Journal of Engineering and Applied Sciences 12 (Special Issue 12): 9502-9507, 2017
ISSN: 1816-949X

Assessment of Packet Header for Security Support in Trivial File Transfer Protocol

Nur Nabila Mohamed, Yusnani Mohd Yussoff, Habibah Hashim
Faculty of Electrical Engineering, Universiti Teknologi MARA, Shah Alam, 40450 Selangor, Malaysia

Abstract: The Trivial File Transfer Protocol (TFTP) has received remarkable attention as a significant medium for transferring information on embedded Machine-to-Machine (M2M) system due to its lightweight features and compatibility. However, major limitation is the lack of security feature where its reliability is questionable, especially during the process of data collection and system update. Therefore, to strengthen TFTP communication protocol, a new key exchange technique based on well-known Diffie Hellman Key Exchange (DHKE) algorithm is proposed to generate a shared secret key. The proposed technique is integrated with TFTP packet header for optimal solution to provide a secure TFTP communication in M2M technology. In this research, simulation of TFTP packet transmission is performed using NS3 as an initial finding for security support in the protocol. This research makes major contribution which has potential to enhance TFTP capability in constrained M2M communication field.

Key words: Trivial file transfer protocol, option extension, Diffie Hellman key exchange, NS3, communication, technology

INTRODUCTION

Internet of Things (IoT) is considered as the world's third wave for current and future research areas, involved by both academia and industry sectors. M2M communication is among the key technologies in the scope of IoT. Due to rapid growth in M2M technology, some approaches have been proposed to improve M2M component for outstanding utilization in various related field (e.g., e-Health, smart home, smart vehicle). One of the most critical components of future M2M systems may be a device known as M2M gateway which aggregates and processes sensor data before sending it onward to the users, tolerates between sensor protocols and so on. Current M2M gateway operates as platforms for application code that processes data and becomes an intelligent part of a device-enabled system (Persson *et al*., 2011). Due to too many data that has been received, researchers have introduced a data collection agent (Wang *et al*., 2015) which resides in the gateway device to obtain, accumulate and store various sensor data from various devices as illustrated in Fig. 1. The data collection agent communicates with the database server using a lightweight communication module by implementing TFTP protocol to reduce the communication load and cost. However, problem arises as TFTP has flaws for trusted connectivity and security in order to ensure the integrity of the network and system in both directions.

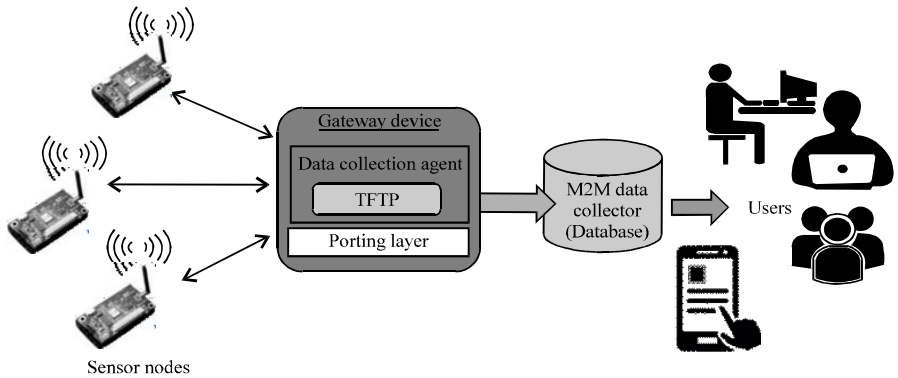

Fig. 1: TFTP implementation in M2M system

Corresponding Author: Nur Nabila Mohamed, Faculty of Electrical Engineering, Universiti Teknologi MARA, Shah Alam, 40450 Selangor, Malaysia

Therefore in this research, a lightweight and compatible security mechanism based on TFTP option extension is proposed to enhance the communication protocol. Specifically, the aim of this research is to devise and reconstruct the packet anatomy for TFTP security support as a means to add security feature in the protocol.

MATERIALS AND METHODS

Security requirement in M2M system: It is believed that to successfully deploy the communications systems for M2M technology, security requirement must be satisfied to prevent compromise in the transmission process (Nagamani and Veni, 2016). In order to defend against the threats and establish a secure M2M communications environment, several security requirements that should be achieved are explained as follows (Hossain *et al*., 2015; Lu *et al*., 2011; Cha *et al*., 2009).

