An E-lottery System with a Fair Purchasing Environment and an Arbitration Mechanism

Chin-Ling Chen¹²³, Yuan-Hao Liao³, Fang-Yie Leu⁴, Ilsun You⁵, Kim-Kwang Raymond Choo⁶, Chia-Yin Ko⁴

¹ School of Computer and Information Engineering, Xiamen University of Technology, China
² School of Information Engineering, Changchun Sci-Tech University, China
³ Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taiwan
⁴ Department of Computer Science, Tunghai University, Taiwan
⁵ Department of Information Security Engineering, SooChunHyang University, Republic of Korea
⁶ Department of Information Systems and Cyber Security, The University of Texas at San Antonio, USA

clc@mail.cuyt.edu.tw, jackmanliao@gmail.com, leufy@thu.edu.tw, ilsunu@gmail.com, Raymond.Choo@utsa.edu, pess@thu.edu.tw

Abstract

Lottery is an attractive game as the winning player can potentially receive a huge amount of prize money. Advances in Internet and communications technologies and the popularity of online shopping resulted in proxy-purchasing-services, and more recently there have been attempts to digitize lotteries into ‘real’ electronic lottery (E-lotteries; i.e. moving away from websites that merely provide proxy purchasing services). However, there are challenges underlying existing E-lotteries. For example, the lottery originator (LO) may forge a winning player and share the prize with the actual winners, by purchasing one or more lottery tickets of the winning number before publishing the number. Other concerns include exploitation by malicious employees of lottery providers. In this paper, we use the verifiable random function, digital signature algorithm (DSA) and bulletin board mechanism to establish an E-lottery system, which has a fair and secure purchasing environment and an arbitration mechanism. The former ensures the rights for both the players and LOs, whereas the latter allows the resolution of a dispute and protects the rights and interests of players as well as lottery providers.

Keywords: Fairness, Non-repudiation, Arbitration, E-lottery, Public verification

1 Introduction

Lotteries have the properties of unpredictability and significant payouts, and players have the opportunity to win significant cash prizes by spending only a small amount of money. Hence, lottery is a very hard to resist attraction (similar to other forms of gambling). In a lottery, players first select their favorite numbers. After the deadline of the purchasing phase, the lottery originator (LO) randomly generates the winning numbers. If no one wins the jackpot, then the prize will snowball to the next round.

There are number of websites, such as Lottery [1], and LoveMyLotto [2], that provide a trading platform (TP) for proxy purchasing services. These online purchasing proxy services enables player to bet on their favorite numbers. On behalf of the players, the websites will purchase the lottery tickets from the trusted LO. Subsequently, the TP will send a scanned a copy of the lottery ticket to the user as evidence of purchase. Holders of winning ticket can claim the prize via the TP’s website. There are limitations in existing system.

(1) Potential risks to a player:
- An employee or the TP owner may abscond with the winnings, particularly if the player has won the grand prize.
- Evidence of purchase may be refuted or can be fabricated.
- Loss of the purchase information may affect the ability of the player to claim the winnings.

(2) Potential risks to a TP:
- Evidence of purchase may be forged.

Not surprising, how to design a fair and secure electronic lottery (E-lottery) protocol has been studied in the literature. For example, the authors in [3-9] attempted to digitize lotteries and transform them into ‘real’ E-lotteries rather than having only a proxy purchase service. Specifically, Chow et al. [3] proposed practical E-lotteries with an offline trusted third party (TTP) [10]. The scheme satisfies most of the identified requirements without the presence of TTP when generating the winning numbers. Although it is publicly verifiable, when a dispute occurs, its arbitration mechanism is not a fully trusted platform in...
the real world.

Lee and Chang [4] introduced an electronic t-out-of-n lottery which is developed following the Chinese Remainder Theorem, allowing lottery players to simultaneously select t out of n numbers in a ticket without an iterative selection. The drawback of the scheme is that there is significant computational overhead for players during purchasing of the lotteries. Lee et al. [5] proposed a non-iterative privacy preservation scheme for E-lotteries, allowing a player to choose t-out-of-n numbers without an iterative selection, and preserving the privacy of the player’s choice. Similar to the work in [4], the computational overhead imposed on the players during purchasing of the lotteries is significant. Thus, neither schemes are suitable for mobile or wireless device users (e.g., players seeking to purchase the lotteries on their mobile devices).

Afterward, Lee et al. [6] based on Aryabhata remainder theorem to realize e-lottery games. The main contribution of this work is to guarantee the security of this multi-billion-dollar game industry and realize the mechanism. Next, Chen et al. [7] proposed a lottery protocol which allowed single player to join a lottery entry to purchase lottery in a mobile environment. It is a novel application compared to previous works. The mobile user can easily purchase lottery numbers using their smart device. The proposed protocol can prevent the various malicious attacks. Recently, some authors [8-9] used the blockchain mechanism to design lottery systems. This decentralized autonomous mechanism is applied to E-lottery scope. But the detail scenarios are not clear and the analysis is roughly.

We also observe that these schemes mainly focus on preserving the players’ privacy or proposing an arbitration mechanism. For example, Chen and Liao [11] proposed to address the transaction fairness problem between players and the TP using subliminal channel [11-14] and arbitration scheme. They also proposed a trustable transaction application based on these two mechanisms.

There are, however, limitations in existing work:

1. Due to the electronic nature of E-lotteries, they can be easily copied and deleted, particularly in the presence of a malicious insider (e.g., LO employee).
2. Lack of suitable mechanisms to prevent a malicious insider from purchasing winning tickets after the winning numbers have been generated.
3. When disputes occur, there is a lack of suitable arbitration mechanism to adjudicate the arbitration in practice.

In this study, we integrate several schemes, including the arbitration approach proposed by Chen and Liao [11], the bulletin board mechanism, verifiable random function [15] and digital signature algorithm (DSA) [16], to design a fair E-lottery system. Specifically, the proposed scheme is designed to mitigate insider collusion and provide an arbitration mechanism. We also remark that our proposed E-lottery system also fulfills the following requirements, typically expected of a secure and practical E-lottery system [3-5]:

1. Public verification. All valid lottery tickets and winning numbers need to be verifiable via a verifiable function.
2. Fairness. No one is able to predict the winning numbers before the numbers are published.
3. Security. No one is able to forge a winning lottery or impersonate a winner to claim the prize.
4. Correctness. The players can verify the correctness of published information via the bulletin board by themselves.
5. Anonymity. No one can identify the participants through the information in a lottery ticket.
6. Convenience. Legitimate players can purchase lottery tickets online (e.g., using an Internet-connected mobile device).
7. Without pre-registration. A player does not need to register with any lottery agent in advance. This requirement should conform to general purchasing behaviors of an electronic lottery to make it more practical.
8. No online Trusted Third Party (TTP). If the security of the mechanism relies on another online TTP of an E-lottery, it is said to be impractical.

In the next section, we will introduce the background materials. Section 3 proposes our protocol, and its security analysis is outlined in Section 4. Section 5 concludes this paper.

2 Preliminary

In this section, we introduce two cryptographic methods employed in this study, namely: a verifiable random function (VRF) [15] and a digital signature algorithm (DSA) [16].

