A method for software trusted update on network security equipment

IOP Conf. Series: Materials Science and Engineering **569** (2019) 052086 doi:10.1088/1757-899X/569/5/052086

A method for software trusted update on network security equipment

Guan Wang[1,2], Zhiwei Yan[1,2], Jianzhong Chen[1,2]

[1] Faculty of Information Technology, Beijing University of Technology, Beijing, 100124 Beijing, China

[2] Beijing Key Laboratory of Trusted Computing, Beijing University of Technology, Beijing, 100124 Beijing, China

*Corresponding author's e-mail: wangguan@bjut.edu.cn ,1957855254@qq.com

Abstract. Network security equipment plays an important role in preventing network security attacks. However, with the change of network environment and the upgrade of network attack means, the protection ability of the security software on the existing network security equipment will gradually decrease with time. For some large organizations or enterprises, their information security operations centers lack attention to software version control and software update processes for network security equipment, resulting in some security crises in the software update process. In this paper, we propose a method for software trusted update on network security equipment. This method can provide trusted identify authentication, secure data transmission and effective software version control. It enables the network security operations centers to more safely manage software update process on network security equipment. This method uses the functions of Trusted Cryptography Module to provide trusted execution environment. In this paper, we had introduced the process design, the prototype design and theoretical analysis to explain the feasibility and safety of this method.

1. Introduction

The security defense software running on the hardware of the network security equipment is the core of the equipment, whose performance and reliability will determine whether the equipment can undertake the mission of fighting against network threats[1-3].

In recent years, some large organizations or enterprises have begun to build the Information Security Operations Center (ISOC), these center devices can improve the overall defense capability of inner network system and reduce the workload of system maintenance personnel though centrally managing to operate decentralized network security equipment[4-5]. After we had researched and analyzed the composition structure of information security operations center, we found some security defects in this system. Some information security operations centers lack attention to software version control and software update processes for network security equipment, so that some virus-infected software update package had been installed or the process data of the software update had been hijacked by attackers. In this paper, we propose a software update method for network security equipment, and this method is suitable for information security operations center system. We utilize the cryptography function of Trusted Cryptography Module (TCM) to realize trusted identity authentication and secure data transmission between devices, so that it ensures that the process of software update can't be hijacked and the software update package come from the trusted central devices[6] .

The rest of this paper is organized as follows. In section 2, we give the relevant work description; in section 3, we give the overall design method; in section 4, we give specific solutions to several key problems of the method; in section 5, we give the prototype design and theoretical analysis of the method; in section 6, we summarize the whole paper.

2. Related work

Some researchers from all over the world have done a lot of research in the field of software update research. We divide their research into two categories: theoretical research of software update method and engineering application research of software update.

For the theoretical research of software update method, the dynamic update of software has been the main aspect of research in recent years. Dynamic software updating (DSU) system supports updating software without interrupting software services[7].In the article [8], Haibo Chen evaluated the security of the traditional DSU and proposed a solution to improve DSU by applying a lightweight consistency model to solve the problem that DSU method is difficult to deal with the dynamic update of multithreaded software, which can solve the state consistency effectively when DSU updates multithreaded software[9]. In the paper [10], Tianxiao Gu proposed an improvement method of DSU, which pays more attention to the error repair in the process of software update. This method refers to ARR technology[11], simplifies the complexity of DSU technology and improves the reliability of DSU technology. In the paper [12], Petr Hosek and Cristian Cadar proposed a novel multi-version execution approach for improving the software update process. This approach is an innovation of DSU, they run the new version in parallel with the old one, and carefully synchronizes their execution to create a more secure and reliable multi-version application. Their solution enables users to benefit from the additional features and bug fixes provided by recent versions, without sacrificing the stability and security of older versions. For the engineering application research of software update. In the paper [13], Kapilan Kulayan proposed a software update method of infotainment system applied to the automobile industry, which adopts integrity measurement and digital signature technology to ensure the safety of the installed software for the automobile industry[14-15].In this paper, our research belongs to engineering application research of software update for network security equipment under ISOC management. We had found that existing ISOC method has two defects in the process of software update on network security equipment, one is weak identity authentication,and the other one is unreliable data transmission.

TCM is a kind of physical chip with cryptographic operation function and data storage function[16].TCM is developed in China and has similar functions to TPM,it can meet the requirements of a highly demanding trusted execution environment[17-18].

