Computer Communications 34 (2011) 326–336

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Robust one-time password authentication scheme using smart card for home network environment

Binod Vaidya [a,*], Jong Hyuk Park [b], Sang-Soo Yeo [c], Joel J.P.C. Rodrigues [a,d,**]

[a] Instituto de Telecomunicações, Covilhã, Portugal
[b] Department of Computer Science and Engineering, Seoul National University of Technology, Seoul, Republic of Korea
[c] Division of Computer Engineering, Mokwon University, Daejeon, Republic of Korea
[d] University of Beira Interior, Covilhã, Portugal

ARTICLE INFO

Article history:
Received 14 September 2009
Received in revised form 29 January 2010
Accepted 11 March 2010
Available online 16 March 2010

Keywords:
Home network
User authentication
Strong-password approach
HMAC-based one-time password
Hash-chaining technique

ABSTRACT

Due to the exponential growth of the Internet users and wireless devices, interests on home networks have been enormously increased in recent days. In digital home networks, home services including remote access and control to home appliances as well as services offered by service providers are alluring. However, the remote control services cause digital home networks to have various security threats. Hence, for digital home networks, robust security services, especially remote user authentication, should be considered. This paper presents a robust and efficient authentication scheme based on strong-password approach to provide secure remote access in digital home network environments. The proposed scheme uses lightweight computation modules including hashed one-time password and hash-chaining technique along with low-cost smart card technology. It aims to satisfy several security requirements including stolen smart card attack and forward secrecy with lost smart card as well as functional requirements including no verification table and no time synchronization. Comparing with the existing representative schemes, it can be validated that the proposed scheme is more robust authentication mechanism having better security properties. We have conducted formal verification of the proposed scheme.

1. Introduction

Due to the exponential growth of the Internet users and wireless devices, interests on home networks have been enormously increased in recent days. Home networks provide remote access control over the connection between digital home appliances and mobile handheld devices through Internet [1–3].

Even though digital home networks provide conveniences to the residential users by providing numerous value-added services, the remote control service causes them to have numerous security threats and vulnerabilities. In particular, mobile and wireless handheld devices having relatively low computing capabilities are often connected to the digital home networks, so they can be vulnerable to several attacks such as eavesdropping, replay attack and other active attacks. Hence, it is essential to provide robust security mechanism in the digital home network. In the home control protocols, authentication and encryption should be considered as security functions [4]. Moreover, home network services may include remote healthcare monitoring that contains sensitive information. Unless home network is well protected, illegitimate users can easily access such home services. Thus user authentication is one of the key security mechanisms required for the remote access control in the digital home networks.

Remote user authentication mechanisms have been extensively developed. And password authentication is regarded as one of the simplest and the most convenient authentication mechanisms because it has the benefits of low implementation cost and convenient to users [5]. To prevent direct wiretapping attacks in open network environments, many modern password authentication schemes use one-time passwords. One-time password (OTP) is one of the simplest and most popular forms of two-factor authentication for securing network access [6]. In recent days, OTP solutions including HMAC-based OTP (HOTP) [19] that use cryptographic processing to generate an unique password are appealing.

Currently, smart card based remote user authentication schemes have been widely adopted due to their low computational cost and convenient portability [9–11]. Password-based authentication with smart cards is one of the convenient and effective remote user authentication mechanisms. This technology has been widely deployed for various kinds of authentication applications including remote host login, on-line banking, e-health services.

* Corresponding author.
** Corresponding author.
E-mail addresses: bnvaidya@gmail.com (B. Vaidya), parkjonghyuk1@hotmail.com (J.H. Park), ssyeo@msn.com (S.-S. Yeo), joeljr@ieee.org (J.J.P.C. Rodrigues).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.03.013

Nevertheless, according to the researches in [27,28], the existing smart cards are vulnerable as sensitive verifier and secret values stored in the smart cards could be extracted by monitoring their power consumption. Thus weaknesses of the authentication schemes using smart card are mainly due to two problems. First, if an adversary obtains a legal user's smart card even without the corresponding password, he can use it to produce a fabricated login message, and then impersonate the user to pass the server's authentication. Secondly, if the adversary captures a server's secret key and smart card at the same time, he can easily impersonate the legitimate user to login the remote system. Due to above reasons, most of the existing schemes using smart card [23–25] are still vulnerable to stolen smart card attacks.

In this paper, we propose an authentication scheme based on strong-password approach to provide secure remote access in digital home network environment. It uses hashed one-time password technique and one-way hash functions to reduce the processing overhead. As a one-time password changes each time the user logs in, it is impossible to reuse the current password although it is captured by the attacker. The proposed scheme aims not only to provide mutual authentication, to avoid time synchronization and to discard password-verifier at the remote server but also to thwart the stolen smart card attacks and provide forward secrecy with lost smart card. Thus the proposed scheme using HMAC-based OTP (HOTP) algorithm, hash-chaining technique along with a smart card technology provides robust and efficient authentication mechanism. We have conducted formal verification of the proposed scheme as well as analysis in terms of security and functional requirements.

The rest of this paper is organized as follows: Section 2 presents theoretical background, while Section 3 discusses related work. Section 4 presents a proposed robust user authentication scheme for digital home networks whereas Section 5 provides protocol analysis and verification of the proposed scheme. Section 6 gives in-depth analysis of the proposed scheme and finally, Section 7 concludes the paper.

2. Theoretical background

This section discusses about the architecture of home network, HMAC-based one-time password (HOTP) and Non-monotonic cryptographic protocol (NCP).

2.1. Architecture of home network

In this subsection we give basic concept of a digital home networking. Digital home allows users to perform out-home accesses where the users can use mobile devices to control their home appliances as well as to obtain value-added services. A typical digital home network contains a home gateway, home appliances, mobile devices, an authentication server and service providers.

The concept of digital home is to remotely access and control the digital and electrical home appliances (home devices), for instance, televisions (TVs), personal computers (PCs), lights, refrigerators, and washing machines. And wireless and mobile devices are used to connect the home gateway and further control the home appliances by the residential users.

A home gateway (HG) plays an essential role in digital home networking. On one hand, it provides network connectivity to the various network terminals at home, interconnects the public network and the subnets of the home network, and implements the remote management. On the other hand, it enables users to utilize the value-added services provisioned by service providers on the Internet. It also provides access authentication and service security functions.

In digital home applications, an Integrated Authentication Server (IAS) exists outside the home network, which manages the home gateway, authenticates users, grants privileges, and controls accounting as the home gateway operator. Whereas service providers supply many kinds of services, such as e-health, music, and other network services, for residential users.

Fig. 1 shows the general architecture for the home networks.

2.2. HOTP: HMAC-based one-time password algorithm

HMAC-based one-time password algorithm (HOTP) [19] uses a monotonically increasing counter value representing the message in the HMAC computation. In order to create the HOTP value, the HMAC-SHA-1 algorithm [20] is used. In the HOTP algorithm, a static symmetric key is known only to the client and the server.

The operation of HOTP algorithm is as follows. Initially, a SHA-1 HMAC generator is initialized using an unique shared secret. Then the HMAC of the current counter, or moving factor, is computed. During dynamic truncation process, certain bytes are extracted from the HMAC. Finally, these bytes are taken modulo 10^n, where

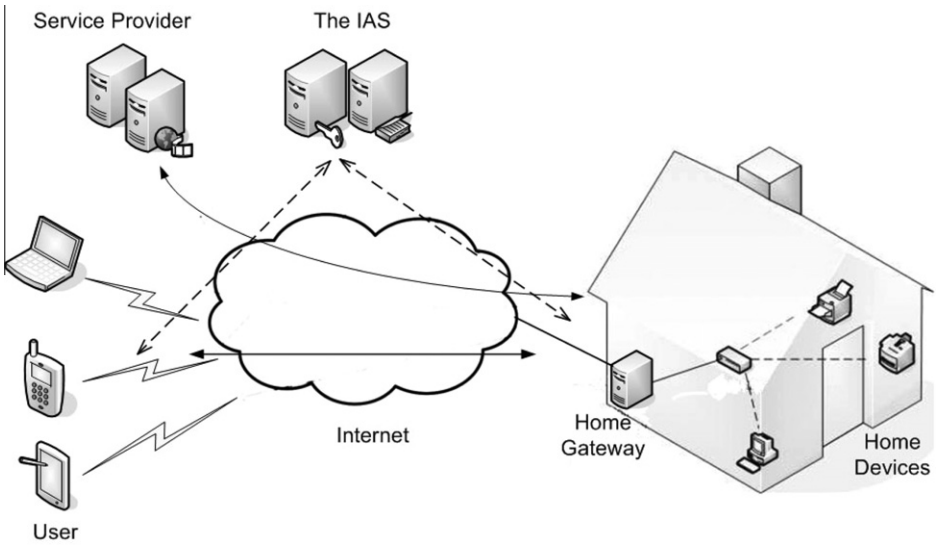

Fig. 1. Home network architecture.

n is the number of digits desired in the passcode, to produce the current passcode. The obtained HOTP value is as follows:

$$\text{HOTP}(K, C) = \text{Truncate}(\text{HMAC-SHA-1}(K, C))$$

where Truncate represents the function that converts, HMAC-SHA-1 value into HOTP value; and key (K), counter (C), and Data values are hashed high-order byte first.