2.1 Verifiable Random Function (VRF)

A VRF, first proposed by Micali et al. [14], is essentially a pseudorandom function [17-18] which provides non-interactively verifiable proof of the output’s correctness. VRF is based on Identity-based Key Encapsulation [19], a variant of decisional Diffie-Hellman (DDH) assumption [20], and decisional bilinear Diffie-Hellman inversion assumption (DBDHI) [21]. VRF can also be implemented using digital signature schemes, such as RSA [22], DSA [16] and ECDSA [23] – see Table 1.
Table 1. A brief overview of three digital signature schemes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>RSA [22]</th>
<th>DSA [16]</th>
<th>ECDSA [23]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Comparison item</strong></td>
<td>Signature length</td>
<td>Digital Signatures Security Strength</td>
<td>Security Basis</td>
</tr>
<tr>
<td><strong>(≥ 112 bits)</strong></td>
<td></td>
<td>(≥ 112 bits)</td>
<td>Prime factorization problem</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Discrete logarithm problem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(≥ 112 bits)</td>
<td>Discrete logarithm problem</td>
</tr>
</tbody>
</table>

Note. n, p, q, r and s are parameters of security length of the related scheme. The signature verification process is shown below.

As observed from Table 1, to achieve similar security strength transitions (≥ 112 bits), ECDSA are shorter than that of RSA [22] and the signature lengths of DSA. It is also clear that in comparison to ECDSA, DSA has less complexity. Therefore, we adopt the DSA as an example to implement VRF.

Lysyanskaya [23] presented a set of functions $F_{y}(*) : \{0, 1\}^{k} \rightarrow \{0, 1\}^{c(k)}$ which is verifiable. Adapting the concepts introduced in [24], we define the following polynomial-time functions:

1. **Gen (k)**: a probabilistic function that produces a secret key $SK$ by invoking a random function, and provides public verification using the corresponding public key $PK$.
2. **Eval (SK, x)**: a function that calculates the value $y=FPK(x)$.
3. **Prove (SK, x)**: a function that proves $\pi$ using $y=FPK(x)$.
4. **Verify (PK, x, y, $\pi$)**: a function that verifies $y=FPK(x)$ using the proof $\pi$.

Details will be presented later.

VRF should satisfy the following properties:

1. **Uniqueness:**
   - $\text{Verify}(PK, x, y_1, \pi_1) = \text{Verify}(PK, x, y_2, \pi_2)$
   - if $y_1 = y_2$.
2. **Computability:** $\text{Eval}(SK, x) = F_{y}(x)$ is efficiently computable.
3. **Provability:** if $(y, \pi) = \text{Prove}(SK, x)$, then $y(PK, x, y, \pi) = 1$.
4. **Pseudorandomness:** The probability that an attacker inputs data $x$ of an arbitrary number in bits to $F_{y}(x)$. A pseudorandom process [18] is a process that generates a number sequence appearing to be random, but it is not absolutely random. Pseudorandom sequences typically exhibit statistical randomness which is produced by an entirely deterministic causal process.

### 2.2 Digital Signature Algorithm (DSA)

DSA is a Federal Information Processing Standard (FIPS) for digital signatures. The National Institute of Standards and Technology (NIST) proposed DSA in August 1991 for use in their Digital Signature Standard (DSS), and subsequently adopted as FIPS 186 in 1993 [16].

Let $M$, $r'$, and $s'$ be, respectively, the received versions of $M$, $r$, and $s$.

1. **P**: a prime modulus, $2L–1 < p < 2L$, where $L$ is the length of $p$ in bits.
2. **Q**: a prime divisor of $(p – 1)$, $2N–1 < q < 2N$, where $N$ is the length of $q$ in bits.
3. **G**: a generator of the subgroup of order $q mod p$, such that $l < g < p$.
4. **X**: a private key which remains secret; $x$ is a random or pseudo random integer, $0 < x < q$, i.e., $x$ is in $[1, q–1]$. 
5. **Y**: a public key, $y = g^x mod p$.
6. **k**: a secret number that is unique to each message; $k$ is a randomly or pseudo random integer, such that $0 < k < q$, i.e., $k$ is in the range of $[1, q–1]$. The signature of a message $M$ consists of two numbers $r$ and $s$ where $r = (g^x mod p) mod q$ and $s = k^{-1}(M + xr) mod q$.

**Step 1.** The verifier checks to determine whether $0 < r' < q$ and $0 < s' < q$; if at least one of the conditions is violated, then the signature shall be rejected. Otherwise, the verifier computes the following parameters.

- $w = s' - 1 mod q$.
- $u1 = M'r mod q$.
- $u2 = r'w mod q$.
- $v = g^w y^{s'}(mod p) mod q$.

**Step 2.** If $v = v'$, then the signature is successfully verified.

### 2.3 Constructing the Session Key Model

Diffie and Hellman [25-27] introduced a key agreement protocol (RFC 2631), which was adopted by IETF in 1999 [28]. Other security schemes can be found in [29-30]. In this paper, we utilize this protocol to establish session keys which are used in our protocol under two situations. First, the player must share the session key with the bank to protect his/her account information when the player purchases a lottery ticket. Second, the player encrypts the purchase details with the session key before sending the information to LO.
3 Our Proposed Scheme

The participants in the proposed scheme (refer to Figure 1) are described as follows:

Figure 1. The structure of our scheme

(1) Player (P).
(2) Lottery originator (LO). The lottery originator issues the E-lottery, draws the winning numbers, and presents the winnings.
(3) Bank (B). The bank is a fair financial institution.
(4) Arbiter (A). The arbiter is tasked with arbitration in the event of a dispute.

Step 1. $P \rightarrow B$: Player (P) opens a new account with a participating bank, say Bank B.
Step 2. $P \rightarrow LO$: Player purchases an E-lottery with LO.
Step 3. $LO \rightarrow B$: The LO forwards the account information of the player to the bank, and the bank withdraws the designated amount from the player’s account and transfers the amount into LO’s account. Finally, the bank confirms the result of the transfer to LO.
Step 4. $LO \rightarrow P$: The LO issues the E-lottery to the player.
Step 5. LO: The LO draws the winning numbers and publishes them on the bulletin board.
Step 6. $P \rightarrow LO$: The winning player submits a claim, and his/her identity will be verified by the LO. Once the identity of the (winning) player is verified, the winnings will be paid to the player.
Step 7. $P \rightarrow A$: The player submits the arbitration to the arbiter (A) if a dispute occurs.

A summary of the notations employed is shown in Table 2.

