Appl. Sci. **2015**, *5*, 446-458; doi:10.3390/app5030446

OPEN ACCESS

applied sciences

ISSN 2076-3417

www.mdpi.com/journal/applsci

Article

Research of Integrity and Authentication in OPC UA Communication Using Whirlpool Hash Function

Kehe Wu, Yi Li *, Long Chen and Zhuxiao Wang

North China Electric Power University, No.2 Beinong Road, Changping District, Beijing 102206, China; E-Mails: epuwkh@126.com (K.W.); easy_cl@163.com (L.C.); wangzx@ncepu.edu.cn (Z.W.)

* Author to whom correspondence should be addressed; E-Mail: liyi174748@163.com; Tel./Fax: +86-10-6177-2747.

Academic Editor: Christos Verikoukis

Received: 21 June 2015 / Accepted: 17 August 2015 / Published: 21 August 2015

Abstract: Currently, the demand for information security of industrial control systems is becoming more and more urgent, but the security model proposed by OPC UA cannot meet the practical requirements of industrial control systems. For this reason, this paper proposes a new security communication model to provide integrity and authentication in OPC UA. This model uses the Whirlpool hash function to check integrity and generates digital signature along with RSA in message transmission. Compared to SHA-1, Whirlpool has a higher calculation speed and lower collision rate. Through this model, terminals in the upper layer can communicate with field devices via a channel with high security and efficiency.

Keywords: industrial control system; information security; OPC UA; Whirlpool hash function

1. Introduction

In recent decades, computers have been widely adopted in industrial control systems, which lead to an extremely increment of industrial automation. Especially after the proposal of OPC (OLE for Process Control) specification, which builds a bridge between Windows applications and field process control applications [1,2], it is possible for industrial control networks to communicate with external networks directly, even they can connect to the Internet. However, the OPC specification lacks essential security mechanisms, thus it on one hand makes it convenient for people managing or acquiring data from field devices, on the other hand the security threats from internal and external industrial control networks become much more serious, such as information exposure and control command tampering caused by

trojans, viruses or other network attacks [3,4]. For this reason, OPC Foundation proposed OPC UA (OPC Unified Architecture) specification in August, 2006 which expatiates the security model of OPC UA. This security model mainly guarantees confidentiality, integrity and authentication of transmission in industrial control networks [5–9]. There are three security profiles included: basic128rsa15, basic256 and none [10–14]. Because both basic128rsa15 and basic256 use AES to encrypt the content of communication in the network, the time of encryption increases along with the increment of message length. However, industrial control systems have high requirement in aspect of real-time. For process control loops, the decrease of real-time will cause delay of control. We take power system as example. The transmission delay time requirement for power system relay protection signals is less than 5 ms, thereby power system is very sensitive to time delay [15,16]. When the transmission delay time of the control instruction is longer than the specified value, it may result in frequency instability by executing insufficient or excessive load curtailments, and even may cause power failure [17–20]. For the two security profiles adopted by OPC UA security model, basic128rsa15 and basic256, according to our experiment shown in Chapter 4, the delay time caused by them are all longer than 5 ms when the data volume transmitted is bigger than 8192 bytes, such a delay time is unacceptable to network communication; therefore, these two security profiles are not efficient enough to be used in practical application [12,14]. Reference [14] advises the inclusion of a compromising scheme that authenticates data only into OPC UA specification to protect network communication and have a lower overhead at the same time. Both basic128rsa15 and basic256 adopt SHA-1 to calculate message digest for messages to be transferred. However, security problems lie in this hash function. According to Reference [21], the speed of deciphering is about 2000 times faster than the prospective speed. Moreover, because the computer is becoming more powerful and cheaper, the security of SHA-1 is reducing day by day. Thereby, the security of SHA-1 is questioned by cipher analysts and vendors. Microsoft and Google have announced the plan to abandon SHA-1 and Microsoft will not trust signatures using SHA-1 from January 2016. NIST (National Institute of Standards and Technology) had announced the step-by-step elimination of this hash function.

Aims at solving the problem of low communication efficiency and security occurred in OPC UA, this paper proposes a security communication model of OPC UA. This model checks the integrity of communication in networks using the Whirlpool hash function and authenticates the source of communication using X.509 specification. Compared to SHA-1 which is currently used in OPC UA, Whirlpool has a higher calculation speed and lower collision rate. The experiment shows that this model can guarantee integrity and authentication for communication in industrial control networks effectively, and brings a low time delay to network communication, thus can basically meet the requirement of security and real-time in industrial control networks.

2. Overview of Whirlpool Hash Function

2.1. Principle of Whirlpool

Whirlpool was a hash function proposed by Vincent Rijmen and Paulo S. L. M. Barreto in 2000. It is one of only two hash functions endorsed by NESSIE (New European Schemes for Signature, Integrity, and Encryption), and has been adopted by ISO (International Organization for Standardization) and IEC

(International Electrotechnical Commission) as ISO/IEC 10118-3 specification [22]. In contrast to traditional hash functions, Whirlpool makes use of a block cipher structure similar to AES, its main calculation procedure is as follows [23,24].

