2011 International Conference on Computational Science and Its Applications

Development of Smart Devices for Secure Communication in the SCADA system

Sugwon Hong, Tran Nhat Phuong, and Myongho Lee
Dept. of Computer Science and Engineering
Myongji University
Yongin, Korea
swhong@mju.ac.kr, phuongnhatit@gmail.com, myunghol@mju.ac.kr

***Abstract*—As the power grid evolves the smart grid, the security in the SCADA system becomes even more important than ever. The basic security requirement is to guarantee the secure exchange of messages between the nodes in the system. The primary security concerns for the secure operations are to guarantee the message authenticity and integrity. For this purpose the several documents which explain the security measures and protocols have been published by the international standard organizations for secured SCADA systems. But developing the substation devices which implement the security function is also challenging task considering the functions should be implemented mostly in embedded system with limited computing resources. In this paper we address the implementation issues when the both security function and the communication function are implemented on the embedded system devices in future power grid including the SCADA network. We propose an approach to implement these functions on a general purpose multi-core microprocessor platform and show the experiment results.**

Keywords-SCADA system, cyber security, IED, CMP, multi-core processor

I. INTRODUCTION

The main purpose of the supervisory control and data acquisition (SCADA) system is gathering real-time data, monitoring and controlling equipments and processes in the critical infrastructure. A SCADA network provides connection between servers which reside inside a control center and control devices which are located at substations, sometimes at remote locations.

Major concerns about cyber attack stem from the notion that the SCADA network is no longer an isolated network which prohibits outsiders from entering the network, nor is the specialized network based on private platforms and protocols, allowing only technical staffs with special knowledge to access to the resources. The reasons are twofold. First, the communication architecture is more relying on the open standard communication protocols. The use of the open communication protocols renders the system more vulnerable to cyber attacks in many applications.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2009-0089793)

Second, more importantly, the SCADA network is moving toward being connected to other networks including cooperate networks for convenience and other business reasons. In a bigger picture, the SCADA network is an integral part of the smart grid which spans from power generation, transmission, and distribution systems to advanced metering infrastructure (AMI) networks, and energy consumer home networks. Thus the SCADA network may open its doors to outsiders who can enter any entry point in the smart grid.

For this reason SCADA security issues have drawn attention in various levels, and the security measures and protocols have been studied and proposed by the industry as well as the academia. In addition to the works in industry and academia, the international standard bodies also have worked to derive the standard documents for the SCADA security [1].

Along with the development of security measures required to devices in the SCADA system, how to implement the security function to these devices is also a challenging task, considering most substation devices operate on embedded systems with limited computing power and exist as unmanned systems.

Recently, microprocessor designers have been considering many design choices to efficiently utilize the ever increasing effective chip area with the increase of transistor density. Instead of employing a complicated processor pipeline on a chip with an emphasis on improving single thread's performance, incorporating multiple processor cores on a single chip, or multi-core processor, has become a main microprocessor design trend [2, 3]. As a Chip Multi-Processor (CMP), it can execute multiple software threads on a single chip at the same time. Thus a multi-core processor provides a larger capacity of computations performed per chip for a given time interval (or throughput).

In this paper we propose an approach to implement the security function with other functions together on general purpose multi-core processor platform which enables to execute multiple software threads on a single chip at the same time.

In the following two sections we explain the security issues required in the devices in the SCADA system briefly. In section IV we introduce the parallel model for implementing the security function and in the following section we show the results of parallel processing, comparing with the case of the serial execution case.

978-0-7695-4404-5/11 $25.00 © 2011 IEEE
DOI 10.1109/ICCSA.2011.58

II. SECURITY REQUIREMENTS FOR SUBSTATION DEVICES

Main components of the SCADA system are intelligent electronic devices (IED), power equipments, and substation controller. The substation controller as a master station, located in a central site, monitors and supervises a large number of IEDs which are field devices located in physical environments. IEDs gather data from sensors which measure current and voltage, and send data to the substation controller. The actuator as a part of IED controls the operation of power equipments by commands issued by other IEDs. The substation controllers have a hierarchical structure. A high-level master station can control several sub-master stations.

The data and command transfer takes place between the substation controller and IEDs, between IEDs, or between IED and sensors (or switchgears). The transferred information is carried over the SCADA network. The SCADA network is based on various communication channels and network technologies including Ethernet, serial links, and wireless communication.

