Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **5**, September **2016**

Research on Secure Aggregation Scheme based on Stateful Public Key Cryptology in Wireless Sensor Networks

Danyang Qin , Shuang Jia, Songxiang Yang, Erfu Wang, Qun Ding

Key Lab of Electronic and Communication Engineering
Heilongjiang University
Harbin, 150080, China
qindanyang@hlju.edu.cn

Received April, 2016; revised June, 2016

Abstract. *Wireless sensor network (WSN) is an important part of the Internet of Things (IoT) and technical support for ubiquitous communication. Data aggregation technology reduces the network traffic overhead and extends the effective working time of the network, but the continued operation of the wireless sensor networks will increase the probability of the aggregation node being captured and probability of the aggregated data being tampered. Thus it will seriously affect the security performance of the network. For network security issues, this paper proposed a secure aggregation scheme based on Stateful Public Key Cryptography (SPKC) for WSNs, which employs a new stateful public key encryption to provide an efficient end-to-end security. Moreover, the security aggregation model will not impose any bound on the aggregation function property, so as to realize the low cost and high security level as the same time*
Keywords: Wireless sensor networks, Secure data aggregation, Homomorphic encryption, Power analysis.

1. **Introduction.** WSN (Wireless sensor network) is a major branch of the Internet of things, which is made up of many sensor nodes with constrained resource. Sensors will be affected by power supply, communication and computing power, so the data transmission is restricted [1]. Data aggregation is thought to be essential technique for WSN, since it may save the computation and communication energy effectively. With such a technique, the data will be captured by sensor nodes and fused by intermediate nodes and then transmitted to the sink nodes through the wireless link. Under the background of the Internet of things, the aggregation node is regarded as a network drive to collect aggregated data and sends them to the cloud [2]. In short, aggregation node manages all interactions between sensor network and the outside world. Data aggregation can be processed in the network, which will reduce the packet transmission and data redundancy and improve the overall life of the network [3].

2. **Stateful public key encryption and cryptographic methods.** The SPKC adopts the hybrid encryption algorithm, which is the combination of asymmetric and symmetric encryption algorithm. This part will briefly introduce the concept of encryption algorithm and some encryption tools, and then put forward the system model and the design target.

2.1. **Stateful public key encryption.** The stateful encryption can significantly reduce the computational cost of the traditional PKE (Public Key Encryption). In SPKE (Stateful Public Key Encryption), the sender uses a state repeatedly in different encryption algorithms. The state is held by the sender, and the same state is reused in different encryption processes. The use of stateful encryption greatly reduces the computational cost and the energy, because it will only calculate once. In this work, the SPKE is utilized to guarantee the convergecast traffic toward BS efficiently in the way that the state is adopted to share a secret with BS and the network nodes use the key to ensure the end-to-end security for aggregated data.

2.2. **Cryptographic methods.** In this paper, we use some cryptographic methods, such as homomorphic encryption, aggregate message authentication code, hash-based message authentication code, pseudo random function and key derivation function.

2.2.1. *Homomorphic encryption.* Homomorphic encryption allows calculation on ciphertext, the effect of which is the same as performing these calculations on the underlying plaintext data. If and only if (1) is founded, an encryption algorithm is considered to be homomorphic.

$$D\left(E\left(\mathrm{x}\right)\Delta E\left(\mathrm{x}\Delta y\right)\right) \tag{1}$$

The operation can support the addition and multiplication or support these two kinds of algorithms at the same time, depending on the characteristics of the encryption scheme. HE (Homomorphic Encryption) supports all the functional operation on the ciphertext, known as FHE (Fully Homomorphic Encryption) [4]. The other class of HE is PHE (Partially Homomorphic Encryption), which includes encryption schemes that have homomorphic property.

2.2.2. *Aggregate message authentication code.* It is clear that MAC (Message Authentication Code) cannot verify the additive property:

$$\mathrm{MAC}\left(\alpha+\beta\right)\neq\mathrm{MAC}\left(\alpha\right)+\mathrm{MAC}\left(\beta\right) \tag{2}$$

MAC can be aggregated using XOR (Exclusive Operation). The result allows integrity and authenticity with the condition to verify all the personal data.

$$\mathrm{MAC}_{agg}=\mathrm{MAC}_1\oplus\mathrm{MAC}_2\oplus\cdots\oplus\mathrm{MAC}_n \tag{3}$$

2.2.3. *Hash-based message authentication code.* HMAC (Hash-based Message Authentication Code) is a kind of scheme based on encryption hash function and it is always used to validate the data integrity and source. HMAC can be used for any iteration encryption hash function with a shared key. Encryption intensity of HMAC depends on the property of the underlying hash function. Let $\mathrm{HMAC}(K,m)$ denote the message digest of m using a key K, assuming that the underlying hash function is SHA-1, which will produce 20 bytes digest.

