Playing Lottery on the Internet

Jianying Zhou[1] and Chunfu Tan[2]

[1] Oracle Corporation
500 Oracle Parkway, Redwood Shores, CA 94065
United States
Jianying.Zhou@oracle.com

[2] Kent Ridge Digital Labs
21 Heng Mui Keng Terrace
Singapore 119613
cftan@krdl.org.sg

Abstract. The Internet is used by more and more people for personal and business related communication. This paper presents an integrated scheme for playing lottery on the Internet, which includes purchase of tickets, generation of winning number, and claiming of prize. Fairness between the customer and the service provider is maintained at the stages of purchasing tickets and claiming prize. The customer's identity is kept anonymous to the service provider. The sum of sold tickets and the sum of winning tickets are publicly verifiable. The winning number is generated randomly but verifiably. These features will increase the customer's trust in the Internet lottery service.

1 Introduction

Lottery is one of the most widespread forms of gambling in the world. A typical lottery service operates in the following way.

1. The customer selects the type of lottery service and his lucky number.
2. The customer gets the quotation from the service provider.
3. The customer and the service provider exchange the lottery ticket and the payment physically.
4. The service provider generates and publishes the winning number.
5. The winning customer claims the prize. The service provider will verify the winning ticket before the prize is awarded.

The traditional lottery service has some limitations. Customers need to go to an outlet to buy lottery tickets. They may have to join a long queue in front of the outlet. Obviously, this is less efficient for customers. If customers do not have time to buy lottery tickets personally, they may trouble their friends or relatives to buy tickets for them. Then the customers' anonymity cannot be well protected. These limitations could be removed if customers can play lottery over the Internet.

S. Qing, T. Okamoto, and J. Zhou (Eds.): ICICS 2001, LNCS 2229, pp. 189–201, 2001.

In this paper, we present a secure and integrated Internet lottery scheme, which includes purchase of tickets, generation of winning number, and claiming of prize. We identify the desired security requirements for playing lottery on the Internet in Section 2, and review the related work in Section 3. Then, we give an overview of our scheme in Section 4, and propose detailed protocols in Section 5. We conclude the paper in Section 6. The following basic notation is used throughout the paper.

- X, Y: concatenation of two messages X and Y.
- $H(X)$: a one-way hash function of message X.
- $eK(X)$ and $dK(X)$: encryption and decryption of message X with key K.
- $sS_A(X)$: party A's digital signature on message X with key S_A.
- S_A and V_A: party A's private signature key and public verification key.
- P_A and P_A^-: party A's public encryption key and private decryption key.
- $A \rightarrow B : X$: party A sends message X to party B.

2 Security Requirements

When playing lottery on the Internet, the following security requirements should be taken into consideration.

R1. *Fairness on purchase of tickets and claiming of prize*

In the traditional lottery service, when a customer buys a lottery ticket from the service provider, they exchange the money and the ticket face to face, thus fairness is maintained. This is also true when a winner claims the prize from the service provider. In the Internet lottery service, however, the customer and the service provider are distributed over the Internet. After one party receives the other party's item, it may refuse to send its item to the other party, which leaves the other party in an unfair situation. Therefore, a security mechanism is needed to achieve fairness such that either each party gets the other party's item, or no party gets the other party's item at the end of a transaction. Several fair exchange mechanisms are available to satisfy the requirement [1,2,3].

R2. *Anonymity of lottery players*

In the traditional lottery service, when the customer buys a lottery ticket at an outlet, he only needs to pay cash for the ticket. No one knows who the buyer is. If the customer is a winner, he can get the prize by showing the winning ticket. The customer's identity is kept anonymous at the stages of purchasing tickets and claiming prize, thus protecting jackpot winners against blackmail. This requirement is also desirable in the Internet lottery service. An anonymous electronic fund transfer system is needed such that purchase of tickets and claiming of prize will not disclose the customer's identity. The requirement of anonymity has been widely discussed in electronic payment systems [4,5,6,13].

R3. *Verifiable sum of sold and winning tickets*

In the traditional lottery service, the sum of sold tickets is counted by the service provider. As the amount of winning prize is usually related to the sum of sold tickets, the service provider might publish a sum smaller than the actual sum of sold tickets. In addition, the amount of winning prize is usually related to the sum of winning tickets, the service provider might fake winning tickets thus the actual winners will get less of a prize. In the Internet lottery service, it is possible to make the sum of sold tickets and the sum of winning tickets publicly verifiable by the use of cryptographic techniques.

