Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 5 (1): 12-23, 2013
ISSN 1815-7432 / DOI: 10.3923/rjit.2013.12.23

An Optimal Key Management Scheme for Group Key Sharing with Polynomial Expression

S. Krishnakumar and R. Srinivasan
SRM University, Chennai, Tamilnadu, India

Corresponding Author: S. Krishnakumar, SRM University, Chennai, Tamilnadu, India

ABSTRACT

In today's arena, multicasting technique plays a vital role in digital communication which has to be made secure with high resistance from middleman attacks. The proposed one is an economic way of generating an optimal number of keys with maximum utility in a sensor network rather than using a single key for each member. However the problem lies in generating the key management system which has a single common key for all the members of a virtual group that are essential for secure communication and should possess availability and least redundancy. This common key which is proposed as ViPKey can be shared among the members of a virtual group with polynomial clustering expression, thereby making it simpler to achieve efficient data transmission instead of using a separate key for each individual member of that group. Thus, proposed ViPKey management system enables least amount of keys involving in a single group and has reduced rekeying and updating process during inclusion or exclusion of members inside a virtual group among the neighbors of sensor network. The accuracy in terms of competence statistically with network simulator NS-2 version 2.26 that offers support for simulating a variety of protocol suites.

Key words: Virtual group, ViPkey management, rekeying, key servers, internet key exchange, user datagram protocol, internet protocol security

INTRODUCTION

In a digital sensor communication network, the important process of securing the transmission is done using the key management scheme, which is the organization of cryptographic keys in a system or in the data that are to be transmitted. The process includes dealing with the production, exchange, storage, utilization and substitution of keys among the members or users involved in the transaction using some clustering polynomial expressions and derivations that are used by the members during intergroup and intra group transmission (Piao *et al*., 2012). The method deals with the cryptographic protocol design, key servers, user events and other associated protocols in this network. Key management (Mukherjee *et al*., 2005) handles keys at the user level, either between users or systems. This is in contrast to key scheduling, which typically refers to the internal handling of key material within the operation of a cipher.

Successful key management is vital to the security of a cryptosystem. In practice, it is questionably the most difficult aspect of cryptography because it involves system policy, user training, organizational and departmental interactions and coordination between all of these

elements. Most IPSec implementations whether consist of an Internet Key Exchange (IKE) daemon that runs in user space and an IPSec stack in the kernel that processes the actual IP packets (Tshering and Sardana, 2011).

User-space daemons have easy access to mass storage containing configuration information, such as the IPSec endpoint addresses, keys and certificates, as required (Harkins and Carrel, 1998). Kernel modules, on the other hand, can process packets efficiently and with minimum overhead, which is important for performance reasons. The IKE protocol uses UDP packets, usually on port 500 and generally requires 4-6 packets with 2-3 turn-around times to create a Security Association (SA) on both sides. The negotiated key material is then given to the IPSec stack. For instance, this could be AES key, information identifying the IP endpoints and ports that are to be protected, as well as what type of IPSec tunnel has been created. The IPSec stack, in turn, intercepts the relevant IP packets if and where appropriate and performs encryption/decryption as required. Implementations vary on how the interception of the packets is done, for example, some use virtual devices and others take a slice out of the firewall, etc.

These policies of encryption are not meant for key sharing among virtual group of a sensor network. In many network applications, including distant learning, audio webcasting, video streaming and online gaming, often a source has to send data to many receivers. IP multicasts and application-layer multicasts provide efficient and scalable one-to-many or many-to-many communications. From Lin *et al.* (2009) study, a common secret key has been derived which is the group key, shared by multiple users can be used to secure the information transmitted in the multicast communication channel. In this study, a new group key management protocol is proposed to reduce the communication and computation overhead of group key rekeying caused by membership changes. With shared key derivation, new keys derivable by members themselves do not have to be encrypted or delivered by the server and the performance of synchronous and asynchronous rekeying operations, including single join, single leave and batch update, is thus improved. He *et al.* (2009) found that the cryptographic key management is challenging due to the following characteristics of wireless *ad hoc* communications.

