Verification of an Authentication Protocol for M2M Communication Signal Processing

Woo-Sik BAE, Kun-Hee Han

Abstrat

With the advancement of RFID/USN system technology, M2M (Machine-to-Machine) communication draws attention as a promising future industry. M2M communication is intended for inter-machine communication especially in a space hardly accessible to humans in such fields as disaster, safety, construction, health and welfare, weather, environment, logistics, national defense, medical care and agro-livestock industries where such systems are installed and operated long hours. In effect, machines replace men in M2M communication to the extent that programs automatically run in accordance with varying situations, e.g. giving failure alarms and taking proper actions for electronic devices and vehicles, with a view to efficiency in information control and device operation. Still, wireless M2M communication between devices could be exposed to attackers causing security vulnerabilities. Therefore, proper security measures including cross-authentication need taking to ensure relevant devices are legitimate.

This paper, quite a few studies have investigated many protocols in relation to M2M security issues. Likewise, the present study attempted to solve M2M security issues by means of public keys, hash functions and XOR and suggested a secure protocol based on M2M authentication. The proposed protocol proved secure in a formal verification using Casper/FDR instead of any theorem proving method, which was used in most of previous studies.

Keywords: M2M Security, Privacy, Security Protocol, Formal Validation

1. Introduction

M2M communication for intelligent and automatic inter-device operation has been extensively studied. As M2M (Machine-to-Machine) communication requires no human intervention, it is applied to dangerous fields that are hard to check and inspect via direct human access or to simple repeat works and military purposes [1]. As a rule, M2M communication refers to inter-device connection and communication using wired or wireless networks and IT. In particular, M2M communication system transmits and exchanges information related to failure diagnosis, repairing, monitoring and information collection in diverse industries [2-5]. Nonetheless, wireless M2M communication may be prone to attackers’ eavesdropping/random alteration/deletion of data in transmission, privacy issues and other security vulnerabilities. Hence, many researchers study on communication protocols for M2M networks [6]. However, most of them use and propose mathematical logical expressions to prove their systems and characteristics required on such systems, which is called the theorem proving, where vulnerabilities arise that are unexpected in the design process and considerable time and follow-up experiments are needed before such systems are applied in reality.

This paper study used a formal method to remedy the issues associated with the existing protocols. To be specific, this study verified the proposed protocol by conducting a formal specification with mathematical and logical expressions and a formal verification to prove whether the specified content met security requirements. Here, Casper [7][8] and FDR[9], both of which are recognized for efficiencies as methods to test models, were used to verify the proposed security protocol. This paper is comprised of the following chapters. Chapter 2 deals with previous studies on M2M communication. Chapter 3 specifies and tests the proposed protocol to verify its security. Finally, chapter 4 presents the conclusion.
2. Relevant studies

2.1 M2M communication framework

In M2M(Machine-to-Machine) communication technology, wire or wireless communication modules are mounted on diverse machines and devices so as to extend communication, broadcasting and internet infra from man-to-man to man-to-machine and even machine-to-machine domains. With the remarkable advancement of RFID/USN technology, M2M technology broadens its applicability based on interactive machine-to-machine communication for collection, processing and transmission of information with no human intervention [10].

![M2M system framework](image)

[Fig. 1] represents the system framework for M2M service. M2M Service Capability uses core network features via interface combinations accessible through a range of approaches from outside and can be accessed through multiple core networks and interfaces. M2M applications include Device Application, Gateway Application and Network Application.

M2M communication is divided into device gateway domain and M2M network domain. An M2M communication reference model should provide an open interface for communication between components [11].

2.2 Casper/FDR

As a compiler developed to facilitate protocol specification in CSP (Communication Sequential Process), Casper (Compiler for the Analysis of Security Protocols)[7][8] is a very complex specification language for a security protocol designer unfamiliar with formal design methods in the formal specification process using the conventional CSP[12] language, which is why even a tiny mistake leads to errors, causing difficulties in design and analysis. Casper is a program developed to help design the actions of security protocols with ease.

The specification includes defining the agents, variables, and functions in the protocol; representing each agent as a process; showing all the messages exchanged between the agents; specifying the security properties to be checked; defining the real variables in the actual system to be checked; defining all the functions used in the protocol; listing the agents participating in the actual system with their parameters instantiated; and specifying the intruder’s knowledge and capabilities. Then, once run, the program automatically performs a conversion to CSP documents. Subsequently, FDR (Failure Divergence Refinements) program is used to verify if security and authentication attributes are met. FDR takes care of safety verification, deadlock verification and live-lock verification and shows feasible attack scenarios when security vulnerabilities are found to facilitate the analysis of vulnerabilities.
2.3 Universal Hash function