Both the client and the server must generate the same passcode. Specifically, assuming that the server has already distributed the shared secret to the client, the client counter and the server counter must be synchronized.

2.3. Non-monotonic cryptographic protocols

Non-monotonic cryptographic protocols (NCP) [12,13] introduced by Rubin, describes the logic of authentication and the beliefs of various entities involved in a protocol as a consequence of communication. This protocol is also known as Rubin logic, in which there is no idealization step in specifying protocols. The specification of protocols, which is close to the actual implementation, is simply the starting point of the analysis. There are two sets in non-monotonic logic, which are applicable of analyzing a protocol. One is global to the protocol and other is local to each entity. The knowledge and belief sets for each principal are modified via actions and inference rules.

2.3.1. Global and local sets

Global set is public to each principal in a protocol specification. It contains Principal Set, Rule Set, Principal Set, Secret Set, and Observers Sets, which are defined in Appendix A.1.1. The first step of the specification of any protocol using Rubin logic is to instantiate the global sets with values.

Local sets are private to each principal in a protocol specification. For each principal, P_i, Rubin logic defines the following sets such as Possession Set, Belief Set, Behavior List, Seen, Haskeys, etc., that are defined in Appendix A.1.2.

2.3.2. Actions

Actions play vital role in the protocol specification, which describe how a principle constructs messages, encrypts and decrypts, compute functions, aborts a protocol, and performs any other operations. Actions can be used to control the knowledge and possessions of the entities in a protocol. The action lists that precede and follow message operations in a principal's behavior list determine sequence of events performed by the principal during a protocol run. Some actions that can be performed by P_i in the proposed protocol are derived from [12,13] (Refer to Appendix A.2) . And the following actions are recently defined as per our requirements.

1. HOTP$(\{Hash(h(\cdot); X)\}, k)$
   Condition: $h(\cdot), X, k \in$ POSS $(P_i), P_i \in$ Observers (k)
   Result: POSS(P_i) : = POSS$(P_i) \cup \{\{(h(X)\}_k\}$
   Description: This action is used to obtain HOTP operation.
2. HashOTP$(h(\cdot), N; X))$
   Condition: $h(\cdot), X, N \in$ POSS (P_i)
   Result: POSS(P_i) : = POSS$(P_i) \cup \{h^N(X)\}$
   Description: This action is used to obtain hash-chaining operation.
3. Increment (N, n). It means that counter N is increased by n step.
4. Decrement (C, n): It means a counter C is decreased by step n.
5. Replace (X_1, X_2): It means value of X_1 is substituted by value of X_2.

2.3.3. Inference rules

Inference rules are used to reason about beliefs during the execution of the protocol. These rules are applied whenever they are relevant.

X contains Y: Y appears as a sub message of X .
S:= f (S): S is replaced by the value f (S).
X from E: X is received from E.
LINK(N) : to link challenge and response.

The useful inference rules are shown in Appendix A.3. Furthermore, a complete list of inference rules are found in [12].

3. Related work

Password-based authentication scheme that was first introduced in [14] is the most widely used method for remote user authentication. In order to meet today's security requirements, many password authentication methods using one-time password have been proposed [15,23,21,22,18,24]. The purpose of a one-time password (OTP) is to make it more difficult to gain unauthorized access to restricted resources. Existing one-time password authentication schemes can be weak-password approach or strong-password approach. In case of strong-password approach, password is with high entropy, thus it cannot be guessed easily for adversary. Furthermore, the computational load of most strong-password authentication schemes is relatively lightweight due to use of simple operations, e.g., one-way hash function and exclusive-OR operation.

Furthermore, due to the fact that a smart card has a capability of simple computation, highly storage capacity, and easy to carry, large number of password authentication schemes with the smart cards [7–11], have been proposed. In many cases, the server stores authentication parameters, even sensitive verifier, into a smart card, and issues it to a user.

In 2002, Yeh et al. [15] proposed a one-time password authentication scheme using smart cards. This scheme is an enhancement of S/KEY [6] and was claimed to be free from any of server spoofing attack, preplay attacks, and off-line dictionary attacks. However, Tsuji and Shimizu [16] and Ku et al. [17] later respectively showed that Yeh et al.'s scheme is vulnerable to stolen-verifier attacks. In 2005, Lee and Chen [21] proposed an improvement of Yeh et al.'s scheme. Lee and Chen claimed that their improvement can effectively withstand the stolen-verifier attack and is as efficient as Yeh et al.'s scheme. Jo and Youn [23] proposed a secure user authentication protocol based on one-time password technique for home network. It employs a three-way challenge-response handshake technique to provide mutual authentication. In 2006, You and Jung [22] proposed a light-weight authentication protocol for digital home networks to improve Lee–Chen scheme [21]. Most parts of this scheme are the same as those of Lee–Chen scheme. Especially, the proposed method can withstand an attack, called the compromise of pass session keys via stolen passwords. However, these schemes have a number of security flaws including stolen smart card attacks.

In [24], a new user authentication (UA) scheme based on OTP protocol using smart cards for home networks was proposed, which not only can protect against illegal access for home services but also does not allow unnecessary service access by legitimate users. And Kim and Chung [25] proposed the modified version of Yoon–Yoo's scheme [26], which can overcome leak of password and impersonation attacks while mentioning merits of Yoon–Yoo's scheme. However, these schemes also have some flaws including lost smart card problems.

In this paper, we propose a user authentication scheme based on strong-password approach to provide secure remote access in home network environment. It uses HMAC-based OTP algorithm, and hash-chaining technique along with low-cost smart cards, which is more secure and robust than the existing schemes. The proposed scheme can thwart the stolen smart card attacks and provide forward secrecy with lost smart card as well as provide mutual authentication, avoid time synchronization and discard password-verifier at the remote server.

4. Robust user authentication scheme for home network environment

In this section, we propose a robust and efficient password authentication scheme based on strong-password approach to furnish secure remote access in digital home networks. It uses HOTP algorithm, and hash-chaining technique along with smart card. The security of our scheme depends on the secure HMAC function, hash-chaining technique and encryptions. The hashed one-time password is used not only to avoid replay attack and the serious time synchronization problem but also to offer more secure solution than the standard one-time password. Table 1 shows the notations used for this scheme.

The proposed scheme consists of registration phase, login/authentication phase, service request phase and password change phase. The user registers to Integrated Authentication Server (IAS) through the secure channel, and the IAS securely issues smart card to the user. During the registration phase, the user can select their own password. In this scheme, the IAS does not store a password table nor hashed password for the user. During login and authentication phase, mutual authentication between the user and the IAS is obtained with HOTP operation, while during service request phase, mutual authentication between the user and the home gateway (HG) is accomplished with hash-chaining technique. As the proposed scheme uses lightweight computation modules including hashed one-time password, one-way hash function, exclusive-OR operation and moderately inexpensive symmetric encryption technique, it has relatively low computational overhead.

It is assumed that IAS is located on the outside of the home network environment, manages the home gateway, and performs authentication, authorization, and accounting (AAA) functions. Service subscribers require security association between IAS and HG, in order to provide home network services. In addition, they must be able to operate service access control when privileged services are granted. To provide robust authentication between HG and users, the authentication ticket granted by the IAS and hash-chaining mechanism are used. The users can use home services until the validity of the authentication ticket is not expired so that they do not have to login each time to the IAS. And the permitted number of access is also used to limit the number of access within the authentication ticket's validity. Fig. 2 illustrates the proposed user authentication scheme.

Table 1
Notations used for the proposed scheme.