Table 2. Summary of notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_i$</td>
<td>secret message which is the evidence selected by the $i$th player and can be used in an arbitration</td>
</tr>
<tr>
<td>$k_i$</td>
<td>random number selected by the $i$th player</td>
</tr>
<tr>
<td>number</td>
<td>set of lottery numbers selected by the $i$th player</td>
</tr>
<tr>
<td>$\alpha_i$</td>
<td>$i$th player’s pseudo name, where $\alpha_i = h(\text{number}_i</td>
</tr>
<tr>
<td>$\mu_i$</td>
<td>$i$th player’s index on the database of the bank which is used to search for the corresponding session key $K_{P-B}$</td>
</tr>
<tr>
<td>$X$</td>
<td>role variable of participant of an E-lottery system; the value maybe player, LO, bank or arbiter</td>
</tr>
<tr>
<td>ID$_X$</td>
<td>$X$’s identity</td>
</tr>
<tr>
<td>Accounts$_X$</td>
<td>$X$’s bank account</td>
</tr>
<tr>
<td>Amount</td>
<td>amount of the lottery prize</td>
</tr>
<tr>
<td>$p_s$</td>
<td>prime number selected by $X$, $2^{L-1} &lt; p_s &lt; 2^L$, which $L$ is the length of $p_s$ in bits</td>
</tr>
<tr>
<td>$q_s$</td>
<td>prime number selected by $X$, where $q_s$ is a divisor of $(p_s - 1)$, $2^{N-1} &lt; q_s &lt; 2^N$, and $N$ is the length of $q_s$ in bits</td>
</tr>
<tr>
<td>$g_{CA}$</td>
<td>generator of the subgroup of order $q_s$ mod $p_s$, where $g_{CA}, 1 &lt; g_{CA} &lt; p_s$, are selected by the certificate authority (CA)</td>
</tr>
<tr>
<td>$(x_s, y_s)$</td>
<td>(X’s private key, X’s public key) generated by the CA. $x_s$ should remain secret, where $x_s$ is a random or pseudo random integer, $0 &lt; x_s &lt; q_s$; the $y_s$ is the corresponding public key, where $y_s = g_{CA}^{x_s}$ mod $p_s$</td>
</tr>
<tr>
<td>$\sigma_i$</td>
<td>arbitration evidence generated by the $i$th player, where $\sigma_i = y_{LO}^{h(M_i)</td>
</tr>
<tr>
<td>$E(K_{X,Y}, m)$</td>
<td>symmetric encryption function which encrypts message $m$ with session key $K_{X,Y}$</td>
</tr>
<tr>
<td>$D(K_{X,Y}, m)$</td>
<td>symmetric decryption function which decrypts message $m$ with the session key $K_{X,Y}$</td>
</tr>
<tr>
<td>$h(\cdot)$</td>
<td>one-way hash function</td>
</tr>
</tbody>
</table>
Table 2. Summary of notations (continue)

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>number of lottery tickets having sold</td>
</tr>
<tr>
<td>chain$_i$</td>
<td>published hash chain set which is used to generate the winning number where a valid seed $\alpha_i$ is generated by the $i$th player, and chain$_0=0$ is the initial vector, chain$_1=h$ (chain$_0$, $\alpha_1$), chain$_2=h$ (chain$_1$, $\alpha_2$), ..., chain$<em>f=h$ (chain$</em>{f-1}$, $\alpha_f$)</td>
</tr>
<tr>
<td>Prize</td>
<td>winning prize</td>
</tr>
<tr>
<td>Rand(.)</td>
<td>public pseudo-random number generation function</td>
</tr>
<tr>
<td>Eval(.)</td>
<td>winning number generation function</td>
</tr>
<tr>
<td>Prove(.)</td>
<td>proof function</td>
</tr>
<tr>
<td>Verify(.)</td>
<td>public verification function</td>
</tr>
<tr>
<td>seed</td>
<td>random seed generated by the public pseudo-random number generating function as seed = Rand (chain$_1$, chain$_2$, ..., chain$_f$)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>$A = B$</td>
<td>determine if A is equal to B</td>
</tr>
<tr>
<td>☜ ➔ secure channel</td>
<td></td>
</tr>
<tr>
<td>➔ ——— insecure channel</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. Summarizes the lottery purchasing phase
Step 1. The ith player Pi and bank B establish a session key K_{Pi,B}. The ith player Pi:
1. selects a favorite number number r_i and a random number k_i, and then computes \( \alpha_i \) as the pseudo name of the ith player with a one-way hash function where
\[
\alpha_i = h(\text{number } r_i || k_i) \quad (1)
\]
1.2 creates a secret message \( M_i \) and uses LO’s public key to compute \( \alpha_i \) as:
\[
\alpha_i = y_i^{h(M_i)} \mod p_{CA} \quad (2)
\]
where \( \alpha_i \) and \( M_i \) are evidences in the arbitration phase.
1.3 uses the hash function to encrypt secret message \( M_i \) so as to generate index \( \mu_i \) as the ith player’s index in the session-key database of the bank used to record the corresponding session key \( K_{Pi,B} \) where
\[
\mu_i = h(M_i) \quad (3)
\]
establishes the session key \( K_{Pi,B} \) with B.
1.4 utilizes session key \( K_{Pi,B} \) to protect the account information, including the purchase amount of lottery (Amount), Pi’s identity (ID_p) and Pi’s account (Account_p) as follows:
\[
C_{Pi,B} = E(K_{Pi,B}, (\text{Amount} \mid \text{ID}_p \mid \text{Amount}_p)) \quad (4)
\]
1.5 sends \( \mu_i \) to B.

Step 2. B stores the session key \( K_{Pi,B} \), and \( \mu_i \) together as a record in its session-key database.

Step 3. Pi establishes the session key with LO, and sends the corresponding purchasing request to LO.

P_i then
3.1 establishes the session key \( K_{Pi,LO} \) with LO.
3.2 uses \( K_{Pi,LO} \) to protect the purchase information, including the pseudo name \( \alpha_i \), arbitration evidence \( \alpha_i \), index \( \mu_i \), ciphertext \( C_{Pi,LO} \) and lottery amount Amount, as a ciphertext \( C_{Pi,LO} \) where
\[
C_{Pi,LO} = E(K_{Pi,LO}, (\alpha_i \mid \sigma_i \mid \mu_i \mid C_{Pi,LO} \mid \text{Amount})) \quad (5)
\]
3.3 sends \( C_{Pi,LO} \) and the corresponding purchasing request \( m_{purch} \) to LO.

Step 4. LO sends the transaction information to B. Upon receiving the message sent by P_i, LO
4.1 decrypts the \( C_{Pi,LO} \) with session key \( K_{Pi,LO} \) as:
\[
(\alpha_i \mid \sigma_i \mid \mu_i \mid C_{Pi,LO} \mid \text{Amount}) = D(K_{Pi,LO}, (C_{Pi,LO})) \quad (6)
\]
sends the transaction information, including the lottery amount \( \text{Amount}' \) provided by LO, LO’s account \( \text{Account}_{LO} \), P_i’s pseudo name \( \alpha_i \), index \( \mu_i \) and ciphertext \( C_{Pi,B} \) to B via a secure channel.

Step 5. P_i proceeds payment, and B sends the payment message to LO.

Upon receiving the message, B
5.1 looks up \( K_{Pi,B} \) in its session-key database according to index \( \mu_i \), and decrypts \( C_{Pi,B} \) as:
\[
(\text{Amount} \mid \text{ID}_p \mid \text{Amount}_p) = D(K_{Pi,B}, (C_{Pi,B})) \quad (7)
\]
5.2 checks to see whether the lottery amount specified by LO is equal to the lottery amount \( \text{Amount} \) provided by P_i.
5.3 If Eq. (8) holds, B further checks to see whether the amount in P_i’s account is equal to or greater than the lottery amount \( \text{Amount} \).
5.4 If the above checks are both correct, then B performs the following processes.
(1) using Schnorr’s signature mechanism [31] to sign the pseudo name \( \alpha_i \) by computing
\[
N_i = g^{u_i} \mod p_{CA}, \text{ where } u_i \in \mathbb{Z}_{p_{CA}} \quad (9)
\]
\[
\gamma_i = h(N_i, \alpha_i) \quad (10)
\]
\[
s_i = (\gamma_i - x_{LO} \cdot \gamma_i) \mod q_{CA} \quad (11)
\]
\[
\delta_{\alpha_i} = (\gamma_i, s_i) \quad (12)
\]
(2) transferring the lottery amount \( \text{Amount}' \) from Pi’s account \( \text{Account}_p \) into LO’s account \( \text{Account}_{LO} \).
(3) setting a validation parameter \( V = \text{“valid”} \).
5.5 If at least one of the above checks is incorrect, the B sets.
If V is “valid”, B sends a message to the LO; otherwise, B sends a message containing V only to LO.

