Acta Technica *62* (2017), No. 6A, 393–404

An optimized aggregation for privacy-preserving data based on homomorphic mac and stochastic data segmentation in wsn

Zhang Zhongxian[1]

Abstract. In the existing aggregation of privacy-preserving data, it is that either the security strength is not high, or the data integrity is not effectively guaranteed, or the communication overhead and the computation load is high, or the aggregation accuracy is low. In order to solve this problem, this paper has combined the encryption and authentication of homomorphic MAC and the technique of stochastic data segmentation to come up with an improved scheme of aggregation of privacy-preserving data. It has effectively reduced the times of encryption and decryption of the data, and the integrity of the aggregation result is guaranteed. At the same time, It not only can reduce the network communication overhead of the nodes, but also can strengthen the protection strength for privacy-preserving data.

Key words. data aggregation, privacy-preserving data, data segmentation, homomorphic, fusion accuracy..

1. Introduction

The effective ways to reduce the energy consumption of WSNs is the aggregation of data [1]. Due to the openness of network deployment and the sharing of channel, and the deployment of the high-complexity algorithm for the encryption is limited by the computing power of the nodes, WSNs is difficult to protect the privacy-preserving data and the integrity of the data aggregation results. It is necessary to deploy a lightweight algorithm for the aggregation of privacy-preserving data with a low- complexity encryption of data. The key to protect privacy-preserving data is to get control rights for the data of the nodes [2]. But the early algorithm of data aggregation was focused on reducing data transmission, computing load, such as TAG[3].The literature [4] uses a private seed to protect the data in aggregating,

[1]College of Information Engineering, Zhangzhou Institute of Technology, Zhangzhou, Fujian, China

it's traffic load and computing load are low, but the protection of privacy data is not high. The paper [5] introduces the cryptography of the elliptic curve in data aggregation, but the integrity of the aggregation results cannot be verified. The literature [6] presents a protection of the privacy data that can guarantee the integrity of data, but high computing load. The literature [7] introduces the technique of data segmentation and mixing for the aggregation of privacy-preserving data that be called SMART, it's is effective to protect privacy-preserving data in the network. But SMART scheme distributes large slices in the data segmentation phase, and have high communicate overhead, and the accuracy of the aggregation is obviously decreased also. Literature [8] using the technique of the homomorphic encryption for the aggregation of privacy-preserving data, it can protect the data integrity and improve the aggregation accuracy of data, but the computing load and communication load are large. Literature [9] came up with a mechanism for the encryption and authentication, named as homomorphic MAC, has very good security strength and lower computing- load of nodes.

On the Basis of SMART, this paper come up an improved aggregation of the privacy- preserving data based on the mechanism of end-to-end homomorphic MAC, and based on the technique of the stochastic data segmentation, called HMSS-SMART. It can effectively protect the privacy-preserving data and the integrity of the results of the aggregation data, can also effectively reduce operation load and communication overhead, and can improve the accuracy of data aggregation.

2. Design for HMSS-SMART

2.1. Assumptions for network

As shown in Figure 1, this scheme uses the network topology of aggregation tree of the TAG [3]. The nodes will divide into base-stations, aggregation-nodes and leaf-nodes. The base-station is responsible for decrypting cipher text and verifying the integrity of data. The leaf-node is perceiving the data and sending data back to its aggregation-node. The aggregation-node is a special leaf-nodes, it can sense data, can also transmute command of capture from base-station to leaf-nodes, still can collect and aggregate the data of leaf-nodes to base-station. Assuming the maximum number of data-slices that any node can generate is $M = 4$, the communication of nodes each other is only one hop ($h = 1$). Suppose attacker can intercept the privacy-data and destroy the integrity of the results of aggregation in the network. Because the encryption algorithm for the large prime factorization can be obtained at a small price to obtain a stronger anti-analytic ability, so the network uses the mechanism of homomorphic encryption based on problem of large prime factorization.

2.2. Design ideas

(1) Using the technique of random segmentation and mixture of the data to reduce the high communication overhead of the SMART scheme, and then improve the accuracy of aggregation data. Because it is difficult to guess the number of

data-slices, it makes that eavesdroppers recover original data is harder, and then the privacy-preserving data is much less likely to be wiretapped than SMART.