2.2.4. *Pseudo random function.* PRF (Pseudo Random Function) is a deterministic function [5]. PRF has two inputs, namely K and m, where K is a hidden key and m is a variable. Generally, the output of PRF by computing is different from the real output value.

2.2.5. *Key derivation function.* KDF (Key Derivation Function) takes a given key and a random number being as the input, and will generate a new key for the encryption algorithm. In this study, the HKDF (Key Derivation Function based on Hash message authentication code) NISTSP800-108 is adopted to generate a dynamic key.

3. **Secure data aggregation models.** SPKC proposed in this paper consists of two main phases, namely the forwarding phase and the aggregation phase. In the former phase, all the sensors will send their states for aggregation phase. In the latter phase, the sensor nodes will encrypt and verify the captured data using states shared with BS. Then, CH use homomorphic operation and XOR to aggregate ciphertext and signature respectively in order to generate a cipher and a new signature. Finally, BS verifies the aggregated data, decrypts aggregation, retrieves plaintext and calls the verification process.

3.1. **Setup phase.** ECC for SPKE is an effective secure method. Suppose BS generates a pair of keys (x, Y) before deployment, where $Y = xG$, and it will keep the private key x secret. Each sensor S_{ij} carries keys shared with BS. Elliptic curve parameters that are the set (Y, E, p, G, n) , where Y is the elliptic curve public key and E is the elliptic curve over prime field p with the base point G of order n. Sensors are also loaded with M, while KDF is on the basis of PRF (NIST SP800-108 HKDF) and a secure MAC (HMAC).

3.2. **Forwarding phase.** In forwarding phase, the sensor nodes will send state St_{ij} to produce keys required by aggregation phase. HKDF is adopted to obtain authentication keys. In fact, the output K_{ij} of the PRF is computed with a nonce as the iteration variable, and then used as keying material for authentication. Each sensor S_{ij} executes Algorithm 1 sending the output St_{ij} and MAC_{ij} to the next hop. In order to extract all the keys from base stations, all packets must be sent. Therefore, at the stage, the CH acts as data forwarder but not a data aggregator, as shown in Figure 1(a).

Algorithm 1: Forwarding phase (S_{ij})

Input: (Y, E, p, G, n), SK_{ij}^{BS}, Nonce

1.Generate a random $r_{ij} \in [1, n-1]$

2.Compute $St_{ij} = r_{ij}G$

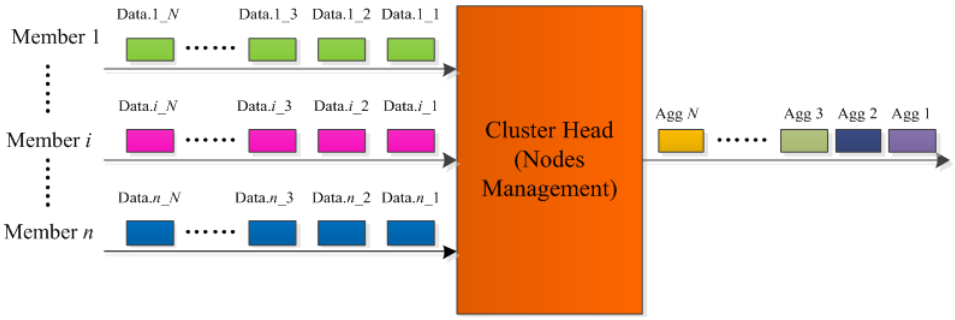
3.Compute $K_{ij} = HKDF\left(St_{ij} \| r_{ij}Y \| SK_{ij}^{\mathrm{BS}}, N_{ij}\right)$

