DNA based Password Authentication Scheme using Hyperelliptic Curve Cryptography for Smart Card

Vijayakumar P¹, Vijayalakshmi V² and Zayaraz G²
¹ Research Scholar, Dept., of ECE, Pondicherry Engineering College, Pondicherry, India
² Vijayrgcet@gmail.com
³ Assistant Professor, Dept., of ECE, Pondicherry Engineering College, Pondicherry, India
³ Associate Professor, Dept., of ECE, Pondicherry Engineering College, Pondicherry, India
Email: {¹vijaykumar@pec.edu, ²vijayrgcet@gmail.com, ³zayaraz@pec.edu}

Abstract— Smart Card technology is the emerging technology which is developing among common masses in our culture and widely used in the sectors of banking and industries. Many research works are undergoing in this area to provide highly confidential data transmission. Existing Scheme provides a security against offline attack for the lost Smart Card using Elliptic Curve Cryptography (ECC) but it requires more communication and computation overhead with higher key length. To overcome this limitation, DNA based Password authentication using Hyper Elliptic Curve Cryptography (HECC) scheme is proposed. It provides more security than existing system which allows server and smartcard to exchange the generated password and verify each other. This system exploits the advantages of Hyperelliptic Curve Cryptography (HECC) technique which is having lesser key size, less communication and computation overhead for Password generation and signature verification process.

Index Terms— Smartcard, Hyper Elliptic Curve Cryptography, Deoxyribo Nucleic Acid, Communication complexity, Computational complexity, Password generation, Authentication, Elliptic Curve Cryptography.

I. INTRODUCTION

Smart cards are widely used in many business systems which provide portable benefits and secure data storage, and it also incorporated in many computing systems. Smart cards are provided with memory card which can enable to store and secure the information using available cryptographic algorithm. Deoxyribo Nucleic Acid (DNA) based computing technology combined with cryptographic algorithm will provide secure authentication for smartcard. DNA is a long linear polymer found in the core part of a cell. DNA is made up of several nucleotides in the form of double helix and it is linked with the transmission of genetic information. DNA based computing technique mainly focuses on storage capacity and its unique property. Hyperelliptic Curve Cryptosystem (HECC) is used in many power constrained devices which offer equal security as any other public key cryptosystem with much smaller key length. This cryptographic system allows highly efficient computation of the underlying field arithmetic. Hyperelliptic Curve Cryptosystem is very much popular among other cryptosystems such as Elliptic Curve Cryptosystem (ECC), Rivest Shamir Adleman (RSA), Digital Encryption Standard (DES), etc., due to its shorter key length [1-5]. This section gives the introduction about smart card, DNA and Hyperelliptic Curve Cryptography. Section II describes related works. Section III deals with different phases for providing robust password authentication scheme.

DOI: 02.ITC.2014.5.581
© Association of Computer Electronics and Electrical Engineers, 2014
using ECC. Section IV briefly explains mathematical background of Hyperelliptic Curve Cryptography and also explains DNA based password authentication scheme using HECC. Section V deals with results and discussion in this context. Finally, it concludes the paper.