Confidentiality: The confidentiality and secrecy of the revealed and stored information should be strictly protected to ensure only the authorized entities have access to the data in the M2M communication system. During the communication, devices may transmit highly sensitive information such as key distribution, thus, it is crucial to employ a feasible security mechanism to preserve against attack. Moreover, in some cases, the identities of the devices and the public keys sharing between devices should also be protected (e.g., using encryption) to some extent in order to secure the data privacy.

Integrity: It is necessary to meet the integrity requirement in M2M system to prevent illegal data alteration such as delaying, replaying and modifying the information. Apart from that, the adversary can alter the whole packet stream during transmission process by injecting or inserting additional flawed data. This result a serious consequences especially in life-critical application such as in elderly monitoring at home, e-Health, etc.

Authentication: Authentication is prerequisite for allowing the application domain to corroborate data of the M2M device, also to verify that data is originally sent by the correct sender to the correct client.

Non-repudiation: Non-repudiation guarantees that M2M machine device cannot deny the transmission it has previously sent. It refers to the ability of the communicating party cannot deny the authenticity of the signature on message it sent.

Availability: Availability ensures that the sever domain is always available whenever M2M application systems access into it.

Privacy: If sensitive information is illegally disclosed or improperly used such as in e-Health care systems, it can cause undesirably negative effects on patient's lives.

In general, the above security requirements in M2M communications can be achieved by various cryptographic techniques. For instance, symmetric and asymmetric encryption primitives can be employed to achieve the confidentiality while the digital signature and Message Authentication Code (MAC) methods can achieve information authentication. These security mechanisms are developed or integrated with M2M system based on the security properties that the system wants to achieve. Therefore, the M2M system features should reflect the characteristics of the security requirement and reveal the related security problems. Referring to related work in securing TFTP in (Isa *et al*., 2012) the security extension was proposed using two cryptographic principles, symmetric and asymmetric encryption as a preliminary step on securing TFTP packet. Meanwhile, an integration of key exchange mechanism with TFTP protocol has been proposed by Mohamed *et al*. (2013, 2014) for the implementation in data encryption/decryption process to provide a significant enhancement in pervasive computing and IoT applications for managing and upgrading embedded infrastructure (Hongsong *et al*., 2011; Kowshayla and Valarmathi, 2016). Besides, a pre-shared technique for exchanging a shared secret was proposed by Mohamed *et al*. (2016, 2017) to significantly improve the TFTP communication performance. According to these previous research, it is believed that targeted security solutions to each aspect of security problems should be based on the protocol's feasibility and compatibility.

RESULTS AND DISCUSSION

TFTP option extension overview: In M2M communication, the communicating device must incorporate with a protocol to enable data transmission operation. TFTP has been generally used for booting M2M nodes from the network (Finlayson, 1984), updating software/firmware (Qiu *et al*., 2008), transferring and collecting data from M2M devices (Iitsuka *et al*., 2012) and so on. Several modifications have been added to enhance the protocol features. Malkin and Harkin (1998) have introduced a simple TFTP extension to allow some option negotiations prior to the file transfer. The option mechanism must be in accordance with TFTP's request-respond-acknowledge sequence which is similar to the lock-step approach. If the

communicating parties want to negotiate multiple options, the options are appended to each other as long as thelength of total request packet is <512 bytes. Meanwhile, if the server supports and agrees with the option negotiation specified in the request packet, it may reply with an Options Acknowledgment (OACK). The extensions that are appended to TFTP Read Request (RRQ) or Write Request (WRQ) packet:

Option extension for read or write request format:

|Opc|Filename|0|Mode|0|optN|0|ValueN|0|

- Opc: the opcode field contains either a 1 for RRQ or 2 for WRQ
- Filename: the name of the file to be read or written as defined by Persson *et al*. (2011)
- Mode: the mode of the file transfer is in "netascii", "octet" or "mail"
- OptN: the option extension (e.g., blksize, windowsize)
- ValueN: the value associated with the option in case insensitive ASCII

In 1998, the blocksize option extension (Blksize) (Harkin and Malkin, 1998) was firstly introduced as the general 512 bytes blocksize is insufficient to be used on local network which require hose MTU of 1500 bytes or greater. This option enables valid blocksize values to be extended until 65464 bytes. If the blocksize is greater than the amount of data to be transferred, the first packet is considered as the final packet to terminate the protocol operation.