Step 6. LO publishes P_i’s pseudo name and signature on the bulletin board.
6.1 When receiving a message from B, if V = “valid”, LO publishes P_i’s pseudo name and signature on the bulletin board, and then proceeds to the next phase. Otherwise, LO replies the message “transaction failure” to P_i.
After the above procedures, the player completes the lottery purchasing phase.

3.3 The Lottery Ticket Issuing Phase

LO generates the related parameters for the lottery, links random seed with the hash chain, publishes the hash chain value on the bulletin board, and issues the lottery tickets to their players. When a player receives the lottery ticket, he/she uses the information on the
bulletin board to verify whether the selected random seed is included in the hash chain. The published hash chain mechanism is to guarantee that the winning numbers are unpredictable in our scheme. The information shown on the bulletin board for sold lottery tickets is outlined in Table 3.

### Table 3. Information published on bulletin board for sold lottery tickets

<table>
<thead>
<tr>
<th>Number of lottery tickets sold</th>
<th>Hash chain value</th>
<th>Player’s pseudo name</th>
<th>Signature of player’s pseudo name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>chain₁</td>
<td>α₁</td>
<td>δ₁</td>
</tr>
<tr>
<td>2</td>
<td>chain₂</td>
<td>α₂</td>
<td>δ₂</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
<tr>
<td>i</td>
<td>chainᵢ</td>
<td>αᵢ</td>
<td>δᵢ</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
<tr>
<td>f</td>
<td>chainₙ</td>
<td>αₙ</td>
<td>δₙ</td>
</tr>
</tbody>
</table>

After a player, e.g., pf (i.e., pᵢ=pf), has purchased a lottery ticket, LO will publish a record on the bulletin board containing the hash chain values (i.e. where \(\text{chain}_0=0\), \(\text{chain}_1=h(\text{chain}_{0}, α_1)\), \(\text{chain}_2=h(\text{chain}_1, α_2)\), ..., \(\text{chain}_f=h(\text{chain}_{f-1}, α_f)\)), the f’s pseudo name \(w_f=h(\text{number}_f || k_f)\) and the signature of f’s pseudo name \(δ_f=(r_f, s_f)\) where \(1 ≤ f\), and \(f\) is the number of lottery ticket that has been so far sold. The procedure of the lottery ticket issuing phase is shown below and illustrated in Figure 3.

![Figure 3. Lottery ticket issuing phase](image)

**Step 1.** LO makes a digital signature of lottery.

1.1 defines the lottery \(\text{Lottery}_i\) as:

\[
\text{Lottery}_i = (σ_i || δ_n) \quad \text{for} \quad i=1, 2, ..., f;
\]

subsequently links Pi’s pseudo name \(α_i\) on the hash chain as:

\[
\text{chain}_i = h(\text{chain}_{i-1}, δ_n) \quad \text{for} \quad i=1, 2, ..., f;
\]

1.2 publishes \((\text{chain}_n, σ_i)\) on the bulletin board.

1.3 uses the session key \(K_{P_{-LO}}\) to protect the ith lottery Lotteryi, the hash chain value chaini and the ith number of lottery tickets that have been so far sold as:

\[
C_{\text{Lottery}_i} = E(K_{P_{-LO}}, (\text{Lottery}_i \parallel \text{chain}_i \parallel i));
\]

1.4 uses Schnorr’s signature mechanism to sign the ciphertext \(C_{\text{Lottery}_i}\) as computing:

\[
\begin{align*}
(1) \quad N_i &= g_{CA}^{u_2} \mod p_{CA}, \quad \text{where} \quad u_2 \in \mathbb{Z}_{p_{CA}}; \\
(2) \quad γ_i &= h(N_i, C_{\text{Lottery}_i}); \\
(3) \quad s_i &= (u_2 - x_{LO} \cdot r) \mod q_{CA}; \\
(4) \quad δ_{C_{\text{Lottery}_i}} &= (γ_i, s_i);
\end{align*}
\]

1.5 defines the signature of ciphertext \(C_{\text{Lottery}_i}\) as:

\[
\delta_{C_{\text{Lottery}_i}} = (γ_i, s_i)
\]
1.6 sends the message \((\delta_{\text{chain}^i}, C_{\text{Lottery}})\) to the \(P_i\).

**Step 2.** The player \(P_i\) verifies digital signature and checks to see whether the hash chain on bulletin board is valid. \(P_i\) then

1.1 uses the LO’s public key \(y_{LO}\) to compute:

\[
\gamma_i' = g_{CA}^{y_{LO}} \cdot y_{LO}^{\gamma_i} \mod p_{CA}\tag{20}
\]

checks to see whether

\[
\gamma_i = h(\gamma_i', C_{\text{Lottery}})\tag{21}
\]

2.2 utilizes the session key \(K_{P_i,\text{LO}}\) to decrypt the ciphertext \(C_{\text{Lottery}}\) as:

\[
(Lottery_i, || \text{chain}_i, || i) = D(K_{P_i,\text{LO}}, (C_{\text{Lottery}}))\tag{22}
\]

2.3 integrates the pseudo name \(\alpha_i\) into the hash chain \(\text{chain}_i\), and checks to see whether the hash chain on the bulletin board is valid, as:

\[
\text{chain}_i = h(\text{chain}_{i-1}, \alpha_i)\tag{23}
\]

### 3.4 The Winning Numbers Generation and Verification Phase

After the lottery purchasing deadline has passed, LO uses the winning number generation function and the random seed to generate the winning numbers, and uses a proof function to compute the proof value. LO then publishes the random seed, winning numbers and proof value on the bulletin board.

If the players wish to determine whether LO is honest, they can use the public verification function to verify the correctness of the winning numbers.

In our scheme, VRF is implemented based on the structure of Schnorr’s signature mechanism [31]. Here, \(y = \text{Eval}(SK, x)\), \(\pi = \text{Prove}(SK, x)\) and \(y = \text{WinNum}\) maybe true or false. The output is \(y = \text{WinNum}\), and the input parameters are \(SK = x_{LO}\), \(x = \text{seed}\) and \(PK = y_{LO}\).

Next, we describe the algorithm of VRF. The winning number generation function \(\text{Eval}(X_{LO}, \text{seed})\) is defined as follows. \(\text{Eval}(X_{LO}, \text{seed})\)

\[
\{ \text{The public key } y_{LO} = g_{CA}^{x_{LO}} \mod p_{CA}; \tag{24} \\
r = g_{CA}^{\text{seed}} \mod p_{CA}; \tag{25} \\
\text{WinNum} = h(r, y_{LO}); \tag{26} \\
\}
\]

The proof function \(\text{Prove}(x_{LO}, \text{seed})\) is defined as follows.

\[
\pi = (\text{seed} - (\text{WinNum} \cdot x_{LO})) \mod q_{CA}\tag{27}
\]

The public verification function \(\text{Verify}(y_{LO}, \text{seed}, \text{WinNum}, \pi)\) is defined as follows. \(\text{Verify}(y_{LO}, \text{seed}, \text{WinNum}, \pi)\)

\[
\{ r = g_{CA}^{\text{seed}} \mod p_{CA}; \tag{28} \\
g_{CA}^{\text{seed} \cdot y_{LO}} \mod p_{CA} = g_{CA}^{\text{seed} \cdot (x_{LO} \cdot \text{WinNum})} \mod p_{CA} \tag{29} \\
\}^*\text{Refer to Eqs. (27) and (24)}/

\[
= g_{CA}^{\text{seed}} \mod p_{CA} = r'; \tag{30}
\]

IF \(\gamma = \gamma'\) and \(\text{WinNum} = h(\gamma', y_{LO})\), return true; \(\text{Verify}(y_{LO}, \text{seed}, \text{WinNum}, \pi)\) is defined as follows.

