Journal of Information Security Research

DLINE JOURNALS

Print ISSN: 0976-4143
Online ISSN: 0976-4151

JISR 2025: 16 (3)
https://doi.org/10.6025/jisr/2025/16/3/110-124

Only 3-way Handshake Pake based on Reversible Cellular Automata

Soufiane Baghor
Computer science department, UDL University, Sidi Bel Abbes. Algeria
baghorsoufiane@yahoo.fr

Kamel Mohamed Faraoun
Computer science department, UDL University, Sidi Bel Abbes. Algeria
kamel_mh@yahoo.fr

ABSTRACT

In this paper, we propose an only three way handshake password authenticated key exchange using only the mechanism of two-dimensional reversible cellular automata. The scheme use simple and elementary operations to handle the problem of mutual authentication between two communicating parties, and establish common secure, valid session key based only on a short, easily memorable password. Security of the proposed scheme shown to be high, while it resist most common attacks on authentications schemes and provide the perfect forward secrecy. Besides, runtime performances of the authentication protocol are very competitive, and outperform those of existing similar schemes. The proposed schemes ensure the three authentication requirements, namely the mutuality, the authenticity and the key exchange using only elementary and parallelizable operations, without requiring any addition cryptographic functions.

Keywords: Password Authentication, Mutual Authentication, Key Exchange, Two-Dimensional Reversible Cellular Automata

Received: 11 February 2025, Revised 17 May 2025, Accepted 1 June 2025

1. Introduction

Entity authentication and secure key exchange are two important building blocks of any multi-user secure distributed/centralized system. While the former aims to prove the identity of a given user and allow it to gain access to a given service, the later aims at establishing a cryptographic key that enable secure communication

between the user and the requested provider during a predetermined communication's session. The user owns a specific and personal information (that can be either a password or a smart card) that enables his authentication within a server, and permits to establish a valid cryptographic key in a secure manner. An eavesdropper or a man in the middle should not be able to obtain enough information permitting to brute force or guess the password, and any unauthorized user should be able to impersonate the server and take the identity of legitimate user.

Combining both entity authentication and secure key exchange leads to the definition of a Password Authenticated Key Exchange (PAKE) protocols permitting to achieve a secure communications between two or more parties based only on the knowledge of single password.

A solution to the key exchange problem first introduced by Diffie-Hellman [1] using exponential algorithm. However, this scheme neither provides authentication nor resists against man-in the-middle attack. It is necessary to verify the identities of the communicating parties, when they establish a connection. This authentication is usually provided in combination with a key exchange protocol between the parties.

The concept of PAKE was first introduced by Bellovin and Merritt in 1992 [2] and there after many PAKE protocols have been proposed and analyzed [3-8]. Most of these schemes involves modular exponentiations with high computational overhead and communication bandwidth, which is unsuitable for low power applications.

In this paper, we propose an only three way handshake password authenticated key exchange using only the mechanism of two-dimensional reversible cellular automata. Thus, the scheme provides a high security and performances level with an optimized computational cost. The proposed scheme handle remote entity authentication, session key exchange and mutual authentication between two communicating parties using a simple and elegant computational model, and it is shown to be resistant against common related attacks.

The remaining of the paper is organized as follows: Section 2 presents related works on password authenticated key exchange using cellular automata, and Section 3 gives elementary details on the cellular automata mechanism. In section 4, the proposed authentication scheme is fully described and detailed, while related security analysis is presented in Section 5 and comparisons analysis are presented in Section 6. Finally, conclusions are drawn in section 7.

2. Related works

Recently, some password-based authentication protocols have been implemented by cellular automata since it has been accepted as a good calculation model and have been used in various domains but little used in authentication systems.

D. K. Bhattacharryya & S. Nandi made the first attempt [9] who proposed an authentication protocol that allows two parties that share a small password to generate a common cryptographic key using simple cellular automata operations to simplify the logical structure. However, this scheme uses numerous instances CA-based hash function, is not resistant to the stolen session key attack, is not robust against the attack on known

key security, hence the protocol does not provide perfect forward secrecy. In addition, the protocol can be executed only by the presence of a shared secret key.