4.Compute $\mathrm{MAC}_{ij} = \mathrm{HMAC}\left(St_{ij}, K_{ij}\right)$

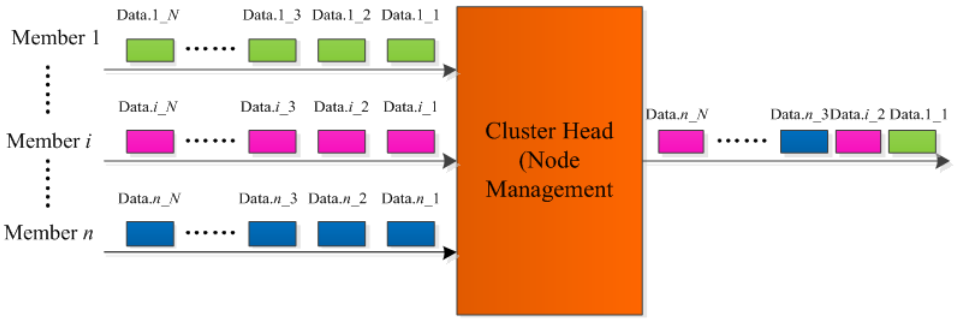
Output: St_{ij}, MAC_{ij}

Each sensor will keep the state (r_{ij}, St_{ij}) and use the state to encrypt. Once the state is received, CH forwards all data to BS or to the nearest CH. Then, BS will verify the integrity by using a private key to identify the entire senders. The verification is done by calculating all the keys corresponding to the received states (see Algorithm 2). If the verification is established, the corresponding state will be stored in BS's database, and is used to decrypt and verify. Otherwise, the state will be rejected.

3.3. **Aggregation phase.** The aggregation phase consists of three steps: encryption, aggregation and verification. These steps work as follows.



(a) Forwarding phase



(b) Aggregation phase

FIGURE 1. Data transmission in SPKC

Algorithm 2: Verification（BS）

Input: (Y,E,p,G,n), SK_{ij}^{BS}, x, Nonce, all pairs $(St_{ij},\mathrm{MAC}_{ij})$

1.**For** each $i\in\{1,...,L\}$, $j\in\{1,...,R\}$

1.1.Compute $K_{ij}=\mathrm{HKDF}\left(St_{ij}\|x_{ij}St_{ij}\|SK_{ij}^{BS},N_{ij}\right)$

2.**For** each $i\in\{1,...,L\}$, $j\in\{1,...,R\}$

2.1.Compute $\mathrm{MAC}_{ij}=\mathrm{HMAC}\left(St_{ij},K_{ij}\right)$

2.2.**If** $\mathrm{MAC}'_{ij}=\mathrm{MAC}_{ij}$,

Then accept

Otherwise reject

Output: MAC verification

3.3.1. *Encryption.* the encryption step, the data m_{ij} is captured by S_{ij} , and it is encoded before encryption. The resulting code e_{ij} uses symmetric encryption. In aggregation phase, HKDF outputs two keys, namely K_{ij1} and K_{ij2} , where $K_{ij1} < M$. The sensors encrypt the encoded plaintext and use K_{ij1} and K_{ij2} to calculate the corresponding MAC respectively. The encryption is performed using addition modulo the large number M (see Algorithm 3), where M must be greater than $e_{agg} = \sum_{i=1...L}^{j} e + ij$. If this property is verified, the decryption will result in a message e_{agg} that is smaller than M .The nonce N_{ij} used in HKDF ensures the dynamic keys needed for the security of encryptions. The MAC is then calculated on ciphertext. Finally, the ciphertext and MAC are sent to the corresponding CH.

Algorithm 3: Encrypt&sign (S_{ij})

Input: m_{ij} , (r_{ij}, St_{ij}) , Y, M, Nonce

1.Encode m_{ij} into $e_{ij} = m_{ij} \| 0^z$, where $z = \lambda \cdot (i-1)$

2.Comput $K_{ij} = \mathrm{HKDF}(St_{ij} \| r_{ij} Y \| Y, N_{ij})$, where $K_{ij} = K_{ij1} \| K_{ij2}$

3.Compute $C_{ij} = K_{ij1} + e_{ij} \bmod M$

4.Compute $\mathrm{MAC}_{ij} = \mathrm{HMAC}(C_{ij}, K_{ij2})$

Output: C_{ij} , MAC_{ij}

3.3.2. *Aggregation.* In the aggregation step, CH acts as the data aggregators and uses dispersion aggregation. For multi-user, the data are randomly aggregated, the process of which is shown in Figure 1(b). CH will aggregate all ciphertext including its own ciphertext aggregated into a cipher $C_{agg}, L-1$ MACs and its own MAC into one MAC_{agg} (see Algorithm 4). Using addition operation modulo M , ciphertext is homomorphically aggregated, and MAC [6] is XORed. After that, the output of Algorithm 4 will be sent to BS or the nearest CH. CH receives the packets from another CH, and forwards the packet to BS. The execution homomorphism aggregation of each CH_j is as follows:

3.3.3. *Verification.* In the verification step, BS will call the decryption and verification process for the aggregation of each cluster after the data packets being received. BS will first calculate the current keys corresponding to all network nodes using the state stored in the database. Then, BS will decrypt the aggregated ciphertext, and retrieve the personal plaintext (see Algorithm 5). Finally, BS will calculate $(C_{ij}, \mathrm{MAC}_{ij})$, and check the integrity of the end-to-end information. If the verification is established, the aggregated data e_{agg} will be accepted, otherwise it will be rejected. BS will send $(C_{ij}, \mathrm{MAC}_{ij})$ by notifying CH_j, and verify each pair of nodes to determine the malicious nodes. Another advantage of the model is that the node does not need to send a response to BS, because all the sensors are involved in the aggregation [7]. In SPKC, even without sensing data, each sensor will produce the encryption and sign MAC. In Algorithm 3, the non-responding node will simply use the zero value of m ; thus, after sending all the sensor data to BS,

Algorithm 4: Homomorphic aggregation $\left(\mathrm{CH}_j\right)$

Input: All pairs $\left(C_{ij}, \mathrm{MAC}_{ij}\right)$, where $i \in \{1,...,L\}$

1.**For** L ciphertexts $\left(C_{1j}...C_{Lj}\right)$

1.1.Compute $C_{agg} = \sum_{i=1...L}^{j} C_{ij} \bmod M$

2.**For** LMACs$\left(\mathrm{MAC}_{1j}...\mathrm{MAC}_{Lj}\right)$

2.1.Compute $\mathrm{MAC}_{agg} = \oplus \mathrm{MAC}_{ij}$

Output: C_{agg}, MAC_{agg}

it can perform any aggregation function, which is a major advantage of hop solutions. SPKC is flexible and does not impose any constraints on the property of the function.

Algorithm 5: End-to-end (BS)

Input: All pairs $\left(C_{agg}, \mathrm{MAC}_{agg}\right)$, where $j \in \{1,...,R\}$

1.Compute $K_{ij} = \mathrm{HKDF}\left(St_{ij} \| x_{ij} St_{ij} \| Y, N_{ij}\right)$

2.**For** each pair $\left(C_{agg}, \mathrm{MAC}_{agg}\right)_j$

2.1.Compute $e_{agg} = C_{agg} - \sum_{i=1...L}^{j} K_{i1} \bmod M$

2.2.Encod$\left(e_{agg}, L, \lambda\right)$: $m_i = e\left[(i-1)\cdot\lambda, \lambda\cdot i - 1\right]$, where $i = 1,...,L$

2.3.**For** m_i, where $i = 1,...,L$

2.3.1.Compute MAC_i

2.4.Compute $\mathrm{MAC}'_{agg} = \oplus \mathrm{MAC}_i$

2.5.**If** MAC'_{agg}=MAC_{agg}

Then e_{agg} is accepted

Otherwise e_{agg} is rejected

Output: MAC verification

Aggregation stage is to be executed for many times until the state is failed. Therefore, a deadline has been designed in this paper. Key expiration is a very important security

measurement, which allows key refreshing and involves a new forward stage so as to increase the security theoretically. In SPKC, a node can be seen as a lifetime sequence epoch, and each period consists of two stages, namely forwarding and aggregation stages (see Figure 2). In fact, the security level can be used to judge whether to meet the required security level of sensitive information. Therefore, the key refreshing is very important for security improvement before deadline, and it may be pre-installed in the sensor practically. Further, due to some faults, some nodes' synchronization may be lost. Then, the synchronization of new forwarding phase can be refreshed. In this case, the base station will request a new forwarding phase by using a specific active message.

