Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

Group Key Management Protocol for Securing a SCADA System

Mrs. Vaishali Patil
PG Scholar, Department of Computer engineering, Thakur College of Engineering and Technology, Mumbai

Mrs. Veena Kulkarni
Assistant Professor, Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai

Mrs. Harshali Patil
Assistant Professor, Department of Computer Engineering, Thakur College of Engineering and Technology, Mumbai

***Abstract*— SCADA (Supervisory Control and Data Acquisition System) requires a secure group communication between control center and remote terminal units. In this paper, we have proposed an improvised group key management system to secure the group communication in SCADA system. The proposed scheme optimizes the storage overhead of the existing system without compromising security of the system. The GKMP lifecycle and rekeying interval has chosen such that frequent initialization should not be required which will reduce the burden of communication cost on the system.**

Keywords—SCADA, Group key management, key distribution centre.

I. INTRODUCTION

Supervisory Control And Data Acquisition (SCADA) is a system which is used for monitoring, controlling and analyzing the industrial processes or operations being carried out at multiple remote locations from a central location/s. These systems are being used in various industrial sectors such as power, oil and gas, transportation, water and waste water etc. In SCADA system remote field equipment are monitored and controlled with the help of remote terminal units (RTUs) over various communication channels. Group key management mechanism is proposed to secure the communication between RTUs and control center.

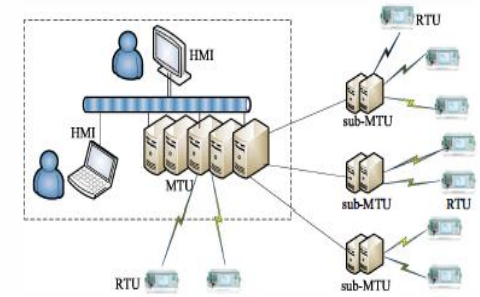
II. BASIC SCADA ARCHITECTURE

Basic SCADA architecture contains several communicating entities as shown in fig.1. In this architecture MTUs monitors the remote devices by sending a status request to RTUs, and in turn RTUs respond to MTUs by sending the required information periodically or by exception.

HMI (Human Machine Interface): It is the user console based on GUI. HMI enables user to monitor and control the various control processes defined in the SCADA system for effective and efficient operation of the remote system in the field. It analyses the received data and processes it and presents to user in such a way that user can take the precise and quick decisions in no time.

MTU (Master Terminal Unit): MTU is the main control device which provides supervisory control of multiple RTUs. Main SCADA control center delegates its powers to MTU. Hence MTU also act as data concentrator for SCADA control center. MTUs are physically secured devices as they are located at the provider's site. MTUs send commands to RTUs to gather the data of remote devices on behalf of SCADA control center.

RTU (Remote Terminal Unit): These devices are composed of a microprocessor based controllers. These controllers gather data from various sensors and control various end equipment with the help of auxiliary devices like actuators, heavy duty relays (HDR) etc. They acquire data from both digital and analog sensors, process the signals and send it to the MTU. Unlike MTUs these RTUs are physically not secured due to challenges and constraints of the environment in which they are installed at remote site. So, it becomes imperative to secure information transmitted and received from SCADA control center.



ig.1.Basic SCADA Architecture

III. SECURITY CHALLENGES AND REQUIREMENTS FOR SCADA SYSTEM

There are many challenges related to security of SCADA system that have been observed and identified in recent years, **Increasing connectivity with business networks:** In view of generating near real-time MIS/dashboards for different verticals of enterprise, it has become inevitable to interconnect control system network with enterprise network. As a result control systems have become vulnerable to attacks through enterprise network. Also it is not possible for any enterprise to setup their own dedicated communication network for MTUs and RTUs. Hence they are forced to avail these network services from third party and it is difficult to monitor the activities in the third party network. In the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

absence of adequate security mechanism control system is exposed to very high risk of cyber-attacks.

Wireless communication: Remote devices are physically located at consumer's site and in the absence of enterprise owned communication medium at those sites, wireless communication services offered by various telecom operators becomes the most convenient. As in wireless communications it is very easy to intercept the packets and they are more vulnerable to cyber-attacks.

