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

Chin-Ling Chen[1,2,3], Yuan-Hao Liao[3], Fang-Yie Leu[4], Ilsun You[5], Kim-Kwang Raymond Choo[6], Chia-Yin Ko[4]

[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 E-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

[*] Corresponding Author: Ilsun You; E-mail: ilsunu@gmail.com

DOI: 10.3966/160792642020052103004

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' 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 compare 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 system. 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

Scheme Comparison item	RSA [22]	DSA [16]	ECDSA [23]
Signature length (≥ 112 bits)	$\|n\|$ = 2048 bits	2×$\|q\|$ = 448 bits	$\|r\| + \|s\|$ =224 +224=448 bits
Digital Signatures Security Strength Transitions by NIST [26]	$\|n\| \geq$ 2048 bits (≥ 112 bits)	$\|p\| \geq$ 2048 bits and $\|q\| \geq$ 224 bits (≥ 112 bits)	$\|n\| \geq$ 224 bits (≥ 112 bits)
Security Basis	Prime factorization problem	Discrete logarithm problem	Discrete logarithm problem

Note. n, p, q, r and *s* are parameters of security length of the related schemeThe 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 $F_{(\cdot)}(\cdot):\{0,1\}^k \rightarrow \{0,1\}^{l(k)}$ 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 π using *y=FPK(x)*.

(4) Verify (PK, x, y, π): a function that verifies *y=FPK(x)* using the proof π.

Details will be presented later.

VRF should satisfy the following properties:

(1) Uniqueness:

$$\begin{aligned} & Verify\,(PK, x, y_1\ \pi_1) \\ & = Verify\,(PK, x, y_2\ \pi_2) \end{aligned}$$

$if\, y1 = y2.$

(2) Computability: $Eval\,(SK, x) = F_{SK}(x)$ is efficiently computable.

(3) Provability: if $(y, \pi) = Prove\,(SK, x)$, then $y(PK, x, y, \pi) = 1$.

(4) Pseudo randomness: The probability that an attacker inputs data x of an arbitrary number in bits to $F_{PK}(x)$. 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*.

P: a prime modulus, $2L–1 < p < 2L$, where *L* is the length of *p* in bits.

Q: a prime divisor of $(p – 1)$, $2N–1 < q < 2N$, where *N* is the length of *q* in bits.

G: a generator of the subgroup of order *q mod p*, such that $1 < g < p$.

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].

Y: a public key, $y = g^x \bmod p$.

k: a secret number that is unique to each message; *k* is a randomly or pseudo randomly 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 $r = (g^k \bmod p) \bmod q$ and $s = k^{-1} (M+ xr) \bmod q$

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.

$w = s' - 1 \bmod q.$

$u1 = M'w \bmod q.$

$u2 = r'w \bmod q.$

$v = g^{u1}\, y^{u2} (\bmod p) \bmod q.$

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:

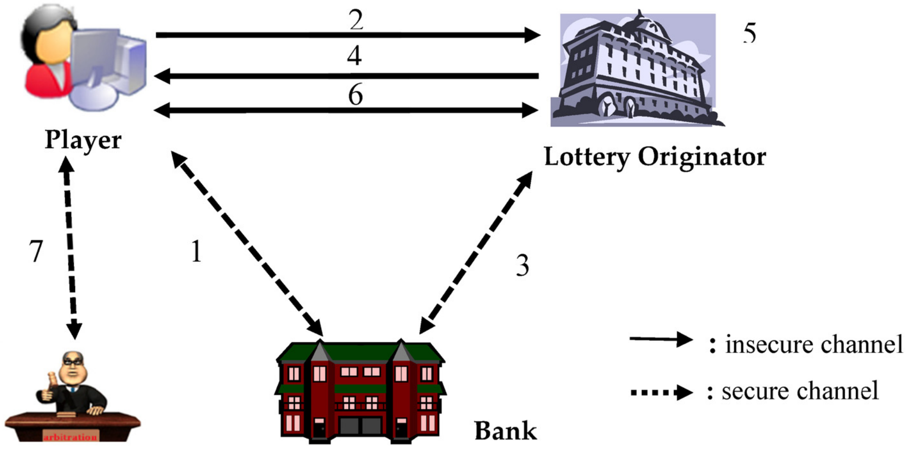


Figure 1. The structure of our scheme

(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←→B: Player (P) opens a new account with a participating bank, say Bank B.

Step 2. P→ LO: Player purchases an E-lottery with LO.

Step 3. LO←→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→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←→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←→A: The player submits the arbitration to the arbiter (A) if a dispute occurs.

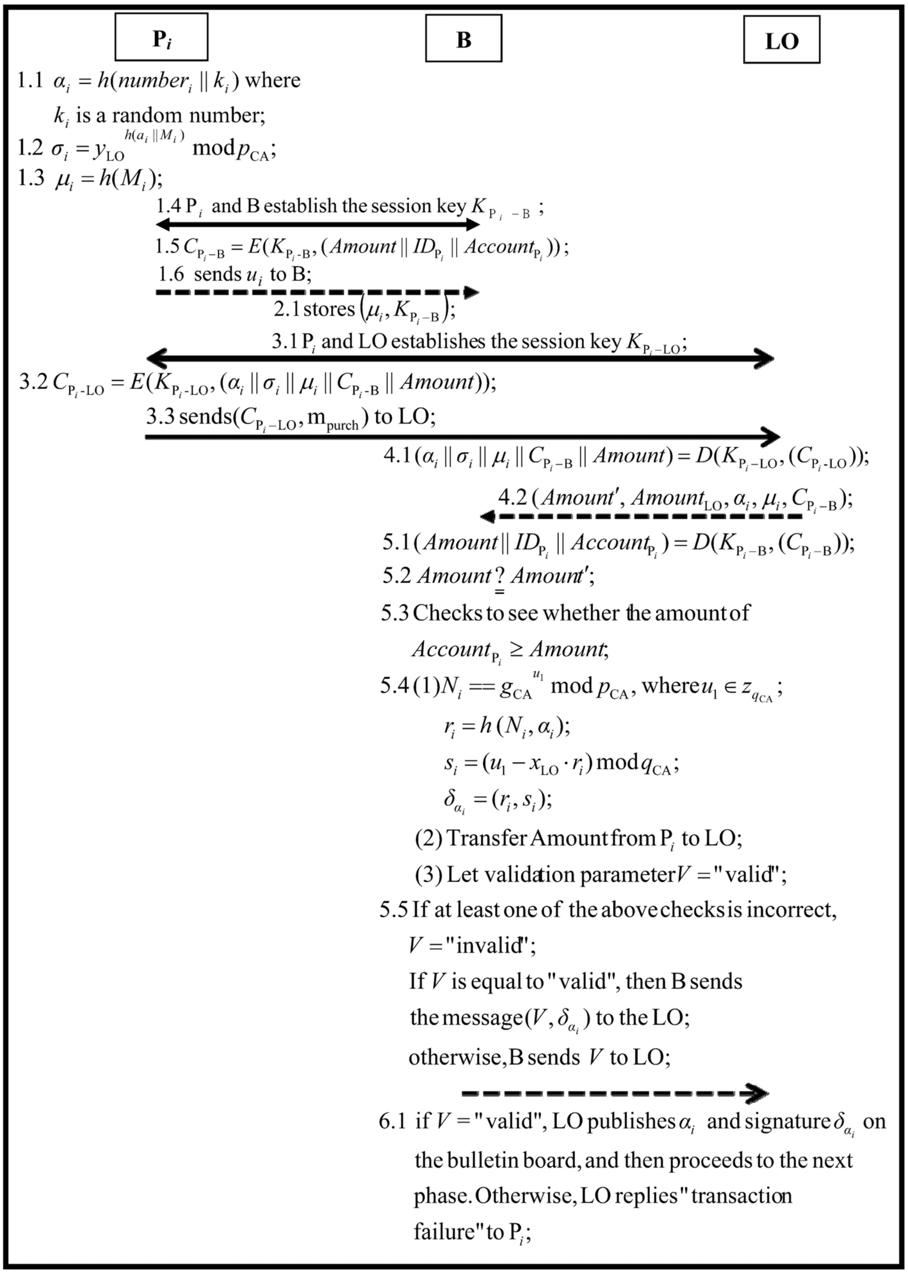
A summary of the notations employed is shown in Table 2.

