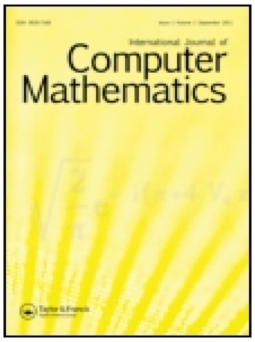

International Journal of Computer Mathematics

ISSN: 0020-7160 (Print) 1029-0265 (Online) Journal homepage: https://www.tandfonline.com/loi/gcom20

Using an efficient hash chain and delaying function to improve an e-lottery scheme

Yining Liu , Lei Hu & Heguo Liu

International Journal of Computer Mathematics
Vol. 84, No. 7, July 2007, 967–970

Taylor & Francis
Taylor & Francis Group

Using an efficient hash chain and delaying function to improve an e-lottery scheme

YINING LIU*†‡, LEI HU§, and HEGUO LIU†

†Faculty of Mathematics and Computer Sciences, Hubei University, Wuhan, China
‡Department of Mathematics, China University of Geosciences, Wuhan, China
§State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing, China

(Received 24 October 2006; revised version received 13 January 2007; second revision received 31 January 2007; third revision received 11 February 2007; accepted 20 February 2007)

A multi-level hash chain structure is employed to improve a fair e-lottery scheme which ensures secure winning number generation and verification and to make verification more efficient. In addition, an implementation of a delaying function based on a secure hash function is presented.

Keywords: e-lottery; Multi-level hash chain; Delaying function

AMS Subject Classification: 94A60

1. Introduction

An e-lottery scheme based on a new cryptographic tool called verifiable random function, which achieves security requirements such as the anonymity of a ticket purchaser and the randomness of a generated winning number, has recently been proposed [1]. The scheme has a compact design, consisting of five sequential phases: Setup, Ticket Purchase, Winning Result Generation, Prize Claiming, and Purchaser Verification. In the Winning Result Generation phase, a linear hash chain is used to chain all tickets and results in a value which ensures that all purchasers can participate in the process of winning number generation. In this paper, we propose an improvement of the scheme by replacing the linear hash chain with a multi-level hash chain. It is shown that the improved scheme reduces the complexity of verification in the final phase.

In addition, based on the output of the hash chain, a delaying function [2] is used to generate a value in the Winning Result Generation phase which is then used as the input to the verifiable random function in order to deduce the winning number. The delaying function is considered to be moderately hard to compute so that the dealer will be unable to forge winning tickets by computing the winning number. However, no practical implementation of a delaying function

*Corresponding author. Email: lyn7311@sina.com

International Journal of Computer Mathematics
ISSN 0020-7160 print/ISSN 1029-0265 online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207160701294426

has been reported to date. In this paper we propose a hash-function-based delaying function with characteristics of parameterization and adjustability of delay time.

2. Multi-level hash chain

We use the same notation as in [1]. A linear hash chain is described as follows:

$$chain_1 = H(ticket_1),\, chain_i = H(chain_{i-1} \| ticket_i).$$

The result is $chain_j$, where j is the number of tickets that have been sold. In the scheme proposed in [1], the hash chain is used to involve all purchasers in the generation of the winning result. However, because of the linearity, the calculation operation needed in the Winning Result Generation phase is also needed in the Purchaser Verification phase, which is efficient for the dealer but troublesome for purchasers.

To simplify the verification, we propose the following multi-level hash chain structure (figure 1):

1. The dealer divides $ticket_i$ into blocks of size T in order:

$$\{ticket_1, \ldots, ticket_T\}\{ticket_{T+1}, \ldots, ticket_{2T}\}\{ticket_{\lfloor (j-1)/T \rfloor T+1}, \ldots, ticket_j, 1, \ldots, 1\}.$$

Pad the last block with 1 if it contains less than T tickets.

2. The dealer computes the first level hash outputs as

$$hash_i = H(ticket_{(i-1)T+1} \| \ldots \| ticket_{iT})$$

and divides $hash_1, hash_2, \ldots, hash_{\lceil j/T \rceil}$ into blocks of size T. If necessary, pad the last block as in step 1 and then hash each of these new blocks and continue constructing the second level.

3. Continue the dividing and hashing process until a final value h is output. Hash values in all levels are published for verification.

With the multi-level hash chain, it is simple to carry out Purchaser Verification, in which a purchaser verifies whether his ticket is involved in a hash chain and whether the hash calculation

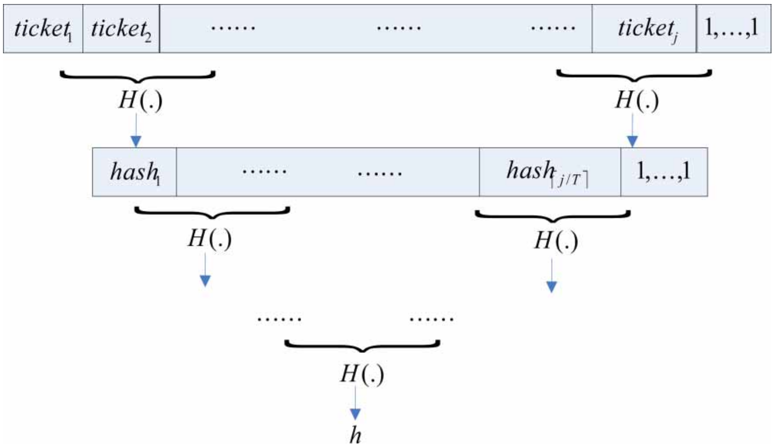

Figure 1. Multi-level hash chain.

is correct. For example, $ticket_i$ is in the $\lceil i/T \rceil$th block in the first level, in the $\lceil i/T^2 \rceil$th block in the second level, and in the $\lceil i/T^l \rceil$th block in the lth level. Since the hash values in each level have been published, the purchaser can directly and thoroughly verify whether his $ticket_i$ is in the corresponding block at each level.

To illustrate the above analysis, assume that $T = 128$ and $j = 1\,000\,000$. A three-level structure is sufficient since $128^3 = 2^{21} > 1\,000\,000$. Assume that $i = 50\,000$. Then $ticket_{50\,000}$ is in the 391st block in the first level, in the fourth block in the second level, and in the first block in the third level. Thus the purchaser of $ticket_{50\,000}$ only needs to verify the hash calculation of these three blocks. If these three verifications pass, his ticket data must be involved in the winning result generation. Further verification for other blocks is not necessary. However, in the scheme in [1] each purchaser has to calculate the hash chain from $ticket_1$ to $ticket_j$.

In summary, the complexity of verification is reduced from $O(j)$ in [1] to $O(\lceil \log_T j \rceil)$ in the improved scheme, which is especially suitable for e-lottery networks with mobile portable terminals.

3. Delaying function

Let $H(\cdot)$ be a cryptographically secure hash function, and define a delaying function $D_L(\cdot)$ with input h and delay parameter L. If the first L bits of $H(h)$ and $H(d)$ are the same (this is the concept of partial collision of hash function in cryptography), d is a valid output of $D_L(\cdot)$. When an output d is found, the function computation halts immediately to make the delaying function a one-to-one mapping. If L is sufficiently large, searching for d will be computationally infeasible because it is hard to find a complete collision for a secure hash function. However, it is very easy to find d when L is small. In view of the relation between L and the complexity of searching for d, the value of L should be chosen to make the computation of the function moderately hard. The parameter L can be adjusted according to the delay time needed.

Since the hash function $H(\cdot)$ is published as part of the system, the output d of $D_L(H)$ can be verified efficiently by all users. Here, we adopt the secure hash function SHA-256 recommended by NIST [3] for$H(\cdot)$.

Even if multiple valid outputs d of $D_L(\cdot)$ are found by the dealer during the valid duration, it does not compromise the fairness requirement of the e-lottery. In the Prize Claiming phase, a purchaser claims a prize if the favourite number x he chose equals the winning number. Since the random number r is kept secret in the Winning Result Generation phase, it is infeasible for the dealer to obtain a favourite number x according to

$$ticket_i = s||x \oplus H_0(r)||H(x||s||r).$$

Therefore it is useless for the dealer to obtain multiple outputs of delaying function for a given input.

Acknowledgements

This research was supported in part by the National Natural Science Foundation of China (Nos. 60573053 and 60672049) and the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20050512002). The authors would like to thank the anonymous referees and the editors for their helpful suggestions.

References

[1] Sherman, S.M.C., Lucas, C.K.H., Yiu, S.M. and Chow, K.P., 2005, An e-lottery scheme using verifiable random function. In: *Proceedings of the International Conference on Computational Science and its Applications, LNCS 3482*, pp. 651–660 (Berlin: Springer Verlag).

[2] Goldschlag, D.M. and Stubblebine, S.G., 1998, Publicly verifiable lotteries: applications of delaying functions. In: *Financial Cryptography (FC'98), LNCS 1465*, pp. 214–226 (Berlin: Springer Verlag).

[3] NIST, 2002, *FIPS 180–2: Secure hash standard* (Washington, DC: NIST, US Department of Commerce).