Notation used	Description
ID_C	User's Identifier
ID_{IAS}	IAS's identifier
ID_{SC}	Smart card's identifier
PW	User's password
x	Secret key maintained by the IAS
$F(\cdot),\ h(\cdot)$	One-way hash function
$F^n(S)$	Hash chaining function value at nth with S
$\oplus$	XOR operation
$\|$	Concentration
$H^i_{(K,C)}$	ith HMAC-Based One-Time Password
C_X	8-byte counter value, the moving factor (C – client, S – server, M – Max allowed)
K	secret shared between client and server
$K_1,\ S_K$	Session keys
K_{IAS-HG}	Symmetric key between IAS and HG
N	Permitted number of access
S	Random seed
$E_{K_X}(M)$	Encryption of message M with K_X
T_{exp}	Expiry time for authentication ticket

For the proposed user authentication scheme for home network environments, the following assumptions have been considered:

- The algorithm is counter-based, that means the HOTP algorithm embedded in smart cards.
- The algorithm uses a strong shared secret. The length of the shared secret must be at least 128 bits and preferably of 160 bits.
- Each HOTP generator has a different and unique secret K shared between client and server.
- The 8-byte counter must be synchronized between the HOTP generator (client) and the HOTP validator (server).
- HOTP algorithm uses resynchronization mechanism as mentioned in [19].
- The IAS has established the security association with home gateway server (HG) using symmetric key K_{IAS-HG}.

4.1. Registration phase

In this phase, the user (U) needs to submit its identity and PW to the remote system for registration. The complete registration process is shown as follows.

Step R1
User (U) chooses ID_C, PW, and sends them over a secure communication channel to the remote integrated authentication server (IAS).
Step R2
IAS does the followings:
– Generate secret K
– Compute
$v_T = h(ID_C \oplus x) \oplus h(PW) \oplus K,\ \ g_T = h(ID_C \| x \| K) \ \ \oplus h(ID_C \| PW)$, and

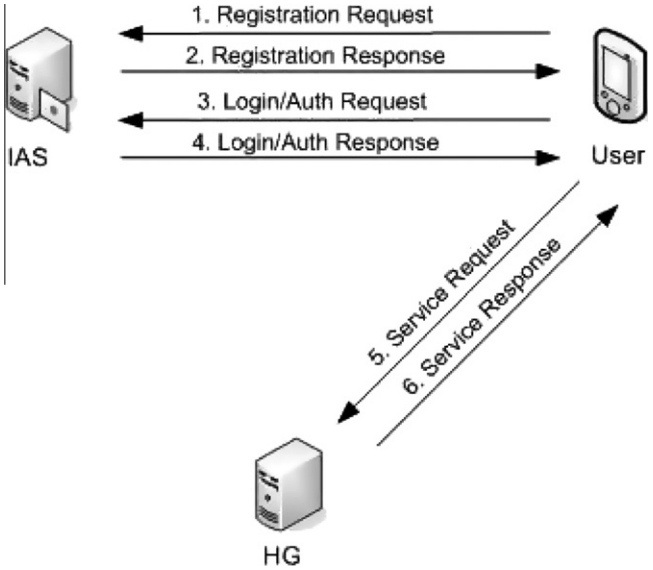

Fig. 2. Proposed user authentication scheme for home networks.

$k_T = K \oplus h(PW \oplus h(PW))$

- Store ID_C, ID_{SC} in user database
- Write $\{ID_C, ID_{SC}, h(\cdot), v_T, g_T, k_T, C_M\}$ to a smart card

Step R3

The IAS will issue the smart card securely to the user (U).

4.2. Login/authentication phase

In this phase, the user U has to send a login request message to the remote IAS server, whenever U wants to log in. The login/authentication procedure is as follows.

Step LA1

U must insert his smart card (SC) into the terminal and input ID_C and PW.

Step LA2

SC performs as follows:

- Verify ID_C. If ID_C is identical to the ID_C in SC, it will then proceed the login procedure.
- Derive $K = k_T \oplus h(PW \oplus h(PW))$, and

  $h(ID_C \oplus x) = v_T \oplus K \oplus h(PW)$
- Compute $u_T = K \oplus h(ID_C \oplus x)$, and

  $a_T = h(ID_{SC}\|K) \oplus h(ID_C\|PW)$
- Generate current HOTP $H^i_{(K,C_C)}$ = HOTP $(K, C_C, h(ID_C\|PW))$
- Increase its counter C_C by 1
- Compute $G = h(u_T \oplus g_T) \oplus H^i_{(K,C_C)}$

Step LA3

The U sends $\{ID_C, u_T, a_T, G\}$ to the IAS server.

Step LA4

IAS performs as follows:

- Verify ID_C. If it is not a valid user identity, login request is rejected.
- Derive $K = u_T \oplus h(ID_C \oplus x)$, and

  $h(ID_C\|PW) = h(ID_{SC}\|K) \oplus a_T$
- Compute $g'_T = h(ID_C\|x\|K) \oplus h(ID_C\|PW)$
- Obtain $H^i_{(K,C_C)} = h(u_T \oplus g'_T) \oplus G$
- Generate HOTP $H^i_{(K,C_S)}$ = HOTP$(K, C_S, h(ID_C\|PW))$
- Compare $H^i_{(K,C_C)} = H^i_{(K,C_S)}$If it is true, then increase its counter C_S by 1
- Obtain $K_1 = H^i_{(K,C_S)}$
- Generate a random number N_a, and

  $S_K = h(K_1\|N_a)$
- Compute $A_S = h(S_K\|ID_C)$
- Encrypt $E_{K_1}(ID_C, ID_{IAS}, N_a, A_S, N, S)$ using K_1, and

  $E_{K_{IAS-HG}}(ID_C, ID_{IAS}, N_a, K_1, N, S, T_{exp})$

  using K_{IAS-HG}

Step LA5

The IAS sends authentication response ($AuthResp$)

$E_{K_1}(ID_C, ID_{IAS}, N_a, A_S, N, S)$, and authentication ticket (TKG)

$E_{K_{IAS-HG}}(ID_C, ID_{IAS}, N_a, K_1, N, S, T_{exp})$ to the user (U)

.

Step LA6

U performs as follows:

- Decrypt $E_{K_1}(ID_C, ID_{IAS}, N_a, A_S, N, S)$ using $K'_1$where $K'_1 = H^i_{(K,C_C)}$
- Derive $S'_K = h(K'_1\|N_a)$
- Compute $A'_S = h(S'_K\|ID_C)$
- Check if $A'_S = A_S$ to verify the $AuthResp$

4.3. Service request phase

In order to use home services, the authenticated users can request services to the home gateway (HG). The service request phase is shown as follows.

Step SR1

U calculates an initial one-time password value $P_0 = F^N(SK \oplus S)$. This is done only once. U will save P_0 for further use.

Step SR2

When home service is required, U sends

$ID_C, E_{K_{IAS-HG}}(ID_C, ID_{IAS}, N_a, K_1, N, S, T_{exp})$ to HG.

Step SR3

HG performs as follows:

- Decrypt $E_{K_{IAS-HG}}(ID_C, ID_{IAS}, N_a, K_1, N, S, T_{exp})$ using K_{IAS-HG}
- Verify ID_C with that in TKG, and T_{exp} if it is expired.
- Derive $S_K = h(K_1\|N_a)$. This is done once only.
- Compute $P_0 = F^N(S_K \oplus S)$. This is done once only.
- Decrease C by 1 to set value of a counter C same as N
- Compute $R = h(S \oplus C \oplus S_K)$
- Compute $PP = P_0 \oplus h(S_K)$

Step SR4

HG sends C, R, PP to U.

Step SR5

U performs as follows:

- Compute $R' = h(S \oplus C \oplus S_K)$, and

  $PP' = P_0 \oplus h(S_K)$.
- Verify $R' = R$, and $PP' = PP$

  If both are true, then compute $P_i = F^{(N-i)}(S_K \oplus S)$ at the ith time.
- Save C
- Replace P_{i-1} by P_i

Step SR6

The U will send $P_i \oplus h(S_K)$ to the HG.

Step SR7

HG performs as follows:

- Obtain P_i from the received message
- Verify $F(P_i) = F(F^{(N-i)}(SK \oplus S)) = F^{(N-i+1)}(SK \oplus S) = P_{i-1}$

  If it is true then replace P_{i-1} by P_i.

Two parameters N and T_{exp} can be used to limit the user access. For instance, $N = 100$ and $T_{exp} = 1$ mth then the valid user can access the home gateway 100 times within 1 month.