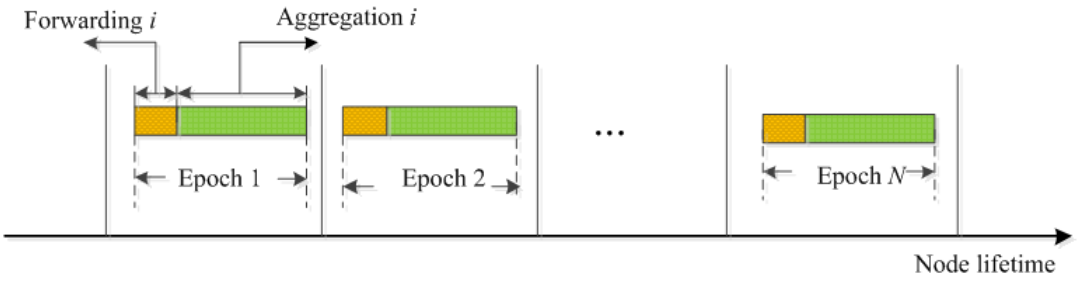


FIGURE 2. Node lifetime

4. **Simulation results and performance analysis.** he confidentiality and integrity, as well as the performance evaluation of SPKC will be analyzed in terms of computation and communication overhead, energy consumption, scaling and portability.

4.1. **Security performance analysis.** SPKC proposed in this paper will increase the data aggregating efficiency and enhance the information security performance at the same time. The metric of average packet successful delivery rate (PSDR) [8] with different number of attackers is adopted to compare different security algorithms. The number of nodes remains 200 with the attack type as the classical SPA (Simple Power Analysis), and the security aggregating algorithms are SHA[9], EVCDA[10] and SPKC. The simulating results in Figure 3 show that the average PSDR will decrease with the number of attackers increasing for all the algorithms. The average PSDR of SPKC, however, is superior to the other two obviously. Especially when the number of attackers approaches to 10% of the total nodes, SHA and EVCDA are not available, while it is still over 40% with SPKC. The error packets are reduced by SPKC since the bidirectional malware detection technology to eliminate malicious node cluster members and CH. Therefore, the security performance can be well provided by SPKC.

4.2. **Communication overhead analysis.** In the forwarding stage, for the non-CH node, the complexity of communication is $O(1)$. For CH node, its complexity is $O(L)$. Because all the data packets are sent to BS, so the total communication overhead at this stage is $R(2L-1)$. In the aggregation stage, the data are aggregating toward the BS to which each sensor sends a data packet to form aggregating communication streams. The complexity of both non-CH and CH node are $O(1)$. Thus, the total cost of communication is N. We consider the case where CH is directly connected to the BS. Otherwise, the costs may increase, and the increase will depend on the depth, the number L and the actual phase.

In terms of communication overhead and energy consumption, the new simulation results, experiments and analysis are presented in the previous sections. The results confirm

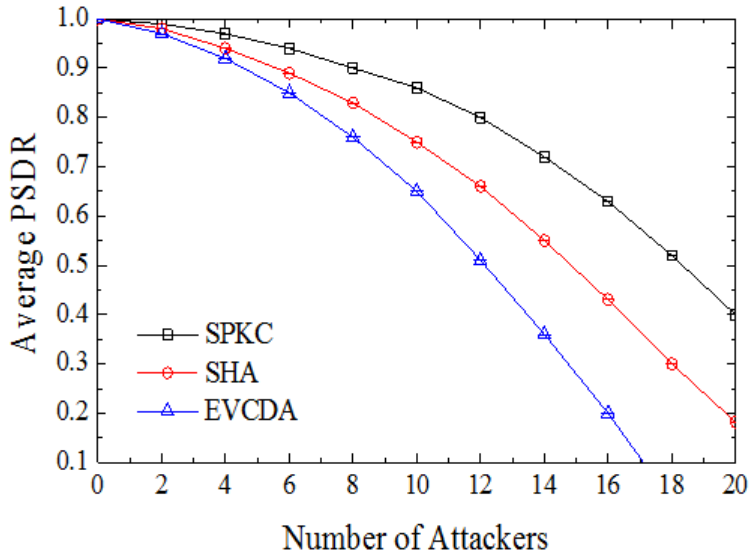


FIGURE 3. Comparison of average PSDR with different number of attackers

the suitability of the protocol and ensure the minimum cost of high level on security. In order to make more correct and accurate results, we simulate in three scenarios, SHA, EVCDA and SPKC. We choose to compare these two options because they offer the same level of security as SPKC. In the three simulation scenarios, 50, 100 and 150 sensor nodes are randomly deployed within a square area with a single base station centered.