The communication between the devices is governed by the standard communication protocols. The most commonly used protocols are IEC 60870-5, DNP3 which is the derivative of IEC 60870-5, and Modbus [4]. Recently the International Electrotechnical Commission (IEC) is working on the new protocol, IEC 61850, which not only defines a new structure for substation automation, but also can provide more enhanced communication functionalities [5].

Communication between IEDs and various power apparatuses has traditionally carried out over hardwired I/O logic to gather physical data and control circuit breakers and switchgears. IEC 61850 replaces this hardwired connection with communication lines such as serial unidirectional multi-drop point to point link or the IEEE 802.3 LAN.

The secure transmission of the messages plays a critical role for normal protected operation in the substation. First of all, receivers need to verify that all messages are sent from claimed senders. IEDs should avoid receiving any illegitimate data or command from unauthorized nodes. Second, receivers should confirm that the message content is original, not altered during transmission for any reason, maliciously or inadvertently. The sample values in a special type of the message are used for the IEDs to decide whether voltage, current, or frequency anomalies happen. If these values are modified, the IEDs are mistaken to understand the current status.

The contents of messages can be leaked to outsiders. Even though information about operation data or commands is exposed to outsiders, this information leakage would not lead critical damage directly to the system operation unless IEDs are forced to function improperly. In some application, messages can deliver highly sensitive information such as secret keys which should be known to only the concerned parties. In this case the message contents should be protected from being exposed to outsiders.

Considering the special nature of the SCADA operations, the primary security concerns for secure operations are to guarantee the message authenticity and integrity. Based on the threat analysis, the following security requirements are considered to be basic and necessary conditions which should be achieved for the secure operation in the SCADA system.

- Receivers should be able to verify that messages they receive are from claimed senders.
- Receivers should be able to verify that messages they receive are not compromised in transit.
- Receivers should be able to verify that messages they receive are not replayed by any attacker.
- Critical contents of messages such as secret keys should be secured in transit.

III. KEY MANAGEMENT

The message authentication code (MAC) is a common method used to verify the authenticity of the sender and the integrity of the message. Since it can avoid the encrypting/decrypting computation, this method is preferable when we apply the authentication algorithm to the devices which have limited computing resources. The Keyed-Hashing (HMAC) is the most widely used algorithm for computing MAC [6]. The paper explains in detail how the MAC algorithm can be applied to the distribution automation system [7].

The key management including key establishment is an integral part of the security mechanisms. The big picture of the key management in the SCADA security mechanism is well summarized in the paper [8]. In the master-to-IED communication which is encountered when the communication is based on the DNP3.0, the key establishment protocols for this mode are based on the ISO/IEC 11770 Part2 server-less protocols [9]. In these protocols, a master station and an IED have a pre-shared symmetric key which is often called a long-term key in many literatures. Using this long-term key, two nodes establish an ephemeral key, which is also often called a session key and is used to encrypt and decrypt messages. The key establishment process takes 1 to 3 passes of message exchange, offering unilateral or bilateral authentication. In this procedure they use random number(nonce) and/or time stamp to protect the replay attack.

The secure DNP standard protocol has two kinds of modes: challenge-response mode and aggressive mode [3]. The challenge-response mode is the typical example based on the common key establishment protocol.

There are no specific remarks about any key establishment protocol in the standard documents for the peer-to-peer model. But some researchers have proposed key establishment protocols based on the symmetric cryptographic algorithm [10]. These protocols are variants of the Kerberos protocol or the ISO/IEC 11770-2 server-based protocols [11,12].

In these protocols, two nodes A and B assume the trusted third party(TTP) which distributes the shared secret key between A and B. When the TTP generates the shared key, the TTP acts as the key distribution center(KDC). On the other hand, when the shared key is generated by an initiating node, the TTP will be the key translation center(KTC). Since

random number generation requires complex computation, it is desirable for the master station to generate the key rather than an IED which normally has limited computer power.

The nodes A and B have the pre-shared keys with the TTP respectively. When a newly generated shared key between A and B(session key) is distributed, the session key is encrypted by the pre-shared key. The additional information such as nonce or time stamp or sequence number may be transmitted together with the key for verifying message freshness or preventing the Man-In-The-Middle attack. Normally the master station can act as the KDC or KTC. But the KDC can be located separately from the master station.

