Eland: An Efficient Lightweight Anonymous Authentication Protocol Applied to Digital Rights Management System

Qing Fan1,2, Jianhua Chen1*, Yihong Wen3, Min Luo4

1 School of Mathematics and Statistics, Wuhan University, China
2 Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, China
3 The 54th Research Institute of China Electronics Technology Group Corporation, China
4 School of Cyber Science and Engineering, Wuhan University, China
qingfan_Cassie@whu.edu.cn, chenjh_ecc@163.com, yihwen@139.com, mluo@whu.edn.cn

Abstract

A digital rights management (DRM) system provides the function of packaging raw content into an appropriate distribution form, protection for content temper-proof transmission and circumvention of unauthorized use. The digital rights contents in DRM system improve people’s spiritual quality and create huge market value. However, in the sharing of digital contents, malicious adversaries could bring security and privacy issues even cause infringement of copyright. To solve these problems, valid user identity authentication and secret key distribution to legal users is an efficient method. Nevertheless, the existing authentication schemes for DRM system have problems of security weakness or efficiency to be enhanced. Thereby, we cryptanalyze these protocols’ vulnerabilities and propose a new lightweight anonymous authentication protocol called Eland that can be efficiently applied to DRM system. We compare our protocol with six DRM-related authentication protocols from the aspects of security properties, computation expense and communication cost. Comparisons result indicate that our protocol achieves a well trade-off between security and performance. Specifically, our protocol reduces 37.5% communication cost and satisfies more security requirements than Yu et al.’s scheme.

Keywords: Digital rights management (DRM), Authentication, Key distribution, Anonymity

1 Introduction

According to explanation from World Intellectual Property Organization (WIPO), the digital rights management (DRM) refers to identification and description of property, rights involved in creation and enforcement of usage restrictions [1]. In a narrow sense, it includes techniques, tools and measures of digital rights protection during the digital content usage. Digital rights contents generally contain rights-related digital consumption products such as e-books, digital music, video, games and so on. The massive quality digital content has met the people’s growing spiritual and cultural needs [2]. Value of DRM system is increasingly prominent and the DRM industry becomes one of the most promising formats. Mordor Intelligence report on DRM market indicates that the DRM market was valued at USD 3215.4 million in 2020, expected to exceed USD 6000 million by 2026 and increase at a compound annual growth rate (CAGR) of 12.18% over the forecast 2021-2026 [3]. In view of DRM operations, the digital rights content is encrypted using a symmetric key and stored in a content server. Meanwhile, the symmetric key and the rights object (i.e. state of permitted ways for digital content consumption) are packaged into a license held by the license server [4]. Generating process of ciphertexts and license is depicted in Figure 1. The user sends a license request information to the license server for developing the digital content’s value. Not all the request could be accepted since the user may be illegal or malicious. Moreover, adversaries may also trace the user’s behavior to steal user privacy. Furthermore, piracy statistics for 2021 shows that digital video piracy is costing US content and distribution sectors between USD 29.2 and USD 71.0 billion each year [5]. Therefore, security and privacy protection of content delivery is particularly significant.

Authentication, as a prominent cryptographic technique, is undoubtedly an efficient way to authenticate the user’s legality. There exist some authenticated protocols applying in digital rights management system [6-9], some of which has improved previous schemes for better security or efficiency. However, these schemes whether the new proposed or enhanced schemes still have some vulnerabilities. For example, Rana and Mishra’s [6] and Rana and Mishra’s [7] are not equipped with three-factor security. Lee et al.’s [8] is resistless to denial of service attack (DoS) and does not provide secure authentication and three-factor security. The scheme [9] does not hold mobile device theft attack resistance, three-factor security and secure authentication. Yu et al.’s [10] further remedy scheme [9]’s limitation and proposed a lightweight three-factor authentication protocol.

Although the above schemes attempt to provide security and privacy protection in DRM system application, they cannot satisfy as many security properties as possible to the best of our knowledge. Moreover, the limited computing power and memory capacity of DRM mobile device [11] asks higher demand for protocol’s computation efficiency and storage expense. Consequently, we present an efficient anonymous authentication protocol applied to DRM system which only uses hash function and XOR operations to realize lightweight.

*Corresponding Author: Jianhua Chen; E-mail: chenjh_ecc@163.com
DOI: 10.53106/160792642022032302007
Our Contributions.

- We propose a new authentication protocol named Eland, which only employs hash function and XOR operations to achieve lightweight anonymous authentication. We also apply it to DRM system and meet the system’s security requirements.
- We compare the proposed protocol with six recent DRM-related authentication protocols. The security and performance comparison results show that our protocol realizes a better trade-off between efficiency and security. Thereby, our protocol is more efficient for digital contents protection.
- We prove our protocol achieves secure mutual authentication under the ROR model, that is we give theory proof of impersonation attack resistance and key distribution security.

Organization. Section 2 introduces DRM system model, fuzzy extractor, threat model and security requirements of DRM authentication protocol. Section 3 reviews Yu et al.’s protocol and analyzes weakness of this protocol. To realize a well trade-off between security and efficiency, we propose a new authentication and key distribution protocol in section 4. Its formal security proof and informal security analysis are provided in section 5. Section 6 compares the performance of existing six authentication schemes. We introduce related works in section 7. Finally, we conclude this paper in section 8.

2 Preliminaries

In this section, we introduce the digital rights management (DRM) system model including compositions and workflow, the biometrics information extraction tool i.e. fuzzy extractor, threat model of authentication protocol as well as security requirements of DRM system. Finally, we give notations used in this paper.

2.1 DRM System Model
verification and the corresponding key \( K_{PC} \) is finally distributed to the user. In the end, the user decodes the ciphertexts from content server and gets the intact digital content.

- **Mobile device.** The mobile device is held by the user which stores some authentication-related information such as auxiliary biometrics string, the hiding password information, authorized secret value from license server and so on. Information stored in mobile device will be used for user identity verification.

In this paper, we mainly consider the critical authentication and key distribution process between the user and license server.

