Acta Technica *62* (2017), No. 6A, 69–78

A zero-knowledge set based communication scheme for sharing economic

Dandan Li[1], Wanxin Xue[1,2], Yilei Pei[1], Jing Mu[1]

Abstract. Nowadays, sharing economy by using mobile-commerce (M-commerce) is very hot topic and phenomenon. People could share car, room, task, money, meal, clothes, knowledge or information, office, equipment and so on. Group communication of M-commerce especially in peer-to-peer model, is becoming increasingly popular in mobile applications. Thus, how to insure the security of group communication in sharing economy has become an important research issue. Most of the proposed methods directed towards wired networks. Although some approaches can be employed in the M-commerce environment of sharing economy, they cannot achieve the same efficiency as them in wired networks. Therefore, a distributed group key management approach is proposed in this paper, which integrates zero-knowledge set and hierarchy structure together. Re-key procedure is finished in sub-group so that it cannot affect the other members, and Pederson commitment is used to provide the anonymity of personal users. It is demonstrated that the new scheme can achieve better efficiency in terms of communication, computation, storage cost, anonymity and security when compared to the other schemes. It is more suitable for M-commerce in sharing economy.

Key words. Pedersen commitment, zero-knowledge set, group key management, sharing economy.

1. Introduction

Sharing economy is a class of economic arrangements in which asset owners and users mutualize access to the products or services associated with these assets [1, 2]. Today, the best-known sharing economy companies do business in a lot of fields, such as bicycle sharing including Mobike, ofo, Unibike; car sharing including Didi, Uber; clothes sharing including rent the runway, Poshmark; knowledge or information sharing including Khan Academy, Zhihu; room sharing including Onefinestay, Airbnb, Couchsurfing; task sharing including TaskRabbit; money sharing including

[1]Management School, Beijing Union University, Beijing, China
[2]Corresponding Author, e-mail: xuewanxin1@126.com

Tilt, Kickstarter; meal sharing including EatWith; office sharing, equipment sharing and so on. In China, the sharing economy market size has reached $570 billion in 2016.

What is sharing economy we using? It is a socio-economic ecosystem that commonly uses information technologies to connect different stakeholders in order to make value by sharing their excess capacities, products and services [3, 4]. So we define it from a broader view: any sales transactions that enhance the utilization of idle resources largely based on online market places. Most sharing economy transactions must depend on mobile E-commerce such us mobile equipment. Users can do transactions at any time and any place without any restriction because of the light-weight mobile devices. However, the core issue of sharing economy multicast communication is security, that is because it is vulnerable to be eavesdropped and access unauthorized in wireless communication. In order to ensure only authorized users can participate in the group, access control must be achieved by encrypting the communication data with a cryptographic key, which is known as group key. Therefore, a secure and efficient key management is a critical issue in group communication for sharing economy [5].

Researchers did a considerable amount of research work on group key management over past few decades. Wong C. K. [6] proposed KEY GRAPH which generates logic key tree and keys by server. The server manages all of the keys in group. In this approach, every member who belongs to different subsets can communicate with each other via different keys. Based on the Logical Key Hierarchy (LKH), a key predistribution scheme is proposed [7], which satisfies the stateless, correctness, group key secrecy, forward secrecy and backward secrecy. Ref. [8] proposes a lightweight key establishment protocol for wireless sensor networks. By optimizing information exchanges in the process of key establishment, this temporal initial key based protocol is able to achieve better extensibility and lower energy consumption.

For key agreement protocols, all communication entities are involved to determine session keys. The most commonly used key agreement protocol is Diffie-Hellman (DH) key agreement protocol [9]. But DH public key distribution algorithm can only provide session key for two entities; not for a group more than two members. So GDH (Group DH) was proposed in 1996 by Steiner M. et al. [10]. Ref. [11] distribute group key based on non-DH key agreement approach. Since all communication entities are involved to determine session keys, the time delay of setting up is too long, especially when there are a large number of group members. A new two-round authenticated contributory group key agreement was proposed by Vankamamidi S Naresh [12] in 2015. It is based on Elliptic Curve Diffie–Hellman protocol with Privacy Preserving Public Key Infrastructure (PP-PKI) which is extended to a dynamic authenticated contributory group key agreement with join and leave protocols for dynamic groups. Yan Lili proposed a user authentication and key agreement scheme based on heterogeneous wireless sensor networks [13]. Turkanović et al.'s proposed a highly efficient and novel user authentication and key agreement scheme (UAKAS) for heterogeneous WSN (HWSN) which was adapted to the IOT notion. Unfortunately Turkanović et al.'s scheme has some security shortcomings and is susceptible to some cryptographic attacks [14]. So Mohammad Sabzinejad Farash et

al. [15] proposed a new and improved UAKAS which enables the same functionality but improves the security level and enables the HWSN to dynamically grow without influencing any party involved in the UAKAS. But they are not suitable to be employed in Mobile E-commerce whose entities are large and change frequently, for example, sharing economy.