4.4. Password change phase

This phase is invoked only when the user desires to change his password. The password change phase is as follows.

Step P1

The user U submits ID_C, PW, PW_N, where PW_N is new password to the smart card (SC).

Step P2

SC performs as follows:

- Check the ID_C if it is valid. Otherwise it will reject.
- Derive $K = k_T \oplus h(PW \oplus h(PW))$, and $h(ID_C \oplus x) = v_T \oplus K \oplus h(PW)$
- Compute $u_T = K \oplus h(ID_C \oplus x)$, and $a_T = h(ID_{SC}\|K) \oplus h(ID_C\|PW)$
- Encrypt $E_K(PW, PW_N)$ using K

Step P3

The user sends $\{ID_C, u_T, a_T, E_K(PW, PW_N)\}$ to the IAS.

Step P4
IAS performs as follows:
- Check the ID_C if it is valid. Otherwise it will reject.
- Compute $u_T \oplus h(ID_C \oplus x)$
- Decrypt $E_K(PW, PW_N)$ using K
- Compute $a'_T = h(ID_{SC}\|K) \oplus h(ID_C\|PW)$
- Verify if a'_T is same as received a_T. If not, it will reject.
- Compute $c_{T_N} = h(ID_C\|PW_N)$, $v_{T_N} = h(ID_C \oplus x) \oplus K \oplus h(PW_N)$, $g_{T_N} = h(ID_C\|x\|K) \oplus c_{T_N}$, and $k_{T_N} = K \oplus h(PW_N \oplus h(PW_N))$
- Encrypt $E_K(ID_C, c_{T_N}, v_{T_N}, g_{T_N}, k_{T_N})$ using K

Step P5
The IAS sends response $E_K(ID_C, c_{T_N}, v_{T_N}, g_{T_N}, k_{T_N})$ to the user U.

Step P6
U performs as follows:
- Decrypt $E_K(ID_C, c_{T_N}, v_{T_N}, g_{T_N}, k_{T_N})$ using K
- Compute $c'_{T_N} = h(ID_C\|PW_N)$
- Verify if c'_{T_N} is same as received c_{T_N}. Otherwise it will reject.
- Replace v_T, g_T, k_T with $v_{T_N}, g_{T_N}, k_{T_N}$ in the SC

5. Protocol analysis and verification

We analyze the proposed scheme using non-monotonic cryptographic protocol (Rubin logic). The method integrates protocol analysis with specification and thus, beliefs and state of knowledge is upgraded as the protocol run is progressed. Only registration phase, login/authentication phase, service request phase are accounted for protocol analysis.

5.1. Protocol specification

Before proceeding with protocol specification, it should be clear that notations used for the protocol specification is same as those in Table 1. Additional notations used for the protocol specification are shown in Table 2.

The specification of the protocol is explained as follows.

Global Sets: The Global set consists of following sets:

A. Principal Set: P = U, RS, HG. U is the initiator of protocol.
B. Rule Set: Inference rules defined in Section 2.3.3.
C. Secret Set: $\{PW, x, K_{RS-HG}\}$
D. Observers Set:
Observers(PW): {U}
Observers(x): {RS}
Observers(K_{RS-HG}): {RS, HG}

Local Sets: The Local set consists of U, RS, and HG.

Entity U
POSS(U) = $\{PW, \{ID_C\}\}$
BEL(U) = $\{\sharp(PW)\}$
BL(U) =

I phase

- Send(RS, $\{ID_C, PW\}$)
- Update($\{ID_C, PW\}$)
- Receive(RS, $\{ID_C, ID_{SC}, h(\cdot), v_T, g_T, k_T, C_M\}$)

II phase

- XOR(k_T, Hash($h(\cdot)$; XOR(PW, Hash($h(\cdot)$; PW)))) $\rightarrow K$
- XOR(v_T, K, Hash($h(\cdot)$; PW)) $\rightarrow J$
- XOR(K, J) $\rightarrow u_T$
- Hash($h(\cdot)$; Concat(ID_C, PW) $\rightarrow c_T$
- XOR(c_T, Hash($h(\cdot)$; Concat(ID_{SC}, K)) $\rightarrow a_T$
- HOTP$\{Hash(h(.); C_C, c_T)\}, K) \rightarrow H^i_{(K,C_C)}$
- Increment(C_C, 1)
- XOR($H^i_{(K,C_C)}$, Hash($h(\cdot)$; XOR(u_T, g_T))) $\rightarrow G$
- Send(RS, $\{ID_C, u_T, a_T, G\}$)
- Update($\{ID_C, u_T, a_T, G\}$)
- Receive (RS, $\{AuthResp, TKG\}$)
- Split ($\{AuthResp, TKG\}$)
- Decrypt ($\{ID_C, ID_{RS}, N_a, N, S, A_S\}_{K_1}, K'_1$)
where $K'_1 = H^i_{(K,C_C)}$
- Split($\{ID_C, ID_{RS}, N_a, N, S, A_S\}$)
- Hash($h(\cdot)$; Concat(K'_1, N_a)) $\rightarrow S'_K$
- Hash($h(\cdot)$; Concat(S'_K, ID_C)) $\rightarrow A'_S$
- Check(A'_S, A_S)

III phase

- HashOTP($h(\cdot), N$; XOR(S_K, S, C)) $\rightarrow P_0$
- Send(HG, $\{ID_C, TKG\}$)
- Update ($\{ID_C, TKG\}$)
- Receive($HG, \{C, R, PP\}$)
- Hash($h(\cdot)$; XOR(S_K, S)) $\rightarrow R'$
- XOR(P_0, Hash($h(\cdot)$; S_K)) $\rightarrow PP'$
- Check(R', R)
- Check(PP', PP)
- HashOTP($h(\cdot), (N-i)$; XOR(S_K, SD)) $\rightarrow P_i$
- Replace(P_{i-1}, P_i)
- Send(HG, {XOR(P_i, Hash($h(\cdot)$; S_K))})
- Update({XOR(P_i, Hash($h(\cdot)$; S_K))})

Entity RS
POSS(RS) = $\{x, K_{RS-HG}, \{ID_{RS}\}\}$
BEL(RS) = $\{\sharp(x), \sharp(K_{RS-HG})\}$
BL(RS) =

I phase

- Receive(U, $\{ID_C, PW\}$)
- Generate-secret(K)
- XOR(K, Hash($h(\cdot)$; PW), Hash($h(\cdot)$; XOR(ID_C, x))) $\rightarrow v_T$
- Hash($h(\cdot)$; Concat(ID_C, PW)) $\rightarrow c_T$
- XOR(c_T, Hash($h(\cdot)$; Concat(ID_C, x, K))) $\rightarrow g_T$
- XOR(K, Hash($h(\cdot)$; XOR(PW, Hash($h(.)$; PW))) $\rightarrow k_T$
- Send(U, $\{ID_C, ID_{SC}, h(\cdot), v_T, g_T, k_T, C_M\}$)
- Update($\{ID_C, ID_{SC}, h(\cdot), v_T, g_T, k_T, C_M\}$)
- Forget($\{v_T, g_T, k_T, K, PW\}$)

II phase

- Receive(U, $\{ID_C, u_T, a_T, G\}$)
- Split($\{ID_C, u_T, a_T, G\}$)
- Check($\{ID_C\} \in$ U, $\{ID_C\} \in$ RS)
- XOR(u_T, Hash($h(\cdot)$; XOR(ID_C, x)) $\rightarrow K$
- XOR(a_T, Hash($h(\cdot)$; Concat(ID_{SC}, K))) $\rightarrow c_T$
- XOR(c_T, Hash($h(\cdot)$; Concat(ID_C, x, K)) $\rightarrow g'_T$
- XOR(G, Hash($h(\cdot)$; XOR(u_T, g'_T))) $\rightarrow H^i_{(K,C_C)}$
- HOTP({Hash($h(\cdot)$; C_S, c_T}, K) $\rightarrow H^i_{(K,C_S)}$
- Check($H^i_{(K,C_C)}, H^i_{(K,C_S)}$)
- Increment(C_S, 1)
- Generate-nonce(N_a)
- Hash($h(\cdot)$; Concat(K_1, N_a)) $\rightarrow S_K$where $K_1 = H^i_{(K,C_S)}$
- Hash($h(\cdot)$; Concat(S_K, ID_C)) $\rightarrow A_S$

Table 2
Additional notations used for the protocol specification.

