An Improvement of Wang et al.’s Authentication Scheme Using Smart Cards

Sandeep K. Sood, Anil K. Sarje, Kuldip Singh
Department of Electronics & Computer Engineering
Indian Institute of Technology
Roorkee, India
{ssooddec, sarjefec, ksconfcn}@iitr.ernet.in

Abstract - In 2009, Wang et al. argued that Das et al.’s scheme is vulnerable to stolen smart card attack if an attacker obtains the smart card of legitimate user and chooses any random password. Then the attacker gets through the authentication process to get access of the remote server. Therefore, Wang et al. proposed an improved scheme to preclude the weaknesses of Das et al.’s scheme. However, we found that Wang et al.’s scheme is vulnerable to impersonation attack, stolen smart card attack and offline password guessing attack. This paper improves Wang et al.’s scheme to resolve the aforementioned problems, while keeping the merits of different dynamic identity based authentication schemes.

Keywords- Network security; Cryptography; Password; Authentication protocol; Smart card; Dynamic identity.

I. INTRODUCTION

Password is the most commonly used authentication technique in smart card based authentication protocols. In 1981, Lamport [1] proposed a password based authentication scheme that authenticates remote users over an insecure communication channel. Lamport’s scheme eliminates the problems of password table disclosure and communication eavesdropping. Since then, a number of static identity based remote user authentication schemes have been proposed to improve security, efficiency and cost. The static identity leaks out partial information about the user’s authentication messages to the attacker. On the other hand, the dynamic identity based authentication schemes provide two-factor authentication based on the identity and password and hence more suitable to e-commerce applications. Therefore in 2004, Das et al. [2] proposed a dynamic identity based remote user authentication scheme to authenticate users that preserves the user’s anonymity. Das et al. claimed that their scheme is secure against stolen verifier attack, replay attack, forgery attack, password guessing attack and identity theft.

However, many researchers [3-10] demonstrated vulnerability of Das et al.’s scheme to different attacks. In 2005, Chien and Chen [3] pointed out that Das et al.’s scheme fails to preserve the user anonymity effectively because the authentication messages belonging to the same user can be identified. They proposed an authentication scheme and claimed that the proposed scheme preserves the user’s anonymity more efficiently. Their scheme is highly computation intensive. In 2005, Liao et al. [4] proposed an authentication scheme that enhanced the security of Das et al.’s scheme and achieves mutual authentication. In 2006, Yoon and Yoo [5] demonstrated a reflection attack on Liao et al.’s scheme that breaks the mutual authentication. They proposed an improved dynamic identity based mutual authentication scheme that eliminates the security flaws of Liao et al.’s scheme. In 2006, Liou et al. [6] suggested a new dynamic identity based remote user authentication scheme using smart cards that achieves mutual authentication. They claimed that their scheme preserves the advantages of Das et al.’s scheme. In 2008, Shih [7] demonstrated that Liou et al.’s scheme fails to achieve mutual authentication. In 2009, Wang et al. [8] argued that Das et al.’s scheme is vulnerable to stolen smart card attack if an attacker obtains the smart card of legitimate user and chooses any random password. Then the attacker gets through the authentication process to get access of the remote server [9]. Therefore, Wang et al. proposed an improved scheme to preclude the weaknesses of Das et al.’s scheme.

In this paper, we found that the Wang et al.’s scheme [8] is vulnerable to Ku and Chang’s impersonation attack [10], stolen smart card attack and offline password guessing attack. To remedy these pitfalls, this paper presents an efficient scheme to resolve the aforementioned problems, while keeping the merits of different dynamic identity based authentication schemes.

The rest of this paper is organized as follows. In Section 2, a brief review of Wang et al.’s scheme [8] is given. Section 3 describes vulnerability of Wang et al.’s scheme to different attacks. In Section 4, a new dynamic identity based authentication scheme is proposed. The security analysis of the proposed improved scheme is presented in Section 5. The comparison of the cost and performance of the proposed scheme with the other related schemes is shown in Section 6. Section 7 concludes the paper.

II. REVIEW OF WANG ET AL.’S SCHEME

In this section, we examine the remote user authentication scheme proposed by Wang et al. [8] in 2009. Wang et al.’s scheme consists of four phases.