The asymmetric key cryptographic algorithm can also be used for the peer-to-peer model. The difficulty in implementing the public key cryptographic algorithm lies in maintaining the private certificate authority(CA) and processing the public-key certificates at each node.

As long as the underlying communication network in the SCADA system is multi-access, especially wireless, we can exploit the advantage of the broadcast communication. The most convincing rationale for using broadcast channels in the SCADA system is that the master station can reach all IEDs by propagating a single message. If an emergent shutdown happen, it would not be desirable for the master station to send a message individually [14].

Another reason to mention the broadcast in the SCADA system is that sensors will be deployed in the SCADA network rapidly in the near future. Researchers on the wireless sensor networks(WSN) have produced numerous protocols [13]. And since the constraints of the WSN are very similar to the SCADA system, the proposed key establishment protocol for broadcast channel can borrow the ideas in the sensor networks [14].

IV. APPROACHES FOR DEVELOPING SMART DEVICES

Devices in the SCADA system are expected to have electrical control function, communication function, and security function, all harnessed in a single box. Since the security functions eventually should be applied to most, though not all, incoming packets to guarantee secure operation in the power grid including the SCADA system. More demanding requirement is that these functions should be implemented mostly on the embedded, microprocessor-based platform. For this reason finding any feasible and economic way of implementing the security functions as well as handling incoming packets in the embedded system will pose very daunting task.

Figure 1 shows the basic functional blocks inside a device in the future power grid. First the device should capture all incoming packets without any loss if possible. Unfortunately capturing packets is not an easy task due to the kernel livelock in which the system spends all its time processing interrupts [15]. Over the past few years many efforts have been tried to improve the performance of packet capture and transmission, eliminating Kernel livelock while processing interrupts [16-18].

For capturing packets, especially high-rate arriving packets, one alternative is to use specialized hardware such as network processors in the monitoring cards. This expensive alternative is mostly adapted in developing network devices in real-life.

We face the same challenge in implementing cryptographic functionality these days. The security function on heavy traffic load is intimidating task to enterprise servers. To overcome the limitation of pure software implementation, some dedicated hardware, which may be FPGA-based co-processors or hardware accelerators or graphic processor (GPU), are used only for security functions.

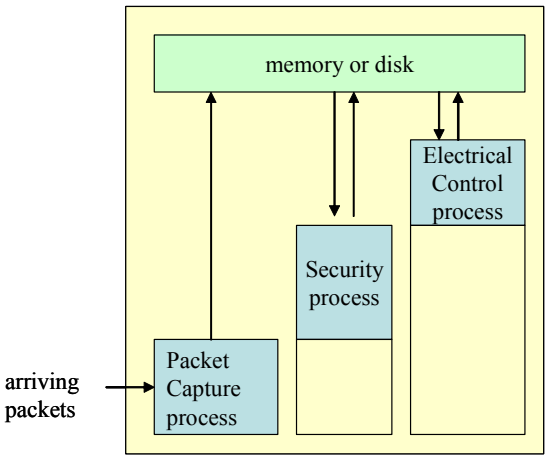

Figure 1. basic functions in a smart device

One possible and less expensive alternative for developing the network devices in the future is to use the general purpose multi-core processor, avoiding any specialized hardware, and consequently reducing development costs significantly. The current main design for multi-core processors is based on the Chip Multi-Processor(CMP). Some multi-core processors go one step further to incorporate Simultaneous MultiThreading (SMT) or similar technologies on a processor core. Figure 1 shows the architecture of an advanced multi-core processor. On each processor chip, there are N-processor cores, with each core having its own cache on chip. The N-cores share a larger cache on or off the processor chip. Each core also has *M* hardware threads performing SMT or similar features. Thus it supports two levels of parallelism. Also, it has a cache hierarchy of private (to each core) and shared (among threads).

However, to program for multiple interrelated application processes to execute in parallel is not an easy problem to solve. Some processes are not independent from the viewpoint of sharing data they have to use and synchronizing their tasks. For example, the process doing encryption function is working on the packets which are captured and moved into the user space buffer by the packet capture process. The task of the former process is triggered when packet copy is completed in shared memory by the latter process.