3. The entities in software trusted update method

In this research, we add a software service center into the ISOC to be responsible for the software update service of all network security equipment. The Software Service Center includes three key entities: Software Update Issue Software (SUIS), Software Service Database (SSD)and Equipment Service Database (ESD). The Network Security Equipment includes two entities: Security Defense Software (SDS) and Software Update Management Software (SUMS).

As shown in figure 1, there is only one Software Service Center in the Information Security Operations Center, but there are a lot of security defence devices(NSE) in the security system.

Software Service Center (SSC): It is in charge of the software update tasks for all the Network Security Equipment in the security system. In addition, the SSC should accept the unified management of ISOC.

Software Update Issue Software (SUIS): It is in charge of the management and coordination tasks of software update service to all NSE.SUIS can provide identity authentication, integrity verification, reliable data transmission of software update package and other services to all NSE.

Software Service Database (SSD): It is in charge of synchronously storing the integrity measurement benchmark library of all Network Security Equipment. It can provide the data records for version control and integrity measurement of software.

Equipment Service Database (ESD): It is in charge of storing devices relevant information about all NSE under the management of ISOC. In addition, it can support SUIS to obtain the device information about the specified NSE.

Network Security Equipment (NSE): It is responsible for protecting network security. It can be a firewall device or intrusion prevention system device. It is the basic unit of security system under the management of ISOC.

Software Update Management Software (SUMS): It is in charge of managing software updates on NSE and providing the function to safely communicate with SSC. In addition, it is also in charge of the installation process of the software, and the software only can be installed after been authorized by SUMS.

Security Defense Software (SDS): It undertakes various network security protection tasks on NSE and is the basic unit of NSE's software system. The version ID and hash value of each SDS will be recorded in the software integrity measurement benchmark library, and the format of a record in the library is shown in figure 2.

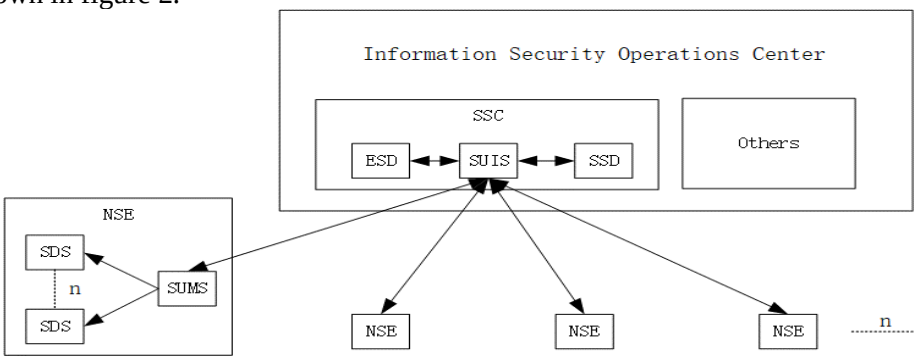

Figure 1. The relationship between entities in software trusted update method.

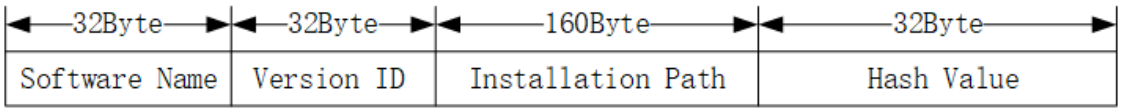

Figure 2. The data field format of the software integrity measurement benchmark library.

In the software integrity measurement benchmark library, each record is 256 bytes, corresponding to the software name, the version ID, the installation path, and the hash value used for integrity verification. Each record would be written into the benchmark library by SUMS after the software had been installed. In addition, SUMS can verify the software whether it is trustworthy though integrity measurement. If it judges the software is untrustworthy, it will prevent the software from installation and write the record into the benchmark library.

4. The process of software trusted update method

4.1. The Definition of the basic concepts

The SSD records the installation information of software on all NSE in the security system. ISOC gives each device a unique device identification code, namely Client ID (CID). Similarly, SSC as part of the ISO, it has a unique server identification code(SID).With SID and CID, the hierarchy between devices can be distinguished.