Table 2. Summary of notations

Notation	Description
M_i	secret message which is the evidence selected by the ith player and can be used in an arbitration
k_i	random number selected by the ith player
$number_i$	set of lottery numbers selected by the ith player
α_i	ith player's pseudo name, where α_i=h($number_i$|| k_i)
μ_i	ith player's index on the database of the bank which is used to search for the corresponding session key K_{P_i-B}
X	role variable of participant of an E-lottery system; the value maybe player, LO, bank or arbiter
ID_X	X's identity
$Account_X$	X's bank account
Amount	amount of the lottery prize
p_x	prime number selected by X, $2^{L-1} < p_x < 2^L$, which L is the length of p_x in bits
q_x	prime number selected by X, where q_x is a divisor of $(p_x - 1)$, $2^{N-1} < q_x < 2^N$, and N is the length of q_x in bits
g_{CA}	generator of the subgroup of order q_x mod p_x, where g_{CA}, $1 < g_{CA} < p_x$, are selected by the certificate authority (CA)
(x_x, y_x)	(X's private key, X's public key) generated by the CA. x_x should remain secret, where x_x is a random or pseudo random integer, $0 < x_x < q_x$; the y_x is the corresponding public key, where $y_x = g_{CA}{}^{x_x} \bmod p_x$
σ_i	arbitration evidence generated by the ith player, where $\sigma_i = y_{LO}{}^{h(a_i\|M_i)} \bmod p_{CA}$
$E(K_{X-Y}, m))$	symmetric encryption function which encrypts message m with session key K_{X-Y}
$D(K_{X-Y}, m))$	symmetric decryption function which decrypts message m with the session key K_{X-Y}
$h(\cdot)$	one-way hash function

Table 2. Summary of notations (continue)

Notation	Description
f	number of lottery tickets having sold
chain_{f}	published hash chain set which is used to generate the winning number where a valid seed α_i is generated by the ith player, and $\text{chain}_0=0$ is the initial vector, $\text{chain}_1=h(\text{chain}_0, \alpha_1)$, $\text{chain}_2=h(\text{chain}_1, \alpha_2)$, ..., $\text{chain}_{\text{f}} =h(\text{chain}_{\text{f-1}}, \alpha_{\text{f}})$
Prize	winning prize
Rand(.)	public pseudo-random number generation function
Eval(.)	winning number generation function
Prove (.)	proof function
Verify(.)	public verification function
seed	random seed generated by the public pseudo-random number generating function as seed = Rand(chain_1, chain_2, ..., chain_{f})
||	concatenation operation
$A \overset{?}{=} B$	determine if A is equal to B
- - - -▶	secure channel
——▶	insecure channel



P$_i$ **B** **LO**

1.1 $\alpha_i = h(number_i \| k_i)$ where k_i is a random number;

1.2 $\sigma_i = y_{\text{LO}}^{h(a_i \| M_i)} \bmod p_{\text{CA}}$;

1.3 $\mu_i = h(M_i)$;

1.4 P_i and B establish the session key $K_{\text{P}_i-\text{B}}$;

1.5 $C_{\text{P}_i-\text{B}} = E(K_{\text{P}_i\text{-B}}, (Amount \| ID_{\text{P}_i} \| Account_{\text{P}_i}))$;

1.6 sends u_i to B;

2.1 stores $(\mu_i, K_{\text{P}_i-\text{B}})$;

3.1 P_i and LO establishes the session key $K_{\text{P}_i-\text{LO}}$;

3.2 $C_{\text{P}_i\text{-LO}} = E(K_{\text{P}_i\text{-LO}}, (\alpha_i \| \sigma_i \| \mu_i \| C_{\text{P}_i\text{-B}} \| Amount))$;

3.3 sends$(C_{\text{P}_i-\text{LO}}, \text{m}_{\text{purch}})$ to LO;

4.1 $(\alpha_i \| \sigma_i \| \mu_i \| C_{\text{P}_i-\text{B}} \| Amount) = D(K_{\text{P}_i-\text{LO}}, (C_{\text{P}_i\text{-LO}}))$;

4.2 $(Amount', Amount_{\text{LO}}, \alpha_i, \mu_i, C_{\text{P}_i-\text{B}})$;

5.1 $(Amount \| ID_{\text{P}_i} \| Account_{\text{P}_i}) = D(K_{\text{P}_i-\text{B}}, (C_{\text{P}_i-\text{B}}))$;

5.2 $Amount \overset{?}{=} Amount'$;

5.3 Checks to see whether the amount of $Account_{\text{P}_i} \geq Amount$;

5.4 (1) $N_i == g_{\text{CA}}^{u_1} \bmod p_{\text{CA}}$, where $u_1 \in z_{q_{\text{CA}}}$;

$r_i = h(N_i, \alpha_i)$;

$s_i = (u_1 - x_{\text{LO}} \cdot r_i) \bmod q_{\text{CA}}$;

$\delta_{\alpha_i} = (r_i, s_i)$;

(2) Transfer Amount from P_i to LO;

(3) Let validation parameter $V =$ "valid";

5.5 If at least one of the above checks is incorrect, $V =$ "invalid";

If V is equal to "valid", then B sends the message(V, δ_{α_i}) to the LO;

otherwise, B sends V to LO;

6.1 if $V =$ "valid", LO publishes α_i and signature δ_{α_i} on the bulletin board, and then proceeds to the next phase. Otherwise, LO replies "transaction failure" to P_i;

Figure 2. Summarizes the lottery purchasing phase

Step 1. The *i*th player Pi and bank B establish a session key $K_{Pi\text{-}B}$.

The *ith* player P_i

1.1 selects a favorite number numberr_i and a random number k_i, and then computes α_i as the pseudo name of the ith player with a one-way hash function where

$$\alpha_i = h(number\ \gamma_i \,\|\, k_i) \quad \textbf{(1)}$$

1.2 creates a secret message M_i, and uses LO's public key to compute α_i as:

$$\alpha_i = y_{LO}{}^{h(\alpha_i \| M_i)} \bmod p_{CA} \quad \textbf{(2)}$$

where α_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 μ_i as the i^{th} player's index in the session-key database of the bank used to record the corresponding session key K_{P_i-B} where

$$\mu_i = h(M_i) \quad \textbf{(3)}$$

establishes the session key K_{P_i-B} with B.

1.4 utilizes session key K_{P_i-B} to protect the account information, including the purchase amount of lottery (Amount), Pi's identity (ID_{P_i}) and Pi's account ($Acount_{P_i}$), as follows:

$$C_{P_i-B} = E(K_{P_i-B}, (Amount \,\|\, ID_{P_i} \,\|\, Amount_{P_i})) \quad \textbf{(4)}$$

1.5 sends μ_i to B.

Step 2. B stores the session key K_{P_i-B}, and μ_i together as a record in its session-key database.

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

P_i then

3.1 establishes the session key K_{P_i-LO} with LO.