In SPKC, the state is a point belonging to an elliptic curve, and it is transmitted in a compressed way and requires only $(1 + \log_2 p)$ bits. In the aggregation stage, we use 20-byte encoded plaintext to produce 20-byte ciphertext. Therefore, we only use short packets to provide both end-to-end confidentiality and integrity. The cost can be calculated by the following ways for transmitting a corresponding data packet, which corresponds to two different methods: (i) sending longer messages, which will increase the bit error rate and reduce the reliability, (ii) splitting the packet into blocks and sending each block separately which will incur not only delay but also additional cluster head overhead. However, no matter how to be used in these scenarios, SPKC will produce less energy consumption. To illustrate this point in a sense, three simulations are run with 500 seconds for each. After the simulations, the average of each result will be obtained. The data are sent to the BS using a simple clustering algorithm based on TDMA. 4, 8 and 12 cluster heads (channel) are selected for three typologies with number of nodes as 50, 100 and 150. Therefore, the number of nodes per cluster is $L \in [8, 13]$. We note that for the same simulation time, an expiration date is considered 5 times in 5E (5 Epochs) and 10 times in 10E (10 Epochs). The results of different models are shown in Figure 4.

Comparing with other models, the simulation results show that SPKC will bring lower communication overhead. Due to the use of the stateful public key encryption, the data using symmetric encryption can produce short ciphertext, resulting in short packets. Even in SPKC, security is improved, its overhead is acceptable. In addition, the advantage of using SPKE can be viewed when the network density increased, as shown in Figure 4 (b) and (c).

4.3. **Energy consumption analysis.** We perform the simulations for three models, with each simulation lasts for 500s, to estimate the energy consumption in the entire network. The results at non-CH and CH node are shown in Figure 5(a) and 5 (b) respectively.

