Hierarchical Concealed Data Aggregation for Wireless Sensor Networks

Suat Ozdemir and Yang Xiao†
Computer Engineering Department, Gazi University, Ankara, TURKEY TR-06570
†Department of Computer Science, The University of Alabama Tuscaloosa, AL 35487-0290
suatozdemir@gazi.edu.tr, yangxiao@cs.ua.edu

***Abstract*—In wireless sensor networks, performing data aggregation while preserving data confidentiality is a challenging task. Recently, privacy homomorphism based secure data aggregation schemes have been proposed to achieve seamless integration of data confidentiality and aggregation. If sensor data are encrypted with different keys, however, these schemes do not allow hierarchical data aggregation, thereby limiting the benefit of data aggregation. This paper presents a novel hierarchial concealed data aggregation protocol that allows the aggregation of data packets which are encrypted with different keys. Hence, regardless of the encryption key, data collected from all sensor nodes can be aggregated without violating data confidentiality. Moreover, during the decryption of aggregated data, the base station is able to classify sensor data based on the encryption key.**

I. Introduction

A wireless sensor network is composed of large number of sensor nodes that have strictly limited computation and communication abilities and power resources [1]. In the near future, wireless sensor networks are envisioned to be employed widely in many applications including critical area surveillance, home and office automation, habitat monitoring, health monitoring, and military tracking. Therefore, security is an essential issue in wireless sensor networks and widespread deployment of these networks could be curtailed without adequate security [2], [3]. However, compared to conventional computer networks, implementing security is not easy in wireless sensor networks due to limited processing power, storage, bandwidth, and energy of sensor nodes. In addition to security, limited battery power and bandwidth of sensor nodes make it a challenging task to provide efficient solutions to data gathering problem. Therefore, in order to reduce the power and bandwidth consumption of wireless sensor networks, several mechanisms are proposed such as data aggregation [4]. Data aggregation protocols aim to combine and summarize data packets of several sensor nodes so that overall in network communication bandwidth and energy consumptions are reduced.

Since both data aggregation and security are essential for wireless sensor networks, providing *secure data aggregation* has been an attractive problem for researchers [5], [6], [7], [8], [9], [10], [11]. In many of the existing secure data aggregation protocols, data aggregators must decrypt every message they receive, aggregate the messages according to the corresponding aggregation function, and encrypt the aggregation result before forwarding it. Therefore, while these data aggregation protocols improve the bandwidth and energy utilization in the network, they negatively affect other performance metrics such as delay and security. To support secure data aggregation without causing delay, a set of data aggregation protocols is proposed. These protocols use privacy homomorphic encryption to allow data aggregation without requiring decryption of the data [12], [13], [14]. Protocols in [12] and [13] utilize symmetric and asymmetric privacy homomorphic encryption to allow aggregation of encrypted data, respectively. However, in [12], sensor data must be encrypted with a single key to perform concealed data aggregation. Therefore, in order to hierarchically aggregate data of the whole network, sensor nodes in the network must share a common key and use it for encryption. Using a single symmetric key in the network is not secure as an adversary can fake the aggregated results through compromising only a sensor node. In addition, symmetric key based privacy homomorphism is shown to be insecure for chosen plaintext attacks for some specific parameter settings [15]. The scheme proposed in [13] relies on asymmetric key based privacy homomorphism but it also requires a single public key to allow hierarchical data aggregation. The scheme proposed in [14] allows using different encryption keys in aggregated data. Authors employ an extension of the one-time pad encryption technique using additive operations modulo n. However, several practical issues are not addressed in this paper such as requirement of a strong synchronization mechanism.