Technology standardization: In order to minimize the operational cost and enhance system scalability and longitivity, most of the SCADA systems are implemented using standard communication protocols. On one side standard protocols offers very high flexibility of integrating multi- vendor systems, but on the other side these protocols are very much vulnerable to cyber-attacks.

IV. Group Key Management Protocol

Group Key Management Protocol (GKMP) is the fundamental mechanism that provides and implements the security framework for group communications by managing the keys. SCADA systems are being used for many critical applications for organizational and social benefits. Hence these SCADA systems become prime targets for terrorists and anti-social organizations to be disrupted by executing cyber-attacks. Hence it has become imperative to secure the communication between MTUs and control center or between RTU and MTU.

The minimum requirements for secure group communication in SCADA devices are as follows:

***Group Confidentiality*:** Only authorized users i.e. group members can have access to the key that can be used to decrypt any data transmitted within the group. Group communication keys should be secure from unauthorized users.

***Backward Secrecy*:** Members can leave or join the group over time. So newly joined member should not have access to the previous keys that can be used to decrypt the data in previous sessions.

***Froward Secrecy*:** If any member leaves the group, it should not be able to access any keys after leaving the group.

***Collusion Free environment*:** A newly joined member can attack in coalition with removed member called as collusion attack. So, the group communication should be free from collusion attack.

V. EXISTING WORK ON GROUP KEY MANAGEMENT

In the beginning authors in [1] have proposed a key management scheme SKE to secure group communication in SCADA system which was based on symmetric key and public key but with low cost security. Also an efficient multicast and broadcast is not supported in this scheme which is an important part of the SCADA system. SKMA [2] also faces the same problem. ASKMA [3] overcomes the issues in SKMA but it is less efficient for multicast communication. ASKMA+ [4] was proposed specially to secure the multicast communication in SCADA system. This scheme provides secure broadcast and multicast communication by using a logical key hierarchy [5] and the Iolas framework scheme [8], but it does not provide collusion free environment and suffers from availability problem. In [11] authors have proposed another key management architecture for SCADA system by considering the availability feature of the system. This scheme uses replacement protocol for availability, but the drawback is system stops working during the replacement.

In addition to re-keying and the node revocation, some recent works address the self-healing issue that a group node can recover the missed session keys from the latest re-keying message on its own. A self-healing group key distribution scheme first presented by authors in [17] based on entropy theory. Later it was improved in [9][10]. Although these self-healing schemes are secure, many of them suffers from heavy overload. In [11] authors have proposed a key distribution scheme for secure group communications in wireless sensor networks which provides self-healing group key scheme with time limited node revocation based on dual directional hash chains which assures forward and backward secrecy. Although this scheme is better in terms of storage and communication cost, it is not collusion resistant and it has limited revocation ability. In [16] authors proposed a new GKM scheme for securing group communications in wireless ad hoc networks which improvises the previous self-healing key distribution schemes by using vector space access structure to reach more flexible performance of the scheme in terms of storage overhead and communication and computation cost. Although this scheme provides forward and backward secrecy, it could not resist collusion attack. Authors in [15] have proposed a robust group key management scheme, called LiSH(Limited Self-Healing) which is more secured as compared to other schemes but have more storage overhead and communication cost.

In our proposed system we have optimized this storage overhead and communication cost without compromising security of the system.

VI. PROPOSED SYSTEM

The existing GKM scheme i.e. LiSH[15] provides better security of group communication in SCADA system in smart grid. But as per the performance analysis this scheme has more storage overhead and communication cost as compared to other scheme. So, our proposed system improvise this existing LiSH [15] key distribution scheme by minimizing the storage overhead and communication cost without compromising security.