R4. *Random and verifiable generation of winning number*

The winning number should be selected randomly and no ticket is predictably more likely to win than any other tickets. In the traditional lottery service, a random process may be executed or monitored by an outside auditor for the generation of winning number. Unfortunately, as the random process is not repeatable, customers have to trust both the process and the auditing organisation. In the Internet lottery service, it is possible to generate the winning number randomly but publicly verifiable by the use of cryptographic techniques.

3 Previous Work

Some research papers on the lottery service have been published in recent years. Rivest proposed a micropayment scheme based on the use of "electronic lottery tickets" in [10]. In such a scheme, the bank provides an electronic credential to the customer with the micropayment account in good standing. With the credential, the customer can generate lottery tickets and use them to pay for some services provided by the vender. The vender can verify whether a lottery ticket is a winning ticket and claim payment via the bank. The bank pays off winning tickets issued by the customer from the customer's account. This scheme greatly reduces the bank's processing costs since the bank handles only winning tickets instead of each micropayment. Obviously, this scheme is different from the real lottery services, where the roles of buyer and seller are turned around.

Goldschlag and Stubblebine proposed a publicly verifiable lottery scheme based on a delaying function [8]. Each lottery ticket has an equal chance of being selected as a winning ticket. Anyone can calculate the winning number based on the parameters of purchased tickets, and the winning number calculation is repeatable. Since the calculation uses a delaying function, nobody can get the result before the lottery closes.

Syverson presented two versions of a lottery scheme based on the application of the weak protection of secrets [12]. The winning number is determined by the ticket numbers purchased, but no one can control the outcome or determine what it is until after the lottery closes. This is because the outcome is kept secret in a way that is breakable after a predictable amount of time and/or computation.

Sako presented the design and implementation of a lottery server on WWW [11]. The server allows users to define and start a lottery session, participate in that session, and verify its outcome. When a lottery session is initiated, each player submits a random number to the server. The server generates the outcome using a one-way hash function with the concatenation of each player's random number as the input. A rule could be defined to select the winner based on the random and verifiable outcome.

All of these schemes are mainly focused on the mechanisms of winning number generation. There are no integrated mechanisms on fair payment for lottery tickets and claiming of prize. Moreover, the customer's anonymity is not considered either.

4 Overview of a New Scheme

The parties involved in our Internet lottery scheme are the lottery service provider S, the customer C, the bank B, and an *off-line* trusted third party *TTP*. There are several publicly announced dates which are announced well in advance of the running of the lottery service.

- *open of ticket sale*: a time after which customers can purchase lottery tickets.
- *close of ticket sale*: a time after which no new tickets can be purchased.
- *close of winning number generation*: a deadline for customers to be involved in the generation of winning number.

We assume that the lottery is *simple parimutuel*, i.e. the amount of winning prize is solely from the sold tickets, and there is no roll-over of winning prize [12]. We also assume that both the lottery service provider and the customer have an account at their bank.

Purchase of Tickets. A run of Internet lottery starts at the time of *open of ticket sale*. The customer first selects his lucky number N. He also generates two random numbers R_1 and R_2, which will be used in the generation of winning number and claiming of prize (if he is a winner) respectively. Then, the customer submits $(N, H(R_1), H(R_2))$ to the service provider, and gets the quotation from the service provider. After this, the customer requests a cash order from the bank, and exchanges the cash order for the lottery ticket with the service provider. When the service provider deposits the cash order into the bank, the service provider's account will be credited and the customer's account will be debited. The lottery ticket issued by the service provider contains a transaction number, a serial number, the ticket value, and $(N, H(R_1), H(R_2))$.

By the use of a new cryptographic primitive, called the *Certificate of Encrypted Message Being a Signature* (*CEMBS*) [3], the customer and the service provider can fairly exchange the cash order and the lottery ticket. An off-line *TTP* will be involved only if there is something wrong in the exchange. To preserve the customer's anonymity, the customer's identity in the cash order is not disclosed to the service provider, and the ownership of a lottery ticket is not identified by the customer's identity. To make the sum of sold and winning tickets publicly verifiable, the service provider needs to maintain a one-way hash chain. Each sold ticket should be linked into the hash chain. The initial output and the final output of the hash chain, as well as all of the sold tickets should be published at the time of *close of ticket sale*.