A. Registration phase

A user $U_i$ has to submit his unique identity ID to the server $S$ for registration over a secure communication channel. The server $S$ computes $N_i = H(P_i) \oplus H(x) \oplus ID_i$, where $x$ is the secret key of remote server $S$ and $P_i$ is the
password of user \( U_i \) chosen by the remote server \( S \). Then the server \( S \) issues the smart card with secret parameters \((H(\cdot), N_i, y)\) and password \( P_i \) to the user \( U_i \) through a secure communication channel, where \( y \) is the remote server’s secret number stored in each registered user’s smart card.

B. Login phase

The user \( U_i \) inserts his smart card into a card reader to login on to the server \( S \) and submits his identity \( ID_i \) and password \( P_i \). The smart card computes \( CID_i = H(P_i) \oplus H(N_i \oplus y \oplus T) \oplus ID_i \), where \( T \) is the current date and time of the input device and sends the login request message \((ID_i, CID_i, N_i, T)\) to the service provider server \( S \).

C. Verification phase

The service provider server \( S \) checks the validity of timestamp \( T \) by checking \((T' - T) < \delta T\), where \( T' \) denotes the server’s current timestamp and \( \delta T \) is expected time interval for a transmission delay. Afterwards, the server \( S \) computes \( H(P_i') = CID_i \oplus H(N_i \oplus y \oplus T) \oplus ID_i \). \( ID_i = H(P_i) \oplus \delta T \). The server \( S \) compares this computed value of \( ID_i \) with the received value of \( ID_i \). If they are not equal, the server \( S \) rejects the login request and terminates this session. Otherwise, the server \( S \) computes \( A_i = H(H(P_i') \oplus y \oplus T') \) and sends the message \((A_i, T')\) back to the smart card of user \( U_i \). On receiving the message \((A_i, T')\), smart card checks the validity of timestamp \( T' \) by checking \((T' - T') < \delta T\), where \( T' \) denotes the client’s smart card current timestamp. Then the client’s smart card computes \( A_i' = H(H(P_i') \oplus y \oplus T') \) and compares it with the received value of \( A_i \). This equivalency authenticates the legality of the service provider server \( S \) and the login request is accepted else the connection is interrupted.

D. Password change phase

The client \( C \) can change his password without the server’s help. A user \( U_i \) inserts his smart card into a card reader and submits his password \( P_i \) corresponding to his smart card and request to change his password with a new password \( P_i^{new} \). The smart card computes \( N_i^{new} = N_i \oplus H(P_i) \oplus H(P_i^{new}) \) and updates the values of \( N_i \) stored in its memory with \( N_i^{new} \) and the password gets changed.

III. WEAKNESSES OF WANG ET AL.’S SCHEME

Wang et al. [8] claimed that their protocol can resist various known attacks. Unfortunately, this protocol is found to be flawed for impersonation attack, stolen smart card attack and offline password guessing attack.

A. Impersonation attack

Ku and Chang [10] demonstrated impersonation attack on Das et al.’s scheme [2]. This attack is also applicable on Wang et al.’s [8] scheme. An attacker can perform impersonation attack as follows.

1. An attacker intercepts a login request message \((ID_i, CID_i, N_i, T)\) of the user \( U_i \) from the public communication channel.

2. Now the attacker gets current time stamp \( T' \) and computes \( \delta T = T \oplus T', N_i = N_i \oplus \delta T \) and \( CID_i = CID_i \oplus \delta T \).

3. Then an attacker frames the message \((ID_i^*, CID_i^*, N_i^*, T')\) and sends this login request message to the server \( S \).

4. The server \( S \) checks the validity of the timestamp \( T' \) by checking \((T' - T') < \delta T\), where \( T' \) denotes the server’s current timestamp. Then the server \( S \) computes:

\[
H(P_i^*) = CID_i^* \oplus H(N_i^* \oplus y \oplus T') \oplus ID_i^*
\]

\[
= CID_i \oplus \delta T \oplus H(N_i \oplus \delta T \oplus y \oplus T \oplus \delta T) \oplus ID_i^*
\]

\[
= CID_i \oplus \delta T \oplus H(N_i \oplus y \oplus T) \oplus ID_i^*
\]

\[
= H(P_i) \oplus \delta T
\]

\[
ID_i^* = N_i^* \oplus H(x) \oplus H(P_i^*)
\]

\[
= N_i \oplus \delta T \oplus H(x) \oplus H(P_i^*) \oplus \delta T
\]

\[
= N_i \oplus H(x) \oplus H(P_i^*)
\]

\[
= ID_i^*
\]

The server \( S \) compares this computed value of \( ID_i^* \) with the received value of \( ID_i^* \). On successful verification, the server \( S \) accepts the forged login authentication request. Therefore, the attacker can impersonate as legitimate user \( U_i \).