Besides, the option for timeout was introduced in Harkin and Malkin (1995). This extension allows the client and server to agree on a specified number of seconds to use for their retransmission timers. Valid values range is between 1 and 255 bytes. Another option, TFTP windowsize option (Masotta, 2015) enables the client and server to negotiate a window size of consecutive blocks to send as an alternative for replacing the old single-block lockstep schema. The reception of the data window with a number of blocks less than the negotiated windowsize will be the final window. Based on the above mentioned option extensions, the proposed security extension should be backward-compatible with the general protocol, also the size of the packet shall be consistent with TFTP's request packet format.

TFTP security extension: Firstly, we elaborate the DHKE operation which is illustrated in Fig. 2. In order to secure TFTP communication protocol, the communicating parties need to do the key exchange first to share the secret key. Figure 2 explains thoroughly the DHKE operation using the communication between client and server as example. Both parties agree on p and g public integers where p is a large prime number and g is a generator of p. They choose the positive personal values, a and b which have not been transmitted over public medium. They will then compute the public values, based on their personal values according to. They can share their public A and B, over insecure communication channel. From these public integers, a key can be generated by either communicating user on the basis of their own personal values where the value of the key turns out to be the same. This technique enables both client and server to generate the exact same secret key without transferring the key in physical manner but the limitation of this operation is the vulnerability to Man-In-The-Middle (MITM) attack where the adversary can intercept all the parameters passing through the public channel and sending its own public value in which finally can compromise the entire communication.

In this research, we aim to achieve the integrity and confidentiality of the information exchange by introducing the security extension using modified DHKE-based key exchange technique in TFTP packet header as illustrated in Fig. 3. Alice and Bob are two communicating parties acted as client and server. At Alice's side, the security

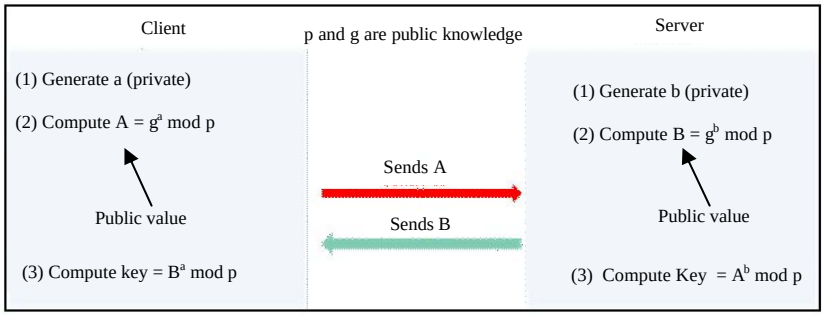

Fig. 2: Diffie Hellman key exchange overview

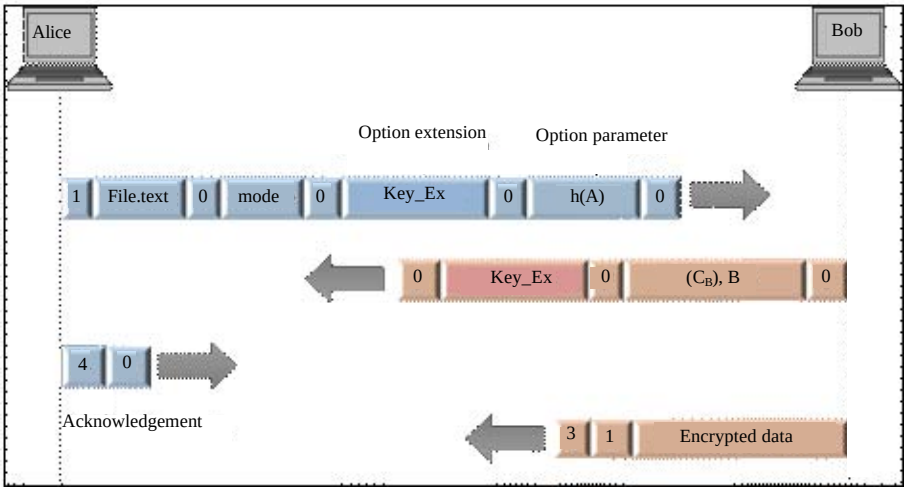

Fig. 3: TFTP with security extension

extension is structured by blending the public value for key exchange and cryptographic hash function to maintain data integrity. While at Bob's side, the public parameter is encrypted using previous secret key and will be hashed using the same hash function to protect the data confidentiality and integrity. To facilitate the TFTP support for enabling the key exchange, the header format and its frame should be adapted according to TFTP feasibility. Specifically, the support for Key_Ex must be implemented in the existing message format of TFTP protocol as in Fig. 3. The option extension packet (Key_Ex) is an option to send request to Bob to exchange the parameter for secret key. Value A packet contains the Alice's public parameter which is hashed with cryptographic hash function SHA256. The total length is 202 bytes where 128 bytes (1024 bits size) is for public parameter plus additional of 20 bytes (160 bits size) for hash value and the remainder 54 bytes is the length for mode, null-terminated, file name and opcode field length.

