Implementing and Verifying a Secure M2m Mutual Authentication Protocol Based on Hash Functions

Kun-Hee Han, Yoon-Su Jeong, Woo-Sik Bae

Abstract*:M2M (Machine To Machine) communication works using the information shared between devices, between a server and devices and between servers without a direct human intervention. As M2M communication service uses the information of devices, it is crucial to authenticate if devices involved are authorized ones prior to the completion of communication. Also, given the potential vulnerability of communication data to such security threats as theft, exposure and modification, it is necessary to develop a secure authentication method against security threats [1,2]. This paper proposes a protocol for mutual authentication and key exchange between devices in M2M communication. The proposed method ensures secure authentication using hash functions and two nonces. To verify the security performance of the proposed authentication protocol for M2M devices, Casper/FDR verification tools are used. The formal verification results highlight the proposed authentication protocol for M2M devices ensures secure mutual authentication and key exchange against masquerade, replay attack and man in the middles

***Index Terms*: Authentication protocol, M2M Service, Integrated authentication protocol, Model Checking, Security Policy**

I. INTRODUCTION

The rapid advancement of network technology has increased the access to superfast communication networks, engendering a range of services and businesses and enabling large amounts of information processing [3,4]. Currently, M2M(Machine To Machine) communication networks are intelligent enough to allow automatic inter-device communication without a direct human intervention, accumulate information and accordingly respond to situations by, say, sounding an alarm. Still, many researchers have dealt with security issues arising in diverse settings of M2M systems in practice [5,6,7]. In M2M environment, wireless communication is needed with devices located in remote areas or where humans cannot access. When infiltrators eavesdrop on any vulnerable and insecure wireless communication, they are highly likely to gather and analyze the communication data for replay attacks and other malicious attempts [8,9]. To address such challenges, the hash lock protocol has been proposed. Yet, tag data may be exposed to spoofing and replay attacks and location tracking in the hash lock protocol using a metaID [10]. The protocol proposed by Kenji et al. has an initial transmission session, where is not encrypted before transmission, which may cause infiltrators to collect data and forge some unencrypted parts in other tags for replay and other attacks [11]. This paper proposes a protocol capable of coping with such security vulnerabilities in wireless communication. Using Casper/FDR [12,13] designed for formal verification of security processes, we verify the security authentication of the proposed protocol in M2M communication.

Revised Manuscript Received on May 22, 2019.

Kun-Hee Han, Department of Information Communication Engineering, Baekseok University, 115, Anseo-Dong, Cheonan 330-704, Chungnam, South Korea

Yoon-Su Jeong, Department of Information Communication Engineering, Mokwon University, 88 Doanbuk-ro,Seo-gu, Daejeon, Korea

Woo-Sik Bae, Department of AIS Center, Ajou Motor College, 106, Daehak Road, Jupo-Myeon, Boryeong-Si 355-769, Chungnam, Korea

II. RELATED WORK

A. Hash lock technique

Proposed by MIT for streamlined communication at lower cost compared to barcodes in cases where large amounts of tags are used, the hash-lock [10] protocol is a method of authenticating keys assuming they are securely shared with tags and databases. The authentication process runs as follows.

1) Hash lock process

① A reader selects a random key, and calculates its hash(key) using a meta ID value.

② Reader records the metaID in a Tag.

③ Tag enters a locked state.

④ Reader saves the metaID and key

2) Hash unlock process Figure 1

① A reader sends a query to a tag about T's metaID.

② Reader searches a database for the metaID and key.

③ Reader sends the key to Tag.

④ If the hash(key) and the metaID match, Tag gets out of the locked state.

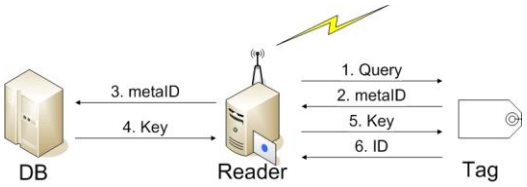

Figure 1: Hash-lock technique

This method uses a fixed metaID as a tag identifier. As the wireless data from a metaID are identical, it is possible to identify the tag that sends the data. Also, as the wireless communication channel between a reader and a tag is

vulnerable to eavesdropping, a malicious attacker may acquire the key and resend the metaID for authentication. Furthermore, a metaID used as an identifier is vulnerable to user tracing and spoofing.