B. Stolen smart card attack

An attacker can extract the stored values through some technique like by monitoring their power consumption and reverse engineering techniques as pointed out by Kocher et al. [11] and Messerges et al. [12]. Therefore, a malicious privileged user \( U_i \) having his own smart card can gather information \( N_i = H(P_i) \oplus H(x) \oplus ID_i \) from his own smart card or from communication channel during the login request message \((ID_i, CID_i, N_i, T)\) to the service provider server \( S \). He can find out the value of \( H(x) \) as \( H(x) = N_i \oplus H(P_i) \oplus ID_i \) because the malicious user knows his own identity \( ID_i \) and password \( P_i \) corresponding to his smart card. In case another user \( U_i \)’s smart card is stolen by this malicious user, he can extract the information \( N_k = H(P_k) \oplus H(x) \oplus ID_k \) from the memory of smart card.

1. Now the malicious user \( U_i \) chooses any arbitrary combination of identity \( ID_k \) and password \( P_k \) so that \( N_k = H(P_k) \oplus H(x) \oplus ID_k \) is equal to extracted value of \( N_k \).

2. Then the malicious user \( U_i \) inserts the stolen smart card into a card reader to login on to the server \( S \) and submits fabricated identity \( ID_k \) and password \( P_k \). The smart card computes \( CID_k = H(P_k) \oplus H(N_k \oplus y \oplus T) \oplus ID_k \), where \( T \) is current date and time of the input device and sends the login request message \((ID_k, CID_k, N_k, T)\) to the service provider server \( S \).

3. The service provider server \( S \) checks the validity of timestamp \( T \) by checking \((T' - T') < \delta T\), where \( T' \) denotes the server’s current timestamp and \( \delta T \) is expected time interval for a transmission delay. Afterwards, the server \( S \) computes:

\[
H(P_k^*) = CID_k \oplus H(N_k \oplus y \oplus T) \oplus ID_k^*
\]

\[
= H(P_k^*)
\]

\[
ID_k^* = N_k \oplus H(x) \oplus H(P_k^*)
\]

\[
= N_k \oplus H(x) \oplus H(P_k^*) \quad \text{because } N_k^* = N_k
\]
Then server \( S \) compares the value of \( ID_0^* \) with the received value of \( ID_0^* \). This equivalency authenticates the malicious user \( U_i \) to the service provider server \( S \).

4. Then the server \( S \) computes \( A_X = H (H (P_i^* \oplus y \oplus T)) \) and sends the message (\( A_X, T \)) back to the smart card of user \( U_i \). On receiving the message (\( A_X, T \)), smart card checks the validity of timestamp \( T \) by checking \((T^* - T) <= \delta T\), where \( T^* \) denotes the client’s smart card current timestamp. Then the client’s smart card computes \( A_X^* = H (H (P_i^* \oplus y \oplus T')) \) and compares it with the received value of \( A_X \). This equivalency authenticates the legality of the service provider server \( S \) and the login request is accepted.

C. Offline password guessing attack

The malicious privileged user \( U_i \) having his own smart card can gather information \( N_i = H (P_i) \oplus H (x) \oplus ID_i \) from his smart card or from own login request message (\( ID_i^*, CID_i, N_i, T \)) to the service provider server \( S \). He can find out the value of \( H (x) \) as \( H (x) = N_i \oplus H (P_i) \oplus ID_i \), because the malicious user \( U_i \) knows his own identity ID and password \( P_i \) corresponding to his smart card.

1. Now this malicious privileged user \( U_i \) intercepts the login request message (\( ID_i^*, CID_i, N_i, T \)) of the user \( U_i \) from the public communication channel.

2. This malicious user \( U_i \) can extract the password verifier information of the user \( U_i \) as \( H (P_i) = N_i \oplus H (x) \oplus ID_i \) because the malicious user \( U_i \) knows the values of \( H (x) \), \( ID_i \) and \( N_i \).