3.2 uses K_{P_i-LO} to protect the purchase information, including the pseudo name α_i, arbitration evidence α_i, index μ_i, ciphertext C_{P_i-LO} and lottery amount Amount, as a ciphertext C_{P_i-LO} where

$$C_{P_i-LO} = E(K_{P_i-LO}, (\alpha_i \,\|\, \sigma_i \,\|\, \mu_i \,\|\, C_{P_i-LO} \,\|\, Amount)) \quad \textbf{(5)}$$

3.3 sends C_{P_i-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_{P_i-LO} with session key K_{P_i-LO} as:

$$\begin{aligned}&(\alpha_i \,\|\, \sigma_i \,\|\, \mu_i \,\|\, C_{P_i-LO} \,\|\, Amount)\\ &= D(K_{P_i-LO}, (C_{P_i-LO}))\end{aligned} \quad \textbf{(6)}$$

sends the transaction information, including the lottery amount $Amount'$ provided by LO, LO's account $Account_{LO}$, P_i's pseudo name α_i, index μ_i and ciphertext C_{P_i-B} to B via a secure channel.

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

Upon receiving the massage, B

5.1 looks up K_{P_i-B} in its session-key database according to index μ_i, and decrypts C_{P_i-B} as:

$$\begin{aligned}&(Amount \,\|\, ID_i \,\|\, Amount_{P_i})\\ &= D(K_{P_i-B}, (C_{P_i-B}))\end{aligned} \quad \textbf{(7)}$$

5.2 checks to see whether the lottery amount specified by LO is equal to the lottery amount *Amount* provided by P_i as:

$$Amount \overset{?}{=} Amount' \quad \textbf{(8)}$$

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 *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 α_i by computing

$$N_i = g_{CA}{}^{u_1} \bmod p_{CA}, \text{ where } u_1 \in z_{q_{CA}} \quad \textbf{(9)}$$

$$\gamma_i = h(N_i, \alpha_i) \quad \textbf{(10)}$$

$$s_i = (\gamma_i - x_{LO} \cdot \gamma_i) \bmod q_{CA} \quad \textbf{(11)}$$

$$\delta_{\alpha_i} = (\gamma_i, s_i) \quad \textbf{(12)}$$

(2) transferring the lottery amount *Amount* from Pi's account $Acount_{P_i}$ into LO's account $Acount_{LO}$.

(3) setting a validation parameter V = "*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. h

After the above procedures, the player completes the lottery purchasing phase.

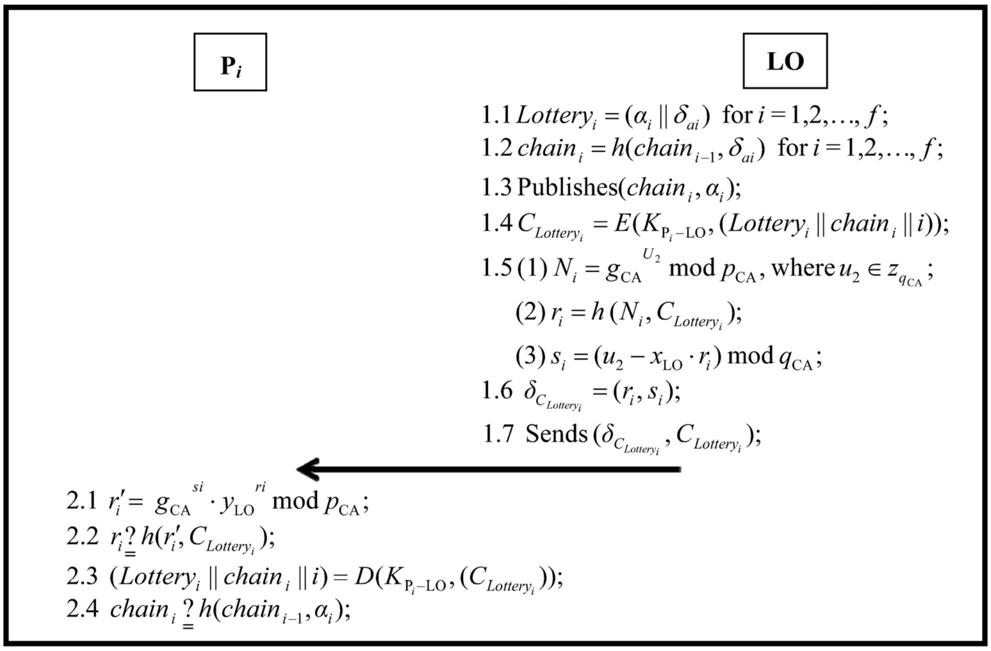
3.3 The Lottery Ticket Issuing Phase

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

Number of lottery tickets sold	Hash chain value	Player's pseudo name	Signature of player's pseudo name
1	$chain_1$	α_1	δ_{α_1}
2	$chain_2$	α_2	δ_{α_2}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	$chain_{\mathrm{i}}$	α_i	δ_{α_i}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
f	$chain_f$	α_f	δ_{α_f}

After a player, e.g., pf (i.e., $p_i=p_f$), has purchased a lottery ticket, LO will publish a record on the bulletin board containing the hash chain values (i.e. where $chain_0=0$, chain1=$h(chain_0, \alpha_1)$, chain$_2$=$h(chain_1, \alpha_2)$, ..., $chain_f = h(chain_{f-1}, \alpha_f)$), the f's pseudo name $\alpha_f = h(number_f \,||\, k_f)$ and the signature of f's pseudo name $\delta_{\alpha_f} = (r_f, s_f)$ where $1 \le f$, 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.



1.1 $Lottery_i = (\alpha_i \,||\, \delta_{ai})$ for $i = 1,2,\ldots, f$;
1.2 $chain_i = h(chain_{i-1}, \delta_{ai})$ for $i = 1,2,\ldots, f$;
1.3 Publishes$(chain_i, \alpha_i)$;
1.4 $C_{Lottery_i} = E(K_{\mathrm{P}_i-\mathrm{LO}}, (Lottery_i \,||\, chain_i \,||\, i))$;
1.5 (1) $N_i = g_{\mathrm{CA}}^{U_2} \bmod p_{\mathrm{CA}}$, where $u_2 \in z_{q_{\mathrm{CA}}}$;
(2) $r_i = h(N_i, C_{Lottery_i})$;
(3) $s_i = (u_2 - x_{\mathrm{LO}} \cdot r_i) \bmod q_{\mathrm{CA}}$;
1.6 $\delta_{C_{Lottery_i}} = (r_i, s_i)$;
1.7 Sends $(\delta_{C_{Lottery_i}}, C_{Lottery_i})$;

2.1 $r_i' = g_{\mathrm{CA}}^{si} \cdot y_{\mathrm{LO}}^{ri} \bmod p_{\mathrm{CA}}$;
2.2 $r_i \overset{?}{=} h(r_i', C_{Lottery_i})$;
2.3 $(Lottery_i \,||\, chain_i \,||\, i) = D(K_{\mathrm{P}_i-\mathrm{LO}}, (C_{Lottery_i}))$;
2.4 $chain_i \overset{?}{=} h(chain_{i-1}, \alpha_i)$;

Figure 3. Lottery ticket issuing phase

Step 1. LO makes a digital signature of lottery.

LO first

1.1 defines the lottery $Lotter_{yi}$ as:

$$Lottery_i = (\sigma_i \,||\, \delta_{\sigma_i}) \text{ for } i=1, 2, \ldots f, \quad \textbf{(13)}$$

subsequently links Pi's pseudo name α_i on the hash chain as:

$$chai\ n_i = h(chai\ n_{i-1}, \delta_{\sigma_i}) \text{ for } i=1, 2, \ldots f, \quad \textbf{(14)}$$

1.2 publishes $(chai\ n_{i-1}, \sigma_i)$ on the bulletin board.