Unreliable communications and limited bandwidth: Due to shared-medium nature of wireless links, flows may frequently interfere with each other. Moreover, a network may be partitioned frequently due to node mobility and poor channel condition. Therefore, the communication overhead for certificate exchange cannot be ignored.

Network dynamics: Mobile nodes may leave and join the *ad hoc* network frequently and new legitimated nodes may join the network later after some nodes have been deployed in the field. Mobility increases the complexity for trust management.

Large scale: The number of *ad hoc* wireless devices deployed at an incident scene depends on specific nature of the incident. In general, the network size can be very large. In addition, an *ad hoc* network should be able to accommodate more mobile devices if necessary, therefore it is necessary to have newly deployed devices and previously deployed devices trust each other without introducing too much overhead.

Resource constraints: The wireless devices usually have limited bandwidth, memory and processing power. Among these constrains, communication bandwidth consumption and memory are two big concerns for key management schemes.

Wireless bandwidth is the scarcest resources in wireless networks. On the other hand, memory concern for key storage is more and more evident, since the requirement on network scalability (or network size) is increasing.

The proposed protocol is shown to be secure and immune to collusion attacks and it outperforms the other comparable protocols from our analysis and simulation. The protocol is particularly efficient with binary key trees and asynchronous rekeying and it can be tuned to meet different rekeying delay or key size requirements. Thus with all the terms, arrived at a performance evaluation of system that gives virtual group node generation in sensor network.

KEY MANAGEMENT SYSTEM

As in our proposed system some of the group key management systems have been derived from many multicasting applications for both intragroup and intergroup digital sensor communication. In all this traffic management should be maintained by distributed and shared secret keys for secure communication within a single group and members of another group (Tshering and Sardana, 2011). A polynomial P to achieve efficient intragroup key refreshment and a polynomial H(x) to create an intergroup key are generated.

The proposed mechanism can reduce the number of rekeying messages. The mechanism reduces the storage overhead of sensor group members and the group controller by adopting a polynomial-based key management scheme, the group controller does not need to broadcast heavy messages which are necessary for creating an intergroup key among the sensor group. Thereby, it introduces only a small amount of broadcast traffic to the group members. The analysis of the proposed mechanism is conducted to demonstrate the improvements.

The system is evident for relative methodologies that can adopt intragroup and intergroup key management scheme in previous researches. One example comes from Yi *et al.* (2001), where the nodes are divided into soldiers, officers and generals.

Between the nodes in different groups routing requests and multicast traffic are needed. For instance, a soldier may send a message that can be read only by all of the generals. Another example comes from Wang and Stransky (2007), three groups of soldiers coming from countries A, B and C. When an event is observed by a soldier of country A, a description with different contents will be provided to different soldier groups. To support such requirements, secret keys must be deployed.

The straightforward solution to this problem is to encrypt messages with a shared secret key, so that entities who do not have the shared secret key cannot decode them. For the above application, when a member A in the research team wants to send an inspection report that should be read only by audit team members, it would be like a sender joining the operation. The audit team leader drops the previous group key which ciphered the past communication and generates a new shared secret key and broadcasts it to the sender and all of the audit team members. When another member B also wants to send a message to the audit team, the audit team leader should regenerate a secret key which is shared between the member B and audit team members. This solution is simple, yet with a disadvantage: the leader will be overwhelmed by the communication overhead for distributing the secret keys to each of senders.

When more senders want to send messages, more will be the communication overhead. They developed a key generation and update method for secure information sharing in the same group and among different groups. Researchers employ a polynomial P to achieve efficient intragroup key refreshment and generate polynomials H(x) from the intragroup key in order to communicate among different group notations P and H(x).