**Step 1.** LO generates and publishes the verification parameters on the bulletin board. LO first

1.1 uses the pseudorandom generator \(\text{Rand}(\ )\) to derive random seed \(\text{seed}\) from the hash chain value \(\text{chain}_i\), for \(i = 1 \to f\) as

\[
\text{seed} = \text{Rand}(\text{chain}_1, \text{chain}_2, ..., \text{chain}_f)\tag{32}
\]

where \(\text{chain}_i = (\text{chain}_1, \alpha_i)\) and \(f\) is the total number of lottery tickets sold.

1.2 utilizes the winning numbers generation function \(\text{Eval}(\cdot)\) to calculate the winner numbers \(\text{WinNum}\) given private key \(x_{LO}\) and the random seed \(\text{seed}\).

\[
\text{WinNum} = \text{Eval}(x_{LO}, \text{seed}); \tag{33}
\]

1.3 employs the proof function \(\text{Prove}(\cdot, \cdot)\) to compute the proof value \(\pi\) given the parameters \(x_{LO}\) and \(\text{seed}\).

\[
\pi = \text{Prove}(x_{LO}, \text{seed})\tag{34}
\]

1.4 publishes the verification parameters \((\text{seed}, \text{WinNum}, \pi)\) on the bulletin board.

**Step 2.** \(P_i\) checks the correctness of the winning number.

2.1 After the winning numbers are published, \(P_i\) can check the correctness of the winning numbers \(\text{WinNum}\) via the public verification function \(\text{Verify}(\cdot, \cdot, \cdot)\) to see whether

\[
\text{Verify}(y_{LO}, \text{WinNum}, \text{seed}, \pi) = \text{true}; \tag{35}
\]

The information of the lottery winners published on the bulletin board is listed in Table 4.

### 3.5 The Claim Prize Phase

After the winning numbers have been published, the winner can submit the related parameters to claim the winnings. To prevent double or multiple claims using the same winning lottery, LO records the information
of the winning lottery in the database. Once the LO has
given out the winnings, related parameters of the
winning lottery ticket(s) will be published on the
bulletin board.

**Step 1.** The winner encrypts the information of the
claim prize and sends the protected information to the
LO.

1.1 The winner uses the session key $K_{P_i-LO}$ to
encrypt the information of the claim prize, including
the ciphertext, the signature $\delta_{\text{Lottery}}$ and the random
number $k_i$ as:

$$C_{P_i-LO} = E(K_{P_i-LO}, (C_{\text{Lottery}}, || \delta_{\text{Lottery}}, || k_i))$$  \hspace{1cm} (36)

1.2 sends $(C_{P_i-LO}, m_{\text{claim}})$ to LO.

The winning number generation and verification
phase is illustrated in Figure 4.

![Figure 4. Winning numbers generation and verification phase](image)

1.3 The winner uses the session key $K_{P_i-LO}$ to
encrypt the information of the claim prize, including
the ciphertext, the signature $\delta_{\text{Lottery}}$ and the random
number $k_i$ as:

$$C_{P_i-LO} = E(K_{P_i-LO}, (C_{\text{Lottery}}, || \delta_{\text{Lottery}}, || k_i))$$  \hspace{1cm} (37)

1.4 sends $(C_{P_i-LO}, m_{\text{claim}})$ to LO.

**Step 2.** LO checks the winner’s information, and sends
the awarded certificate (with signature) to the winner.
Upon receiving this message, LO

2.1 uses the session key $K_{P_i-LO}$ to decrypt the
ciphertext $C_{\text{Lottery}}$ as:

$$\gamma_i = h(\gamma'_i, C_{\text{Lottery}})$$  \hspace{1cm} (40)

2.3 If they are equal, LO decrypts the ciphertext
$C_{\text{Lottery}}$ with the session key $K_{P_i-LO}$ as:

$$(|| \text{chain}_i || i) = D(K_{P_i-LO}, (C_{\text{Lottery}}))$$ \hspace{1cm} (41)

2.4 uses the random number $k_i$ and the winner
numbers WinNum to compute the parameter $\alpha'_i$ as:

$$\alpha'_i = h(\text{WinNum} || k_i)$$ \hspace{1cm} (42)

2.5 checks to see whether the equation holds as:

$$\gamma_i = \alpha'_i$$ \hspace{1cm} (43)

2.6 checks to see whether the pseudo name $\alpha_i$ was
published on the bulletin board.

<table>
<thead>
<tr>
<th>Number of the winning lottery</th>
<th>Hash chain value of the winner</th>
<th>Winner’s pseudo name</th>
<th>Signature of the winner’s pseudo name</th>
<th>Winner’s random number</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>chain$_3$</td>
<td>$\alpha_3$</td>
<td>$\delta_{\alpha_3}$</td>
<td>$k_3$</td>
</tr>
<tr>
<td>5</td>
<td>chain$_5$</td>
<td>$\alpha_5$</td>
<td>$\delta_{\alpha_5}$</td>
<td>$k_5$</td>
</tr>
<tr>
<td>9</td>
<td>chain$_9$</td>
<td>$\alpha_9$</td>
<td>$\delta_{\alpha_9}$</td>
<td>$k_9$</td>
</tr>
</tbody>
</table>

**Note.** Assume that the third, fifth and ninth players are winners.
2.7 If the above checks are all correct, \( P_i \) is a winner. LO provides the award-winning information \( m_{\text{price}} \) including LO’s identity \( ID_{LO} \), LO’s account \( Account_{LO} \) and the winner prize \( \text{price} \) as:

\[
m_{\text{price}} = (ID_{LO} \parallel Account_{LO} \parallel \text{price})
\]  

(44)

2.8 uses Schnorr’s signature mechanism to sign the award-winning information \( m_{\text{price}} \) as:

\[
N_i = g_{CA}^{u_i} \mod p_{CA}, \text{ where } u_i \in z_{p_{CA}}
\]  

(45)

\[
\gamma_i = h(N_i, m_{\text{price}})
\]  

(46)

\[
s_i = (u_i - x_{LO} \cdot \gamma_i) \mod q_{CA}
\]  

(47)

defines the signature of award-winning information \( m_{\text{price}} \) as:

\[
\delta_{m_{\text{price}}} = (r_i, s_i)
\]  

(48)

2.9 encrypts the award-winning information \( m_{\text{price}} \) and the signature \( \delta_{m_{\text{price}}} \) with the session key \( K_{P_i-L_0} \) as:

\[
C_{P_i-L_0} = E(K_{P_i-L_0}, (m_{\text{price}} \parallel \delta_{m_{\text{price}}}))
\]  

(49)

2.10 uses Schnorr’s signature mechanism to sign the ciphertext \( C_{P_i-L_0} \) as:

\[
N_i = g_{CA}^{u_4} \mod , \text{ where } u_4 \in z_{q_{CA}}
\]  

(50)

\[
\gamma_i = h(N_i, C_{P_i-L_0})
\]  

(51)

\[
s_i = (u_4 - x_{LO} \cdot \gamma_i) \mod q_{CA}
\]  

(52)

2.11 defines the signature of ciphertext \( C_{P_i-L_0} \) as:

\[
\delta_i = (\gamma_i, s_i)
\]  

(53)

sends the message \((\delta_i, C_{P_i-L_0})\) to \( P_i \).