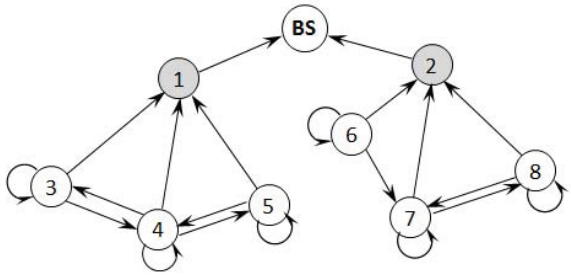

Fig. 1. AggregationTreeExample (h = 1, M = 4)

(2) Additive data aggregation has the homomorphic conditions of homomorphic encryption. So, we can use the technique of end-to-end authentication and encryption of homomorphic MAC to put together the decryption and integrity certification of the result in the base-station, and then reduce the times of encryption and decryption of nodes in data segmentation, data mixing and data aggregation. It can effectively protect against malicious attacks from both inside and outside the network, and also reduce the overhead of the nodes.

(3) The homomorphic MAC includes signature, aggregation, and validation of three time algorithms of probability polynomial [9]. Data segmentation and mix includes segmentation, mixing, and aggregation stage [7].Therefore, the working process of the aggregation of privacy-preserving data for HMSS-SMART can be divided into seven stages, such as network building, signature and data encryption, data segmentation, mixing, data aggregation, data decryption and integrity verification.

2.3. The working process of the improved aggregation scheme

(1) Network construction

Let D_i is the information for the node I, ρ_i is the information weight of node I, and id_i is the only identity of D_i. By the definition of the homomorphic MAC [9], D_i can be changed into n-dimensional vector space on the finite domain F_q^n, that is: $D_i = (D_{i1}, D_{i2}...D_{in}), D_{ij} \in F_q$. Before the network is deployed, the base-station and sensor nodes have preset a pseudo-random generator GEN and a pseudo-random function RND on the finite domain F_q^n. So, in the early times of the network deployment, we can generate the MAC key-pair (sk_1, sk_2) shared by the base-station and the nodes through them. Let the large prime numbers a and b are the private keys of the mechanism of homomorphic encryption for the problem of the big prime factorization, and then the public key $w = a*b$. At this stage, each vector component of the plaintext Di has also generated the corresponding private key-pair (u_i, v_i),here $i \in [1, n]$, n is the dimension of the vector space of the clear text D_i .

(2) Data encryption

After the initialization of the parameter described above is complete, the nodes

begin to generate cipher text and labels ti of the homomorphic information of the original data with the undermentioned operations. And then, it forms a new cipher-packet C_i with the weight ρ_i of the perceived information. Finally, C_i will be returned to its parent aggregation-node. Homomorphic MAC has three algorithms of probabilistic polynomial about time, such as signature, aggregation and algorithm. Here the algorithm of probability polynomial of signature about time is called $Sign(Sk_1, Sk_2, id_i, D_i, I) = t_i$, it is used to generate the information label t_i for the plaintext m_i. Substitute sk_1, sk_2, id_i into the pseudo-random generator of GEN and pseudo-random function of RND. After get the vector space y and the integer x_i on the finite field F_q, and then put the x_i, y in the function of signature algorithm, can get the following formula:

$$x_i + y * D_i = t_i \,. \tag{1}$$

Here $y = GEN(Sk_1), x_i = RND(Sk_2, id_i), y * D_i = y_1 * D_{i1} + y_2 * D_{i2} ... + y_n * D_{in}$. The encryption algorithm is called $ENC(u_i, v_i, a, b, D_i)$, this means is using the preset key-pair u_i, v_i, a, b to encrypt the plaintext D_i. The formula is as follows:

$$ENC(u_i, v_i, a, b, D_i) = (u_1.D_{i1} \ mod \ a, v_1.D_{i1} \ mod \ b) ... = (\alpha_{i1}, \beta_{i1}) ..., (\alpha_{in}, \beta_{in}) \,. \tag{2}$$

(3) Data segmentation

The nodes randomly split the cipher-packet C_i into m slices of data, then keeps one slice at random and sends the rest to the neighboring node randomly. Here m is an independent random event within [2, m]. Its distribution of probability is the function F (M, m) shown in the formula (12).The data-slices of node are shown in Figure 1. Let the slice is called s_{ij}, here i and j is the ID of node that send and receive slice. For example, the random number of data-slices from node 5 is 3, s_{55} is a reserved slices, and the other are randomly distributed to node 1 and node 4 for s_{51} and s_{54} data-slices.