A hash function is a method to generate short output values from any length of data. Here, results, or hash values, are generated by cutting and repositioning data. There are many different ways used to deter any inference of hash values. Universal Hash function is one of those methods and generated as below. It is assumed that when M is a decimal fraction there is a key value k of (R+1). When the value of the number of digits for the key value has a number between 0 and (m-1), the hash function is represented as the following formula.

\[ h_a(k) \sum_{i=0}^{r} a_i k_i \mod m \]

To prove the above formula, two discrete key values, x and y, whose number of digits is \((r+1)\) are assumed as below:

\[
\begin{align*}
X &= \{x_1, x_2, x_3, \ldots, x_r\} \\
Y &= \{y_1, y_2, y_3, \ldots, y_r\}
\end{align*}
\]

When the key values, x and y are subject to a hash-function collision, it is represented in the following order:

\[
\begin{align*}
\sum_{i=0}^{r} a_i x_i &\equiv \sum_{i=0}^{r} a_i x_i \pmod m \\
\sum_{i=0}^{r} a_i (x_i - y_i) &\equiv 0 \pmod m \\
a_0 (x_0 - y_0) + \sum_{i=1}^{r} a_i (x_i - y_i) &\equiv 0 \pmod m \\
a_0 &\equiv \left( - \sum_{i=1}^{r} a_i (x_i - y_i) \cdot (x_0 - y_0)^{-1} \right) \pmod m
\end{align*}
\]

That is, when an a0 value is selected under the condition that the key values X and Y collide, the maximum number of r branches in a1, … ar will be \((m^r)\) [13].

3. The proposed protocol

M2M devices use several communication methods to exchange information between gateway and network domains. These communication sections have diverse security threats. To provide safe and secure communication environment against security threats from attackers, the present study used mathematical functions based on key authentication and hash functions for the design and test. Data transmitted between sessions are encrypted with complex formulae per session, leading data to vary in transmission. In addition, the time stamp and XOR applied prevents attackers from using any information eavesdropped for other purposes or for replay attacks. In short, the proposed protocol is safe and secure against spoofing attacks, replay attacks, location tracking, eaves-dropping and traffic analysis. The signs used in the proposed protocol are defined as in [Table 1].
Table 1. Notations and definitions

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DV</td>
<td>Device</td>
</tr>
<tr>
<td>AR</td>
<td>Application reader</td>
</tr>
<tr>
<td>SERVER</td>
<td>Server</td>
</tr>
<tr>
<td>pkdb</td>
<td>Publickey</td>
</tr>
<tr>
<td>(+)</td>
<td>Exclusive OR</td>
</tr>
<tr>
<td>H</td>
<td>Hashfunction</td>
</tr>
<tr>
<td>Key A, key B</td>
<td>Sessionkey</td>
</tr>
<tr>
<td>Ta, Tb</td>
<td>Timestamp</td>
</tr>
</tbody>
</table>

3.1 Casper specification

[Fig. 2] shows the proposed protocol’s Casper specification codes. 3 important domains used for security protocols, i.e. variable type declaration, action procedure and attacker model, are listed. Variable and function types are defined in #Free variables. InverseKeys = (keyA,keyB),(pkdb,skdb),(keyT,keyT) is declared to mean that each function returns inverse keys. #Protocol description defines the messages transmitted in the protocol. Integers 0, 1 and 2 stand for the steps of message transmission. #Intruder Information defines the name and initial information of an attack host. The name of the attacker here is Mallory, who is aware that hosts are Application reader, Device, Mallory, DataBase and SM.

#Free variables

DV, AR : Agent
Server : Database Server
pkdb : PublicKey
skdb : SecretKey
keyA, keyB : SessionKey
H : HashFunction
Ta, Tb : TimeStamp
InverseKeys = (keyA,keyB),(pkdb,skdb),(keyT,keyT)

#Protocol description

0. DV -> AR : Ta,{AR,keyB}{pkdb}%enc
   [AR !=DV]
1. AR -> Server : Tb,{DV,keyA} {pkdb},enc{AR,keyB} {pkdb}
2. Server -> AR : keyB(+)keyA(+)H(AR), Tb
3. AR -> DV : {AR} {keyB}(+)H(AR), Ta
4. DV -> AR : H(keyB), Ta

#Intruder Information
Intruder = Mallory
Intruder Knowledge = {Application reader, Device, Mallory, DataBase, SM}

Fig. 2 Casper specification in the proposed protocol

3.2 Description on important actions

Following are stepwise descriptions.
Step 1 : DV ➔ AR
A device DV receives from an application reader the Query. Then, DV generates $T_a$, AR value, SessionKey and PublicKey, concatenates each value with the variable $\%Emc$ and transmits it to AR. Here, the values generated are eigenvalues that cannot be generated on other devices.