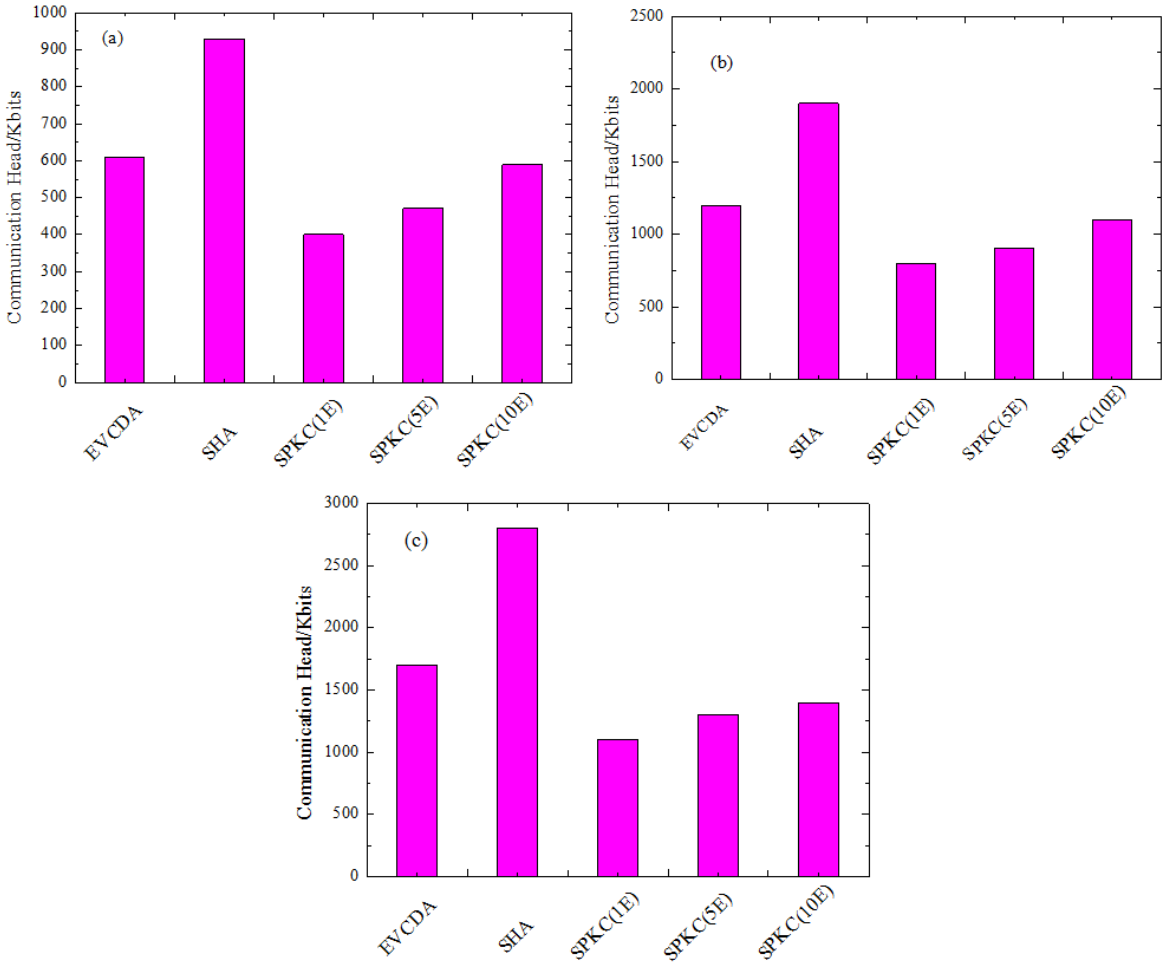


FIGURE 4. The communication overhead for different typologies: (a) 50 nodes, (b) 100 nodes and (c) 150 nodes

In Figure 5(a), the comparison with other models shows that SPKC will greatly reduce the energy consumption. Due to the use of symmetric primitive and the effective implementation of asymmetric operations, there is less computational overhead in SPKC. Thus, for providing the same level of security, network lifetime is infinitely increased. Figure 5(b) shows that the average energy, no matter at CH or at non-CH, has significant reduction in SPKC. On the one hand, the node will continuously perform a large amount of calculation in EVCDA, while the complex operations of SHA are simple in forwarding stage; on the other hand, in addition to participate in the aggregation process, CH also performs an aggregate function (same-state operation). In SHA and EVCDA, the operation requires the addition on the elliptic curve, while it requires only a small amount of calculation in SPKC.

The simulation results show that the energy cost of communication using asymmetric calculation can be ignored. However, if the energy required for cryptographic computations is reduced, the cost will become crucial. By saving two ECC scalar multiplications, SPKE can reduce the computational cost of the traditional PKE. Therefore, we need to consider the cost of communication to highlight the advantages of using data aggregation in wireless sensor networks. In Table 1, the estimated total energy costs of SHA, EVCDA

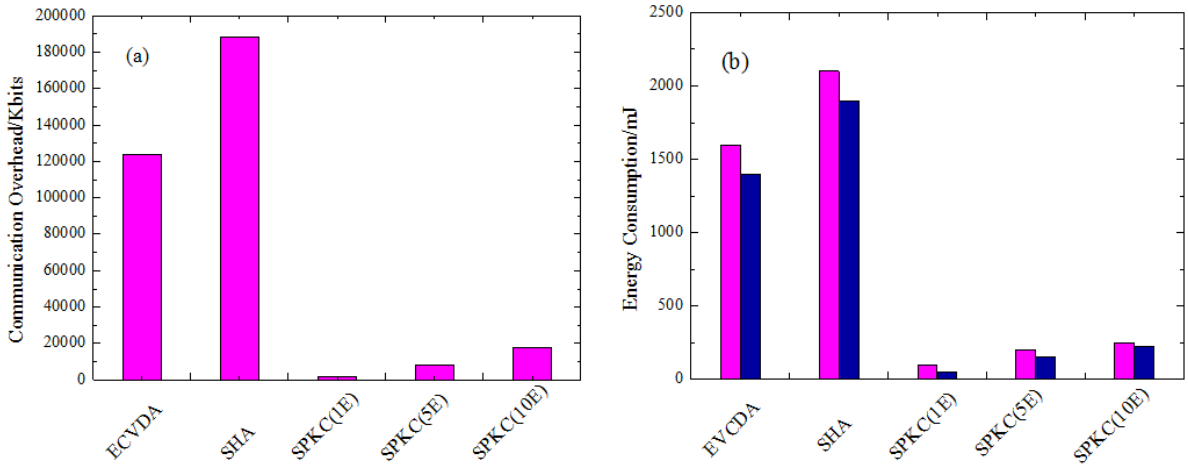


FIGURE 5. The estimated total energy: (a) in the whole network, (b) at CH and non-CH nodes in a network of 100 nodes

and SPKC at the CH node and non-CH node are presented. The results show that, compared to SPKC, computations compose almost the overall cost of SHA and EVCDA on TelosB mote.

