Original Research Paper | Computer Science | Volume : 5 ¦ Issue : 9 ¦ Sep 2020 ¦ e-ISSN : 2456-5040

A ROBUST LIGHTWEIGHT THREE-FACTOR AUTHENTICATION SCHEME FOR IOT IN A CLOUD COMPUTING ENVIRONMENT

Yanbao Han[1], Baoyuan Kang[1], Kun Qian[1]

[1]School of Computer Science and Technology, Tiangong University, Tianjin, 300387, China.

ABSTRACT

With the advancement of computer science and technology, the society pays more and more attention to the application of Internet of Things technology. The Internet of Things technology brings convenience to people's lives, but various security vulnerabilities in IoT devices still threaten people's privacy. We describe the occurrence process of these security flaws and propose a robust lightweight three-factor authentication scheme based on IoT in a cloud computing environment. Then we analyzed the security and performance of the proposed scheme.

KEYWORDS: Cloud Computing; Key Agreement;Anonymity; Internet of Things

1. INTRODUCTION:

In recent years, the rapid development of the Internet of Things has brought great convenience and efficient collaboration to people's lives. The definition of Internet of Things for smart environments is Interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications[1]. Furthermore, cloud computing technology is an important branch of the Internet of Things. Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services[2]. Cloud computing [3] is the most recent paradigm to emerge which promises reliable services delivered through next generation data centres that are based on virtualised storage technologies. This platform acts as a receiver of data from the ubiquitous sensors; as a computer to analyze and interpret the data[1]. Applying cloud computing technology to a wide range of IoT devices, when users need to obtain the resources of the cloud computing server, user identity authentication is inevitable. Remote user authentication is a mechanism to authenticate remote users over insecure communication network. User authentication is a central component of any security infrastructure. Other security measures depend upon verifying the identity of the sender and receiver of information. Authorization grants privileges based upon identity. Audit trails would not provide accountability without authentication. If we cannot reliably differentiate an authorized entity from an unauthorized entity, confidentiality and integrity are broken. To access resources at remote systems, users should have proper access rights. One of the simplest and most convenient security mechanisms is the use of a password authentication scheme [4]. Therefore, how to strengthen the security of IoT devices is a very important and urgent issue. In this paper, we propose an effective and secure three-factor authentication scheme based on the IOT environment.

2. RELATED WORKS:

In order to improve the authentication and key agreement schemes based on cloud computing environment, many schemes have been proposed[5-10]. In 2007, Liao et al. [11] proposed a key agreement protocol using the concept of dynamic identity for multi-server environment based on cryptographic hash function. Two years later, Hsiang et al. in [12] point out that Liao et al.'s protocol is not secure to several threats and designed an extended protocol. In 2011, Sood et al. [13] prove the flaws of the Hsiang et al.'s protocol and its password change process is not accurate. Afterwards, Sood et al. [13] raised a dynamic identity based multi-server authentication protocol. In 2012, Li et al.[14] confirmed that Sood et al.'s scheme had security flaws, in order to improve these security flaws, they developed a counter measure protocol. In 2014, Xue et al.[15] proposes a better security improvement protocol to enhance the scheme proposed by Li et al. In 2018, Amin et al.[16] proposes a security attacks free authentication protocol which can be used in distributed cloud environment and demonstrated that Xue et al.'s protocol cannot not resist user impersonation and session key discloser attack. In 2019, Zhou et al. [17] consider that Amin et al.'s scheme cannot against off-line guessing attacks. Then they proposed a scheme based on hash function and exclusive-or operation to provide authentication on large-scale IoT and cloud computing deployment. Same year, Pelaez et al.[18] find Zhou et al.'s scheme has security defects. In particular, the scheme is vulnerable to user impersonation attacks. Moreover, Pelaez et al. propose an enhanced lightweight IoT-based authentication scheme in cloud computing circumstances. Afterwards, SungJin Yu et al.[19] demonstrate that Pelaez et al.'s scheme is not secure and propose a secure and lightweight three-factor authentication scheme for IoT in cloud computing environment.