Our proposed approach drastically reduces the amount of broadcast traffic in the intergroup communication. The additional number of rekeying messages and communication overhead caused by the proposed scheme has been properly justified. Through using the proposed mechanism, the system can improve system efficiency and increase the network lifetime under the same traffic scenarios. The contributions of the proposed schemes are: Sharing the intragroup key between the group controller and group members do not need to adopt any encryption/decryption mechanisms. When membership changes happen, the keys are renewed immediately. The designed mechanism reduces the number of rekeying messages during group membership changes.

The adoption of the polynomial which is used for deriving an intragroup key can reduce the key storage overhead at the group members and the group controller. After the intragroup key is derived, the members self generate the polynomial functions which are necessary for creating an intergroup key. It helps to reduce the communication overhead at the group controller.

In Fig. 1 to ensure secure intergroup digital sensor communication researchers (Wang and Stransky, 2007) have proposed a polynomial-based scheme. A polynomial based scheme was first used to implement threshold secret sharing (Shamir, 1979; Staddon *et al*., 2002) propose a self-healing group key distribution mechanism that adopt polynomials to support the distribution of personal key shares. They use t-degree polynomial H(x) to determine the personal key shares and protect intergroup multicast traffic. In Fig. 1, H2,1(v) means personal key share to encrypt multicast traffic from v in the group one (G1) to the members of group two (G2). H2, 1(x) means a polynomial to determine the keys for decrypting the multicast traffic from a node in G1 to the members of G2. A node v in G1 will get its personal key shares H2,1(v) from the group controller. The group controller of G1 will request H2,1(v) from the group controller of G2 and node v encrypts the message using H2,1(v) and sends it to the members in G2. In this time, the nodes in G2 already get H2,1(x) from the group controller and they know that the message comes from node v, so, the nodes in G2 can calculate H2,1(v) and decrypt the message from node v. For example, H2, 1(x) = 5x+8, all the members in G2 have H2,1(x), if the value of node v is 5, the intergroup key will be calculated as H2,1(v) = 5×5+8 = 33. So, in this case only the sender v and members in G2 will be able to read the information which is encrypted with 33, because other nodes cannot calculate the polynomial. Here, the polynomials H(x) is generated by the group controller, the group controllers consume energy not only when they generate the polynomials but also when they send it to the group members.

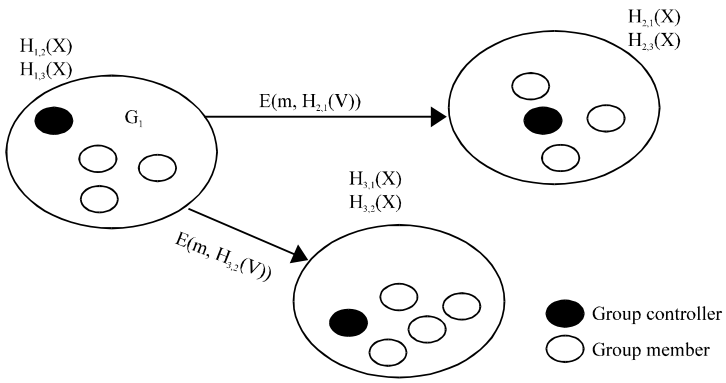

Fig. 1: Member v in G1 sends message to all members in G2 and G3, H(x): To create an intergroup key, V-member

METHODOLOGY

In proposed system, contribution on the resource management and generate polynomial time for key sharing model where the process of resource allocation among the sensors is made dynamically with the available number of members in virtual group as that of Virtual group key management framework (Fig. 2) and the performance efficiency. The process starts with the generation of virtual group that consists of minimum three member nodes and then generating a polynomial time which is based on a key management scheme that protect both intra sensor group and inter sensor group multicast traffic. This time is undertaken by the group controller and he generates the common key for the entire group and thus the controller becomes the cluster head and updating process of member modification is eliminated because of having no individual key for each member of a cluster. Then the response sent to pair key generator which produces combination of keys that are going to be processed during data transmission.