**Step 3.** \( P_i \) verifies LO’s signature, makes a signature for the award-information, and then sends it to B.

When receiving this message, \( P_i \)

1. verifies the signature \( \delta_i \) with LO’s public key \( y_{LO} \) to compute the parameter \( \gamma_i \) as:

\[
\gamma_i' = g_{CA}^{u_i} \cdot y_{LO}^{\gamma_i} \mod p_{CA}
\]  

(54)

2. checks to see whether the equation holds, as:

\[
\gamma_i = h(\gamma_i', C_{P_i-L_0})
\]  

(55)

3.3 If the equation holds, \( P_i \) decrypts the ciphertext \( C_{P_i-L_0} \) with session key \( K_{P_i-L_0} \) as:

\[
(m_{\text{price}} \parallel \delta_{m_{\text{price}}}) = D(K_{P_i-L_0}, (C_{P_i-L_0}))
\]  

(56)

3.4 defines, the signature \( \delta_{m_{\text{price}}} \), the award-information \( m_{\text{price}} \), \( P_i \)’s identity \( ID_{P_i} \) and \( P_i \)’s account \( Account_{P_i} \) as:

\[
m'_{\text{price}} = (\delta_{m_{\text{price}}} \parallel m_{\text{price}} \parallel ID_{P_i} \parallel Account_{P_i})
\]  

(57)

3.2 uses Schnorr’s signature mechanism to sign the award-winning information \( m_{\text{price}} \) as computing

\[
N_i = g_{CA}^{u_i} \mod , \text{ where } u_i \in z_{q_{CA}}
\]  

(58)

\[
\gamma_i' = h(N_i, m'_{\text{price}})
\]  

(59)

\[
s_i' = (u_i - x_{LO} \cdot \gamma_i') \mod q_{CA}
\]  

(60)

3.6 sends the defined signature \( \delta'_{m_{\text{price}}} = (\gamma_i', s_i') \) and the award-winning information \( m'_{\text{price}} \) to B.

**Step 4.** B checks the signature and transfers the amount of prize into the winner’s account.

To verify the signature \( \delta'_{m_{\text{price}}} \), B

1. uses \( P_i \)’s public key \( y_{P_i} \) to compute the parameter \( \gamma_i' \) as:

\[
\gamma_i' = g_{CA}^{u_i} \cdot y_{P_i}^{\gamma_i} \mod p_{CA}
\]  

(61)

2. checks to see whether the equation holds, as:

\[
\gamma_i = h(\gamma_i', m_{\text{price}})
\]  

(62)

3. verifies the signature \( \delta_{m_{\text{price}}} \) by using the following two equations:

\[
\gamma_i' = g_{CA}^{u_i} \cdot y_{LO}^{\gamma_i} \mod p_{CA}
\]  

(63)

\[
\gamma_i = h(\gamma_i', m_{\text{price}})
\]  

(64)

4.4 If the above two signatures are both correct, B transfers the amount of prize into the winner’s account \( Account_{P_i} \) from LO’s account \( Account_{LO} \).

For winning player who has an account with a bank different from the LO, inter-bank transfer will be utilized by the banks concerned. The claiming of prize phase is illustrated in Figure 5.
Figure 5. Claiming of prize phase

1.1 \( C_{P-LO} = E(K_{P-LO}, (C_{Lottery} \| \delta_{C_{Lottery}}) \| k_i) \);

1.2 \((C_{P-LO}, m_{claim})\);

2.1 \((C_{Lottery} \| \delta_{C_{Lottery}}) \| k_i) = D(K_{P-LO}, (C_{P-LO})\);

2.2 \( r'_j = g_{CA}^{s_j \cdot y_{LO}^{r_j}} \mod p_{CA} \);

2.3 \( r_j \equiv h(r'_j, C_{Lottery}) \);

2.4 \((Lottery, \| chain, \| i) = D(K_{P-LO}, (C_{Lottery})\);

2.5 \( \alpha'_i = h(WinNum \| k_j) \);

2.6 \( \alpha_j \equiv \alpha'_i \);

2.7 LO checks to see whether the pseudo name \( \alpha_i \) was published on the bulletin board;

2.8 \( m_{prize} = (ID_{LO} \| Account_{LO} \| prize) \);

2.9 \( N_i = g_{CA}^{u_3} \mod p_{CA} \), where \( u_3 \in z_{q_{CA}} \);

\( r_i = h(N_i, m_{prize}) \);

\( s_j = (u_3 - x_{LO} \cdot r_j) \mod q_{CA} \);

2.10 \( \delta_{m_{prize}} = (r_j, s_j) \);

2.11 \( C_{P-LO} = E(K_{P-LO}, (m_{prize} \| \delta_{m_{prize}})) \);

2.12 \( N_i = g_{CA}^{u_4} \mod p_{CA} \), where \( u_4 \in z_{q_{CA}} \);

\( r_i = h(N_i, C_{P-LO}) \);

\( s_i = (u_4 - x_{LO} \cdot r_j) \mod q_{CA} \);

2.13 \( \delta_i = (r_j, s_i) \);

2.14 Sends the message \((\delta_i, C_{P-LO})\) to \( P_i \);

3.1 \( r'_j = g_{CA}^{s_j \cdot y_{LO}^{r'_j}} \mod p_{CA} \);

3.2 \( r_j \equiv h(r'_j, C_{P-LO}) \);

3.3 \((m_{prize} \| \delta_{m_{prize}}) = D(K_{P-LO}, (C_{P-LO})\);

3.4 \( m_{prize}' = (\delta_{m_{prize}} \| m_{prize} \| ID_P \| Account_P) \);

3.5 \( N_i = g_{CA}^{u_3} \mod p_{CA} \), where \( u_5 \in z_{q_{CA}} \);

\( r'_i = h(N_i, m_{prize}') \);

\( s'_i = (u_5 - x_{LO} \cdot r_j) \mod q_{CA} \);

3.6 \( \delta_{m_{prize}}' = (r'_i, s'_i) \) and \( m_{prize}' \) are sent to \( B \);

4.1 \( r'_j = g_{CA}^{s_j \cdot y_{LO}^{r'_j}} \mod p_{CA} \);

4.2 \( r_j \equiv h(r'_j, m_{prize}') \);

4.3 \( r'_j = g_{CA}^{s_j \cdot y_{LO}^{r'_j}} \mod p_{CA} \);

4.4 If the above two signatures are both correct,

B transfers the amount of prize \( prize \)

into the winner's account from LO's account \( Account_{LO} \).
3.6 The Arbitration Phase

There are two possible dispute scenarios. First, a player may submit a winning claim more than once. When this happens, the LO can apply for arbitration with the published information shown on the bulletin board. Second, if the LO deems that a submitted lottery ticket for a claim is invalid, then it denies the winner’s claim. In this case, the winner can apply for arbitration with the arbiter.

Upon receiving an arbitration case, the arbiter will adjudicate the arbitration with the evidence provided, including the bulletin board information, the signature and the secret message. If the evidence fails the verification, then the complaint fails; otherwise, the arbitration succeeds. The flowchart is illustrated in Figure 6.
4 Security Analysis

Here, we analyze the proposed fair E-lottery system under the assumption that there exists an intruder Eve in the network system who is capable of eavesdropping on communication messages transmitted between LO and a player.