Notation used	Description
U	User
RS	Remote server such as IAS
HG	Home Gateway
K_{RS-HG}	Symmetric key between RS and HG
$AuthResp$	Authentication Response
TKG	Authentication Ticket
$X_1 \rightarrow X_2$	X_1 is replaced by X_2
I phase	Registration phase
II phase	Login/Authentication phase
III phase	Service Request phase

- Encrypt($\{ID_C, ID_{RS}, N_a, N, S, A_S\}, K_1$) → *AuthResp*
- Encrypt($\{ID_C, ID_{RS}, N_a, N, S, K_1, T_{exp}\}, K_{RS-HG}$) → *TKG*
- Send (U, {*AuthResp*, *TKG*})
- Update({*AuthResp*, *TKG*})

Entity HG

POSS(HG) = $\{K_{RS-HG}, \{ID_C\}\}$
BEL(HG) = $\{\sharp(K_{RS-HG})\}$
BL(HG) =

III phase

- Receive (U, $\{ID_C, TKG\}$)
- Decrypt($\{ID_C, ID_{RS}, N_a, N, S, K_1, T_{exp}\}_{K_{RS-HG}}, K_{RS-HG}$)
- Split($\{ID_C, ID_{RS}, N_a, N, S, K_1, T_{exp}\}$)
- Check($ID_C, \{ID_C\} \in TKG$)
- Check-freshness(T_{exp})
- Hash($h(\cdot)$; Concat(K_1, N_a)) → S_K
- HashOTP($h(\cdot), N$; *XOR*(S_K, S)) → P_0
- Decrement($C, 1$)
- Hash($h(\cdot)$; XOR(C, S_K, S)) → R
- XOR(P_0, Hash($h(\cdot); S_K$)) → PP
- Send (U, $\{C, R, PP\}$)
- Update($\{C, R, PP\}$)
- Receive(U, {XOR(P_i, Hash($h(\cdot); S_K$))})
- XOR(P_i, Hash($h(\cdot), S_K$), Hash($h(\cdot); S_K$)) → P_i
- Check(Hash($h(\cdot)$; HashOTP($h(\cdot), (N-i)$; XOR(S_K, S))), HashOTP($h(\cdot), (N-i+1)$; XOR(S_K, S)))
- Replace(P_{i-1}, P_i);where
  P_{i-1} = HashOTP($h(\cdot), (N-i+1)$; XOR(S_K, S))

5.2. Analysis and verification using Rubin logic

In this subsection, we discuss the analysis of the proposed protocol using Rubin logic. We will analyze three phases of the proposed protocol, namely, I, II, and III phases separately.

In the I phase, actions in BL(U) are executed first as the U is the initiator of the protocol. The first two actions in BL(U) are executed. After the Update action, the next action to be executed is in RS's behavior list because the Send action specifies RS.

Then the actions of BL(RS) in the I phase are executed and with the Update action, the next action to be executed is in U's behavior list as the Send action specifies U. With Forget statement, RS removes $\{v_T, g_T, k_T, K, PW\}$ from its POSS(RS) and $\sharp(PW)$ from BEL(RS).

At the end of I phase, there are no changes in global sets, however local sets of U and RS will be changed as follows.

POSS(U) = $\{PW, \{h(\cdot), v_T, g_T, k_T, C_M, \{ID_C, ID_{SC}\}\}$ from RS}
BEL(U) = $\{\sharp(PW)\}$
POSS(RS) = $\{x, K_{RS-HG}, \{ID_{RS}, ID_C, ID_{SC}\}\}$
BEL(RS) = $\{\sharp(x), \sharp(K_{RS-HG})\}$

When II phase begins, the U will be the initiator. The first ten actions of BL(U) are executed. After completion of Update action, the following changes will be occur in the entity U.

POSS(U) = $\{PW, K, H^i_{(K,C_C)}, C_C, u_T, a_T, G,$
$\{h(\cdot), v_T, g_T, k_T, \{ID_C, ID_{SC}\}\}$ from RS}
BEL(U) = $\{\sharp(PW), \sharp(K), LINK(H^i_{(K,C_C)})\}$

The global set will change as follows:

Observers(K) : {U}

There are still no relevant inference rules applied.
The next action is in RS's BL as follows.

Receive(U, $\{ID_C, u_T, a_T, G\}$)

Then the execution of rest of the actions of BL(RS) in II phase are occurred. After applying Update action, the following changes occur in the entity RS.

POSS(RS) = $\{x, K, K_1, S_K, K_{RS-HG}, H^i_{(K,C_S)}, \{u_T, a_T, G\}$ from U,
$\{ID_{RS}, ID_C, ID_{SC}\}, \{(ID_C, ID_{RS}, N_a, N, S, A_S)\}_{K_1}$,
$\{(ID_C, ID_{RS}, N_a, K_1, N, S, T_{exp})\}_{K_{RS-HG}}\}$
BEL(RS) = $\{\sharp(x), \sharp(K), \sharp(K_1), \sharp(K_{RS-HG})$, LINK$\left(H^i_{(K,C_S)}\right)\}$

At this point, the conditions for the Linkage rule are satisfied. Once the linkage rule is applied, the freshness of sub message is added to belief set of RS. And LINK is removed from the belief set so that $\left(H^i_{(K,C_S)}\right)$ cannot be used again.

Now the global sets will also change as follows.

Observers(K) : {U, RS}
Observers (K_1) : {RS}
Observers(S_K) : {RS}

The next action in U's BL is executed as the Send action specifies U.

The last eight actions of BL(U) in II phase are executed. Check action will verify A'_S and A_S.

Again, the conditions for the linkage rule are satisfied. Once the linkage rule is applied, the freshness of sub message is added to belief set of U. And LINK is removed from the belief set so that $\left(H^i_{(K,C_C)}\right)$ cannot be used again.

At the end of the II phase, the global sets will be as follows.

Observers(K) : {U, RS}
Observers(K_1) : {U, RS}
Observers(S_K) : {U, RS}

In III phase, the U is the initiator. First three actions in BL(U) in III phase are executed. After Update action is executed, a next action in BL(HG) is executed. Then the first twelve actions in BL(HG) are executed. Upon the execution of the Update action, a next action will execute in BL(U). Now POSS(HG) has S_K and BEL(HG) contains $\sharp(S_K)$, and LINK(P_{i-1}).

Then the last eight actions in BL(U) are executed. Upon the execution of the Update action, a next action in BL(HG) will be executed. At this moment, BEL(U) contains LINK(P_{i-1}). And at this point, the conditions for the Linkage rule are satisfied. Once the linkage rule is applied, the freshness of sub message is added to belief set of U. And LINK is removed from the belief set so that (P_{i-1}) cannot be used again.

Finally, last five actions in BL(HG) in III phase are executed. With Check action, the HG will verify $h(P_i)$ with P_{i-1}. At this moment, the conditions for the Linkage rule are satisfied. Once the linkage rule is applied, the freshness of sub message is added to belief set of HG. And LINK is removed from the belief set so that (P_{i-1}) cannot be used again.

The Global sets have now changed and finally contained the followings.

Observers(K) : {U, RS}
Observers(K_1) : {U, RS}
Observers(S_K) : {U, RS, HG}

All the actions of the protocol specification have been executed and it can be concluded that

- Session key S_K is shared by U and HG.
- Session key K_1 and shared secret key K are shared by U and RS.
- The entity U is mutually authenticated with RS and HG.

6. Analysis of proposed scheme

This section provides in-depth analysis of the proposed scheme in terms of security requirements as well as functional requirements. Our scheme not only satisfies several current security requirements as defined in [5] but also fulfills functional requirements.

6.1. Analysis of security requirements

In this subsection, security analysis of the proposed scheme is discussed. The security requirements for an ideal password authentication scheme defined in [5] should be satisfied in order to achieve robust and efficient user authentication mechanism. Thus we provide in-depth analysis of the proposed scheme in terms of most predominant security properties.

S1. Password Guessing Attacks: Some passwords having low entropy is vulnerable to password guessing attacks, where an attacker intercepts authentication messages and stores them locally and then attempts to use a guessed password to verify the correctness of his guess using these authentication messages. In our scheme, if the attacker intercepts a login request message $\{ID_C, u_T, a_T, G\}$, it cannot guess the password PW from u_T and a_T because it does not know the server's secret key x, and shared secret K.

S2. Eavesdropping attack: A host is configured to listen to and capture data not belonging to it. Carefully written eavesdropping programs can take usernames and passwords when users first login to the network. Broadcast networks like Ethernet and wireless local area network (LAN) are especially vulnerable to this type of attack. Our scheme can resist eavesdropping attack as all the important messages such as the authentication response, and authentication ticket are encrypted with the K_1, and symmetric key respectively.