The proper parallel programming model for this purpose might be the producer-consumer model [19]. In this model a stream of data pass through successive processes, each of

which performs some tasks on data. A proceeding process can be considered as a producer of the data stream which the following process consumes. Production of new data by a producer process triggers the action of a task by a consumer process. A chain of producer processes and consumer processes can be viewed as a pipeline. One of drawbacks of this approach would be load balancing due to heavily coarse granularity. Because workload of each process is different and varied, some process will take longer time to produce or consume data, causing unbalanced CPU utilization between cores.

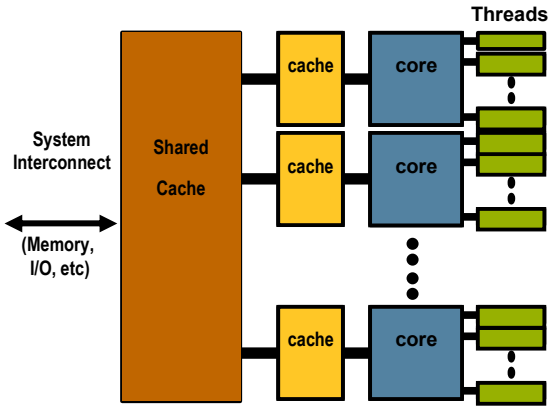

Figure 2. Architecture of an advanced multi-core processors

One of serious issues to implement this parallel model is management of access to shared data. As shown in the multi-core processor architecture in figure 1, when data is retrieved, data traverses from shared memory to dedicated cache to each core. Most multi-core processor have more than two-level caches that are called L1 and L2 cache that are dedicated to or shared between cores depending on which level the cache is. The cache hit rate which implicates data locality on memory affects performance significantly. When a process has to use data which are scattered widely on memory, it is likely to have low cache hit rate, which means to retrieve data from memory more frequently. One example is a packet-processing application that is performing TCP reassembly on a large number of TCP flows, which means to access a large number of data over many memory locations [20]. The result in the article [20] shows how much the number of TCP connections affects cache hit rate and throughput.

In addition to avoiding cache-trashing to minimize the number of packet copies between buffers in two processes is critical in exploiting parallel implementation. The occurrence of packet copy triggers another task in the pipelined model, which imposes the cost of synchronization.

V. EXPERIMENTS

In this experiment we execute a packet capture process and an encryption process in parallel based on the producer-consumer model explained in the previous section. We use the SEED algorithm as an encryption method which is a 128-bit symmetric key block cipher developed by Korea Information Security Agency in 1998, and has been since adopted by most of the security systems in Korea [21]. The parallel programming is implemented using OpenMP.

In this implementation to avoid excessive accesses to buffers and decrease the cost of synchronization, double buffers are used to store the packets in PF_PACKET and libpcap methods. At first encryption process must wait until one buffer is full. While the encryption process is accessing one buffer, arriving packets are being stored in another buffer. When one buffer is full, a coordinating function triggers the encryption process. When one buffer is full and encryption of packets in another buffer has not completed yet, captured packets are stored in temporary buffer until encryption is completed. There will be cost for copying from the temporary buffer to any buffer and for waiting buffer full at the initial time. But it can decrease the cost needed for synchronization when two CPU access the same buffer more frequently. Before storing packets into buffers, the length of packets is normalized to be aligned with multiple of 16 bytes.

In this experiment we use the same system as a receiver which is used for the packet capture experiment in figure 2 (Intel Core 2 Duo 2.2GHz CPU and 2GB RAM running the Centos 5.5 operating system). We generate 10,000,000 packets with the size of 512 bytes and arriving rate of 60Mbps. We obtain packet capture rates for three different cases as shown in table 1. Performance of the parallel implementation of two processes with respect to the serial execution is compared based on the packet capture rate, since it highlights the improvement of speed for processing packets with the help of parallelism.

The result of the test B shows how much the packet capture rate is degraded due to the encryption process running together in serial mode, comparing with the result of the test A. Comparison of the results of the test B and C shows that modest improvement of packet capture rate can be achieved when two processes are running on two cores in parallel mode. But we admit that this gain should be final, considering that there must be room of optimization of the codes to reduce the synchronization cost mentioned in the previous section.

TABLE I. PACKET CAPTURE RATE AT DIFFERENT MODES

	Test A	Test B	Test C
PF_PACKET	97.21%	83.69%	91.59%
LIBPCAP	87.49%	74.61%	82.59%
PF_RING	96.90%	84.31%	95.36%

- Test A is to capture packets without encryption.
- Test B is to capture packets and encrypt packets in serial mode.
- Test C is to capture packets and encrypt packets in parallel mode.