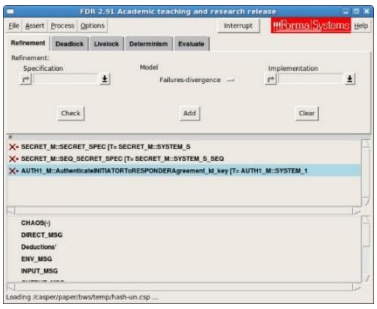

Figure 2: Verification result of hash lock

As shown in figure 2, Casper/FDR verify three security vulnerabilities of the hash lock technique, which are marked x.

B. Kenji et al. protocol technique

Kenji et al. protocol uses a secret value pre-distributed to a user by a database server to create a one-t ID, incrementing the value by 1 in each operation in a synchronized manner. The reader and server apply a complex formula for calculation to deter attackers from attempting to use the secret value shared with I incremented by 1. Afterwards, the server receives and identifies the one-t ID by comparing it with the value of its own operating. Thus, attackers cannot reversely infer the one-t ID created as per the complex formula [11]. However, this protocol has an initial session where is transmitted without being encrypted, which may allow attackers to forge the data they acquire, if any, for attacks. Also, in case attackers randomly write messages for attacks, system integrity is compromised with security vulnerabilities ensuing. Kenji et al. technique transmits data as below.

(1) A → B : $S, O_{AS,}(B, g^x, O_{AS})_{kAS}$
(2) B → S : $O_{AS,}(B, g^x, O_{AS})_{kAS,}(g^y, O_{as})_{kBS}$
(3) S → B : $S, (A, g^x, g^y, (g^x, g^y)_{kAS})_{kBS}$
(4) B → A : $g^x, (g^x, g^y)_{kAS}, (g^x, N_b)_{kAB}$
(5) A → B : $(N_b - 1)_{kab}$

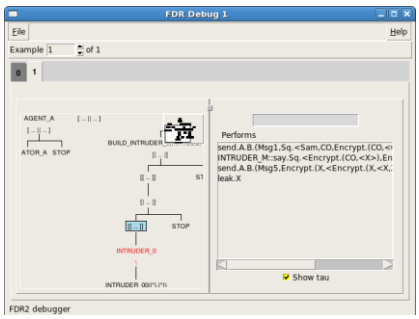

Figure 3: Debugging Kenji et al. technique with Casper/FDR

As the status of Kenji et al. technique verified with Casper/FDR in figure 3, vulnerability lies in the unencrypted transmission.

III. PROPOSED PROTOCOL FOR M2M

The proposed protocol is applicable to sending and receiving information in wireless security communication between tags and readers in the M2M inter-device communication section [14,15,16]. Given the presence of diverse security threats in the inter-device wireless communication section, this paper propose this protocol with intent to provide secure communication environment against such security threats as hacking and device manipulation by infiltrators. The proposed protocol is designed to draw on different types of nonce, nonce1, shared, session keys and hash functions, as well as TimeStamp to preclude timing attacks. table 1 defines the symbols used in the proposed inter-device transmission security protocol for M2M.

Table 1: Symbols and definition

Symbols	Definition
Tm_i^j	AgentTag
Rm_i^j	AgentReader
S_{DB}	DBServer
H	Hash Function
Shared	Agent x Agent -> SharedKey
Td, Td`	TimeStamp
$(A)_{pkb}$, $(B)_{skb}$	SessionKey
Request	Nonce1
N_x , N_y	Nonce2

A. Casper specification

figure 4 shows the essential part of the Casper-specified codes of the proposed security protocol verifying the M2M inter-device wireless communication. Specified function types and variables are defined under #Free variables. InverseKeys = (Request,Request),(N_x,N_x),$((A)_{pkb},(A)_{pkb})$,(N_y,N_y),$((B)_{skb},(B)_{skb})$,(Tm_i^j,Tm_i^j)

method each agent and function return their inverse key values. The core part of the protocol is specified under #Protocol description, where the sequences of calculation and transmission in the protocol are defined. Integers 1, 2, 3 etc. indicate the order of data to be sent.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

0. ->Tm_i^j : Rm_i^j
1. Tm_i^j ->Rm_i^j : Request{N_x}{$(A)_{pkb}$}%mem1
2. Rm_i^j ->S_{DB} : Shared{m1%{N_x}{$(A)_{pkb}$},$(B)_{skb}$,N_y}
3. S_{DB} ->Rm_i^j : Shared{H(Tm_i^j (S_{DB}))},N_x,{N_y}{$(A)_{pkb}$}%mem2}{$(B)_{skb}$,Td}
[Td==now or Td+1==now]
4. Rm_i^j ->Tm_i^j : {$(A)_{pkb}$,Td`}mem2%{N_y}{$(B)_{skb}$}
[Td'==now or Td'+1==now]
5. Tm_i^j ->Rm_i^j : {$(B)_{skb}$}H(Tm_i^j)

Figure 4: Casper specification in the protocol

B. Algorithm specification of proposed protocol

The proposed security protocol for M2M inter-device communication runs in the order described below.