Zero-knowledge set was proposed by Silvio Micali et al. [16] in 2003. They show how a polynomial-time prover can commit to an arbitrary finite set S of strings so that, later on, for any string x, reveal with a proof whether x belongs to S or not, without revealing any knowledge beyond the verity of these membership assertions. In 2005, ref. [17] proposed a new group key distribution protocol based on zero-knowledge set and Pedersen commitment, so that the identities and number of the group members can be concealed and realize key distribution at the same time. In order to employ the advantage of zero-knowledge set, a group key management scheme is proposed in this paper, which is called ZKD (Zero-knowledge set based Distributed Group Key Management) for short. It integrates the framework of Iolus [18] and can be applied in Mobile E-commerce of sharing economy.

2. Proposed scheme

2.1. Basic idea

Group key exchange protocol based on zero-knowledge set (GKPZS) guarantees the confidentiality and security of every member who participates in group. It is suitable for some special network applications, such as secret network meeting. Pedersen commitment provides its safety and rationality. Iolus [19] is a high efficiency structure which is able to achieve better extensibility, because it completes key distribution and access control in sub-group. Based on the above analysis and the needs of M-commerce communication in sharing economy, a distributed group key management with zero-knowledge quality (ZKD) is proposed.

In ZKD scheme, every sub-group controller p_i generates key k_i as its sub-group session key itself. p_i and Group Secure Controller (GSC) share private keys k_{vi}. GSC also stores k_i of every sub-group. The key tree of sub-group p_i is Merkel tree. The example of framework tree for generating and managing keys is Fig. 1.

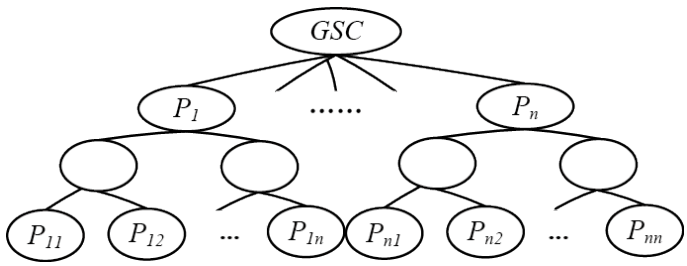

Fig. 1. A Framework Tree of ZKD

2.2. Sub-group key generation algorithm

A tree level size h is assumed in sub-group tree. An example of the sub-group key k_1 generation procedure as follows, and takes the idea of GKPZS for reference.

The leaf node is called real node only if it participates in communication. For intermediate node, if at least one of its child nodes is real node, the intermediate node is called real node. All of the nodes are called virtual node except real node. Each node stores four values which are (m_v, c_v, h_v, K) from the leaf nodes up to the second level nodes of sub-group p_1.

2.3. Commitment generation algorithm of leaf node

The value c_v of leaf node is $c_v = g^{m_v} h_v^{r_v} \bmod p$, where p is a prime.

If it is a virtual node, $m_v = 0$, and $h_v = g^{e_v}$, where e_v is a random value selected by p_1. Else, $m_v = A^{r_i}$, and $h_v = h^{e_v}$, where $r_i \in Z_q$ is a random value generated by p_1; A and h and g are generators by the cyclic subgroup of Z_p^* of order q; A, h and g are the exponential relationship; the exponential relationship of h and g is confidential.

2.4. Commitment generation algorithm of intermediate node

The calculation of intermediate node value c_v is the same as leaf node, The value m_v is $m_v = A^{(c_{v_0}, c_{v_1})}$. If it is a real node, $h_v = h^{e_v}$, else $h_v = g^{e_v}$, where c_{v_0} and c_{v_1} is commitment of its left and right children; e_v is a random value selected by p_1.