3. Then malicious user \( U_i \) can launch offline dictionary attack on \( H (P_i) \) to know the password \( P_i \) corresponding to the smart card of user \( U_i \).

4. In case user \( U_i \)'s smart card is stolen by this malicious user, he can masquerade as a legitimate user to the service provider server \( S \) because the malicious user possesses the smart card, knows the identity ID and password \( P_i \) corresponding to the user \( U_i \).

IV. PROPOSED PROTOCOL

In this section, we describe a new remote user authentication scheme which resolves the above security flaws of Wang et al.’s [8] scheme. Fig. 1 shows the entire protocol structure of the new authentication scheme.

A. Registration phase

A user \( U_i \) has to submit his identity \( ID_i \) and password \( P_i \) to the server \( S \) via a secure communication channel to register itself to the server \( S \).

Step 1: \( U_i \to S: ID_i, P_i \)

The server \( S \) chooses random value \( y_i \) and computes the security parameters \( N_i = H (ID_i) \oplus P_i \oplus H (x), A_i = H (ID_i) \oplus P_i \oplus H (y_i), B_i = y_i \oplus ID_i \oplus P_i \), and \( D_i = H (ID_i) \oplus y_i \). The server \( S \) stores the value of \( y_i \) corresponding to each user in such a way that the value of \( D_i \) must be unique for each user. The server \( S \) stores \( y_i \oplus x \) and \( ID_i \oplus H (x) \) corresponding to \( D_i \) in its database. Then the server \( S \) issues the smart card containing security parameters \((N_i, A_i, B_i, H (x))\) to the user \( U_i \) through a secure communication channel.

Step 2: \( S \to U_i: \) Smart card

B. Login phase

A user \( U_i \) inserts his smart card into a card reader to login on to the server \( S \) and submits his identity \( ID_i^* \) and password \( P_i^* \). The smart card computes \( y_i^* = B_i \oplus ID_i^* \oplus P_i^* \). Then the server \( S \) computes \( A_i^* = H (ID_i^* \oplus P_i^* \oplus H (y_i^*)) \) and compares it with the stored value of \( A_i \) in its memory to verify the legality of the user.

Step 1: Smart card checks \( A_i^* = A_i \)

After verification, the smart card computes \( H (x) = N_i \oplus H (ID_i \oplus P_i \oplus H (x) \oplus y_i \oplus y_i \oplus T) \) and \( M_i = H (ID_i \oplus H (x) \oplus y_i \oplus T), \) where \( T \) is current date and time of the input device. Then the smart card sends login request message (\( CID_i, M_i, T \)) to the service provider server \( S \).

Step 2: Smart card \( \to S: CID_i, M_i, T \)

C. Verification and session key agreement phase

After receiving the login request message from the user \( U_i \), the service provider server \( S \) checks the validity of timestamp \( T \) by checking \((T^* - T) <= \delta T\), where \( T^* \) is current date and time of the server \( S \) and \( \delta T \) is expected time interval for a transmission delay. The server \( S \) computes \( D_i^* = CID_i \oplus H (H (x) \oplus T) \) and finds \( D_i \) corresponding to \( D_i^* \) in its database and then extracts \( y_i \oplus x \) and \( ID_i \oplus H (x) \) corresponding to \( D_i^* \) from its database. Now the server \( S \) computes \( y_i^* \oplus x \) and \( ID_i \oplus H (x) \) because the server \( S \) knows the value of \( x \). Then the server \( S \) computes \( M_i^* = H (ID_i \oplus H (x) \oplus y_i \oplus T) \) and compares \( M_i^* \) with the received value of \( M_i \).

Step 1: Server \( S \) checks \( M_i^* = M_i \)

This equivalency authenticates the legality of the user \( U_i \) and the login request is accepted. Finally, the user \( U_i \) and the server \( S \) agree on the common session key as \( H (H (x) \oplus ID_i \oplus T \oplus y_i) \). Afterwards, all the subsequent messages between the user \( U_i \) and the server \( S \) are XOR-ED with the session key. Therefore, either the user \( U_i \) or the server \( S \) can retrieve the messages between themselves because both of them know the common session key.