In [10] *S* Tripathy & S Nandi (2006) proposed a password-based authentication protocol using bi-dimensional reversible cellular automata. This protocol provides mutual authentication and key agreement on an insecure channel, it is also robust against all passive and active attacks while providing perfect forward secrecy, but through the addition of more instances of hash function, that increases the computational overhead of the scheme.

Finally, authentication based on the A Secure CA Based 2-Party Key Exchange Protocol was proposed in [11] using the group and non-group characteristics of cellular automata. The security of the protocol is based primarily on asymmetric encryption, where each participant has a certified key pair. The protocol depends on a PKI infrastructure and a complete management of keys (certification request, verification of the validity of certificates, etc.). Another required module is the generator of the secret from the digital fingerprint; this requires complicated processes and expensive devices for companies.

In addition to these limitations, we note an application limit in the case of an application (user-to-user authentication); the prior sharing of the digital fingerprint remains a problem.

3. Cellular Automata Preliminaries

A cellular automaton consist of a number of cells arranged in a regular lattice, when each cell has its own state changing in a discrete time step. States of the CA'$_s$ cells are updated synchronously using a local transition rule defining each new cell's state using its old state, and the states of the corresponding neighbors. The neighbors are a specific selection of cells relatively chosen with respect to a given cell's position that can be defined for each cell using a radius *r* on the lattice. This gives 2r+1different neighbor including the cell itself.

The boundaries cells of the lattice are concatenated together in a cyclic form to deal with finite size automaton. If the same update rule is used for all the cells then the resulting CA is named uniform. Otherwise, if a different transition rule is used each time the cell's position change, the resulting CA is named non-uniform.

Formally, a *CA* is defined by 4-tuple (*d*, *S*, *N*, *f*) where,

- d: dimension 1, 2, 3....
- S: finite state set {1,2,,....,*n*}
- N: Neighborhood vector $(x_1,.. x_m) \in Z^m$
- f: Local update rule mapping from S^m to *S*.

If we define the state of a cell i at a time *t* with s^t_i, , its state on the time *t*+1 depends only on the states of the corresponding neighborhood at time *t,* by applying a transition rule that define the way states are updated. If the neighborhood radius is *r*, and if two cell states are defined (0 or 1), then the length of each transition rule is then 2^{2r+1} bit, and the number of possible rules is equal to $2^{(2^{2r+1})}$.

This feature of simple local-rule leading to complex global-behavior motivated many researchers to design security primitives. Interested reader is referred to [12]-[16].

The transition rule of one dimensional binary CA_s is generally coded using the integer value of the corresponding binary representation, when the different CA'_s configurations are represented by binary blocks.

Given some evolution rule, a sequence of states has a single evolution, but it can be generated by one, many or no ancestors. A cellular automaton is reversible if each configuration has a single ancestor. Such a cellular automaton is called reversible cellular automata and is briefly explained in next.

3.1. Reversible Cellular Automata (RCA)

Reversible Cellular Automata (RCA) are dynamical systems, which conserve their initial information. For this reason, they present a very interesting mathematical theory, and have been used as models for data ciphering [19], information coding [18] and simulation of reversible physical phenomena [17] among other applications. The theory of Reversible Cellular automata (RCA), are realized in [20-22]. Algorithms for finding reversible one-dimensional *CA* are proposed in [19].

CA A is reversible (called invertible) if its global map G_A is invertible i.e. there exists another automaton B (with global map G_B) such that $G_B = G_A^{-1}$. Here the automaton B is called the inverse of A. For every possible configuration C: $G_B o G_A(C) = C$. Therefore, for every possible configuration the global map specifies one and only one successor. However, the problem of deciding, whether a configuration has a predecessor is proved to be NLOG complete for one-dimensional and NP-complete for all higher ($d \geq 2$) dimensional *CA* [23].