Generation of Winning Number. After the ticket-selling session is closed, each customer could be involved in the generation of winning number by submitting his random number R_1 to the service provider. If there is a *denial of service* attack, customers can submit their R_1 to an off-line *TTP*, which are then forwarded to the service provider. The service provider can verify R_1 by checking whether there is a ticket containing $H(R_1)$. Only valid submissions are used in the generation of winning number. The outcome remains random even if most of customers do not submit R_1 for the generation of winning number.

The service provider needs to publish all valid submissions received before the deadline of *close of winning number generation*, thus the process of winning number generation is publicly verifiable, and nobody can predict the outcome before the deadline. As all of the sold tickets are chained and the chain is publicly verifiable, the sum of winning tickets is also publicly verifiable. The service provider cannot forge winning tickets without being detected.

Claiming of Prize. After the winning number is generated, the winning customer claims his prize. The customer first submits his winning ticket to the service provider for verification. If the service provider does not hold R_2 corresponding to $H(R_2)$ in the winning ticket, it means the prize has not been claimed. Then, the service provider requests a cash order from the bank, and exchanges the cash order for R_2 with the customer. When the customer deposits the cash order into the bank, the customer's account will be credited and the service provider's account will be debited.

We use the same techniques as in the ticket-selling session to keep the customer's identity anonymous to the service provider and make the process of prize claim fair to both parties.

Definition of CEMBS. The *CEMBS* technique can be used to prove that an encrypted message is a certain party's signature on a public file without revealing the signature.

Suppose $s = sS_A(m)$ is party A's digital signature on m, and $c = eP_{TTP}(s)$ is the cipher text of s encrypted with the trusted third party's public encryption key. A can generate a *CEMBS*, denoted as *Cert*, to prove that c is indeed the encryption of the signature s without disclosing s. There exists a public verification algorithm **Veri** to check whether $(m, c, Cert)$ is valid.

$$\mathbf{Veri}(m, c, Cert, V_A, P_{TTP}) = yes \text{ or } no$$

If *yes*, the verifier will be convinced that $dP_{TTP}^{-}(c) = sS_A(m)$.

A *CEMBS* could be constructed on the ElGamal public key encryption scheme [7] and the Guillou-Quisquater signature scheme [9].

5 An Internet Lottery Scheme

The following notation is used in the description of our Internet lottery scheme.

- N: the lucky number selected by C.
- R_1, R_2: the random numbers generated by C.
- \$_C, \$_S: the amount of payment for purchase of ticket and for claiming of prize, respectively.
- *TID*: the transaction ID generated by S.
- *ticket_no*: the serial number of a lottery ticket generated by S.
- $ticket = sS_S(TID, ticket\_no, \$\_C, N, H(R_1), H(R_2))$: a lottery ticket issued by S.
- $salt_1, salt_2$: the random salts generated by C.
- *Account_C, Account_S*: the bank account numbers of C and S, respectively.
- $form\_C = eP_B(Account\_C, salt_1), Account\_S, \$\_C, TID, N, H(R_1), H(R_2)$: the content of an electronic cash order for purchase of ticket.
- $form\_S = Account\_S, eP_B(Account\_C, salt_2), \$\_S, ticket$: the content of an electronic cash order for claiming of prize.
- $cash\_C = sS_B(form\_C)$: an electronic cash order issued to C by B.
- $cash\_S = sS_B(form\_S)$: an electronic cash order issued to S by B.
- $cipher\_cash\_C = eP_{TTP}(cash\_C)$: the cipher text of *cash_C* encrypted with the *TTP*'s public encryption key.
- $cipher\_cash\_S = eP_{TTP}(cash\_S)$: the cipher text of *cash_S* encrypted with the *TTP*'s public encryption key.
- *Cert_C*: a *CEMBS* generated by C which can be used to verify whether *cipher_cash_C* is the cipher text of B's signature on *form_C*, i.e.

$$\mathbf{Veri}(form\_C, cipher\_cash\_C, Cert\_C, V_B, P_{TTP}) = yes \text{ or } no$$

- *Cert_S*: a *CEMBS* generated by S which can be used to verify whether *cipher_cash_S* is the cipher text of B's signature on *form_S*, i.e.

$$\mathbf{Veri}(form\_S, cipher\_cash\_S, Cert\_S, V_B, P_{TTP}) = yes \text{ or } no$$

5.1 Protocol 1: Purchase of Tickets

The protocol for purchase of ticket is as follows.