This generated pair of keys is sent to the pool which has the shared key pair. Then the transmission gets initiated between one virtual group members to the other. The cluster head collects the appropriate key from the pool and encrypts the information with the available key and transmission is made through data masking within data stores. It ensures that sensitive data is replaced with realistic but not real data. The goal is that sensitive customer information is not available outside of the authorized environment. The shuffling technique uses the existing data as its own substitution dataset and moves the values between rows in such a way that the numbers of values are present in their original rows. Data masking is typically done while provisioning non-production sensor environments so that copies created to support test and development processes are not exposing sensitive information and thus avoiding risks of leaking.

Masking algorithms are designed to be repeatable so that referential integrity is maintained. Once the information reaches the target virtual group as shown in Fig. 3 encryption is done and decryption is made at the transmission accomplishment stage thus achieving secure transmission with minimum utilization of resources. As the sensor cluster VGx1 generating the keys H1 with the combination H2,3 it makes an encryption of E(M,H2,1) to the virtual group x2 where the combination lies as H2 with H1,3. This formation of new key will be extracted by the cluster head and shares among the members of that cluster.

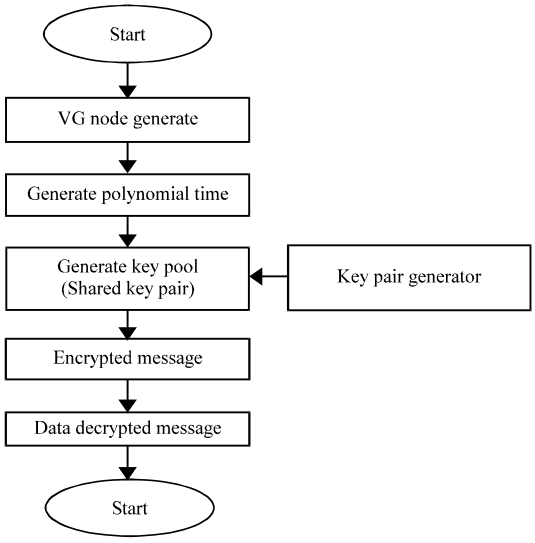

Fig. 2: ViPKey: Virtual group key management framework

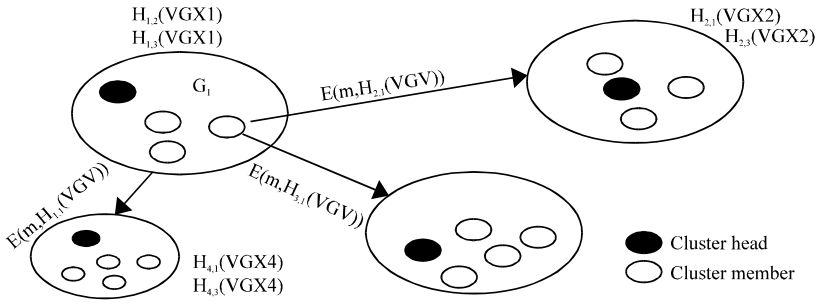

Fig. 3: Virtual group key generating outlay

As the sensor cluster VGx1 generating the keys H1 with the combination H2,3 it makes an encryption of E(M,H2,1) to the virtual group x2 where the combination lies as H2 with H1,3. This formation of new key will be extracted by the cluster head and shares among the members of that cluster.

Likewise as the encryption E(M,H3,1) combination moves over virtual group x3 forms H3 with effort from H1,2. And finally E(M,H4,1) form key H4 with combination of H1,3 at virtual group x4 and overall process is shown in Fig. 3 which lead to optimized key pair generation in terms of number and functionality.

SYSTEM IMPLEMENTATION

The proposed system ViPKey is having optimization resource utilization to virtual group which is implemented in terms of above mentioned model with polynomial inference. The following flow will show the entire process in terms of codes with vital load balance mechanism, mapping and threshold management.