Basing on this theory, RCA can be used as cryptosystem. Let $T = (d, S, N, f)$ be an RCA and $T^{-1} = (d, S, N', f')$ be its inverse. The message M is written in ($d \geq 2$) dimensional array using symbols in S. Now the cipher can be obtained by applying T on M. In reverse, operating T^{-1} on the cipher, the original message can be deduced. Higher dimensional ($d \geq 2$) RCA is to be used for this purpose, as all one-dimensional RCA can be effectively inverted [24]. Apart from that, the RCA to be used in the proposed system is required to have the following properties.

- Given the RCA T it is computationally infeasible to find T^{-1};
- However, it is easier to find T by knowing T^{-1}.

Composition of some easy invertible RCA can be used for this purpose. Selecting some easily invertible RCA ($T_1, T_2 \ldots T_n$), and their composition $T = (T_1 \text{ o } T_2 \ldots \text{ o } T_n)$ can be published and used as a public key keeping $T_1, T_2 \ldots T_n$ in secret. $T^{-1} = (T_n^{-1} \text{ o } T_{n-1}^{-1} \ldots \text{ o } T_1^{-1})$ can be used as private key without sacrificing the secrecy. The proposed scheme exploits such RCA, which is illustrated in the next section.

4. The Proposed Scheme

4.1. The Issue

The problem comes down to the ability to overcome a brute force attack to find the secret (the low entropy password). Therefore, to design a strong password-based authentication system, we need to consider the following particular properties and requirements:

• Users have only a low entropy secret. More precisely, it is possible for the opponent to search in all possible secrets within a reasonable time.

• Offline dictionary attacks should not be possible. This means that a passive indiscretion that registers the transcription of one or more sessions cannot eliminate a large number of possible passwords.

• Online dictionary attacks should not be possible. This means that the active opponent cannot abuse the protocol in order to eliminate a large number of possible passwords. An active opponent can always eliminate at least one password per attempt to pass himself off as the legitimate user by using this password.

Our main contribution is the minimization of the number of message flows exchanged during the authentication and key confirmation phase, as well as the minimization of expensive computational operations such as (hash functions ...), ensuring a high level of security.

4.2 Protocol Design Assumptions

Assumption 1: This is a case where the opponent can listen to all messages sent or received. This is a realistic assumption in typical communication systems.

Hypothesis 2: Another realistic hypothesis where the opponent can intercept, modify and replay messages.

Hypothesis 3: Again, a hypothesis very realistic - it is widely believed that internal participants are often more a threat that foreigners. This brings us to another security hypothesis.

Hypothesis 4: A typical scenario is where two participants commit to a shared key protocol and one returns a challenge that was intended for itself. This attack can only be possible if parallel execution of the same protocol is allowed, but this is often a realistic assumption.

4.3 The Protocol Architecture

There are three characteristics that we consider architectural criteria to classify different authentication protocols and cryptographic key establishment: the existing cryptographic keys, how the session key is established and the number of users of a protocol is designed to serve.

Design Criterion 1: Alice and Bob share a pre-shared secret to ensure the authentication of the entity and the establishment of secure key.

Design Criterion 2: The session key is generated by Bob and transported to Alice safely (Key Transport Protocol).

Design Criterion 3: Only one of the two parties contributes to generating the session key and transporting it to the peer entity.

4.4 The Desired Goals

We will cite in this section a detailed list of objectives desired by our protocol, the list is illustrated in the following table:

4.5 What do we want to protect? Against what dangers?

The confidentiality of the session key and the shared secret is mandatory in order to counteract the passive attacks. In addition to passive attacks: it is necessary for designers of authentication protocols to propose techniques in order to counteract active attacks.

The following table presents the set of attacks that we wish to counteract during the design and construction of our protocol. In order to clarify the objectives of our protocol. Since for a protocol to achieve particular objectives, it depends on its strength to thwart all attacks considered possible.