TABLE 1. The estimated total energy costs (in mJ) of SHA, EVCDA and SPKC at CH and non-CH node in a network of 100 nodes

Scheme		CH	non-CH
SPKC(1E)	Comm	33.446(60%)	15.029(48%)
	Comp	21.999(40%)	16.649(48%)
	Total	55.465	31.678
SPKC(5E)	Comm	35.711(32%)	15.191(20%)
	Comp	77.461(68%)	63.038(80%)
	Total	113.172	78.229
SPKC(10E)	Comm	38.103(21%)	15.284(11%)
	Comp	141.145(79%)	128.369(89%)
	Total	179.248	143.653
EVCDA	Comm	39.007(3%)	16.748(1%)
	Comp	1497.143(97%)	1408.692(99%)
	Total	1536.15	1425.44
SHA	Comm	90.744(4%)	27.488(1%)
	Comp	2038.263(96%)	1950.672(99%)
	Total	2129.007	1978.16

5. **Conclusion.** Wireless sensor network is an important component of modern communication systems, and the security of the network is an important guarantee for the success rate of data transmission, so the study of WSN security is very important and necessary. In this paper, we propose a secure data aggregation model based on stateful public key namely SPKC, which uses an addition homomorphic encryption and aggregated MAC to provide end-to-end confidentiality and integrity. Experimental and simulation results

confirmed the efficiency of SPKC in terms of security and energy, comparing with traditional models. Moreover, relative data and curves illustrate that SPKC can achieve security level of higher performance and produce very little energy consumption. In the topology of the 100 nodes, SPKC generates energy consumption in the cluster head node only 179.248mJ, while the energy consumption is 1536.15mJ for EVCDA and 2129.007mJ for SHA. Therefore, SPKC is able to achieve an efficient low-power and safe data aggregation. Future research will extend to the network with moving nodes and will consider new attacks such as selective forwarding in order to provide the best way for subsequent research on aggregated data. Some of the results will provide ideas for robust multihop routing in future ubiquitous communication network.

REFERENCES

[1] T. K. Dao, T. S. Pan, T. T. Nguyen, S. C. Chu, A compact articial bee colony optimization for topology control scheme in wireless sensor networks, *Journal of Information Hiding and Multimedia Signal Processing* , vol. 6, no. 2, pp. 311-322, 2015.

[2] Y. H. Huang, K. H. Fan, W. S. Hsieh, Message authentication scheme for vehicular ad-hoc wireless networks without RSU, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 1, pp. 113-122, 2015.

[3] F. C. Chang, H. C. Huang, A survey on intelligent sensor network and its application, *Journal of Network Intelligence*, vol. 1, no. 1, pp. 1-15, 2016.

[4] T. T. Nguyen, T. K. Dao, M. F. Horng, C. S. Shieh, An energy-based cluster head selection algorithm to support long-lifetime in wireless sensor networks, *Journal of Network Intelligence*, vol. 1, no. 1, pp. 23-37, 2016.

[5] K. C. Egemen, B. Dan, D. Amit, Modelling communication network challenges for future internet resilience, survivability, and disruption tolerance: a simulation-based approach, *Telecommunication Systems*, vol. 2, no. 6, pp. 751-768, 2013.

[6] M. Liu, H. G. Gong, Y. C. Mao, Collection and aggregation protocol on high efficiency and energy saving of sensor network data. *Journal of software*, vol. 16, no. 12. pp. 2106-2116, 2012.

[7] V. Kumar, S. K. Madria, Secure hierarchical data aggregation in wireless sensor networks: performance evaluation and analysis. *Proceeding of the IEEE 13th International Conference on Mobile Data Management*, pp. 196201, 2012.

[8] J. Rabaey, J. Ammer, Distributed Framework for Correlated Data Gathering in Sensor Networks, *IEEE Transactions on Mobile Computing*, vol. 57, no. 1, pp. 578-593, 2010.

[9] J. Albath, S. K. Madria, Secure hierarchical data aggregation in wireless sensor networks, *Proceeding of Wireless Communications and Networking Conference*, pp. 16, 2009.

[10] H. M. Sun, Y. H. Lin, Y. C. Hsiao, C. M. Chen, An efficient and verifiable concealed data aggregation scheme in wireless sensor networks, *Proceeding of the International Conference on Embedded Software and Systems*, pp. 1926, 2008.