II. RELATED WORKS

Xing Wang applied DNA computing theories for cryptography security which transmits the message securely and effectively. This paper shows how cryptography works with DNA computing technique. Here, most famous symmetric key RSA algorithm is used to encrypt the message and decrypt the message to provide greater level of security with 1024 bit key size. Major drawback of this algorithm is increased key size tends to increased computational complexity [6]. Reference [7] propose a novel DNA based Elliptic Curve Cryptography algorithm which provide the same level of security as [6] with lesser key size by employ the advantage of ECC. As a result, it gives lesser communication and computation overhead. Guozhen Xiao [8] pointed out the biological background of DNA cryptography and principle of DNA computing. This paper compares the status, security and application fields of DNA cryptography with traditional cryptography and quantum cryptography. Guangzhao Cui [9] provides information about background principle of DNA computing, challenges behind DNA computing based cryptology, DNA encryption techniques and application of DNA Computing security field. It gives brief introduction about DNA steganography and DNA authentication. Wen-Bing Horng [10] offers a secure and efficient user authentication scheme for smart card which will improve the level of security of the Peyravian-Zunic scheme. He also reveals the weakness of Kwon et al.'s protocol concerning off-line password forgery attack and guessing attack. Xiaoyi-Ying [11] proposed a novel key authentication scheme which combines the fuzzy extractor concept with Smart Card. This scheme can avoid guessing attack, parallel attack and masquerade attack. Seoul [12] uses symmetric key cryptosystem with modular exponentiation to make an efficient authentication scheme for non-tamper resistant smart card. He showed that Song's scheme is weak to the offline password guessing attack and the insider attack. Nenghai Yu [13] proposed a secure scheme which is very efficient, both in term of computational complexity and storage capacity. This scheme is very suitable for providing remote authentication in distributed application and also it was developed against password guessing attack, masquerade attack and replay attack. Roy prevents the clogging attack by implementing two party identity-based authenticated key agreement protocols [14].

III. EXISTING PASSWORD AUTHENTICATION SCHEME

Existing password authentication scheme for smart card consists of five phases: Parameter generation phase, registration phase, pre-computation phase, login phase and password changing phase. In parameter generation phase, server generates a large prime number, two field elements. With the help of these number fields, it will generate a point from order n. Server also selects private key and public key for exchanging a data between the users and distributes the generated parameter. In registration phase, user uses the Smart card to send identification information to server for authentication purpose. In log-in phase, session key is generated, verified and exchanged between user and Smart Card. In password changing phase, user can able to change the password frequently with the help of session key which is produced in log-in phase [15].

IV. PROPOSED DNA BASED PASSWORD AUTHENTICATION SCHEME USING HYPERELLIPTIC CURVE
CRYPTOGRAPHY

Existing scheme needs to enter the password directly which will create insecure environment between smart card and server. It exploits the advantages of Elliptic Curve Cryptography for key exchange, encryption and decryption process. It require 160 bit key length to provide greater level of security, tends to high communication and computational overhead. In order to avoid the above limits, DNA based password authentication scheme using Hyperelliptic Curve Cryptography is proposed. This will generate the password based on DNA molecule and will authenticate both smart card and server. In order to provide higher level of security, password generation phase and authentication phase is added in this proposed scheme. The proposed scheme employs the advantages of Hyperelliptic Curve Cryptography to provide same level of security with 80-bit key size, and less communication and computational overhead. In 1988, Neal Koblitz
proposed an expansion of Elliptic Curve cryptosystem known as Hyperelliptic Curve Cryptosystem. Hyperelliptic Curve Cryptosystem was widely used fast public key cryptosystem in many power constrained devices with high efficiency and security. HECC was very much famous because of its high efficiency, shorter key length, easily implemented for software and hardware applications, less communication and computational overhead, less consuming power, less processing time. The security of HECC is based on the Hyperelliptic Curve Discrete Logarithmic Problem (HECDLP), (i.e) \( k \in \mathbb{Z}_p \), the computation of \( K = k \times P \) where \( K \) is the private key and \( P \) is the public key of the user. So, the security of HECC lies on the discrete logarithm problem in the Jacobian of the curve [16-20]. These security features and characterize of HECC allows to use in less memory and less power smart card device. Hyperelliptic points are generated from the curve \( C \) which form a Jacobian group and divisor. These two key elements are useful for cryptographic scheme which is transformed from Hyperelliptic Curve. Proposed scheme enhances the server performance when Smart card content is disclosed. Proposed scheme consists of six phases: Parameter generation phase, Password generation phase, Registration phase, Pre-Computation phase, Authentication phase and Password changing phase.