1.3 uses the session key K_{P_i-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:

$$C_{Lottery_i} = E(K_{P_i-LO}, (Lottery_i) \,||\, chain_i \,||\, i) \quad \textbf{(15)}$$

1.4 uses Schnorr's signature mechanism to sign the ciphertext $C_{Lottery_i}$ as computing

$$(1)\ N_i = g_{CA}^{U_2} \bmod p_{CA}, \text{ where } u_2 \in z_{q_{CA}} \quad \textbf{(16)}$$

$$(2)\ \gamma_i = h(N_i, C_{Lottery_i}). \quad \textbf{(17)}$$

$$(3)\ s_i = (u_2 - x_{LO} \cdot \gamma_i) \bmod q_{CA} \cdot . \quad \textbf{(18)}$$

1.5 defines the signature of ciphertext $C_{Lottery_i}$ as:

$$\delta_{C_{Lottery_i}} = (\gamma_i, s_i) \quad \textbf{(19)}$$

1.6 sends the message $(\delta_{C_{Lottery_i}}, C_{Lottery_i})$ 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

2.1 uses the LO's public key y_{LO} to compute:

$$\gamma_i' = g_{CA}{}^{si} \cdot y_{LO}{}^{\gamma i} \bmod p_{CA} \qquad \textbf{(20)}$$

checks to see whether

$$\gamma_i \overset{?}{=} h(\gamma_i', C_{Lottery_i}) \qquad \textbf{(21)}$$

2.2 utilizes the session key K_{P_i-LO} to decrypt the ciphertext $C_{Lottery_i}$ as:

$$(Lottery_i \,\|\, chain_i \,\|\, i) = D(K_{P_i-LO}, (C_{Lottery_i})) \qquad \textbf{(22)}$$

2.3 integrates the pseudo name α_i into the hash chain $chai\ n_{i-1}$, and checks to see whether the hash chain published on the bulletin board is valid, as:

$$chain_i \overset{?}{=} h(chain_{i-1}), \alpha_i) \qquad \textbf{(23)}$$

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, $y = Eval\,(SK, x)$, $\pi = Prove\,(SK, x)$ and $y = WinNum$ maybe true or false. The output is $y = WinNum$, and the input parameters are $SK = x_{LO}$, $x = seed$ and $PK = y_{LO}$.

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

{

The public key $y_{\mathrm{LO}} = g_{\mathrm{CA}}{}^{X_{\mathrm{LO}}} \bmod p_{\mathrm{CA}}$; **(24)**

$$r = g_{\mathrm{CA}}{}^{seed} \bmod p_{\mathrm{CA}}; \qquad \textbf{(25)}$$

$$WinNum = h(r, y_{\mathrm{LO}}); \qquad \textbf{(26)}$$

}

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

$$\pi = (seed - (WinNum \cdot x_{\mathrm{LO}})) \bmod q_{\mathrm{CA}} \qquad \textbf{(27)}$$

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

{

$$r = g_{\mathrm{CA}}{}^{seed} \bmod p_{\mathrm{CA}}; \qquad \textbf{(28)}$$

$$g_{\mathrm{CA}}{}^{\pi} \cdot y_{\mathrm{LO}}{}^{WinNum} \bmod p_{\mathrm{CA}} = g_{\mathrm{CA}}{}^{seed-(x_{\mathrm{LO}} \cdot WinNum)} \cdot g_{\mathrm{CA}}{}^{(x_{\mathrm{LO}} \cdot WinNum)} \bmod p_{\mathrm{CA}} \qquad \textbf{(29)}$$

/*Refer to Eqs. (27) and (24)*/

$$= g_{\mathrm{CA}}{}^{seed-(x_{\mathrm{LO}} \cdot WinNum)+(x_{\mathrm{LO}} \cdot WinNum)} \bmod p_{\mathrm{CA}}$$

$$= g_{\mathrm{CA}}{}^{seed} \bmod p_{\mathrm{CA}}$$

$$= r'; \qquad \textbf{(30)}$$

IF $\gamma = \gamma'$ and $WinNum = h(\gamma', y_{LO})$, returen true; **(31)**

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

1.1 uses the pseudorandom generator Rand( ) to derive random seed *seed* from the hash chain value $chai\ n_i$, for $i = 1$ to f as

$$seed = Rand(chain_1, chain_2, ..., chain_f) \qquad \textbf{(32)}$$

where $chai\ n_i = (chai\ n_i, \alpha_i)$ and f is the total number of lottery tickets sold.

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

$$WinNum = Eval(x_{LO}, seed); \qquad \textbf{(33)}$$

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

$$\pi = Prove(x_{LO}, seed) \qquad \textbf{(34)}$$

1.4 publishes the verification parameters $(seed, WinNum, \pi)$ 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 WinNum via the public verification function $Verify(\cdot,\cdot,\cdot,\cdot)$ to see whether

$$Verify(y_{LO}, WinNum, seed, \pi) \overset{?}{=} true; \qquad \textbf{(35)}$$

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.

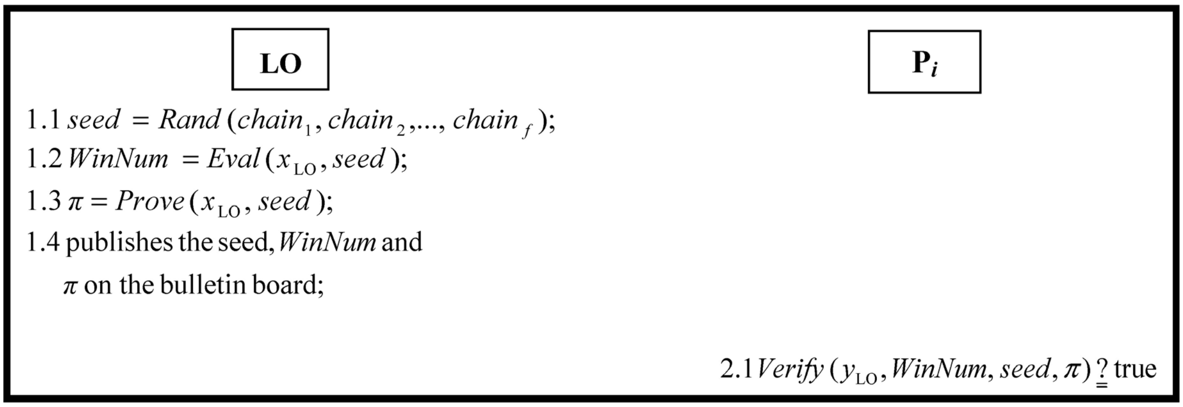
Step 1. The winner encrypts the information of the claim prize and sends the protected information to the LO.

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 $\delta_{Lottery_i}$ and the random number k_i as:

$$C_{P_i-LO} = E(K_{P_i-LO}, (C_{Lottery_i} \| \delta_{C_{Lottery_i}} \| k_i)) \qquad \textbf{(36)}$$

1.2 sends (C_{P_i-LO}, m_{ciaim}) to LO.