3. THE PROPOSED SCHEME:

We propose a secure authentication and key agreement scheme. The proposed scheme consists of four phases: registration phase, login phase, authentication and key agreement phase, and password change phase. The list of notations used in the proposed protocol is given in Table 1. The details of each phase are as follows.

Table 1: The symbols and definitions used in this paper.

Symbol	Description
U_i	User
S_j	Cloud server
CS	Control server
SC	Smart card of user
ID_i	Identity of user
SID_j	Identity of cloud server
BIO_i	Biometric template of user
c_i	A unique identification code for user
c_j	A unique identification code for cloud server
PW_i	Password of user
X_{CS}	Master key of control server

X_{sj}	Secret key of cloud server
SK	Session key
$E_k(\cdot)$	Encrypt the text with a key k using an encryption algorithm
$D_k(\cdot)$	Using a decryption algorithm to decrypt a cipher text using a key k
$Gen(\cdot)$	the biometric key generation algorithm
$Rep(\cdot)$	the bio-key recovery algorithm
$h(\cdot)$	One-way hash function
$\oplus$	Bit-wise XOR operation
$\|$	Concatenate operation

3.1 Registration phase:

3.1.1 User registration:

When a new user wants to use cloud server services, they need to register with the control server. This phase is described below.

The user (U_i) chooses his/her identity ID_i, password PW_i and imprints biometric BIO_i. Then U_i computes $\langle R_i, P_i \rangle = Gen(BIO_i)$ through the biometric key generation algorithm $Gen(\cdot)$ in the fuzzy extractor [21], where R_i is the bio-key, and P_i is the public information. Then U_i computes $A_i = h(ID_i \| PW_i \| R_i)$ and sends user registration messages $\{ID_i, A_i\}$ to control server CS via a secure channel.

Upon receipt of the registration request, the control server CS generates a unique identification code c_i for U_i. Afterwards, the CS computes

$$RID_i = h(ID_i \| X_{CS} \| c_i),$$
$$B_i = h(RID_i \| X_{CS} \| c_i),$$
$$EID_i = ID_i \oplus h(RID_i \| c_i \| X_{CS})$$

with its master key X_{CS}, and computes $Q_i = h(A_i) \oplus RID_i$, $D_i = h(A_i \| B_i)$, $E_i = A_i \oplus B_i$. After that, the CS stores a table $\{RID_i, c_i, EID_i\}$ into its database and distributes a new smart card $SC=\{D_i, E_i, Q_i, h(\cdot), Rep(\cdot), E_k(\cdot), D_k(\cdot)\}$ to U_i through a secure channel.

On receiving SC, the U_i uses the smart card to store P_i. Therefore, $SC=\{D_i, E_i, Q_i, h(\cdot), Rep(\cdot), E_k(\cdot), D_k(\cdot), P_i\}$.

3.1.2 Cloud server registration:

When a new cloud server S_j is deployed, it should register in CS. This phase of the scheme is run as shown below.

The cloud server S_j selects its identity SID_j, generates a random nonce b_j, computes $F_j = h(SID_j \| X_{sj} \| b_j)$ with its secret key X_{sj}. The S_j sends cloud server registration messages $\{SID_j, F_j\}$ to control server CS via a secure channel.

Upon receipt of the registration request, the control server CS generates a unique identification code c_j for S_j. Afterwards, the CS computes

$$RSID_j = h(SID_j \| X_{CS} \| c_j),$$
$$Y_j = h(RSID_j \| X_{CS} \| c_j),$$
$$L_j = F_j \oplus Y_j,$$
$$ESID_j = SID_j \oplus h(RSID_j \| c_j \| X_{CS}).$$

After that, the CS stores a table $\{RSID_j, c_j, ESID_j\}$ into its database and sends $\{L_j, RSID_j\}$ to S_j through a secure channel.