In the existing LiSH [15] scheme the storage overhead is due to the number of keys required to be stored. In this proposed system we are computing the hash chains, which are required to calculate session key, on the fly. There is no need to store the whole hash chains which is done in previous system. This reduces storage as well as computation cost. To minimize the communication cost we can add a new agent i.e. KDC which will handle new joining RTUs and distributes the load of communication as shown in Fig.2.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

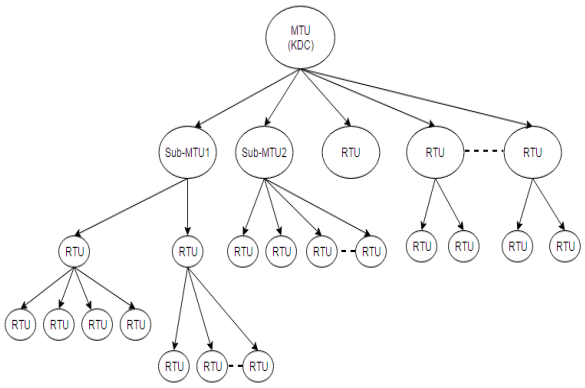


Fig.2.Distributed architecture for improvising LiSH scheme.

The proposed scheme consists of a secured group communication between Key distribution Centre i.e. KDC(located in MTU or sub-MTU) and RTUs. KDC collects status information of each RTU node and RTU sends the periodic information to KDC. The proposed scheme contains following phases:

Initializtion: Initially a network is built by authenticating the RTUs by KDC based on some identification. Here KDC sends some queries by using asymmetric encryption to RTU and RTU response with answers. Upon checking response from RTU, KDC authenticates that RTU. If the RTU is a legal node then KDC provide it a key encryption key(KEK) to that leagal RTU. If authentication fails KDC does not except any data or send any key to the illegal node. Here KEK is a key encryption key which is required to encrypt TEK i.e. traffic encryption key required for sending or reciving information or commands from KDC to RTU. TEK is different for each session. The lifetime of the network is divided into sessions. TEK is also called as session key.

Session key generation: In the preprocessing time, KDC selects two randomley generated key seeds, forward key seed fw and backward key seed bw. To produce two hash chains, on these two key seeds KDC repeatedly aplies same hash function of equal length z+1. So forward hash chain generated as { fw, H(fw),…,H^i(fw),…,H^z(fw)} and backward hash chain generated as { bw , H(bw),…, H^i(bw),…, H^z(bw)}.

During the lifecycle of a RTU from t1 to t2, KDC assigns a pair of hash chains (H^{t1}(fw), H^{t2}(bw)). So, for the GKM system of lifecycle 0 to z, the traffic encryption key(TEK) at the time x can be defined as,

$TEK_x = f(H^x(fw), H^{z-x}(bw), RK_x)$, $0 \le x \le z$

Here, RK_x is a secret number for session x which is generated by KDC and it is periodically send to RTU by KDC during the session based upon its timer.

KDC randomly selects a RK seed and generates a RK hash chain by applying a one way hash function. Each session will have a different and unique RK. During the transmission if a rekeying message lost, RTU can recover the lost RK by using one way hash funtion and the last RK received i.e. with the self healing mechanism.

Initial Group Communiction: At the initial phase to initiate rekeing parameters InitTEK massege is sent to nodes by KDC. For this communication each node n keeps a key encryption key(KEK) for message encryption and its authentication. At the time t_{init}, KDC sends the following message to node n whose lifecycle is (t1,t2).

KDC→ n : { E_{KEK}(RK_buf_t, RK_$T_{refresh}$, H^{t1}(fw), H^{t2}(bw), rk2, kdc_start_time, dt1), MAC(RK_buf_t, RK_$T_{refresh}$, H^{t1}(fw), H^{t2}(bw), rk2, kdc_start_time, dt1)}

Where RK_buf_t is the length of RK buffer, RK_$T_{refresh}$ is the rekeying period, H^{t1}(fw) and H^{t2}(bw) are initial forward and backward key seeds with lifecycle (t1, t2), rk2 is initial RK and dt1 is the current time and date.

In the proposed scheme RK buffer can be variable according to the number of nodes. An operator can set the buffer size according to the network latency which reduces the storage cost.

When node n receives this message first it decrypts the meassage and check if message is tampered or not by calculating and comparing the message digest received with the meassage. If meassage is not tampered it allocates a key buffer of size RK_buf_t and two key slots. Then it calculates RK sequence and stores it in key buffer and two key slots are filled with two most recent RKs. Sets a parameter RK_w which tracks the most recent inactive RK and RK_e which tracks the number of RKs that a node fails to receive.