S3. Replay attack: Having intercepted previous communications, an attacker can impersonate the legal user to login to the system. The attacker can replay the intercepted messages. An attack in which a valid data transmission is maliciously or fraudulently repeated either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack. Our scheme can resist a replay attack as HMAC-based OTP is used for authenticity of the authentication messages. Furthermore, during service request phase also, hash-chaining technique is used to avoid the replay attack.

S4. Forged User Attacks: An attacker attempts to modify intercepted communications to masquerade the legal user and login to the system. If the attacker tries to impersonate the user, it must be able to forge a valid login message $\{ID_C, u_T, a_T, G\}$. However, it is not possible to compute u_T and a_T without knowledge of secret key x, PW and K, and similarly G cannot be computed without g_T. Thus the proposed scheme resists the forged user attack.

S5. Masquerading Server Attacks: If the masquerading server attempts to cheat the requesting user, it has to forge a valid authentication response message. However, this is not feasible, as the attacker cannot compute K_1 without the knowledge of K and $H^i_{(K,C_S)}$.

S6. Man-in-the-middle attack: An attacker intercepts and modifies the messages between two parties with a malicious intent without either party knowing that the link between them has been compromised. In the proposed scheme, the attacker may alter the login message $\{ID_C, u_T, a_T, G\}$ into $\{ID_C, u^*_T, a^*_T, G^*\}$. However, this malicious attempt will not be successful, because such a modification will fail during verification process in Step LA4 at Section 4.2. And also the messages are encrypted so the adversaries cannot modify the message.

S7. Parallel Session Attacks: Without knowing a user's password, an attacker can masquerade as the legal user by creating a valid login message out of some eavesdropped communication between the user and the server. The attacker may launch a parallel attack by replaying the server's response message as the user's login message at a later time. However, this attack is not possible in the proposed scheme, as the authentication response message is well protected by K_1.

S8. Denial of Service (DoS) Attacks: The denial of service (DoS) attack prevents or inhibits the normal use or management of communications facilities. This attack may act on a specific user; for instance, an adversary may cause the server to reject the logins of a specific user until re-registration. The DoS attack rejects all or specific users by means of an offensive action on the server or by means of a falsification of user's password-verifier. Then the attacker can inconvenience the user but cannot imitate the user. Password change protocols allow an authenticated user to change his password, however, this protocol is very vulnerable to DoS attacks. As there is no password-verfier stored at the remote server, the adversary cannot update false verification information of the legal user. Further, suppose the adversary intercepts the password change request message and replace with $\{ID_C, u^*_T, a^*_T, E_K(PW, PW_N)\}$ before transmitting to the remote server. This attack cannot be successful in our scheme because the verification process in the Step P4 will not be successful. Thus our scheme can thwart the DoS attacks.

S9. Stolen-verifier Attacks: In most applications the server stores hashed passwords instead of clear text passwords. The stolen-verifier attack means that an adversary who steals the password-verifier (e.g., hashed passwords) from the server can use it directly to masquerade as a legitimate user during the user authentication phase. In the proposed scheme, the server does not store any password-verifier (verification table) so our scheme is resist to stolen verifier attack.

S10. Secret key Forward Secrecy: It ensures that the previously generated passwords in the system are secure even if the system's secret key has been revealed in public by accident or is stolen. In the proposed scheme, even if the server's secret key x happens to be revealed, the attacker would not be able to compute v_T and g_T without the knowledge of user's PW and K.

S11. Smart Card Loss Attacks: When the smart card is lost or stolen, unauthorized users can easily change the password of the smart card, or can guess the password of the user by using password guessing attacks, or can impersonate the user to login to the system. If the attacker tries to impersonate the user with stolen smart card to login to the system, he must be able to produce a valid login message $\{ID_C, u_T, a_T, G\}$. However, it is impossible to derive K and $h(ID_C \oplus x)$ from the smart card without knowing PW, thus the attacker cannot produce valid login message. Thus the proposed scheme can resist smart card loss attacks.

S12. Forward Secrecy with Lost Smart Card: Suppose the server's secret key x is revealed and if the attacker tries to get passwords or other login information from the stolen smart card, it can easily impersonate the user to login to the system. Thus the scheme should be capable to provide forward secrecy even if the smart card is lost or stolen. The proposed scheme can provide forward secrecy with lost smart card as the adversary cannot compute u_T or a_T without knowing user's PW.

Table 3
Comparison among the existing representative authentication schemes in terms of security properties (Y – Yes; N – No).

Auth. scheme	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
Jo–Youn scheme [23]	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
Jeong et al.'s scheme [24]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N
Kim–Chung scheme [25]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N
Proposed scheme	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

The comparison among the existing representative authentication schemes with the proposed scheme in terms of security properties is shown in Table 3. From Table 3, it can be seen that Jo-Youn scheme [23] cannot not only resist against stolen smart card attacks but also provide secret key forward secrecy. Suppose the smart card having $\{K, SEED, ID\}$ is stolen, an unauthorized user easily impersonate the user to login to the system. The adversary can send valid ID to the server, which will then send authentication data $\{N_i, SEED \oplus T_i, P_i \oplus T_i \oplus N_i)\}$ to the former. So the adversary can easily generate and send valid $\{N_{i+1}, \alpha, \beta\}$ to the server for authentication process. Furthermore, as the secret key K is stored in the smart card, this scheme cannot provide forward secrecy.

Jeong et al.'s scheme [24] cannot provide forward secrecy when the smart card is stolen. When x is leaked from the IAS, the adversary can compute $v_i = h(U_{ID}, x)$ with stolen smart card having $\{h(\cdot), e_i\}$. And knowing v_i and e_i, the adversary can easily calculate $F_N(password) = v_i \oplus e_i$.

In case of Kim–Chung scheme [25], it cannot provide forward secrecy when the smart card is stolen. Assuming x is revealed and with stolen smart card having $\{K_1, K_2, R, h(\cdot)\}$, the adversary can compute $h(PW) = R \oplus K_1$ and $N = K_1 \oplus h(ID \oplus x)$. Then knowing N, it can obtain $h(PW \oplus h(PW)) = K_2 \oplus h(ID \oplus x \oplus N)$. Thus even without knowledge of PW, the adversary can compute $C_1^* = R \oplus h(PW)$ and $C_1' = K_2 \oplus h(PW \oplus h(PW))$, then $C_2^* = h(C_1' \oplus T_1^*)$. So the adversary can impersonate the user and send login message $\{ID, T_1^*, C_1^*, C_2^*\}$ to the server, which will be passed during the verification.

Our scheme can satisfy all above-mentioned security requirements. It can be said that the proposed scheme offers robust authentication mechanism and better security properties. Thus our scheme is more secure and robust than the existing representative schemes.

6.2. Analysis of functional requirements

In this sub-section, we analyze the proposed scheme in terms of some of the following functional requirements of authentication schemes.

- **F1**. Freely chosen password by the users: Users should be able to choose their password freely. In our scheme, each user can choose his own password, not decided by the system.
- **F2**. No verification table: The remote system should not have a dictionary of verification tables such as clear-text passwords or hashed passwords to authenticate users. Our scheme does not require any storage of password verifiers.
- **F3**. No time synchronization: In time stamp-based authentication schemes [29,30], the clocks of the system and all users' computers must be synchronized with one another and the transmission delay time of the login message also has to be limited. The timestamp should be discarded to avoid serious time synchronization problem. To eliminate the requirement of clock synchronization and the limitation of transmission delay time, our scheme is based on counter and nonces instead of timestamps.
- **F4**. Mutual Authentication Mutual Authentication should be provided between the user and remote systems. Not only can the server verify the legal users, but the users should be able to verify the legal server. Mutual authentication can help withstand the server spoofing attack where an attacker pretends to be the server to manipulate sensitive data of the legal users. Any illegal server cannot cheat a user to log into its system without knowing (K, C) in the proposed scheme. Since it cannot obtain the correct corresponding HOTP for that particular user, the login process will be terminated by the user by verifying $A_S' = A_S$. Also the user will authenticate the HG by verifying $h(S \oplus C \oplus S_K)$ and $P_0 \oplus h(S_K)$. If only both these verifications are true, the user will further proceed the service request phase. Thus the user cannot be cheated by the illegitimate HG as well.
- **F5**. Secure Password Change: Users should be able to change their passwords after the registration phase. In our scheme, the user can securely change his password and update the information stored in the smart card.
- **F6**. Lower Computational cost: The computational cost should be lower as possible to obtain the efficient solution. Due to usage of one-time password technique such as HOTP algorithm, hash function, Exclusive-OR, the computational cost of the proposed scheme is still low.
- **F7**. Session key agreement: A session key agreed by the user and the remote system generated in every session. Session key changes with the change in HMAC-based OTP.