(1) Append padding bits. The message is padded so that its length is congruent to 256 modulo 512, that is to say, the length of padded message is some times of 512 minus 256, and the reserved 256 bits will be used in step 2. The pattern of padding is the same, it consists of a single 1-bit followed by the necessary of 0-bits.

(2) Append length. The padded message is appended by a block of 256 bits representing the length of message before the padding. After that the message m can be divided into t 512-bit blocks $m_1, \cdots, m_t$, each block can be treated as an 8×8 byte array called hash state.

(3) Substitute bytes. This is a non-linear byte substitution, each byte of a state is substituted individually according to a table called S-box.

(4) Shift columns. The shift column transformation is a simple cyclical permutation performed against a state column by column except column 0. For column i, it is shifted downwards by $i - 1$ bytes.

(5) Mix rows. In this step, each row of a state array is treated as a polynomial $a(x)$ over Galois field $GF(2^8)$, $a(x)$ is multiplied by a fixed polynomial $c(x)$ modulo $x^8 + x^4 + x^3 + x^2 + 1$. The polynomial $c(x)$ is expressed in the following formula in the designing of Whirlpool (the factors are expressed in hexadecimal numbers).

$$c(x) = '09'x^7 + '02'x^6 + '05'x^5 + '08'x^4 + '01'x^3 + '04'x^2 + '01'x + '01' \quad (1)$$

(6) Add round key. This step is a simple bitwise XOR calculation between the state and round key. The round key is derived from a seed key and the length of round key is equal to the block size. The key schedule is same as the round function.

There should be iteration for 10 times to obtain the hash value. The procedure of Whirlpool is illustrated in Figure 1.

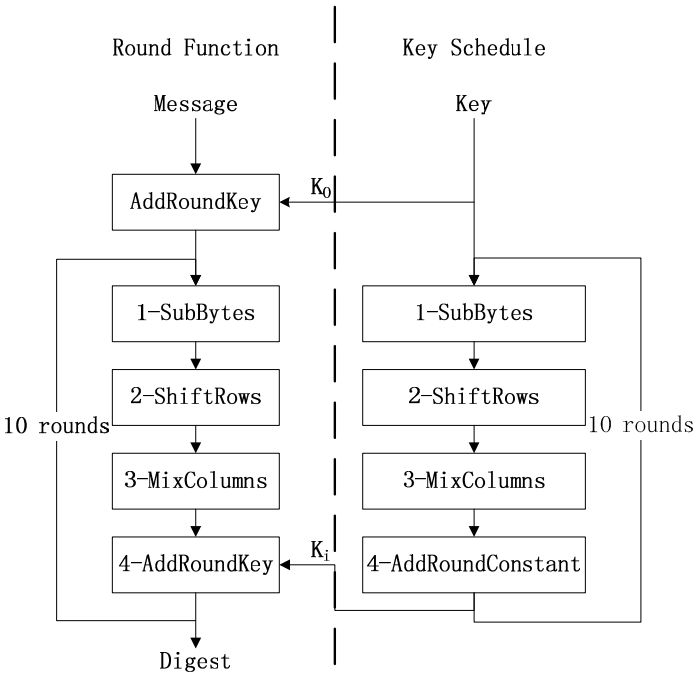


Figure 1. Procedure of Whirlpool.

2.2. Features of Whirlpool

2.2.1. Adoption of AES-Like Structure to Construct Compression Function

Whirlpool uses a block cipher that is specially designed for using in hash function, but this block cipher cannot be used as a standalone encryption function. The motion is that the designers wanted to take advantage of security and efficiency of a block cipher like AES while at the same time provide a potential security equal to SHA-512. The structure and elementary functions this block cipher uses are similar to AES, but the block size and key size it uses is 512 bits [22].

2.2.2. Using Miyaguchi-Preneel Scheme to Construct Hash Function

Traditional hash functions, for example MD5 and SHA-1, usually make use of the Davies-Meyer scheme to construct hash function, whereas Whirlpool adopts Miyaguchi-Preneel scheme to construct hash function. Miyaguchi-Preneel scheme is one of the few still unbroken methods to construct a hash function from an underlying block cipher [23]. In this scheme, the message block and key is equal in status, that is to say, the message block and key can be exchanged, which makes Whirlpool more flexible than hash functions constructed from Davies-Meyer.

2.2.3. Convenient for Hardware Implementation

As Whirlpool uses a similar block cipher structure with AES, it has similar performance and memory characteristics with AES. It occupies less memory during its implementation and has a better efficiency in execution, thus it is convenient for software and hardware implementation [24–26].