### 2.2 Fuzzy Extractor

In this paper, we use the biometric tool fuzzy extractor to extract a certain secret value \( \sigma \) from biometric data \( B\text{IO} \), where \( \sigma \) can be seen as a biometric key for user authentication and could be obtained with input an error-tolerant biometrics and an auxiliary string. Fuzzy extractor consists of \( \text{Gen} \) algorithm and \( \text{Rep} \) algorithm. \( \text{Gen} \) algorithm extracts two strings \( \sigma, \theta \) and \( \text{Rep} \) algorithm recovers \( \sigma \) using an error-tolerant biometrics \( B\text{IO}' \) and \( \theta \). Details of fuzzy extractor are described as follows.

- **Gen** is a probabilistic algorithm and takes biometrics \( B\text{IO} \) as input. It is responsible for extracting a uniformly random string \( \sigma \) and an auxiliary string \( \theta \). This algorithm can be written as \( (\sigma, \theta) \leftarrow \text{Gen}(B\text{IO}) \).

- **Rep** is a deterministic algorithm to reproduce the string \( \sigma \) with inputting biometrics \( B\text{IO}' \) and auxiliary string \( \theta \), where the distance between \( B\text{IO}' \) and \( B\text{IO} \) must keep within an allowable range. This algorithm can be written as \( \sigma \leftarrow \text{Rep}(B\text{IO}', \theta) \).

### 2.3 Threat Model

Abdalla et al. [14] proposed the Real-Or-Random (ROR) model for two-party authenticated key exchange protocols. It demonstrates that ROR model provides more stronger security than BPR model and applied in many authenticated schemes [15-18]. According to [14, 17], the adversary could execute Send query, Execute query, Capture query, Reveal query and many-time Test query.

- **Hash
dot(\):** This query holds a hash list \( L_h \) with the form of \((x, h(x))\). When \( \mathcal{A} \) executes this query with message \( x \), the oracle first checks whether \( x \) in \( L_h \) and returns the corresponding \( h(x) \) if it is. Otherwise, the oracle computes the hash value \( h(x) \) with \( x \) and adds \( h(x) \) into \( L_h \) and returns \( h(x) \) to \( \mathcal{A} \).

- **Execute\((U_i, S_j, \text{sid})\):** This query models passive attacks where the adversary \( \mathcal{A} \) could eavesdrop on all messages transmitted between \( U \) and \( S \) as protocol \( \Pi \) with session identifier \( \text{sid} \). On receiving this query, oracle \( \Pi_{\text{sid}}^{U, S} \) and \( \Pi_{\text{sid}}^{S, U} \), respectively simulating \( U \) and \( S \), will launch protocol \( \Pi \). Then the messages exchanged between them will be recorded and transmitted to \( \mathcal{A} \).

- **SendServer\((U_i, S_j, \text{sid}, M')\):** This query models active attacks against a server where \( \mathcal{A} \) sends messages \( M' \) to server oracle \( \Pi_{\text{sid}}^{S, U} \). Then the oracle will calculate a response honestly as \( \Pi \) and sends the response to \( \mathcal{A} \).

- **SendUser\((S_j, U_i, \text{sid}, M')\):** This query models active attacks against a user where \( \mathcal{A} \) sends messages \( M' \) to user oracle \( \Pi_{\text{sid}}^{U, S} \). If \( M' = \lambda \), the oracle will launch a new session and transmit authenticated request messages to \( \mathcal{A} \). Otherwise, the oracle will execute \( \Pi \) to generate corresponding messages and send them to \( \mathcal{A} \).

- **Capture\((U_i)\):** This query models user’s mobile device theft attack in which all the values stored in the user mobile device will be captured by \( \mathcal{A} \) through executing this query.

- **Reveal\((\Pi, U_i, S_j, \text{sid})\):** This query models key exposure attack between user \( U_i \) and license server \( S_j \). This query could only be executed when mutual authentication is realized. On receiving this query, the oracle will return KEY to \( \mathcal{A} \).

- **Test\((U_i, S_j, \text{sid})\):** This query can be executed by \( \mathcal{A} \) after secure authentication and the KEY has been shared with \( U_i \). Then the oracle \( \Pi_{\text{sid}}^{S, U} \) tosses a coin \( b \). The oracle returns either a random one if \( b = 0 \) or the real session key if \( b = 1 \). The adversary could ask as many Test queries as he wants.

In the end of above game, \( \mathcal{A} \) outputs a guess bit \( b \). Assuming the probability of \( \mathcal{A} \) success is \( \Pr[\text{Succ}] \), we define advantage of \( \mathcal{A} \) breaking the protocol \( \Pi \) is \( \text{Adv}_{\Pi, \mathcal{A}}^M = \Pr[\text{Succ}] - 1 \).

**Definition 1 (Secure KEY distribution):** If there is no probabilistic polynomial time (PPT) adversary could correctly guesses \( b \) in the above game with non-negligible advantage \( \text{Adv}_{\Pi, \mathcal{A}}^M \), we say the proposed protocol is MA-secure.

**Definition 2 (Hash function):** A hash function \( h: \{0,1\}^* \rightarrow \{0,1\}^n \) is a deterministic function that takes a variable length bit string as input and outputs a fixed length string of \( n \) bits. Collision-resistance and one-wayness of hash function are described as follows.

- One-wayness. Given a hash value \( h(x) \) , it is computable for PPT adversary to find the \( x \).

- Collision-resistance. Let \( \text{Adv}_{\Pi, \mathcal{A}}^{\text{HASH}} \) denote the advantage of PPT \( \mathcal{A} \) finding a hash collision, i.e. \( \text{Adv}_{\Pi, \mathcal{A}}^{\text{HASH}} = \Pr[(x_1, x_2): x_1 \neq x_2, h(x_1) = h(x_2)] \), where \( x_1, x_2 \) are randomly selected with arbitrary length. Then, \( \text{Adv}_{\Pi, \mathcal{A}}^{\text{HASH}} \) is negligible.

### 2.4 Security Requirements

According to [8-10, 19], an authentication protocol for DRM system has the following security requirements.