ViPKey: Virtual group key management framework:

Claim: Resource management and virtual group key management
Step1: Admission access (no generation = ack)
if (ack == success)
do(VG generation proceed)
for (index packet = 0 to max no of members n)
do (maintain record of existing members)
if (member failure == 1)
do (update record)
else if (failure == 0)
do (maintain existing record shared key)
index number = already generated key number
else (maintain record of updated members)
else (ack == fail)
do (current virtual group = enable)
Step 2:
if (ack == +ve)
do (generate ViPKey)
else if (ack == -ve)
do (group update)
go to Step 2.

The whole system is constructed on the basis of above flow where essentiality is maintained at the resource manager to optimally generate the number of keys and routing path to search of virtual group available with handful resource to fit the generated keys.

DISCUSSION

The intangible influence of ViPKey model is executed to prove the efficiency standards of optimal resource manager in the sensor network and exponent when compared to the predecessors. The obtained is, more accuracy in terms of competence statistically with network simulator NS-2 version 2.26 that offers support for simulating a variety of protocol suites. The consideration is made in order to show the advancement of every parameter like time, resource allocation and available virtual group in terms of key generation time, cost of memory, power consumption and communication overheads, end to end delay and importantly number of keys generated.

In our proposed system (ViPKey), during resource allocation, the accuracy is calculated with the number of available virtual system within the sensor cluster in the mean interval of time. Figure 4 indicates the analysis result of the power consumption with respect to simulation time which increases exponentially. In some instances this may require exchanging identical keys.When the simulation time is 70, the power consumption is 8.1 in ViPkey but increased to 8.8 for Distributed key. Figure 5 represents the analysis results of key management time with respect to simulation time. When the simulation time is 2 msec, the key management time is 24 in ViPkey but increased to 25 for Distributed key. In other way, it may require possessing the other end public key. While public keys can be openly exchanged, symmetric keys must be exchanged over a secure communication channel.

Formerly, exchange of such a key between the sensor nodes was extremely troublesome and was greatly eased by access to secure channels such as a diplomatic bag. Clear text exchange of