(Step ① : $Tm_i^j \rightarrow$Rm_i^j)

The Tm_i^j receives the Query from the Rm_i^j, calculates Nonce 1, generates {N_x}{$(A)_{pkb}$}%mem1 for concatenation and saves this value in the variable %mem1. The Tm_i^j sends the calculated Request{N_x}{$(A)_{pkb}$}%mem1 to the Rm_i^j. Here, the value generated is based on random numbers and session keys and unique to the very Tm_i^j, not generated by other Tm_i^js. The communication data sent here cannot be used for replay attacks in case of eavesdropping and are designed to deter attackers from using them for attacks, since Nonce1, Nonce2 and SessionKey generate different values in each operation.

(Step ② : R$m_i^j \rightarrow S_{DB}$)

The Rm_i^j calculates the Request{N_x}{$(A)_{pkb}$}%mem1 sent by the Tm_i^j using its own shared{$(B)_{skb}$,N_y} to yield the following value: Shared{mem1%{N_x}{$(A)_{pkb}$},$(B)_{skb}$,N_y}, which is calculated with the Request{N_x}{$(A)_{pkb}$}%mem1 data sent from the Tm_i^j and subject to a concatenation operation. Then, the Rm_i^j checks the data received and saves the data in the variable mem1. It sends Shared{mem1%{N_x}{$(A)_{pkb}$},$(B)_{skb}$,N_y} data normally generated to S_{DB}. Here, it is secure for the Rm_i^j and S_{DB} to operate on a wire or wireless communication network.

(Step ③ : $S_{DB} \rightarrow$Rm_i^j)

S_{DB} checks the Shared{mem1%{N_x}{$(A)_{pkb}$},$(B)_{skb}$,N_y} value received from the Rm_i^j and uses the received data to generate a value for cross-certification by checking the value sent by the Rm_i^j and the Tm_i^j value. Then, the session keys $(A)_{pkb}$ and $(B)_{skb}$ are generated, and S_{DB} and Tm_i^j (S_{DB}) undergo a hash operation. Now, it performs a calculation to send a value to the Rm_i^j, and generates a value such as Shared{H(Tm_i^j(S))},N_x,{N_y}{$(A)_{pkb}$}%mem2}{$(B)_{skb}$,Td}. S_{DB} calculates the hash value as per the formula $h_a(x) = h_{\text{int}}\left(\sum_{i=0}^{k} x_i \cdot a^i\right) \bmod p\right)$.

(Step ④ : R$m_i^j \rightarrow Tm_i^j$)

The R m_i^j checks the Shared{H(Tm_i^j(S_{DB}))},N_x,{N_y}{$(A)_{pkb}$}%mem2}{$(B)_{skb}$, Td} value received from S_{DB} and calculates it for mutual authentication. Then, the Rm_i^j checks its own value mem2%, uses it to generate {$(A)_{pkb}$,Td`}mem2%{$N_y$}{$(B)_{skb}$}, calculates hash values $h_a(x) = h_{\text{int}}\left(\sum_{i=0}^{k} x_i \cdot a^i\right) \bmod p\right)$ and $N_x$,{$N_y$}{$(A)_{pkb}$}%mem2}{$(B)_{skb}$,Td}, and concatenates each data to generate {$(A)_{pkb}$,Td`}mem2%{N_y}{$(B)_{skb}$}. The Rm_i^j sends the generated {$(A)_{pkb}$,Td`}mem2%{N_y}{$(B)_{skb}$} value to the Tm_i^j.

(Step ⑤ : $Tm_i^j \rightarrow$Rm_i^j)

Finally, the Tm_i^j receives {$(A)_{pkb}$,Td`}mem2%{N_y}{$(B)_{skb}$} from the Rm_i^j and compares it with the value it holds. Once the two values match, the Tm_i^j calculates using {$(B)_{skb}$}H{Tm_i^j} and sends the value to the Rm_i^j, completing its device security authentication session. Then, the R m_i^j sends {$(B)_{skb}$}H{Tm_i^j} received from the Tm_i^j to S_{DB}, which in turn checks if the value matches the one it holds prior to authentication. Once normal authentication completes, it is possible to validate hash and $(B)_{skb}$ codes and proceed.

