Aggregation of Encrypted data in Wireless Sensor Networks for Secure Multi-Application Data

S. Nazeera Banu*1, T. Pavithra*2, K. Priyadharshini*3, B.Devendar Rao#4, Dr. D.C. Joy Winnie Wise#5

*1,2,3U.G Students, B.E CSE, Alpha College of Engg, Chennai, T.N, India.
#4Assistant Professor, Dept of CSE, Alpha College of Engg, Chennai, T.N, India.
#5Professor, HOD, Dept of CSE, Alpha College of Engg, Chennai, T.N, India.

[1]nazeerecse@gmail.com

Abstract---Generally, aggregative operations are algebraic, such as the addition or multiplication of received data, or statistical operation, such as a median, a minimum, or a maximum of a data set. It reduces a large amount of transmission is the most practical technique. Although data aggregation could significantly reduce transmission, it is vulnerable to some attacks. Some of the existing schemes do not provide secure counting. Thus, they may suffer from unauthorized aggregation attacks. We propose 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.

Keywords---Concealed data aggregation, elliptic curve cryptography, wireless sensor networks

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of thousands of sensor nodes (SN) that gather data from deployed environments. Currently, there are plenty of rich applications proposed for WSNs, such as environment monitoring, accident reporting, and military investigation. 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 a better energy utilization, cluster-based WSNs [1] have been proposed. In cluster-based WSNs, 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. Generally, aggregative operations are algebraic, such as the addition or multiplication of received data, or statistical operation, such as a median, a minimum, or a maximum of a data set. Although data aggregation could significantly reduce transmission, it is vulnerable to some attacks. An alternative approach for this problem is to aggregate encrypted messages directly from SN, thereby avoiding the forgery of aggregated result. Since CHs are not capable of encrypting messages, compromising a CH earns nothing in forging aggregated results. In this paper, the proposed scheme, called CDAMA. CDAMA is a modification from Boneh et al.'s PH scheme [2]. Here, we 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. Previously, the ciphertexts of different applications cannot be aggregated together; otherwise, the decrypted aggregated result will be incorrect. The only solution is to aggregate the ciphertexts of different applications separately. As a result, the transmission cost grows as the number of the applications increases. By CDAMA, the ciphertexts from different applications can be encapsulated into "only" one ciphertext. Conversely, the base station can extract application-specific plaintexts via the corresponding secret keys. The second scenario is designed for single application WSNs. Compared with conventional schemes [3], [4]. CDAMA mitigates the impact of compromising SN through the construction of multiple groups. An adversary can forge data only in the compromised groups, not the whole system.

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; leaking count knowledge will suffer maliciously selective aggregation and repeated aggregation. In CDAMA, the base station exactly knows the number of messages aggregated to avoid above attacks.

II. RELATED WORKS

A. 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 [3], [4]. By PH schemes, SNs encrypt their sensed readings and allow AGs to homomorphically aggregate their ciphertexts without decryption. Therefore, compromising AGs earns no advantage of forging aggregated results. Westhoff et al. [3] and Girao et al. [4] proposed CDA based on symmetric PH to facilitate the aggregation of encrypted data. In contrast to symmetric PH construction, Mykletun et al. adopted public-key based PH to construct their systems, and Girao et al. 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 [3], [4], an adversary can forge the aggregated results by simply compromising one SN. To solve this problem, Castelluccia et al. proposed an encryption scheme similar to onetime pad. If an adversary tries to forge aggregated results, he must compromise all SNs. 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.

III. DESIGN AND IMPLEMENTATION

A. Outline and Architecture

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. We 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).

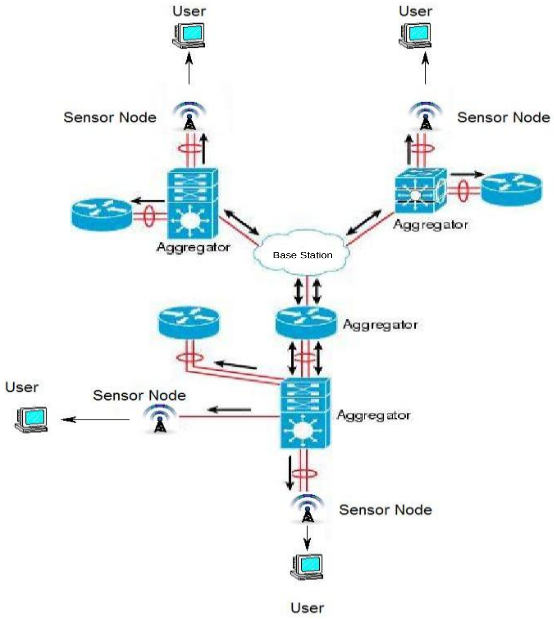


Fig .1. System Architecture

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. CDAMA contains four procedures: Key generation, encryption, aggregation, and decryption. All sensor nodes are divided into two groups. The encryptions of messages of two groups can be aggregated to a single ciphertext, but the aggregated message of each group can be obtained. Considering deployment, the private keys should be kept secret and only known by the BS.