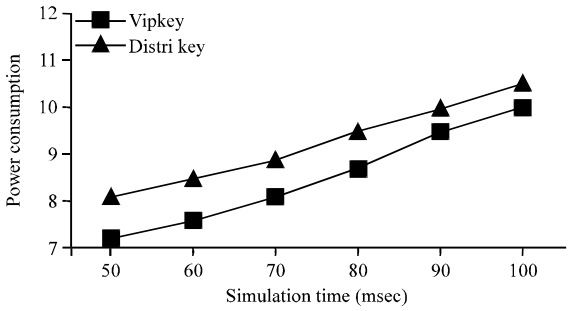

Fig. 4: Power consumption vs. simulation time

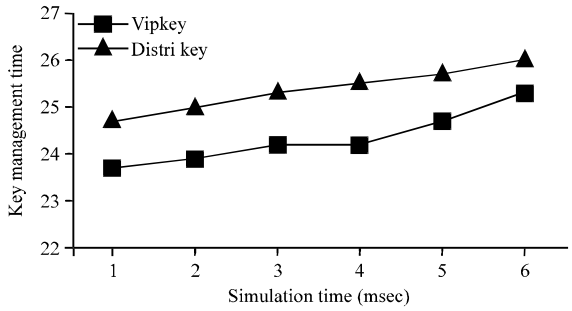

Fig. 5: Key management time (s) vs. simulation time

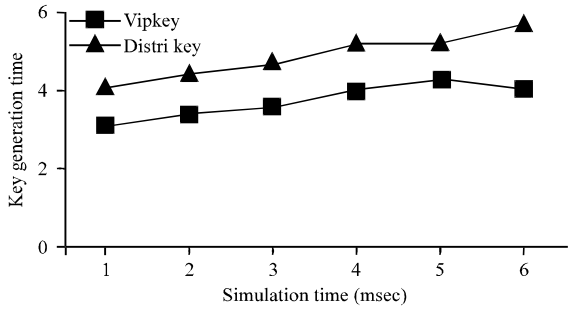

Fig. 6: Key generation time (s) vs. simulation time

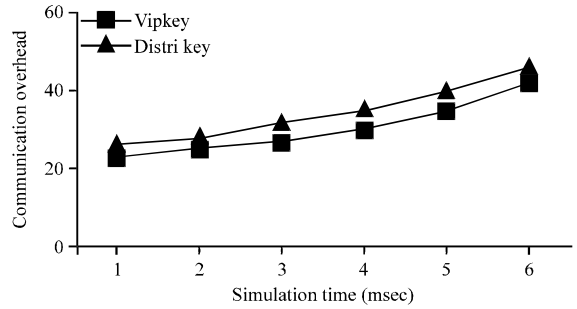

Fig. 7: Communication overhead vs. simulation time

symmetric keys would enable any interceptor to immediately learn the key and any encrypted data and the time to manage the key is reduced in relation with simulation time. Figure 6 show an important enhancement due to usage of common key of a single virtual group and the time to generate key is relevantly diminished compared to the prior work. When the simulation is 3 msec, the key generation time is 3.5 msec in ViPkey but increased to 4.5 msec for distributed key. Figure 7 explains the performance of communication overhead with respect to simulation time. When the simulation time is 4 msec, communication overhead is 30 for ViPkey but increased to 35 for distributed key. Communication overheads increase as the number of people increases. The number of different communication channels increases along with the square of the number of people; doubling the number of people results in four times as many different conversations. Everyone working on the same task needs to keep in sync, so as when more people are added they spend more time trying to find out what everyone else is doing and the communication overhead is reduced nearly twice the previous key generation models.

Figure 8 show the average communication time with respect to simulation time, the reduction in average communication overhead. With pre conditions, if simulation time is 4. Then the average communication overhead will be 30 for distributed key but 25 for ViP key. Figure 9 show the PDR% with respect to simulation time. All these works of experimental studies were carried out with ViPKey where the Virtual group environment is formulated with standards that was verified through code and the parameters of the resultant graph is extracted. Figure 10 shows the memory cost with respect to simulation time. Figure 11 shows the compromised node with respect to time. Figure 12 shows the end to end delay with respect to time. Figure 13 show the memory utilized with respect to process time. Figure 14 shows the average remaining energy with respect to simulation time. Finally the important role of key reduction is given in Fig. 15 which directly proportional to the network size.