VI. CONCLUSION

The SCADA system is not immune to cyber attacks any more, especially when we are moving into the smart grid. Most devices in the smart grid are required to have security

function with communication function as basic capability. How to implement these functions on embedded system is as important as what to do for secure communication.

Implementing multiple operations including security function in a general-purpose multi-core processor requires careful design of a parallel model and consideration to reduce the synchronization cost caused by packet copy and cache-trashing. In this paper we experiment one parallel model to implement a packet capture process and an encryption process in parallel on two CPU cores, and show performance gain that can be achieved by parallelization.

Although multi-core processors promise to deliver higher throughput performance than the traditional single-core processors, resources on the multi-core processors such as caches, cache/memory bus, functional units and the like are shared among the cores on the same chip. Software processes or threads running on the cores of the same processor chip compete for the shared resources, which can cause conflicts and hurt performance. Thus exploiting the full performance potential of multi-core processors is still a challenging task

REFERENCES

[1] V. M. Igure, S. A. Laugher, and R. D. Williams, "Security issues in SCADA networks," Computers & Security Vol. 25, pp. 498-506, 2006.

[2] L. Spracklen and S. Abraham, Chip MultiThreading: Opportunities and Challenges, 11th International Symposium on High-Performance Computer Architecture (HPCA-11), pp 248-252, 2005..

[3] Y. Li, D. Brooks, Z. Hu, K. Shadron, "Performance, Energy, and Thermal Considerations for SMT and CMP Architectures," 11th International Symposium on High-Performance Computer Architecture, 2005.endon, 1892, pp.68–73.

[4] DNP User Group, http://www.dnp.org.

[5] IEC, "Communication networks and systems in substations- Part 1: introduction and overview," IEC TR 61850-1, 2003.

[6] H. Krawcryk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[7] I. H. Lim, S. Hong, M. S. Choi, S. J. Lee, T.W. Kim, S. W. Lee and B. N. Ha, "Security Protocols against Cyber Attacks in the Distribution Automation System," IEEE TRANS. ON POWER DELIVERY, VOL. 25, NO. 1, JANUARY 2010.

[8] L. Pietre-Cambacedes and P. Sitbon, "Cryptographic key management for SCADA systems – issues and perspectives," International Conference on Information Security and Assurance, 2008.

[9] ISO, Information Technology – Security Techniques – Key Management – Part 2: Mechanisms Using Symmetric Techniques, ISO/IEC 11770-2, 1996.

[10] R. Dawson, C. Boyd, E. Dawson, and J.M.G. Nieto, "SKMA-A Key Management Architecture for SCADA Systems," Proceedings of the Australasian workshops on Grid computing and e-research, 2006.

[11] C. Beaver, D. Gallup, W. Neuman, and M. Torgerson, "Key management for SCADA," Technical Report, SANDIA, 2002.

[12] B. Clifford Neuman and T. Tso, "Kerberos: An authentication service for computer networks", IEEE Communications Magazine, Vol. 32, No. 9, pp33-28, September 1994.

[13] S.A. Camtepe and B. Yener, "Key Distribution Mechanisms for Wireless Sensor Networks: a Survey," TR-05-07, Dept. of Computer Science, Rensselaer Polytechnic Institute, 2005.

[14] Y. Wang and B.-T. Chu, "sSCADA: Securing SCADA Infrastructure Communications," http://eprint.iacr.org/2004/265.pdf.

[15] J. C. Mogul and K. K. Ramakrishnanl "Eliminating receive livelock in an interrupted-driven kernel," ACM Trans. On Computer System, Vol. 15, No 3, pp217-252, 1997.

[16] L. Rizzo, "Device Polling support for FreeBSD," the EuroBSDCon 2001.

[17] L. Deri, "Improving passive packet packet capture: Beyond device polling," the 4th Int. System Administration and Network Engineering Conference 2004.

[18] L. Deri, "nCap: Wire-speed packet capture and transmission," the IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services 2005.

[19] M. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw Hill, 2004.

[20] E. Verplanke, "Understand packet processing with multi-core processors," EE Times-India, April 2007.

[21] H.J.Lee, S.J. Lee, J.H.Yoon, D.H.Cheon, J.I.Lee, "The SEED Encryption Algorithm," IETF RFC 4269, December 2005.

.