B. Key distribution

In the end of this section, we briefly address how to
deliver the group public keys to SNs securely. There are two main approaches.

Key predistribution: If we know the locations of deployed SNs, we can preload necessary keys and functions into SNs and AGs so that they can work correctly after being spread out over a geographical region.

Key postdistribution: 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 in LEAP [5] and the master secret key in SPINS [6]. Once these SNs are deployed, they can run the LEACH protocol [1] to elect the AGs and construct clusters. After that, the BS sends the corresponding CDAMA keys, encrypted by the preshared key, to SNs and AGs.

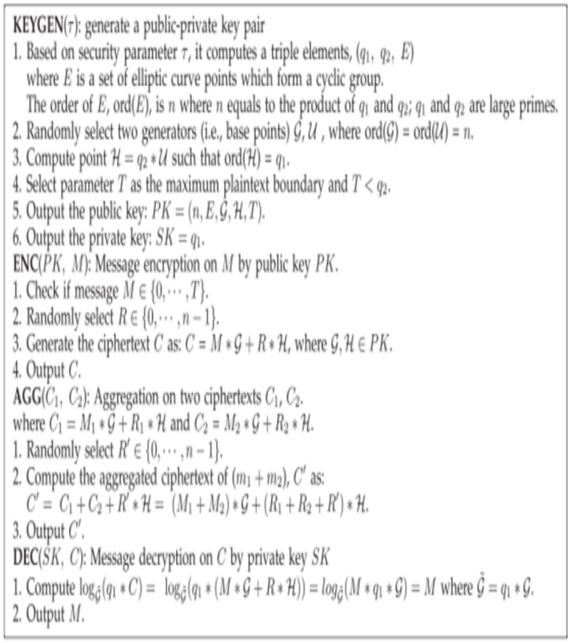
C. Aggregation and secure counting

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 (here, we call "count"), 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.

The BS obtains the aggregated result M and its count ζ. If a malicious AG launches unauthorized aggregations, such as repeated or selective aggregation, ζ's value would be changed to a bigger or smaller value than the reference count . Since the AG does not know the base points P and Q, unauthorized aggregations have to alter the values of ζ and M simultaneously; it is impossible to alter M without changing ζ. Meanwhile, the BS knows the number of deployed sensors through gathering topology information, the BS can detect unauthorized aggregation based on the value of ζ. For each application, one group sums its messages and the other group counts the number of messages aggregated. As a result, unauthorized aggregation by AG can be mitigated. Although this mechanism will fail after a SN has been compromised (knowing the base point Q), it nevertheless mitigates unauthorized aggregation by AGs, where most asymmetric schemes cannot achieve.