(4) Data mixing

After the data segmentation is complete, the node will be mixed with its own data-slice and all other data-slices that it receives, and generate new packets. If U_j is ID-set of the data-slices from the node j, Q_j is a new packet generated by the data-slices in the node j, the mixed operation expression for the data-slice is:

$$Q_j = \sum_{j \in U_j} S_{ij} \,. \tag{3}$$

(5) Data aggregation

In this stage, the method of data fusion is an additive aggregation; the function is $Agg((t_1, m_1, \rho_1), ..., (t_n, m_n, \rho_n)) = (\sum_{i=1}^{n} \rho_i \cdot m_i, \sum_{i=1}^{n} \rho_i \cdot t_i) = (m, t)$. Let ID-set of child node is R, the function of the fusion for Cipher and labels of aggregation-node

is $Agg(C_i)$, and can get the following formula of aggregation algorithm:

$$t = \sum_{i \in R} t_i \cdot \rho_i, \beta = \sum_{i \in R} C_i' \cdot \rho_i \,. \tag{4}$$

As Figure 1, the data is converged to the aggregation-node 1 and 2 by leaf-nodes. In order to be able to receive the cipher, aggregation-nodes will wait a while. And then, do additive aggregation, integrate the aggregated label t_i, cipher C_i, and node weight ρ_i into the new packet C_i. And finally, C_i will be back to the base-station.

(6) Data decryption

The aggregation results of data from aggregation-nodes are encrypted. If want to get the aggregation results and verify the integrity of the data aggregation result, the base-station must be decrypt the aggregated data in the decryption function by the weight of the node ρ_i, the key-pair of a, b and the modulo-inverse of the key-pair. If D is the result of data aggregation after decryption, then:

$$D = \sum_{i \in R} Dec(C_i) = \sum_{i \in R} (\alpha_i^{-1} \cdot b \cdot b^{-1} + \beta_i^{-1} \cdot \alpha \cdot \alpha^{-1} \; mod \; v) \,. \tag{5}$$

(7) Integrity verification

According to the principle of homomorphic encryption, in this phase, the base-station can recomputed the aggregation result of the label, and compare it with the decrypted aggregation results of label data. If they are equal, the data is intact.

$$y = G(Sk_1), x = \sum\nolimits_{i=1})^n F(Mk_2, id_i) \cdot \varepsilon_i, t' = x + y * \sum_{i=1}^{n} m_(i) \; mod \; q \,. \tag{6}$$

3. The simulation and analysis

This paper mainly will simulate about the protection probability of privacy-preserving data, communication overhead, computing load and aggregation accuracy for HMSS-SMART and SMART. As a reference, the simulation will also consider the TAG scheme as a typical mechanism of unsecured data protection. The simulation environment is the simulator embedded by Tiny OS, the 200 sensor nodes will randomly assign to the rectangular area of 200m* 200m, the probability of becoming an aggregation-node is $P_s = 0.3$.

3.1. Exposed probability of privacy-preserving data

In SMART scheme, assuming that the original data of the node is fixed with M pieces of data-slices and the number of data-slice received by node is n. If the in-degree of node is marked $in - degree$ and the out-degree of node is marked $out - degree$, then $out - degree = M - 1, in - degree = n$. Because the necessary condition for the node's privacy-preserving data can be recovered to original data by attacker is that all of the incoming links and all of the outgoing links are eavesdropped, so if

the probability of a communication link be eavesdropped is q, the maximum value of in-degree is id_{max}, and the probability that the in-degree will be n is $P_{(ind=n)}$, and then the node's exposure probability $P_{smt}(q, M)$ of privacy-preserving data in the SMART is:

$$P_{smt}(q, M) = q^{M-1} \sum_{n=0}^{id_{max}} (P_{(id=n)} q^n) . \tag{7}$$

In the formula (7), $\sum_{n=0}^{id_{max}} (P_{(id=n)} q^n)$ is the probability that all of the incoming links of the node will be eavesdropped; q^{M-1} is the probability that all of the outgoing links of the node will be eavesdropped; and Id_{max} depends on M.