Both SSC and NSE need to use built-in TCM's Endorsement Key (EK) to create a Platform Identity Key (PIK), PIK represents platform identity at the level of cryptography and is used to sign data generated inside TCM[19-20]. ISOC can as the trusted third parties, it issues PIK certificates to all PIK for building a trusted network. TCM can provide hash algorithm SM3, symmetric encryption

algorithm SM4 and asymmetric encryption algorithm SM2[21-23]. In addition, all operations of cryptography occur inside the TCM, so that the calculated process is more secure and reliable.

In order to explain the software trusted update method more clearly, we first define the basic concepts related to the software update service process:

Definition 1: PIK public key and private key created by SSC and NSE represent as pk_{SSC}, sk_{SSC}, pk_{NSE}, sk_{NSE} respectively, encryption and decryption are respectively expressed as Enc_X and Dec_X, where x is the key used for encryption and decryption;

Definition 2: the software set in NSE represents as S, and $S=\{s_1, s_2, ...s_i...s_n\}, i\in[0,n]$, where n is the number of software on NSE, and s_i is a security defense software running on NSE;

Definition 3: the software integrity measurement benchmark library in NSE is expressed as Lib_{NSE}, and $Lib_{NSE}=\{sn_i, VID_i, inp_i, Hash(s_i)\}, i\in[1,n]$, where sn_i represents the name of the software s_i, VID_i represents the version identification of the software s_i, inp_i represents the installation path of the software s_i, and $Hash(s_i)$ represents the hash value of the software s_i;

Definition 4: the software registration record of the software s_i in the NSE is expressed as $lib_{NSE}(s_i)$, and $lib_{NSE}(s_i)=\{sn_i, VID_i, inp_i, Hash(s_i)\}$;

Definition 5: the temporary session keys created in SSC and NSE communication are represented as $SesK_{SSC}$ and $SesK_{NSE}$, the random number generated by SSC and NSE is represented as rdm;

Definition 6: $A\rightarrow B$ represents the process of entity A sending data to entity B, the request message is represented as MAC, and the request message of different attributes are different MAC. The software update packages are represented as $appdata$.

4.2. The establishment process of software update service relationship

When NSE attempts to build connection with SSC for the first time, NSE needs to send data containing the CID request message to the SSC. After SSC has received the message data, it will decrypt the data and the verify the message. If the SSC makes sure the message comes from the legitimate NSE, the SSC will request for the Lib_{NSE} of this NSE. When the Lib_{NSE} is received, the SSC will write the Lib_{NSE} into the SSD. The detailed process is as follows:

(i) SUMS sends a message data to SUIS for requesting the registration of software update service: $Enc_{pk_{SSC}}\{CID, MAC, pk_{NSE}\}\rightarrow SUIS$;

(ii) After SUIS gets the message data, it will obtain the CID and pk_{NSE}, the network address and port number of the NSE, then it writes these data into the ESD.

(iii) SUIS sends a temporary session key to SUMS:

$Enc_{sk_{NSE}}\{SID, MAC, SecK_{SSC}\}\rightarrow SUMS$;

(iv) SUMS decrypts the message data by using sk_{NSE}, and saves the SID into the TPM non-volatile memory.

(v) SUMS uses $SesK_{SSC}$ to encrypt the reply message and sends the message data to SUIS:

$Enc_{SecK_{SSC}}\{CID, MAC\}\rightarrow SUIS$;

(vi) SUIS decrypts the message data, if SUMS finds that the MAC contains the information about establishing the software update service relationship, it will send request information to obtain the Lib_{NSE}: $Enc_{SecK_{SSC}}\{SID, MAC\}\rightarrow SUMS$;

(vii) SUMS decrypts the message and sends a reply message to SUIS:

$Enc_{SecK_{ssc}}\{CID, MAC, Lib_{NSE}, Enc_{sk_{NSE}}\{Hash(Lib_{NSE})\}\}\rightarrow SUIS$;

(viii) SUIS decrypts the message and verifies the integrity of the Lib_{NSE}. If the signature has been authenticated successfully, SSC will record the message into the SSD and send a reply message data to SUMS: $Enc_{SecK_{SSC}}\{SID, MAC\} \rightarrow SUMS$;

(ix) After the SUMS has received the message data, the software in NSE will accept SSC's centralized management.