- **Impersonation attack resilience:** In the DRM system, it requires that no illegal user could impersonate the legitimate client to obtain digital rights content, which is an essential demand for digital rights content protection. Thus, the authenticated and key distribution should hold impersonation attack resilience.

- **Mobile device theft attack resilience:** Each user of DRM system has a mobile device to store secret information and execute necessary operations. Even if the private data in the mobile device is theft, the adversary still cannot acquire other information such as password and secret string from biometrics to obtain digital rights content access eventually.
• **Offline password guessing attack resilience:** This attack is that the adversary or malicious server attempts to guess the user password in the offline manner. Password exposure may result in other private information leakage even exposing digital rights content. Thereby, offline password guessing attack resilience is important for authentication protocol of DRM system.

• **Replay attack resilience:** This attack is that the adversary could take previous authentication messages in the public channel for current protocol execution, which cannot be found by the server. Replay attack may bring resource waste of license server and should be resisted in the authenticated and key distribution for DRM.

• **Stolen verifier attack resilience:** This attack could happen in the protocol registration phase. If the server holds a verifier table, the adversary may acquire user authenticated information by intruding into the server database. An efficient authentication scheme needs to keep stolen verifier attack resilience.

• **Privileged insider attack resilience:** This attack mainly takes place in the user registration phase. Privileged insider in the server side may acquire more information to break the protocol. Privilege insider attack resilience requires this breaking will not happen.

• **Server spoofing attack resilience:** Server spoofing attack means the adversary impersonates server to spoof users. This attack could bring perplex to the user’s regular work. Thereby, server spoofing attack resilience is a necessary security requirement for DRM system.

• **Denial of service attack resilience:** This attack is that an adversary tries to make service resource unavailable to intended users by disrupting services of a client connected to the Internet. It is usually achieved by flooding the targeted server with overloaded authentication requests to prevent some or all legitimate requests from being fulfilled. To guarantee regular digital rights content access, DRM system should have denial of service attack resilience.

• **Three-factor security:** In the biometrics-based authenticated scheme, the smart device, biometrics and password consist of three factors to guarantee the scheme’s security. Three-factor security requires that any two of factors are exposed cannot lead to the other one factor’s leakage. For example, the adversary could not guess the user’s password even if theft of smart device and biometrics.

• **Secure mutual authentication:** In the DRM system, we assume that the license server is honest and trustworthy. Therefore, authentication to the user who tries to obtain KEY of digital rights content is important. Secure authentication in this paper demands any possible attacks from the user side should be prevented.

• **Anonymity:** In the digital rights content access process, the user always wants to hide his true identity and the requested digital rights content in case the adversary maliciously track the user’s behavior. For protecting the user’s identity and behavior privacy, anonymity of authentication protocol needs to be held.

• **Untraceability:** To hide the digital rights access footprint, the user may expect each authentication and digital rights content access is independent, that is the adversary cannot find relationship between any two authentication phases. Untraceability could protect the user’s privacy from statistical analysis and is another security requirement for DRM system.

Notations used in this paper is explained in Table 1.

<table>
<thead>
<tr>
<th>Table 1. Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notations</strong></td>
</tr>
<tr>
<td>$U_i$</td>
</tr>
<tr>
<td>$LS_j$</td>
</tr>
<tr>
<td>$ID_i$</td>
</tr>
<tr>
<td>$ID_{DC}$</td>
</tr>
<tr>
<td>$PW_i$</td>
</tr>
<tr>
<td>$BIO_i$</td>
</tr>
<tr>
<td>$r_1$</td>
</tr>
<tr>
<td>$r_2$</td>
</tr>
<tr>
<td>$X_{LS}$</td>
</tr>
<tr>
<td>$KEY_{DC}$</td>
</tr>
<tr>
<td>$h(\cdot)$</td>
</tr>
<tr>
<td>$\oplus$</td>
</tr>
<tr>
<td>$|$</td>
</tr>
</tbody>
</table>

# 3 Review of Yu et al.’s Protocol

In this section, Yu et al.’s three-factor authentication protocol will be reviewed briefly. Their protocol consists of user registration, login and authentication, and password change. Each phase of Yu et al.’s protocol is presented in the following subsection.

## 3.1 Yu et al.’s Protocol

• **Registration phase:** A mobile user $U_i$ registers its identity with license server $LS_j$ as the following steps.

  - $U_i$ chooses $ID_i$, $PW_i$ and biometrics $BIO_i$. Then $U_i$ calculates $Gen(BIO_i) = (\sigma_i, \theta_i), RPW_i = h(PW_i||\sigma_i)$ and transmits $(ID_i, RPW_i)$ to $LS_j$ through a secure channel.

  - On receiving the request information from $U_i$, $LS_j$ calculates $X_i = h(ID_i||X_{LS_j}), d_i = X_i \oplus h(ID_i|||RPW_i)$ and $f_i = h(RPW_i||X_i)$. Then $LS_j$ stores $ID_i, X_i$ in a database and transmits $(d_i, f_i)$ to $U_i$ via a secure channel.

  - Having received messages from $LS_j$, $U_i$ stores $(d_i, f_i)$ in a device.

• **Authentication and key distribution phase:** The goal of this phase is to realize mutual authentication between $U_i$ and $LS_j$, and distribute decryption key $KEY_{DC}$ to a legitimate user, which can be seen in Figure 3.

  - $U_i$ inputs $ID_i, PW_i, BIO_i$, computes $\sigma_i = Rep(BIO_i, \theta_i), RPW_i = h(PW_i||\sigma_i), X_i = d_i \oplus h(ID_i||RPW_i)$ and verifies whether $h(RPW_i||X_i)$ is equal to $f_i$ in the device. Then $U_i$ randomly selects a nonce $r_3$ and calculates $M_1 = X_i \oplus r_3, M_2 = ID_i \oplus r_3, M_3 = ID_{DC} \oplus r_3, M_{DS} = h(ID_i||ID_{DC}||X_i||r_3)$. Finally, $U_i$ transmits $(M_1, M_2, M_3, M_{DS})$ to the $LS_j$.