2.3. Comparison between Whirlpool and SHA-1

2.3.1. Computation Speed

To compare the calculation speed between Whirlpool and SHA-1, we generate random strings in different lengths (10,000 strings for each length). Let these two algorithms calculate the hash value of these strings respectively, then compare the average time of them during one time of calculation procedure. The test is performed under Inter Pentium Dual E2180 2.00GHz and 2GB memory environment. Figure 2 shows the comparison of these two algorithms.

From this figure we can learn that when the length of string is short, these two algorithms have a similar calculated performance, even SHA-1 is a bit better than Whirlpool. However, when the string is long (longer than 256 bytes), the calculation speed of Whirlpool is faster than SHA-1, and with the increment of length, the gap between them in calculation speed is bigger and bigger.

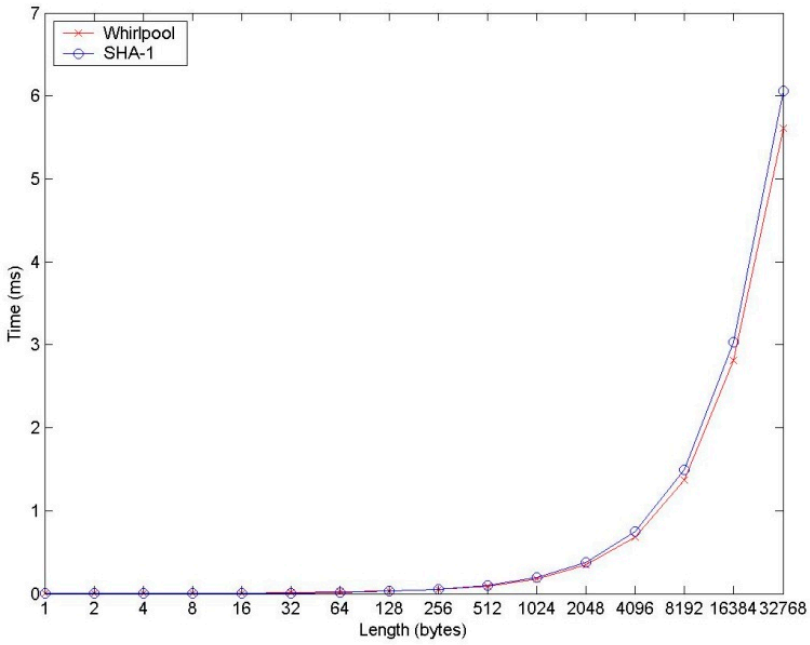


Figure 2. Comparison between Whirlpool and SHA-1.

2.3.2. Security

According to the birthday attack theory, assuming the number of possible outputs a hashing function H have is 2^m, *i.e.*, the length of output is m bits. If the probability of at least two from k random input of H have the same output is larger than 0.5, as a result, $k \approx \sqrt{2^m} = 2^{m/2}$. As the output length of Whirlpool is 512 bits, whereas it is 160 bits for SHA-1, $2^{512/2} \gg 2^{160/2}$, it is obvious that Whirlpool is much more secure than SHA-1 against birthday attack.

3. Security Communication Model of OPC UA Based on Whirlpool

3.1. Architecture of Model

The target of this model is to build a security communication channel between client and server in OPC UA. This channel is always active during the process of communication, and it guarantees the integrity and authentication of all the messages exchanged. This means that the client and server need to authenticate each other only once. In this model, the client sends and receives requests and responses of corresponding services using its APIs. The server mainly provides two types of services for the client, one is to accept the connection requests and notification subscription requests from the client, the other is to publish the occurrence of some events to the client, such as alarms, changes of data values, events and execution results of programs. The CA is mainly responsible for building, issuing and managing certificates for clients and servers. There may be several clients and servers included in this model. Each client may be connected to one or more servers, and each server may be connected to one or more clients. An application may include two parts of modules performing functions of client and server respectively, hereby it can meet the requirement of connecting to other clients and servers. The architecture is illustrated in Figure 3.

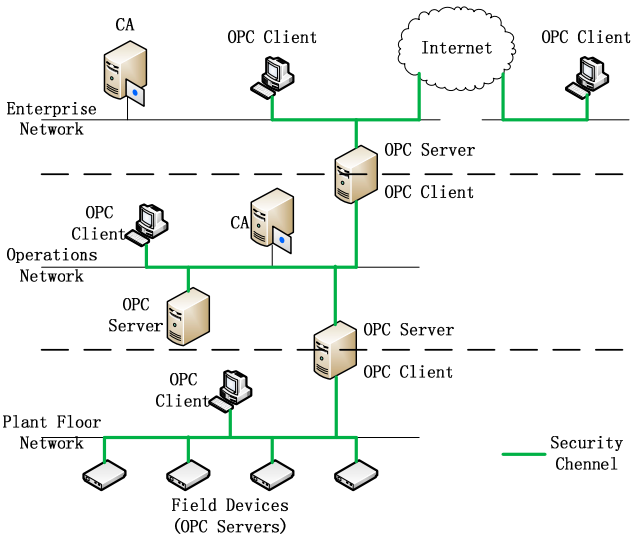


Figure 3. Architecture of security communication model.