In order to not lose any generality, we consider the virtual node of $(h-1)$-th which has two virtual leaf children nodes. m_v is calculated by using (2), that is $m_v = A^{g^{e_{v_0}(r_{v_0}+1)+e_{v_1}(r_{v_1}+1)}}$, where $e_{v_0}, r_{v_0}, e_{v_1}, r_{v_1}$ are parameters selected by server p_1 when calculates c_{v_0} and c_{v_1}. $e_{v_0}, r_{v_0}, e_{v_1}, r_{v_1}$ can be selected appropriately to satisfy $m_v = A$. The selection of e_v and r_v still satisfies $m_{parnet}(v) = A$, if its parent node is virtual node. Else e_v and r_v are selected randomly. c_v can be computed by e_v and r_v. Following the steps above, all commitment values can be calculated along the path from leaf nodes up to the root e, commitment value c_1 is made public at last.

2.5. Sub-group key generation algorithm

k_1 can be got through verification process. $P(v)$ is a path from v up to the root, u is intermediate node on $P(v)$ except leaf nodes and root. As $m_v = A^{(c_{u_0}, c_{u_1})}$, $h_u = h^{e_u}$, $c_u = g^{m_u} h_u^{r_u} \bmod p$, which are used to recursively calculate commitment value of every node on $P(v)$ until the root e, that is c_1. c_1 is used to verified whether public value is equal to intermediate node of $P(v)$. Verification process is finished if the commitment value is equal, and the user can ensure that he/she is selected to be a member of group him/herself. He/She can compute A_1^R using r_i and $A_1^{r_i^R}$ as k_1.

Exponentiation is the most expensive operation in group key generation algo-

rithm. From the above calculation of commitment value in ZKD, every leaf node does exponentiation 3 times, while intermediate node does it 4 times. Obviously, the commitment value can be computed on polynomial time for a group with n members.

2.6. Group key generation algorithm

GSC generates group key K and sends it to all sub-group controllers p_i. K can be got through decrypting private key k_{0i} shared with p_i and GSC. Then p_i sends K to its all sub-group members. The members gain K through decrypting messages by k_i. Here is an example of p_1.

GSC $\Rightarrow \{p_1, p_2, \ldots, p_m\}$: $\{K\}k_{0i}; p_i \Rightarrow \{p_{i1}, p_{i2}, \ldots, p_{in}\}$: $\{K\}k_i$; $(i = 1, 2, \ldots, m)$, where $\{K\}k_{0i}$ means message K is encrypted by the key k_{0i}; $A \Rightarrow \{B\}$: *message* means A sends message to B via broadcast or multicast.

This procedure needs multicast twice, and encryption $(m + 1)$ times.

2.7. Group key updating algorithm

In a secure group, whenever there is a change in access structure, the corresponding group secret key value has to be changed to a new value. Otherwise, the new joining users would be able to understand previous communications whereas a leaving user would continue to read the ongoing communication after it left. These situations are undesirable. The following protocols take care of such changes.

2.8. User join

When a user wants to join a sub-group (take p_1 for example), it sends a JOIN request to p_1. p_1 dose 3.1 again to compute new commitment value and generates new sub-group key k_1' which is sent to GSC after encrypting. GSC gains k_1' through decrypting message by k_{01}. Then, GSC generates new group key K' and sends it to p_i after encrypting K' by k_{0i}. p_i sends K' to its members.

$p_1 \rightarrow \{\text{GSC}\}$: $\{k_1'\}k_{01}$; GSC $\Rightarrow \{p_1, p_2, \ldots, p_m\}$: $\{K'\}k_1'$, $\{K'\}k_i$; $p_i \Rightarrow \{p_{i1}, p_{i2}, \ldots, p_{in}\}$: $\{K'\}k_1'$, $\{K'\}k_i$, where $i = 1, 2, \ldots, m$; members in sub-group p_1 gain K' through decrypting by k_1'; members in other sub-group gain K' through decrypting by k_i, because joining user does not have k_i; $A \rightarrow \{B\}$: *message* means A sends message to B via unicast.

This protocol needs unicast once and multicast m times. That is, GSC does encryption and communication twice when one user joins group.

2.9. User leave

When a user wants to leave a sub-group (take p_1 for example), it sends a LEAVE request to p_1 first. p_1 changes this node to be a virtual node and dose 3.1 again to compute new commitment value. Hence, members in p_1 can gain new sub-group key through computation themselves. Leaving affects nodes' value whose position above this node in key tree because of deleting real node. p_1 sends new sub-group key k_1'

to GSC. GSC gains k_1' through decrypting message by k_{01}. Then GSC generates new group key K' and sends it to p_i. At last p_i sends K' to its members.