1. $C \rightarrow S : N,\ H(R_1),\ H(R_2)$
2. $S \rightarrow C : \mathit{TID},\ \$\_C,\ Account\_S$
3. $C \rightarrow B : form\_C,\ sS_C(form\_C)$
4. $B \rightarrow C : eP_C(cash\_C)$
5. $C \rightarrow S : \mathit{TID},\ eP_B(Account\_C, salt_1),\ cipher\_cash\_C,\ Cert\_C$
6. $S \rightarrow C : \mathit{TID},\ ticket\_no,\ ticket$
7. $C \rightarrow S : \mathit{TID},\ cash\_C$
8. $S \rightarrow B : form\_C,\ cash\_C$

At Step 1, the customer selects his lucky number N, generates two random numbers (R_1, R_2) and calculates their hash values. The customer sends $(N, H(R_1), H(R_2))$ to the service provider, and keeps (R_1, R_2) confidential. Upon receiving the customer's purchase request, the service provider offers the quotation *\$_C* at Step 2. *TID* is used to identify the transaction, and *Account_S* is the service provider's bank account number for receiving the customer's payment.

Upon receiving the quotation, the customer requests a cash order from the bank at Step 3. After authenticating the request with the customer's signature and checking the balance of the customer's account, the bank issues the cash order *cash_C* at Step 4. The cash order specifies the debiting account (*Account_C*), the crediting account (*Account_S*), the amount of payment (*\$_C*), and the purpose of payment $(\mathit{TID}, N, H(R_1), H(R_2))$. The bank may freeze the amount of payment in the customer's account until *cash_C* expires or the recipient claims the payment with *cash_C*. To preserve the customer's anonymity, *Account_C* is encrypted with the bank's public encryption key [1]. To prevent *cash_C* from being intercepted by the service provider before issuing the ticket, it is encrypted with the customer's public encryption key in transmission at Step 4.

After receiving the cash order from the bank, the customer generates the cipher cash order (*cipher_cash_C*) and a *CEMBS* certificate (*Cert_C*), and sends them to the service provider at Step 5. The service provider can verify whether *cipher_cash_C* is indeed the bank's signature on *form_C*, and whether the crediting account number and the amount of payment are correct. If so, it issues the lottery ticket to the customer at Step 6 [2]. Upon receiving *ticket*, the customer checks $(N, H(R_1), H(R_2))$ to see whether the ticket is what he intends to buy. If so, the customer sends the cash order (*cash_C*) to the service provider at Step 7. The service provider can get the payment by depositing *cash_C* into the bank at Step 8. The bank gets the debiting account number by decrypting $eP_B(Account\_C, salt_1)$, and transfers *\$_C* from *Account_C* to *Account_S*.

[1] To make the customer's repeat transactions unlinkable by the service provider, a random salt is attached to *Account_C* before encryption.

[2] If the service provider aborts after Step 5, the transaction status may not be decided until the time of *close of ticket sale*.

Occasionally, the customer may dislike the lottery ticket he just bought, and wants to abort the transaction by not sending *cash_C* to the service provider at Step 7. Obviously, this is unfair to the service provider where it has issued the ticket to the customer without receiving the payment. With the recovery sub-protocol below, the service provider could force the transaction by sending the lottery ticket and the encrypted cash order to the *TTP* in exchange of the decrypted cash order.

$$\begin{array}{l} 7.1\ S \rightarrow TTP: \mathit{form\_C},\ \mathit{cipher\_cash\_C},\ \mathit{ticket\_no},\ \mathit{ticket} \\ 7.2\ TTP \rightarrow S: \mathit{TID},\ \mathit{cash\_C} \\ 7.3\ TTP \rightarrow C: \mathit{TID},\ \mathit{ticket\_no},\ \mathit{ticket} \end{array}$$

The service provider sends out the recovery request at Step 7.1. The *TTP* can get *cash_C* by decrypting *cipher_cash_C*. Then the *TTP* checks whether *cash_C* is the payment for *ticket* by comparing $(\$\_C, N, H(R_1), H(R_2))$ in *cash_C* and *ticket*. The *TTP* also checks whether the time of *close of ticket sale* has not passed yet. If so, the *TTP* sends *cash_C* to the service provider at Step 7.2, and *ticket* to the customer at Step 7.3. If the recovery request arrives after the time of *close of ticket sale* but before the deadline of *close of winning number generation*, the *TTP* only issues a signed revocation notice to both parties, with which the service provider could exclude the ticket as a valid one.