4.1 Public verification

**Security Issue 1.** Assume that a player questions the correctness of the winning numbers.

**Analysis.** This particular player can use the public verification function (see Eq. (35)) Figure 5. Claiming of prize phase

\[ \text{Verify}(y_{LO}, \text{seed}, \text{WinNum}, \pi) = \text{true} \text{ or false} \text{ to verify the correctness of the winning numbers.} \]

The algorithm of the verification function Verify \((y_{LO}, \text{seed}, \text{WinNum}, \pi)\) is computed as (see Eqs. (28)-(30)):

\[
\text{Verify}(y_{LO}, \text{seed}, \text{WinNum}, \pi) =\{
\begin{array}{l}
\text{false} \text{ if } \gamma = \gamma' \text{ and } \text{WinNum} = h(y_{LO}) \text{, return true;} \\
\text{else return false;} \\
\end{array}
\]

If the result is true, then the process of the winning number generated by LO is considered legitimate. Hence, the requirement of public verification is achieved.

**Security Issue 2.** Assume that a player suspects the correctness of the winning numbers (e.g. once the winning numbers have been chosen, the LO or a malicious insider purchases one or more lottery tickets before publishing the winning number).

**Analysis.** In the proposed scheme, when the player completes the purchasing behaviors, the bank will compute the signature of pseudo name \(\delta_{\alpha_i} \). Since the bank’s signature cannot be forged, and the signature will be published immediately after the player completes the purchase, our scheme can mitigate insider collusion.

4.2 Fairness

**Security Issue 3.** If a player wishes to predict or bias the winning result, s/he will fail.

**Analysis.** Since each winner’s hash chain value \(chain_i\) (for \(i = 1 \text{ to } f\)) is random and occasional, and the random seed \(seed\) of winner number generation (see Eq. (32)) is generated by random function \(Rand()\) as,

\[
\text{seed} = \text{rand}(\text{chain}_1, \text{chain}_2, ..., \text{chain})
\]

Hence, no one can predict the winning result.

4.3 Security

**Security Issue 4.** If Eve attempts to forge the winning lottery numbers to claim the prize, she will fail.

**Analysis.** In reviewing the lottery purchasing phase, LO uses DSA to sign the lotteries. For Eve to successfully forge a winning lottery, she must successfully forge or have access to LO’s private key \(x_{LO}\). The former will require Eve to solve the discrete logarithm problem in the underlying DSA.

**Security Issue 5.** If Eve attempts to forge a winning player, she will fail.

**Analysis.** In the claiming of prize phase, the lottery winner \(P\) must submit the winning lottery numbers \(\text{Lottery}_i\), where \(\text{Lottery}_i \equiv (\alpha_i \parallel \sigma_i)\), and random number \(k_i\) to prove his/her identity. If Eve uses the forged random number \(k'_i\) to claim the prize, LO can perceive the illegality via following equation:

\[
\alpha_i = h(\text{WinNum} \parallel k'_i),
\]

(see Eq. (42) and (43)). Now \(k'_i\) is replaced by \(k_i\).

Where \(\alpha_i\) is the pseudo name of \(P_i\). In fact, based on a one-way hash function, it is impossible to obtain \(k_i\) from \(\alpha_i\).

4.4 Correctness

**Security Issue 6.** A player suspects either of the following:

(1) The correctness of the value of final hash chain \(chain_f\).

(2) The correctness of random seed \(seed\).

**Analysis.** The suspecting player can use the published bulletin board information to verify the correctness of \(chain_i\) (for all \(is, \ i = 1 \text{ to } f\)) and \(seed\) as follows.

(1) The suspecting player can recalculate the hash chain value \(chain_i\) (for all \(is, \ i = 1 \text{ to } f\)) in sequence as (see Eq. (15)):

\[
\text{Initial condition } chain_0 = 0 \\
chain_i = h(chain_i \parallel \alpha_i) \quad \text{with } i = 1, 2, ..., f \quad \text{and seed as follows.}
\]

(2) The suspecting player then uses the pseudorandom number generator \(Rand()\) to derive the random seed \(seed^*\) as (see Eq. (32)):

\[
\text{seed} = \text{rand}(\text{chain}_1, \text{chain}_2, ..., \text{chain})
\]

If \(seed^*\) is equal to \(seed\), then the random seed of generation function is correct.

This is achieved in our proposed scheme, as shown.
4.5 Anonymity

Security Issue 7. If LO attempts to identify the player’s identity in the lottery purchasing and claiming of prize phases, then LO’s malicious behavior will fail.

Analysis. In the lottery purchasing phase, the player uses a random number \( k \) and the selected numbers \( \text{number} \), to compute \( \alpha_i \) where \( \alpha_i = h(\text{number} \mid k) \) (see Eq. (1)). In the claiming of prize phase, LO employs \( k \) and \( \text{number} \) to verify the legitimacy of the winner. The verification equation is as follows:

\[
\alpha_i' = h(\text{number} \parallel k)
\]

(see Eq. (42)) and then checks to see whether the equation holds, as (see Eq. (43)):

\[
\gamma = \alpha_i = \alpha_i'
\]

Notably, the player uses \( \alpha \) as his/her pseudo name; hence, anonymity is achieved in this issue.

Security Issue 8. If Eve steals the lottery by eavesdropping transmitted messages and attempts to identify the player’s identity from the lottery, Eve’s malicious behavior will fail.

Analysis. In our scheme, the player’s pseudo name \( \alpha_i \), where \( \alpha_i = h(\text{number} \mid k) \) (see Eq. (1)) and the arbitration evidence \( \alpha_i' \), where \( \alpha_i' = \gamma_{\text{LO u (M)}} \text{mod} \text{pCA} \) (see Eq. (2)), are combined to form Lottery. Thus, Eve attempts to obtain a player’s personal information from the lottery will fail since the lottery does not contain any personal information; hence, anonymity is achieved.

4.6 Convenience

In our scheme, the player can purchase the lottery via the Internet and receive prizes using the (inter)bank transfer system. Clearly, the proposed fair E-lottery mechanism can achieve this requirement.

4.7 Without Pre-registration

In our scheme, the players only need to register with the CA and bank in advance.

4.8 No Online Trusted Third Party (TTP)

An online TTP is not required in our E-lottery mechanism.

4.9 Arbitration Mechanism

The arbitration applicant needs to provide evidence to support the claim using the proposed arbitration mechanism. The applicant only needs to provide the relevant evidence for the arbiter to decide whether the complaint is valid. More specifically, the arbiter only need to verify whether the signature is valid using public-key verification or compare it with the data stored in database.

4.10 Insider Collusion Resilience

In this section, we assume that the LO is dishonesty and we now explain how our scheme can resist the various related attacks. We also provide a comparative summary of the security properties.

(1) Cheating attack

Security Issue 9. In order to share the winning prize, LO attempts to determine the player’s numbers after the winning numbers are published and publishes invalid pseudo name \( \alpha_i \) (see Table 4).

Analysis. In the lottery purchasing phase, LO publishes a pseudo name \( \alpha_i \) where \( \alpha_i = h(\text{number} \mid k_i) \) (see Eq. (11)) for \( i = i \) to \( f \), on the bulletin board (see Table 3). In other words, the pseudo name of the valid winning lottery \( \alpha_i \) must be published on the bulletin board before the winning numbers generation. Hence, LO’s malicious behavior will be detected (see Tables 3 and 4).