Next, a simulation of sending TFTP's RRQ packet header is performed to assess the transmission time of proposed security extension which will be compared with TFTP's Blksize request packet. The system runs on Laptop Intel® Core™ i3-3217U CPU @ 1.80 GHz with 4 GB of memory. The basic software environment is Linux (Ubuntu 14.04), NS3 (Release 3.26), G++ (GCC 4.8.4) and Python (2.7). The simulation parameters are described in Table 1 and the comparison of blksize and Key_Ex header size is depicted in Fig. 4. In this research, one character string is assumed 2 bytes length. The total length of request packet header with proposed security extension (Key_Ex) is 202 bytes, meanwhile the total length of blksize request packet is 58 bytes. From Fig. 5, it can be

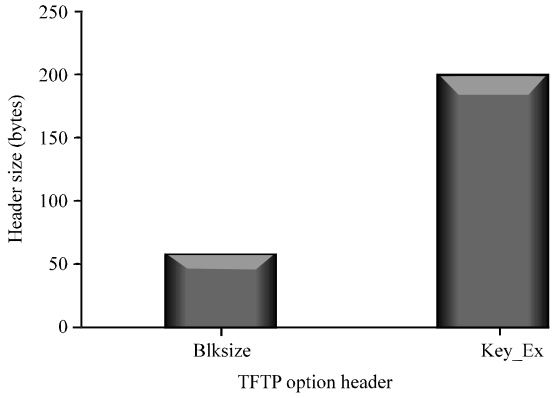

Fig. 4: TFTP Blksize and Key_Ex header size comparison; Time to send RRQ with Key_Ex option = 0.06431 sec; Time to send RRQ with Blksize option = 0.02832 sec

Table 1: TFTP simulation parameters

Simulation parameters	Description
Client's name	Device 0
Client's IP	10.1.1.1
Server	Device 1
Server's IP	10.1.1.2
TFTP's option extension	Blksize, Key_Ex
File	File.txt
Mode	Octet
Packet size	58 bytes and 202 bytes

seen that the size of the proposed header's length (Key_Ex) is approximately 3.5 times larger than Blksize option header. This is due to additional security parameters in the Key_Ex header packet.

From the simulation, it can be seen that the execution time to send Key_Ex option is 0.06431 sec while it takes about 0.02832 sec ecto send Blksize option header. Although, it shows high gap which may affect the

performance of RRQ's transmission time, the length of total request packet with additional security extension for key exchange and cryptographic hash is still below than 512 bytes. This is an indication for possible implementation of our proposed security support in TFTP where the length of the proposed RRQ packet is in acceptable value and consistent with the general option extension format. According to the result, we believe that the proposed security extension is backward-compatible with the general protocol, also the size of the packet is consistent with existing TFTP's request packet format.

CONCLUSION

As conclusion, in order to achieve the security properties for transferring file on low cost constrained M2M system, a security framework is proposed as an initial finding to merge the DHKE-based key exchange technique with TFTP communication protocol. The proposed work makes noteworthy contributions by presenting the new packet header formats that create support for securing TFTP. It is believed that through cryptographic security implementation in TFTP using new proposed technique, information can be protected from being tampered or eavesdropped by unauthorized third party. Compared to normal TFTP which has no assurances the messages that is being sent will arrive to exact destination, this simple security solution on TFTP will satisfy the security requirements during file transmission in M2M communication technology. In the next stage of this research, real-time experiment will be conducted to implement and formulate the new packet format with additional security mechanism on embedded microcontroller device.

ACKNOWLEDGEMENTS

The researchers would like to thank the Ministry of Higher Education for providing the FRGS grant, 600-RMI/FRGS 5/3 (141/2015), Research Management Institute (RMI) and also Faculty of Electrical Engineering, Universiti Teknologi MARA (UiTM) for financial support of this research.

REFERENCES

Cha, I.C.I., Y. Shah, A.U. Schmidt, A. Leicher and M.V. Meyerstein, 2009. Trust in M2M communication. IEEE. Veh. Technol. Mag., 4: 69-75.

Finlayson, R., 1984. Bootstrap loading using TFTP. Master Thesis, Stanford University, Stanford, California.