A. Parameter Generation Phase

The related parameters are generated using Hyperelliptic Curve \( C \) for encryption and decryption process. From the curve \( C \), private and public keys are generated using contor algorithm. The process involved to generate both keys is shown as follows:

Input : Public parameters are Hyperelliptic curve \( C \), prime \( p \) and divisor \( D \).

Output  : Public key \( Ps \) and Private key \( Xs \).

Process :

Step 1: Server choose a Hyperelliptic curve Equation of genus \( g \) (\( g > 2 \)) over \( F(q) \) as shown in (1)

\[ y^2 + h(x) \cdot y = f(x) \quad (1) \]

where,

- \( h(x) \) is a polynomial of degree, \( g \).
- \( f(x) \) is monic polynomial of degree \( 2g+1 \), which satisfies the equation ,

\[ h(x)' \cdot y = f(x) \quad (2) \]

\[ 2y + h(x) = 0 \quad (3) \]

Step 2 : From the points of Hyperelliptic curve \( C \), server generates a set of elements of Jacobian over \( J(F_q) \)

\[ D = \sum m_i \cdot P_i \quad (4) \]

where

- \( m_i \geq 0 \)
- \( D \) - Reduced divisor
- \( P_i \) - Finite points

Step 3: The server generates a point \( G \) from order \( n \), satisfies \( n \times G = 0 \).

Step 4: The server picks a random number \( Xs \) to be the private key and computes the public key \( Ps = Xs \times G \).

Step 5: The server issues the parameter \( (Ps, G, D, C, n) \).

B. Password Generation Phase

Each user receives the parameter before joining into the network, which is provided by server. Instead of giving password directly, password is mapped with DNA molecule along with the number to provide greater level of security which is not known to the eavesdropper who always tries to retrieve the password. Password is generated by combining DNA molecules such as Adenine (A), Thymine (T), Guanine (G) and Cytosine(C) as shown in Table II.

Step 1: Server can map the password message with DNA nucleotide using the Table I.

Step 2: Convert the DNA nucleotide into number using the Table II.

_Example_

| Password | : Hyper |
| DNA standard | : ATG AAA ACA TTT ACT |
| Password | : 104030 101010 102010 404040 102040 |

519
TABLE I. CONVERSION OF PLAIN TEXT TO DNA MOLECULE

<table>
<thead>
<tr>
<th></th>
<th>DNA Molecule</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>CCA</td>
</tr>
<tr>
<td>B</td>
<td>GTT</td>
</tr>
<tr>
<td>C</td>
<td>TTG</td>
</tr>
<tr>
<td>D</td>
<td>GGT</td>
</tr>
<tr>
<td>E</td>
<td>TTT</td>
</tr>
<tr>
<td>F</td>
<td>TCG</td>
</tr>
<tr>
<td>G</td>
<td>CGC</td>
</tr>
<tr>
<td>H</td>
<td>ATG</td>
</tr>
<tr>
<td>I</td>
<td>AGT</td>
</tr>
<tr>
<td>J</td>
<td>CGA</td>
</tr>
<tr>
<td>K</td>
<td>GAA</td>
</tr>
<tr>
<td>L</td>
<td>CGT</td>
</tr>
<tr>
<td>M</td>
<td>CCT</td>
</tr>
<tr>
<td>N</td>
<td>TCT</td>
</tr>
<tr>
<td>O</td>
<td>CGG</td>
</tr>
<tr>
<td>P</td>
<td>ACA</td>
</tr>
<tr>
<td>Q</td>
<td>CAA</td>
</tr>
<tr>
<td>R</td>
<td>ACT</td>
</tr>
<tr>
<td>S</td>
<td>GCA</td>
</tr>
<tr>
<td>T</td>
<td>CTT</td>
</tr>
</tbody>
</table>

TABLE II. CONVERSION OF NUCLEOTIDE TO NUMBER

<table>
<thead>
<tr>
<th>DNA Molecule</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-10</td>
</tr>
<tr>
<td>C</td>
<td>-20</td>
</tr>
<tr>
<td>G</td>
<td>-30</td>
</tr>
<tr>
<td>T</td>
<td>-40</td>
</tr>
</tbody>
</table>

**C. Registration Phase**

The user can use the smart card to send identification information for the server to authenticate as shown in Figure 1.