3.2. Procedure of Security Channel Establishment

In the procedure of building security communication channel between client and server, both of the ends will authenticate each other through checking whether the digital signature of CA is right. This model uses Whirlpool to obtain a message digest, then uses the private key of RSA in certificate to encrypt the message digest to generate digital signature. The procedure of building a security communication channel between client and server is illustrated in Figure 4.

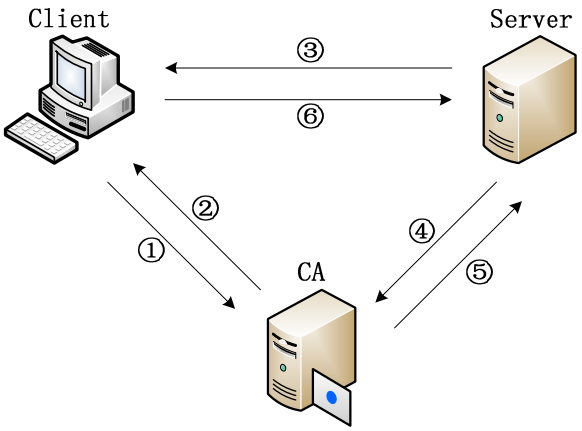


Figure 4. Procedure of security channel establishment.

(1) Before building a security channel, the client requests CA to authenticate whether the server is legal. The contents of authentication include CA signature, issue and expiry date, and CA revocation list.

(2) The CA returns the result of authentication to the client.

(3) The client requests the server to build security connection. The client provides its certificate and a nonce to the server. The data to be sent are signed with the private key of the client, and then encrypted by the public key of the server.

(4) After the connection request is received by the server, the server requests CA to authenticate the client. The contents of authenticate also include CA signature, issue and expiry date, and CA revocation list.

(5) The CA returns the result of authentication to the server.

(6) After the client is authenticated, the server responses the connection request from the client. The server sends a nonce, security token, and the lifetime of the token to the client. The contents to be sent is encrypted by the private key of the server, and then encrypted by the public key of the client. After that the client and the server can communicate with each other through the security channel. In the subsequent communication, the receiver of the message calculates the digest of the received message using Whirlpool, then compares it with the digest in the signature to check whether the message is intact or not.

4. Experiment and Analysis

To evaluate the performance of the secure communication model we proposed in Chapter 3, we construct an experimental environment. As depicted in Figure 4, this experimental environment contains a server, a client and a CA. We adopt opcsvrda2 to build the server. Opcsvrda2 is a software development kit based on Win32 platform, it can meet the demand of OPC DA 1.0/2.0/3.0 specification and can develop an OPC server fast. Besides, we develop an OPC client using Visual Studio 2008. In this client we can configure the data volume transmitted and the security profile we want to use. There are five security profiles we can use in total, they are: none, WhirlpoolRSA, SHA1RSA, basic128rsa15 and basic256. The program interface of client configuration is shown in Figure 5. Finally, we construct a CA using OpenSSL to issue certifications to the client and server.

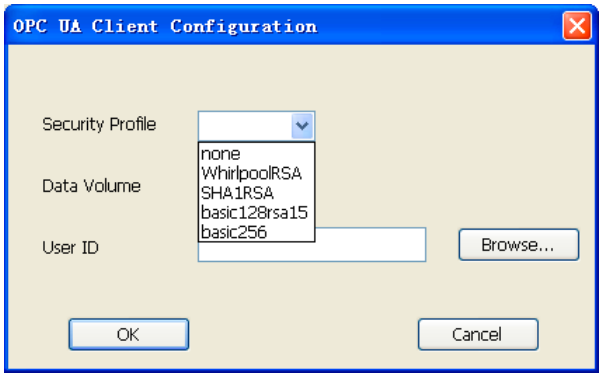


Figure 5. Program interface of client configuration.

In our experiment, the client constructs a secure connection with the server and read different lengths of data from the server. When a client requests the information for reading or monitoring, the server always returns a StatusCode which contains a ServerTimestamp and a SourceTimestamp. From these timestamps we can calculate the delay time brought by our model, and in turn we can obtain the time passed during data transmission after taking security measures and evaluate the effect these security profiles bring to data transmission. In this experiment, we select four kinds of length to perform the experiment: 1024 bytes, 4096 bytes, 8192 bytes and 16,384 bytes. The experiment is done 12,000 times for each security profile and each length of message. Figure 6 shows the delay time of the five security

profiles when transiting messages in four kinds of lengths. In this set of figures the x-axis presents the chronological order of messages in a measurement and the y-axis represents delay time in milliseconds. In the legend of Figure 6, none means there is no secure measure taken; WhirlpoolRSA represents security profile using Whirlpool and RSA; SHA1RSA means security profile using SHA-1 and RSA; basic128rsa15 means security profile using AES-128, SHA-1 and RSA; and basic256 is security profile utilizing AES-256, SHA-1 and RSA.