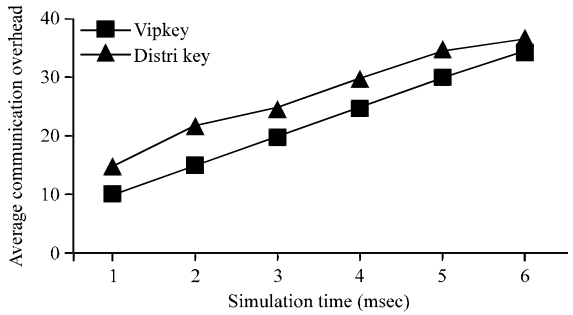

Fig. 8: Average communication overhead and simulation time

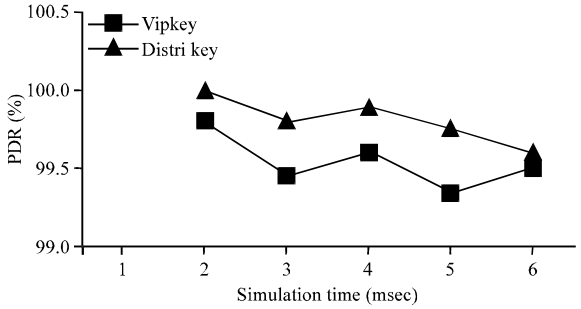

Fig. 9: PDR% vs. simulation time

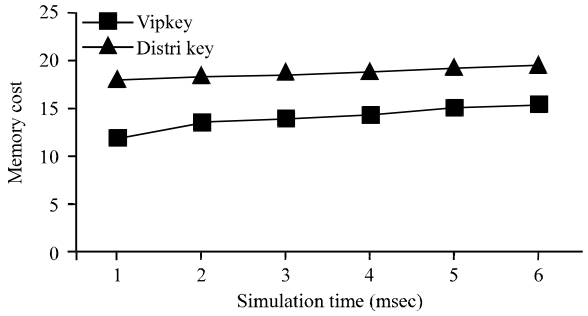

Fig. 10: Memory cost vs. simulation time

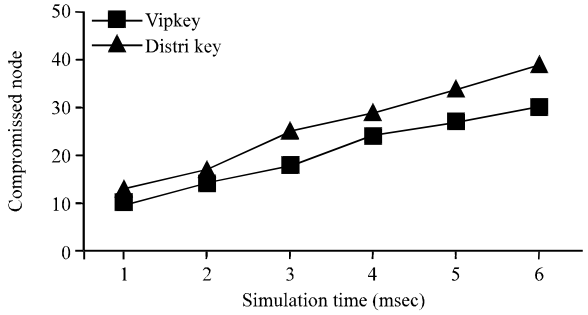

Fig. 11: Compromised node vs. simulation time

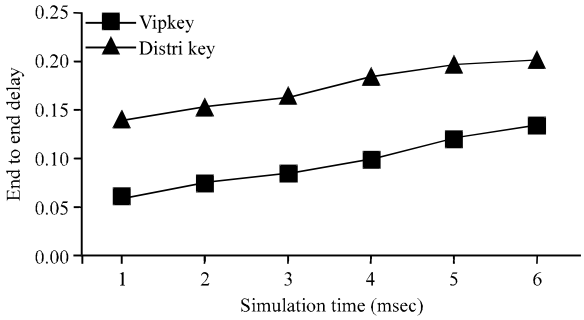

Fig. 12: End to end delay vs. simulation time

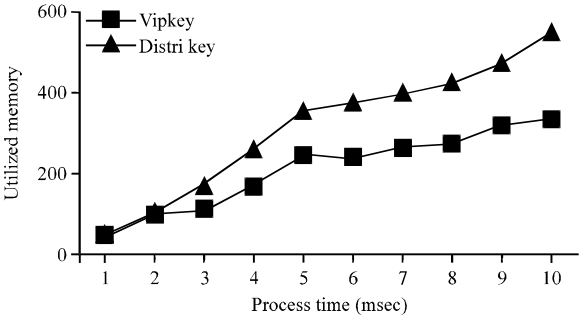

Fig. 13: Utilized memory vs. process time

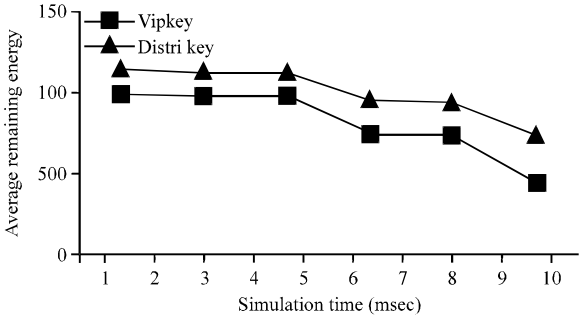

Fig. 14: Average remaining energy vs. simulation time

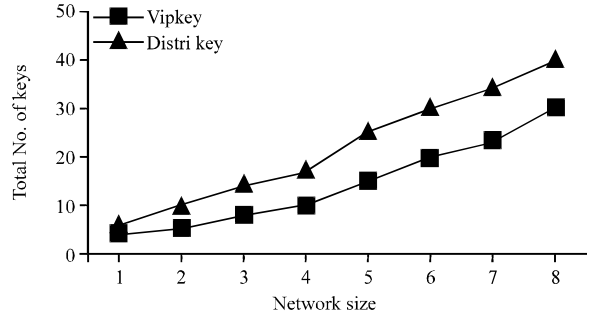

Fig. 15: Total numbers of keys vs. network size

Table 1: Simulation parameter

Parameter	Values
Number of nodes (N)	100
Network coverage	1500×300
Initial energy	20 J
Data packet size	512 bytes
Simulation time	50-100
Cluster size	200-1000