IV. TEST RESULTS OF PROPOSED PROTOCOL

Using CASPER/FDR security model verification programs, we tested the proposed M2M device communication protocol specification in terms of livelock, safety and deadlock.

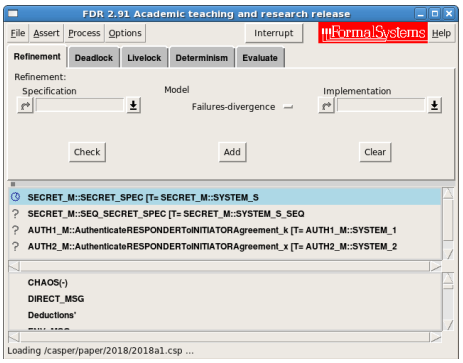

Figure 5: Verifying the proposed protocol

The verification process shown in figure5 involves using CASPER to successfully convert the proposed device authentication protocol into its source codes and using FDR to load it to test its security performance. The ? symbols on the left indicates the protocols verification is not started. figure 6 shows the completed checks on grammar and process in the designed source code loaded. The proposed M2M protocol underwent the verification programs in terms of security and process. As shown in the figure of the completed verification, all attributes verified met requirements. The verification programs display X, on detecting any security vulnerability, and run Debug to determine and correct

relevant issues for repetitive verification.

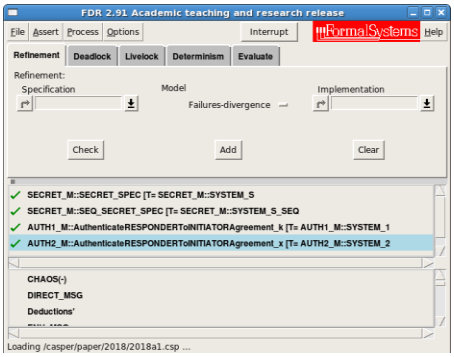

Figure 6: Verification results of proposed M2M protocol

Figure6 shows four verification results with each result analyzed as below.

a) 

a)The proposed M2M security authentication protocol's overall performance and security against multiple attacks were verified. The tick marks displayed before messages represent it is secure without being exposed to attackers. Specifically, the security of inter-agent transmission and session keys were verified against a range of attacks. Hence, the proposed authentication protocol is secure and safe as shown in the figure.

b) 

b)This item verifies if the proposed M2M device authentication protocol seamlessly runs in each step. The protocol is secure and safe against errors, attacks and exposure in each step as shown in figure 6

c, d) 

c, d) This item verifies if the Responder and Initiator can mutually authenticate each other via k without security issues. The proposed protocol ensures the secure mutual authentication between agents.

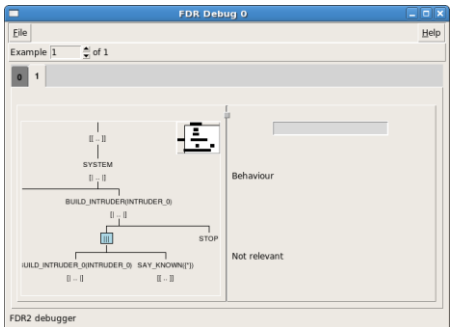

Figure7: Status window showing completed verification of protocol

figure7 shows a status window about the completed verification of the proposedauthentication protocol, which satisfies the inter-session safety and fully operates in each step without coming to a deadlock. Also, it completes the verification without causing any system issues resulting from memory errors, infinite loops and processing delays. Thus, the proposed protocol meets security requirements as verified with its entire process stably running.

V. CONCLUSION

The evolving IoT technology has led the advancement of M2M devices, research on their applications, and installation thereof in practice. M2M devices process information without a direct human intervention and engage in inter-device wireless communication, where information protection overrides anything else. Also, any manipulated, leaked or modified device information can result in serious issues on systems [17]. To address the security-related challenges, and to ensure the security of M2M inter-device communication, this paper proposes a authentication protocol designed to prevent timing attacks using hash functions, nonce1, nonce2, session keys and timestamps.The verification results substantiate the security of the proposedauthentication protocol against vulnerabilities in M2M communication in comparison to other protocols drawing on different operations. Moreover, the formal verification of the operation status of the proposed protocol is more effective and less prone to mistakes than other approaches to designing formulaic security protocols. Further research will adopt more diverse functions and operations for efficient and secure authentication by linking devices in meteorology and minimal computation.

ACKNOWLEDGMENT

This research is supported by 2019 Baekseok University research fund.