In this paper, we propose Hierarchical Concealed Data Aggregation (HCDA) protocol which allows concealed aggregation of data that are encrypted with different keys. HCDA protocol virtually partitions the network into several regions and employs a different public key in each region. Due to the privacy homomorphic encryption scheme [20] of HCDA, the data collected in a region can be encrypted using the public key of the region and the encrypted data of several regions can be hierarchically aggregated into a single piece of data without violating data confidentiality. Moreover, during the decryption of aggregated data, the base station is able to determine the origin of the data based on the encryption key. This is

Dr. Ozdemir's work is supported in parts by the Gazi University Scientific Research Project Fund No:06/2008-44. Prof. Y. Xiao's work is supported in parts by the US National Science Foundation (NSF) under grants numbers: CCF-0829827, CNS-0716211, and CNS-0737325.

particulary useful when the base station needs data from a certain region of the network. In order to use multiple keys in the network area, HCDA protocol employs a group based network deployment scheme where sensor nodes in a group use the same public key. In addition, as HCDA protocol is based on elliptic curve cryptography, it is not affected by node compromise attacks whereas symmetric key based concealed data aggregation protocols [12] are significantly affected from these attacks. Our theoretical analysis shows that HCDA is feasible for resource constrained sensor nodes.

Our contribution in this work is that we provide a concealed data aggregation technique that allows hierarchical aggregation of data encrypted with different keys. Note that to the best of our knowledge this property cannot be efficiently achieved by any other existing concealed data aggregation scheme. Figure 1 presents an example for the motivation behind HCDA scheme.

The rest of the paper is organized as follows. In Section II, the state-of-the-art in secure data aggregation is presented. Section III explains the system model and preliminaries along with HCDA's network deployment scenario. HCDA protocol is given in Section IV. Concluding remarks are made in Section V.

II. Related Work

In wireless sensor network domain, secure data aggregation problem is studied extensively [5], [6], [7], [8], [9], [10], [11]. In [5], the security mechanism detects node misbehaviors such as dropping or forging messages and transmitting false data. In [6], random sampling mechanisms and interactive proofs are used to check the correctness of the aggregated data at the base station. In [8], witness nodes of data aggregators also aggregate data and compute MACs to help verify the correctness of the aggregators' data at base station. Because the data validation is performed at base station, the transmission of false data and MACs up to base station affects adversely the utilization of sensor network resources. In [9], sensor nodes use the cryptographic algorithms only when a cheating activity is detected. Topological constraints are introduced to build a secure aggregation tree (SAT) that facilitates the monitoring of data aggregators. In [10], a Secure Hop-by-hop Data Aggregation Protocol (SDAP) is proposed. The authors of SDAP are motivated by the fact that, compared to low level sensor nodes, more trust is placed on the high-level nodes (i.e., nodes closer to the root) during a normal hop-by-hop aggregation process in a tree topology. In [11], the authors propose a protocol that makes use of a *web of trust* to overcome the shortcomings of cryptography based secure data aggregation solutions.

Privacy homomorphism is introduced by Rivest et al. [16]. For example, Rivest's asymmetric key algorithm RSA is multiplicatively homomorphic. Due to their high computational overhead, such asymmetric key homomorphic encryption algorithms are not feasible for sensor nodes [17]. The privacy homomorphic encryption algorithm introduced by Domingo-Ferrer [18] is symmetric key based. The concealed data aggregation algorithm that is proposed in [12] employs Domingo-Ferrer's privacy homomorphic encryption algorithm. However, in order to hierarchically aggregate the data of the all network, the proposed scheme must uses a secret key known by all sensor nodes which leads to the following attack. If a sensor node is compromised, it can decrypt data of any sensor node which is encrypted by the secret key. Hence, in this paper, we use a privacy homomorphic function that is based on elliptic curve cryptography (ECC). Compared to RSA, ECC provides the same security level with shorter key size and ciphertexts. It is shown that 160-bit ECC key provides the same security as 1024-bit RSA key provides [19]. Since communication communication overhead of wireless sensor networks depends on the size of data packets, ECC based privacy homomorphic encryption schemes are more preferable.

III. System Model and Preliminaries

We consider a large sensor network with densely deployed sensor nodes. Due to the dense deployment, sensor nodes have overlapping sensing ranges and events are detected by multiple sensor nodes. Hence, data aggregation is needed to reduce data transmission. Some sensor nodes are dynamically designated as data aggregators to aggregate data from their neighboring sensor nodes, although every sensor node is assumed to be capable of doing data aggregation. To balance the energy consumption of sensor nodes, the role of data aggregator is rotated among sensor nodes based on their residual energy levels. Sensor nodes have limited computation and communication capabilities [17]. All messages are time-stamped and nonces are used to prevent reply attacks. Sensor nodes encrypt their data prior to data transmission. Encrypted data are decrypted only at the base station. The base station is interested in data of a region rather than data of a single sensor node. Therefore, the network deployment area is divided into several regions as described below.