RK in one key slot is used for TEK generation which is called as active RK and RK for next session will be stored in other key slot. When key update $T_{refresh}$ timer is triggered, the node switches the active key slot to the one with the new RK in other key slot as shown in Fig.3.

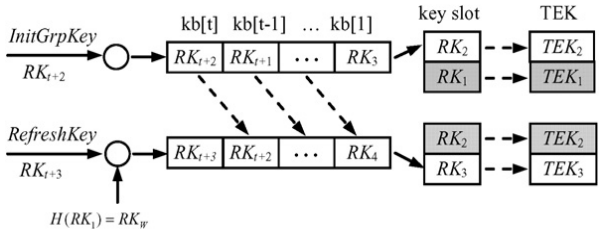


Fig.3. Initial setup and TEK refreshment

TEK and RK refreshment: Rekeying is the process of refreshing RK. KDC periodically sends next RK in the key sequence to nodes. Accordingly TEK is updated with new RK. Upon receiving new RK, node verifies it by camparing its hash with RK_w. If its hash value matches then it is valid RK otherwise it is tampered.

If some RKs lost during the communication, a node(RTU) can recover those lost RKs by self healing technique. Suppose RTU receives RK_1, RK_2, RK_3 continuously. Here number of lost rekey messages RK_e=0. If in the next two intervals rekey messages lost then RK_e=2. After that, the node receives an authentic rekey message with RK_6. Now node can recover the lost keys as $RK_4=H^2(RK_6)$, $RK_5=H(RK_6)$, since $H(RK_3)=RK_w$. On recovering lost RKs, RTU derives corresponding TEKs.

TEK is used by KDC to send commands to RTUs and to receive data and status information from RTUs. Counter is maintained to check the packet loss.

Request Key: When a node fails to receive RKs upto t intervals of $T_{refresh}$, it sends a request key message to KDC to get current RK in the sequence. Upon getting request key message KDC reinitialize the group communication for the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

requesting node by sending initGroupKey message with current configuration.

Re-initialization of system: The system is re-initialized when all the RKs in the RK sequence have been used up and when a GKM explicitly receives request key message from a node.

VII. SECURITY AND PERFORMANCE ANALYSIS

We implemented the proposed system and gathered the results by simulating different scenarios that may exist in real environment. The scenarios were created considering variation in no. of nodes (50 to 2000), rate of data being exchanged, communication reliability (60% to 100%), duration of GKMP time slots, rekeying interval, buffer size (t) and simulation of malicious nodes. The analysis of the data gathered was done to evaluate the performance of the system. The analysis results are classified as follows:

Security Analysis: In the proposed system, each session uses unique TEK among all the nodes in the group. Also the TEK is refreshed periodically at the set key refresh interval. The minimum key refresh interval we tested is 4 minutes with 2000 nodes in the group. The TEK is derived using forward key, backward key and Refresh Key, which are again unique and dynamic for all nodes with time. Suppose a node n joins the system at time interval t1 to t2. This node n can generate its TEK at time interval x ($x \geq t1$ and $x \leq t2$) by using its initial key seeds $H^{t1}(fw)$ and $H^{z-t2}(bw)$ and RK_x for the current session. So the TEK for node n at time x in lifecycle of GKMP system 0 to z is derived as,

$$TEK_x = f(H^x(fw), H^{z-x}(bw), RK_x)$$

Our proposed system satisfies all the criteria that are supposed to be complied by any GKMP, including collusion resistant criteria.

Apart from security compliances, our system demonstrates the satisfactory performance in timings for end-to-end data exchanged even at the group size of 2000. The results for the same are as follows:

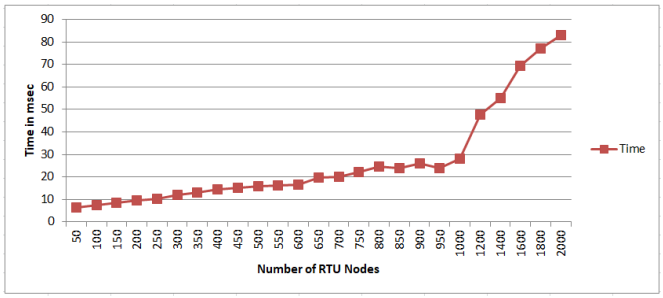


Fig.4. End-to-end system performance with network size

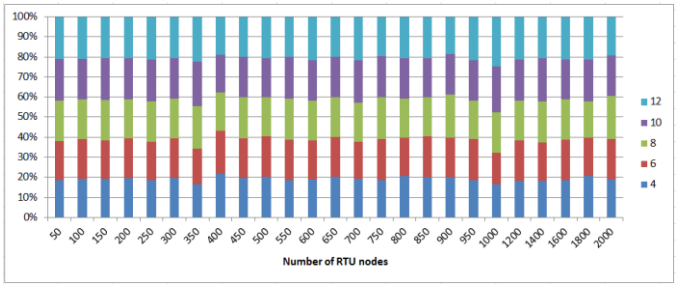


Fig.5. Impact of RK buffer size on End-to-end system performance

Above results in Fig-4 signify that the performance of proposed system is in-line with the anticipated performance as well as within the tolerable limits of real-time monitoring and control systems. In Fig-5 it is also seen that for a given network/group size of RTUs there is no significant impact of RK buffer size on the performance of the system. Thus RK buffer size can be independently choosen based on the capapility of hardware, in order to develop, the more resilient system design.

Storage Cost Analysis: In our proposed system storage cost has been optimized both at KDC and RTU. At KDC end single hash chain for rekeying has been implemented, instead of having individual chains for all the nodes. However to achieve unique refresh key a unique base RK seed key for all the nodes was generated and using this seed key and a common key from rekeying hash chain, a unique refresh key was generated for all nodes by applying a set function.

At RTU node end the storage requirement is only for base forward hash key at time interval t1, base backward hash key at time interval t2. All the subsequent hash keys for applicable time intervals are derived from these base keys. Apart from this, RTU node is also required to store advance rekeys based on size of the buffer (t). In our implementation we tested the implementation with maximum buffer size of t=10. There is no need to store the entire RK chain.

With all the above optimization we have observed satisfactory performance with minimal deterioration of time as shown in the results above in Fig-4.

Communication Cost Analysis: Cost of communication depends upon following aspects:

GKMP lifecycle duration: GKMP lifecycle is the time after which entire system needs to be re-initialized. System initialization/ re-initialization are the event which requires maximum communication cost, as working of GKMP mainly depends on the successful initialization of system. Hence GKMP lifecycle should be chosen such that frequent initialization should not be required.

Rekeying interval: Rekeying interval is the duration in which refresh keys are sent from KDC to RTU nodes. If rekeying interval is too short, then there is a risk of all the RKs available with RTU getting exhausted in a high loss of communication scenario. In the case of all RKs exhausted, the system for that node needs to be re-initialized. If this happens frequently with significant number of nodes, this will drastically increase the cost of communication.

Probability and duration of loss of communication: Reliability of communication determines the frequency of re-initialization of GKMP for nodes, where all RKs get exhausted due to non-receipt of consecutive k numbers of RKs on account of loss of communication.

In our proposed system we have chosen the above parameters so that cost of communication can be optimized. With the combination of GKMP lifecycle, rekeying interval and RK buffer (t) cost of communication can be minimized.

Stability and scalability of the system: During the various iterations of our proposed system under the various simulated scenarios with various constraints imposed, we have not observed any failure of the system. Hence we have confirmed that the system is stable under adverse scenarios.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

During the tests we have observed that beyond 2000 nodes the timing of end-to-end data exchange starts deteriorating, hence single instance of the system cannot cater large number of nodes. However this can be implemented with distributed architecture to scale up to more number of nodes. Hence the system is easily scalable.

VIII. CONCLUSION

Though we have achieved reduction in storage cost in our proposed implementation, but as we all know that with the advancements in microprocessor based controllers, storage constraints are no more a challenge today. Hence we can leverage the microprocessor capability of computing and storage to further optimize the parameters like lifespan of the GKMP, refresh key interval and RK buffer in RTU node, such that cost of communication can be further reduced. These capabilities can also be leveraged to further strengthen the security aspect. All in all more robust and stable GKMP system can be implemented even under the context of unreliable communication mediums within the practical limits.