D. BGN Scheme

Boneh et al. proposed a public-key PH scheme, which integrates the Paillier with the Okamoto- Uchiyama encryption schemes. We call it BGN for simplicity. BGN provides additive and multiplicative homomorphism. Since the multiplicative property, based on the bilinear pairing, is much expensive and inefficient for SNs, we only utilize the additive homomorphism of BGN. In this paper, we first provide a possible application for BGN, data aggregation. Furthermore, we modify BGN to fit multigroup construction for stronger security and better applicability in CDA. BGN is constructed on a cyclic group of elliptic curve points. Precisely, these points form an algebraic group, where the identity element of the group is the infinite point, ∞. Notation ord(P) denotes the order of a point P. Supposing ord(P) =q, it indicates that q is the minimum integer that satisfies $q * P = \infty$. In the KEYGEN function, the order of E is equivalent to the number of points in E.



```
**KEYGEN**($\tau$): generate a public-private key pair
1. Based on security parameter $\tau$, it computes a triple elements, $(q_1,\ q_2,\ E)$
   where $E$ is a set of elliptic curve points which form a cyclic group.
   The order of $E$, ord($E$), is $n$ where $n$ equals to the product of $q_1$ and $q_2$; $q_1$ and $q_2$ are large primes.
2. Randomly select two generators (i.e., base points) $\mathcal{G}, \mathcal{U}$ , where $\mathrm{ord}(\mathcal{G}) = \mathrm{ord}(\mathcal{U}) = n$.
3. Compute point $\mathcal{H} = q_2 * \mathcal{U}$ such that $\mathrm{ord}(\mathcal{H}) = q_1$.
4. Select parameter $T$ as the maximum plaintext boundary and $T < q_2$.
5. Output the public key: $PK = (n, E, \mathcal{G}, \mathcal{H}, T)$.
6. Output the private key: $SK = q_1$.
**ENC**($PK,\ M$): Message encryption on $M$ by public key $PK$.
1. Check if message $M \in \{0, \cdots, T\}$.
2. Randomly select $R \in \{0, \cdots, n-1\}$.
3. Generate the ciphertext $C$ as: $C = M * \mathcal{G} + R * \mathcal{H}$, where $\mathcal{G}, \mathcal{H} \in PK$.
4. Output $C$.
**AGG**($C_1,\ C_2$): Aggregation on two ciphertexts $C_1$, $C_2$.
where $C_1 = M_1 * \mathcal{G} + R_1 * \mathcal{H}$ and $C_2 = M_2 * \mathcal{G} + R_2 * \mathcal{H}$.
1. Randomly select $R' \in \{0, \cdots, n-1\}$.
2. Compute the aggregated ciphertext of $(m_1 + m_2)$, $C'$ as:
   $C' = C_1 + C_2 + R' * \mathcal{H} = (M_1 + M_2) * \mathcal{G} + (R_1 + R_2 + R') * \mathcal{H}$.
3. Output $C'$.
**DEC**($SK,\ C$): Message decryption on $C$ by private key $SK$
1. Compute $\log_{\tilde{\mathcal{G}}}(q_1 * C) = \log_{\tilde{\mathcal{G}}}(q_1 * (M * \mathcal{G} + R * \mathcal{H})) = log_{\tilde{\mathcal{G}}}(M * q_1 * \mathcal{G}) = M$ where $\tilde{\mathcal{G}} = q_1 * \mathcal{G}$.
2. Output $M$.
```

BGN Scheme

The ENC function is based on point addition and scalar multiplication over points G and H. As we can see, the ciphertext is composed of the message part (the scalar of the point G) and the secure randomness (the scalar of the point H). Due to homomorphic properties, the AGG function aggregates ciphertexts via point addition; it is trivial to see that the scalar values of point G were added in the end, yielding the sum of the corresponding message. Consequently, the final result will be the form of M * G + R*H, where M is the sum of the messages and R is the sum of the randomness.

The DEC function decrypts the aggregated result to obtain the plaintext value, M. Recall that the order of points G and H are different. Hence, the DEC function removes the randomness of point H by multiplying the result with the private key (i.e., ord(H)). Now, the cipher text contains only the product of G (i.e., ord(H) *M * G) such that we can apply the discrete logarithm to retrieve the value M. In fact, discrete logarithm can be solved by Pollard's λ method whose efficiency is O(√T). When AGG operations is performed on all ciphertexts it receives, e.g., AGG(…AGG (AGG (C1,C2), C3)…..Ci). Then, AG sends the aggregated result to the next aggregator. Finally, BS decrypts the aggregated result through the DEC function with the private key SK.

V. CONCLUSION

For a multi-application environment, CDAMA is the first CDA scheme. Through CDAMA, the ciphertexts 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. Besides the 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 the future, we wish to apply CDAMA to realize aggregation query in Database-As-a-Service (DAS) model [7]. 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. The most important of all is that we do not have to consider the computation cost and the impact of compromising secret keys (i.e., compromising a client in DAS model is harder than compromising a sensor). Those drawbacks will no longer be issues in CDAMA. For the policy matching procedure, we described algorithms to efficiently search the policy database assessment.

We extended the Postgre-SQL open source to implement our methods. Specifically, we added support for new system catalogues to hold policy related data, for the policy administration tasks and integrated the policy matching code with the query processing subsystem of Postgre-SQL. The experimental evaluation of our policy matching algorithms showed that our techniques are efficient.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," ACM Trans. Sensor Networks, vol. 2, no. 4, pp. 500-528, 2006.

[6] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. Culler, "SPINS: Security Protocols for Sensor Networks," Wireless Networks, vol. 8, no. 5, pp. 521 534, 2002.

[7] B. Iyer, C. Li, and S. Mehrotra, "Executing Sql over Encrypted Data in the Database-Service-Provider Model," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 216-227, 2002.