After receiving the messages from CS, the S_j stores $\{L_j, RSID_j, b_j\}$ in its database.

3.2 Login phase:

When a user U_i wants to get service from the cloud server, s/he must first perform the login phase.

The U_i inputs his/her ID_i and PW_i, then imprints his/her biometric information BIO_i using the smart card. The smart card SC computes $R_i^* = Rep(BIO_i^*, P_i)$ using the bio-key recovery algorithm of the fuzzy extractor[21]. Next, SC computes

$$A_i^* = h(ID_i \| PW_i \| R_i^*),$$
$$RID_i = Q_i \oplus h(A_i^*),$$
$$B_i = E_i \oplus A_i^*,$$
$$D_i^* = h(A_i^* \| B_i)$$

Then the SC checks $D_i^* ? = D_i$. If the condition does not hold, it rejects the connection. Otherwise, the smart card SC generates a random nonce RU_i and a timestamp T_1. Next, SC asks the user U_i for the SID_j of the cloud server to connect to. The U_i inputs SID_j, and the smart card computes, $M_1 = RU_i \oplus h(ID_i \| B_i)$ and $V_1 = h(RU_i \| ID_i \| SID_j \| B_i \| RID_i \| T_1)$. Finally, the smart card sends login messages $MSG_1 : \{RID_i, M_1, V_1, T_1\}$ to the cloud server S_j through a public channel.

3.3 Authentication and key agreement phase:

After receiving the message from the U_i, S_j checks the validity of the timestamp $|T_2 - T_1| < \Delta T$. Then the S_j generates a random nonce RS_j and a timestamp T_3, and computes $F_j^* = h(SID_j \| X_{sj} \| b_j)$ with its secret key X_{sj}, computes

$$Y_j^* = L_j \oplus F_j^*,$$
$$M_2 = RS_j \oplus h(SID_j \| Y_j^*),$$
$$V_2 = h(RS_j \| Y_j^* \| SID_j \| RSID_j \| T_1 \| T_3).$$

Afterward, the S_j sends messages $MSG_2 : \{RID_i, M_1, V_1, RSID_j, M_2, V_2, T_1, T_3\}$ to the control server CS through a public channel.

After receiving the message from the S_j, CS checks the validity of the timestamp $|T_4 - T_3| < \Delta T$ and searches whether RID_i and $RSID_j$ is in its database. If conditions are not met, CS rejects current session. Then the CS computes

$$ID_i^* = EID_i \oplus h(RID_i^* \| c_i \| X_{CS}),$$
$$SID_j^* = ESID_j \oplus h(RSID_j^* \| c_j \| X_{CS}),$$
$$B_i^* = h(RID_i^* \| X_{CS} \| c_i),$$
$$Y_j^* = h(RSID_j^* \| X_{CS} \| c_j)$$

with its master key X_{CS}. After that CS computes

$$RU_i = M_1 \oplus h(ID_i^* \| B_i^*),$$
$$RS_j = M_2 \oplus h(SID_j^* \| Y_j^*).$$

In addition, CS computes $V_1^* = h(RU_i \| ID_i^* \| SID_j^* \| B_i^* \| RID_i \| T_1)$, $V_2^* = h(RS_j \| Y_j^* \| SID_j^* \| RSID_j \| T_3)$. If $V_1^* \neq V_1$ or $V_2^* \neq V_2$, CS rejects the current session. Otherwise, the CS generates a random nonce Rcs and a timestamp T_5. Then CS computes

$$Z_1 = h(ID_i^* \| RU_i),$$
$$SK = h(Z_1 \| SID_j \| RS_j \| Rcs),$$
$$M_3 = (Z_1 \| Rcs) \oplus h(Y_j^* \| SID_j^*),$$
$$M_4 = (RS_j \| Rcs) \oplus h(B_i^* \| ID_i^*),$$
$$V_3 = h(Rcs \| M_4 \| V_4 \| SK \| T_5).$$