In the HMSS-SMAR, let the limit of the number of data-slices of node is M, and the actual data-slice is m, $P_{(od=m-1)}$ is the probability of that the number of outgoing link is $m-1$, other parameters with SMART. So, the probability of that all incoming links are eavesdropped is $\sum_{n=0}^{id_{max}} (P_{(id=n)} q^n)$, the probability of that all outgoing links are eavesdropped is $q^{m-1} \sum_{m=2}^{M} (P_{(od=m)}))$, then the expression for the probability $P_{hsmt}(q, M)$ of the node's privacy-preserving data is:

$$P_{hsmt}(q, M) = \sum_{n=0}^{id_{max}} (P_{(id=n)} q^n) q^{m-1} \sum_{m=2}^{M} (P_{(od=m-1)}) . \tag{8}$$

Different from the SMART, the value of m in the formula (8) is obeyed by the probability distribution function $F(M, m), (m \in [2, M])$. If the ratio of the unexposed probability of the privacy-preserving data and the number of data-slice is defined to be safe and cost-effective, the function of the safe and cost-effective of the nodes under different m is $f(m)$. And then expression for $F(M, m)$ is:

$$F(M, m) = \frac{f(m)}{\sum_{m=2}^{M} f(m)} . \tag{9}$$

In formula (9), f(m) is:

$$f(m) = \frac{1 - P_{smt}(q, m)}{m} = \frac{1 - q^{m-1} \sum_{n=0}^{id_{max}} \left(P_{(id=n)} q^n\right)}{m} . \tag{10}$$

In formula (9), $1 - Psmt(q, m)$ is the non-exposed probability of the node's privacy- preserving data under different m. According to the formula (9) and the formula (10), when M is equal to 6, and q is 0.1%, can see that the probability distribution for different m have those peculiarities: The characteristics of probability distribution of function $F(M, m)$ is: the smaller the m, the higher the probability of m occurring. And according to formula (10) again, the smaller the number of data-slices, the more the safety and cost-effective of nodes. Because of $m \in [2, M]$, so the technique of random data segmentation can achieve higher performance of data security with smaller communication overhead. IN this simulation environment, we set $M = 4, P_s = 0.3$.

According to the above formula of the exposed probability of privacy-preserving

data, can get the simulation results shown in Figure 2(Because of TAG scheme has not mechanism for privacy-preserving data aggregation, so here is only SMART and HMSS-SMART.).The simulations show the exposed probability of privacy- preserving data of HMSS-SMART is significantly lowers than SMART. There are two reasons: First, HMSS-SMART has reduced the transfer between aggregation-nodes. According to the formula (7) and (8), there is the lower the transmission of data, and the lower the exposed probability of privacy-preserving data. Second, due to the slices of HMSS-SMART is random, so it's very difficult for an eavesdropper to guess the number of the slices, and are harder to recover the original data completely.

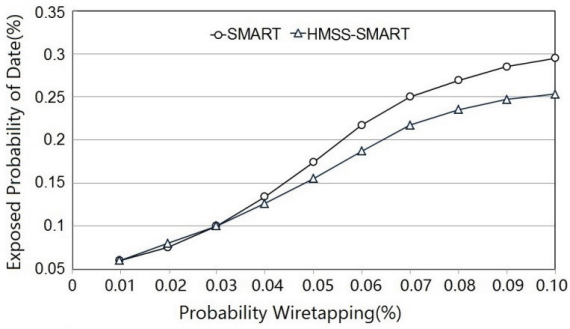

Fig. 2. Exposed Probability of Privacy-preserving data for SMART and HMSS-SMART

3.2. Computing load

Because the key-pair generation, Cipher decryption and integrity verification are done in base-station, so they do not produce computing load of nodes. In the two schemas, except for that the aggregation-nodes of HMSS-SMART reduce encryption and decryption each time, the other is basically the same. So, in the simulation experiment, we will ignore the computing load of the aggregation-node of SMART. In order to simplify the operation, assuming the data segmentation, sending, receiving and mixing are equal to one standard unit of computing load. Assuming that encryption and decryption have the same weights $k_{enc} = 5$, and N is the number of nodes, P_s is the probability of nodes becoming aggregation-node.

In SMART, each leaf-node will encrypt and receive data (2M-1), and segmentation and mix one time, so node computing load CO_{smt} can be expressed as:

$$CO_{smt} = N\,(1 - Ps)\,((k_{enc} + 1)\;(2M + 1)\;. \tag{11}$$

In HMSS-SMART, each leaf-node is encrypted one time, send and receive data $(2m_i - 1)$, segmentation and mix one time, the operations of encryption and decryption of aggregation-node are less than SMART one time. So, computing load

CO_{hsmt} of the node can be expressed as:

$$CO_{hsmt} = \sum_{i=1}^{N(1-P_s)} (2m_i + 1) + N\left(2\left(1 - P_s\right) + \left(1 - 3P_s\right) k_{enc}\right) . \tag{12}$$

Here $m_i \in [2, M]$, and mi is obeyed by the function $F(M, M)$ of probability distribution described in the preceding text.