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



1.1 $seed = Rand(chain_1, chain_2, ..., chain_f)$;

1.2 $WinNum = Eval(x_{\text{LO}}, seed)$;

1.3 $\pi = Prove(x_{\text{LO}}, seed)$;

1.4 publishes the seed, $WinNum$ and π on the bulletin board;

2.1 $Verify(y_{\text{LO}}, WinNum, seed, \pi) \overset{?}{=}$ true

Figure 4. Winning numbers generation and verification phase

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 $\delta_{Lottery_i}$ and the random number *ki* as:

$$C_{P_i-LO} = E(K_{P_i-LO}, (C_{Lottery_i} \| \delta_{C_{Lottery_i}} \| k_i)) \qquad \textbf{(37)}$$

1.4 sends (C_{P_i-LO}, m_{ciaim}) 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_{P_i-LO} as:

$$(C_{Lottery_i} \| \delta_{C_{Lottery_i}} \| k_i) = D(K_{P_i-LO}, (C_{P_i-LO})) \qquad \textbf{(38)}$$

2.2 verifies the signature $\delta_{Lottery_i}$ by using its public key y_{LO} to compute the parameter γ_i' as:

$$\gamma_i' = g_{CA}{}^{si} \cdot y_{LO}{}^{\gamma_i} \bmod p_{CA} \qquad \textbf{(39)}$$

checks to see whether the following equation holds as:

$$\gamma_i \overset{?}{=} h(\gamma_i', C_{Lottery_i}) \qquad \textbf{(40)}$$

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

$$(\| chain_i \| i)) = D(K_{P_i-LO}, (C_{Lottery_i})) \qquad \textbf{(41)}$$

2.4 uses the random number k_i and the winner numbers WinNum to compute the parameter α_i' as:

$$\alpha_i' = h(WinNum \| k_i) \qquad \textbf{(42)}$$

2.5 checks to see whether the equation holds as:

$$\alpha_i \overset{?}{=} \alpha' \qquad \textbf{(43)}$$

2.6 checks to see whether the pseudo name α_i was published on the bulletin board.

Table 4. Information of lottery winners published on the bulletin board

Number of the winning lottery	Hash chain value of the winner	Winner's pseudo name	Signature of the winner's pseudo name	Winner's random number
3	$chain_3$	α_3	δ_{α_3}	k_3
5	$chain_5$	α_5	δ_{α_5}	k_5
9	$chain_9$	α_9	δ_{α_9}	k_9

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_{prize} including LO's identity ID_{LO}, LO's account $Account_{LO}$ and the winner prize *prize* as:

$$m_{prize} = (ID_{LO} \,||\, Account_{LO} \,||\, prize) \quad \textbf{(44)}$$

2.8 uses Schnorr's signature mechanism to sign the award-winning information m_{prize} as:

$$N_i = g_{CA}{}^{u_3} \bmod p_{CA}, \text{ where } u_3 \in z_{P_{CA}} \quad \textbf{(45)}$$

$$\gamma_i = h(N_i, m_{prize}) \quad \textbf{(46)}$$

$$s_i = (u_3 - x_{LO} \cdot \gamma_i) \bmod q_{CA} \quad \textbf{(47)}$$

defines the signature of award-winning information m_{prize} as:

$$\delta_{m_{prize}} = (r_i, s_i) \quad \textbf{(48)}$$

2.9 encrypts the award-winning information m_{prize} and the signature $\delta_{m_{prize}}$ with the session key K_{P_i-LO} as:

$$C_{P_i-LO} = E(K_{P_i-LO}, (m_{prize} \,||\, \delta_{m_{prize}})) \quad \textbf{(49)}$$

2.10 uses Schnorr's signature mechanism to sign the ciphertext C_{P_i-LO} as:

$$N_i = g_{ca}{}^{u_4} \text{ mod, where } u_4 \in z_{q_{CA}} \quad \textbf{(50)}$$

$$\gamma_i = h(N_i, C_{P_i-LO}) \quad \textbf{(51)}$$

$$s_i = (u_4 - x_{LO} \cdot \gamma_i) \bmod q_{CA} \quad \textbf{(52)}$$

2.11 defines the signature of ciphertext C_{P_i-LO} as:

$$\delta_i = (\gamma_i, s_i) \quad \textbf{(53)}$$

sends the message (δ_i, C_{P_i-LO}) 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

3.1 verifies the signature δ_i with LO's public key y_{LO} to compute the parameter γ_i' as:

$$\gamma_i' = g_{CA}{}^{si} \cdot y_{LO}{}^{\gamma_i} \bmod p_{CA} \quad \textbf{(54)}$$

3.2 checks to see whether the equation holds, as:

$$\gamma_i \overset{?}{=} h(\gamma_i', C_{P_i-LO}) \quad \textbf{(55)}$$

3.3 If the equation holds, Pi decrypts the ciphertext C_{P_i-LO} with session key K_{P_i-LO} as:

$$(m_{prize} \,||\, \delta_{mprize}) = D(K_{P_i-LO}, (C_{P_i-LO})) \quad \textbf{(56)}$$

3.4 defines, the signature $\delta_{m_{prize}}$, the award-information m_{prize}, P_i's identity ID_{P_i} and P_i's account $Acocunt_{P_i}$ as:

$$m_{prize}' = (\delta_{mprize} \,||\, m_{mprize} \,||\, ID_{P_i} \,||\, Account_{P_i}) \quad \textbf{(57)}$$

3.2 uses Schnorr's signature mechanism to sign the award-winning information m_{prize}' as computing

$$N_i = g_{CA}{}^{u_5} \text{ mod}, \text{ where } u_5 \in z_{q_{CA}} \quad \textbf{(58)}$$

$$\gamma_i' = h(N_i, m_{prize}') \quad \textbf{(59)}$$

$$s_i' = (u_5 - x_{LO} \cdot \gamma_i) \bmod q_{CA} \quad \textbf{(60)}$$

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

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

To verify the signature $\delta'_{m_{prize}'}$, B

4.1 uses P_i's public key y_{P_i} to compute the parameter γ_i' as:

$$\gamma_i' = g_{CA}{}^{si} \cdot y_{P_i}{}^{\gamma_i} \bmod p_{CA} \quad \textbf{(61)}$$

4.2 checks to see whether the equation holds, as:

$$\gamma_i \overset{?}{=} h(\gamma_i', m_{prize}) \quad \textbf{(62)}$$

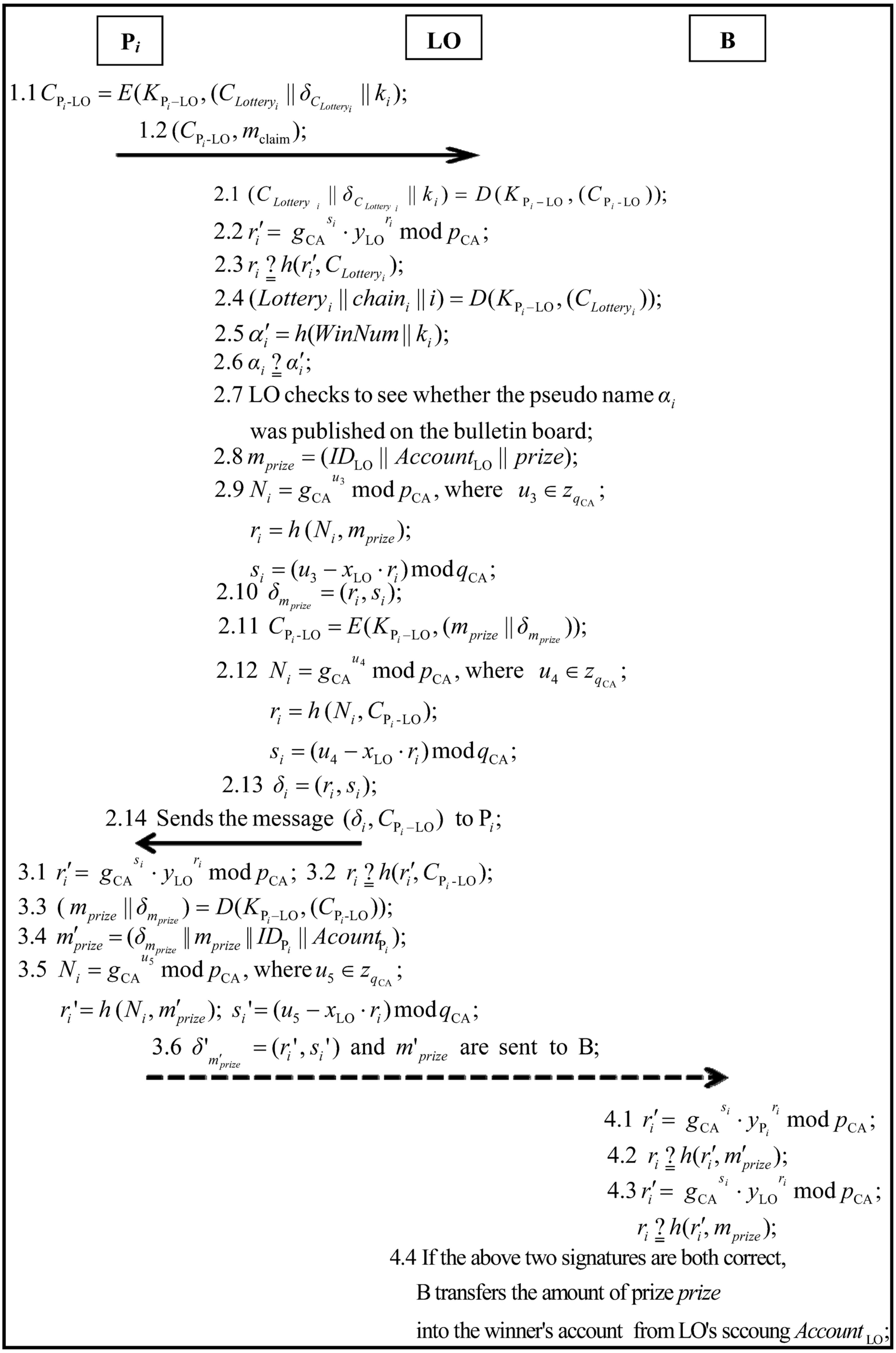
4.3 verifies the signature $\delta_{m_{prize}}$ by using the following two equations:

$$\gamma_i' = g_{CA}{}^{si} \cdot y_{LO}{}^{\gamma_i} \bmod p_{CA} \quad \textbf{(63)}$$

$$\gamma_i \overset{?}{=} h(\gamma_i', m_{prize}) \quad \textbf{(64)}$$

4.4 If the above two signatures are both correct, B transfers the amount of prize into the winner's account $Acocunt_{P_i}$ from LO's account $Acocunt_{LO}$

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.



1.1 $C_{P_i\text{-LO}} = E(K_{P_i\text{-LO}}, (C_{Lottery_i} \| \delta_{C_{Lottery_i}} \| k_i);$

1.2 $(C_{P_i\text{-LO}}, m_{\text{claim}});$

2.1 $(C_{Lottery_i} \| \delta_{C_{Lottery_i}} \| k_i) = D(K_{P_i\text{-LO}}, (C_{P_i\text{-LO}}));$

2.2 $r_i' = g_{CA}^{s_i} \cdot y_{LO}^{r_i} \bmod p_{CA};$

2.3 $r_i \overset{?}{=} h(r_i', C_{Lottery_i});$

2.4 $(Lottery_i \| chain_i \| i) = D(K_{P_i\text{-LO}}, (C_{Lottery_i}));$

2.5 $\alpha_i' = h(WinNum \| k_i);$

2.6 $\alpha_i \overset{?}{=} \alpha_i';$

2.7 LO checks to see whether the pseudo name α_i was published on the bulletin board;

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

2.9 $N_i = g_{CA}^{u_3} \bmod p_{CA}$, where $u_3 \in z_{q_{CA}};$

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

$s_i = (u_3 - x_{LO} \cdot r_i) \bmod q_{CA};$

2.10 $\delta_{m_{prize}} = (r_i, s_i);$

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

2.12 $N_i = g_{CA}^{u_4} \bmod p_{CA}$, where $u_4 \in z_{q_{CA}};$

$r_i = h(N_i, C_{P_i\text{-LO}});$

$s_i = (u_4 - x_{LO} \cdot r_i) \bmod q_{CA};$

2.13 $\delta_i = (r_i, s_i);$

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

3.1 $r_i' = g_{CA}^{s_i} \cdot y_{LO}^{r_i} \bmod p_{CA};$ 3.2 $r_i \overset{?}{=} h(r_i', C_{P_i\text{-LO}});$

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

3.4 $m'_{prize} = (\delta_{m_{prize}} \| m_{prize} \| ID_{P_i} \| Acount_{P_i});$

3.5 $N_i = g_{CA}^{u_5} \bmod p_{CA}$, where $u_5 \in z_{q_{CA}};$

$r_i' = h(N_i, m'_{prize});$ $s_i' = (u_5 - x_{LO} \cdot r_i) \bmod q_{CA};$

3.6 $\delta'_{m'_{prize}} = (r_i', s_i')$ and m'_{prize} are sent to B;

4.1 $r_i' = g_{CA}^{s_i} \cdot y_{P_i}^{r_i} \bmod p_{CA};$

4.2 $r_i \overset{?}{=} h(r_i', m'_{prize});$

4.3 $r_i' = g_{CA}^{s_i} \cdot y_{LO}^{r_i} \bmod p_{CA};$

$r_i \overset{?}{=} h(r_i', m_{prize});$

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 sccoung $Account_{LO}$;

Figure 5. Claiming of prize phase

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.