The lottery ticket issued by the service provider need not be kept confidential. Nobody can claim the prize from the service provider even if holding a winning ticket unless R_2 is presented, which is only known to the customer who bought the ticket.

Once a ticket is sold to the customer, the service provider should link the ticket to a one-way hash chain. Suppose the sum of lottery tickets sold by the time of *close of ticket sale* is j. The hash chain could be created as follows.

$$\begin{aligned} chain_1 &= H(ticket_1) \\ chain_2 &= H(chain_1, ticket_2) \\ &\vdots \\ chain_j &= H(chain_{j-1}, ticket_j) \end{aligned}$$

The service provider needs to publish $chain_1$ and $chain_j$, as well as $ticket_i$ $(i = 1, 2, \cdots, j)$ at the time of *close of ticket sale*. Then, each customer can check whether his ticket is included in the hash chain, and the total number of sold tickets is publicly verifiable. *ticket_no* could be used to quickly identify the location of a ticket in the hash chain [3].

[3] To maintain the scalability, the service provider could create multiple hash chains and allow customers to select which chain their tickets are linked to. The name of the selected hash chain will be added into *ticket* and used with *ticket_no* to identify the ticket location.

The service provider might issue "free" lottery tickets to itself. However, this does not bring profit to the service provider in the *simple parimutuel* lottery. As any valid ticket has to be linked into the hash chain and the total number of sold tickets is publicly verifiable, these "free" tickets will increase the corresponding amount of winning prize. As we will see in Section 5.2 that the generation of winning number is random and out of the service provider's control, these "free" tickets have no higher chance than other tickets to be winning tickets. If they are not selected as winning tickets, the service provider has to pay for them to compensate the actual amount of winning prize.

5.2 Protocol 2: Generation of Winning Number

After the ticket-selling session is closed, each customer could be involved in the generation of winning number by submitting his random number R_1 to the service provider. A signed receipt might be requested by the customer. The protocol for generation of winning number is as follows.

1. $C \rightarrow S : \mathit{ticket\_no},\ R_1$
2. $S \rightarrow C : sS_S(\mathit{ticket\_no}, R_1)$

Of course, if a customer does not want to be involved in the generation of winning number, he can simply give up the right. To make the outcome unpredictable to anybody, we assume there are at least two non-colluding customers who submitted their R_1 to the service provider. The customer's optional involvement improves the scalability compared with the proposal in [11] where the winning number cannot be generated as long as one of the customers does not co-operate.

If there is a denial of service attack when a customer submits his R_1 to the service provider for the generation of winning number, he may invoke the service from an off-line TTP [4].

1. $C \rightarrow TTP : \mathit{ticket\_no},\ R_1$
2. $TTP \rightarrow C : sS_{TTP}(\mathit{ticket\_no}, R_1)$

The TTP passes these submissions received by the deadline of *close of winning number generation* to the service provider. Thus, the service provider cannot influence the outcome by deliberately rejecting some submissions.

Suppose $R_{11}, R_{12}, \cdots, R_{1k}$ are the random numbers received by the deadline of *close of winning number generation*. Without knowing the customer's identity, the service provider can verify R_{1i} $(i = 1, 2, \cdots, k)$ by checking whether there is a ticket containing $H(R_{1i})$. Only valid submissions are used in the generation

[4] It mainly protects against the denial of service attack from a dishonest service provider that intends to influence the outcome of winning number generation. Other measures are needed to protect against the distributed denial of service attacks.

of winning number. The service provider needs to publish these submissions. Thus, customers who made valid submissions can check whether their random numbers are used in the generation of winning number. If not, customers can use the signed receipts to prove the service provider's misbehaviour.

Suppose all of the above submissions are valid. The winning number could be generated by the use of a one-way hash function with R_{1i} ($i = 1, 2, \cdots, k$) as its input. A pre-defined rule could be used to map the winning number to the winning tickets.

$$\text{winning number} = H(R_{11}, R_{12}, \cdots, R_{1k})$$

The number of submissions and each submission received from customers are random, thus nobody (not even the service provider) can predict the outcome before the deadline of *close of winning number generation*. Further, with a one-way hash function, it is computationally infeasible to find the pre-image of a designated winning number.