4.3. The trusted transmission process of software update package

When there is a new software update package that needs to be pushed for installation, SUIS will search the NSE device record in SSD to get a list of NSE devices that need to be updated the software, and communicates with multiple NSE simultaneously. After the two parties have successfully verified each other's identities, the SSC will send the software update package for the NSE. The detailed process is as follows:

(i) SUIS sends software update request message data to SUMS in NSE:

$Enc_{pk_{NSE}}\{SID, MAC, SecK_{SSC}\} \rightarrow SUMS$;

(ii) The SUMS decrypts the message data from SSC received, and then sends a verification message data containing random numbers rdm to SUIS:

$Enc_{pk_{SSC}}\{CID, MAC, rdm\} \rightarrow SUIS$;

(iii) SUIS decrypts the message data, gets the random numbers rdm and sends a reply message to SUMS: $Enc_{SecK_{SSC}}\{CID, MAC, rdm\} \rightarrow SUMS$;

(iv) When the rdm in decrypted message data is matched with rdm generated in step (ii) and the identity of SID has been thought legitimate SSC, SUMS will make sure the SUIS is from the SSC and check whether the software installation records with the software name or the version ID of the corresponding software name are available, then SUMS sends a message to SUIS:

$Enc_{SecK_{SSC}}\{CID, MAC\} \rightarrow SUIS$;

(v) SUIS receives the message and sends the software update package to SUMS:

$Enc_{SecK_{SSC}}\{SID, MAC, appdata, Enc_{sk_{SSC}}\{Hash\{appdata\}\}\} \rightarrow SUMS$;

(vi) SUMS decrypts the data and verifies the integrity of the software installation package firstly. If the verification passes,it can explain that the transmission process is successful.Finally,SUMS a reply message is sent to SUIS: $Enc_{SecK_{SSC}}\{CID, MAC\} \rightarrow SUIS$.

4.4. The trusted synchronous update of software version record

When the SUMS has received the software update package from SUIS, it will update the software, and then update the local Lib_{NSE}.When SUMS updates the Lib_{NSE},it search Lib_{NSE} for the software's name and version number firstly, and then modifies the specific software information of the updated software in Lib_{NSE} or inserts a new software registration information record $lib_{NSE}(s_i)$ into the Lib_{NSE}.When SUMS has updated the Lib_{NSE} successfully,it needs to submit the latest Lib_{NSE} to SUIS. The synchronize process of Lib_{NSE} is as follow:

(i) SUMS sends a message data to SUIS for requesting synchronization update Lib_{NSE}:

$Enc_{pk_{SSC}}\{CID, MAC, SecK_{NSE}\} \rightarrow SUIS$;

(ii) SUIS verifies the identity of SUMS and sends random numbers rdm to SUMS:

$Enc_{pk_{NSE}}\{SID, MAC, rdm\} \rightarrow SUMS$;

(iii) SUMS decrypts the message data and sends the reply message containing the rdm to SUIS:

$Enc_{SecK_{NSE}}\{CID, MAC, rdm\} \rightarrow SUIS$;

(iv) SUIS decrypts the message data and verifies whether it is the same as the random number rdm generated in step (iii),if the verification is successful, it will send a reply message data to SUMS: $Enc_{SecK_{NSE}}\{SID, MAC\} \rightarrow SUMS$;

(v) If the MAC means permission to synchronize data, it will send its Lib_{NSE} to SUIS:

$Enc_{SecK_{NSE}}\{CID, MAC, Lib_{NSE}, Enc_{sk_{NSE}}\{Hash\{Lib_{NSE}\}\}\} \rightarrow SUIS$;

(vi) SUIS will find the corresponding information of NSE in the SSD of the SSC according to CID, and it will write the data in the form of overwrite.

5. Prototype design and analysis

5.1. The prototype design of the method

In order to test the feasibility of the method, we provide the prototype design of the method. As the figure 3, the following prototype design structure diagram is designed to simulate the processing flow between the server of the SSC and a single NSE.