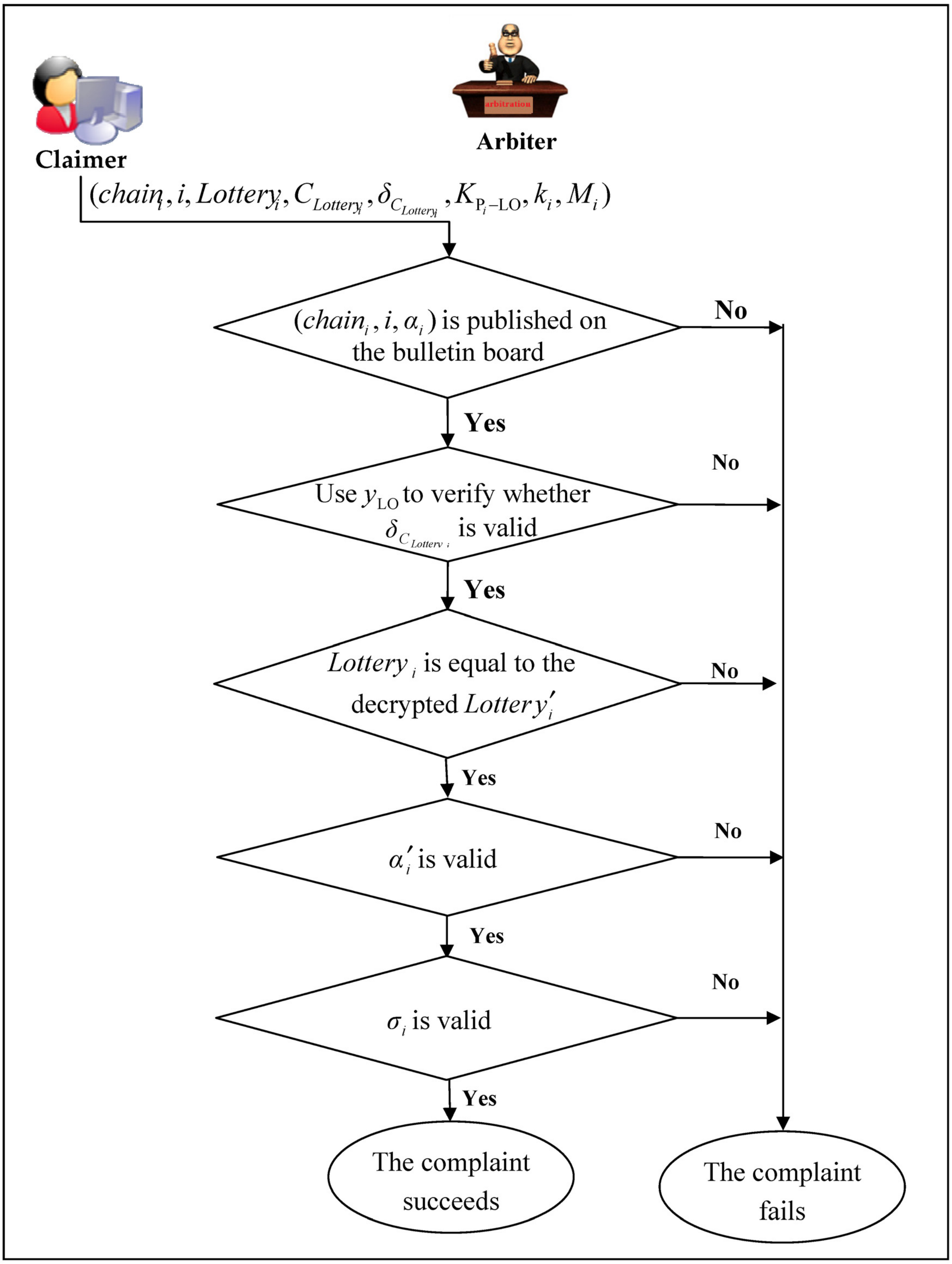

Arbiter
arbitration
Claimer
$(chain_i, i, Lottery_i, C_{Lottery_i}, \delta_{C_{Lottery_i}}, K_{P_i-LO}, k_i, M_i)$
$(chain_i, i, \alpha_i)$ is published on the bulletin board
No
Yes
Use y_{LO} to verify whether $\delta_{C_{Lottery_i}}$ is valid
No
Yes
$Lottery_i$ is equal to the decrypted $Lottery_i'$
No
Yes
α_i' is valid
No
Yes
σ_i is valid
No
Yes
The complaint succeeds
The complaint fails

Figure 6. Arbitration phase

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

Verify (y_{LO}, *seed*, *WinNum*, π) = true or false to verify the correctness of the winning numbers.

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

$Verify\ (y_{LO}, seed, WinNum, \pi)$

{

$r = g_{CA}{}^{seed} \bmod p_{CA};$

$g_{CA}{}^{\pi} \cdot y_{LO}{}^{WinNum} \bmod p_{CA}$

$= g_{CA}{}^{seed-(x_{LO}\cdot WinNum)} \cdot g_{CA}{}^{(x_{LO}\cdot WinNum)} \bmod p_{CA}$

$= g_{CA}{}^{seed-(x_{LO}\cdot WinNum)+(x_{LO}\cdot WinNum)} \bmod p_{CA}$

$= g_{CA}{}^{seed} \bmod p_{CA} = \gamma';$

If $\gamma = \gamma'$ and $WinNum = h(\gamma', y_{LO})$, return true;

else return false;}

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 lotteries (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 δ_{α_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 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,

$$seed = rand\ (chain_1, chain_2, ..., chain_f)$$

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$_i$ must submit the winning lottery numbers *Lottery*$_i$, where *Lottery*$_i$ =($\alpha_i \parallel \sigma_i$), 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:

$$\alpha_i \overset{?}{=} h(WinNum \parallel k_i'),$$

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

Where α_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 α_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 *i*s, *i* =1 to *f*), and *seed* as follows.

(1) The suspecting player can recalculate the hash chain value *chain*$_i$ (for all *i*s, *i* = 1 to *f*) in sequence as (see Eq. (15)):

$$\begin{aligned} &Initial\ condition\ chain_0 = 0 \\ &chain_1 = h(chain_0, \alpha_1) \\ &chain_2 = h(chain_1, \alpha_2) \\ &\qquad\vdots \\ &chain_f = h(chain_{f\text{-}1}, \alpha_f) \end{aligned}$$

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

$$seed = rand\ (chain_1, chain_2, ..., chain_f)$$

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_i and the selected numbers $number_i$ to compute $\alpha_{i,}$ where $\alpha_i = h(number_i \,||\, k_i)$ (see Eq. (1)). In the claiming of prize phase, LO employs k_i and $number_i$ to verify the legitimacy of the winner. The verification equation is as follows.

$$\alpha'_i = h(number_i \,||\, k_i)$$

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

$$\alpha_i \overset{?}{=} \alpha'_i$$

Notably, the player uses α_i 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 pseudo name $\alpha_{i,}$ where $\alpha_i = h(number_i \,||\, k_i)$ (see Eq. (1)) and the arbitration evidence α_i, where $\alpha_i = y_{LO}{}^{h(a_i||M_i)} mod p_{CA}$ see Eq. (2)), are combined to form $Lottery_i$. 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 α'_i (see Table 4).*

Analysis. In the lottery purchasing phase, LO publishes a pseudo name α_i, where $\alpha_i = h(number_i \,||\, k_i)$ (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 α_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_i 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 $number_i$ and k_i, respectively; therefore, the pseudo name α_i is computed as follows (see Eq.(11)).

$$\alpha_i = h(number_i \,||\, k_i)$$

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

$$\alpha'_i = h(WinNum \,||\, k_i)$$

On the basis of one-way hash function, it is clear that $\alpha_i \neq \alpha'_i$; i.e., it is impossible to find $(WinNum||k_i)$ which satisfies $h(number_i||k_i) = h(WinNum||k_i)$. 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, Lee *et al.* [5], Lee et al. [6] and Chen et al. [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 6. 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

	Lee *et al.* [5]	Lee *et al.* [6]	Chen *et al.* [7]	Ours
Public verification	Yes	Yes	Yes	Yes
Fairness	Yes	Yes	Yes	Yes
Security	Yes	Yes	Yes	Yes
Correctness	Yes	Yes	Yes	Yes
Anonymity	Yes	Yes	Yes	Yes
Convenience	Yes	Yes	Yes	Yes
No pre-registration required	Yes	Yes	Yes	Yes
No online trusted third party	Yes	Yes	Yes	Yes
Arbitration mechanism	No	No	No	Yes
Random generation	Yes	Yes	Yes	Yes
Against the conspiracy attack	No	No	No	Yes
Against the cheating attack	No	No	No	Yes
Against the LO purchasing lottery tickets to share winners' prize	No	Yes	No	Yes
Description of the cash flow	sketchy	sketchy	sketchy	detailed

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

	TheLottery.com [1]	Ours
Support arbitration mechanism	No	Yes
Player need not register with the TP	No	Yes
Support digital signature to verify the legality of lottery	No*^{a}	Yes^{*b}
Is there a mechanism to redeem when the lottery agent refuses to give out the prize?	No	Players can request arbitration
Non-repudiation of evidence	Depend on the scanned archives and credit card transaction receipts	Yes ^{*b}

Note. [a] Players manually verify the legitimacy of a lottery.
[b] Our scheme adopts DSA signature mechanism.