The protocol must ensure authentication of the entity; i.e. one of the parties must be assured of the identity of the second party involved in the protocol and that the second party must have actually participated;
The protocol must provide mutual authentication between Bob and Alice; That is to say, the first objective must be ensured in both directions;
The protocol must establish a fresh session key, known only to session participants; The protocol must provide key integrity;
The protocol must provide key confirmation from Alice to Bob i.e. That Alice must have the assurance that the session key K_i is a good key to communicate with Bob, and that Bob is in possession of that key.
The purpose of the protocol is to provide mutual trust in the K_i key, i.e., Bob's prediction that K_s is a good key for use with Alice, and Alice wishes to communicate with Bob using the K_i key, which Alice believes is good for this use.
Finally, the protocol aims to provide forward secrecy i.e. if the long-term secret is compromised, session keys that have been previously established using this secret should not be compromised too.

Table 1. The desired goals

Password guessing attack (Online/Offline);
Replay attack and clock un-synchronization;
Mutual authentication and man in the middle attack;
Session key agreement, forgery attack and known key security;
No verifier-table stored and stolen verifier attack;
Reflection attack;
Denial Of Service (DOS).

Table 2. List of possible attacks

4.6 Protocol Construction

This approach is a theoretical design that depends on any cryptographic mechanism (symmetric, asymmetric and one-way function). This is called modularity of the crypto-system. We let total freedom for the integrator of our solution to choose any cryptographic mechanism if it is safe and proven, this choice is made as required.

We recommend mechanisms based on proven cellular automata in order to obtain a fast, light and efficient implementation. For symmetric cryptography, we opt for the crypto-system of our proven approach [22]; for asymmetric cryptography we recommend the crypto-system of Kari [23] and for the one-way functions we choose [26] from [24-26].

Using the mechanism of second-order reversible cellular automata, we propose a new system PAKE Password Authenticated Key Exchange. The security of our approach relies solely on a small low entropy password (PWi) shared between the two communicators (Alice and Bob), and on the difficulty of reversing a two-dimensional RCA Reversible Cellular Automata.

The concept of challenge in a similar way to the principle of nonces. We use the previous session key Ki-1 as a challenge instead of generating and sending new nonces in each execution, which reduces the number of messages exchanged during the execution of the protocol. The use of this challenge at the same time eliminates the time synchronization problem between the participants.

The Ki-1 key is used in a challenge-response scenario in which both participants retain the value of the previous one-time session key of each session. Alice then sends Ki-1 encoded by the derived S key from the shared password (PWi) in order to meet the challenge Bob is expecting. Alice checks the validity of the challenge,

H()	A one-way cryptographic function
ψ ()	The expansion function
K_{i-1}	The previous session key used as a challenge
ID_i	The identifier of the user Ui (Alice)
PW_i	The user's password Ui (Alice)
S	The secret key derived from the PWi
K_i	The current session key
T	The public key generated by Alice
T^{-1}	The generated private key corresponding to T, generated by Alice

Table 3. The rating table

otherwise Bob refuses Alice's request.

The proposed authentication system comprises two phases: an authentication and key exchange phase, and a phase of confirmation of key knowledge. The following notations are adopted in the remainder of this article:

Our protocol dispense stage registration since the protocol needs only a small password to be completed successfully. This phase allows mutual authentication between the two parties (Alice and Bob), so that each verifies the identity of the other. After the authentication phase, a shared session key is set up to enable secure communication between the two parties during a predetermined communication session. An intruder who has access to the messages exchanged should not be able to draw information about the secret S, or the established session key K_i.

4.6.1 Authentication and key Exchange Phase

The phase of mutual authentication and key exchange takes place in 3 steps:

• **Make-1:** For Alice who wishes to authenticate with Bob, he must first enter his identifier ID_i and his password PW_i to start an access session:

- A secret key S is derived from the password PW_i;

- Alice generates a reversible cellular automaton T and its inverse T^{-1};

- Then, Alice calculates the cryptogram $X = E_s (K_{i-1}, T)$;

- Finally, the couple $<ID, X>$ is sent to Bob.

Note that Alice does not need to wait for a challenge from Bob because he is aware of the challenge (K_{i-1}) he needs to code it and send it to Bob. Bob on his side waits for him to receive the challenge in the form of a cryptogram so that he can authenticate Alice.