Furthermore, CS updates RID_i to RID_i^{new}, which $RID_i^{new} = h(RID_i \| RU_i)$, and computes $B_i^{new} = h(RID_i^{new} \| X_{cs} \| c_i)$, $S = E_{B_i^*}(B_i^{new})$, which $E(\cdot)$ is an encryption algorithm like AES (Advanced Encryption Standard). Finally, the CS computes $V_4 = h(Rcs \| ID_i^* \| B_i^* \| S \| SK \| RID_i^{new} \| T_5)$ and sends messages $MSG_3:\{M_3, M_4, V_3, V_4, S, T_5\}$ to the cloud server S_j through a public channel.

After receiving the message from the CS, S_j checks the validity of the timestamp $|T_6 - T_5| < \Delta T$. S_j generates a timestamp T_7 and computes

$$(Z_1^* \| Rcs^*) = M_3 \oplus h(Y_j \| SID_j),$$
$$SK = h(Z_1^* \| SID_j \| RS_j \| Rcs^*),$$
$$V_3^* = h(Rcs^* \| M_4 \| V_4 \| SK \| T_5).$$

Then S_j checks $V_3^* ? = V_3$, if the verification does not hold, rejects the current session. At last, the S_j sends messages $MSG_4:\{M_4, V_4, S, T_5, T_7\}$ to the user U_i through a public channel.

After receiving the message from the S_j, U_i checks the validity of the timestamp $|T_8 - T_7| < \Delta T$. Then U_i computes

$$(RS_j^* \| Rcs^*) = M_4 \oplus h(B_i^* \| ID_i^*),$$
$$SK = h(h(ID_i \| RU_i) \| SID_j \| RS_j^* \| Rcs^*),$$
$$RID_i^{new} = h(RID_i \| RU_i),$$
$$V_4 = h(Rcs^* \| ID_i^* \| B_i^* \| S \| SK \| RID_i^{new} \| T_5).$$

The U_i checks $V_4^* ? = V_4$, if it is not true, U_i rejects the session. Otherwise, U_i updates

$$B_i^{new} = D_{B_i}(S) = h(RID_i^{new} \| X_{cs} \| c_i),$$
$$E_i^{new} = A_i \oplus B_i^{new},$$
$$Q_i^{new} = h(A_i) \oplus RID_i^{new},$$
$$D_i^{new} = h(A_i \| B_i^{new})$$

into the smart card SC.

3.4 Password change phase:

A legal user updates his/her old password PW_i and the biometric BIO_i as follows:

U_i inserts own SC, inputs his/her ID_i, PW_i and imprints his/her biometrics BIO_i. SC computes

$$R_i^* = Rep(BIO_i^*, P_i),$$
$$A_i^* = h(ID_i \| PW_i \| R_i^*),$$
$$RID_i = Q_i \oplus h(A_i^*),$$
$$B_i = E_i \oplus A_i^*,$$
$$D_i^* = h(A_i^* \| B_i).$$

Then SC checks D_i^*, if $D_i^* \neq D_i$, terminates password update. SC then compares D_i^* with the stored D_i. If this condition is not satisfied, SC terminates this phase.

U_i inputs a new password PW_i^{new} and a new biometrics BIO_i^{new}. SC computes

$$\langle R_i^{new}, P_i^{new} \rangle = Gen(BIO_i^{new}),$$

$A_i^{new} = h(ID_i \| PW_i^{new} \| R_i^{new})$,

$Q_i^{new} = h(A_i^{new}) \oplus RID_i$,

$E_i^{new} = A_i^{new} \oplus B_i$,

$D_i^{new} = h(A_i^{new} \| B_i)$.

SC replaces corresponding parameters. Ultimately, SC contains $\{D_i^{new}, E_i^{new}, Q_i^{new}, P_i^{new}\}$.