The experimental analysis is made with ViPKey consisting of 3 virtual group and more than 100 msec. Thus all the values which are necessary for the study are collected and simulation of sensor network is made according to the outcome of the experiment and accuracy in performance evolution shown clearly. All these works of experimental studies were carried out with NS-2 which is a network simulator kit version 2.27 where the virtual group key model standards have verified through code and the parameters of the resultant graph is extracted through NS-2. The area of operation with initial energy of 20 J and base station analysis are made under coverage region of 1500×300 consisting of 100 mobile nodes and their data size of 512 bytes packet size sent between those cluster heads shown in Table 1 in detail. Thus all the values which are necessary for the study is collected and simulation is made according to the outcome for the experiment and accuracy in performance evolution are given evidently.

CONCLUSION AND IMPLICATION

ViPKey delivers elevated liveliness of secure group key sharing across the transmission medium with efforts from polynomial expression that reduces the number of keys within a virtual group as almost one. Hence, achieved least amount of keys involving in a single group and reduced rekeying and updating process during member modification inside a virtual group of a sensor network. This suggests that the proposed model and utilization of optimization ViPKey may be useful in terms of dynamically altering member group thereby increasing profitability of increased throughput, efficient usage of generated keys, reduced overall end to end delay and reduced power consumption. Thus our system achieves high performance in key management system under digital sensor transmission involving encryption and decryption of shared information.

REFERENCES

Harkins, D. and D. Carrel, 1998. The internet key exchange (IKE). http://tools.ietf.org/html/rfc2409

He, W., Y. Huang, R. Sathyam, K. Nahrstedt and W.C. Lee, 2009. SMOCK: A scalable method of cryptographic key management for mission-critical wireless Ad-Hoc networks. IEEE Trans. Inform. Forensics Secur., 4: 140-150.

Lin, J.C., K.H. Huang, F. Lai and H.C. Lee, 2009. Secure and efficient group key management with shared key derivation. Comput. Standards Interfaces, 31: 192-208.

Mukherjee, A., A. Gupta and D.P. Agrawal, 2005. Totally distributed key management for dynamic groups in MANETs. Proceedings of the 24th IEEE International Performance, Computing and Communications Conference, April, 7-9, 2005, Phoenix, Arizona, USA, pp: 185-192.

Piao, Y., J. Kim, U. Tariq and M. Hong, 2012. Polynomial-based key management for secure intragroup and intergroup communication. Comput. Math. Appli., 10.1016/j.camwa.2012.02.008

Shamir, A., 1979. How to share a secret. Communi. ACM, 22: 612-613.

Staddon, J., S. Miner, M. Franklin, D. Balfanz, M. Malkin and D. Dean, 2002. Self-healing key distribution with revocation. Proceedings of the Symposium on Research in Security and Privacy, May 12-15, 2002, IEEE CS, Berkeley, California, pp: 241-257.

Tshering, F. and A. Sardana, 2011. A review of privacy and key management protocol in IEEE 802.16e. Int. J. Comput. Appli., 20: 25-31.

Wang, W. and T. Stransky, 2007. Stateless key distribution for secure intragroup and intergroup multicast in mobile wireless network. Comput. Networks, 51: 4303-4321.

Yi, S., P. Naldurg and R. Kravets, 2001. Security-aware *ad hoc* routing for wireless networks. Proceedings of ACM International Symposium on Mobile *ad hoc* Networking and Computing, (MADNC'01), ACM, New York, USA., pp: 299-302.