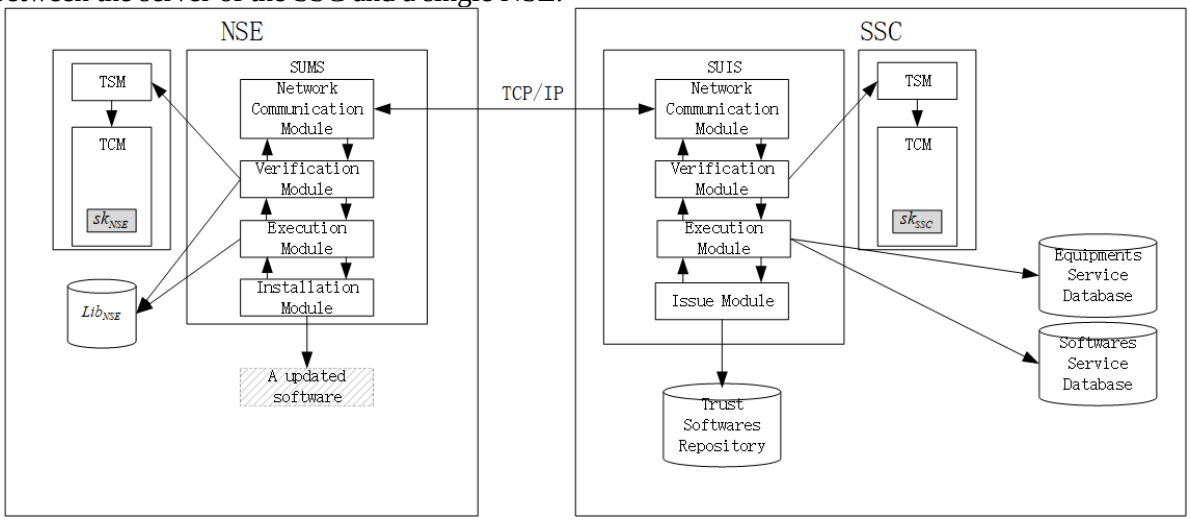

Figure 3. Prototype design structure of software trusted update method.

As the figure 3, the SUMS and SUIS are divided into several modules, each module is responsible for different function. They use TCP/IP protocol to communicate each other through the *Network Communication Module*. The *Verification Module* in SUMS and SUIS are responsible for executing the integrity measurement, digital signature and identity authentication and other tasks.the module needs to use the hash algorithm and encryption and decryption algorithm in TCM. Due to the function only been invoked by SUMS and SUIS through the upper TSM (TCM service module) protocol, so *Verification Module* will actually exchange data with TSM directly.The TCM in the NSE and SSC stores sk_{SSC} and sk_{NSE} respectively, and the TCM is bound to the devices, so the private key of PIK cannot be accessed and obtained externally.

In NSE,the *Execution Module* is responsible for querying and modifying Lib_{NSE} and the *Installation Module* is responsible for booting the Installation process of the software update package.The *Verification Module* is also responsible for querying the Lib_{NSE} to verify the integrity of software.The *Installation Module* is responsible for booting the Installation process of the software update package

In SSC,the *Execution Module* is responsible for querying and modifying the ESD and SSD. The *Trusted Software Repository* is responsible for storing some important softwares for NSE, which is a very secure and reliable storage space. The *Software Issue Module* is responsible for managing the *Trusted Software Repository* and issues new software update**.**

5.2. The theoretical analysis

Some information security operations centers lack attention to the software update process of network security equipment,thus the result may result in some serious consequences. In this method, some security problems can be solved:

(i) Trusted identity authentication. Third-party devices cannot pretend to be ISOC and send fake messages and modified software update packages data to network security equipment. Because that the TCM chips are bound to the SSC and all NSE, the chip internal PIK private key as the core authentication identity information of the devices, no attackers can obtain the PIK private key, thus the attackers cannot go through the authentication process.

(ii)Secure data transmission. Data communication in the process of software update uses the cryptography functions of TCM. All cryptography operations based on TCM are performed inside the chip, and the execution process is non-interference. The combination of identity authentication and secure cryptography operation can effectively protect the software update package from being hijacked and tampered in the transmission process.

(iii)Effective version control. The SSC of ISOC records the software installation information of all NSE, so that it is continent for ISOC to manage the software version information and the process of software update. In addition, it will be harder for an attacker to modify the security defense software, because that the software integrity measurement benchmark library has recorded the hash value of software for integrity verification. Timed integrity checks can detect modified software.

6. Conclusion

The software vulnerabilities can be repaired and the operating efficiency can be enhanced by updating the security protection software on the network security equipment. Unfortunately, at the moment, some information security operations centers lack attention to the process of software update.So that the inappropriate software update methods have brought system security crisis. In this paper, we propose a kind of software update method for the network security equipment in ISOC's security system. we apply the cryptography functions of the TCM to realize trusted identity authentication, secure data transmission and effective version control for the process of software update. Finally, we gave the prototype design and theoretical analysis to explain that the method can enhance the security of software update process for network security equipment. Our next work is to study how to realize the monitoring of software running state on network security equipment, which will enable network security equipment to timely find the software failure.