REFERENCES

[1] C. Beaver, D. Gallup, W. Neumann, and M. Torgerson, "Key management for SCADA", *Tech. Rep., Cryptog. Information Sys. Security Dept., Sandia Nat. Labs, SAND2001-3252. 2002.*

[2] R. Dawson, C. Boyd, E. Dawson, and J. M. G. Nieto, "SKMA: a key management architecture for SCADA systems", *in Australasian workshops on Grid computing and e-research, 2006, pp. 183-192.*

[3] D. Choi, H. Kim, D. Won, and S. Kim, "Advanced key-management architecture for secure SCADA communications", *IEEE Trans. on Power Delivery, vol. 24, no. 3, pp. 1154-1163, 2009.*

[4] Donghyun Choi, Sungjin Lee, Dongho Won, and Seungjoo Kim, "Efficient Secure group communications for SCADA", *IEEE transaction on power delivery, vol. 25, no. 2, April 2010*.

[5] C.K. Wong, M.G. Gouda, and S.S. Lam, "Secure Group Communications Using key Graphs", *IEEE/ACM transactions on networking, vol. 8, no.1, February 2000.*

[6] Marcel Waldvogel, Germano Caronni, Dan Sun, Nathalie Weiler, and Bernhard Plattner, "The Versakey Framework: Versatile Group Key Management", *IEEE Journal on selected areas of communications, 1999.*

[7] M. Eltoweissy, H. Heydari, L. Morales, et al. "Combinatorial Optimization of Key Management in Group Communications", *Journal of Network and Systems Management: Special Issue on Network Security, March 2004.*

[8] S. Mittra, "Iolus: A framework for scalable secure multicasting", in *Proc. ACM SIGCOMM*, 1997, pp. 277–88.

[9] C. Du,M . Hu,H . Zhang, and W. Zhang, "Anti-collusive self-healing key distribution scheme with revocation capability", *Information Technology Journal, vol. 8, no. 4, pp. 619--624, 2009.*

[10] B. Tian,S . Han,T. S. Dillon,a nd S. Das, "A self-healing key distribution scheme based on vector space secret sharing and one way hash chains", *IEEE WoWMoM, 2008, pp. 1-6*.

[11] Y. Jiang,C . Lin,M . Shl,a nd X. S. Shen," Self-healing group key distribution with time-limited node revocation for wireless sensor networks", *Ad Hoc Networks, vol. 5, no. 1, pp. 14-23, 2007.*

[12] Y. Wang, "sSCADA: securing SCADA infrastructure communications", *International Journal of Communication Networks and Distributed Systems, vol. 6, no. 1, pp. 59-78, 2011.*

[13] D. Choi, H. Jeong, D. Won, and S. Kim, "Hybrid key management architecture for robust SCADA systems", *Journal of Information Science and Engineering, vol. 27, pp. 197-211,2011.*

[14] Rong Jiang, Rongxing Lu, Chengzhe Lai, Jun Luo and Xuemin (Sherman) Shen, "Robust Group Key Management with Revocation and Collusion Resistance for SCADA in smart grid", *IEEE Global Communications Conference (GLOBECOM),* 2013

[15] Rong Jiang, Jun Luo, Fang Tu, Jin Zhong, "LEP : A Lightweight Key Management Scheme based on EBS and Polynomial for Wireless Sensor Networks", *IEEE International Conference on Signal Processing, Communications and Computing, 2011, pp. 1-5.*

[16] R. Dutta, S. Mukhopadhyay, and M. Collier, "Computationally secure self-healing key distribution with revocation in wireless ad hoc networks", *Ad Hoc Networks, vol. 8, no. 6, pp. 597-613, 2010.*

[17] J. Staddon, S. Miner, M. Franklin, "Self-healing key distribution with revocation", in: Proceedings of IEEE Symposium on Security and Privacy, 2002, pp. 241–257.

[18] C. Beaver, D. Gallup, W. Neumann, and M. Torgerson, "Key management for SCADA", Tech. Rep., Cryptog. Information Sys. Security Dept., Sandia Nat. Labs, SAND2001-3252. 2002.