As the random numbers used in the generation of winning number are published, the process of winning number generation is publicly verifiable. In addition, as the initial output and the final output of the chained tickets have also been published, each winning ticket is publicly verifiable as well. It is computationally hard for the service provider to forge a winning ticket without being detected.

In practice, random numbers submitted to the service provider may not be published instantaneously and the clock may not be well synchronized among all participants. Then, a dishonest service provider may try to fiddle the outcome of winning number generation by adding the favorite random numbers of its valid tickets soon after the deadline of *close of winning number generation*. To prevent such kind of possible cheating, a delaying function [8] could be used in the generation of winning number thus the service provider cannot get the result of winning number until actually publishing all valid random numbers received by the deadline.

5.3 Protocol 3: Claiming of Prize

The protocol for claiming of prize is as follows.

1. $C \to S: \ TID,\ ticket\_no,\ \$\_C,\ N,\ H(R_1),\ H(R_2),\ ticket,$
   $eP_B(Account\_C, salt_2)$
2. $S \to B: form\_S,\ sS_S(form\_S)$
3. $B \to S: eP_S(cash\_S)$
4. $S \to C: \ TID,\ Account\_S,\ cipher\_cash\_S,\ Cert\_S$
5. $C \to S: \ TID,\ R_2$
6. $S \to C: \ TID,\ cash\_S$
7. $C \to B: form\_S,\ cash\_S$

At Step 1, the customer sends the winning ticket to the service provider. The customer's bank account number (*Account_C*) is also provided for receiving the prize. To keep the customer's identity anonymous, *Account_C*, to which a random salt is attached, is encrypted with the bank's public encryption key.

Upon receiving the winning ticket from the customer, the service provider checks whether it is indeed a winning ticket by verifying its lucky number N. The service provider further checks whether the prize has been claimed by searching R_2 corresponding to its $H(R_2)$. If N is the winning number and R_2 is not found, the service provider requests a cash order from the bank at Step 2.

After authenticating the request with the service provider's signature and checking the balance of the service provider's account, the bank issues the cash order *cash_S* at Step 3. The bank may freeze the amount of payment in the service provider's account until *cash_S* expires or the recipient claims the payment with *cash_S*. To prevent *cash_S* from being intercepted by the customer before releasing R_2, it is encrypted with the service provider's public encryption key in transmission at Step 3.

After receiving the cash order from the bank, the service provider generates the cipher cash order (*cipher_cash_S*) and a *CEMBS* certificate (*Cert_S*), and sends them to the customer at Step 4. The customer can verify whether *cipher_cash_S* is indeed the bank's signature on *form_S*, and whether the crediting account number and the amount of payment are correct [5]. If so, the customer releases R_2 to the service provider at Step 5. Upon receiving R_2, the service provider checks whether R_2 is the random number matching $H(R_2)$ in the winning ticket. If so, the service provider sends the cash order (*cash_S*) to the customer at Step 6. The customer can get the payment by depositing *cash_S* into the bank at Step 7.

In the above prize claim protocol, a dishonest service provider might use its advantage in the transaction, i.e. holding R_2 at Step 5, to refuse the payment at Step 6 by falsely claiming the prize related to the winning ticket has already been paid. Obviously, this is unfair to the customer where he has released R_2 of his winning ticket to the service provider without receiving the payment. With the recovery sub-protocol below, the customer could get the payment by sending the winning ticket plus R_2 as well as the encrypted cash order to the *TTP* in exchange of the decrypted cash order.

$$\begin{aligned}
6.1\ C \rightarrow TTP :\ & form\_S,\ cipher\_cash\_S,\\
& TID,\ ticket\_no,\ \$\_C,\ N,\ H(R_1),\ H(R_2),\ R_2\\
6.2\ TTP \rightarrow C :\ & TID,\ cash\_S\\
6.3\ TTP \rightarrow S :\ & TID,\ R_2
\end{aligned}$$

[5] Although *Account_C* is in cipher text, the customer can verify *Account_C* which is encrypted by himself with the bank's public encryption key at Step 1.

The customer sends out the recovery request at Step 6.1 [6]. The TTP can get *cash_S* by decrypting *cipher_cash_S*. Then the TTP checks whether R_2 matches $H(R_2)$ in *ticket* which is specified in *cash_S* as the winning ticket for receiving the prize. If so, the TTP sends *cash_S* to the customer at Step 6.2, and R_2 to the service provider at Step 6.3. Thus, the exchange remains fair.