References

[1] Stytz, M. R. Whittaker, J. A. (2003) Software protection: security's last stand?. Security & Privacy IEEE, 1(1):95-98.
[2] Condon E, Cukier M. (2007) Applying Software Reliability Models onSecurity Incidents.In: The 18th IEEE International Symposium on Software Reliability. Trollhattan, Sweden. pp.159-168
[3] Zhang, Kun Y. (2014) Computer network security threats and security technology research. Advanced Materials Research, 971-973:1440-1443.
[4] Miloslavskaya N. (2018) Information security management in SOCs and SICs. Journal Intelligent & Fuzzy Systems. pp. 2637-2647.
[5] Eldardiry O.M., Caldwell B.S. (2015) Improving information and task coordination in cyber security operation centers. In: IIE Annual Conference and Expo 2015. Proceedings pp. 1224-1233.
[6] Wang J., Shi Y., Peng G., et al. (2016) Survey on Key Technology Development and Application in Trusted Computing. China communications (English version). pp.70-90.
[7] Seifzadeh H., Abolhassani H., Moshkenani, M.S. (2013) A survey of dynamic software updating. Journal of Software Evolution and Process, 25(5), 535-568.

[8] Chen H., Yu J., Hang C., et al. (2011) Dynamic Software Updating Using a Relaxed Consistency Model. IEEE Transactions on Software Engineering, 37(5):679-694.
[9] Trümper J., Bohnet J. (2010) Understanding Complex Multithreaded Software Systems by Using Trace Visualization. In: International Symposium on Software Visualization. Salt Lake City, United States.pp.134-142.
[10] Gu T., Zhao Z., et al. (2017) Improving Reliability of Dynamic Software Updating Using Runtime Recovery. In: Software Engineering Conference. Hamilton, New Zealand pp.257-264.
[11] Carzaniga A., Gorla A., et al. (2013) Automatic Recovery from Runtime Failures. In: 35th International Conference on Software Engineering. San Francisco, CA, USA.pp.782-791.
[12] Hosek P., Cadar C. (2013) Safe software updates via multi-version execution. In: Internation Conference on Software Engineering. IEEE.pp.612-621.
[13] Gandhi K.K.A., Arumugam C. (2017) An approach for secure software update inInfotainme-nt system. In: the 10th Innovations in Software Engineering Conference. Jaipur, India. 127-131.
[14] Ziwen L. TPM-Based Dynamic Integrity Measurement Architecture[J]. Journal of Electronics & Information Technology, 2010, 32(4):875-879.
[15] Dods, C., Smart, N.P. (2005) Hash Based Digital Signature Schemes. Cryptography and Coding. Cirencester, United Kingdom. pp.96-115
[16] Xiaofen C., Dengguo F. (2010) Model checking of trusted cryptographic module. Editorial Board of Journal on Communications. 31:59-64+72.
[17] Trusted Computing Group.(2011) TPM Main Specification. https://www.trustedcomputing group.org/resource/tpm-main-specification.
[18] Trusted Computing Group. (2016) TPM 2.0 Library Specification. https://trustedcomputing group.org/resource/tpm-library-specification.
[19] China National Standards, GB/T 0011-2012, (2013) Functionality and interface specification of cryptographic support platform for trusted computing, Standards Press of China. Beijing,
[20] China National Standards, GB/T 0012-2012, (2013) Interface specification of trusted cryptography module. Standards Press of China. Beijing, China.
[21] Zhenwei Z., Guoqiang B. (2014)Exploring the speed limit of SM2.In: 3rd IEEE International Conference on Cloud Computing and Intelligence Systems. Shenzhen, China. pp. 456-460.
[22] Zhibo D.,Zhen W. (2016) Power analysis attack of HMAC based on SM3. Editorial Board of Journal on Communications.37:38-43.
[23] Jian Z., Wenling W. (2016) Security of SM4 against (related-key) differential cryptanalysis. In: 12th International Conference on Information Security Practice and Experience. Zhangjiajie, China.pp.65-78.