  - On receiving $(M_1, M_2, M_3, M_{DS})$, $LS_j$ calculates $r_3 = M_1 \oplus X_i, ID_3 = M_2 \oplus r_3, ID_{DC} = M_3 \oplus r_3$ and verifies $M_{DS} = h(ID_i||ID_{DC}||X_i||r_3)$. Then $LS_j$ retrieves secret key $KEY_{DC}$ according to $ID_{DC}$ and calculates $M_4 = r_2 \oplus X_i, M_5 = KEY_{DC} \oplus X_i, M_{SU} =$
\[ h(ID_i||X_i||KEY_{DC}||r_2) \]. Eventually, \( LS_j \) transmits \( \{M_4,M_5,M_{SU}\} \) to \( U_i \).

- On receiving \( \{M_4,M_5,M_{SU}\} \), \( U_i \) recovers \( r_2 = M_4 \oplus X_i \). \( KEY_{DC} = M_5 \oplus X_i \) and verifies \( h(ID_i||X_i||KEY_{DC}||r_2) = M_{SU} \). If the equation holds, \( U_i \) stores the secret key \( KEY_{DC} \) and has the access right to the digital content.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.png}
\caption{Authentication and key distribution phase of Yu et al.’s protocol}
\end{figure}

- **Private information change phase:** This phase is performed offline. A legitimate user \( U_i \) could change password and imprinted biometrics as follows in any cases.

  - \( U_i \) takes \( ID_i \) and the present \( B1O_i, PW_i \) as inputs and calculates \( Gen(B1O_i) = (\sigma_i,\theta_i) \), \( RPW_i = h(PW_i||\sigma_i) \). Then \( \{ID_i,RPW_i\} \) are sent to the device.

  - The device calculates \( X_i = d_i \oplus h(ID_i||RPW_i) \), checks \( f_i = h(RPW_i||X_i) \) and aborts if the check fails.

- On receiving the permitting message, \( U_i \) chooses a new password \( PW_{i}^{\text{new}} \) and a new biological information \( B1O_{i}^{\text{new}} \) to compute \( Gen(B1O_{i}^{\text{new}}) = (\sigma_{i}^{\text{new}},\theta_{i}^{\text{new}}) \), \( RPW_{i}^{\text{new}} = h(\text{PW}_{i}^{\text{new}}||\sigma_{i}^{\text{new}}) \). Then the \( RPW_{i}^{\text{new}} \) is sent to the mobile device.

- Eventually, the mobile device computes \( d_{i}^{\text{new}} = X_i \oplus h(ID_i||RPW_{i}^{\text{new}}) \), \( f_{i}^{\text{new}} = h(RPW_{i}^{\text{new}}||X_i) \) and replaces \( \{d_i,f_i\} \) with \( \{d_{i}^{\text{new}},f_{i}^{\text{new}}\} \).

### 3.2 Security Analysis of Yu et al.’s Protocol

In this section, we analyze that a malicious attacker \( A \) could impersonate a legitimate user \( U_i \) by computing the login request message of \( U_i \). \( A \) could intercept the valid messages \( \{M_1,M_2,M_3,M_{US}\} \) from the public channel. \( A \) could guess the correct random nonce \( r_1 \) by verifying the equation
\[
h(M_1 \oplus r_1||M_2 \oplus r_1||M_1 \oplus r_1||r_2) = M_{US} .
\]
Having gotten \( r_1 \), \( A \) can get \( X_i = M_1 \oplus r_1, ID_i = M_2 \oplus r_1, ID_{DC} = M_3 \oplus r_3 \). Then \( A \) could achieve impersonation attack even break properties of secure mutual authentication and user anonymity.

- **Impersonation attack:** \( A \) could impersonate \( U_i \) and obtains secret key \( KEY_{DC} \) as follows.

### 4 The Proposed Protocol

In this section, a more secure lightweight authentication protocol for the DRM system is proposed to resist the impersonation attack. The proposed protocol comprises three phases: registration phase, login and authentication phase, and password change phase.

#### 4.1 Registration Phase

Each user holds a device to execute necessary computation and store secret information. Through the device, a user could
complete the authentication protocol for DRM system. A new user \( U_i \) who tries to access to digital content will first register the identity \( ID_i \) through the license server \( LS_j \). Registration phase of the proposed protocol is shown in Figure 4. Detailed descriptions are as follows.

- \( U_i \) inputs the identity \( ID_i \), password \( PW_i \) and imprints the biometrics \( BIO_i \) into the device. The device could obtain \( \sigma_i \) and \( \theta_i \) through \( Gen(BIO_i) = (\sigma_i, \theta_i) \). Then the device selects a random nonce \( r_i \) and computes \( d_i = h(PW_i[|\sigma_i|]|ID_i), WPWR_i = d_i \oplus r_i \). Finally, \( U_i \) transmits \( \{ID_i, r_i, d_i\} \) to \( LS_j \) through a secure channel.
- On receiving \( \{ID_i, r_i, d_i\} \), \( LS_j \) calculates \( X_i = h(ID_i||X_i||LS_i||r_i) \) and sends \( f_i \) to \( U_i \) through a secure channel.
- Eventually, \( U_i \) stores \( \{\theta_i, WPWR_i, f_i\} \) in the device.

### 4.2 Authentication and Key Distribution Phase

Having received a confirmation message of \( ID_{DC} \) from \( LS_j \) over a secure channel, the user \( U_i \) will transmit a login request to the corresponding license server \( LS_j \). Then \( LS_j \) authenticates \( U_i \) and sends the secret key \( KEY_{DC} \) to \( U_i \) under the premise of \( U_i \) legitimate. As Figure 5, the detailed login and authentication steps are described in the following.