D. Password change phase

The client \( C \) can change his password without the help of the server \( S \). The user \( U_i \) inserts his smart card into a card reader and enters his identity \( ID_i^\text{new} \) and password \( P_i^\text{new} \) corresponding to his smart card. The smart card computes \( y_i = B_i \oplus ID_i^\text{new} \oplus P_i^\text{new}, A_i^\text{new} = H (ID_i^\text{new} \oplus P_i^\text{new} \oplus H (y_i^\text{new})) \) and compares it with the stored value of \( A_i \) in its memory to verify the legality of the user. Once the legality of card holder is verified then the user \( U_i \) can instruct the smart card to change his password. Afterwards, the smart card asks the card holder to resubmit a new password \( P_i^\text{new} \) and then smart card computes \( N_i^\text{new} = N_i \oplus H (ID_i \oplus P_i \oplus H (ID_i \oplus P_i^\text{new})), A_i^\text{new} = H (ID_i \oplus P_i^\text{new} \oplus H (y_i^\text{new}), B_i^\text{new} = y_i \oplus ID_i \oplus P_i^\text{new} \). Thereafter, smart card updates the values of \( N_i, A_i \), and \( P_i \).
and $B_i$ stored in its memory with $N_i^{\text{new}}$, $A_i^{\text{new}}$ and $B_i^{\text{new}}$. The identity and the password of the user is verified before the password update procedure, while the password change phase of Wang et al.'s [8] scheme is insecure. If an attacker manages to obtain the smart card of user $U_i$ for a very short time, he can change the password of user $U_i$ without knowing the correct password.

V. SECURITY ANALYSIS

Smart card is a memory card that uses an embedded micro-processor from smart card reader machine to perform required operations specified in the protocol. Kocher et al. [11] and Messerges et al. [12] pointed out that all existing smart cards cannot prevent the information stored in them from being extracted such by monitoring their power consumption. Some other reverse engineering techniques are also available for extracting information from smart cards. That means once a smart card is stolen by an attacker, he can extract the information stored in it. A good password authentication scheme should provide protection from different feasible attacks.

1. Impersonation attack: In this type of attack, the attacker impersonates as a legitimate client and forges the authentication messages using the information obtained from the authentication scheme. The attacker can attempt to modify a login request message $(C_{iD}, M_i, T)$ into $(C_{iD}^*, M_i^*, T)$, where $T'$ is the attacker's current date and time, so as to succeed in the authentication phase. However, such a modification will fail in Step 1 of the verification and session key agreement phase because an attacker has no way of obtaining the values of $ID_i$, $y_i$ and $H(x)$ to compute the valid parameters $C_{iD}^*$ and $M_i^*$. Therefore, the proposed protocol is secure against impersonation attack.

2. Stolen smart card attack: In case a user's smart card is stolen by the attacker, he can extract the information stored in its memory. The attacker extracts $N_i = H(ID_i | P_i) \oplus H(x)$, $A_i = H(ID_i | P_i) \oplus P_i \oplus H(y_i)$ and $B_i = y_i \oplus ID_i \oplus P_i$ from the memory of smart card. Even after gathering this information, the attacker has to guess $ID_i$ and $P_i$ correctly at the same time. It is not possible to guess out two parameters correctly at the same time in real polynomial time. Therefore, the proposed protocol is secure against stolen smart card attack.

3. Offline dictionary attack: In offline dictionary attack, the attacker can record messages and attempt to guess the user's identity $ID_i$, password $P_i$ and other secret parameters from recorded messages. The attacker first tries to obtain the card verification information $C_{iD} = H(ID_i | y_i) \oplus H(H(x) | T)$, $M_i = H(ID_i | H(x) | y_i) \oplus T$, and then try to guess the values of $ID_i$, $y_i$ and $H(x)$ by offline guessing. Even after gathering this information, the attacker has to guess all three parameters $ID_i$, $y_i$ and $H(x)$ correctly at the same time. It is not possible to guess all three parameters correctly at the same time. In another option, the attacker requires valid smart card of the legitimate user and then has to guess the identity $ID_i$ and password $P_i$ correctly at the same time. It is not possible to guess two parameters correctly at the same time. Therefore, the proposed protocol is secure against offline dictionary attack.

