Energy-Efficient Communication for Ad-Hoc Wireless Sensor Networks," Proc. Conf. Record of the 35th Asilomar Conf. Signals, Systems and Computers, vol. 1, 2001.

[3] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure Information Aggregation in Sensor Networks," Proc. First Int'l Conf. Embedded Networked Sensor Systems, pp. 255-265, 2003.

[4] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," Comm. ACM, vol. 47, no. 6, pp. 53-57, June 2004.

[5] L. Hu and D. Evans, "Secure Aggregation for Wireless Networks," Proc. Symp. Applications and the Internet Workshops, pp. 384-391, 2003.

[6] H. Cam, S. O ¨ zdemir, P. Nair, D. Muthuavinashiappan, and H.O. Sanli, "Energy-Efficient Secure Pattern Based Data Aggregation for Wireless Sensor Networks," Computer Comm., vol. 29, no. 4, pp. 446-455, 2006.

[7] H. Sanli, S. Ozdemir, and H. Cam, "SRDA: Secure Reference-based Data Aggregation Protocol for Wireless Sensor Networks," Proc.
IEEE 60th Vehicular Technology Conf. (VTC '04-Fall), vol. 7, 2004.

[8] Y. Wu, D. Ma, T. Li, and R.H. Deng, "Classify Encrypted Data in Wireless Sensor Networks," Proc. IEEE 60th Vehicular Technology Conf., pp. 3236-3239, 2004.

[9] D. Westhoff, J. Girao, and M. Acharya, "Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution, and Routing Adaptation," IEEE Trans. Mobile Computing, vol. 5, no. 10, pp. 1417-1431, Oct. 2006.

[10] J. Girao, D. Westhoff, M. Schneider, N. Ltd, and G. Heidelberg, "CDA: Concealed Data Aggregation for Reverse Multicast
Traffic in Wireless Sensor Networks," Proc. IEEE Int'l Conf. Comm. (ICC '05), vol. 5, 2005

[11] E. Mykletun, J. Girao, and D. Westhoff, “Public Key Based Cryptoschemes for Data Concealment in Wireless Sensor Networks,” Proc. IEEE Int’l Conf. Comm. (ICC ’06), vol. 5, 2006.

[12] J. Girao, D. Westhoff, E. Mykletun, and T. Araki, “Tinypeds: Tiny Persistent Encrypted Data Storage in Asynchronous Wireless Sensor Networks,” Ad Hoc Networks, vol. 5, no. 7, pp. 1073-1089, 2007.

[13] D. Boneh, E. Goh, and K. Nissim, “Evaluating 2-DNF Formulas on Ciphertexts,” Proc. Second Int’l Conf. Theory of Cryptography (TCC), vol. 3378, pp. 325-341, 2005.

[14] C. Castelluccia, E. Mykletun, and G. Tsudik, “Efficient Aggregation of Encrypted Data in Wireless Sensor Networks,” Proc. Second Ann. Int’l Conf. Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous ’05), pp. 109-117, 2005.

[15] S. Peter, D. Westhoff, and C. Castelluccia, “A Survey on the Encryption of Convergecast-Traffic with In-Network Processing,” IEEE Trans. Dependable and Secure Computing, vol. 7, no. 1, pp. 20- 34, Jan.-Mar. 2010.