- \( U_i \) inputs the identity \( ID_i \), password \( PW_i^r \) and biometrics \( BIO_i^r \) into the device. Then the device uses \( BIO_i^r \) and the stored \( \theta_i \) to compute \( \sigma_i^r = Rep(BIO_i^r, \theta_i) \) and calculates \( d_i^r = h(PW_i^r[|\sigma_i^r|]|ID_i) \). Then \( U_i \) chooses a random \( r_i^r \) and computes \( M_1 = (d_i^r || r_i^r) \oplus h(ID_{DC}), M_2 = X_i \oplus (ID_{DC}||r_i^r), M_3 = (ID_{DC}||r_i^r) \oplus h(X_i), M_{US} = h(ID_{DC}||M_1||M_2||M_3) \), where \( T_1 \) is the current timestamp. Eventually, \( U_i \) sends \( \{T_1, M_1, M_2, M_3, M_{US}\} \) to \( LS_j \).

- Upon receiving \( \{T_1, M_1, M_2, M_3, M_{US}\} \) at the time \( T_2 \), \( LS_j \) first checks whether \( |T_2 - T_1| \leq \Delta \) and aborts if the check fails. Otherwise, \( LS_j \) computes \( (d_i||r_i) = M_1 \oplus h(ID_{DC}), X_i = h(d_i||X_i||r_i) \) and gets \( (ID||r_i) = M_2 \oplus X_i, (ID||ID_{DC}) = M_3 \oplus h(X_i||r_i)||T_1 \). Then \( LS_j \) verifies whether the equation \( M_{US} = h(ID_{DC}||ID_{DC}||r_i||X_i||r_i||T_1) \) holds. If aborts if the equation does not hold. Otherwise, the secret key \( KEY_{DC} \) to the digital content is retrieved. Finally, \( LS_j \) chooses a random nonce \( r_2 \) to compute \( M_4 = X_i \oplus (ID||r_2) \oplus KEY_{DC} \), \( M_{SU} = h(r_2||ID_{DC}||X_i||ID_{DC}||KEY_{DC}) \) and transmits \( \{M_4, M_{SU}\} \) to \( U_i \).
- On receiving \( \{M_4, M_{SU}\} \), \( U_i \) calculates \( r_2 = M_{SU} \oplus X_i \) and \( KEY_{DC} = M_4 \oplus h(X_i||r_2) \). Then he checks the equation \( M_{SU} = h(r_2||ID_{DC}||X_i||ID_{DC}||KEY_{DC}) \). If the equation holds, \( U_i \) obtains the correct secret key \( KEY_{DC} \) and has access authority to the digital content.

![Figure 4. Registration phase](image_url)

![Figure 5. Authentication and key distribution phase](image_url)
4.3 Private Information Change Phase

In this phase, we suppose that $U_i$ requests to change the secret password $PW_i$ and biological information $BIO_i$. This phase is executed by the user $U_i$ and the mobile device with the aid of license server $LS_j$. The detailed steps are described in the following, which is depicted in Figure 6.

- $U_i$ inputs identity ID$_i$, the present password $PW_i$, biometrics $BIO_i$ and the new password $PW_i^{\text{new}}$, biometrics $BIO_i^{\text{new}}$ into the device. Then the device computes $\sigma_i^* = \text{Rep}(BIO_i, \theta_i)$, $d_i^* = h(PW_i | \sigma_i^* | ID_i)$, $r_i^* = RPW_i \oplus d_i^*$ and $X_i^* = h(d_i^* | r_i^*) \oplus f_i$. Then it selects a new random $s_i$ and computes $Gen(BIO_i^{\text{new}}) = (\sigma_i^{\text{new}}, \theta_i^{\text{new}})$, $d_i^{\text{new}} = h(PW_i^{\text{new}} | \sigma_i^{\text{new}} | ID_i)$. $RPW_i^{\text{new}} = d_i^{\text{new}} \oplus s_i$. Finally, the device transmits $(ID_i, r_i^*, d_i^*, X_i^*, s_i, d_i^{\text{new}})$ to the LS$_j$ via a secure channel.

- On receiving messages from $U_i$, LS$_j$ computes $X_i = h(ID_i | d_i^* | X_i^*)$ and verifies the equation $X_i = X_i^*$. If it does not hold, LS$_j$ refuse the $U_i$’s request. Otherwise, it computes $X_i^{\text{new}} = h(ID_i | d_i^{\text{new}} | X_i^*)$ and $f_i^{\text{new}} = h(d_i^{\text{new}} | s_i) \oplus X_i^{\text{new}}$. Eventually, LS$_j$ sends $f_i^{\text{new}}$ to $U_i$. Upon receiving $f_i^{\text{new}}$, the mobile device stores $(\theta_i^{\text{new}}, RPW_i^{\text{new}}, f_i^{\text{new}})$ securely.

5 Security Analysis

In this section, we firstly give the proof of Eland’s $MA$-security under the security model in section 2.3. Then we explain that Eland protocol satisfies the security requirements mentioned in section 2.4.

5.1 Formal Security Proof

Theorem 1. Assume that the PPT adversary $\mathcal{A}$ running in time $t$ under the ROR model and $PD$ is a proper password dictionary, $\rho$ is the bit length of the biometrics key $\sigma_i$, the advantage of $\mathcal{A}$ in breaking the MA-secure of protocol $\Pi$ is estimated as $Adv_{\mathcal{A}}^{\text{MA}} \leq q_{Send}^2 + q_{Send} \cdot 2^{|PD|}$, where $q_{Send}$ is the number of Hash queries, Send queries, the size of $PD$ and the range of $h(\cdot)$ respectively.

Proof: We follow the similar proof as . In this proof, four games denoted by $G_j (j = 0, 1, 2, 3)$ are considered where the success probability is $Pr[\text{Succ}_j]$ for guessing the right bit $b$. Descriptions of these games are given as follows.