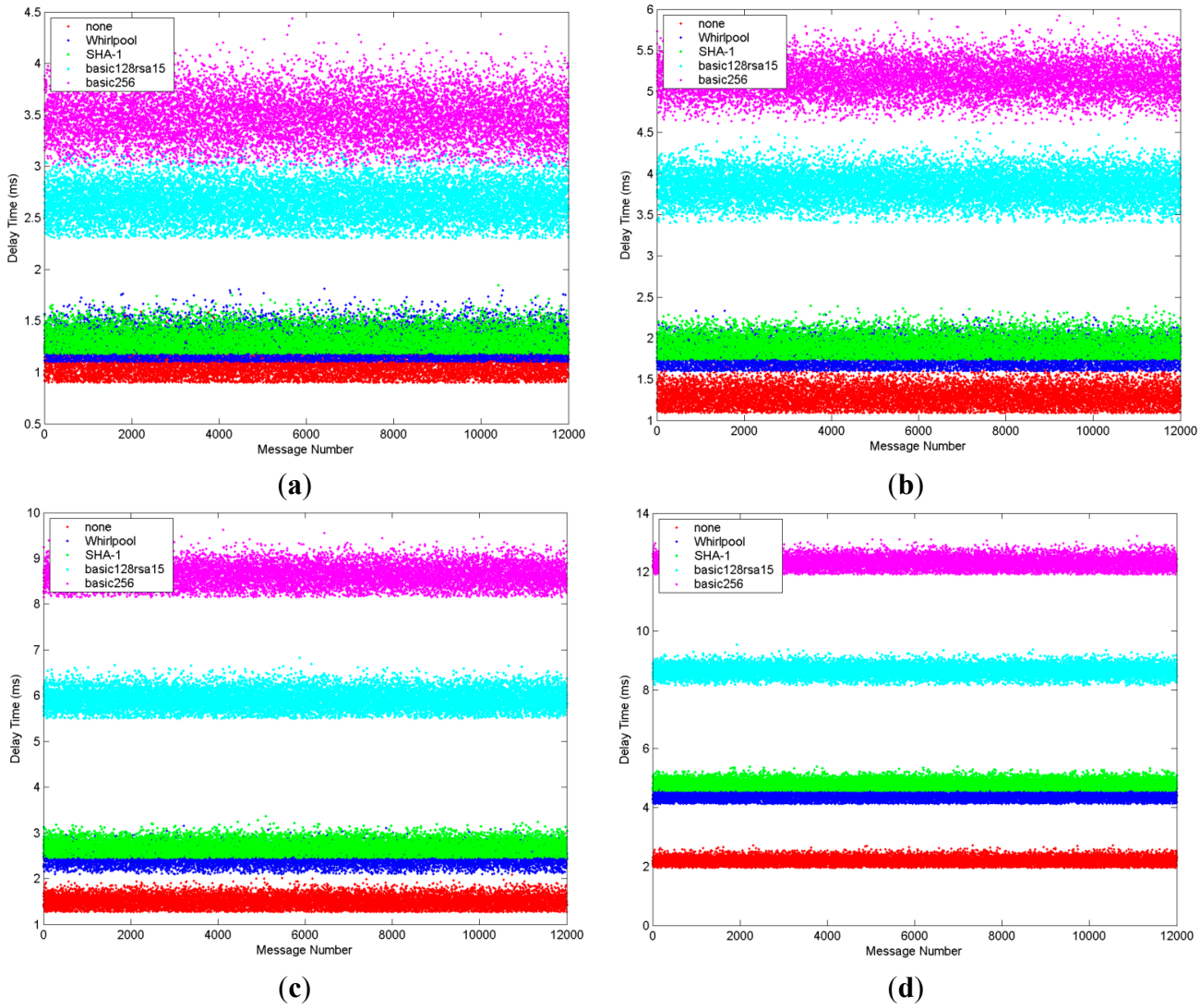


Figure 6. Comparison of delay time in different lengths of messages. (**a**) In the length of 1024 bytes; (**b**) In the length of 4096 bytes; (**c**) In the length of 8192 bytes; (**d**) In the length of 16,384 bytes.