Step 2 : AR $\rightarrow$ SERVER

Receiving the $T_a$, {AR, SkeyB}{pkdb}$\%$enc value from DV, the reader calculates the $T_b$,{DV,keyA}{pkdb}, enc%{AR,keyB}{pkdb} value, which is in turn transmitted to the database server.

Step 3 : Server $\rightarrow$ AR

Using the $T_b$,{DV,keyA}{pkdb},enc%{AR,keyB}{pkdb} value transmitted by the reader, the database server calculates the $keyB(+keyA(+H(AR),Tb$ value, which in turn is transmitted to the reader.

Step 4 : AR $\rightarrow$ DV

AR authenticates the $keyB(+keyA(+H(AR),Tb$ value received from the database and generates the {AR}{keyB}(+)H(AR)$,Ta value. Here, the fixed length data are hashed as follows. The first hash function is calculated as $h_a(x)^prima = (\sum_{i=0}^{k} a_i \cdot x_i \mod 2^w) + 2^w$, which is substituted for the varying length string and gets $h_a(x) = h_{int}(\sum_{i=0}^{k} x_i \cdot a_i \mod \rho)$.

Therefore, the results calculated as $[R][SkeyB] \oplus h_a(AR) = h_{int}(\sum_{i=0}^{k} x_i \cdot a_i \mod \rho)$ are transmitted to the DV.

Step 5 : DV $\rightarrow$ AR

Finally, on receiving the {AR}{SkeyB}$\oplus$ $h_a(AR)$, $Ta value from AR, DV compares it with the one generated by the tag. When the value is confirmed, DV has its ID encrypted with a hash operation $h_a(SkeyB) = h_{int}(\sum_{i=0}^{k} x_i \cdot a_i \mod \rho)$, transmits it to AR and completes the authentication session in the tag. Later on, AR transmits the $h_a(SkeyB) = h_{int}(\sum_{i=0}^{k} x_i \cdot a_i \mod \rho)$ value from DV to the database server to verify the AV’s hash value saved. Upon completion of normal authentication, the hashed codes and DV codes are confirmed. Then, the authentication session ends followed by secure and stable communication.

3.3 Tests and results

FDR 2.91 model verification tool was used to verify the actions of the M2M protocol designed here in terms of its safety, deadlock and livelock.
Fig. 3 Verification in progress

[Fig. 3] illustrates the state where the source file is loaded and run with no fundamental errors. Upon completion of the verification, a green √ is marked, indicating the verification result is safe. The security protocol proposed was verified using the FDR tool and as in [Fig. 4] it met the requirement for every security attribute.

Fig. 4 Results of security verification of the proposed protocol

[Fig. 4] presents 3 verification results, each of which representation is analyzed as follows.

SECRET_M::SECRET_SPEC[T=SECRET_M::SYSTEM_S

With the security of the proposed protocol ensured, the tick mark before the message represents the protocol has not been exposed to intruders. The security of the inter-agent communication, PublicKey and SessionKey is verified.

SECRET_M::SEQ_SECRET_SPEC[T=SECRET_M::SYSTEM - S_SEQ

This item is the result of verifying if the protocol operates in a normal process in the system. In short, the proposed protocol operates in a safe and secure process.

AUTH1_M::AuthenticateRESPONDERToINITIATORAgreement_k[T=AUTH1_M::SYSTEM_1
state that whether the Responder and Initiator can authenticate each other via k and x is verified. In short, the authentication between the two is secure.

4. Conclusion

M2M has been extensively studied and applied with the advancement of RFID/USN technology. It is widely in use in such fields as theft tracking, person/animal protection, controlling vending machine, weather, environment, national defense and medical care. However, any external intruder could have a crippling impact on the security of M2M system operation and control. Security issues may cause many issues such as device malfunction and privacy that would get in the way of further development of M2M system. Thus, to solve security issues, diverse study efforts have been made using encryption protocols. This paper used PublicKey, Hash Function and Exclusive OR for designing and testing the proposed protocol and applied the time stamp together with the hash lock operation

$$h_a(SkeyB) = h_{int}(\sum_{i=0}^{k} x_i \cdot a^i \mod p)$$

to transmit a different value per session, whilst enhancing the transmission efficiency by encrypting data in transmission. Then, using the specifications in Casper language, the proposed protocol was verified in terms of whether it met the security attributes of the FDR tool. In brief, the proposed protocol was found to satisfy every security aspect of FDR, e.g. replay attack, man-in-the-middle attack, forgery attack, safety verification, deadlock verification and live-lock verification. The present findings suggest the following implications. First, the formal verification in protocol proposal beyond the theorem proving is to reduce human error in protocol verification, ensuring more effective security verification. Second, the method used in this study will remedy several vulnerabilities of different protocols, facilitating fast system development. Future studies will delve into and verify a safer and more efficient protocol using more complex and potent functions applicable to important M2M devices used in medical care and national defense systems.

5. References