$G_0$: $G_0$ denote a real attack executed by $\mathcal{A}$ against the proposed protocol $\Pi$ under ROR model, where $\mathcal{A}$ tries to guess the correct bit $b$ to break MA-secure. Suppose that $\mathcal{A}$ succeeding in this game is $Pr[\text{Succ}_0]$. According to definition of advantage, there exists the following equation:

$$Adv_{\mathcal{A}}^{\text{MA}} = |2Pr[\text{Succ}_0] - 1|$$

$G_1$: This game simulates an eavesdropping attack executed by eavesdropper, who can intercept $MSG_1 = \{T_1, M_1, M_2, M_3, M_{US}\}$ and $MSG_2 = \{M_4, M_5, M_{SU}\}$ in the public channel. In other words, $\mathcal{A}$ in ROR model could make $\text{Execute}$ query. At the end of the game, $\mathcal{A}$ executes $\text{Test}$ query which outputs the actual $KEY$ or a random string according to bit $b$. Note that $KEY$ is computed as $KEY_{\mathcal{A}} = M_5 \oplus h(X_i | r_i^* | r_j^*)$, where $X_i = h(ID_i | PW_i | \sigma_i | [5LS] | r_i^*)$. To compute $KEY_{\mathcal{A}}$, $\mathcal{A}$ must have $M_5$ and $h(X_i | r_i^* | r_j^*)$, which further needs to know the master $x_{LS}$. password $PW_i$ and biometrics key $\sigma_i$. $\mathcal{A}$ also requires $ID_i, r_i^*, r_j^*$ which are secret to $\mathcal{A}$. Therefore, probability of $\mathcal{A}$ succeeding in the game $G_1$ is not increased by eavesdropping attack, that is $Pr[\text{Succ}_1] = Pr[\text{Succ}_0]$.

$G_2$: Through adding $\text{Send}$ query and $\text{Hash}(\cdot)$ query to $G_1$, it is transformed into $G_2$ which simulates the active attack. In this game, $\mathcal{A}$ tries to deceive an honest participant to accept modified messages. Specifically, $\mathcal{A}$ is allowed to execute several $\text{Hash}(\cdot)$ queries to check whether there exists hash collisions. Moreover, $\mathcal{A}$ may perform two kinds of $\text{Send}$ queries i.e. $\text{SendServer}(U_i, S_i, sid, M)$ query to server oracle and $\text{SendUser}(S_j, U_i, sid, M)$ query to user oracle. Any changes on $\{T_1, M_1, M_2, M_3, M_{US}\}$ and $\{M_4, M_5, M_{SU}\}$ during authentication and key distribution phase contain random numbers, timestamp and user’s identity. As a consequence, there does not exist hash collision when $\mathcal{A}$ queries $\text{Send oracle}$. According to birthday paradox, we have

$$|Pr[\text{Succ}_1] - Pr[\text{Succ}_2]| \leq q_{Send}^2 / (2^{|PD|})$$

$G_3$: Through adding $\text{Capture}(U_i)$ query to $G_2$, it is transformed into $G_3$ which simulates the mobile device theft attack. In this game, $\mathcal{A}$ may perform dictionary attack through using the information stored in user device to guess the low-entropy password. In addition, $\mathcal{A}$ may attempt to get biometrics key $\sigma_i$ according to the stored information in device. We use fuzzy extractor and biometrics $BIO$ to uniformly extract random $\rho$-bit biometrics key $\sigma_i \in \{0, 1\}^{\rho}$. Then the probability of $\mathcal{A}$ in guessing the correct $\sigma_i$ is approximately $\frac{1}{2^\rho}$. Moreover, the system permits the limited number of wrong password entries. Thereby, we have

$$|Pr[\text{Succ}_2] - Pr[\text{Succ}_3]| \leq q_{Send} / (2^\rho \cdot |PD|)$$
According to equation (2-4), we have
\[ |Pr[\text{Succ}_1] - Pr[\text{Succ}_2]| \leq q_{\text{send}}/(2|H(\text{HASH})|) + q_{\text{send}}/(2^p + |PD|) \]
Combining with equation (1) and \( Pr[\text{Succ}_2] = \frac{1}{2} \), we have
\[ Adv_{P,A}^{\text{MA}} \leq \frac{q^2}{2|H(\text{HASH})|} + \frac{q_{\text{send}}}{2^p + |PD|} \tag{5} \]

5.2 Informal Security Analysis

Next, we show that the Eland protocol is equipped with the system’s security requirements as section 2.4.

- **Impersonation attack resilience**: The adversary \( A \) attempting to impersonate a user \( U_i \) must produce valid authentication messages \( \{M_1, M_2, M_3, M_{US}\} \) where natural secret values are \( \{PW_i, \sigma_i, X_i, r_i, ID_L, ID_{DC}\} \). \( A \) could not recover any one of these secret value neither by establishing equation utilizing \( \{M_1, M_2, M_3, M_{US}\} \) nor through breaking security of hash function \( h(\cdot) \). Thus, \( A \) cannot forge valid authenticated messages and successfully deceive license server. Thereby, the proposed Eland protocol resists impersonation attack.

- **Mobile device theft attack resilience**: The user stores \( \{\theta_i, RPWi, f_i\} \) in the mobile device where \( \theta_i \) is an auxiliary string, \( RPWi \) is \( d_i \oplus r_i \), \( r_i \) is a random number, \( f_i \) is \( h(d_i||r_i)||X_i \). Even if \( A \) acquires \( \{\theta_i, RPWi, f_i\} \), he still cannot get \( ID_L, PW_i, \sigma_i \) due to secrecy of master key \( X_{LS} \). Therefore, the proposed protocol holds mobile device theft attack resilience.

- **Replay attack resilience**: Suppose that \( A \) intercepts messages \( \{T_1, M_1, M_2, M_3, M_{US}\} \) and replaces \( T_1 \) with the new timestamp \( T' \). Then he attempts to impersonate \( U_i \) by replaying \( \{T'_1, M_1, M_2, M_3, M_{US}\} \) to \( LS_i \). \( LS_i \) could find the attack by checking \( M_{US} = h(ID_{DC}||d_i||X_{LS}||PW_i||\sigma_i|||ID_L) \). Even if \( A \) acquires \( PW_i \) and biometrics \( \sigma_i \), he still cannot recover information stored in mobile device due to unknown \( r_i \) and \( X_{LS} \). Therefore, the proposed protocol withstands replay attack.

