CA-KEP : A Secure CA Based 2-Party Key Exchange Protocol

Monowar H Bhuyan¹, D K Bhattacharyya¹ and J K Kalita²

¹Department of Computer Science & Engineering¹, Tezpur University
Napaam, Tezpur-784028, Assam, India.
{mhb, dkb}@tezu.ernet.in

²Department of Computer Science, University of Colorado
Colorado Springs, CO 80918, USA.
jkalita@uccs.edu

Abstract: Password based authenticated key exchange (PAKE) allows sharing short passwords between two users to agree on a cryptographically strong session key. However, the main challenge in designing this type of protocol is that it must be secure from offline dictionary and various types of man-in-the-middle attacks. In this paper, we present a fast, easily implementable, yet secure cellular automata (CA) based 2-party key exchange protocol called CA-KEP. The proposed protocol is designed based on the features of group and non-group programmable CAs. CA-KEP is superior in comparison to [1–5] considering speed and invulnerabilities to most of the known attacks.

Keywords: Protocol, key exchange, 2PAKE, 3PAKE, Cellular Automata, authentication.

I. Introduction

PAKE is an important, but simple password based authentication protocol useful for sharing and protecting from unauthorized access, information exchanged between two parties over an insecure channel in a network environment. There are many Internet applications that require password authentication only. These also include access to Internet banking, government organizations, private corporations and school systems. Bellovin and Merritt [1] presented a novel protocol relying on the counter-intuitive notion of using a secret key to encrypt a public key. Essentially, they tried to protect users with weak passwords. It has been claimed that such an approach protects networked systems easily and cost effectively. For client-server architectures, two-party PAKE protocols are suitable. However, most existing two party PAKE protocols are inconvenient and costly to use in large peer-to-peer systems. Another serious shortcoming of PAKE protocols is that most suffer from password guessing attacks [4]. Recently, several three-party password-based authenticated key exchange (PAKE) protocols have been proposed to solve the problems of two-party PAKE protocols in large scale systems [2, 3, 6]. To authenticate each other, in a three party PAKE protocol, a trusted server is used. Apart from this, the user does not need to keep a large number of secrets for a group of users. However, three major limitations of these three party PAKE protocols are (a) scalability, (b) overhead, and (c) dependency on a third party.

To address the above issues, this paper presents a fast and easily implementable, yet secure 2-party key exchange protocol (CA-KEP) based on features of linear and non-linear group and non-group CAs. CA-KEP is secure and is superior compared to [1, 2, 4, 5] in view of the following points.

• Generation of a DH-base using two secure cryptographic modules (viz., a maximum length group CA based PRSG (Pseudo Random Sequence Generator) and a non-maximum length group CA based permutation scheme) based on the biometric features of the users, rather than a small shared secret.

• The use of a non-linear MACA (Multiple Attractor CA) based hash function during the generation of $k$, the shared session key, makes CA-KEP cost effective and secure from man-in-the-middle attacks.

• Both key generation and confirmation within six steps make the protocol cost effective and fast.

• Periodic change of $S$ makes the scheme further attractive from the man-in-the-middle attacks. It is achieved by ANDing the mutually exchanged bit patterns generated from fingerprint image using z-ordering [7] by both parties as an index to a minutiae index table to initiate the generation of $S$.

• Modular, cascadable and local neighbourhood based programmable CA structure makes CA-KEP easily implementable and scalable.

The rest of the paper is organized as follows. Characteristics of PAKE protocols are discussed in Section 2. Related
work is reported in Section 3. Section 4 provides the basics of group and non-group cellular automata. In Section 5, the basic building blocks of the proposed CA-KEP protocol are described. This section also discusses the proposed protocol. A detailed security analysis is given in Section 6. Section 7 presents applications of CA-KEP. Finally, concluding remarks are given in Section 8.

II. Characteristics of PAKE Protocols

Generally, most password based authentication systems are vulnerable to dictionary attack [1]. Developing immunity from dictionary attack for a password based authentication system is a challenging task. Online dictionary attack can be easily detected and thwarted based on the number of access failures. But offline dictionary attack is more complex and requires special effort to handle. These attacks are often launched by legitimate users. They misuse their privileges for authentication or somebody else observes data or key exchange session for legitimate users to purloin information. Using tiny amounts of information leakage during an exchange, such an attack can be easily launched. Characteristics of password based authenticated key exchange (PAKE) protocols focusing on user-to-host authentication include the following:

- They use smaller (easy to remember) weak passwords for authentication.
- They try to prevent off-line dictionary attack.
- They are able to survive online dictionary attack.
- They support mutual authentication.
- They support integrated key exchange.
- They do not need any persistent record or host specific data for authentication.

The two parties know their shared secret. The protocol attempts to generate a session key for securing a subsequent authenticated session between the two parties using the password. The desirability of integrated key exchange in authentication is discussed in [8]. The participation of both parties should be an integral part of the process to ensure strongly authenticated key exchange [5]. Lack of the need for persistent record or host specific data characteristics means that users need no additional symmetric, public, or private keys.

In this paper, we develop a secure 2-party key exchange protocol where the shared secret becomes an independent factor. Thus, it simplifies the user’s side of the system. It also attempts to fulfill the desirable characteristics of the PAKE protocol.

III. Related Work

Many password based authenticated key exchange protocols that have promised increased security have been developed over the last few decades [1, 9–22, 37–40]. We classify them into two categories: (i) 2PAKE protocols and (ii) 3PAKE protocols. We discuss some of the popular protocols belonging to each of these categories in brief.

A. 2PAKE Protocols

A detailed discussion on some existing protocols based on the BPR (Bellare, Pointcheval, and Rogaway) and BMP (Boyko, MacKenzie, and Patel) models are found in [20]. The authors prove their security models based on the random-oracle and ideal-cipher models and establish that they work even in the presence of adaptive adversaries, capable of corrupting players at any time and learning their internal states. Even though the idea of the password-based authentication scheme dates back to the seminal work by Bellovin and Merritt [1], it took several years for the first formal security models to appear in the literature [12, 13]. BPR [13] was proposed as an indistinguishability-based security model for extending the framework of Bellare and Rogaway [23] while in [12] BMP was proposed as a simulation-based security model based on the framework of Shoup [24]. These authors develop formal security proofs against both passive and active adversaries, and provide secure mutually explicit authentication models.

Gennaro [18] present an improved version of the password-based authenticated key exchange protocol in the common reference string model. The protocol is based on a framework introduced by Gennaro and Lindell [16], which generalizes the KOY (Katz, Ostrovsky and Yung) key exchange protocol of Katz et. al. [25]. Both the KOY and the GL protocols use one-time signatures as a non-malleability tool in order to prevent a man-in-the-middle attack against the protocols. The efficiency of the resulting protocols is negatively affected, since they use regular signatures and require a large amount of computation (almost as much as the rest of the protocols) and need further computational assumptions. If one-time signatures are used, they substantially increase the bandwidth requirement. Generally, protocols for PAKE allow two users who share only a short, low-entropy password to agree on a cryptographically strong session key. The main challenge in designing such protocols is that they must be immune to off-line dictionary attack in which an eavesdropping adversary exhaustively enumerates a dictionary of likely passwords in an attempt to match a password to the set of observed transcripts. Groce et. al. [22] generalize a protocol introduced by [14] to give a new methodology for realizing a PAKE protocol in the common reference string model. In addition to giving a new approach to the problem, the resulting construction offers several advantages over prior work. The authors also describe an extension of their protocol that is secure within the universal composability (UC) framework. It is shown to be more effective in comparison to a previous protocol of Canetti et al. [26]. To address the existing challenges, Saeed et al. [39] propose another improved version of two-party PAKE protocol.

Advantages and limitations of 2PAKE: Password guessing attacks are classified into three different categories [4, 27]: (a) detectable online password guessing attacks, (b) undetectable online password guessing attacks and (c) off-line password guessing attacks. A 2PAKE protocol is often incapable of handling all three attacks.

The advantages of 2PAKE protocols include the following:

- They protect information from unauthorized parties.
- They facilitate exchange of messages over insecure
channels.

- They are useful in many applications such as bootstrapping new system installations, diskless workstations, and user-to-user applications.

The limitations of 2PAKE protocols include the following.

- They are vulnerable to man-in-the-middle attack.
- Users need to remember the human-readable password with low entropy, which is simple and efficient. However, if chosen passwords are of low entropy, it is not a trivial task to protect against password guessing attacks.

To address the limitations of the 2-party PAKE protocols, significant efforts have been made to develop three-party password-based key exchange (3PAKE) protocols, where a client is allowed to share a human memorizable password with a trusted server such that two clients can agree on a secret session key for secure connectivity.

B. 3PAKE Protocols

Recently, several 3PAKE protocols [2–4, 6, 28, 37] have been developed. Lin et al. [6] report a new three-party EKE protocol and claim that their protocol is secure against both offline guessing attacks and undetectable on-line guessing attacks. The protocol also supports the properties of perfect forward secrecy and known-key security. Lu and Cao [2] present a three party PAKE protocol known as S-3PAKE using the server’s public key. The authors report that the protocol resists various known attacks, viz., trivial attacks, online guessing attacks, off-line guessing attacks and replay attacks. The protocol also provides perfect forward secrecy and known key security. The work of Kim and Choi [4] and Lu and Cao [2] establish that the STPKE (simple three-party password-based key exchange) protocol is still vulnerable to undetectable on-line password guessing attacks by using the formal description BPR model. It is due to the fact that the messages of the communicants are not appropriately encrypted into the exchanged cryptographic messages. Therefore, they suggest a countermeasure to resist their attacks while keeping the original protocol unchanged. However, not all proposed protocols can simultaneously achieve high security and efficiency. Without using server’s public key, Huang [3] proposes a simple three-party password-based authenticated key exchange scheme and claims that the proposed scheme is not only efficient, but also secure. Huang also claims that the proposed HS-3PAKE protocol [3] is not only secure against various attacks, but also more efficient than the previously proposed 3PAKE protocols. However, Yoon and Yoo [28] demonstrate that HS-3PAKE protocol is vulnerable to undetectable online password guessing attacks [29] as well as off-line password guessing attacks [4, 27]. Zeng et al. [40] suggest an improved version of a three party protocol based on weil pairing. Again, another communication efficient 3PAKE protocol can be found in [37]. This protocol is developed mainly based on LHL-3PAKE protocol, which does not require either a symmetric cryptosystem or the necessity of server public keys.

C. Discussion

Based on our survey of 2PAKE and 3PAKE protocols, we observe the following.

- As chosen passwords are usually of low entropy, it is not trivial to protect password information against password guessing attack (dictionary attack).
- 3PAKE protocols are vulnerable to man-in-the-middle attack. An adversary can sit between the client and the server, intercepting all traffic and altering or deleting messages at will.
- The KOY protocol relies on the DDH assumption. Gennaro and Lindell [16] demonstrate additional instantiations of the framework and develop a password-based authenticated key exchange protocol that is more effective.
- Each party does not need to remember and manage multiple passwords in a 3PAKE protocol but shares only a single password with a trusted server to assist clients in establishing a session key by providing authentication services to them.
- The S-3PAKE protocol [2] is potentially vulnerable to a man-in-the-middle attack as described by [30].

To address these issues, in this paper we present a secure and easily implementable 2-party key exchange protocol, called CA-KEP. The proposed protocol is designed by utilizing features of group and non-group CAs. Next, we report the basics of these CAs [31–34].

IV. Basics of Group and Non-Group CA

Cellular Automata (CA) have already been established as an effective tool for a wide range of applications in various fields such as Computer Science, Physics, Biology, Chemistry, Mathematics, Social Sciences and Engineering. Researchers have also explored the applications of CA rules in cryptography and network security. In this work, we use CA rules in designing a PAKE protocol by utilizing the non-group and group or periodicity properties of 2-stage, 3-neighbourhood and extended neighbourhood CAs.

A. CA Basics

CAs represent an idealized parallel processing machine consisting of a number of cells containing values called states together with an updating rule. A cell value is updated using this updating rule, which involves the cell value as well as values of other cells in its neighbourhood. A CA is completely defined with the following five parameters:

- the number of states,
- the size of the neighbourhood,
- the length of the CA,
- the guiding rules (uniform or hybrid), and
- the number of times the evolution of the automata takes.
The structure of a CA can be viewed as a discrete lattice of cells, where each can assume either the value 0 or 1 [31]. At discrete clock cycles, evolution of the cell value takes place using a rule (i.e., a combinational function), which is a function of the present state of \( k \) of its neighbours for a \( k \)-neighbourhood CA. For a 2-state, 3-neighbourhood CA, the evolution of the \( i^{th} \) cell can be represented as a function of the present states of \((i - 1)^{th}, i^{th} \) and \((i + 1)^{th} \) cells as:
\[
q_i(t + 1) = f(q_{i-1}(t), q_i(t), q_{i+1}(t)),
\]
where \( f \) represents a combinational function. In effect, each cell as shown in Figure 1, consists of a storage element or flip-flop \( D \) and a combinational logic implementing the next state function.

In Cellular Automata, if there are \( 2^3 \) distinct neighbourhood configurations and \( 2^3 \) distinct mappings from all these neighbourhood configurations to the next state, each mapping represents a CA rule.

The top row gives all the 8 possible states of the 3 neighbouring cells at time instant \( t \), while the second and third rows present the corresponding states of the \( i^{th} \) cell at time instant \( t + 1 \) for two consecutive CA rules. A detailed discussion of CA rules with truth tables is found in [31]. If the same CA rules are applied to all cells, the CA is known as uniform CA, otherwise the CA is known as hybrid CA. If a CA rule involves only XOR logic, it is called a linear CA rule. If a rule involves XNOR logic, it is referred to as a complemented rule. If each cell contains only linear rules, it is called a linear CA, whereas a CA with non-linear rules (i.e., involving AND-OR logic) is called a non-linear CA.

The state transition diagram for a cellular automata is characterized using a matrix \( T \) and its characteristic polynomial. Depending on the nature of their state transition behaviour, CAs are broadly classified into two categories.

(a) **Additive/Linear CAs**: CAs that follow the constraints of XOR/XNOR logic for its next state function.

(b) **Non-linear CAs**: CAs which do not possess such constraints.

Algebraic methods can be used in the analysis of linear/additive CA. Since the next state function applied at each cell follows the operations of a Galois field, the properties of the field can be applied to characterize its state transition behaviour. The minimal and characteristic polynomials of the characteristic matrix show various interesting features of CA behaviour [35]. Linear/additive CAs can be classified into two (i) **Group CA** and (ii) **Non-group CA**.

### B. Group CA and Its properties

An effective application of null boundary group CA has been proposed for pseudo-random pattern generation [31]. A maximum-length group CA with all non-zero states lying in a single cycle produces high quality pseudo-random patterns. It has been established that the maximum length cycle can be produced only if the characteristic polynomial is primitive [35]. A maximum length CA cannot be generated by a periodic boundary CA since its characteristic polynomial can be factorized.

In the state transition diagram of a group CA, all states are cyclic. Each state has a unique successor and a unique predecessor state. The state transition diagram of two group CAs are shown in Figure 2 and Figure 3. A detailed characterization of these CAs based on matrix algebra can be found [31]. Next we characterize a general class of group CA called Extended Neighbourhood CA (ENCA), which we use in the design of the present schemes.

![Figure 1: A typical CA rule](image)

The characterization of ENCA (Extended Neighbourhood CA) is based on extensions of the group and non-group 3-neighbourhood additive CA [31]. If the ENCA is a non-singular CA with a finite set of elements and the entire set of states ultimately converges in a single cycle or in multiple cycles, it is called a group ENCA; otherwise, it is called non-group ENCA. Both these categories of ENCAs have important properties, which can be successfully utilized in the field of cryptography for pseudo-random pattern generation and hashing.

In general, an ENCA generates trajectories, starting from some elements which one can call non-reachable or starting states, because they have no predecessors. The trajectories may have different starting points and end in possibly more than one cycles where the lengths of a cycle may even be only one, i.e., the cycle contains only one attractor. In other words, an ENCA partitions the elements of the opened...
set into partially ordered sets (POSET), referred to as basins, where the structure of each basin is characterized by the type of CA chosen. An example of ENCA with starting state $\alpha_0$ and with periodicity 7 is shown in Figure 4.

**Figure 4:** Basin structure showing aperiodic behaviour

These basin structures exhibit some peculiar characteristics which can be of use in different contexts. A detailed analysis of these basin structures induced by group and non-group ENCA is found in [31]. Two examples for non-maximum and maximum length group ENCA, respectively are presented next. **Example 1:** Consider the characteristics matrix

$$
T = \begin{bmatrix}
1 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 1
\end{bmatrix}
$$

We see that $T^3 = I$ (i.e., the identity matrix) and the characteristic equation is $x^5 = x^4 + x^3 + x^2 + x + 1$. Also, $x^6 = 1$, i.e., $T^6 = I$. Hence, $T^2 \cup T^4 \cup T^6 = I$, i.e., length of cycles either may be 2 or 3 or 6. We can verify that $T^2 \neq I$ but $T^4 = I$. It follows that there can be cycles of length 3 only. Also since $T$ is non-singular, every state has only one parent (possibly itself). Hence, in the basin structure induced by this CA, there can be either attractors or cycles of length three only. Every state is either its own parent or it is a member of a 3-cycle. To obtain all the attractors of $T$, one has to only find the solutions of $(T + I)\Delta = 0$, where $\Delta$ is a boolean variable. The only possible solutions are $(\Delta 0 \Delta \Delta \Delta \Delta')$. Thus, there are only two attractors, viz., 0 itself and 23. The other 30 states in the space are partitioned into 10 ($= 30/3$) cycles as follows.

\{1, 5, 14\}, \{2, 18, 7\}, \{3, 29, 9\}, \{4, 31, 27\} \cdots

**Example 2:** Consider the characteristic matrix

$$
T = \begin{bmatrix}
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 0
\end{bmatrix}.
$$

We see that $T$ is non-singular and the characteristic equation is $x^6 = x^5 + x^3 + x^2 + x + 1 = 0$. It does not possess any attractor other than 0. Also, we have $x^{31} = I$. So, this CA gives rise to a fully periodic basin (maximum length cycle) of all 31 non-zero states, as shown in Figure 6.

We select a group ENCA $T$ of size $S$ and a complement vector $S_c$ of the same size, which generate say, $p$ number of cycles, i.e., $B_1, B_2, \cdots B_p$. Each state with a repeated multiplication of $T$ and exclusive ORing with $S_c$ results in $p$ different cycles shown in the state transition diagram. These cycle structures induced by $T$ and $S_c$ can be utilized for encryption of any binary file that does not have more than $p$ distinct $r$-bit blocks. During encryption, for each distinct $r$-bit block, a unique cycle is identified and is mapped to any element taken at random (with equal chance, for example) from this cycle. The decryption process also uses the same CAs. By a repeated application of $T$ and $S_c$ on the ciphertext blocks, one will get the smallest element in the cycle, which is uniquely associated with a plaintext block and hence decrypted.

A permutation $\pi$ of a finite set $V$ is defined to be an injection, $\pi : V \longrightarrow W$, where, $V = \{u_1, u_2, \cdots, u_d\}$. The set of all possible permutations on $V$ form a non-commutative group $G_d$ of order $d!$ under the operation of permutation multiplication. One distinct feature of permutation is its cyclic structure, i.e., for any permutation $\pi$ in $V$, one find $\pi(z_j) = z_{j+1}$, where, $j = 1, 2, \cdots, (l - 1)$ for a cycle $(z_1, z_2, \cdots, z_l)$ of length $l$. It has been established that every permutation can be uniquely expressed as a product of disjoint cycles. An even permutation is one that is expressible as a product of an even number of transpositions; otherwise a permutation is called an odd permutation. It is a well-known result [34] that all even permutations on $V$ form a normal subgroup $G_d$, which is the alternating group, say $A_d$ of degree $d$ and order $d!/2$.

Consider a CA, say $T_1$ which through repeated application upon the elements of a finite set, say, $F$ over GF(2), ultimately ends in one of $p$ different cycles. Every permutation can be uniquely expressed as a product of these $p$ disjoint cycles. Let us consider an $r$-dimensional vector space $V_r$, which consists of all the message blocks, each $r$-bit long. Here, each $r$-bit message block $b_i$ is subjected to a permutation operation. We apply the group ENCA based permutation scheme to generate the intermediate message.

These new pseudo random pattern generators also have im-
plication advantages when generating pseudo random numbers in that they can be designed to require only adja-
cent neighbour communication. They are also cascadable, i.e., the physical length of the generator can be increased or
decreased by simply adding or subtracting cells. It should be
noted that the area of each CA cell is somewhat larger than
an LFSR cell. This means that a CA-based test pattern gener-
ator is very appropriate for incorporation in a CAD tool. For
physical length \( n \), where \( n \) is the number of cells or bits in the
test pattern generator, we have a maximum cycle length of
\( 2^n - 1 \). As the length of the CA increases the maximum poss-
ible cycle length of the pseudo random sequence generally
increases, but this growth is not monotonic and the seed, or
initial state, used in the CA affects the length of the sequence
produced. The path length for cyclic boundary condition-
s increases rapidly, mainly due to the increase in maximum
cycle length. For null boundary conditions, the maximum cycle
length does not rapidly increase with CA size and we
see that the maximum possible path length becomes nearly
constant as the size of CA increases. Thus the paths created
have null boundary.

C. Non-Group CA and Its Properties

The non-group CA is a degenerate case of a non-singular
(group) matrix. The isomorphism of the tree structures of
non-group CA has two important consequences: first, the
non-group CA can be mapped to a table structure with its
cyclic states producing the address of the table and second,
the linear and complemented variants of a non-group CA pro-
duces interesting symmetry within themselves. Non-group
CAs have more potential applications than group CAs. Interest-
ing classes of non-group CA which have been extensively
used are: multiple attractor cellular automata (MACA),
depth-1 cellular automata (D1\*CA) and single attractor cellular
automata (SACA). They have been used in a wide range
of functions such as hashing, classification, designing easy
and fully testable FSM, and authentication.

Non-group ENCA from a special class of CA, in which some
of the states are not reachable from any other state. In con-
trast to the group ENCA, (i) in the state transition graph of
such CA, the reachable states can have multiple predeces-
sors, and (ii) the characteristic matrix is singular in nature.
The study of non-group CA behaviour has not received suf-
icient attention. From the state transition graph of such non-
group ENCA, one can see cyclic states, i.e., one or more cy-
cles. Other states form inverted trees rooted at one of the
cyclic states. Such inverted trees are referred to here simply
as trees. The cycles in the state transition diagram of a non-
group ENCA are referred to as attractors. Thus an attractor
is a cycle of length \( l, l \geq 1 \). The tree rooted at a cyclic-state
\( \alpha \) is denoted as \( \alpha \)-basin. The depth of such a basin is defined to be the minimum number of clock cycles [31] required to
reach the nearest cyclic state from any non-reachable state in
the state-transition graph. Example 3 illustrates the construc-
tion of a basin structure due to a 5-cell non-group ENCA.

Thus, we have the characteristic equation: \( x^5 + x^4 + x^3 = 0 \)
and here, \( x^5 = x^3 \). Hence, \( T_6 = T_3 \). Thus, for any
state \( S \), one has \( T_3 \). \( S = T_3.S \) and any sequence
\( \{S_n|S_n = T.S_{n-1}\} \) will have at most only 5 distinct states,
the sixth state \( S_6 \) in the sequence being identical with \( S_3 \).
The sequence is ultimately periodic with periodicity either 3
or a factor of 3. Of course, since 3 is prime, the ultimate pe-
riodicity is either 1 or 3. Again, since \( x^5 + x^4 + x^3 = 0 \), the
only admissible eigenvalue is \( x = 0 \). Solving the equation
\( T.S = 0 \), we have \( T[00101]^3 = T.5 = 0 \) as the only solution,
other than 0, of \( T.S = 0 \). Also, \( T.S = b \) has no solution if
\( b_1 + b_2 + b_4 = 1 \), i.e., there are \( 2^2 \cdot (3/1 + 3/1) = 16 \)
states obtained by choosing components \( b_1, b_2 \) and \( b_4 \). The
value of each component is either one of them or all are 1.
The elements \( b_2 \) and \( b_0 \) being arbitrarily 0 to 1, which cannot
be reached from any other state. These only can be the
non-reachable states of a trajectory. Since \( T.5 = 0 \), we have
\( T.(S + 5) = T.S \), only for all \( S \). Hence if \( S \) has \( T.S = b \),
\( T.(S + 5) = b \) also, i.e., for any state in the trajectory other
than a starting state, there are two parents, which differ by
the state 5. Thus, for instance, since \( T.28 = 1 \) we have
\( T.(28 + 5 = 25) = 1 \). Also, since all the columns of \( T \) have
been used as a parent, with its progeny obtained, one can compute \( T^2, T^3, \cdots \) by simple table lookup. The struc-
ture and contents of the table are as shown in Table 1. Again
from the characteristic equation, i.e., \( x^5 + x^4 + x^3 = 0 \), we
observe that \( x = 1 \) is not a root, and therefore, there is no
\( S \) such that \( T.S = S \). Also, we have \( T^6 = T^3 \). Then the
periodicity is \((6 - 3) = 3\). So, cycles of length 3 can exist as
the terminal points of the trajectories.

Example 3: Consider the CA

\[
T = \begin{bmatrix}
1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

Since each offspring has only two parents, \( T.S = b \), if at all solvable, has only two solutions, and since each element in a
cycle necessarily has one parent in the cycle, there should be
another parent to it, which is not in the cycle. If the parent is
itself not a starting state, it should have exactly two parents.
Further from \( T^3 = \{15, 0, 4, 15, 4\} \), we find, for \( (T^3 + I) \)
that there are exactly four eigenvalues of \( T^3 \), including the

![Figure. 6: State transition diagram of maximum length group ENCA](image-url)
trivial 0 state. Hence, there is only one cycle of length 3 (for illustration, see Figure 7).

MACA is a special class of non-group CA. A subclass of MACA has been established to be an efficient hashing function generator. The state transition graph of an example of MACA is found in [31]. The graph consists of a set of distinct components, each component containing equal numbers of inverted states. The CA state can be viewed simply as a vector of key values. If the CA is loaded with a particular key and allowed to run autonomously for a number of cycles to the depth of such tree, it evolves through a number of states before reaching the root of the corresponding tree. A root is not a cyclic state with a self-loop and is referred to as an attractor. The set of attractors of all these basins contains all the PE-bit patterns at certain bit positions. Each PE-bit pattern of the root states can be used as an identifier of the basin. Thus, by selecting the particular configuration with the required T matrix, it is possible to get CA with varying number of attractors [31].

CA-KEP uses a CA-based hashing scheme, designed considering the properties of group, non-group and non-linear CAs. The protocol hashes a user input or persistent piece of information on disk, rather than being computed based on the properties of the user’s fingerprint, it avoids the stolen key attack.

(ii) Suitable Base: The use of randomized, expanded and permuted S as the DH-base avoids the dictionary attack.

(iii) Compactness: Both key generation and confirmation are performed in six steps.

(iv) Periodic change of S: It avoids replay attack.

Like other PAKE protocols, it is also based on a small shared r-bit secret, S, which is generated by a module M based on the transformation invariant features (true minutiae and their positions) of the user’s fingerprint image. An illustration of M is given in Figure 8. It accepts the user’s fingerprint as input, extracts the transformation invariant features, and generates an n-bit pattern using z ordering technique [7] (one dimensional grid approximation of the minutiae positions). For each minutiae position an n-bit pattern is generated, where n depends on the level of grids of the image space. Finally, it combines all the n-bit patterns in order to generate S. A sample index table is shared by both the parties to generate S (shown in Table 2).

<table>
<thead>
<tr>
<th>Index bits</th>
<th>Bits pattern for minutiae position</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>11100000111111011110001111110011</td>
</tr>
<tr>
<td>00001</td>
<td>11011101101111101111010110111011</td>
</tr>
<tr>
<td>00010</td>
<td>0011010000010111100000111111100</td>
</tr>
<tr>
<td>00011</td>
<td>10000101101101111110010001010101</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>11111</td>
<td>01011110101111101010010111110001</td>
</tr>
</tbody>
</table>

To convert S into a suitable base for the exponentiation operation, CA-KEP applies a function F to randomize, expand and permute the r-bit password S into S". F comprises two cryptographic functions: f1 and f2, where f1 is a maximum-length group CA based PRSG, whereas f2 is a non-maximum

D. Discussion

The properties exhibited by group and non-group CAs are advantageous in the design of a PAKE protocol considering the following.

- Maximum length group CAs with all non-zero states lying in a single cycle produce high quality pseudo-random patterns.

- MACA is important because of its strong one-way properties and hence useful in the digest generation.

V. CA-KEP: The Proposed Protocol

This section describes the proposed CA-KEP protocol with necessary illustrations. Like others, CA-KEP also comprises of two phases: key generation and key confirmation. The major attractions of CA-KEP are the following.

(i) Immunity from stolen-key attack: Since S is not a user input or persistent piece of information on disk, rather it is computed based on the features of users fingerprint, it avoids the stolen key attack.

(ii) Suitable Base: The use of randomized, expanded and permuted S as the DH-base avoids the dictionary attack.

(iii) Compactness: Both key generation and confirmation are performed in six steps.

(iv) Periodic change of S: It avoids replay attack.

Like other PAKE protocols, it is also based on a small shared r-bit secret, S, which is generated by a module M based on the transformation invariant features (true minutiae and their positions) of the user’s fingerprint image. An illustration of M is given in Figure 8. It accepts the user’s fingerprint as input, extracts the transformation invariant features, and generates an n-bit pattern using z ordering technique [7] (one-dimensional grid approximation of the minutiae positions). For each minutiae position an n-bit pattern is generated, where n depends on the level of grids of the image space. Finally, it combines all the n-bit patterns in order to generate S. A sample index table is shared by both the parties to generate S (shown in Table 2).

<table>
<thead>
<tr>
<th>Index bits</th>
<th>Bits pattern for minutiae position</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>11100000111111011110001111110011</td>
</tr>
<tr>
<td>00001</td>
<td>11011101101111101111010110111011</td>
</tr>
<tr>
<td>00010</td>
<td>0011010000010111100000111111100</td>
</tr>
<tr>
<td>00011</td>
<td>10000101101101111110010001010101</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>11111</td>
<td>01011110101111101010010111110001</td>
</tr>
</tbody>
</table>

To convert S into a suitable base for the exponentiation operation, CA-KEP applies a function F to randomize, expand and permute the r-bit password S into S". F comprises two cryptographic functions: f1 and f2, where f1 is a maximum-length group CA based PRSG, whereas f2 is a non-maximum
Key Generation

The proposed protocol are given below. CA-KEP requires six steps to accomplish both the key generation and key confirmation phases. The main steps of CA-KEP are presented in Table 3.

CA-KEP requires six steps to accomplish both the key generation and key confirmation phases. The main steps of the proposed protocol are given below.

Key Generation

1. Alice computes: \( Q_A = F(S)^{R_A} \mod p \),
   \( A \rightarrow B : E_B(Q_A, h(Q_A), R_A) \), using B’s public key

2. Bob computes: \( Q_B = F(S)^{R_B} \mod p \)

S3. Alice computes: \( k = h(Q_B^{R_A} \mod p) \)

S4. Bob computes: \( k = h(Q_A^{R_B} \mod p) \)

Key Confirmation

S5. Alice verifies: \( A \rightarrow B : h(R_B - 1) \)
S6. Bob verifies: \( B \rightarrow A : h(R_A - 1) \)

The three basic modules used by CA-KEP (i) to generate \( S \) based on fingerprint features, (ii) to generate expanded, randomized and permuted \( S \) based on \( F \), and (iii) MACA based hashing, are discussed next.

1) Module: \( M \)

This module is used to generate \( S \). An illustration of this module is depicted in Figure 8. It provides an option for periodic change of the selection of the point of initiation for extraction of the position-wise true minutiae information from user’s fingerprint image. The programmable structure of MACA further facilitates the periodic replacement of \( S \) by both the parties.

2) Cryptographic Expansion Function: \( F \)

It comprises of two submodules: \( f_1 \) and \( f_2 \). An illustration of \( F \) is shown in Figure 9. We discuss the design of \( f_1 \) and \( f_2 \) in brief.

(a) Maximum Length Group CA Based PRSG (\( f_1 \)): Linear CA-based PRSG can be designed with an \( n \) cell maximum length CA, where all the \( (2^n - 1) \) non-zero patterns can be generated starting from any initial seed. It has already been established that CA-based PRSG is better than conventional random generators in terms of the pseudo-randomness quality of the generated patterns [31]. The cascadable structure of CA involving non-linear rule based PRSG is useful in designing a secure protocol by generating a large number of non-repeated states or patterns. An illustration is given in Figure 10 with generated patterns \( (p_1, p_2, \ldots, p_i, \ldots) \). For a group CA of size \( r \times r \), the length of the maximal-length cycle is exponentially large at \( (2^r - 1) \). The probability of occurrence of a pattern, while running a maximal
length CA is $1/2^r$ [31]. However, the complexity of synthesizing an $n$-cell maximal length CA is $O(n^3)$.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>module to preprocess and extract fingerprint features to generate an $r$-bit shared secret, $S$</td>
</tr>
<tr>
<td>$f_u$</td>
<td>fingerprint of the user mutually shared by both the parties</td>
</tr>
<tr>
<td>$S$</td>
<td>a small ($r$-bit) shared secret</td>
</tr>
<tr>
<td>$p$</td>
<td>modulus, a huge (2048 bits) prime number</td>
</tr>
<tr>
<td>$q$</td>
<td>a large prime factor of $p-1$</td>
</tr>
<tr>
<td>$F$</td>
<td>a two phase group CA based cryptographic module to generate a large ($2r$ bit) shared secret based on $S$</td>
</tr>
<tr>
<td>$R_A, R_B$</td>
<td>random nonces chosen by Alice and Bob</td>
</tr>
<tr>
<td>$Q_A, Q_B$</td>
<td>exponential values sent by Alice and Bob</td>
</tr>
<tr>
<td>$E_A(m), E_B(m)$</td>
<td>RSA based encryption of $m$ using the public keys of A and B shared between them</td>
</tr>
<tr>
<td>$h$</td>
<td>non-group MACA based one-way hash function</td>
</tr>
<tr>
<td>$A \rightarrow B : m$</td>
<td>Alice sends $m$ to Bob</td>
</tr>
<tr>
<td>$k$</td>
<td>generated session key</td>
</tr>
</tbody>
</table>

Table 3: Symbols used

In our protocol, the function $f_1$ is the maximum-length group CA-based PRSG with $(2^r - 1)$ maximum-length cycle. We need a $r$-bit $S$ as the initial seed to generate a $2r$-bit pseudo-random pattern. However, in common applications, it is impractical to run the full length $(2^r - 1)$ cycle of a CA to generate pseudo-random patterns. For a large value of $n$, rather than searching for a maximal length cycle, a CA with a large cycle length with its rules suffices to maintain the high quality of pseudo-randomness for the pseudo random pattern generator (PRPG), which is immune from dictionary attack.

(b) Non-maximal Length Group CA-based Permutation ($f_2$): Assume that the ENCA is non-singular. By repeated application of CA with a finite set of elements, the entire set of states ultimately converges to a single cycle or multiple cycles. It is referred to as a group ENCA; otherwise, it is called non-group ENCA. ENCA can be successfully used in cryptography for PRSG and hashing [34]. ENCA partitions the elements of the opened set into partially ordered sets (POSET), referred to as basins, where the structure of each basin is characterized by the type of CA chosen.

In the case of our protocol, the function $f_2$ uses the output of $f_1$, i.e., a $2r$-bit $S'$ as input to the non-maximal length group CA based permutation submodule to generate the final output, i.e., $S''$. Both submodules help to enhance the avalanche effect on $S''$ significantly (at least 2-bit) due to change of single bit of $S$. To support this fact, we provide a theorem in the following.

**Theorem 1.** A change of 1-bit in $S$ causes change of at least 2-bits in $S''$.

**Proof.** Here, the $2r$-bit $S''$ is generated by $f_2$ based on the $2r$-bit input $S'$, which again is generated by $f_1$ based on the $r$-bit input $S$. Since $f_1$ is a qualitative PRSG and since it generates a random pattern twice the size of the input bit pattern $S$, it ensures that $S'$ differs from $S$ by at least 2-bits. Again, since $f_2$ is a CA based permutation scheme, which enables an one-to-one mapping (among the elements of the non-maximum length cycle structure) between $S'$ and $S''$, it ensures that there’s at least 1-bit change in $S''$ due to change of 1-bit in $S'$. Hence, it ensures that a change of 1-bit in $S$ causes change of at least 2-bits in $S''$. □

3) Non-linear MACA based Hash Function: $h$

CA-KEP uses a non-linear MACA based hashing scheme, which is designed by deriving the benefits of one-way properties of non-linear CAs. This scheme hashes a $2r$-bit message into an $r$-bit hash value in feedback mode. It also exploits the complement operation to incorporate non-linearity during the hashing process. A detailed design of an MACA based hash function is available in [31].

4) Periodic Change of $S$

Another attractive feature of CA-KEP is that it enables $S$ to be changed periodically. In order to facilitate such changes, the protocol requires an additional piece of information (i.e., a bit pattern generated based on the minutiae position shown above in the minutiae index table) to be mutually exchanged. The mutually exchanged bit patterns is ANDed by both the parties to determine the index to the minutiae index table to initiate the generation of $S$. However, during extraction of the minutiae position bit pattern, it is essential to do the necessary alignment to avoid the position ambiguity of the fingerprint image.

**Figure. 10: CA states as PRSG**
B. Discussion

Based on our survey, we pick four parameters to compare the proposed protocol with several existing 2PAKE and 3PAKE protocols. See Table 4. We discuss how the various protocols compare.

(a) No of steps: Like most existing protocols, the proposed protocol requires four steps to generate the key. However, some protocols need four steps to complete the confirmation phase, whereas the proposed protocol requires only two steps. Hence, CA-KEP is more compact.

(b) Use of $S$: Based on the usage of $S$ in the key generation process, the existing protocols can be categorized as follows.

- $S$ as a DH-base: Some of the existing protocols [2, 3, 5, 12, 15, 34] use $S$ as the base for the exponentiation operation. However, depending on the suitability of $S$, such usage may lead to several types of serious attacks. One such attack is the partition attack, caused mainly due to leakage of information. When an arbitrarily small number is used as base, it may lead to serious vulnerability, especially due to its group confinement.

- $S$ as an Exponent: Some other protocols [5, 34] which exploit $S$ as an exponent may suffer from the discrete log attack. Any leakage of information about $S$ (even a single bit) may be devastating.

- $S$ as an Encryption Key: While transmitting $Q_A$ and $Q_B$ by both the parties, some protocols [2, 5, 12, 13, 16, 18, 22, 26, 27, 34] use encryption technique for sending the key. Apart from the usual limitations of secret key cipher, such protocols also may suffer from (i) stolen key attack, (ii) partition attack, and (iii) low exponent (if $S$ is small) attack.

However, the proposed CA-KEP protocol is free from all these limitations, since here $S$ is neither a persistent shared secret (rather computed by both the parties independently) nor is used directly as the base nor as a key for encrypted exchange.

(c) Encrypted Data Exchange: As seen in the table (columns 6 and 7) the existing protocols either use secret key or public key encryption during intermediate data exchange. However, as discussed in [5], public key based data exchange is always advantageous than the others. The proposed protocol also takes advantage of the public key based data encryption scheme while transmitting the intermediate data.

(d) Handling of known Attacks: The proposed CA-KEP protocol is capable of handling all known attacks reported in the table. A security analysis is presented below.

VI. Security Analysis

The proposed CA-KEP is developed based on the DH-base. The CA-KEP protocol is shown invulnerable to all known attacks such as (i) offline dictionary attack, (ii) online dictionary attack, (iii) man-in-the-middle attack, (iv) replay attack, (v) forward security, (vi) forgery attack, and (vii) discrete logarithm attack as per the security analysis in [36]. Below, we give a brief discussion on each of these attacks.

Definition 1. A secure hash function, $h() : x \rightarrow y$ is an MACA based one-way hash function, if given $x$, it is easy to compute $h(x) = y$, however, given $y$, it is hard to compute $h^{-1}(y) = x$.

Definition 2. $S_i$ is a small ($r$-bit) shared secret generated based on the features of user’s fingerprint, $f_u$. If $f_u$ and $M$ are known, it is easy to generate $S_i$ but $S_i \neq S_{i+1}$, where $S_{i+1}$ is the shared secret generated for the next session.

Definition 3. The DH-base $F(S)$ is a 2r-bit pattern generated based on $r$-bit $S$ by applying $f_1$ (a CA-based PRSG) and $f_2$ (a CA-based permutation function). Here, change of $1$-bit in $S$ leads to a different $F(S)$ with at least 2r-bit change.

Definition 4. $k$ is a large session key generated mutually based on $R_A$, $R_B$ and $F(S)$ which are changed for each session.

In our proposed protocol, $A$ applies the shared secret $S$ after a secure cryptographic transformation into $F(S)$ in the generation of the intermediate secret $Q_A$ and sends it along with its hash value and $R_A$, i.e., $E_B(Q_A, h(Q_A), R_A)$ to $B$, where $E_B$ represents the encryption using $B$’s public key. On the other hand, $B$ encrypts $(Q_B, h(Q_B), R_B)$ with $A$’s public key and sends it to $A$. In the next step, it computes the common session key $k = h(Q_A) \mod p$ and checks whether $A$ verifies $h(R_A - 1)$ or not and $B$ verifies $h(R_A - 1)$ or not. Without knowing $R_A$, $R_B$ and $F(S)$, it is not helpful for the adversary to obtain confirmation of the valid common session key $k$ for the users $A$ and $B$, and hence an adversary cannot impersonate valid users. Therefore, the proposed protocol provides secure connectivity between the users $A$ and $B$.

The detailed mathematical proofs of the concreteness of the protocol as whole is out of the scope of this paper. However, some more properties based on the specific attacks of the proposed protocol are given below.

A. Offline Dictionary Attack

The proposed CA-KEP is safe from this attack because even if an adversary extracts the large intermediate values, i.e., $Q_A$ and $Q_B$ by managing the private keys of recipients, it is extremely difficult for him/her to derive $S$, $R_A$ and $R_B$. Since $R_A$ and $R_B$ are large randomly generated numbers, guessing them correctly for the exact estimation of $S$ is exceedingly difficult. Again, $S$ is not directly used as the base for the exponentiation operation, rather it depends on a cryptographic function $F$, which is again a combination of two cryptographic secure function, i.e., $f_1$ and $f_2$ (established to be secure in subsection V.A.2). Therefore, our proposed protocol can thwart offline guessing attack. Now, we present a theorem to establish that the proposed CA-KEP protocol is safe from stolen key attack.

Theorem 2. The CA-KEP protocol is immune from stolen key attack.

Proof. CA-KEP uses a cryptographically secure, large DH-base for the exponentiation operation during key generation. The base is neither a persistent piece of information nor a
small shared secret. It is generated using two graphically secure modules, \( M \) and \( F \). \( M \) exploits a MACA based one-way hash function to generate the \( r \)-bit shared secret \( S \), which requires cracking complexity of the order of \( O(2^r) \times O(2^{2S}) \), while the function \( F \) offers a cracking complexity for the generation of \( 2r \)-bit \( S' \) of the order of \( O(r^2) \times O(2^{2S'}) \), an extremely large number! Hence, CA-KEP is immune from stolen key attack.

### B. Online Dictionary Attack

To mount a successful online guessing attack, initially, the adversary needs to change the recipient’s private key to extract the values of \( Q_A \) and \( Q_B \). Since the recipient is also provided an additional provision for verifying the integrity of \( Q_A \) based on \( h(Q_A) \) where \( h \) is secure one way hash function, any improper or incorrect guess of \( Q_A \) or \( Q_B \) is easily detected, even if an adversary manages to guess \( Q_A \) and \( Q_B \) correctly. He/she needs to guess \( R_A, R_B \) and \( F(S) \) correctly. But, the size of \( R_A \) or \( R_B \) is \( 2r \) bits and \( F(S) \) is estimated by applying cryptographically secure function \( F \) on \( S \). Even if \( Q_A \) and \( Q_B \) are extracted, guessing \( R_A, R_B \) and \( F(S) \) correctly is extremely difficult for an online guessing attack. Hence, it is not possible in real-time and must be an offline guessing attack.

### C. Man-in-the-middle Attack

In this attack, an adversary say \( C \) attempts to impersonate either \( B \) or \( A \) to communicate with either \( A \) or \( B \) by capturing either the communication \( E_B(Q_A, h(Q_A), R_B) \) or \( E_A(Q_B, h(Q_B), R_A) \). Even if \( C \) attempts to send a substituted message to either \( A \) or \( B \) by encrypting with their public key, the legitimate party will be detected during Step 5 and Step 6. Moreover, during Step 3 and Step 4, it leads to two different \( k \) values. Also, guessing the correct \( F(S) \) is extremely difficult in real-time due to its extremely large key space \( (\approx 2^{2S}) \) and due to change of \( R_A, R_B \) and \( F(S) \) for each session. Hence, the proposed CA-KEP can easily resist on the man-in-the-middle attack.

### D. Replay Attack

The proposed CA-KEP generates \( k \) based on a new set of \( R_A, R_B \) (a large \( 2r \) bit number chosen at random) and \( F(S) \) (\( F \) is supported by an effective CA based PRSG) for each session. Hence, even if an adversary captures the intermediate communications, i.e., \( E_A(Q_B, h(Q_B), R_B) \) or \( E_B(Q_A, h(Q_A), R_A) \) and manages to extract \( R_A, R_B \) and \( F(S) \) offline, it is no use for the next session. It is extremely difficult for the adversary to generate/produce the correct \( k \) during the current session. Again, based on the previously captured information, it is also not an easy task for the adversary to pretend to be \( A \) or \( B \) in the successive communications, because in each session the legitimate parties use a different set of \( R_A, R_B \) and \( F(S) \) for the generation of \( k \).

### E. Forward Secrecy Attack

In this attack, an adversary attempts to extract the current \( R_A, R_B \) and \( F(S) \) and ultimately the \( S \) based on the previously used or past \( k \). However, extraction of \( R_A, R_B \) and finally \( S \) based on \( k \) is an extremely difficult task due to the use of a large exponent and a secure one-way hash function. Also, \( k \) is generated for each session based on a new set of \( R_A, R_B \) and \( F(S) \). Hence, the proposed CA-KEP provides forward secrecy even if the current session key \( k \) is compromised.

### F. Forgery Attack

In this attack, a masked party attempts to impersonate either \( A \) or \( B \) to deceive the other party. In other words, a malicious party tries to manage \( R_A, R_B \) and \( F(S) \) for successful computation of \( k \). However, as discussed in the preceding subsection under man-in-the-middle attack, the proposed CA-KEP is secure from this attack, since (i) \( R_A, R_B \) and \( Q_A \) or \( Q_B \) are sent in encrypted form using recipients public key, (ii) in each session a different set of \( R_A, R_B \) and \( F(S) \) are used, and (iii) they are too large to mount a dictionary attack in real-time. Hence, CA-KEP is safe from this attack.

### Table 4: Comparing various existing PAKE protocols with CA-KEP

<table>
<thead>
<tr>
<th>Protocol &amp; Year</th>
<th>No. of Steps</th>
<th>N used as</th>
<th>Intermediate DE*</th>
<th>Security Analysis considering</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLE1 [24] &amp; 2000</td>
<td>4</td>
<td>2</td>
<td>Base</td>
<td>( PK_E ) ( SK_A ) ( IL^* ) ( DA^* ) ( FR^* ) ( SC^* ) ( PC^* )</td>
</tr>
<tr>
<td>S-PaKE [14] &amp; 2004</td>
<td>4</td>
<td>2</td>
<td>DH</td>
<td>Yes</td>
</tr>
<tr>
<td>PKE [14] &amp; 2004</td>
<td>4</td>
<td>2</td>
<td>AKE</td>
<td>Yes</td>
</tr>
<tr>
<td>CP-AKE [14] &amp; 2000</td>
<td>4</td>
<td>2</td>
<td>DH</td>
<td>No</td>
</tr>
<tr>
<td>S-PaKE [16] &amp; 2006</td>
<td>4</td>
<td>2</td>
<td>DH</td>
<td>No</td>
</tr>
<tr>
<td>CP-AKE [16] &amp; 2006</td>
<td>4</td>
<td>2</td>
<td>DH</td>
<td>Yes</td>
</tr>
<tr>
<td>CP-AKE [13] &amp; 2009</td>
<td>4</td>
<td>2</td>
<td>DH</td>
<td>Yes</td>
</tr>
<tr>
<td>CP-AKE [13] &amp; 2009</td>
<td>4</td>
<td>2</td>
<td>DH</td>
<td>Yes</td>
</tr>
<tr>
<td>Proposed CA-KEP</td>
<td>4</td>
<td>2</td>
<td>DH</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Note: \( * \left[ PK_E \ (Encrypted \ key \ exchange \ specific \ to \ a \ model), \ DE \ (Data \ Exchange), \ SK_A \ (Symmetric \ key \ encryption), \ IL \ (Public \ key \ encryption), \ IL \ (Information \ Leakage), \ DA \ (Dictionary \ Attack), \ FR \ (Persistent \ Record), \ SC \ (Small \ Sub-group \ Confinement), \ PC \ (Perfect \ Forward \ Secrecy), \ DDH \ (Decisional \ Diffie-Hellman), \ AKE \ (Authenticated \ Key \ Exchange), \ CRS \ (Common \ Reference \ String), \ DDH \ (Decisional \ Diffie-Hellman) \right] \)
G. Discrete Log Computation

In our protocol, \( p \) is a large prime number, and \( R_A \) and \( R_B \) are the two random numbers used to generate \( Q_A \) and \( Q_B \). To get \( Q_A \) and \( Q_B \), an attacker at least needs to guess \( R_A \), \( R_B \) and \( F(S) \), which is extremely difficult. Again, \( R_A \), \( R_B \), \( F(S) \) and also \( k \) are valid for the current session only. In subsequent sessions, we shall have a different set of these secrets. Also, to obtain \( S \) from \( F(S) \) is exceedingly difficult, since \( F \) is composed of two cryptographically secure functions, \( f_1 \) and \( f_2 \). Therefore, the CA-KEP is secure from the discrete log attack.

VII. Applications of CA-KEP

2PAKE protocols are broadly useful in any application where prolonged key storage is risky or impractical, and where the communication channel may be insecure. Some common applications include user-to-user applications, disk-less workstations, bootstrapping new system installation, cellular phones or other key-pad systems, and multi-factor password + key systems. From an economic point of view as well as to address the stolen key problem, such authentication schemes are always preferable than smart cards. Applications of 2PAKE protocols for authentication can be classified into two: user-to-host authentication and user-to-user authentication.

A. User-to-Host Authentication

For systems where only a numeric key-pad is available, e.g., simple cellular telephone authentication, such a CA-based 2-party key exchange protocol is especially convenient, as it is based on a small shared secret. Remote-controlled set top boxes for TV might be another area of application for the proposed scheme. It is ideal in these applications when used with an additional low-cost module for capturing and preprocessing fingerprint images, because it avoids the necessity of long-term storage of persistent keys. Diskless workstation represent another class of devices where it is inconvenient to have locally stored keys. The proposed 2-party protocol can be ideal for establishing an initial connection to a trusted host, and to obtain the user’s safely-stored credentials, such as fingerprint, and other keys.

B. User-to-User Authentication

So far, we have made the case for the usage of CA-KEP in user-to-host authentication. This protocol is equally useful for direct user-to-user authentication also. [12] describes the use of an interactive questionnaire session to authenticate the identity of a user across a network. The main idea behind this authentication is that the users share some common facts (e.g., sharing their fingerprints), which have to be proved to each other, without revealing the facts. Such general authentication paradox can easily be solved by CA-KEP. For example, in a bank application, the banker may want to know that his client really possesses the fingerprint that he is sharing, and at the same time, the client may also like to know that the bank knows his fingerprint, but neither wants to reveal the information directly to the other. In solving such a problem, such a CA based 2-party key exchange protocol can be successfully applied.

VIII. Concluding Remarks

A fast, yet secure CA-based 2-party authenticated key exchange protocol is presented in this paper. Unlike the other traditional and 3-PAKE protocols, the proposed protocol does not depend on a persistent small shared secret (i.e., password). Rather, it exploits the biometrics of the participating users, in order to generate a large password-like piece of information, which is used during the key generation process. We establish the proposed protocol to be secure and effective in terms of all known attacks and competing protocols. The CA-based logic structure makes the protocol attractive for implementation. There is the scope to increase the intruder complexity by introducing a multi-biometric approach while designing \( M \). Also, there is the scope for incorporating programmable structures for dynamic change of rule vectors (i.e., keys) in the CA based modules to increase the cracking complexity.

Acknowledgments

This work is supported by Council of Scientific & Industrial Research (CSIR), Government of India. It is also partially supported by NSF (US) grants CNS-0958576 and CNS-0851783. The authors are thankful to the funding agencies.

References


Author Biographies

**Monowar Hussain Bhuyan** received his AMIETE degree in Computer Science & Engineering from The Institution of Electronics and Telecommunication Engineers (IETE) in 2007 and received his M.Tech. in Information Technology from Department of Computer Science & Engineering, Tezpur University in 2009. Currently, he is pursuing his Ph.D. in Computer Science & Engineering from the same university. He is a life member of IETE, India. His research areas include biometric authentication, data mining, and network security. He has published seven papers in international journals and referred conference proceedings.

**Dhruba Kr Bhattacharyya** received his Ph.D. in Computer Science from Tezpur University in 1999. Currently, he is a Professor in the Computer Science & Engineering Department at Tezpur University. His research areas include Data Mining, Network Security and Bioinformatics. Prof. Bhattacharyya has published more than 140 research papers in leading international journals and conference proceedings. Dr. Bhattacharyya also has written/edited 8 books. He is on the editorial boards of several international journals and also on the programme committees/advisory bodies of several international conferences/workshops.

**Jugal K. Kalita** is a professor of Computer Science at the University of Colorado at Colorado Springs. He received his Ph.D. from the University of Pennsylvania in 1990. His research interests are in natural language processing, machine learning, artificial intelligence and bioinformatics. He has published more than 100 papers in international journals and referred conference proceedings and has written a book. Professor Kalita is a frequent visitor of Tezpur University where he collaborates on research projects with faculty and students.