REFERENCES

1. Seyed Mohammad Alavi, Karim Baghery, Behzad Abdolmaleki, Mohammad Reza Aref,"Traceability Analysis of Recent RFID Authentication Protocols," Wireless Personal Communications, 2015, pp. 1663-1682.
2. Lin, X.-J.a, Sun, L.b, Qu, H.a.,"Insecurity of an anonymous authentication for privacy-preserving IoT target-driven applications," Computers and Security, 2015, pp. 142-149.
3. M. Duraipandian, C. Palanisamy.,"Analysis of a Combined Parameter-Based Multi-objective Model for Performance Improvement in Wireless Networks," Wireless Personal Communications, 2015, pp. 2425-2437.
4. S. R. Mugunthan, C. Palanisamy.,"A Dynamic Interoperability Mobility Management Architecture for Mobile Personal Networks," Wireless Personal Communications, 2015, pp. 1683-1697.
5. G. Wu, S. TalwReader, K. Johnsson, N. Himayat, and K. D. Johnson. "M2M: from mobile to embedded internet," IEEE Communications Magazine, 2011, pp. 36-43..
6. H. S. Ahn, K. D. B, E. J. Yoon, I. G. Nam. RFID Mutual Authentication Protocol Providing Stronger Security, The KIPS Transactions, 2009, pp. 325-334.
7. M. Aiash, J. L.,"An integrated authentication and authorization approach for the network of information architecture," Journal of Network and Computer Applications, 2015, pp. 73-79.
8. Yu-Yi Chen, Jun-Chao Lu, Jinn-Ke Jan.,"A Secure EHR System Based on Hybrid Clouds," Journal of Medical Systems, 2012, pp. 3375-3384.
9. Y. Liu et al.,"Double verification protocol via secret sharing for low-cost RFID tags," Future Generation Computer Systems, 2018, pp. 118–128.
10. Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest and Daniel W. Engels.,"Security and Privacy Aspects of Low-cost Radio Frequency Identification Systems," Security in Pervasive Computing, 2004, pp. 201-202.
11. Kenji Imamoto and Kouichi Sakurai.,"Design and Analysis of Diffie-Hellman Based Key Exchange Using ID by SVO Logic, Proc." Electronic Notes in Theoretical Computer Science, 2005, pp.79-94.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

12. G. Lowe, P. Broadfoot, C. D. M. H.,"Casper: A compiler for the analysis of security protocols," in Technical report, Oxford, 2009.
13. Formal Systems(Europe) Ltd, "Oxford University Computing Laboratory, Failures-Divergence Renement," FDR2 User Manual, 2010.
14. L. Gao, M. Ma, Y. Shu, F. Lin, L. Zhang, Y. Wei.,"A low-cost RFID authentication protocol against desynchronization with a random tuple," Wireless Personal Communications, 2014, pp. 1941–1958.
15. M. Aiash.,"A formal analysis of authentication protocols for mobile devices in next generation networks," Concurrency and Computation: Practice and Experience, 2014.
16. Niu, B., Zhu, X.a, Li, Q.c, Chen, J.a,Li, H.a.,"A novel attack to spatial cloaking schemes in location-based services," Future Generation Computer Systems, 2015, pp. 125-132.
17. Cirani, S., Picone, M.,Gonizzi, P.,Veltri, L.,Ferrari, G.,"IoT-OAS: An oauth-based authorization service architecture for secure services in IoT scenarios," IEEE Sensors Journal, 2015, 1224-1234.

AUTHORS PROFILE

Kun-Hee Hanreceived Ph.D. degree from Chungbuk National University in 2000, and curretly he is working at Baekseok University as associate professor. His interests are secure database, cloud computing, and information security. He is a member of the KCS and the SDPM.

Yoon-Su Jeongwas born in Cheong-Ju, Korea in 1975. He receivedthe B.S. degree in the Department of Computer Science, CheongjuNational University in February 1998. He received the M.S. degreeand Ph.D. in the Department of Computer Science, Chungbuk National University in February 2000 and 2008. He is currently working professorin the Department of Information and Communication ConvergenceEngineering, Mokwon University. His research interests alsoinclude cryptography, network security, information security, AAA,wire/wireless communication security.

Woo-Sik Baereceived his Ph.D. degree in Computer Education fromChungbuk National University, Korea in 2012. He has published morethan 52 papers on IoT security in international and Korean journals andconferences. His research interest includes: VANET, computer andnetwork security, IoT security, authentication protocol, and Convergence.He steering committees member of the International ConferenceConvergence Technology (ICCT), International Conference onDigital Policy and Management (ICDPM), and International Conferencefor Small and Medium Business (ICSMB). He is a member ofKCS and SDPM.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication