Available online at www.sciencedirect.com

ScienceDirect

Computer Communications 29 (2006) 2830–2840

computer communications

www.elsevier.com/locate/comcom

# Practical electronic lotteries with offline TTP

Sherman S.M. Chow *, Lucas C.K. Hui, S.M. Yiu, K.P. Chow

*Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong*

Available online 7 December 2005

## Abstract

A number of electronic lottery (e-lottery) schemes have been proposed; however, none of them can satisfy all the identified requirements. In particular, some of them require either a certain subset of players or a trusted third party (TTP) to remain online in order to generate the winning number(s), and some suffer from the forgery ticket attack. Based on various advanced cryptographic techniques, we propose a new e-lottery scheme that can satisfy all the identified requirements without the presence of TTP for generating the winning numbers, yet the result of this generation is publicly verifiable.

*Keywords:* e-lottery; Applied cryptography; Verifiable random function

## 1. Introduction

Lottery is a multi-billion-dollar industry. Apart from gambling, lottery exists in other forms such as fund-raising coupon for charity. In a typical lottery, there is one dealer and a large number of players. Depending on the rules of the game, players bet on a single number or a combination of numbers chosen from a pre-defined domain. A random process (e.g., drawing of lots) is used to determine the combination of winning numbers. Due to the randomness of the process, it is not repeatable; therefore, the process is usually executed or monitored by a trusted auditing organization. With the popularity of the Internet, it is natural to ask whether we can have a secure electronic lottery (e-lottery) scheme, that is, whether it is possible to have ticket purchase, winning result generation and prize claiming all done over the Internet. Indeed, e-lottery scheme may provide something extra to a traditional lottery game. Apart from the obvious advantage that the players can buy the ticket whenever and wherever they are, without possibly queueing for a long time in front of the lottery ticket outlets; the players may enjoy anonymity of ticket buying and prize claiming [1]. From another point of view, a wide variety of lottery game can be supported by a single implementation by merely assigning different interpretation on the random stream of bits generated.

To design a good e-lottery scheme is not an easy task. It must be *practical*. Usability is one of the issues. It should resemble a traditional lottery game. For example, after the players' purchase of ticket, they do not need to keep an eye on the game until the winning result is announced. For its counterpart in Internet, the players are not required to be online for the whole duration of the game, except at the moment of ticket purchase and prize claiming. However, existing schemes may require some players to be online to ensure the fairness of the game. As the Internet extends to wireless devices, this requirement is simply not realistic. For fairness reason, the dealer cannot discriminate against a player who is not online at a certain time.

Security is another important issue. A huge amount of money is involved in each game. For example, we should prevent premature disclosure of winning result and forgery of winning ticket. In the real world, to let the players assured of the unbiased and unpredictable generation of winning result we usually employ mechanical lotteries that

* Corresponding author.
*E-mail addresses:* smchow@cs.hku.hk (S.S.M. Chow), hui@cs.hku.hk (L.C.K. Hui), smyiu@cs.hku.hk (S.M. Yiu), chow@cs.hku.hk (K.P. Chow).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.10.034

are performed on TV live show with the presence of a certified auditor. However, it is not a trivial task to show the winning result generation process is not biased if the lottery game is electronic instead of a mechanical one. Besides the security requirements related to the winning result generation, we need to investigate the related security threats of all distinct phases of a typical lottery game, such as ticket purchase and prize claiming.

### 1.1. Requirements

While there are many different interpretations of "security" issues in an e-lottery, we define a concept of a "fair lottery game over the Internet". We believe the following criteria are common to most practical e-lottery schemes:

(1) *Random generation of the winning result.* Each possible combination of numbers from the domain is equally likely to be the winning result. No player can predict the result better than guessing. No one (including the dealer) can bias the generation.

(2) *The winning result is publicly verifiable.* Every player can verify the winning result, i.e., the dealer cannot cheat in the generation process.

(3) *The total revenue and the number of winning tickets are publicly verifiable.* The amount of winning prizes is usually related to the dealer's revenue and hence the number of sold tickets should be publicly verifiable. In some lottery games, the prize for each winning player is not fixed but depends on the total number of winning tickets; hence, the number of winning tickets should also be publicly verifiable.

(4) *Forgery of ticket is impossible.* Either the players or the dealer cannot forge a ticket. In particulars, players cannot create a winning ticket which they did not purchase. The dealer cannot forge a ticket which is the same as a winning ticket after the winning result is generated.

(5) *User is not required to be online for the generation of the winning result.* In some previous schemes, the generation of the winning result requires some players to remain online. This is not realistic, especially in a large scale game or due to the proliferation of mobile devices.

(6) *Anonymity of players.* Players' anonymity is of paramount importance especially for the winning player(s). If a winning player's anonymity is compromised, he or she may face the threat of blackmail, beggars, etc.

(7) *Confidentiality of the ticket's value.* In some lottery games, the players bet on a single number or a combination of numbers chosen from a pre-defined domain. In the case more than one winning players bet on the same value, the prize will be shared equally among them. In this setting, we need to keep the confidentiality of the ticket's value, or various attacks are possible. For examples, cutting down a certain ticket value's winning share by intentionally "duplicating" other's ticket value, or eavesdropping the channel between the dealer and the clients to learn the "popularity" of the values and betting on the most "profitable" number.

(8) *Fairness on purchase of tickets and claiming of prize.* Purchase of tickets and claiming of prizes are based on the principle of fair exchange. That is, either both parties (the dealer and the player in our case) get the other party's item, or no party gets the other party's item at the end of a transaction.

(9) *No early registration of player is required.* Some schemes (e.g. [2]) require early registration to avoid collusion of players. This is an unrealistic requirement since early registration is not required in traditional lottery.

(10) *No online trusted third party is assumed.* A huge financial damage is resulted usually if cheating occurred, either by insiders or outsiders. The security of an e-lottery game should not be entirely relied up on the presence of a trusted third party.

### 1.2. Previous work

There are many existing e-lottery schemes (e.g. [1–9]) but only three of them [1,2,8] address similar issues as our scheme without resort to a trusted third party (TTP)[1]. Some schemes assume a model different from ours: Hall and Schneier [4] consider the lottery game as a *real-time* interactive game between the players and the dealer in the casino, while the security of [3] is guaranteed by using *multiple* dealers. Some schemes [5,6,9] simply assume the existence of a *trusted random source*, for example, from the random number generation process executed by a TTP. Moreover, Sako [9] does not address the anonymity requirement, the forgery attack and the fairness on the purchase of tickets and claiming of prize; while [6] is shown to be insecure by Ham and Kim [5].

Recently an e-lottery proposal "electronic national lotteries" [7] in proposed; however, a TTP (called "verifier" in their terms) is assumed to take into accounts of all players' winner tickets in the winning result generation (so that the dealer cannot predict the winning result until the last player's decision is made) and will not "insert" any forged tickets to affect the winning result generation. Due to the above mentioned reasons, we mainly compare our scheme with [1,2,8]. Table 1 shows a comparison between the existing schemes and our new scheme. None of these existing e-lottery schemes can satisfy all the requirements that we have identified.

In [2], only the tickets sold in the "critical purchase phase" are used to determine the winning result. This

[1] TTP may be required for fair exchange, and we still require a conventional certificate authority for the distribution of public keys, as other existing schemes.

Table 1
A comparison of e-lottery schemes

| Scheme | Requirements | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Verifiable lotteries [2] | Y | Y | N | N | Y | N | Y[b] | N | N | Y |
| Digital lottery server [9] | Y | Y | N[c] | Y | Y | N | N | N | Y | N |
| Fair e-lotteries [8] | Y | Y | N | Y | N | N | Y[b] | N | Y | Y |
| Lottery on the Internet [1] | Y | Y | Y | Y | N | Y | N[c] | Y | Y | Y[a] |
| Our proposed scheme | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y[a] |

[a] TTP may be required for fair exchange, and we still require a conventional certificate authority for the distribution of public keys, as other existing schemes.

[b] In their settings, players are not betting on any particular value.

[c] But we believe that the scheme may be modified to satisfy this requirement.

makes forgery of winning ticket possible: after the winning result is generated, the dealer can "insert" the winning ticket to the tickets sold before the critical purchase phase, without altering the winning result. The dealer can do so since the sequence numbers and the timestamps associated with the tickets are not guaranteed to be sequential. For example, when multiple purchase requests are made simultaneously but one of the buyers refuses to pay for the ticket afterwards. Besides, the dealer may simply replace the ticket value of a certain ticket which is under his control and "sold" before the critical purchase phase. These malicious acts may not be noticed easily except there is a trusted third party or the players themselves who checks *each ticket one by one* with previously downloaded tickets list *from time to time*, which is impractical since the number of tickets sold is potentially in millions. As forgery of winning ticket is possible, the total number of winning tickets can be altered easily.

The first e-lottery scheme that uses the delaying function (see the definition in Section 2) to prevent forgery ticket attack is [2]. In fact, our scheme also uses a similar idea to prevent the forgery ticket attack. However, their scheme applies the function only on a portion of the tickets sold while our scheme applies the function to all tickets sold so as to achieve a higher level of security.

Instead of applying the delaying function in the winning result generation phase, Kushilevitz and Rabin [8] apply the delaying function in the verification phase. Their scheme tries to prevent the forgery ticket attack by imposing a time limit for the prize claiming phase. Players must do the verification through a delaying function immediately afterwards, or they will suffer from losing the game due to the insufficient time for verification. This requirement is not realistic. Based on their scheme, the exact number of winning tickets cannot be accurately calculated unless all the winning tickets are *claimed* accordingly. In addition, this scheme has not addressed the anonymity and fair exchange issues.

The first piece of work integrating fair exchange with e-lottery is [1], this scheme satisfies most of the requirements we have identified. However, their winning result generation requires some players to be online, which make the scheme not realistic and not robust. For example, the winning result cannot be generated if none of the players participates in the generation process. Also, the winning result may be biased if all participating players collude, which is an important issue for small scale e-lottery game or when the e-lottery game is not yet popular in the beginning. Moreover, they have not considered the security issue of *not* encrypting the ticket's value.

### 1.3. Our contribution

In this paper, we propose a new e-lottery scheme which satisfies all of the identified requirements by using the *verifiable random function*: a pseudorandom function that provides a non-interactive verifiable proof for the correctness of its output. With the help of hash chain, our scheme makes sure that the lottery dealer generates the winning number(s) based on all the tickets sold. The generation of the winning number(s) does not depend on the participation of the players or a trusted third party, but its randomness is still verifiable.

### 1.4. Organization

In Section 2, we describe the core cryptographic protocols for our proposed scheme. We describe our proposed e-lottery scheme in Section 3, discussing its design philosophy and also its security. Then, we conclude the paper and discuss some open problems in Section 4.

## 2. Preliminaries

Before presenting our results, we review the definitions of bilinear pairing, the cryptographic primitive for building some of the cryptographic protocols used in our scheme. Related complexity assumptions are discussed in this section too. After that, we discuss the properties of various cryptographic schemes to be used in our scheme, which includes hash function, delaying function, verifiable random function and verifiably encrypted signature. We will also review concrete implementations of verifiable random function and verifiably encrypted signature scheme from bilinear pairings.

### 2.1. Bilinear pairings

Bilinear pairing (see [10–13] for implementation details) is an important primitive for many cryptographic schemes [10,14,11,15–18]. Here, we describe some of its key properties.

Let $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, \cdot)$ be two cyclic groups of prime order $q$. The bilinear pairing is given as $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, which satisfies the following properties:

(1) *Bilinearity.* For all $P, Q, R \in \mathbb{G}_1$, $\hat{e}(P+Q, R) = \hat{e}(P,R)\hat{e}(Q,R)$, and $\hat{e}(P, Q+R) = \hat{e}(P,Q)\hat{e}(P,R)$.

(2) *Non-degeneracy*. There exists $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq 1$.
(3) *Computability*. There exists an efficient algorithm to compute $\hat{e}(P, Q)\ \forall P, Q \in \mathbb{G}_1$.

Weil pairing [10] and Tate pairing [12] are two admissible bilinear pairings.

**Definition 1.** *A Bilinear Diffie–Hellman (BDH) parameter generator* is a probabilistic algorithm that takes a security parameter $k$ as input, and outputs a 5-tuple $(q, P, \mathbb{G}_1, \mathbb{G}_2, \hat{e}(\cdot,\cdot))$ satisfying the following conditions: $q$ is a prime where $2^{k-1} < q < 2^k$, $\mathbb{G}_1$ and $\mathbb{G}_2$ are group of order $q$, $P$ is a generator of $\mathbb{G}_1$ and $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is an admissible bilinear map.

More details on this BDH parameter generator can be found in [10]. Now we define the complexity assumptions related to the cryptographic primitives used in our scheme.

**Definition 2.** Given a generator $P$ of a group $\mathbb{G}$ and a 3-tuple $(aP, bP, cP)$, the *Decisional Diffie–Hellman problem* (DDHP) is to decide whether $c = ab$.

**Definition 3.** Given a generator $P$ of a group $\mathbb{G}$ and a 2-tuple $(aP, bP)$, the *Computational Diffie–Hellman problem* (CDHP) is to compute $abP$.

**Definition 4.** If $\mathbb{G}$ is a group such that DDHP can be solved in polynomial time but no probabilistic algorithm can solve CDHP with non-negligible advantage within polynomial time, then we call $\mathbb{G}$ a Gap Diffie–Hellman (GDH) group.

**Definition 5.** Given a generator $P$ of a group $\mathbb{G}$ and a $q+1$-tuple $(xP, x^2P, \ldots, x^qP, \tau) \in \mathbb{G}_1^q \times \mathbb{G}_2$, the *q-Decisional Bilinear Diffie–Hellman inversion problem* ($q$-DBDHI) is to decide whether $\tau = \hat{e}(P, P)^{1/x}$.

We assume the existence of a bilinear map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ that one can solve Decisional Diffie–Hellman problem in polynomial time, but Computational Diffie–Hellman problem and $q$-Decisional Bilinear Diffie–Hellman inversion problem are intractable.

### 2.2. Hash function

Informally, a *hash function* maps a string of arbitrary length to a string of fixed length. *Cryptographic* hash function should be one-way (i.e., hard to invert), collision-free (i.e., hard to find inputs with the colliding output) with random distribution. More formally, a secure cryptographic hash function should satisfy the following properties:

(1) *Mixing transformation*. On any input $m$, the output hashed value $H(m)$ is computationally indistinguishable from a uniform binary string in the interval $[0, 2^{|H|})$, where $|H|$ denotes the output length of $H$.
(2) *Preimage resistant* (*one-way, or hard to invert*). Given $y$ from the range $H$, it is hard to find $m$ such that $H(m) = y$.
(3) *Second-preimage resistant*. Given $m$ from the domain of $H$, it is hard to find $m' \neq m$ such that $H(m) = H(m')$.
(4) *Collision resistant* (*or collision-free*). It is hard to find a pair of distinct messages $m$,$m'$ such that $H(m) = H(m')$.

In our scheme, cryptographic hash function is used. We refer our readers to work like [19] for various definitions of hash functions, some generic constructions and attacks.

### 2.3. Delaying function

*Delaying function* was introduced by Goldschlag and Stubblebine [2], which is a function that is moderately hard to compute and cannot be parallelized. The time to compute the function depends on the parameters chosen. It may take several hours or several days to complete, using the fastest existing implementation. The output is *publicly verifiable* by going through the same computation. An example of the delaying function is to use the input bits as the key for a cipher with a very long period, run the cipher in output feedback mode (OFB) and use the resulting bits as the result. Notice that in this implementation, it is possible to release the intermediate result such that the verifier can choose to do a limited verification by going through only a portion of the process. Please refer to [2] for more details of the delaying function.

### 2.4. Verifiable random function

*Verifiable random function* (hereafter referred as VRF) was introduced by Micali et al. [20]. Basically it is a pseudorandom function [21] providing a non-interactively verifiable proof for the output's correctness.

Based on the notation in [18], a function family $F_{(\cdot)}(\cdot)\colon \{0,1\}^k \mapsto \{0,1\}^{l(k)}$ is a verifiable random function if there exists polynomial-time algorithms ($\texttt{Gen}(\cdot)$, $\texttt{Eval}(\cdot,\cdot)$, $\texttt{Prove}(\cdot,\cdot)$, $\texttt{Verify}(\cdot,\cdot,\cdot,\cdot)$) such that

- $\texttt{Gen}(k)$ is a probabilistic algorithm which output a secret key $SK$ (the secret seed of the random function) and the corresponding public key $PK$ (made available to the public for verification).
- $\texttt{Eval}(SK, x)$ is an algorithm that computes the VRF's output $y = F_{SK}(x)$.
- $\texttt{Prove}(SK, x)$ is an algorithm that computes the proof $\pi$ that $y = F_{SK}(x)$.
- $\texttt{Verify}(PK, x, y, \pi)$ is an algorithm that verifies $y = F_{SK}(x)$.

And a VRF has to satisfy the following properties:

- *Uniqueness.* If $\texttt{Verify}(PK, x, y_1, \pi_1) = \texttt{Verify}(PK, x, y_2, \pi_2) = 1$, $y_1 = y_2$.

- *Computability*. $\texttt{Eval}(SK, x) = F_{SK}(x)$ is efficiently computable.
- *Provability*. $(y, \pi) = (\texttt{Eval}(SK, x), \texttt{Prove}\ (SK, x)) \Rightarrow \texttt{Verify}(PK, x, y, \pi) = 1$.
- *Pseudorandomness*. The probability that an adversary can tell any bit of $F_{SK}(x)$ for $x$ he chose is negligible even if he has seen the values of many $F_{SK}(x')$ given $x' \neq x$.

VRF is applicable in many protocols. For example, it can be used to construct a micropayment scheme [22], a verifiable transaction escrow scheme [23] and an offline anonymous compact e-cash scheme [15]. In fact, such a powerful cryptographic primitive is not easy to be efficiently constructed. To give some high level idea, a signature scheme which is existentially unforgeable against adaptive adversaries can only give rise to a verifiable unpredictable function (VUF) [17]. The notion of VUF is similar to that of the VRF but it only achieves unpredictability, a weaker property when compared with pseudorandomness of the VRF. It is possible to convert a VUF into a VRF, but existing constructions [16,18,20] involve inefficient Goldreich–Levin hardcore bit [24]. (Please refer to [16–18,20] for more formal discussions of the VRF and the related notions.) Recently a simple and efficient construction of VRF is proposed [17]. The proofs and the keys involved are short, and can be made distributed such that at least $t+1$ out of $n$ $(1 \leqslant t+1 \leqslant n)$ parties holding shares of the secret key can jointly compute the function using their shares, but not for any coalition of $t$ parties.

Here, we review the most efficient existing VRF construction in [17].

$\texttt{KeyGen}(k)$: It first executes the BDH parameter generator with $k$ as the input security parameter. Let $\mathbb{G}_1$ be a GDH group and $P$ be a generator of $\mathbb{G}_1$. Randomly choose $s \in_R \mathbb{Z}_q^*$, keep it as the secret key (i.e., $SK = s$) and compute the corresponding public key $PK = sP$.

$\texttt{Eval}(SK, x)$: $F_{SK}(x)$ is defined as $(x+s)^{-1}\hat{e}(P, P)$.

$\texttt{Prove}(SK, x)$: The proof of correctness of $F_{SK}(x)$ is $\pi_{SK}(x) = (x+s)^{-1}P$.

$\texttt{Verify}(PK, x, y, \pi)$: Return true if both of the following equalities hold: $\hat{e}(xP + PK, \pi) = \hat{e}(P, P)$ and $y = \hat{e}(P, \pi)$, false otherwise.

The security of the above scheme depends on intractability of $q$-DBDHI. Security analysis and extensions of the above scheme can be found in [17].

### 2.5. Verifiable encrypted signature

*Verifiable encrypted signature* (*VES*) is an encrypted signature which can be easily verified that the decryption of it gives a certain party's signature on a public message, without revealing the signature. Besides, the recipient of VES can assure that there exists an adjudicator who can decrypt this VES to get back the signature. This class of cryptographic schemes is useful in applications such as online contract signing [25], since the recipient of the VES cannot get the signature directly, but in case of dispute (i.e., another party refuses to provide the original signature even after obtained the "information" in exchange of the signature), the recipient can ask for an offline TTP to get back the signature by decryption.

It is also known as *certificate of encrypted message being a signature* (*CEMBS*) in some previous work [25]. VES can be constructed from aggregate signature scheme [14] or ElGamal public key encryption scheme [26]. Below we review the pairing-based construction, which has the feature of a short signature [11]. Let $H(\cdot)$ be a cryptographic hash function where $H : \{0,1\}^* \rightarrow \mathbb{G}_1$.

$\texttt{KeyGen}(k)$: It is the algorithm used by both of the signer and the adjudicator to generate his or her own key pairs. It first executes the BDH parameter generator with $k$ as the input security parameter. Let $\mathbb{G}_1$ be a GDH group and $P$ be a generator of $\mathbb{G}_1$. The algorithm randomly chooses $s \in_R \mathbb{Z}_q^*$, keeps it as the secret key and computes the corresponding public key $PK = sP$.

$\texttt{Sign}(SK, m)$: The signature to be "encrypted" by the VES creation algorithm is in the form of $\sigma = sH(m)$, it is indeed the form of short signature proposed in [11].

$\texttt{Verify}(\sigma, PK, m)$: If $\sigma$ is a valid signature on message $m$ corresponding to the public key $PK$, $\hat{e}(P, \sigma) = \hat{e}(PK, H(m))$. Again, it matches with the verification process of the short signature scheme in [11].

$\texttt{VESCreate}(SK, x)$: It is the algorithm for creation of a VES, taking the message $m$, the secret key $s$, the adjudicator's public key $PK_A$ as the input, the creator performs the following steps:

(1) Randomly choose $r \in_R \mathbb{Z}_q^*$.
(2) Compute $W = sH(m) + rPK_A$.
(3) Compute $U = rP$.
(4) Output the VES as $\{W, U\}$.

$\texttt{VESVerify}(VES, PK, PK_A, m)$: It is the algorithm for the recipient of VES to verify whether the VES is a valid one, i.e., the holder of the secret key corresponding to $PK_A$ can decrypt the VES $\{W, U\}$ and get the signature on the message $m$ signed by the secret key corresponding to $PK$. The VES is considered as correctly generated if and only if $\hat{e}(W, P) = \hat{e}(H(m), PK)\hat{e}(U, PK_A)$.

$\texttt{Adjudicate}(VES, SK_A)$: It is the algorithm for the adjudicator to get back the signature from the VES $\{W, U\}$ in case of dispute. The original signature can be obtained by $\sigma = W - xU$.

The security of the above scheme depends on intractability of CDHP. Security analysis of the above scheme can be found in [14].

## 3. A new e-lottery scheme

In this section, we describe our proposed e-lottery scheme. We show how to integrate many cryptographic techniques, which includes VRF, hash chain and delaying function to build a scheme that satisfies all the identified requirements. To ease our discussion, we assume that we only generate one winning number and each ticket bets on one number. It is easy to extend the scheme to cover a combination of numbers.

### 3.1. Entities involved

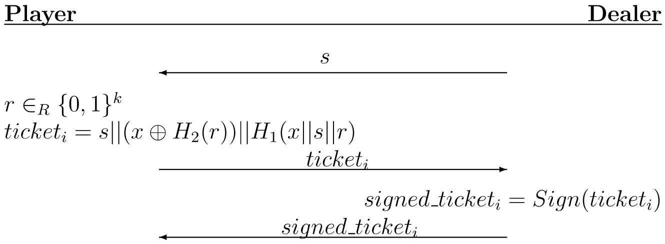
The entities involved in our e-lottery game includes:

(1) *Dealer*. The lottery organizer who launches the lottery, sells tickets to gain revenue, generates winning result and gives out prizes.
(2) *Players*. The buyers of tickets who can claim the prize if they win.
(3) *Bank*. The organization which both the dealer and the players have current account in it, served only when players purchase tickets or when players claim the prizes.
(4) *Trusted third party* (*TTP*). The organization for ensuring fair exchange between the dealer and the players.
(5) *Certificate authority* (*CA*). The authority on Internet that issues and manages issues digital certificates, which contains public keys and certifying information about the public key owners.

### 3.2. Our proposed scheme

Our scheme consists of five phases, namely, Setup, Ticket Purchase, Winning Result Generation, Prize Claiming and Player Verification. We assume that the set of numbers that the players can bet on is $\{1, 2, \ldots, u\}$ and the output domain for the VRF is $\{1, 2, \ldots, v\}$ with $v \geqslant u$. Let $k$ be the security parameter, $H_1(\cdot)$ be a cryptographic hash function that maps a bit string of arbitrary length (i.e., $\{0,1\}^*$) into a bit string of fixed length (e.g., 160 bits) and let $H_2(\cdot)$ be another cryptographic hash function that maps $\{0,1\}^k$ to $\{0, 1\}^{\lceil \log_2 u \rceil}$.

- Setup:
  (1) Dealer generates a secret key $SK$ and a public key $PK$ by the algorithm $\texttt{Gen}$ with the security parameter $k$ as the input.
  (2) Dealer publishes the following items.
    (a) Hash functions $H_1(\cdot)$ and $H_2(\cdot)$, delaying function $D(\cdot)$, and VRF's algorithms: $\texttt{Gen}(\cdot)$, $\texttt{Eval}(\cdot,\cdot)$, $\texttt{Prove}(\cdot,\cdot)$ and $\texttt{Verify}(\cdot,\cdot,\cdot,\cdot)$.
    (b) VRF public key $PK$ and the dealer's digital certificate.
    (c) The amount of time $t$ in which the dealer must release the winning ticket value generated. (This is the input parameter controlling the time complexity of the delaying function.)
- Ticket Purchase (Fig. 1):
  (1) Player chooses his favorite number $x$ and randomly picks a random bit string $r$ from $\{0,1\}^k$. $r$ is kept in secret.



Player
Dealer
$s$
$r \in_R \{0,1\}^k$
$ticket_i = s||(x \oplus H_2(r))||H_1(x||s||r)$
$ticket_i$
$signed\_ticket_i = Sign(ticket_i)$
$signed\_ticket_i$

Fig. 1. Ticket purchase protocol.

  (2) Player obtains a sequence number $s$ of the ticket from the dealer.
  (3) Player computes $H_2(r)$ and $H_1(x\|s\|r)$, then sends $ticket_i = s\|(x \oplus H_2(r))\|H_1(x\|s\|r)$ to the dealer.
  (4) Dealer publishes every single ticket $ticket_i$.
  (5) Dealer returns a signed ticket $signed\_ticket_i$ to player to acknowledge the recipient of player's purchase request. Any digital signature scheme which is existentially unforgeable against adaptive chosen message attack can do and the signature does not need to be encrypted.(Players do not need to enroll to the public key infrastructure as they need to verify the validity of the signature only.)
  (6) Dealer links the ticket to a one-way hash chain. This chain could be created by $chain_1 = H_1(ticket_1)$, $chain_i = H_1(chain_{i-1}\|ticket_i)$ for $i > 1$.
  (7) Dealer publishes $chain_j$ where $j$ is the number of tickets sold so far.
- Winning Result Generation:
  (1) Suppose the final value of the hash chain is $h$, computes $d = D(h)$ by the delaying function, and publishes it.
  (2) Dealer calculates $(w, \pi) = (\texttt{Eval}(SK, d), \texttt{Prove}(SK, d))$.
  (3) If $w > u$, VRF is applied again on $w\|d$ with padding bits if necessary (one possible source of bits is the VRF public key) to determine the winning ticket value.
  Dealer publishes $(w, \pi)$ (and intermediate tuples with number of the times VRF is applied, if VRF is applied more than once) within $t$ units of time after the closing of the lottery session.
- Prize Claiming:
  (1) If $x = w$, player wins.
  (2) Player submits $(s, r)$ to the dealer (in a secure channel, e.g., established by Secure Sockets Layer (SSL) protocol [27]).
  (3) Dealer checks whether a ticket of $s\|(w \oplus H_2(r))\| H_1(w\|s\|r)$ is received.
  (4) If so, dealer checks whether the tuple $(s, r)$ has already been published (i.e., the prize has been claimed by someone already).
  (5) If the prize is not yet claimed, dealer pays the player and publishes $r$.

- Player Verification:
  (1) Player checks whether his or her ticket(s) is (are) included in the hash chain and checks whether the final output of the hash chain $h$ is correct, using the knowledge of $ticket_i$'s.
  (2) Player verifies whether $d = D(h)$.
  (3) Player checks whether $\texttt{Verify}(PK, w, d, \pi) = 1$. (Intermediate $(w, \pi)$ tuples may also be checked if necessary.)
  (4) For each winning ticket published, players verify the validity of $s\|(w \oplus H_2(r))\|H_1(w\|s\|r)$.
  (5) If any mismatch occurs, player should report to the auditing organization by simply providing the parameters involved in the checking.

### 3.3. Design philosophy

The rationales behind some important steps of our scheme are discussed here before we proceed to the security analysis.

#### 3.3.1. Use of VRF

To see how VRF helps in the construction of e-lottery scheme, the following shows some key ideas. The random winning number generation is guaranteed by the pseudorandomness of the VRF and the generation result is verifiable by the commitment made by the dealer (public key). By keeping the private key secret, players are prevented from cheating even if they have access to the VRF's input.

Moreover, we can further distribute the trust on the VRF result generation by the dealer by employing distributed VRF, which can be easily extended from the VRF scheme we discussed [17].

To maximize the security level of the scheme, a new pair of VRF keys can be generated for each lottery session to prevent both the dealer and the players from gaining knowledge of some output values of the VRF.

#### 3.3.2. Use of sequence number

The purpose of using sequence number is not the same as that in [2]. In our scheme, it only serves the purpose of making $H_1(x\|s\|r)$ distinct even if two players pick the same $x$ and $r$. The sequence numbers are not required to follow each other one by one strictly. If the sequence number $s$ is not used, the situation that two different winning players who used the same $r$ may occur (although the probability is not high). In such case, when one of the players claimed the prize, the dealer has already known the value of $r$ and hence the dealer can falsely claim that the prize of the second winning player has been already paid. In our current design, even the values $x$ and $r$ associated with two different winning tickets are the same, the corresponding sequence numbers are different and hence the values of $H_1(x\|s\|r)$ are different.

#### 3.3.3. Adding fair exchange feature

By using VES (or CEMBS), fair exchange between the player and the dealer can be ensured. The fair exchange protocol used in [1] can be adopted in our scheme as well. For the purchase of tickets, if the player aborts the transaction, the dealer can send the cipher cash (an encrypted version of the signature on the message about payment), sequence number $s$ and $signed\_ticket_i$ to the TTP. Then TTP will send the e-cash to the dealer and the $signed\_ticket_i$ to the player. For the prize claiming, if the dealer aborts the transaction, the player can send the cipher cash, $ticket_i$ together with $r$ to the TTP. Then TTP will send the e-cash to the player and $r$ to the dealer.

Since there may be more than one player winning the same prize by betting on the same value $x$, we note that the e-cash used in the prize claiming should not be of fixed value. It should be an agreement of the dealer to award the player a certain prize. By the publicly verifiability of the number of winning tickets, the amount of award shared by each player is guaranteed.

#### 3.3.4. Use of delaying function

The dealer may cheat by trying to "insert" tickets to the hash chain (and "discard" these newly added tickets if the result is not favorable) until a favorable result is obtained. The use of the delaying function prevents this kind of attack. One may argue that if the last ticket is sold much before deadline, then the dealer is able to compute the delaying function twice before deadline. However, it is not so probable in a large scale e-lottery. On the other hand, a malicious dealer can corrupt the late submissions, but this is prevented by the fair exchange protocol[2]. Schemes that do not employ the delaying function, for instance [7], usually assume the integrity of the ticket submission pools is ensured by some TTP. We remark that [1] also mentioned using the delaying function is necessary to make their scheme secure.

#### 3.3.5. Probability of winning

The winning probability of a ticket is $\frac{1}{u}$. Suppose $v = u + b$ is the domain size of the VRF's output. The winning probability ensured by Step 3 of the winning result generation is $\sum_{i=0}^{\infty}\left(\frac{b}{u+b}\right)^i \frac{1}{u+b} = \frac{\frac{1}{u+b}}{1-\frac{b}{u+b}} = \frac{1}{u}$. We can control the domain size of the VRF's output (setting $u \gg b$) so the expected time of application of VRF can be kept small.

#### 3.3.6. Mapping of the winning result to winning ticket

In most of the previous e-lottery schemes, the mapping of the winning result to the ticket value is not discussed in details. There may be a situation that the domain of the generated winning result is not the same as the domain of the ticket value. In particular, if the domain size of the winning result and the domain size of the ticket value are relatively prime, it is not straight-forward how one can

[2] For denial of service attack by the dealer, there are some standard practices to resolve this issue and it is outside the scope of our discussion.

ensure the pre-determined winning probability of each possible ticket value. In our scheme, the mapping of the winning result to the winning ticket value is trivial as it is a 1–1 mapping.

### 3.4. Security analysis

Assuming the lottery is *pari-mutuel*, i.e., the amount of prize is solely based on the sold tickets, our scheme satisfies all the requirements we have identified.

(1) *Random generation of winning number*. The random generation of winning number is guaranteed by the VRF's pseudorandomness. Each ticket submission from the players is random, thus nobody (not even the dealer) can predict the outcome before the time when the last ticket is sold.With the use of the delaying function, the dealer cannot bias the generation of winning number by trying to insert tickets into the hash chain one by one. So, the dealer has no way to make sure that which number or player will win or lose even all the numbers purchased by the players are known.

(2) *The winning result is publicly verifiable*. The winning number generation is verifiable by the provable property of the VRF. Besides, the value of hash chain, which served as the input of the VRF, and the output of the delaying function are also publicly verifiable.

(3) *The total revenue and the number of winning tickets are publicly verifiable*.
   (a) The number of sold tickets is verifiable by the first step of player's verification since each player can check whether his or her ticket(s) is (are) included in the hash chain.
   (b) For each ticket sold, the dealer will publish the corresponding number $x$, so each player can verify the total number of winning tickets easily after the winning result is announced.

(4) *Forgery of ticket is impossible*.
   (a) The dealer cannot forge winning tickets after the winning number has been generated as all the tickets sold have been published and can be publicly verified by all the players.
   (b) Players cannot forge tickets that they have not actually purchased. The hash function $H_1(\cdot)$ makes sure that it is difficult to compute $r$ based on the published ticket $s\|(x\oplus H_2(r))\|H_1(x\|s\|r)$.

(5) *User is not required to be online at the time of winning number generation*. The generation process does not depend on any player.

(6) *Anonymity of players*. Players' identities are not reflected in the process of buying ticket, verification and prize claiming. Players' account number can be encrypted as well to ensure anonymity.

(7) *Confidentiality of the ticket's value*. The ticket's value is encrypted. Neither cutting down a certain ticket value's winning share nor eavesdropping for the "global statistics" is possible.

(8) *Fairness on purchase of tickets and claiming of prize*. Fairness can be achieved by using CEMBS or VES as the protocol used in [1].

(9) *No early registration of player is required*. Our scheme does not require players to register before buying tickets.

(10) *No online trusted third party is assumed*. Our scheme does not require an online trusted third party to satisfy the above security requirements, an offline trusted third party is only needed to resolve dispute for the fair exchange.

### 3.5. Other issues

There are some other minor issues that have been addressed in the previous e-lottery schemes. We briefly discuss how our scheme addresses these issues. For the completeness of discussion, we briefly discuss these issues.

(1) *Total purchase of tickets.* It is natural that total purchase of all tickets guarantees a winning probability of 1. This threat can be resolved with external means, e.g., by setting the prize obtained by total purchase is less than the cost of total purchase. Because the total revenue is publicly verifiable, there is no use for the malicious dealer inserting all possible winning values to the hash chain and succeed to claim tie whenever some one claim to have a prize, since it will greatly increase the revenue and hence the value of prize to be given out by the dealer.

(2) *Unclaimed tickets.* The dealer may not be aware of any unclaimed tickets as the tickets' value are encrypted. A prize claiming period can be imposed, so that any unclaimed tickets presented after the imposed time limit are not entertained. It is a natural solution since it resembles the traditional lottery. Indeed, the player's verification is very efficient (in contrast to the time-consuming verification in [8]) and hence even a temporarily disconnected player would not suffer.

(3) *Duplication of ticket others purchased (or cutting down certain player's potential winning share)*. The tickets are encrypted (by the random $r$ chosen), so duplication of ticket others have purchased (in the sense of choosing the same ticket value) is impossible. Moreover, the submission of ticket buying request is anonymous, no player can intentionally cut down a specific player's potential winning share.

(4) *Other threats. Bogus server*, *system database damage*, *reveal of network address of player*, etc., are not considered as there are other practices to resolve these threats. For examples, the well-studied Secure Sock-

ets Layer (SSL) protocol for server-side authentication [27], standard practices to backup the database and onion routing (e.g. [28]) for anonymous Internet communications.

### 3.6. Efficiency analysis

We analyze the computational requirement of player. For ticket purchase, in addition to the server authentication (which can be done by SSL, for example), players only need to perform one pseudorandom number generation, three hash functions (one is for verification of being included in the hash chain) and one signature verification. It is simply an equality test for players to check whether he or she gets the prize. For prize claiming, players can employ SSL again and encryption can be done by efficient encryption algorithm like the RSA encryption with a low exponent.

For space requirement, in the ticket purchase phase, players only need to store $k$-bit string $r$ and the signed ticket from the dealer, the bandwidth is also minimal as it essentially contains two hash values and a signature only. If the player win, the VES he or she needs to get from the dealer and store temporarily is just having size double than that of a signature. We can achieve short signature (e.g., 150 bits) since the signature encrypted in the VES scheme we discussed is actually a short signature [11]. For the VRF, the analysis in [20] shows that the key $PK$ and the proof $\pi$ only takes 125 bytes each for a 160-bit input.

For verification of the winning result generation, verifying the proof of the VRF takes one point addition, one point multiplication and two pairing operations. It is most computationally expensive to verify the correctness of the delaying function's result. Now we suggest some alternative to remedy this situation.

The first approach is actually similar to what is actually done for tradition lottery, in which we require a number of offline TTPs to verify the correctness of the delaying function's result. Another approach is to ask the dealer to publish a portion of the intermediate result of the delaying function, so that players can randomly choose a subset of these results to verify. We can treat it as a kind of probabilistic cooperative verification. Players who joined the game using devices with low computation power such as mobile devices can choose to trust the offline TTPs and other players helping verification.

From the above analysis, our scheme is practical even for player with mobile device to join the e-lottery game.

## 4. Concluding remarks and future research direction

A new e-lottery scheme, based on various advanced cryptographic techniques including verifiable random function, is proposed. We define a concept of a "fair lottery game over the Internet", with usability, privacy and security concerns. Our scheme satisfies all identified requirements and does not assume the existence of an online TTP for the winning ticket generation.

To make the e-lottery a reality, the way the e-lottery game is conducted should be similar to a traditional one in which players' interaction with the dealer should be kept as simple and minimal as possible. Our scheme is simple and has a high degree of resemblance with a traditional one. Players are neither required to be online at the time of winning ticket generation nor to perform time-intensive operations for checking whether the tickets purchased are indeed winning tickets. Early user registration is not required as our scheme is secure even in the presence of a large size of colluding players. Players are also allowed to buy as many tickets as they want.

A disadvantage of our scheme is the use of the time-consuming delaying function to prevent the dealer from cheating (e.g., by trying to buy each of the possible tickets one by one in order to bias the generation result). To address this issue, we introduce an offline TTP in our e-lottery scheme to perform this verification after the generation of winning result. It is still an open problem to design a publicly verifiable e-lottery scheme without using the delaying function and the online participation of players during the generation of the winning result. Another research direction is to make a delaying function which is efficiently verifiable but the computation is still hard and cannot be parallelized.

There is no doubt that more electronic commerce applications are deployed on the Internet everyday. We hope that the result of this work can help to increase the confidence level and also the interest of customers in participating in e-lottery games. On the other hand, we hope that the technique and the design philosophy involved in this work can stimulate the research of next generation protocols and the new techniques to be applied in the future.

## References

[1] J. Zhou, C. Tan, Playing lottery on the Internet, in: S. Qing, T. Okamoto, J. Zhou (Eds.), Information and Communications Security, Third International Conference, ICICS 2001, Xian, China, November 13–16, 2001, Lecture Notes in Computer Science, vol. 2229, Springer, Berlin, 2001, pp. 189–201.

[2] D.M. Goldschlag, S.G. Stubblebine, Publicly verifiable lotteries: applications of delaying functions, in: R. Hirschfeld (Ed.), Financial Cryptography, Second International Conference, FC'98, Anguilla, British West Indies, February 23–25, 1998, Proceedings, Lecture Notes in Computer Science, vol. 1465, Springer, Berlin, 1998, pp. 214–226.

[3] P.-A. Fouque, G. Poupard, J. Stern, Sharing decryption in the context of voting or lotteries, in: Y. Frankel (Ed.), Financial Cryptography, 4th International Conference, FC 2000 Anguilla, British West Indies, February 20–24, 2000, Proceedings, Lecture Notes in Computer Science, vol. 1962, Springer, Berlin, 2001, pp. 90–104.

[4] C. Hall, B. Schneier, Remote electronic gambling, in: 13th Annual Computer Security Applications Conference, ACM Press, 1997, pp. 227–230.

[5] W. Ham, K. Kim, A secure on-line lottery using bank as a notary, in: Conference on Information Security and Cryptology (CISC) 2002, Korea, 2002, pp. 121–124.

[6] K. Kobayashi, H. Morita, M. Hakuta, T. Nakanowatari, An electronic soccer lottery system that uses bit commitment, IEICE Trans. Inf. Syst. E83-D (5) (2000) 980–987.

[7] E. Konstantinou, V. Liagkou, P. Spirakis, Y.C. Stamatiou, M. Yung, Electronic national lotteries, in: Financial Cryptography, 8th Inter-

national Conference, FC 2004, Key West, Florida, USA, February 9–12, 2004, Lecture Notes in Computer Science, vol. 3110, Springer, Berlin, 2004, pp. 147–163.

[8] E. Kushilevitz, T. Rabin, Fair e-lotteries and e-casinos, in: D. Naccache (Ed.), Topics in Cryptology – CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8–12, 2001, Proceedings, Lecture Notes in Computer Science, vol. 2020, 2001, pp. 100–109.

[9] K. Sako, Implementation of a digital lottery server on WWW, in: R. Baumgart (Ed.), Secure Networking – CQRE (Secure) '99, International Exhibition and Congress Düsseldorf, Germany, November 30–December 2, 1999, Proceedings, Lecture Notes in Computer Science, vol. 1740, Springer, Berlin, 1999, pp. 101–108.

[10] D. Boneh, M. Franklin, identity-based encryption from the weil pairing, in: J. Kilian (Ed.), Advances in Cryptology – CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19–23, 2001, Proceedings, Lecture Notes in Computer Science, vol. 2139, Springer-Verlag, Heidelberg, 2001, pp. 213–229.

[11] D. Boneh, B. Lynn, H. Shacham, Short signatures from the weil pairing, in: C. Boyd (Ed.), Advances in Cryptology – ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9–13, 2001, Proceedings, Lecture Notes in Computer Science, vol. 2248, Springer, Berlin, 2001, pp. 514–532.

[12] S.D. Galbraith, K. Harrison, D. Soldera, Implementing the tate pairing, in: C. Fieker, D.R. Kohel (Eds.), Algorithmic Number Theory, 5th International Symposium, ANTS-V, Sydney, Australia, July 7–12, 2002, Proceedings, Lecture Notes in Computer Science, vol. 2369, Springer, Berlin, 2002, pp. 324–337.

[13] T. Kerins, W.P. Marnane, E.M. Popovici, P.S.L.M. Barreto, Efficient hardware for the tate pairing calculation in characteristic three, Cryptology ePrint Archive, Report 2005/065, available from <http://eprint.iacr.org/>, 2005.

[14] D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and verifiably encrypted signatures, in: E. Biham (Ed.), Advances in Cryptology – EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003, Proceedings, Lecture Notes in Computer Science, vol. 2656, Springer, Berlin, 2003, pp. 416–432.

[15] J. Camenisch, S. Hohenberger, A. Lysyanskaya, Compact e-cash, in: R. Cramer (Ed.), Advances in Cryptology – EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22–26, 2005, Proceedings, Lecture Notes in Computer Science, vol. 3494, Springer, Berlin, 2005, pp. 302–321, also available at Cryptology ePrint Archive, Report 2005/260.

[16] Y. Dodis, Efficient construction of (distributed) verifiable random functions, in: Y. Desmedt (Ed.), Public Key Cryptography – PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings, Lecture Notes in Computer Science, vol. 2567, 2003, pp. 1–17.

[17] Y. Dodis, A. Yampolskiy, A verifiable random function with short proofs and keys, in: Public Key Cryptography – PKC 2005, 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23–26, 2005, Proceedings, Lecture Notes in Computer Science, vol. 3386, 2005, pp. 416–431, also available at Cryptology ePrint Archive, Report 2004/310.

[18] A. Lysyanskaya, Unique signatures and verifiable random functions from the DH-DDH separation, in: M. Yung (Ed.), Advances in Cryptology – CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18–22, 2002, Proceedings, Lecture Notes in Computer Science, vol. 2442, Springer, Berlin, 2002, pp. 597–612.

[19] B. Preneel, The state of cryptographic hash functions, in: I. Damgård (Ed.), Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998, Lecture Notes in Computer Science, vol. 1561, Springer, Berlin, 1999, pp. 158–182.

[20] S. Micali, M.O. Rabin, S.P. Vadhan, Verifiable random functions, in: IEEE Symposium on Foundations of Computer Science, 1999, pp. 120–130.

[21] O. Goldreich, S. Goldwasser, S. Micali, How to construct random functions (extended abstract), in: 25th Annual Symposium on Foundations of Computer Science, IEEE, 1984, pp. 464–479.

[22] S. Micali, R.L. Rivest, Micropayments revisited, in: D. Naccache (Ed.), Topics in Cryptology – CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8–12, 2001, Proceedings, Lecture Notes in Computer Science, vol. 2020, Springer, Berlin, 2001, pp. 149–163.

[23] S. Jarecki, V. Shmatikov, Handcuffing big brother: an abuse-resilient transaction escrow scheme, in: C. Cachin, J. Camenisch (Eds.), Advances in Cryptology – EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2–6, 2004, Proceedings, Lecture Notes in Computer Science, vol. 3027, Springer, Berlin, 2004, pp. 590–608.

[24] O. Goldreich, L.A. Levin, A hard-core predicate for all one-way functions, in: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing (STOC '89), ACM Press, 1989, pp. 25–32.

[25] F. Bao, R.H. Deng, W. Mao, Efficient and practical fair exchange protocols with off-line TTP, in: IEEE Symposium on Foundations of Security and Privacy, 1998, pp. 77–85.

[26] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, in: G.R. Blakley, D. Chaum (Eds.), Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19–22, 1984, Proceedings, Lecture Notes in Computer Science, vol. 196, Springer, Berlin, 1985, pp. 10–18.

[27] A. Freier, P. Karlton, P. Kocher, The SSL protocol Version 3.0, November, 1996. Available at http://www.netscape.com/eng/ssl3/draft302.txt .

[28] R. Dingledine, N. Mathewson, Tor: the second-generation onion router, in: Proceedings of the 13th USENIX Security Symposium, USENIX Association, 2004, pp. 303–320.

**Mr. Sherman S.M. Chow** is currently a Ph.D. candidate in the Department of Computer Science at the New York University. He has obtained his B.Eng in Computer Engineering and M.Phil in Computer Science degrees from the University of Hong Kong. In 2005, he has been a visiting scholar of the Information Security Institute at Queensland University of Technology. He has published a number of papers in the areas of pairing-based, identity-based and group-oriented cryptography, privacy enhancing technologies, electronic commerce applications and anti-malware solutions. He is a member of International Association for Cryptologic Research, and served as program chair of Applied Cryptography and Information Security in 2006.

**Dr. Hui** is the founder and Honorary Director of the Center for Information Security & Cryptography, and concurrently an associate professor in the Department of Computer Science, The University of Hong Kong. Besides actively publishing research papers, he is also involved in several industrial collaboration projects. For instance, he is the Principal Investigator of the Strong Cryptographic Infrastructure for Electronic Commerce project. The technology developed in this project had been used by various government and commercial organizations. Dr. Hui also provides consultancy services for security evaluation of software systems. Dr. Hui received his B.Sc. and M.Phil. degrees in computer science from the University of Hong Kong, and his M.Sc. and Ph.D. degrees in computer science from the University of California, Davis. He is a senior member of IEEE.

**Dr. Yiu** is currently a research assistant professor of the Department of Computer Science of the University of Hong Kong. He obtained his B.Sc., M.S., Ph.D. degrees from the Chinese University of Hong Kong, Temple University, and the University of Hong Kong, respectively. His current research interests include Computational Biology, Data Mining and Computer Security.

**Dr. Chow** began his academic career in the Department of Computer Science, The University of Hong Kong, upon completion of his Ph.D. from the University of California, Santa Barbara. His current research interests are computer forensics, information security and cryptography. He is currently the Associate Director of Center for Information Security and Cryptography. He has involved in the development of several software projects, such as the design and implementation of the computer forensic tool DESK (Digital Evidence Search Kit).