Assuming that $M = 4, P_s = 0.3$, the simulation results are shown in Figure 3. Of the three schemes, the SMART has the highest computing load. The main reason is a large amount of distribution for data-slice, and each slice is encrypted and decrypted. Because the TAG has not deployed the mechanism of the protection of privacy-preserving data, its network computing load is much lower than SMART. Although HDSS-SMART also requires data-slices, but because the number of slices is random in the [2, M] range, the total number of data-slices drops significantly. At the same time, due to the technique of encryption and authentication for Homomorphic MAC has reduced the times encrypted and decrypted of nodes, the decryption and integrity verification of the aggregation result is also performed at the base-station. It is effectively reducing the computing load of the fuse nodes. So, there is a significant improvement in computing load than SMART, even lower than that of the TAG.

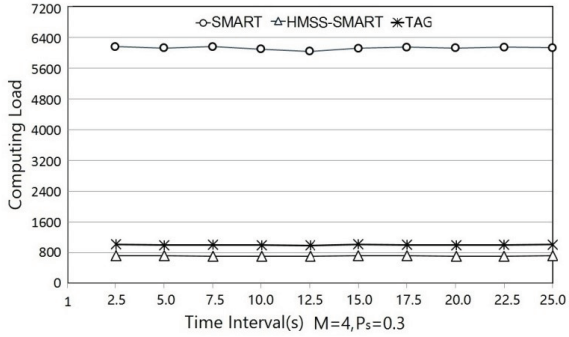

Fig. 3. Simulation Results for Computing Load of Nodes

3.3. Communication overhead

The technique of data segmentation and mixture will necessarily lead to the increase of the communication overhead of the network. Assuming N is the number of network nodes, each data transfer is one standard overhead unit of communication.

In the SMART, the nodes need to be split into M slices for the raw perceptual data, so the formula for communication overhead is:

$$O_{smt} = N.M . \tag{13}$$

In the HMSS-SMART, the aggregation-node does not split the data, so its data

transfer is $N.P_s$. Let the number of data-slices in the leaf-node I is m_i, and the number of outgoing links of the node is $m_i - 1$, the transmission quality is $\sum_{i=1}^{N(1-P_s)}(m_i - 1)$. Therefore, the formula of communication overhead under the HMSS-SNART is:

$$O_{hsmt} = N.P_s + \sum\nolimits_{i=1}^{N(1-P_s)} (m_i - 1)(m_i \in [2, M]) . \tag{14}$$

Because of the probability distribution of m is $F(M, m)$. So, we can get this:

$$O_{hsmt} = N \cdot P_s + \sum_{m=2}^{M} F\left(M, m\right) \cdot N\left(1 - P_s\right)\left(m - 1\right) . \tag{15}$$

Let the rate of communication overhead for SMART and the HMSS-SMART $R = O_{smt}/Ohsmt$, the formula (18) is available.

$$\begin{aligned} R &= \frac{N.M}{N \cdot P_s + \sum_{m=2}^{M} F\left(M, m\right) \cdot N\left(1 - P_s\right)\left(m - 1\right)} \\ &= \frac{M}{P_s + \left(1 - P_s\right) \sum_{m=2}^{M} F\left(M, n\right)\left(m - 1\right)} . \end{aligned} \tag{16}$$

From the definition of the function $F(M, m)$ for probability distribution, because $\sum_{m=2}^{M} F(M, m)(m - 1) < M$. So:

$$R > \frac{M}{P_s + \left(1 - P_s\right).M} = \frac{M}{M + P_s\left(1 - M\right)} . \tag{17}$$

Because $M >= 2$ and $P_s \in (0, 1)$, so there is $P_s(1 - M) < 0$. So, we know: $M + P_a(1 - M) < M$, and then we can derive that $R > 1$, that is: $O_{hmdsmt} <= O_{smt}$. It is to be proved that the network communication overhead of the HMSS-SMART is smaller than the SMART. The simulation results of communication overhead for $M = 4$ and $P_s = 0.3$ are shown in Figure 4. Because TAG does not use the segmentation of data, so it has lowest communication overhead. Conversely, SMART needs to distribute a lot of slices, so communication overhead is highest. Because the number of slices of HMSS-SMART is changed in [2, M] randomly, so the communication overhead of HMSS-SMART reduced significantly compared to SMART.

3.4. Aggregation accuracy of data

As data volumes increase, so does the probability of data collisions and the number of the BER, and then the aggregation accuracy will be reduced. As shown in Figure 5.