From this set of figures we can see that the delay time of basic128rsa15 and basic256 are much longer than WhirlpoolRSA and SHA1RSA. This means that WhirlpoolRSA and SHA1RSA perform much better than the security profiles adopted in OPC UA security model in term of efficiency. Besides, the minimum delay time of the security profile using SHA-1 and RSA is longer than that using Whirlpool and RSA for integrity and authentication. It should be noted that as the network situation is occasionally

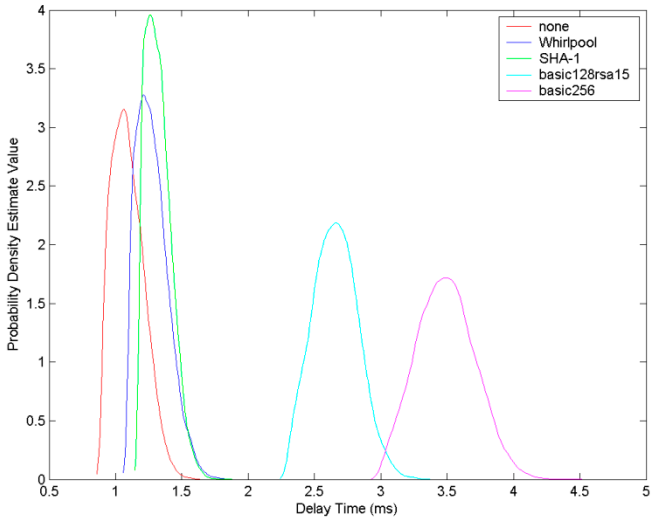
not very stable, the delay time of transmission in the same security profile is slightly different. Table 1 shows the average delay time of each security profile in different length of data.

Table 1. Experiment results.

Profile	1024 Bytes	4096 Bytes	8192 Bytes	16384 Bytes
none	1.052 ms	1.265 ms	1.684 ms	2.179 ms
WhirlpoolRSA	1.215 ms	1.736 ms	2.485 ms	4.387 ms
SHA1RSA	1.268 ms	1.859 ms	2.659 ms	4.753 ms
basic128rsa15	2.648 ms	3.839 ms	5.914 ms	8.647 ms
basic256	3.473 ms	5.164 ms	8.627 ms	12.304 ms

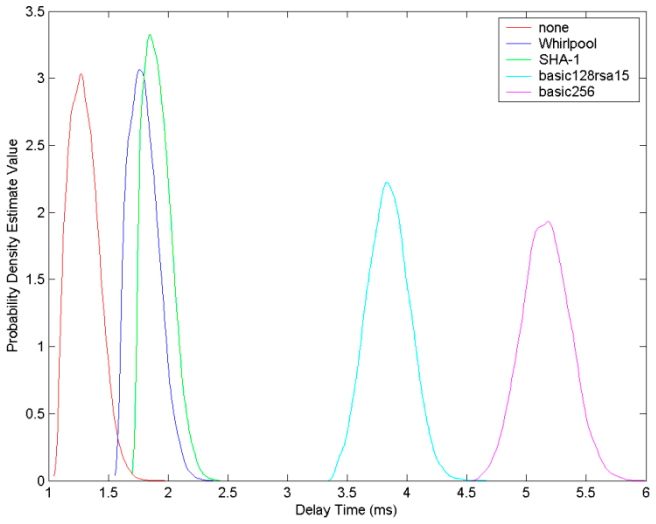
From this table we can also find that basic128rsa15 and basic256 cost a much longer delay time than WhirlpoolRSA and SHA1RSA, and when the data volume transmitted is 8192 bytes, the delay time caused by using basic128rsa15 and basic256 are all longer than 5 ms, such a delay is not acceptable for some industrial control systems like power systems. By contrast, WhirlpoolRSA and SHA1RSA cost less than this specified value even the data volume transmitted is 16,384 bytes, thus these two profiles can be adopted in such systems. In addition, the security profile using Whirlpool and RSA has a smaller average delay time compared to the security profile using SHA-1 and RSA: the difference of delay time between these two profiles is approximately 4%–6%.

To help analyze the experimental results better, we draw a probability density estimate figure shown in Figure 7. This set of figures mainly depict distribution situation of delay time for different security profiles in different lengths of data. In each of these figures, the x-axis presents the delay time in milliseconds and the y-axis represents probability density estimate value. From this set of figures, we can see that for the same probability density estimate value, the delay time value of security profile using Whirlpool and RSA is smaller than the one using SHA-1 and RSA. From this we can also conclude that the security profile using Whirlpool and RSA performs better than the security profile using SHA-1 and RSA in delay time.

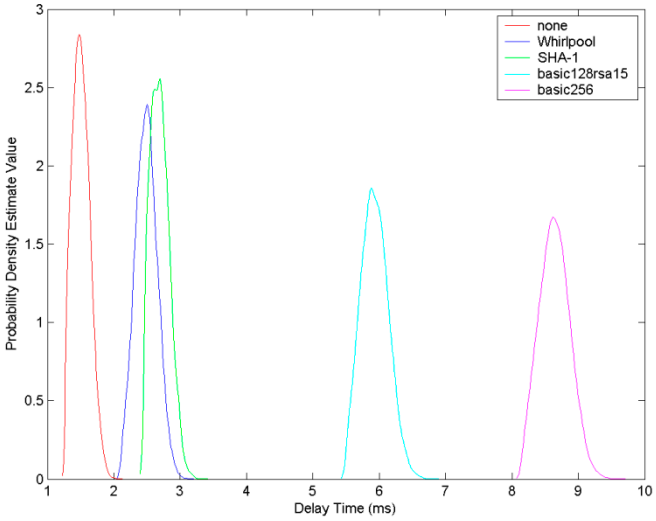


(a)