• **Make-2:** When receiving the above pair, first Bob must retrieve the key K_{i-1} from the cryptogram $X = E_s(K_{i-1},T)$ using the secret key S so that, It can authenticate Alice. After comparing the recovered key to the key of the previous stored session, Bob authenticates and accepts Alice's request in case of a tie, otherwise the user's request is denied.

In order to perform mutual authentication, the server must also prove its identity to the user. To do this, the received ephemeral public key is used. Bob generates a communication key for the current session and then encodes this key with the previous session key under the cryptogram $Y = T (K_{i-1}, K_i)$ using the ephemeral public key T. Alice tries to recover the encoded challenge New fresh and unique shape. If the equality is verified, then Bob is authenticated with Alice. If not, Bob is considered an intruder, and the connection is interrupted. Figure 1 illustrates the fully detailed diagram.

4.6.2 Key Confirmation Phase

After mutual authentication has been successfully completed, both parties must have the knowledge of the

key used for secure communication in the current session confirmed. The session key K_i is generated by Bob and sent to Alice during the messages exchanged during the authentication phase (MAKE-2). A spy that collects all the exchanged messages is unable to deduce all the information about the session key since Alice is the only one who owns the T^{-1} decryption private key.

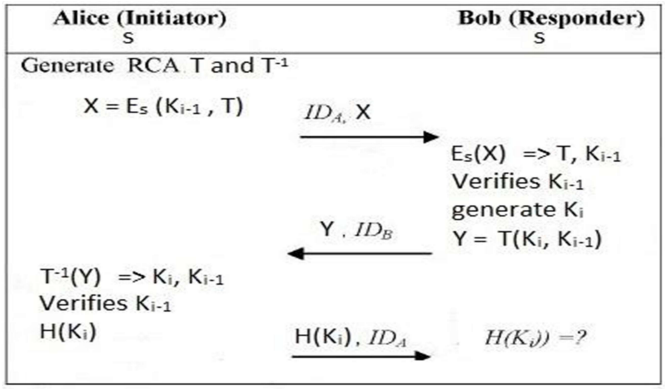

Figure 1. Proposed Authenticated Key Exchange Diagram

- **KC-1:** Confirming Bob's key with Alice is an integral part of the message (MAKE-2) since the key transport mechanism is used.

- **KC-2:** Confirmation of the Alice key with Bob is done by sending the currently hashed session key $H(K_i)$; it is clear that this message is fresh and unique and cannot be forged by a malicious user. On receipt of this hash, Bob performs the same operation and compares the two hashes; if they are equal, Bob will be sure that Alice has the right communication key. Finally, a secure communication can be carried out.

5. Experimental Results

In the following, we discuss the safety properties of the proposed scheme in relation to the potential joint attacks. It is assumed that the attacker has the potential to listen to the communication channel between Alice and Bob and can modify any message exchanged. A comparative table is also presented in the following section between the proposed scheme and some existing CA-based authentication schemes.

5.1 Password Guessing Attack

In the proposed scheme, the user's password PW_i is transformed into a secret key S by means of an expansion function Ψ where $S = \Psi(PW_i)$. In addition, the secret key derived from the password is used to encrypt the RCA T (which is itself generated randomly) and the previous session key K_{i-1}, knowing that T and K_{i-1} have not been transmitted anywhere in the clear on the network. The only password packet passed from Alice to Bob contains $<ID, X = ES(K_{i-1}, T)>$. As a result, the attacker is unable to retrieve information about the password from the identifiers and messages being played between Alice and Bob, so there is no way to mount an offline attack to guess the password (off-line password guessing attack).

Finally, to resist an online attack (on-line password guessing attack), the system limits the number of attempts to a limited number of times.