6 Conclusion

The traditional lottery game may exclude busy people to play because of its inefficient ticket-selling channel. The Internet lottery game can remove the limitation and even provide some new features.

We proposed an integrated Internet lottery scheme covering purchase of tickets, generation of winning number, and claiming of prize. It has the following features.

– The customer and the service provider need not trust each other.
– Both the customer and the service provider are guaranteed not to be cheated when purchasing tickets and claiming prize.
– The customer's identity is not disclosed to the service provider throughout the service.
– The total number of sold tickets and the total number of winning tickets are publicly verifiable. The service provider cannot gain profit by hiding lottery revenue or faking winning tickets.
– The winning number is generated randomly. Nobody, not even the service provider, can predict the outcome.
– Each customer has the freedom to be involved or not in the winning number generation without affecting the randomness of the outcome.
– The process of winning number generation is publicly verifiable.

These features make the Internet lottery service attractive to customers.

Acknowledgements. We thank the anonymous referees for valuable comments. The first author's work was carried out at Kent Ridge Digital Labs. The second author's work was sponsored under the postgraduate scholarship of National University of Singapore.

References

1. N. Asokan, V. Shoup and M. Waidner. *Optimistic fair exchange of digital signatures.* Lecture Notes in Computer Science 1403, Advances in Cryptology: Proceedings of Eurocrypt'98, pages 591–606, Helsinki, Finland, June 1998.
2. G. Ateniese. *Efficient verifiable encryption (and fair exchange) of digital signatures.* Proceedings of 6th ACM Conference on Computer and Communications Security, pages 138–146, Singapore, November 1999.

[6] The winning ticket is included in *form_S*.

3. F. Bao, R. H. Deng and W. Mao. *Efficient and practical fair exchange protocols with off-line TTP*. Proceedings of 1998 IEEE Symposium on Security and Privacy, pages 77–85, Oakland, California, May 1998.
4. M. Bellare, J. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, E. Van Herreweghen and M. Waidner. *Design, implementation and deployment of the iKP secure electronic payment system*. IEEE Journal on Selected Areas in Communications, 18(4):611–627, April 2000.
5. J. Camenisch, U. Maurer and M. Stadler. *Digital payment systems with passive anonymity-revoking trustees*. Lecture Notes in Computer Science 1146, Computer Security: Proceedings of 1996 European Symposium on Research in Computer Security, pages 33–43, Rome, September 1996.
6. G. Davida, Y. Frankel, Y. Tsiounis and M. Yung. *Anonymity control in e-cash systems*. Lecture Notes in Computer Science 1318, Proceedings of 1997 Financial Cryptography, pages 1–16, Anguilla BWI, February 1997.
7. T. ElGamal. *A public-key cryptosystem and a signature scheme based on discrete logarithms*. IEEE Transactions on Information Theory, IT-31(4):469–472, July 1985.
8. D. M. Goldschlag and S. G. Stubblebine. *Publicly verifiable lotteries: Applications of delaying functions*. Lecture Notes in Computer Science 1465, Proceedings of 1998 Financial Cryptography, Anguilla BWI, February 1998.
9. L. C. Guillou and J. J. Quisquater. *A paradoxical identity-based signature scheme resulting from zero-knowledge*. Lecture Notes in Computer Science 403, Advances in Cryptology: Proceedings of Crypto'88, pages 216–231, Santa Barbara, California, August 1988.
10. R. Rivest. *Electronic lottery tickets as micropayments*. Lecture Notes in Computer Science 1318, Proceedings of 1997 Financial Cryptography, pages 307–314, Anguilla BWI, February 1997.
11. K. Sako. *Implementation of a digital lottery server on WWW*. Lecture Notes in Computer Science 1740, Proceedings of CQRE'99, pages 101-108, Dusseldorf, Germany, December 1999.
12. P. Syverson. *Weakly secret bit commitment: Applications to lotteries and fair exchange*. Proceedings of 11th IEEE Computer Security Foundations Workshop, Rockport, Massachusetts, June 1998.
13. E. Van Herreweghen. *Secure anonymous signature-based transactions*. Lecture Notes in Computer Science 1895, Computer Security: Proceedings of 2000 European Symposium on Research in Computer Security, pages 55–71, Toulouse, France, October 2000.