Harkin, A. and G. Malkin, 1995. TFTP timeout interval and transfer size options status. Hewlett Packard Co., Alexandria, Virginia. https://tools.ietf.org/html/rfc1784

Harkin, A. and G.S. Malkin, 1998. TFTP block size option. Hewlett Packard Co., Alexandria, Virginia.

Hongsong, C., F. Zhongchuan and Z. Dongyan, 2011. Security and trust research in M2M system. Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES), July 10-12, 2011, IEEE, Beijing, China, ISBN:978-1-4577-0576-2, pp: 286-290.

Hossain, M.M., M. Fotouhi and R. Hasan, 2015. Towards an analysis of security issues, challenges and open problems in the internet of things. Proceedings of the IEEE World Congress on Services (SERVICES), June 27-July 2, 2015, IEEE, New York, USA., ISBN:978-1-4673-7276-3, pp: 21-28.

Iitsuka, T., N. Saze, N. Chiba, N. Kase and Y. Hiro *et al*., 2012. Hitachi cloud computing solutions for enterprise information systems. Hitachi Rev., 61: 53-59.

Isa, M.A.M., N.N. Mohamed, H. Hashim, S.F.S. Adnan and J.A. Manan *et al*., 2012. A lightweight and secure TFTP protocol for smart environment. Proceedings of the IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE), December 3-4, 2012, IEEE, Kota Kinabalu, Malaysia, ISBN: 978-1-4673-3032-9, pp: 302-306.

Kowshalya, A.M. and M.L. Valarmathi, 2016. Towards trustworthy and secure communications in Social Internet of Things (SIoT). Asian J. Inf. Technol., 15: 3957-3964.

Lu, R., X. Li, X, Liang, X. Shen and X. Lin, 2011. GRS: The green, reliability and security of emerging machine to machine communications. IEEE Commun. Magazine, 49: 28-35.

Malkin, G. and A. Harkin, 1998. TFTP option extension (RFC 2347). Internet Soc., 1: 1-7.

Masotta, P., 2015. TFTP window size option. Transfer, 32: 64-86.

Mohamed, N.N., H. Hashim, Y.M. Yussoff and A.M. Isa, 2013. Securing TFTP packet: A preliminary study. Proceedings of the IEEE 4th Conference on Control and System Graduate Research Colloquium (ICSGRC), August 19-20, 2013, IEEE, Shah Alam, Malaysia, ISBN:978-1-4799-0551-5, pp: 158-161.

Mohamed, N.N., H. Hashim, Y.M. Yussoff, M.A.M. Isa and S.F.S. Adnan, 2014. Compression and encryption technique on securing TFTP packet. Proceedings of the IEEE Symposium on Computer Applications and Industrial Electronics, April 7-8, 2014, Penang, pp: 198-202.

Mohamed, N.N., Y.M. Yussoff, M. Anuar, M. Isa and H. Hashim, 2016. A pre-shared diffie-hellman key exchange scheme for a secure TFTP protocol. Proceedings of the 3rd International Conference on Science and Social Research, December 6-7, 2016, Universiti Teknologi MARA, Shah Alam, Malaysia, pp: 1-13.

Mohamed, N.N., Y.M. Yussoff, M.A.M. Isa and H. Hashim, 2017. Symmetric encryption using pre-shared public parameters for a secure TFTP protocol. J. Eng. Sci. Technol., 12: 098-112.

Nagamani, T.S. and G.K. Veni, 2016. A comparative analysis on cloud security issues. Intl. J. Adv. Trends Comput. Sci. Eng., 5: 1-5.

Persson, C., G. Picard, F. Ramparany and O. Boissier, 2011. A multi-agent based governance of machine-to-machine systems. Proceedings of the European Workshop on Multi-Agent Systems, November 14-15, 2011, Springer, Berlin, Germany, pp: 205-220.

Qiu, S.B., B. Yuan and K.L. Zhang, 2008. Building TFTP server on embedded system. Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing WiCOM'08, October 12-14, 2008, IEEE, Dalian, China, ISBN:978-1-4244-2107-7, pp: 1-4.

Wang, Q., C. Lv, Y. Shen and J.M. Chen, 2015. Compressed sensing and mobile agent based sparse data collection in wireless sensor networks. Proceedings of the IEEE International Conference on Instrumentation and Measurement Technology (I2MTC), May 11-14, 2015, IEEE, Pisa, Italy, ISBN:978-1-4799-6115-3, pp: 1789-1794.