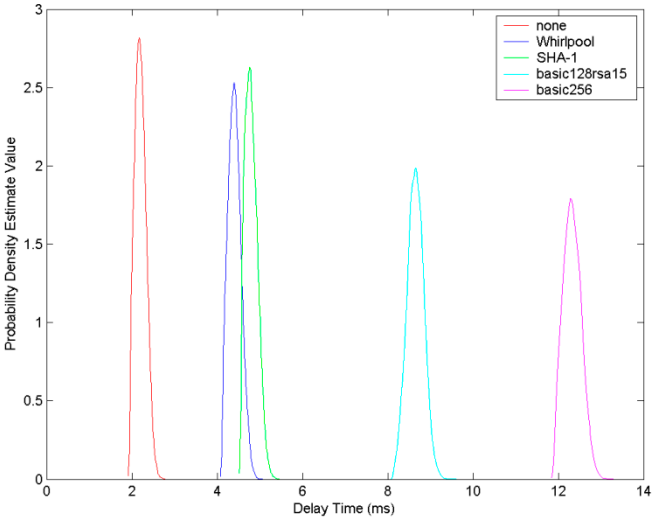
Figure 7. *Cont.*



(**b**)



(**c**)



(**d**)

Figure 7. Probability distribution comparison of delay time in different lengths of messages. (**a**) In the length of 1024 bytes; (**b**) In the length of 4096 bytes; (**c**) In the length of 8192 bytes; (**d**) In the length of 16384 bytes.

5. Conclusions

This paper presents a new security communication model using Whirlpool instead of SHA-1 to protect communication of PCs and field devices in industrial control networks. Through establishing secure channels between the ends of communication, this model can provide integrity and authentication with a higher level of security and considerable efficiency, basically meeting the requirement of security and real-time modeling in industrial control networks. In the future this security profile can be included into OPC UA specification. With the increment of terminal calculation ability in the future, symmetric encryption algorithms with a higher efficiency can be integrated to this security profile to implement confidentiality, integrity and authentication of messages completely transmitted in networks.

Acknowledgments

The authors thank the editor and referees for their helpful comments. This research was supported by the National Natural Science Foundation of China (No. 61300132) and the Specialized Research Fund for the Doctoral Program of Higher Education (No. 2012003612003).

Author Contributions

All authors have contributed to the manuscript. Kehe Wu conceived the underlying idea of the paper. Yi Li performed the numerical calculations and designed the communication model. Long Chen and Zhuxiao Wang performed the experiment. Yi Li and Long Chen wrote the paper. All authors analyzed and discussed the experiment results.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Fernbach, A.; Granzer, W.; Kastner, W. Interoperability at the management level of building automation systems: A case study for BACnet and OPC UA. In Proceedings of the 2011 IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA), Toulouse, France, 5–9 September 2011; pp. 1–8.
2. Huang, R.; Liu, F. Research on OPC UA based on electronic device description. In Proceedings of the 3rd IEEE Conference on Industrial Electronics and Applications (ICIEA 2008), Singapore, 3–5 June 2008; pp. 2162–2166.
3. Van Tan, V.; Yoo, D.S.; Yi, M.J.; Security in automation and control systems based on OPC techniques. In Proceedings of the International Forum on Strategic Technology (IFOST 2007), Ulaanbaatar, Mongolia, 3–6 October 2007; pp. 136–140.
4. Gaitan, A.M.; Popa, V.; Gaitan, V.G.; Petrariu, A.I.; Ungurean, I. Products Authentication and Traceability using RFID Technology and OPC UA Servers. *Elektron. Elektrotech.* **2012**, *18*, 73–76.
5. OPC Foundation. *OPC Unified Architecture Specification, Release 1.00*; OPC Foundation: Scottsdale, AZ, USA, 2006.