4. Malicious user attack: A malicious privileged user having his own smart card can gather information like $N_i = H(ID_i | P_i) \oplus H(x)$, $A_i = H(ID_i | P_i) \oplus P_i \oplus H(y_i)$ and $B_i = y_i \oplus ID_i \oplus P_i$ from the memory of smart card. This malicious user can not generate smart card specific values of $C_{iD} = H(ID_i | y_i) \oplus H(H(x) | T)$ and $M_i = H(ID_i | H(x) | y_i) \oplus T$ to masquerade as other legitimate user $U_k$ to the service provider server $S$ because the values of $C_{iD}$ and $M_i$ is
TABLE I. EFFICIENCY COMPARISON AMONG SMART CARD BASED SCHEMES

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>384 bits</td>
<td>256 bits</td>
<td>384 bits</td>
<td>384 bits</td>
<td>256 bits</td>
<td>384 bits</td>
</tr>
<tr>
<td>E2</td>
<td>3 * 128 bits</td>
<td>6 * 128 bits</td>
<td>5 * 128 bits</td>
<td>6 * 128 bits</td>
<td>4 * 128 bits</td>
<td>4 * 128 bits</td>
</tr>
<tr>
<td>E3</td>
<td>4 T_H</td>
<td>2 T_H</td>
<td>3 T_H</td>
<td>3 T_H</td>
<td>2 T_H</td>
<td>2 T_H</td>
</tr>
<tr>
<td>E4</td>
<td>6 T_H</td>
<td>3 T_H</td>
<td>4 T_H</td>
<td>6 T_H</td>
<td>5 T_H</td>
<td>2 T_E + 2 T_S</td>
</tr>
<tr>
<td>E5</td>
<td>4 T_H</td>
<td>3 T_H</td>
<td>5 T_H</td>
<td>4 T_H</td>
<td>2 T_H + 2 T_E + 2 T_S</td>
<td>3 T_H</td>
</tr>
</tbody>
</table>

VII. CONCLUSION

In this paper, we presented a cryptanalysis of Wang et al.’s scheme and showed that their scheme is vulnerable to impersonation attack, stolen smart card attack and offline password guessing attack. An improvement to Wang et al’s scheme is proposed that inherits the merits of different dynamic identity based authentication schemes and resists different attacks. The security of proposed scheme depends upon the one-way hash function and security analysis proved that the improved scheme is more secure and practical.

REFERENCES


smart card specific and depends upon the values of ID_k, y_k and H (x). Although malicious user can extract H (x) from his own smart card but he does not have any method to calculate the values of ID_k and y_k. Therefore, the proposed protocol is secure against malicious user attack.

5. Server spoofing attack: In server spoofing attack, the attacker can manipulate the sensitive data of legitimate users via setting up fake servers. In the proposed protocol, malicious server can not compute the session key H (H (x) | ID_i | T | y_i) because the malicious server does not know the value of H (x), ID_i and y_i. Therefore, the proposed protocol is secure against server spoofing attack.

VI. COST AND PERFORMANCE ANALYSIS

An efficient authentication scheme must take communication and computation cost into consideration during user’s authentication. The performance comparison of the proposed scheme with the relevant smart card based authentication schemes is summarized in Table 1. Assume that the identity ID_i, password P_i, x, y_i and timestamp values are all 128-bit long. Moreover, we assume that the output of secure one-way hash function is 128-bit. Let T_H, T_E and T_S denote the time complexity for hash function, exponential operation and symmetric key encryption respectively. Typically, time complexity associated with these operations can be roughly expressed as T_E, T_H >> T_S, T_H >> T_E. In the proposed scheme, the parameters stored in the smart card are N_c, A_c, B_c and the memory needed (E1) in the smart card is 384 (= 3*128) bits. The communication cost of authentication (E2) includes the capacity of transmitting message involved in the authentication scheme. The capacity of transmitting message (E1) is 384 (= 3*128) bits. The computation cost of registration (E3) is the total time of all operations executed in the registration phase. The computation cost of registration is 4T_H. The computation cost of the user and the service provider server is the time spent by the user and the service provider server during the process of authentication. Therefore, the computation cost of the user (E4) is 6T_H and that of the service provider server (E5) is 4T_H. The proposed scheme requires nearly the same computation as other related schemes [2][4][5][6][8] and requires very less computation as compared to Chien and Chen scheme [3] but it is highly secure as compared to the related schemes.