As data volumes increase, so does the probability of data collisions and the number of the BER, and then the aggregation accuracy will be reduced. When the BER of data transfer between nodes is 0.04, with time intervals of data aggregation;

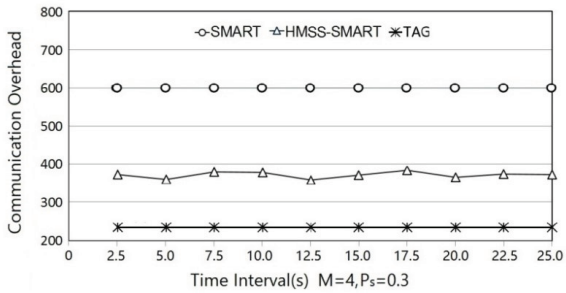

Fig. 4. Simulation Results for Communication Overhead of Nodes

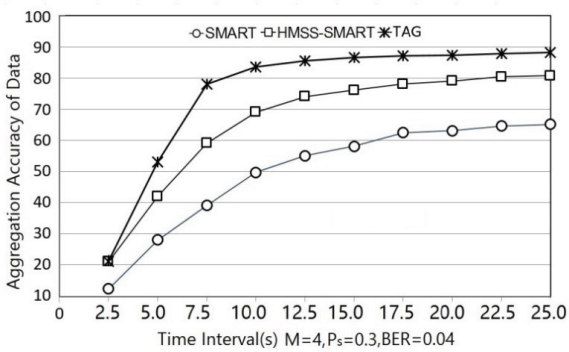

Fig. 5. Simulation Results for Aggregation Accuracy of Data

the data aggregation accuracy of the three is increasing rapidly. But TAG and HDSS-SMART are rising fast, and SMART is slower. After 10s, the HDSS-SMART aggregation precision is about 35%-40% higher than SMART. The accuracy of data aggregation of HDSS-SMAR's is significantly higher than SMART.

4. Conclusion

Compared to SMART, due to the end-to-end authentication encryption mechanism for homomorphic MAC can effectively reduced the times of encryption and decryption, and the integrity of the aggregation data is guaranteed; at the same time the technique of stochastic segmentation has reduced the number of data-slices, not only can reduce the communication overhead, but also can strengthen the protection for privacy- preserving data. Simulation also shows that: HMSS-SMART has reduced the exposed probability of privacy-preserving data, improved the accuracy of data aggregation, and reduced communication overhead and computational load significantly.

References

[1] B. Krishnamachari, D. Estrin and S. Wicker: *The Impact of Data Aggregation in Wireless Sensor Networks*. Distributed Computing Systems Workshops, Proceedings of 22nd IEEE International Conference. *22* (2002), 575–578.

[2] J. W. Bista chang: *Privacy-preserving data Aggregation Protocols for Wireless Sensor Network*. Sensors *5* (2010), 4577–4601.

[3] S. Madden, M. J. Franklin and J. M. Hellerstein: *TAG: A tiny aggregation service for ad-hoc Sensor networks*. Proceedings of the 5th Symposium on Operating Systems Design and Implementation *5* (2002), 131–146.

[4] M. Yoon, M. Jang and H. I. Kim: *A Signature-based data Security Technique for Energy-efficient Data Aggregation in Wireless Sensor Networks*. International Journal of Distributed Sensor Networks *7* (2014), 576–592.

[5] J. M. Bahi, C. Guyeux: *Efficient and Robust Secure Aggregation of Encrypted Data in Sensor networks*. Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on IEEE *4* (2010), 472–477.

[6] C. M. Chen, Y. H. Lin and Y. C. Lin: *RCDA: Recoverable Concealed Data Aggregation for Data Integrity in Wireless Sensor Networks*. IEEE Transactions on Parallel and Distributed Systems *4* (2012), 727–734.

[7] W. He, X. Liu and H. N. Guyen: *Pda: Privacy-preserving data aggregation in wireless sensor networks*. INFOCOM 2007, Proceedings of 26th IEEE International Conference on Computer Communications. IEEE Press *26* (2007), 2045–2053..

[8] S. Papadopoulos, A. Kiayias and D. Papadias: *Exact in-network Aggregation with Integrity and Confidentiality*. IEEE Transactions on Knowledge and Data Engineering *10* (2012), 1760–1773.

[9] S.Agrawal, D.Bench: *homomorphic MACs: MAC-based Integrity for Network Coding*. Proceedings of the 7th International Conference on Applied Cryptography and Network Security *7* (2009), No. 8, 292–305.

Received October 18, 2017