A. Network Deployment

In order to virtually divide the network area into several regions, sensor network is deployed using a strategy described in [21]. In this deployment scenario, before the deployment, sensor nodes are divided into several groups and each group is deployed from a certain location over the network area. The network deployment is usually achieved dropping the sensor node groups from a plane or a helicopter. Hence each deployment group covers a part of the network. Using this deployment scenario, HCDA assigns a public key to each deployment group so that the base station is able to classify the data of the groups based on the public key that is used to encrypt the data.

In the network deployment scenario, we assume that sensor nodes are distributed with the Gaussian distribution. Gaussian distribution allows us to compute the maximum *distance* between two deployment points over the network area. Because, in Gaussian distribution, the distances between the deployment point of sensor nodes and their final locations are guaranteed to be less than 3σ with probability 0.9987, where σ is the standard deviation of the Gaussian distribution. If each sensor

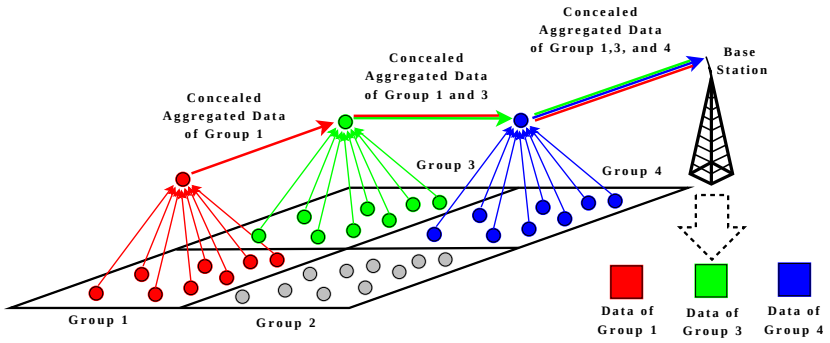

Fig. 1. The motivation behind HCDA protocol. The network consists of four deployment groups. Each group's data are hierarchically aggregated without violating the data confidentiality. During the decryption of the aggregated data, the base station is able extract data of each deployment group.

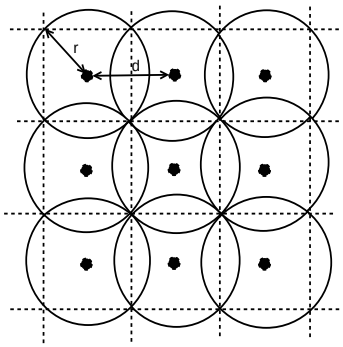

Fig. 2. Determining the positions of the deployment points: Radius (r) of each circle is 3σ by Gaussian distribution, therefore the maximum distance (d) between any two deployment points cannot exceed $3\sqrt{2}\sigma$ to provide full coverage.

node group covers a circular area with radius 3σ centered at its deployment point, the network area is fully covered. Therefore, in order to have full coverage of the network area, the distance (d) between two deployment points should not exceed $3\sqrt{2}\sigma$ as shown in Figure 2.

B. Privacy Homomorphism

A privacy homomorphism is an encryption transformation that allows direct computation on encrypted data. Let E denotes *encryption* and D denotes *decryption*. Also let $+$ denotes addition and $\times$ denotes multiplication operation over a data set Q. Assume that K_r and K_u are the private and public keys of the base station, respectively. An encryption transformation is accepted to be additively homomorphic, if

$$a+b = D_{K_r}(E_{K_u}(a)+E_{K_u}(b)) \quad where \quad a,b \in Q.$$

and it is accepted to be multiplicatively homomorphic, if

$$a \times b = D_{K_r}(E_{K_u}(a) \times E_{K_u}(b)) \quad where \quad a,b \in Q.$$

Since, additively and multiplicatively homomorphic cryptographic functions support additive and multiplicative operations on encrypted data, respectively, data aggregators can perform addition and multiplication based data aggregation over the encrypted data. Privacy homomorphic encryption can be achieved using symmetric or asymmetric cryptography. Recently, privacy homomorphism based on symmetric key cryptography is shown to be insecure for chosen plaintext attacks for some specific parameter settings [15]. Therefore, for mission critical networks, asymmetric cryptography based privacy homomorphism should be used instead of symmetric cryptography based privacy homomorphism. However, public key based privacy homomorphism is prohibitively expensive for resource limited wireless sensor networks.

Realizing that asymmetric cryptography based privacy homomorphism incurs high computational overhead, HCDA protocol employs the elliptic curve cryptography based privacy homomorphism proposed in [20] which allows concealed aggregation of data that are encrypted with different keys. Although the encryption scheme of [20] provides additive and multiplicative homomorphism, HCDA protocol only takes advantage of the additive homomorphism property because multiplicative homomorphism is prohibitively expensive. In what follows, we describe additive homomorphic encryption process of HCDA protocol as explained in [20].

- ***Key generation:*** Given a security parameter $\tau \in \mathbb{Z}$ compute $\varphi(\tau)$ to generate the tuple (q_1, q_2, E, n). E is a set of elliptic curve points that form a cyclic group. The set E should be order of n where $n = q_1 q_2$. Randomly select two points (u and g) of order n from E. Set $h = u^{q_2}$ where h's order is q_1. Set the public key as $P_u = (n, E, g, h)$ and the private key as $P_r = q_1$.
- ***Encryption:*** Set an integer T where $T < q_2$ and let the bit length of T be approximately close to the bit length of q_2. The message M space should consist of integers in the set $\{0, 1, \cdots, T\}$. To encrypt a message m using public key P_u, pick a random $r \leftarrow \{0, 1, \cdots, n-1\}$ and compute the ciphertext $\mathcal{C} = g^m + h^r$ where $+$ is the addition of elliptic curve points and a^b is the scalar multiplication of elliptic curve points a and b. It should be noted that the encryption process relies on the random number r, the resulting ciphertext is probabilistic, and therefore the scheme is resilient to chosen plaintext attacks [15].
- ***Decryption:*** To decrypt a ciphertext $\mathcal{C}$ using the private key $P_r = q_1$, observe that $\mathcal{C}^{q_1} = (g^m + h^r)^{q_1} = (g^{q_1})^m$. Let $\hat{g} = g^{q_1}$, then to recover m, it suffices to compute

the discrete log of C^{q_1} base $\hat{g}$. It should be noted that the message m is between 0 and T, and therefore the decryption operation takes $O(\sqrt{T})$ time using Pollards lambda method [22].

- ***Aggregation:*** Two ciphertexts $\mathcal{C}_1 = g^{m_1} + h^{r_1}$ and $\mathcal{C}_2 = g^{m_2} + h^{r_2}$ are aggregated into a ciphertext of $\mathcal{C}'$ as follows:

$$\mathcal{C}' = \mathcal{C}_1 + \mathcal{C}_2 = g^{(m_1+m_2)} + h^{(r_1+r_2)}.$$

For more details such as proof of homomorphism, interested readers are referred to [20]. Let us give an example to show how this encryption scheme can be employed in the concept of wireless sensor networks. In order to encrypt a message m_i, a sensor node N_i first chooses a random number r_i, and computes the ciphertext $\mathcal{C}_i = g^{m_i} + h^{r_i}$ using the public key (n, E, g, h). Similarly, in order to encrypt a message m_j, a sensor node N_j first selects a random number r_j, and computes the ciphertext $\mathcal{C}_j = g^{m_j} + h^{r_j}$ using the public key (n, E, g, h). Assume that a data aggregator aggregates $\mathcal{C}_i$ and $\mathcal{C}_j$ into $\mathcal{C}_{agg}$ and sends it to the base station. Then, the base station computes the aggregated message by calculating the discrete logarithm of ${\mathcal{C}_{agg}}^{q_1}$ to the base $\hat{g}$ where q_1 is the private key and $\hat{g} = g^{q_1}$.