- **Denial of service attack resilience**: According to the denial of service, \( A \) generally produces superfluous authenticated messages to \( LS_i \) which results in overloaded computation. \( LS_i \) first verifies whether the timestamp is new enough which is the first step to hinder needless computation. Then he verifies \( M_{US} \) to avoid more useless calculated load. These two steps shows the proposed protocol does not have denial of service.

- **Stolen verifier attack resilience**: As shown in Figure 4, \( LS_j \) does not keep a verification table for the following authentication. Therefore, \( A \) cannot break our protocol through stealing verifier.

- **Privileged insider attack resilience**: In the registration phase, the user transmits \( ID_{dc}, r_1 \) and \( d_1 = h(PW_i||\sigma_i||ID_L) \) to the server. Then the privileged insider cannot request \( PW_i \) from \( d_i \) because he does not know the biometric information \( \sigma_i \). Therefore, the proposed protocol could resist privileged insider attack.

- **Server spoofing attack resilience**: To impersonate \( LS_j \), \( A \) has to produce a correct authenticated message \( M_{SU} = h(r_2||ID_{dc}||X_{LS}||ID_{DC}||KEY_{DC}) \). Due to \( A \) cannot obtain server’s master key \( X_{LS} \) and \( h(\cdot) \) is a secure hash function, he could not acquire \( X_i = h(d_i||X_{LS}||r_i) \). Therefore, \( A \) could not generate a valid \( M_{SU} \) and the proposed protocol resists server spoofing attack.

- **Anonymity**: As shown in Figure 5, \( U_i \)’s identity is included in \( M_2 = X_i \oplus (ID_i||r_i), M_3 = (ID_i||ID_{DC} \oplus h(X_i||r_i||T_i), M_{US} = h(ID_i||ID_{DC}||d_i||r_i||X_i||r_i||T_i) \). Because of secrecy of \( PW_i, \sigma_i, r_i, X_{LS} \), \( A \) cannot compute \( X_i \) to get \( ID_i \). Therefore, the Eland protocol holds user anonymity.

- **Offline password guessing attack resilience**: The adversary \( A \) may collect \( M_2, M_3, M_{US} \) to guess password \( PW_i \) which is included in \( X_i = h(ID_i||PW_i||\sigma_i|||X_{LS}||r_i) \). In order to derive \( PW_i, A \) must know the master key \( X_{LS} \), biometrics \( \sigma_i \), the random number \( r_i \) and the user’s real identity \( ID_i \). Due to user anonymity and only legitimate user/server could have \( (\sigma_i, r_i)||X_{LS} \), \( A \) cannot obtain these secret values to guess the password \( PW_i \).

- **Secure mutual authentication**: The above analysis indicates that \( A \) could neither impersonate a legitimate user nor replace license server by any type of attacks. Therefore, this protocol provides secure mutual authentication.

- **Three-factor security**: As shown in Figure 4, the smart device stores \( \{\theta_i, RPWi, f_i\} \). If \( f_i = h(d_i||RPWi \oplus d_i) \) then he could not acquire \( ID_i, PW_i, \sigma_i \). Even if user’s biometrics and smart device are exposed to \( A \), he still cannot guess correct password due to secrecy of master key \( X_{LS} \). Similarly, if \( A \) acquires password \( PW_i \) and biometric information \( \sigma_i \), he still cannot recover information stored in mobile device due to unknown \( r_i \) and \( X_{LS} \). Therefore, the proposed protocol provides secure mutual authentication.

- **Un-traceability**: As shown in Figure 5, messages in each authentication whether from the user side or the server side include random factors. On the one hand, \( M_1 \) is computed by \( d'_i, r'_i \) and \( ID_{DC} \) and we assume that the user will not request the same digital rights content \( ID_{DC} \) thus \( ID_{DC} \) guarantees randomness of \( M_1 \). In addition, \( \{M_2, M_3, M_{US}\} \) include the random number \( r_i \) which would not establish connection with other authentication messages. On the other hand, \( \{M_2, M_3, M_{SU}\} \) contain the random factor \( r_2 \) which is different in each authentication.

6 Performance Analysis

In this section, we compare our protocol with six authenticated and authorized access schemes for DRM which are published in recent three years, i.e. [6-10] and [20]. We only consider the main body of scheme i.e. the authentication and key distribution phase, which plays a key role in the authenticated digital content distribution. Then we will perform comparisons in three aspects of security, computation cost and communication cost.

6.1 Security Comparison (Table 2)

According to analysis of section 3.2, Yu et al.’s protocol cannot withstand impersonation attack, replay attack, stolen verifier attack and cannot provide secure authentication and user anonymity. Rana and Mishra’s [6], [7] do not provide

### 6.2 Computational Cost Comparison

We adopt the running time in [13] to count computation cost of each scheme’s main phase, where the experiment platform is Pentium IV computer with 512-MB RAM offering a maximum clock speed of 3 GHz. One hash operation (denoted as $T_h$) is 0.32ms, one fuzzy extractor (denoted as $T_{fe}$) is 17.1ms and one modular exponentiation (denoted as $T_{me}$) is 59.2ms. Due to the bit XOR operation cost is far from smaller than other operations, it is not counted in the computation cost comparison. According to Figure 5, authentication and key distribution of our protocol consumes $7T_h + 1T_{fe}$ i.e. 19.34ms in the user ($U_i$) side and $6T_h$ i.e. 1.92ms in the license server ($L_Sj$) side.

Computation cost statistics of other four schemes can be seen in Table 3, that is Yu et al.’s [10] consumes 18.7ms and 0.64ms respectively in $U_i$ side and $L_Sj$ side, Rana and Mishra’s [6] consumes 19.34ms and 1.28ms respectively in $U_i$ side and $L_Sj$ side, Mishra and Rana’s [7] consumes 19.98ms and 2.56ms respectively in $U_i$ side and $L_Sj$ side, Lee et al.’s [8] consumes 19.66ms and 2.88ms Lee et al.’s [9] consumes 19.66ms and 1.92ms respectively in $U_i$ side and $L_Sj$ side, Chen et al.’s [20] consumes 76.94ms and 119.04ms respectively in the $U_i$ side and $L_Sj$ side. It demonstrates that our protocol’s authentication and key distribution has lower computation cost than [8-9] and [20] in both $U_i$ side and $L_Sj$ side.