$p_1 \to \{\text{GSC}\}$: $\{k_1'\}k_{01}$; GSC $\Rightarrow \{p_2, \ldots, p_m\}$: $\{K'\}k_1'$, $\{K'\}k_i$; $p_i \Rightarrow \{p_{i1}, p_{i2}, \ldots, p_{in}\}$: $\{K'\}k_1'$, $\{K'\}k_i$, where $i = 2, \ldots, m$. This protocol needs unicast once and muliticast m times. Encryption and communication GSC does, is equal to the number of sub-groups when a user leaves group.

3. Performance discussion

In this section, ZKD is compared and analyzed with GDH and KEY GRAPH using MATLAB software. Members who leave and join the group are called changing member, the others are non-changing member. In order to analyze simply, we assume the degree of key tree is d, level is h, so there are $n = d^h$ members in every sub-group.

3.1. Anonymity

ZKD is based on zero-knowledge set which can ensure the anonymity of members in group. Members compute commitment value to get group key themselves and cannot get any other auxiliary information about other members. It is very favorable to participants of M-commerce.

3.2. Computation efficiency

We take one changing user for example. Sub-group controller does sub-group key generation algorithm when member joins and leaves sub-group, to compute new commitment value. It does $h - 1 = \log_d n - 1$ times of exponentiations. Changing member does $h+1 = \log_d n+1$ times of exponentiations. For non-changing members, (1) is the average computation times. Equ. (2) is computation cost of the whole sub-group itself. Additionally, there are $(d-1)d^{h-i}$ members who compute key i times.

$$\frac{1}{n-1}\sum_{i=1}^{k} i(d-1)d^{h-i} = \frac{d}{d-1} - \frac{\log_d n}{n-1}\,, \tag{1}$$

$$(\frac{d}{d-1} - \frac{\log_d n}{n-1})(n-1) = \frac{d(n-1)}{d-1} + \log_d n\,. \tag{2}$$

Figure 2 depicts the computation cost of ZKD, which abscissa is degree of key tree, and ordinate is computation cost whose unit is calculation number.

Figure 3 depicts the computation cost of ZKD, KEY GRAPH and GDH, which abscissa is group size, and ordinate is computation cost whose unit is calculation times. The three curves are under different degree of key tree d, because it has proved that they are the best performance in their different degrees respectively. From Fig. 3 we can see that, ZKD is better than GDH and close to KEY GRAPH at computation efficiency. However, most computation of ZKD is exponentiation whose speed is much faster than logarithm operation.

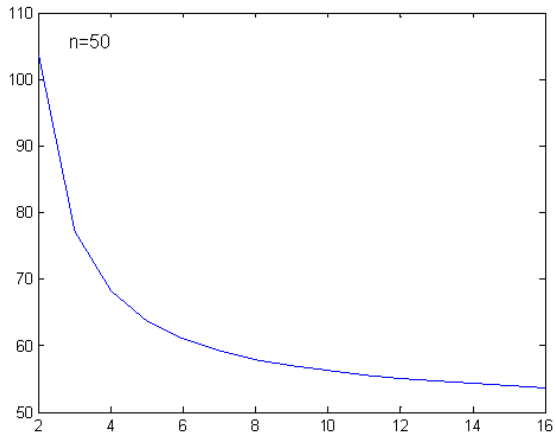

Fig. 2. Computation Cost of ZKD

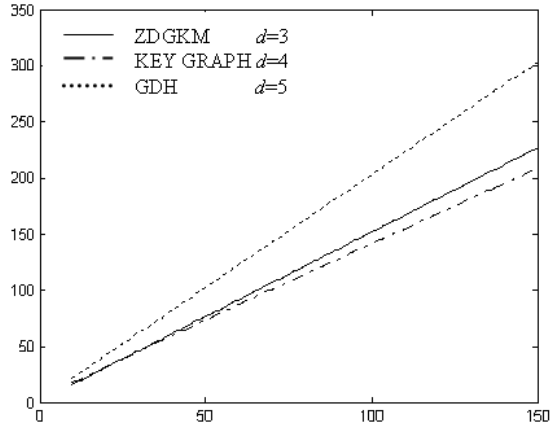

Fig. 3. Computation Cost of three schemes

3.3. Storage efficiency

The storage efficiency is the number of keys stored in the group controller and user side. In ZKD, every user stores three values which are used to compute commitment value. Furthermore, users also store sub-group communication key and the commitment value on their authentication path nodes. So, for the users in the group, the number of stored values is $dh + 4 = d \log_d n + 4$ whereas in the whole sub-group, the number of stores values is $dn \log_d n + 4n$. We tabulate the storage and computation cost in the Table 1. Figure 4 depicts the storage cost of ZKD. Its abscissa is degree of key tree, and ordinate is storage cost whose unit is number of keys. From Fig. 4 and Fig. 3 we can see that, the bigger degree is, the smaller computation and storage cost of ZKD is. Furthermore, in Fig. 4, the storage cost is the smallest when $d \approx 3$, that means, ternary-tree is the most appropriate structure of group key management.

Table 1. Storage and computation cost of ZKD

		Changing member	Non-changing member	Sub-group controller
Computation	join	$O(\log_d n)$	$d/(d-1) - \log_d n/(n-1)$	$O(\log_d n)$
(calculation times)	leave	0	$d/(d-1) - \log_d n/(n-1)$	$O(\log_d n)$
Storage (keys number)		$O(d \log_d n)$		

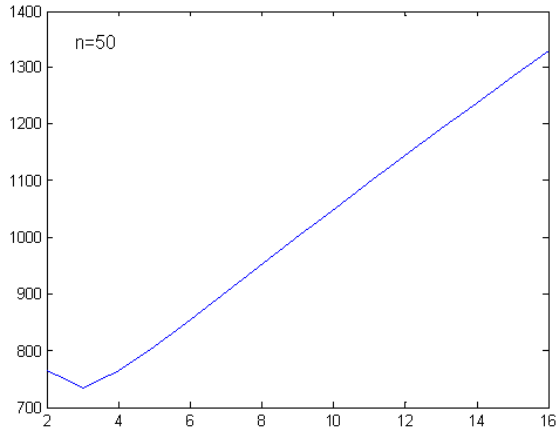

Fig. 4. Storage Cost of ZKD

Figure 5 depicts the storage cost of ZKD, KEY GRAPH and GDH, which abscissa is group size and ordinate is storage cost (number of keys). The three curves are under different degree of key tree d, because it has proved that they are the best performance in their different degrees respectively. Although the storage cost of ZKD is larger than KEY GRAPH, it is smaller than GDH.

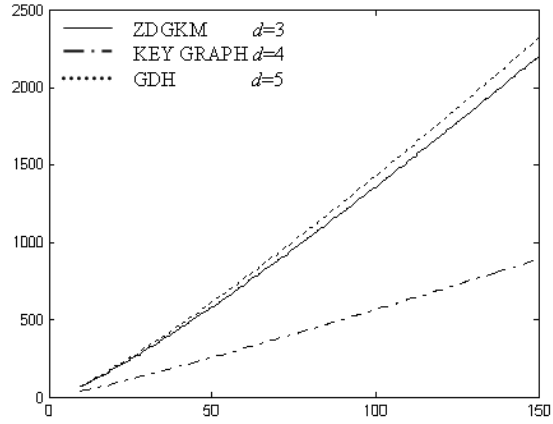

Fig. 5. Storage cost of three schemes

3.4. Communication efficiency

Communication overhead is recorded as a measure of rekeying messages transmitted per operation by the group controller. We evaluate the communication cost of the joining and leaving operations and tabulate our approach in the following Table 2.

Table 2. Communication cost

Operation		GDH	KEYGRAPH	ZKD
join	Unicast times	$n-1$	2	1
	Multicast times	n	1	n
	Exchange times	n	3	1
	number of message	3	3	3
leave	Unicast times	$n-1$	0	1
	Multicast times	n	1	n
	Exchange times	n	1	1
	number of message	3	1	3

4. Conclusions

Based on the analysis of current group key management schemes and the feature of M-commerce, ZKD protocol which based on zero-knowledge set is proposed. It is security against passive attackers, also could guarantee forward security and backward security. Moreover, the proposed protocol employs one group key in all of its sub-group that makes sub-group does not interact with each other, and it avoids the decryption and encryption repeating; hence, it is efficient at communication cost. Furthermore, the hierarchical structure of Iolus ZKD employs to manage the keys of group, has good scalability. It's concluded that the number of encryptions and re-key messages are less than the other protocols. So, ZKD is good for M-commerce and other large-scale multicast environments of sharing economy.

Acknowledgement

This work is supported by the Premium Funding Project for Academic Human Resources Development in Beijing Union University (BPHR2017DS14, BPHR2017 DS10, BPHR2017ES03); social science Foundation of Beijing (grant no. 17GLC050); Beijing Municipal Education Commission project of "Evolutionary Game and Supervision Mechanism of Beijing Municipal Government and Sharing Economic Enterprises"; National Social Science Foundation of China (17BGL265); Beijing Municipal Education Commission Social Science Program General Project of China (sm201711417002).

References

[1] R. Belk: *You are what you can access: sharing and collaborative consumption online.* Journal of Business Research *67* (2014), No. 8, 1595–1600.

[2] S. P. Fraiberger, A. Sundararajan: *Peer-to-peer rental markets in the sharing economy.* Social Science Electronic Publishing *68* (2015), No. 9, 1510–1555.

[3] A. Sundararajan: *The sharing economy: the end of employment and the rise of crowd-based capitalism.* Cambridge, The MIT Press (2016).

[4] J. Hamari, M. Sjoklint, A. Ukkonen: *The sharing economy: why people participate in collaborative consumption.* Journal of the Association for Information Science and Technology *67* (2016), No. 9, 2047–2059.

[5] Y. Amir, Y. Kim, C. Nita-Rotaru, J. Schultz, J. Stanton: *Secure group communication using robust contributory key agreement.* IEEE Transaction on parallel and distributed systems *15* (2004), No. 4, 468–480.

[6] C. K. Wong, M. Gouda, S. S. Lam: *Secure group communications using key graphs.* IEEE Transactions on Networking *8* (2000), No. 1, 16–30.

[7] J. Yan, F. Li, J. Ma: *A stateless key pre-distribution scheme for wireless sensor networks.* ACTA Electronica Sinica *37* (2009), No. 1, 2199–2204, 2210.

[8] W. Liu, R. Luo, H. Yang: *A lightweight key establishment protocol for wireless sensor networks.* Journal of Electronics & Information Technology *32* (2010), No. 4, 869–874.

[9] W. Diffie, M. E. Hellman: *New directions in cryptography.* IEEE Transactions on Information Theory *22* (1976), No. 6, 644–654.

[10] M. Steiner, G. Tsudik, M. Waidner: *Diffie-Hellman key distribution extended to group communication.* In Proceedings of the 3rd ACM Conference on Computer and Communications Security, ACM Press (1996), 31–37.

[11] W. G. Tzeng: *A Secure Fault-Tolerant Conference Key Agreement Protocol.* IEEE Trans. Computers *51* (2002), No. 4, 373–379.

[12] S. N. Vankamamidi, V. E. S. M. Nistala: *A new two-round dynamic authenticated contributory group key agreement protocol using elliptic curve Diffie–Hellman with privacy preserving public key infrastructure.* Sadhana *40* (October 2015), No. 7, 2143–2161.

[13] L. L. Yan, Y. Chang, S. B. Zhang: *A user authentication and key agreement scheme for heterogeneous wireless sensor networks.* Journal of University of Electronic Science and Technology of China (2017), No. 01, 55–60.

[14] T. Muhamed, B. Bostjan, H. Marko: *A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion.* Ad Hoc Networks *20* (2014), No. 2, 96–112.

[15] S. F. Mohammad, T. Muhamed, K. Saru, H. Marko: *An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment.* Ad Hoc Networks 36 Part 1 (January 2016), 152–176.

[16] S. Micali, M. Rabin, J. Kilian: *Zero-Knowledge sets.* 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS'03), Cambridge *11* (Oct. 2003), 80–91.

[17] H. Sun, D. Lin: *A new group key exchange protocol based on zero-knowledge set.* ACTA Electronica Sinica *33* (2005), No. 2, 345–348.

[18] M. Suvo: *Iolus: a framework for sealable secure multicasting.* Proceedings of the ACM SIGCOMM '97 conference Applications, technologies, architectures, and protocols for computer communication *27* (Sep. 1997), No. 3, 277–288.

[19] P. Torben: *Pedersen. Non-Interactive and information–theoretic secure verifiable secret sharing.* Proceedings of the 11th Annual International Cryptology Conference Advances in Cryptology, London UK (Aug. 1991), 129–140.

Received September 6, 2017