5.2 Replay Attack and Clock Un-Synchronization

The timestamp-based authentication system may suffer from the replay attack since the transmission delay is unpredictable in a network environment. For this reason we use a nonce based model for the proposed scheme instead of a timestamp-based design, so that the system can prevent such a serious problem from the non-synchronization of the clock. In order to better optimize our system the nonces are not generated every time to launch a challenge, but the idea is to use the previous session key K_{i-1} as a nonce that is both known by Alice and Bob.

During the authentication phase, an opponent can claim to be a valid user and identify himself to the server (Bob) by sending messages that were previously transmitted by a legitimate user (Alice). During the authentication phase, an opponent can claim to be a valid user and attempts to connect to the server by sending him messages previously transmitted by a legitimate user. To avoid such an attack, the user generates an unpredictable and different RCA T for each session, and the server generates an novel fresh session key K_i which makes messages exchanged from a given session valid only for that session, and could not be reused in other sessions. Therefore, an opponent who replays a previous X message is unable to perform the last phase of the protocol.

Therefore, an opponent cannot overcome the verification process with the server with legitimate values previously collected.

Obviously, the replay attack fails. Therefore, the proposed scheme can withstand a replay attack.

5.3 Mutual Authentication and Man in the Middle Attack

It is a basic requirement of an authentication system that the user and server authenticate themselves to each other. In the proposed scheme, the server authenticates the user by verifying the authentication message using any symmetric encryption mechanism. If the server successfully retrieves K_{i-1} from the connection request message, the server confirms the legitimacy of the user. The server then responds with a specific message ($Y = (T (K_i, K_{i-1})$) to the user, which contains the new session key with the old one in order. The user authenticate the server by checking this pair, which cannot be constructed only by the legitimate server.

Therefore, mutual authentication is guaranteed by the $MAKE_{-1}$ and $MAKE_{-2}$ steps of the protocol. We know that the user's spoof attack and the fake server attack cannot be invoked by a malicious user U_z. Therefore, it is impossible for U_z to conduct a man-in-the-middle attack. It is obvious that the protocol is resistant to man in the middle attack while ensuring mutual authentication.

5.4 Session key Agreement, Forgery Attack and known key Security

After a successful mutual authentication, a session key is safely transported, and finally shared between the client and the server for use during the rest of the communication session. Our protocol ensures that the session key is known only by the legitimate entities (the client and the server). Since Alice generates the *RCA* T and its inverse T^{-1} randomly and freshly, so only Alice can retrieve the current session key from the Y cryptogram; therefore, no third party can send a forged message to the client or server. Since the challenge (K_{i-1}) and the randomly generated RCA are different for each session, even if a given session key is compromised, neither the secret key nor the session keys (past and future) will be compromised. Therefore, the proposed scheme ensures (known key security).

5.5 No Verifier-Table Stored and Stolen Verifier Attack

As long as neither Alice nor Bob store no verification table, authentication depends only on a small, low-entropy password. As a result, the system resists storage table modification attack and ensures lower storage cost. Bob uses the secret key S computed from the shared password PW_i to authenticate Alice while guaranteeing the freshness of the cryptogram; While Alice uses the *RCA T* and its inverse T^{-1} generated during the authentication phase to authenticate Bob. Therefore, the proposed system is stolen verifier attack.

5.6 Reflection Attack

A reflection attack can only be performed if and only if parallel executions of the same protocol are allowed. For example, if a participant is an Internet host, it can accept sessions from several other participants. In our case, a reflex attack cannot be performed even if the protocol allows several parallel executions for two reasons. The first reason - no challenge runs in clear during the authentication phase. While de second - even if a challenge can be unveiled or predicted, such an attack cannot be accomplished for the Alice's way of authenticating with Bob (By a symmetrical mechanism) which is completely different to the Bob's way of authenticating with Alice (by an asymmetric mechanism).

Security Criterion	Cpake [9]	Rpake [10]	Ca-kep [11]	Proposed
Perfect forward secrecy	No	Yes	Yes	Yes
known key security	No	Yes	Yes	Yes
Stolen session key attack	No	Yes	Yes	Yes
Offline-password guessing	No	Yes	Yes	Yes
Resistance to Stolen verifier attack	No	Yes	No	Yes
Resistance to replay attack	Yes	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes
Resistance to Man in The Middle attack	Yes	Yes	Yes	Yes
User Impersonating Attack	Yes	Yes	Yes	Yes
Server Counterfeit Attack	Yes	Yes	Yes	Yes
Resistance To Forgery Attack	Yes	Yes	Yes	Yes

Table 4. Security aspects comparison

5.7 Perfect Forward Secrecy

The idea of forward secrecy is that when a long-term key (*S*) is compromised, session keys that were previously established with this long-term key should not be compromised too. The crucial concept for providing forward secrecy is the principle of the ephemeral public key (in our case we used the RCA *T* and its inverse T^{-1}); This is a public key *T* which is used only during the duration of the key establishment protocol and is then destroyed with its corresponding private key T^{-1}. If the public key is used only to authenticate the session key, and the session key cannot be retrieved without the ephemeral private key, then the key K_i cannot be retrieved from

the cryptogram $Y = T(K_i, K_{i-1})$ without knowledge of the private key T^{-1}. Therefore, our protocol provides persistent confidentiality.

6. Comparative Analysis

In order to illustrate the advantages of the proposed system, a comparative study was carried out against several existing CA-based authentication schemes. It is clear that the proposed approach ensures all the listed properties while requiring less computational cost.

6.1. Security Analysis

The comparison is made with respect to the security properties and the complexity of implementation. Table 4 presents several aspects of security provided by the proposed scheme compared to most known CA-based authentication schemes proposed in [9], [10] and [11].

6.2. Implementation Analysis

Since the CA operations use the simple binary operators that are executed independently for each bit, the CA-based evolution mechanism has an inherent parallelism, and therefore provides an extremely fast and efficient implementation, especially on hardware design. This also prevents denial-of-service (DoS) attacks based on server overload with spurious service requests.

Mutual Authentication phase	**Cpake [9]**	**Rpake [10]**	**Ca-kep [11]**	**Proposed**
Number of message flows	4	2	4	2
Instances of hash functions	10	6	4	0
Number of random numbers	6	2	2	0
Instances of exponential modulo operation	0	0	4	0
Instances of symmetric encryption	2	0	0	2
Instances of asymmetric encryption	0	2	4	2
Intermediate data exchange	S_K	-	P_K, S_K	-
Key confirmation Phase				
Number of message flows	4	2	2	1
Instances of hash functions	-	6	4	2
Total				
Number of message flows	8	4	6	3
Instances of hash functions	-	12	8	2

Table 5. Performance Comparison

of both the proposed approach and those already in place. The comparative Table 5 shows that the proposed approach takes advantage in terms of cost of communication and calculation.

7. Conclusion

This paper proposes a new cellular automaton based authenticated key exchange scheme that ensure both authentication and session key agreement in a fast and highly secure manner using only a small shared password. The scheme provides a high security and performances level with optimized computational cost, while performed analysis and experiments shows that it resists known attacks. The proposed scheme handle remote user authentication, session key agreement and mutual authentication between the user and the server using a simple and elegant computational model. Computational experiments performed to measure the runtime performances shows that the scheme provides best performances with respect to existing CA-based ones. In future works, we plan to extend the scheme to handle situations when multiples (n>2) parties contribute to share a common session cryptographic key.

References

[1] Diffie, W., Hellman, M. E. (1976). New directions in cryptography. *In: Proceedings of the IEEE International Symposium on Information Theory*, 29–40.

[2] Bellovin, S. M., Merritt, M. (1992). Encrypted key exchange: Password-based protocols secure against dictionary attacks. *Proceedings of the IEEE Symposium on Research in Security and Privacy*, 72–84.

[3] Gong, L., Lomas, M. L., Needham, R. M., Saltzer, J. H. (1993). Protecting poorly chosen secrets from guessing attacks. *IEEE Journal on Selected Areas in Communications,* 11(5), 648–656.

[4] Jablon, D. P. (1997). Strong password-only authenticated key exchange. *ACM SIGCOMM Computer Communication Review*, 5–26.

[5] Boyko, V., MacKenzie, P., Patel, S. (2000). Provably secure password authenticated key exchange using Diffie-Hellman. *Proceedings of Eurocrypt 2000*, 156–171.

[6] MacKenzie, P., Patel, S., Swaminathan, R. (2000). Password authenticated key exchange based on RSA. *Proceedings of Asiacrypt 2000*, 599–613.

[7] Zhu, F., Wong, D. S., Chen, A. H., Ye, R. (2002). Password authenticated key-exchange based on RSA for imbalanced wireless networks. *Proceedings of ISC 2002*, 150.

[8] Bao, F. (2003). Security analysis of password authenticated key-exchange protocol. *Proceedings of ISC 2003*, 208.

[9] Bhattacharyya, D. K., Nandi, S. (2000). CA based password-only authenticated key exchange. In *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS 2000)* (pp. 820–827). IEEE.

[10] Tripathy, S., Nandi, S. (2006, September). RPAKE: RCA-based password authenticated key exchange. In *2006 Annual IEEE India Conference* (pp. 1–5). IEEE.

[11] Bhuyan, M. H., Bhattacharyya, D. K., Kalita, J. K. (2012). CA-KEP: A secure CA-based 2-party key exchange protocol. *Journal of Information Assurance and Security,* 7(3), 193–206.

[12] Wolfram, S. (1984). Cellular automata as models of complexity. *Nature,* 311(5985), 419.

[13] Wolfram, S. (1985). Cryptography with cellular automata. In *Advances in Cryptology – CRYPTO'85* (pp. 429–432).

[14] Wolfram, S. (1986). *Theory and applications of cellular automata*. World Scientific.

[15] Palchaudhuri, P., Roy Chowdhury, D., Nandi, S., Chattopadhyay, S. (1997). *Additive cellular automata: Theory and applications* (Vol. 1). *IEEE Computer Society*.

[16] Wolfram, S. (2002). *A new kind of science*. Wolfram Media Inc.

[17] Wolfram, S. (1986). *Theory and applications of cellular automata*. World Scientific.

[18] Lind, D., Marcus, B. (1995). *An introduction to symbolic dynamics and coding*. Cambridge University Press.

[19] Mora, J., et al. (2005). Procedures for calculating reversible one-dimensional cellular automata. *Physica*.

[20] Kari, J. (1994). Reversibility and surjectivity problems of cellular automata. *Journal of Computer and System Sciences,* 48(1), 149–182.

[21] Kari, J. (1996). Representation of reversible cellular automata with block permutations. *Mathematical Systems Theory, 29*, 47–61.

[22] Durand-Lose, J. (2001). Representing reversible cellular automata with reversible block cellular automata. In *Discrete Mathematics and Theoretical Computer Science Proceedings AA (DM-CCG)* (pp. 145–154).

[23] Sutner, K. (1995). On the computational complexity of finite cellular automata. *Journal of Computer and System Sciences,* 50(1), 87–97.

[24] Amoroso, S., Patt, Y. (1972). Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures. *Journal of Computer and System Sciences, 6*, 448.

[25] Baghor, S., Faraoun, K. M. (2016). A novel fast and secure mutual remote user authentication protocol with session key agreement using second order cellular automata.

[26] Kari, J. (1992). Cryptosystems based on reversible cellular automata [Manuscript].

[27] Palchaudhuri, P., Roy Chowdhury, D., Nandi, S., Chattopadhyay, S. (1997). *Additive cellular automata: Theory and applications* (Vol. 1). *IEEE.*

[28] Daemen, J., Govaerts, R., Vandewalle, J. (1993). A framework for the design of one-way hash functions based on a cellular automaton. In *Lecture Notes in Computer Science (Vol. 739), In:* Proceedings of Asiacrypt 1991.

[29] Bhattacharyya, D. K., et al. (1997). Efficient one-way hash function design using cellular automata. In *Proceedings of the 4th International Conference on Advanced Computing*, Chennai.