6. Chou, J.; Chen, L.; Zhang, Y.J.; Pan, L.H. OPC Unified Architecture for Industrial Demand Response. *Int. J. Secur. Appl.* **2012**, *6*, 275–280.
7. Renjie, H.; Feng, L.; Dongbo, P. Research on OPC UA security. In Proceedings of the 2010 the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), Taichung, Taiwan, 15–17 June 2010; pp. 1439–1444.
8. Fernbach, A.; Kastner, W. Certificate management in OPC UA applications: An evaluation of different trust models. In Proceedings of the 2012 IEEE 17th Conference on Emerging Technologies & Factory Automation (ETFA), Krakow, Poland, 17–21 September 2012; pp. 1–6.
9. Schwarz, M.H.; Borcsok, J. A survey on OPC and OPC-UA: About the standard, developments and investigations. In Proceedings of the 2013 XXIV International Symposium on Information, Communication and Automation Technologies (ICAT), Sarajevo, Bosnia and Herzegovina, 30 October–1 November 2013; pp. 1–6.
10. Cavalieri, S. Evaluating Overheads Introduced by OPC UA Specifications. In *Human–Computer Systems Interaction: Backgrounds and Applications 2*; Zdzisław, S.H., Juliusz, L.K., Teresa, M., Eds.; Springer Berlin Heidelberg: Heidelberg, Germany, 2012; pp. 201–221.
11. Braune, A.; Hennig, S.; Hegler, S. Evaluation of OPC UA secure communication in web browser applications. In Proceedings of the 6th IEEE International Conference on Industrial Informatics (INDIN 2008), Daejeon, Korea, 13–16 July 2008; pp. 1660–1665.
12. Cavalieri, S.; Cutuli, G.; Monteleone, S. Evaluating impact of security on OPC UA performance. In Proceedings of the 2010 3rd Conference on Human System Interactions (HSI), Rzeszow, Poland, 13–15 May 2010; pp. 687–694.
13. Post, O.; Seppälä, J.; Koivisto, H. Certificate based security at device level of automation system. In Proceedings of the Preprints of 4th IFAC Wokshop on Discrete-Event System Design, Gandia Beach, Spain, 6–8 October 2009; pp. 120–124.
14. Post, O.; Seppälä, J.; Koivisto, H. The Performance of OPC-UA Security Model at Field Device Level. In Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics, Milan, Italy, 2–5 July 2009; INSTICC Press: Milan, Italy, 2009; pp. 337–341.
15. Chen, G.Y.; Yin, X.; Zhang, K. Communication modeling for wide-area relay protection based on IEC 61850. *Telkomnika* **2012**, *10*, 1673–1684.
16. Feng, L.H.; Gui, W.H.; Yang, F. Application of Communication Optimization Strategy Based on Cascade PLC MODBUS in Fire Water System of Hydropower Station. In Proceedings of the Second International Conference on Intelligent Computation Technology and Automation, Changsha, China, 10–11 October 2009; IEEE: Changsha, China, 2009; pp. 45–48.
17. Silva, R.M.; Martins, H.; Nascimento, I.; Baptista, J.M.; Ribeiro, A.L.; Santos, J.L.; Jorge, P.; Frazão, O. Optical Current Sensors for High Power Systems: A Review. *Appl. Sci.* **2012**, *2*, 602–628.
18. Rosyadi, M.; Muyeen, S.M.; Takahashi, R.; Tamura, J. A Design Fuzzy Logic Controller for a Permanent Magnet Wind Generator to Enhance the Dynamic Stability of Wind Farms. *Appl. Sci.* **2012**, *2*, 780–800.
19. Sjolte, J.; Tjensvoll, G.; Molinas, M. Power Collection from Wave Energy Farms. *Appl. Sci.* **2013**, *3*, 420–436.

20. Li, Y.; Zhang, B.; G, Z.; Bo, Z. Influences of the time delay on the control effect of under-frequency load shedding in power systems. In Proceedings of the 2015 27th Chinese Control and Decision Conference (CCDC), Qingdao, China, 23–25 May 2015; IEEE: Qingdao, China, 2015; pp. 5182–5186.
21. Wang, X.; Yin, Y.L.; Yu, H. Finding collisions in the full SHA-1. In Proceedings of the 25th Annual International Cryptology Conference (CRYPTO 2005), Santa Barbara, CA, USA, 14–18 August 2005; Springer Verlag: Santa Barbara, CA, USA, 2005; pp. 17–36.
22. Stallings, W. The Whirlpool secure hash function. *Cryptologia* **2006**, *30*, 55–67.
23. Barreto, P.; Rijmen, V. *The Whirlpool Hashing Function*; First open NESSIE Workshop: Leuven, Belgium, 2000; pp. 1–20.
24. Hartikainen, A.; Toivanen, T.; Kiljunen, H. *Whirlpool Hashing Function*; Lappeenranta University of Technology: Lappeenranta, Finland, 2006; Volume 26, pp. 1–19. Available online: http://www2.it.lut.fi/kurssit/05-06/Ti5318800/assign/Whirlpool.pdf (accessed on 8 September 2014).
25. Alho, T.; Hämäläinen, P.; Hännikäinen, M.; Hamalainen, T.D. Compact hardware design of Whirlpool hashing core. In Proceedings of the conference on Design, Automation and Test in Europe, Nice Acropolis, France, 16–20 April 2007; Institute of Electrical and Electronics Engineers Inc.: Nice Acropolis, France, 2007; pp. 1247–1252.
26. Krawczyk, K.; Tomaszewicz, P.; Rawski, M. SoPC Implementation of Whirlpool Hash Function. In Proceedings of the 2011 21st International Conference on Systems Engineering (ICSEng), Las Vegas, NV, USA, 16–18 August 2011; IEEE Computer Society: Las Vegas, NV, USA, 2011; pp. 461–462.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).