We compare the proposed scheme with existing representative schemes as per the above-mentioned features. Table 4 shows the comparison among existing representative schemes in terms of functional requirements. It can be seen that our scheme satisfies all above-mentioned functional requirements.

It can be seen that in Jo–Youn scheme [23], V_i is stored in the server, which may be used to impersonate as server. This scheme is time stamp-based, so it may confront severe time synchronization problem. Furthermore it has neither provision of changing password nor session key agreement.

In Jeong et al.'s scheme [24], $F_N(password)$ is stored in the server. Once an adversary has stolen the verifier, he could masquerade the corresponding server to get more information from the legitimate user.

Since Kim–Chung scheme [25] use time stamps, it has stringent requirement of clock synchronization and may have transmission delay time. This scheme does not have session key agreement.

Table 5 shows the efficiency analysis among existing representative schemes. And for convenience, registration, login, and authentication phases are considered for efficiency analysis.

It can been seen that our scheme requires more computational costs than the existing schemes to acquire better security. However, due to lightweight computation modules such as hashed one-time password, one-way hash function and exclusive-OR operation as well as relatively inexpensive symmetric encryption technique, the proposed scheme is still efficient.

Table 4
Comparison among some existing representative schemes in terms of functional requirements.

Auth. scheme	F1	F2	F3	F4	F5	F6	F7
Jo–Youn scheme [23]	Y	N	N	Y	N	Extremely low	NA
Jeong et al.'s scheme [24]	Y	N	Y	Y	Y	Low	Y
Kim–Chung scheme [25]	Y	Y	N	Y	Y	Low	NA
Proposed scheme	Y	Y	Y	Y	Y	Low	Y

Y – Yes; N – No; NA – Not applicable

Table 5
Efficiency analysis among some existing representative schemes.

Auth. scheme	Reg. phase	Login/Auth. phase	Total
Jo-Youn scheme [23]	2⊕, 4h	14⊕, 3h	16⊕, 7h
Jeong et al.'s scheme [24]	1⊕, 2h	2⊕, 8h	3⊕, 10h
		2SE, 1D	2SE, 1D
Kim-Chung scheme [25]	7⊕, 5h	13⊕, 8h	20⊕, 13h
Proposed scheme	6⊕, 5h	14⊕, 15h	20⊕, 20h
		2SE, 1D	2SE, 1D

⊕ – XOR operation; h – hash function; SE – Symm Encryption; D – Decryption

7. Conclusions

Nowadays, not only remote access control to digital home appliances but also services offered by service providers are appealing in digital home network environments. Nonetheless, the remote control services cause digital home networks to have major security threats. To provide secure remote access in home network environments, we propose a robust and efficient user authentication scheme based on strong-password approach, which is relatively lightweight. The proposed user authentication scheme uses the HOTP algorithm, hash-chaining technique along with low-cost smart cards. Our scheme satisfies several security requirements including stolen smart card attack and forward secrecy with lost smart card as well as functional requirements including no verification table and no time synchronization. We have analyzed and verified the security of the proposed using non-monotonic cryptographic logic (Rubin logic). While comparing with existing representative schemes in terms of security requirements and functional requirements, it can be seen that even though the proposed scheme has slightly higher computational overheads than the existing representative schemes, it is more robust authentication mechanism and has better security properties than those schemes.

Acknowledgment

Part of this work has been supported by *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal.

And also this work is partially supported by the Ubiquitous Computing and Network (UCN) Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy (MKE) in Korea and a result of subproject UCN 09C1-T2-10M.

Appendix A. Rubin logic – NCP descriptions

A.1. Global and local sets

A.1.1. Global sets

Principal set: This set contains the principals who participate in a protocol. $P = \{P_1, P_2, \ldots, P_n\}$. Any P_i may be marked as an initiator of the protocol.
Rule set: This set contains inference rules for deriving new statements from existing assertions. $R = \{R_1, R_2, \ldots\ldots, R_n\}$, where R_i is of the form $\frac{C_1, C_2 \ldots\ldots C_n}{D}$, C_i is a condition and D is a statement.
Secret set: This set contains all of the secrets that exist at any given time in the system. $S = \{S_1, S_2, \ldots\ldots, S_n\}$
Observers set: For each S_i, Observers(S_i) contains all the principals who could possibly know the secret S_i by listening to network traffic or generating it themselves.

A.1.2. Local sets

Possession set (P_i): This set contains all the data relevant to security that this principal knows or possesses. This includes secret encryption keys, public keys, data that must remain secret, and any other information that is not publicly available. $\mathrm{POSS}(P_i) = poss_1, poss_2, \ldots\ldots.., poss_n$. possi contains two fields: the actual data and the origin of the data.
Belief set (P_i): This set contains all the beliefs held by a principal. This includes the belief that the keys it holds between itself and other principals are good, beliefs about jurisdiction, beliefs about freshness, and beliefs about the possessions of other principals. $\mathrm{BEL}(P_i) = bel_1, bel_2, \ldots\ldots., bel_n$.
Seen (P_i): This set contains plaintext message parts that P_i sees from messages sent across the network. The Seen sets collectively contain the same information as the observers sets.
Behavior List (P_i): This item is a list rather than a set because the elements are ordered. BL = $AL, bvr_1, bvr_2, \ldots\ldots\ldots, bvr_n$. AL is an action list.
Haskeys (P_i): This set contains keys that P_i sees either because they are in the initial possession set, or because they appear in a message sent across the network and added to P_i's Seen set.

A.2. Actions

A.2.1. Actions derived from [12]

1. Generate-nonce(N):
   Result: $\mathrm{POSS}(P_i)$:= $\mathrm{POSS}(P_i) \cup \{N\}$
   $\mathrm{BEL}(P_i)$:= $\mathrm{BEL}(P_i) \cup \{\sharp(LINK(N))\}$
   Description: This action is used when a principal generates a nonce to link a challenge and a response. LINK(N) is removed from $\mathrm{BEL}(P_i)$ when the response is received. This is used to determine freshness of the data.
2. Encrypt(X, k)
   Condition: $X,\ k \in \mathrm{POSS}(P_i),\ P_i \in \mathrm{Observers}(k)$.
   Result: $\mathrm{POSS}(P_i)$: = $\mathrm{POSS}(P_i) \cup \{\{X\}_k\}$.
   Description: This action is used when a principal encrypts data. If P_i possesses X and knows k then he can possess $\{X\}_k$.
3. Decrypt($\{X\}_k, k$)
   Condition: $P_i \in$ Observers(k), $\{X\}_k,\ k \in \mathrm{POSS}(P_i)$.
   Result: $\mathrm{POSS}(P_i)$: = $\mathrm{POSS}(P_i) \cup \{X\}$.
   Description: This action is used when a principal decrypts data. If P_i possesses X, encrypted under k, and P_i knows k, then P_i can possess X.
4. Concat($X_1, X_2, \ldots, X_n$)
   Condition: $X_1, X_2, \ldots, X_n \in \mathrm{POSS}(P_i)$.
   Result: $\mathrm{POSS}(P_i)$: = $\mathrm{POSS}(P_i) \cup \{X_1, X_2, \ldots\ldots., X_n\}$.
   Description: This action is used when a principal constructs a message, X , out of submessages $X_1, X_2, \ldots, X_n$.
5. Split(X)
   Condition: X contains $x_1, x_2, \ldots\ldots, x_n, X \in \mathrm{POSS}(P_i)$.
   Result: $\mathrm{POSS}(P_i)$: = $\mathrm{POSS}(P_i) \cup \{x_1, x_2, \ldots\ldots., x_n\}$.
   Description: This action is used to break a message into its components. Split is the opposite of concatenation.
6. Generate-secret(s)
   Result: $S := S \cup \{s\}, Observers(s) = \{P_i\}, POSS(P_i) := POSS(P_i) \cup \{s \bowtie P_i\}$

   $BEL(P_i) = BEL(P_i) \cup \sharp(s)$

   Description: This action is used when a principal generates a secret data item, such as a key. A new secret, s, is added to S, and the Observers and possession sets are updated.
7. Send(P_j, X): It means that P_i sends X to P_j.