### Table 2. Security comparison

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Impersonation attack</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mobile device theft</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Offline password guessing</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Replay attack</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Stolen verifier</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Privileged insider attack</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Server spoofing attack</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Denial of service</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Three-factor security</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Secure authentication</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Anonymity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Un-traceability</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

### Table 3. Computation cost of authentication and key distribution phase: ms

<table>
<thead>
<tr>
<th>Protocols</th>
<th>$U_i$ Side</th>
<th>$L_Sj$ Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref. [10]</td>
<td>$1T_{fe} + 5T_h$ (18.7)</td>
<td>$2T_h$ (0.64)</td>
</tr>
<tr>
<td>Ref. [6]</td>
<td>$1T_{fe} + 7T_h$ (19.34)</td>
<td>$4T_h$ (1.28)</td>
</tr>
<tr>
<td>Ref. [7]</td>
<td>$1T_{fe} + 9T_h$ (19.98)</td>
<td>$8T_h$ (2.56)</td>
</tr>
<tr>
<td>Ref. [8]</td>
<td>$1T_{fe} + 8T_h$ (19.66)</td>
<td>$9T_h$ (2.88)</td>
</tr>
<tr>
<td>Ref. [9]</td>
<td>$1T_{fe} + 6T_h$ (19.66)</td>
<td>$6T_h$ (1.92)</td>
</tr>
<tr>
<td>Ref. [20]</td>
<td>$1T_{me} + 2T_h + T_{fe}$ (76.94)</td>
<td>$2T_{me} + 2T_h$ (119.04)</td>
</tr>
<tr>
<td>Ours</td>
<td>$1T_{fe} + 7T_h$ (19.34)</td>
<td>$6T_h$ (1.92)</td>
</tr>
</tbody>
</table>

### 6.3 Communication Cost Comparison

Considering the 1024-bit RSA public key cryptosystem security level and calculation requirement, we take the size of hash function output (denoted as $|h|$), timestamp (denoted as $|T|$), identifier (denoted as $|ID|$), random number (denoted as $|r|$) and modular number (denoted as $|n|$) are respectively 256 bits, 32 bits, 128 bits, 128 bits and 1024 bits. As shown in figure [Fig: Login and authentication], authentication and key distribution phase of our protocol separately consumes $4|h| + |1|T|$ in the $U_i$ side and $3|h|$ in the $L_Sj$ side.

Moreover, communication cost of other four schemes can be analyzed separately in the $U_i$ side and the $L_Sj$ side. Yu et al.’s [10] respectively requires $2|h| + 2|ID|$ and $3|h|$. Rana and Mishra’s [6] respectively requires $3|h| + |1|T|$ and $2|h| + |1|T|$. Mishra and Rana’s [7] respectively requires $4|h| + |1|T|$ and $2|h|$. Lee et al.’s [8] respectively requires $3|h| + |1|T|$ and $3|h| + |1|T|$. Lee et al.’s [9] respectively requires $4|h| + |1|ID| + |1|T|$ and $3|h| + |1|r| + |1|T|$. Chen et al.’s [20] respectively requires $2|n| + |1|r| + 2|ID| + |1|T|$ and $1|n| + |1|r| + 2|ID| + |1|T|$. The comparison results are shown in Figure 7 and Figure 8, which indicates that our protocol needs narrower communication bandwidth than Lee et al.’s [9] and Chen et al.’s [20] in both user side and license server side.

![Figure 7. User side communication cost](image-url)
encryption and authentication technologies. But these security schemes employ complex operations and cannot realize lightweight.

8 Conclusion

This paper proposed a new lightweight anonymous authentication protocol called Eland, which only adopts time-saving hash function and XOR operations. Eland provides protocol’s security and user anonymity and is equipped with 13 security properties according to informal security analysis. We also introduced ROR threat model and proved Eland’s mutual authentication security in the model which means the hash function security ensures Eland’s security. Additionally, we compared Eland with six recent authentication protocols in aspects of security and performance. Comparisons result indicated that our protocol realizes a well trade-off between security and performance. Thereby, Eland is an efficient protocol for DRM system.

9 Acknowledgement

The work was supported by the National Natural Science Foundation of China (Nos. 62172307, 61972294, 61932016), the Blockchain Core Technology Strategic Research Program of Ministry of Education of China (No. 2020KJ010301), the Shandong Provincial Key Research and Development Program (No. 2020CXGC010107), the Special Project on Science and Technology Program of Hubei Province (No. 2020AEA013), the Natural Science Foundation of Hubei Province (No. 2020CFA052), the Wuhan Municipal Science and Technology Project (No. 202010601012187) and the Foundation of Guangxi Key Laboratory of Trusted Software and Technology Project (No. kx202001).

References


[34] D. Kumar, S. Chand, B. Kumar, Cryptanalysis and improvement of a user authentication scheme for wireless sensor networks using chaotic maps, *IET Networks*, Vol. 9, No. 6, pp. 315-325, November, 2020.


**Biographies**

**Qing Fan** received the B.S. degree in applied mathematics from School of Mathematics and Statistics, Shandong Normal University, Jinan, China, in 2017. She is currently taking a successive postgraduate and doctoral program at School of Mathematics and Statistics, Wuhan University, Wuhan, China. Her research interests include cryptography and information security.

**Jianhua Chen** received the Ph.D. degrees in applied mathematics from Wuhan University, Wuhan, China, in 1994. He is currently a Professor with School of Mathematics and Statistics, Wuhan University. His current research interests include number theory, cryptography, and information security.

**Yihong Wen** received his Ph.D. degree in computer application from BeiHang University in 2011. He is currently a senior engineer of CETC54. His research interests mainly include remote sensing image analysis and processing, aerospace blockchain, aerospace System simulation.

**Min Luo** received the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2003. He is currently an Associate Professor with the School of Cyber Science and Engineering, Wuhan University. His research interests mainly include applied cryptography and blockchain technology.