Step 1: If the smart card wants to register at the server with its own identity (IDi) and password (PWi), the user has to compute the password as shown in the above example and send it with username to the server over a secure channel. The smart card chooses a random number RN1, Identity IDi, and calculates U1 value using the generated password F(PWi) as shown in (5).

\[ U_1 = h(F(PW_i \oplus RN_1)^{-1}) \]  

Then, smart cards send \{IDi, h(F(PW) \oplus RN1, U1)\} to the server.

\[ \text{Smart Card} \rightarrow \text{Server} : \{ID_i, h(F(PW) \oplus RN_1, U_1)\} \]  

Step 2: The server generates a random number S1 as a secret key and chooses another random number RN2 and calculates U2 value using U1 and RN2. The server also provides the expiry date (EDi) and time stamp (Tsi) for each user to check its validity and time period respectively as shown in (8).

\[ U_2 = U_1 \cdot RN_2 \]  

\[ Y_i = h(ED_i, T_s) \]  

\[ Q_i = E_{s1}(h(F(PW) \| RN_1) \| U_2) \| h(ID_i) \| CI_i) \| h(F(PW) \| RN_1 \| h(ED_i, T_s)) \]  

\[ V_i = h(ID_i, S_1, CI_i) \]  

Smart card memory consists of the following parameters:

\[ e = r \cdot G \]  

\[ c = r \cdot P_s = r \cdot x \cdot G \]  

Then, the server issues a certificate to user i that contains the parameters (IDi, CIi, Qi, Vi, Yi).

Step 3: The user receives these information (IDi, CIi, Qi, Vi, Yi) and stores into the smart card.

**D. Pre-Computation phase**

The smart card chooses a random number r and calculates e=(r*G) and c=(r*P)=r*x*G. Then, (e,c) is stored in card memory for use in the authentication phase.

**E. Authentication Phase**

If user i log-in to the server by using his own smart card content and respective password as shown in Figure 2.

Step 1: The smart card calculates Ev(e) and send Ev(e) and Q, to the server e=(r x G).

Step 2: The server uses secret key S1 to decrypt Q, = (U2 \| IDi \| CIi \| h(F(PW)) \| RN1 \| h(EDi, Tsi)) and calculates,

\[ U_1 = U_2 \cdot RN_2 \]  

520
\[ Y = h((E_D, T_s)) \]  \hspace{1cm} (14) \\
\[ V = h(I_D, Y, U_1, C_{I_i}). \]  \hspace{1cm} (15)

**F. Mutual Authentication**

The server will verify the parameter comparing with the calculated values

- Is \( C_{I_i} \) is stored in the registration table
- Is \( I_D \) in the registration
- Is Date is expired
- Is Time Stamp is equal

\[ c = e \cdot x = r \cdot x \cdot G \]  \hspace{1cm} (16) \\
\[ M_s = h(c \| R_s \| V_i \| Y_i) \]  \hspace{1cm} (17)

Server sends \( (c, M_s) \) to the smartcard

**G. Password-Changing Phase**

User ‘i’ wants to change password. Change the message using session key, can encrypt smart card. Session key is produced in authentication phase. Smart card selects a random number \( R_{N_1} \) and another new password \( F(PW_i) \) and sends, \( E_s(k, h(F(PW_i) \| R_{N_1})) \) to the server. Server receives the messages. It recalculates \( Q_i^* \), \( Q_i^* = E_s(h(F(PW_i) \| R_{N_1})) \| I_D \| C_I \| h(I_D \| C_I \| h(F(PW_i) \| R_{N_1}))) \). Sends \( E_s(Q_i^*) \) to smart card. Smart card will decrypt \( Q_i^* \) using session key and store in its memory.