8. Receive(P_j, X): It means that P_i receives X from P_j. In this case, X will be marked as coming from P_j and added to POSS(P_i).
9. Update(X): The Update function is used to maintain the observers of X.
10. Forget(X): This action is used when P_i no longer is in possession of X.
11. Abort : This could happen under various circumstances where there is an inconsistency or other flaws in the protocol specification. If protocol run is illegal, analysis reports failure.

A.2.2. Actions adopted from [13]

1. Hash$(h(\cdot); X)$
   Condition: $h(\cdot), X \in \text{POSS}(P_i)$
   Result: $\text{POSS}(P_i) := \text{POSS}(P_i) \cup \{h(X)\}$
   Description: This action is used to hash data
2. XOR$(X_1, X_2, \ldots, X_n)$
   Condition: $X_1, X_2, \ldots\ldots, X_n \in \text{POSS}(P_i)$
   Result: $\text{POSS}(P_i) := \text{POSS}(P_i) \in \{X_1, X_2, \ldots., X_n\}$
   Description: This action is used to XORing data.
3. Check(X, Y)
   Condition: $X, Y \in \text{POSS}(P_i)$
   Result: Valid if $X = Y$, else Invalid

A.3. Inference rules

1. Nonce verification rule:

$$\frac{\sharp(X) \in BEL(P), X \text{ from } Q \in POSS(P)}{BEL(P) := BEL(P) \cup \{Q \text{ believes } \sharp(X)\}}$$

2. Message meaning rule:

$$\frac{\{X\}_k \text{ from } Q \in POSS(P), \{P, Q\} \subseteq Observers(k)}{BEL(P) := BEL(P) \cup \{X \in POSS(P)\}}$$

3. Submessage freshness rule:

$$\frac{\sharp(x_1) \in BEL(P), \{X \text{ contains } x_1, X \text{ contains } x_2\} \subseteq POSS(P)}{BEL(P) := BEL(P) \cup \sharp(x_1)}$$

4. Origin rule :

$$\frac{X \in POSS(P), X \text{ contains } x_1, Q \in Observers(x_1)}{x_1 \text{ from } Q \in POSS(P)}$$

5. Submessage origin rule :

$$\frac{X \in POSS(P), X \text{ contains } \{x_1, x_2\} \text{ from } Q}{x_2 \text{ from } Q \in POSS(P)}$$

6. Linkage rule (symmetric keys)

$$\frac{\sharp(k) \in BEL(P), P \in Observers(k), LINK(N_a) \in BEL(P), X \text{ contains } f(N_a), X \text{ contains } x_1, \{X\}_k \text{ from } Q \in POSS(P)}{BEL(P) := (BEL(P) - LINK(N_a)) \cup \sharp(x_1)}$$

References

[1] B. Rose, Home networks: a standard perspective, IEEE Communication Magazine 39 (12) (2001) 78–85.
[2] H. Schulzrinne, W. Xiaotao, S. Sidiroglou, S. Berger, Ubiquitous computing in home networks, IEEE Communications Magazine 41 (11) (2003) 128–135.
[3] K.S. Choi, S.O. Lim, Y.C. Park, K.M. Jung, Home station, novel architecture of home gateway and its implementations. in Proceedings of the 4th WSEAS International Conference on Applied Informatics and Communications (AIC'04), Tenerife, Canary Islands, Spain, 2004.
[4] M. Ise, Y. Ogasahara, K. Watanabe, M. Hatanaka, T. Onoye, H. Niwamoto, I. Keshi, I. Shirakawa, Design and implementation of home network protocol for appliance control based on IEEE 802.15.4, in: IJCSNS International Journal of Computer Science and Network Security, vol. 7, issue 7, 2007, pp. 20–30.
[5] C.S. Tsai, C.C. Lee, M.S. Hwang, Password authentication schemes: current status and key issues, International Journal of Network Security 3 (2) (2006) 101–115.
[6] N.M. Haller, The S/KEY one-time password system, in: Proceedings of the Internet Society Symposium on Network and Distributed System Security, 1994, pp. 151–158.
[7] J.J. Shen, C.W. Lin, M.S. Hwang, A modified remote user authentication using smart card, IEEE Transactions on Consumer Electronics 49 (2) (2003) 414–416.
[8] S.W. Lee, H.S. Kim, K.Y. Yoo, Improved efficient remote user authentication scheme using smart cards, IEEE Transactions on Communications 50 (May) (2004) 565–567.
[9] E.J. Yoon, E.K. Ryu, K.Y. Yoo, An improvement of Hwang–Lee–Tang's simple remote user authentication schemes, Computers and Security 24 (2005) 50–56.
[10] J.Y. Liu, A.M. Zhou, M.X. Gao, A new mutual authentication scheme based on nonce and smart cards, Computer Communications 31 (2008) 2205–2209.
[11] H.C. Hsing, W.K. Shin, Weaknesses and Improvements of the Yoon–Ryu–Yoo remote user authentication using smart cards, Computer Communications 32 (2009) 649–652.
[12] A.D. Rubin, P. Honeyman, Nonmonotonic cryptographic protocol, in: Proceedings of the Computer Security Foundation Workshop VII, June 1994, pp. 100–116.
[13] M.L. Das, V.L. Narasimhan, Towards a formal verification of an authentication protocol using non-monotonic logic, in: Proceedings of 5th International Conference on Information Technology: New Generation, 2008.
[14] L. Lamport, Password authentication with insecure communication, Communications of the ACM 24 (11) (1981) 770–772.
[15] T.C. Yeh, H.Y. Shen, J.J. Hwang, A secure one-time password authentication scheme using smart cards, IEICE Transaction on Communication E85-B (11) (2002) 2515–2518.
[16] T. Tsuji, A. Shimizu, One-time password authentication protocol against theft attacks, IEICE Transactions on Communications E87-B (3) (2004) 523–529.
[17] W.C. Ku, H.C. Tsai, M.J. Tsaur, Stolen-verifier attack on an efficient smartcard-based one-time password authentication scheme, IEICE Transactions on Communications E87-B (8) (2004) 2374–2376.
[18] N.W. Wang, and Y.M. Huang, User's authentication in media services by using one-time password authentication scheme, in: Proc. of the 3rd International Conference on International Information Hiding and Multimedia Signal Processing (IIH-MSP 2007) 01, 2007, pp. 623–626.
[19] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, O. Ranen, HOTP: An HMAC-based One-Time Password Algorithm, IETF RFC 4226, December 2005.
[20] H. Krawczyk, M. Bellare, R. Canetti, HMAC: Keyed-Hashing for Message Authentication, IETF RFC 2104, February 1997.
[21] N.Y. Lee, J.C. Chen, Improvement of one-time password authentication scheme using smart card, IEICE Transaction on Communications E88-B (9) (2005) 3765–3769.
[22] I. You, E.S. Jung, A Light Weight Authentication Protocol for Digital Home Networks. Computational Science and Its Applications – ICCSA 2006, LNCS 3983, 2006, pp. 416–423.
[23] H.S. Jo, H.Y. Youn, A Secure User Authentication Protocol Based on One-Time-Password for Home Network. Computational Science and Its Applications – ICCSA 2005, LNCS 3480, 2005, pp. 519–528.
[24] J. Jeong, M.Y. Chung, H. Choo, Integrated OTP-based user authentication scheme using smart cards in home networks, in: Proc. of the 41st Annual Hawaii International Conference on System Sciences (HICSS'08), January 2008.
[25] S.-K. Kim, M.G. Chung, More secure remote user authentication scheme, Computer Communications 32 (2009) 1018–1021.
[26] E.J. Yoon, K.Y. Yoo, More efficient and secure remote user authentication scheme with smart cards, in: Proceedings of 11th International Conference on Parallel and Distributed System, vol. 2, 2005, pp. 73–77.
[27] P. Kocher, J. Jaffe, B.B. Jun, Differential power analysis, in: Proceedings of Advances in Cryptology (CRYPTO'99), 1999, pp. 388–397.
[28] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Transactions on Computers 51 (5) (2002) 541–552.
[29] S.T. Wu, B.C. Chieu, A user friendly remote authentication scheme with smart cards, Computers and Security 22 (6) (2003) 547–550.
[30] J.J. Shen, C.W. Lin, M.S. Hwang, Security enhancement for the time-stamp-based password authentication scheme using smart cards, Computer and Security 22 (7) (2003) 591–593.