4. SECURITY ANALYSIS:

This section performs an informal security analysis of the proposed scheme to evaluate the security performance.

4.1 Stolen smart-card attack:

In the proposed scheme, suppose the adversary has already acquired the smart card, s/he can only extract $\{D_i, E_i, Q_i, P_i\}$ from the smart card. Because the adversary lacks some information $\{ID_i, B_i\}$, s/he cannot calculate $\{RID_i, M_1, V_1\}$ and establish a new session. Assume that the adversary can intercept the channel and achieve RID_i, s/he still cannot derive ID_i from RID_i, because ID_i is hash-protected.

4.2 Replay attacks:

Occurs when malicious messages are forwarded maliciously to disrupt traffic or produce unauthorized effects. Assume that the adversary eavesdrops $\{RID_i, M_1, V_1, T_1\}$ from the authentication and key agreement phase through a public channel. And The adversary tried to resend this message but because a timestamp exists, s/he cannot use the message to continue executing the protocol.

4.3 Perfect forward secrecy:

Even if the adversary obtains the master key of the control server, s/he still cannot calculate the old session key. Because every time a session key is generated, a new random number is generated. Therefore, the perfect forward secrecy is supported in the proposed scheme.

4.4 User impersonation attacks:

To impersonate the user U_i, the adversary has to send a valid message $\{RID_i, M_1, V_1, T_1\}$ to the cloud server S_j. If the adversary intends to calculate $M_1 = RU_i \oplus h(ID_i \| B_i)$, s/he needs to know B_i which is impossible. Therefore, it is impossible for the adversary to falsify the messages.

4.5 Privileged-insider attack:

In the Registration Phase, CS stores $\{RID_i, c_i, EID_i\}$ and $\{RSID_j, c_j, ESID_j\}$, but the adversary cannot calculate ID_i and SID_j, because the master key X_{CS} of the CS is in RID_i and $RSID_j$. Assume the adversary eavesdrop messages from a full session, s/he cannot compute any vital information, because it protected by the one-way hash function.

4.6 Denial of service attack:

In the proposed scheme, U_i, CS and S_j all verify the validity of the timestamp. Each message to be verified contains a new timestamp. The proposed scheme can resist denial of service attack.

5. PERFORMANCE ANALYSIS:

We compared the calculated cost of the proposed scheme with the existing schemes[11-14]. Table 2 shows the computational load during the registration, login, and authentication phases. We define T_h and T_s as one-way hash operation and symmetric-key decryption operation, respectively. Conclusions show that our protocol is suitable for running in cloud-based IoT environments.

Table 2: Performance comparison

Schemes		Registration	Login	Authentication	Total
Amin	U_i	$3T_h$	$6T_h$	$3T_h$	$30T_h$
	S_j	$0T_h$	$1T_h$	$3T_h$	
	CS	$4T_h$	$0T_h$	$10T_h$	
Zhou	U_i	$3T_h$	$6T_h$	$4T_h$	$43T_h$
	S_j	$0T_h$	$3T_h$	$4T_h$	
	CS	$4T_h$	$0T_h$	$19T_h$	
Pelaez	U_i	$2T_h$	$3T_h$	$4T_h+3T_s$	$48T_h+8T_s$
	S_j	$1T_h$	$3T_h$	$2T_h+3T_s$	
	CS	$12T_h$	$0T_h$	$21T_h+2T_s$	
SungJin Yu	U_i	$2T_h$	$6T_h$	$4T_h$	$34T_h$
	S_j	$0T_h$	$2T_h$	$4T_h$	
	CS	$6T_h$	$0T_h$	$10T_h$	
Ours	U_i	$1T_h$	$5T_h$	$5T_h+1T_s$	$42T_h+2T_s$

	S_j	$1T_h$	$3T_h$	$3T_h$	
	CS	$8T_h$	$0T_h$	$16T_h+1T_s$	

6. CONCLUSIONS:

We proposed a new three-factor authentication and key agreement protocol based on smart cards and biometrics in the cloud computing environment of the Internet of Things. We prove that the proposed scheme can meet the required security requirements and can resist various attacks. This protocol effectively addresses the security threats that IoT devices may suffer. In order to continue to improve the algorithm, future work may optimize the effectiveness of the algorithm to improve the algorithm.