IV. HCDA: Hierarchical Concealed Data Aggregation

In the previous section, we explained the homomorphic encryption scheme of [20]. If the sensor network uses a single public-private key pair, this encryption scheme can be used in HCDA protocol directly. However, HCDA protocol aims to hierarchically aggregate data of multiple sensor node groups that use different public-private key pairs. Therefore, in what follows, we describe HCDA protocol's modified homomorphic encryption scheme that allows aggregation of k deployment groups' data. At the end of this section, we present a concrete example of HCDA protocol for a wireless sensor network that consists of two deployment groups.

- ***Key generation:*** Given a security parameter $\tau \in \mathbb{Z}$ compute $\varphi(\tau)$ to generate the tuple $(q_1, q_2, \cdots, q_{k+1}, E, n)$. E is a set of elliptic curve points that form a cyclic group. The order of E is n where $n = q_1 q_2 \cdots q_{k+1}$. Randomly select $k+1$ points $(u_1, u_2, \cdots u_{k+1})$ from E where the order of u_i is n for $i = 1$ to $k+1$. Set h as follows:

$$h = u_{k+1}^{\beta} \quad where \quad \beta = \prod_{i=1}^{k} q_i \quad and \quad i = 1, \cdots, k$$

  The order of h is q_{k+1}. Now, we need k public keys for k deployment groups, hence we compute a P value for each deployment group as follows:

$$P_z = g_z^{\alpha} \quad where \quad \alpha = \prod_{i=1, i\neq z}^{k+1} q_i \quad and \quad z = 1, \cdots, k$$

  The public key of deployment group z is P_u^z = (n, E, P_z, g, h) for $z = 1$ to k and the private key is P_r = $(q_1, q_2, \cdots, q_{k+1})$.
- ***Encryption:*** Set $T_z < q_z$ and let the bit length of T_z be approximately close to the bit length of q_z. The message M space of a sensor node that belongs to deployment group z should consist of integers in the set $\{0, 1, \cdots, T_z\}$. To encrypt a message m using public key P_u^z, pick a random $r \leftarrow \{0, 1, \cdots, n-1\}$ and compute the ciphertext $\mathcal{C} = P_z^m + h^r$ where $+$ is the addition of elliptic curve points and a^b is the scalar multiplication of elliptic curve points a and b.
- ***Aggregation:*** Let $\sum m_i$ denotes that the aggregated message of ith deployment group, then k ciphertexts $\mathcal{C}_z = P_z^{m_z} + h^{r_z}$ for $z = 1$ to k are aggregated into a ciphertext of $\mathcal{C}'$ as follows:

$$\mathcal{C}' = \sum_{i=1}^{k} P_i^{\sum m_i} + h^{\sum r_i}$$

- ***Decryption:*** During the decryption the base station is able to separately decrypt the data of each deployment group z from the aggregated ciphertext $\mathcal{C}'$. Let $\hat{g_z}$ be

$$\hat{g_z} = g_z^{\alpha} \quad where \quad \alpha = \prod_{i=1, i\neq z}^{k+1} q_i \quad and \quad z = 1, \cdots, k$$

  then the base station can recover the aggregated data $\sum_{i=z} m_i$ of each deployment group z by computing the discrete log of $(\mathcal{C}')^{\alpha}$ base $\hat{g}_z$. Therefore, decrypted data of deployment group z is

$$\sum_{i=z} m_i = log_{\hat{g_i}} (\mathcal{C}')^{\alpha} where$$

$$\hat{g_i} = g_z^{\alpha}, \quad \alpha = \prod_{i=1, i\neq z}^{k+1} q_i, \quad and \quad z = 1, \cdots, k$$

A. Example

Now, we present an example to show how HCDA protocol achieves hierarchical concealed aggregation of multiple deployment groups' data. For the sake of simplicity, let us assume that the network consists of four deployment groups and only two groups send data to the base station. Group 1 has the public key P_u^1 = (n, E, P_1, g, h) and group 2 has the public key P_u^2 = (n, E, P_2, g, h). As shown in Figure 3 each group has two sensor nodes and a data aggregator. Group 1 has sensor nodes SN_1^A, SN_1^B and data aggregator DA_1. Similarly, group 2 has sensor nodes SN_2^A, SN_2^B and data aggregator DA_2. There is also another data aggregator DA_3 of group 3 that aggregates and transmits data of DA_1 and DA_2 to the base station. In order to keep the example simple, order of P_1, P_2, and h are set to small numbers as follows:

- Order of P_1 and value of q_1 is 11
- Order of P_2 and value of q_2 is 13
- Order of h and value of q_3 is 17
- Order of $n = q_1 q_2 q_3$ is 2431

Sensor nodes in group 1 and 2 encrypt and send their data as follows (note that r values are randomly generated by sensor nodes)

- SN_1^A generates message $M_1^A = 1$ and encrypts it as $\mathcal{C}_1^A = P_1^1 + h^4$

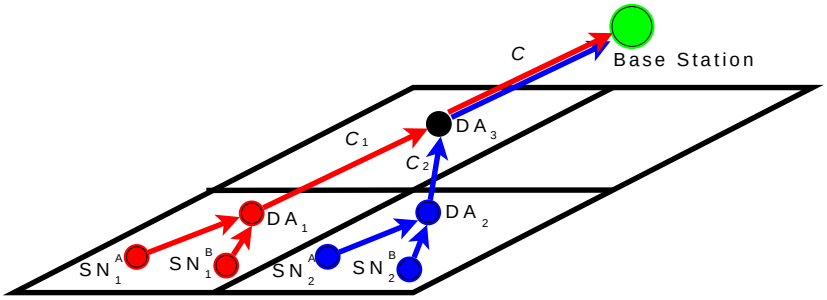

Fig. 3. Example of HCDA protocol. For the sake of simplicity, only two groups send data to the base station. DA_3 aggregates data of DA_1 and DA_2

- SN_1^B generates message $M_1^B = 3$ and encrypts it as $C_1^B = P_1^3 + h^6$
- SN_2^A generates message $M_2^A = 4$ and encrypts it as $C_2^A = P_2^4 + h^2$
- SN_2^B generates message $M_2^B = 2$ and encrypts it as $C_2^B = P_2^2 + h^7$

Sensor nodes send their messages to data aggregators. Data aggregator DA_1 aggregates C_1^A and C_1^B as $C_1 = P_1^4 + h^{10}$. Similarly, data aggregator DA_2 aggregates C_2^A and C_2^B as $C_2 = P_2^6 + h^9$. DA_1 and DA_2 send their aggregated data to DA_3. DA_3 aggregates C_1 and C_2 as $C = P_1^4 + P_2^6 + h^{19}$. Since order of h is 17, $h^{17} = \infty$, and ∞ is the additive unit element in elliptic curve arithmetic, we can write $C = P_1^4 + P_2^6 + h^2$. DA_3 sends C to the base station.

In order to obtain the data of group 1, the base station first computes $C^{q_2 q_3} = (P_1^4 + P_2^6 + h^2)^{221}$. Since a^b donates scalar multiplication of elliptic curve points, $C^{q_2 q_3}$ equals $P_1^{884} + P_2^{1326} + h^{442}$. Note that $h^{17} = \infty$, $P_1^{11} = \infty$, and $P_2^{13} = \infty$, then, using elliptic curve arithmetic, we can write $Cq_2q_3 = P_1^4$. Finally, the base station obtains data of group 1 by computing the discrete logarithm of $C^{q_2 q_3} = P_1^4$ to the base $\hat{g_1}$ where $\hat{g_1} = g_1^{q_2 q_3}$.

V. Conclusion

In this paper, we have presented our ongoing work on hierarchical concealed data aggregation. The proposed scheme allows the aggregation of data packets which are encrypted with different keys and therefore increases data aggregation efficiency without compromising security. Currently, we are working on adding an integrity check mechanism to proposed scheme and implementing the proposed scheme to evaluate its security and performance.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks", IEEE Communications Magazine, 40(8), pp. 102-114, Aug. 2002.

[2] H. Chan and A. Perrig, "Security and Privacy in Sensor Networks", *IEEE Comp. Mag.*, Oct. 2003, pp. 10305.

[3] E. Shi and A. Perrig, "Designing Secure Sensor Networks", *Wireless Commun. Mag.*, vol. 11, no. 6, Dec. 2004 pp. 3843.

[4] K. Akkaya, M. Demirbas, and R. S. Aygun, "The Impact of Data Aggregation on the Performance of Wireless Sensor Networks", Wiley Wireless Communications and Mobile Computing (WCMC) Journal, Vol. 8 pp. 171-193, 2008.

[5] L. Hu and D. Evans, "Secure aggregation for wireless networks", Proc. of Workshop on Security and Assurance in Ad hoc Networks, Jan 28, Orlando, FL, 2003.

[6] B. Przydatek, D. Song, and A. Perrig, "SIA : Secure information aggregation in sensor networks", Proc. of SenSys'03, pp. 255-265, 2003.

[7] H. Çam, S. Ozdemir, P. Nair, and D. Muthuavinashiappan, and H.O. Sanli, "Energy-Efficient and secure pattern based data aggregation for wireless sensor networks", Special Issue of Computer Communications on Sensor Networks, pp. 446-455, Feb. 2006.

[8] W. Du and J. Deng and Y. S. Han and P. K. Varshney, "A Witness-Based Approach for Data Fusion Assurance in Wireless Sensor Networks", in Proc. GLOBECOM'03, pp. 1435-9, 2003.

[9] K. Wu, D. Dreef, B. Sun, and Y. Xiao, "Secure data aggregation without persistent cryptographic operations in wireless sensor networks", Ad Hoc Networks, vol. 5, no.1, pp. 100-111, 2007.

[10] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks", Proc. of ACM MOBIHOC'06, May 2006.

[11] S. Ozdemir, "Secure and Reliable Data Aggregation for Wireless Sensor Networks", LNCS 4836, H. Ichikawa et al. (Eds.), pp. 102-109, 2007.

[12] D. Westhoff, J. Girao, M. Acharya, "Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution and Routing Adaptation", IEEE Transactions on Mobile Computing, Vol. 5, No. 10, pp. 1417-1431, October 2006.

[13] S. Ozdemir, "Concealed Data Aggregation in Heterogeneous Sensor Networks using Privacy Homomorphism", Proc. of ICPS'07 : IEEE International Conference on Pervasive Services, pp. 165-168, Istanbul, Turkey, 2007.

[14] C. Castelluccia, E. Mykletun, G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks", Proc. of Conference on Mobile and Ubiquitous Systems: Networking and Services, pp.109-117, 2005.

[15] D. Wagner, "Cryptanalysis of an Algebraic Privacy Homomorphism", *in Proc. Sixth Information Security Conf. (ISC03)*, Oct. 2003.

[16] R.L. Rivest, L. Adleman, and M.L. Dertouzos, "On Data Banks and Privacy Homomorphisms", *Foundations of Secure Computation,* pp. 169-179, 1978.

[17] Crossbow Technologies Inc., http://www.xbow.com.

[18] J. Domingo-Ferrer, "A Provably Secure Additive and Multiplicative Privacy Homomorphism", *in Proc. Information Security Conf.,* pp. 471-483, Oct. 2002.

[19] N. Koblitz, A. Menezes and S. Vanstone, "The State of Elliptic Curve Cryptography", Journal of Designs, Codes, and Cryptography, vol. 19, pp. 173-193, March 2000.

[20] D. Boneh, Eu-Jin God, and K. Nissim, "Evaluating 2-DNF Formulas on Cipertexts", Proc. Theory of Cryptography Conference, LNCS vol. 3374, pp. 325-321, Jan 2005.

[21] W. Du, J. Deng, Y.S. Han, and P.K. Varshney, "A Key Predistribution Scheme for Sensor Networks Using Deployment Knowledge", IEEE Transactions on Dependable and Secure Computing, vol.03, no.1, pp. 62-77, January-March, 2006.

[22] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, "Handbook of Applied Cryptography", p. 128, CRC Press, 1997.

[23] S. Peter, D. Westhoff, and C. Castelluccia, "A Survey on the Encryption of Convergecast-Traffic with In-Network Processing", IEEE Transactions on Dependable and Secure Computing, vol. 99, no. 2.