Table 7. A comparative summary: computation cost

Phase	Lee *et al.* [6]		Chen *et al.* [7]		Ours	
	Player	ELD	Player	ELD	Player	ELD
First Phase	–	–	$3E+1H+6I$	–	$2H+1E+1S$	$1H+1S+E+1I$
Second Phase	$1E + (t1)\cdot I+ 4H +1S +1A$	$1E + 3\cdot H+ 1S +1A$	–	$1I+1E$	$2H+1E+1S$	$(t+1)H+1 +1E+1I$
Third Phase	–	R	–	$(2t+1)H (t+4)I+(2t +3)E$	–	R
Fourth Phase	$1S + 1A$	$1E+2S +1A$	–	$(t+2)H+(t+1)I =(t+1)E$	$2S+3E+2H$	$4H+2S+5E$
Fifth Phase	–	–	$1H+1E+3I$	$2H+1 H+4E$	–	–

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 participators 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

[1] TheLottery.com, http://www.thelotter.com/.

[2] LoveMyLotto.com, https://www.lovemylotto.com/.

[3] S. S. M. Chow, L. C. K. Hui, S. M. Yiu, K. P. Chow, Practical Electronic Lotteries with Offline TTP, *Computer Communications*, Vol. 29, No. 15, pp. 2830-2840, September, 2006.

[4] J. S. Lee, C. C. Chang, Design of Electronic t-out-of-n Lotteries on the Internet, *Computer Standards & Interfaces*, Vol. 31, No. 2, pp. 395-400, February, 2009.

[5] J. S. Lee, C. S. Chan, C. C. Chang, Non-iterative Privacy Preservation for Online Lotteries, *IET Information Security*, Vol. 3, No. 4, pp. 139-147, December, 2009.

[6] J. S. Lee, W. C. Kao, B. Li, Aryabhata Remainder Theorem-based Non-iterative Electronic Lottery Mechanism with Robustness, *IET Information Security*, Vol. 7, No. 3, pp. 172-180, September, 2013.

[7] C. L. Chen, M. L. Chiang, W. C. Lin, D. K. Li, A Novel Lottery Protocol for Mobile Environments, *Computers & Electrical Engineering*, Vol. 49, 146-160, January, 2016.

[8] D. Y. Liao, X. H. Wang, Design of a Blockchain-based Lottery System for Smart Cities Applications, *Proceeding of 2017 IEEE 3rd International Conference on Collaboration and Internet Computing*, San Jose, CA, USA, 2017, pp. 275-282.

[9] P. Saichua, S. Khunthi, T. Chomsiri, Design of Blockchain Lottery for Thai Government, *2019 IEEE Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*, Nan, Thailand, 2019, pp. 9-12.

[10] K. L. Tsai, Y. L. Huang, F. Y. Leu, I. You, TTP Based High-efficient Multi-Key Exchange Protocol, *IEEE Access*, Vol. 4, pp. 6261-6271, September, 2016.

[11] C. L. Chen, J. J. Liao, A Fair online Payment System for Digital Content via Subliminal Channel, *Electronic Commerce Research and Applications*, Vol. 10, No. 3, pp. 279-287, May-June, 2011.

[12] C. L. Chen, M. H. Liu, A Traceable E-cash transfer System against Blackmail via Subliminal Channel, *Electronic Commerce Research and Applications*, Vol. 8, No. 6, pp. 327-333, November-December, 2009.

[13] G. J. Simmons, The Prisoners' Problem and the Subliminal Channel, *Proceedings of Crypto '83*, Santa Barbara, California, USA, 1983, pp. 51-67.

[14] C. L. Chen, T. F. Shih, K. H. Wang, C. H. Chen, W. J. Tsaur, An Investigator Unearths Illegal Behavior via A Subliminal Channel, *Journal of Internet Technology*, Vol. 19, No. 2, pp. 573-580. March, 2018.

[15] S. Micali, M. O. Rabin, S. P. Vadhan, Verifiable Random Functions, *Proceedings of IEEE Symposium on Foundations of Computer Science*, New York, USA, 1999, pp. 120-130.

[16] PUB, N. F., 186, *Digital Signature Standard (DSS)*, National Institute of Standards and Technology, 1994.

[17] O. Goldreich, S. Goldwasser, S. Micali, How to Construct Random Functions, *Journal of the ACM (JACM)*, Vol. 33, No. 4, pp. 792-807, October, 1986.

[18] Y. L. Huang, F. Y. Leu, K. C. Wei, A Secure Communication over Wireless Environments by Using a Data Connection Core, *Mathematical and Computer Modelling*, Vol. 58, No. 5-6, pp. 1459-1474, September, 2013.

[19] M. Abdalla, D. Catalano, D. Fiore, Verifiable Random Functions from Identity-based Key Encapsulation, in: A. Joux (Ed.), *Lecture Notes in Computer Science (LNCS)*, Vol. 5479, Springer, Berlin, Heidelberg, 2009, pp. 554-571.

[20] Y. Dodis, A. Yampolskiy, A Verifiable Random Function with Short Proofs and Keys, in: S. Vaudenay (Ed.), *Lecture Notes in Computer Science (LNCS)*, Vol. 3386, Springer, Berlin, Heidelberg, 2005, pp. 416-431.

[21] Y. Dodis, Efficient Construction of (Distributed) Verifiable Random Functions, in: Y. G. Desmedt (Ed.), *Lecture Notes in Computer Science (LNCS)*, Vol. 2567, Springer, Berlin, Heidelberg, 2003, pp. 1-17.

[22] R. L. Rivest, A. Shamir, L. Adelman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Communications of the ACM*, Vol. 21, No. 2, pp. 120-126, February, 1978.

[23] Accredited Standards Committee, *American National Standard X9.62-2005*, Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA), 2005.

[24] A. Lysyanskaya, Unique Signatures and Verifiable Random Functions from the DH-DDH Separation, in: M. Yung (Ed.), *Lecture Notes in Computer Science (LNCS)*, Vol. 2442, Springer, Berlin, Heidelberg, 2002, pp. 597-612.

[25] W. Diffie, M. E. Hellman, New Directions in Cryptography, *IEEE Transactions on Information Theory*, Vol. 22, No. 6, pp. 644-654, November, 1976.

[26] I. You, Y. L. Huang, F. Y. Leu, J. C. Liu, L. J. Lo, A Secure Wireless Communication System Integrating PRNG and

Diffie-Hellman PKDS by Using a Data Connection Core, *Journal of Internet Technology*, Vol. 15, No 5, pp. 713-726, Sepember, 2014.

[27] Y. L. Huang, F. Y. Leu, I. You, Y. K. Sun, C. C. Chu, A Secure Wireless Communication System Integrating RSA, Diffie-Hellman PKDS, Intelligent Protection-key Chains and a Data Connection Core in a 4G Environment, *Journal of Supercomputing*, Vol. 67, No. 3, pp. 635-652, March, 2014.

[28] Internet Engineering Task Force (IETF) Working Group, *Diffie-Hellman Key Agreement Method*, RFC 2631, June, 1999.

[29] K. L. Tsai, F. Y. Leu, Secure Data-sharing Using Distributed Environmental Matching Keys, *Pervasive and Mobile Computing*, Vol. 42, pp. 513-525, December, 2017.

[30] J. C. Liu, Y. L. Huang, F. Y. Leu, F. C. Chiang, C. T. Yang, W. C. C. Chu, Square Key Matrix Management Scheme in Wireless Sensor Networks, *Computing and Informatics*, Vol. 36, No. 1, pp. 169-185, January, 2017.

[31] C. P. Schnorr, Efficient Signature Generation by Smart Cards, *Journal of Cryptology*, Vol. 4, No. 3, pp. 161-174, January, 1991.

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.