(2) Conspiracy attack

Security Issue 10. In order to have a share of the winnings, LO conspires with another player to modify the selected numbers into the winning numbers; LO’s malicious behavior will fail.

Analysis. In the winning number generation and verification phase, LO publishes the winner’s random number \( k \) and information of the winner’s lottery on the bulletin board, as illustrated in Table 4. Assume that the original number and the random number are \( \text{number} \) and \( k \), respectively; therefore, the pseudo name \( \alpha_i \) is computed as follows (see Eq. (11)).

\[
\alpha_i = h(\text{number} \mid k)
\]

Before the winning numbers are published, LO conspires with another player to modify his/her lottery numbers into the winning numbers \( \text{WinNum} \); hence, the pseudo name is calculated as follows (see Eq. (42)).

\[
\alpha_i' = h(\text{WinNum} \mid k)
\]

On the basis of one-way hash function, it is clear that \( \alpha_i \neq \alpha_i' \); i.e., it is impossible to find \( \text{WinNum} | k \) which satisfies \( h(\text{number} | k) = h(\text{WinNum} | k) \). Based on the above assumption, LO’s malicious behavior will fail.

(3) A comparative summary: Security properties

A comparative summary of security properties between our proposed scheme and those of [5], [6] and [7] is presented in Table 5. Our scheme offers a higher level of security, since we are able to mitigate three specific insider attacks.

(4) A comparative summary with an existing E-lottery website

Since existing commercial lottery websites, TheLottery [1] and LoveMyLotto [2] have the same modes of operation, we will compare our proposed scheme against TheLottery – see Table 4. Basically, TheLottery and the other two websites only support a lottery purchasing service. Therefore, the player, unlike our scheme, needs to register with the TP in advance.
Table 5. A comparative summary: Security properties

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Public verification</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Fairness</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Security</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Correctness</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Anonymity</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Convenience</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No pre-registration required</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No online trusted third party</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Arbitration mechanism</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Random generation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Against the conspiracy attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Against the cheating attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Against the LO purchasing lottery tickets to share winners’ prize</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Description of the cash flow</td>
<td>sketchy</td>
<td>sketchy</td>
<td>sketchy</td>
<td>detailed</td>
</tr>
</tbody>
</table>

Table 6. A comparative summary with an existing e-lottery website

<table>
<thead>
<tr>
<th></th>
<th>TheLottery.com [1]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support arbitration mechanism</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Player need not register with the TP</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Support digital signature to verify the legality of lottery</td>
<td>No*&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Yes*&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>Is there a mechanism to redeem when the lottery agent refuses to give out the prize?</td>
<td>No</td>
<td>Players can request arbitration</td>
</tr>
<tr>
<td>Non-repudiation of evidence</td>
<td>Depend on the scanned archives and credit card transaction receipts</td>
<td>Yes*&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

*<sup>a</sup> Players manually verify the legitimacy of a lottery. <sup>b</sup> Our scheme adopts DSA signature mechanism.

Table 7. A comparative summary: computation cost

<table>
<thead>
<tr>
<th>Phase</th>
<th>Lee et al. [6]</th>
<th>Chen et al. [7]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Player ELD</td>
<td>Player ELD</td>
<td>Player ELD</td>
</tr>
<tr>
<td>First Phase</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Second Phase</td>
<td>$1E + (t+1)I + 4H + 1S + 1A$</td>
<td>$1E + 3H + 1S + 1A$</td>
<td>–</td>
</tr>
<tr>
<td>Third Phase</td>
<td>–</td>
<td>$R$</td>
<td>–</td>
</tr>
<tr>
<td>Fourth Phase</td>
<td>$1S + 1A$</td>
<td>$1E + 2S + 1A$</td>
<td>–</td>
</tr>
<tr>
<td>Fifth Phase</td>
<td>–</td>
<td>–</td>
<td>$1H + 1E + 3I$</td>
</tr>
</tbody>
</table>

Note. ELD: the e-lottery dealer; I: multiplicative or multiplicative inverse operation; E: modular exponentiation; H: hash function R: computational costs of PRNG; S: symmetric encryption/decryption; A: asymmetric encryption/decryption.

(5) Computation cost

We show the computation cost with the other related works in Table 7.

In our previous work [7], it focused to deal with the lottery participants can join a participant group to purchase a lottery in a mobile environment. So, the coordinator can negotiate with the e-lottery dealer and the participants also can obtain the lottery prize via lottery prize claim phase. The [5-6] schemes are relative simple. Therefore, it spent more cost in computation. In addition, our scheme involves the bank role to issue the lottery prize; and design a processing flow of arbitration phase when the arbitration case occurs. The proposed protocol is comprehensive with practical considerations. Therefore, the time complexity of our scheme spent more computation overhead than [5-6].

5 Conclusions

In our human society, gambling practices such as
purchasing lotteries are unlikely to fade away in the foreseeable future since players have the opportunity (even though the probability is very low) to receive a significant cash prize by spending only a small amount of money. Currently, E-lottery has several mentioned problems needed to be conquered. In this paper, an E-lottery scheme that has a fair purchasing environment and an arbitration mechanism by using verifiable random function (VRF), digital signature algorithm (DSA) and bulletin board mechanism is proposed. It provides a secure approach for players to purchase E-lottery tickets and for winners to claim the corresponding prize. It also safe for Lo to check the tickets when receiving winners’ claims. Based on our analyses, this scheme actually fulfills those requirements typically expected by a secure and practical E-lottery system. This system also provides users with an arbitration mechanism, with which both operators and the players can effectively protect their interests and rights.

References


**Biographies**

**Chin-Ling Chen** received his Ph.D. from National Chung Hsing University, Taiwan in 2005. From 1979 to 2005, he was a senior engineer at Chunghwa Telecom Co., Ltd. He is a professor. His research interests include cryptography, network security and electronic commerce. He has published over 90 articles in SCI/SSCI international journals.

**Yuan-Hao Liao** was born in Taiwan in 1982. He received the B.Sc. degree in Information Management from HSIUPING Institute of Technology, Taichung, Taiwan. He received his M.Sc. degree at the Department of Computer Science and Information, Chaoyang University of Technology. His research interests include cryptography and electronic commerce.

**Fang-Yie Leu** received his bachelor, master and Ph.D. degrees all from National Taiwan University of Science and Technology, Taiwan, in 1983, 1986 and 1991, respectively. His research interests include wireless communication, network security, Grid applications and Sensor Network. He is currently a professor of Computer Science Department, Tunghai University, Taiwan.

** Ilsun You** received the second Ph.D. degree from Kyushu University, Japan, in 2012. He is an associate professor at the Department of Information Security Engineering, Soonchunhyang University. He is the EiC of Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA). He is a Fellow of the IET.

**Kim-Kwang Raymond Choo** received the Ph.D. in information security from the Queensland University of Technology, Brisbane, QLD, Australia, in 2006. He currently holds the Cloud Technology Endowed Professorship with The University of Texas at San Antonio, San Antonio, TX, USA. He is also a fellow of the Australian Computer Society.

**Chia-Yin Ko** received her Ed D in Education from Seattle Pacific University, Washington, USA. Currently, she is an assistant professor in the Computer Science Department of Tunghai University, Taiwan. Her research interests include machine learning, data mining in education, healthcare, wireless communication, wireless and sensor network.