REFERENCES:

I. [1] Jayavardhana Gubbi,,Rajkumar Buyya,,Slaven Marusic & Marimuthu Palaniswami.(2013).Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems(7),. doi:

II. Armbrust, M. , Fox, A. , Griffith, R. , Joseph, A. D. , Katz, R. , & Konwinski, A. D. , et al. (2010). A view of cloud computing. Communications of the ACM.

III. Levine, John R, & Young, Margaret Levine. (1994). Internet for dummies. Calif Idg Books Worldwide Inc(100).

IV. Madhusudhan, R. , & Mittal, R. C. . (2012). Dynamic id-based remote user password authentication schemes using smart cards: a review. Journal of Network and Computer Applications, 35(4), 1235–1248.

V. Chien, H. Y. , Jan, J. K. , & Tseng, Y. M. . (2002). An efficient and practical solution to remote authentication: smart card. Computers & Security, 21(4), 372-375.

VI. Jianming Zhu, & Jianfeng Ma. (2004). A new authentication scheme with anonymity for wireless environments. IEEE Press.

VII. Wong, K. H. M. , Zheng, Y. , Cao, J. , & Wang, S. . (2006). A dynamic user authentication scheme for wireless sensor networks. IEEE International Conference on Sensor Networks. IEEE.

VIII. Yunho Lee, Seungjoo Kim, & Dongho Won. (2010). Enhancement of two-factor authenticated key exchange protocols in public wireless LANs. Pergamon Press, Inc.

IX. Kim, J. , Lee, D. , Jeon, W. , Lee, Y. , & Won, D. . (2014). Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks. Sensors, 14(4), 6443-6462.

X.

XI. Wang, D. , & Wang, P. . (2014). On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions. Computer Networks, 73(nov.14), 41-57.

XII.

XIII. Liao, Y. P. , & Wang, S. S. . (2009). A secure dynamic id based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces, 31(1), 24-29.

XIV. Hsiang, H. C. , & Shih, W. K. . (2009). Improvement of the secure dynamic id based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces, 31(6), 1118-1123.

XV. Sood, S. K. , Sarje, A. K. , & Singh, K. . (2011). A secure dynamic identity based authentication protocol for multi-server architecture. Journal of Network & Computer Applications, 34(2), 609-618.

XVI. Li, X. , Xiong, Y. , Ma, J. , & Wang, W. . (2012). An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. Journal of Network & Computer Applications, 35(2), 763-769.

XVII. Xue, K. , Hong, P. , & Ma, C. . (2014). A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. Journal of Computer & System ences, 80(1), 195-206.

XVIII. Amin, R. , Kumar, N. , Biswas, G. P. , Iqbal, R. , & Chang, V. . (2016). A light weight authentication protocol for iot-enabled devices in distributed cloud computing environment. Future Generation Computer Systems, 78(PT.3), 1005-1019.

XIX. Zhou, L. , Li, X. , Yeh, K. H. , Su, C. , & Chiu, W. . (2019). Lightweight iot-based authentication scheme in cloud computing circumstance. Future generation computer systems, 91(FEB.), 244-251.

XX. Rafael Martínez-Peláez, Toral-Cruz, H. , Parra-Michel, J. R. , Vicente García, & Ochoa, A. . (2019). An enhanced lightweight iot-based authentication scheme in cloud computing circumstances. Sensors, 19(9), 2098.

XXI. Yu, S. J. , Park, K. S. , & Park, Y. H. . (2019). A secure lightweight three-factor authentication scheme for iot in cloud computing environment. Sensors, 19(16), 3598.