**V. SIMULATION RESULTS AND DISCUSSIONS**

The simulation parameters are processing time and key size: Processing time is the total time taken to finish the task (phase) by the personal computer. Key size is the size of the key used for encrypts and decrypt the message using HECC. MATLAB software is used to implement the proposed scheme which consists of parameter generation phase, password generating phase, registration phase, pre-computation phase, authentication phase, and password-changing phase. Simulation results show the variation of processing time of each and every phase with respect to the key size. From this result, it is inferred that the number of bits involved providing authentication using HECC is less than ECC. Finally, performances of both existing and proposed scheme are compared for each and every phase with respect to key size as show in table.3.
Parameter generation phase took more processing time for generating parameters using existing scheme (ECC) than proposed scheme (HECC). From the Figure.3, it is inferred that for key size 35 bits, ECC takes the processing time of 389 milliseconds whereas HECC takes only 289 milliseconds. Figure.4 shows that the total time taken to map DNA molecule with password, and mapping of Nucleotide to number is 270 ms (52 bit key size). ECC takes the processing time of 320 milliseconds for the same key size. During registration process, user receives parameter from the trusted server, and stores it in smart card memory which takes processing time of 479ms for ECC, and 343ms for HECC (52 bit key size) as shown in Figure.5. Pre-Computation phase generates two parameter e and c which are used for mutual authentication phase take 250ms (ECC) and 198ms (HECC) for the same key size as shown in Figure.6. Figure.7 shows that total processing time taken to authenticate both server and smart card with the help of generated parameter for 52 bit key size. For the key size 52 bits, ECC takes the processing time of 512 milliseconds, whereas HECC takes only 353 milliseconds. User wants to change the password in password changing phase takes only 275ms for HECC and 510ms for ECC (35 bit key size) as shown in Figure.8.
VI. CONCLUSION

Existing authenticated key agreement scheme fails to save the password from the eavesdroppers whereas DNA based password authentication scheme avoids password hacking by mapping original message with DNA molecule along with number to improve the level of security. Proposed scheme replaces ECC by extended cryptosystem for encryption and decryption of message which consumes less power and less processing time suitable for power constrained device. Addition of mutual authentication phase enables to check validity and identity of both user and server which avoid denial of service, non-repudiation, data integrity and forgery. Password changing phase avoid phishing and hacking of password.
REFERENCES


AUTHORS

P. Vijayakumar is currently working as Assistant Professor (Sr.) in School Electronic Science Engineering at VIT university Chennai campus, India and pursuing Ph.D in Pondicherry University. He completed his B.Tech in Rajiv Gandhi College of Engineering and Technology and M.Tech in Pondicherry Engineering College which is affiliated to Pondicherry University. He has 7 years of teaching experience. To his credit, he has published more than 15 research papers relating to Cryptography and Network Security in several National / International Journals and Conferences. He can be reached by email at vijayrgcet@gmail.com
Dr. V. Vijayalakshmi is currently working as Assistant Professor in Electronics & Communication Engineering Department at Pondicherry Engineering College, Puducherry, India. She completed her B.Tech, M.Tech and PhD in Pondicherry Engineering College which is affiliated to Pondicherry University. She has 20 years of teaching experience. To her credit, she has published more than 25 research papers relating to Network Security and software Engineering in several National / International Journals and Conferences. She can be reached by email at vijizai@pec.edu

Dr. G. Zayaraz is currently working as Professor in Computer Science & Engineering Department at Pondicherry Engineering College, Puducherry, India. He received his Bachelor’s, Master’s and Doctorate degree in Computer Science & Engineering from Pondicherry University. He has published more than twenty five research papers in reputed International Journals and Conferences. His areas of specialization include Software Architecture and Information Security. He is a reviewer for several reputed International Journals and Conferences and Life Member of CSE, and ISTE. He can be reached by email at gzayaraz@pec.edu