Electronic Lottery Tickets as Micropayments

Ronald L. Rivest

MIT Lab for Computer Science
(RSA / Security Dynamics)
rivest@theory.lcs.mit.edu

Abstract. We present a new micropayment scheme based on the use of "electronic lottery tickets." This scheme is exceptionally efficient since the bank handles only winning tickets, instead of handling each micropayment.

1 Introduction

We present a paradigm for micropayments: probabilistic payments with "electronic lottery tickets." The probabilistic nature of lottery tickets makes payment of small values simple. For example, an electronic lottery ticket for a $10.00 prize with a 1/1000 chance of winning has an expected value of one cent. A user can pay a vendor one cent by giving the vendor such a lottery ticket.

With conventional payment schemes, a bank or broker must process each payment: the bank issues each digital coin, and processes it again when it is redeemed. *Electronic lottery tickets are the first payment scheme in which the bank does not have to process each payment*, since the bank only sees the "winners." From a bank's point of view, lottery tickets are significantly more efficient than all previously known micropayment schemes.

We assume the reader is familiar with the general notions of public-key cryptography, digital signatures, hash functions, and electronic payment schemes.

The next section introduces the details of electronic lottery tickets; following sections describe a standard implementation and variations, and discuss issues arising from this proposal.

2 Electronic lottery tickets

An *electronic lottery ticket* contains the following items of information (either explicitly or implicitly):

- The name of the *issuer* who created the electronic lottery ticket.
- The name of the *buyer* who is using the electronic lottery ticket as a means of payment. (The buyer may be the same as the issuer.)
- The name of the *recipient* who will collect the payment if the lottery ticket turns out to be a winner. We also call the recipient the "vendor," since the buyer gives the ticket to the vendor to pay for some good or service.

- A *ticket number*, which is compared to the "winning number" to determine if the ticket wins or not.
- A *winning number indicator* that indicates how the winning number will be determined.
- A *ticket face value* that specifies the payment to be received if the lottery ticket turns out to be a winner.
- The name of the *payer* who will make the payment to the recipient if the lottery ticket turns out to be a winner.
- A *ticket credential* that provides the recipient with evidence that the payer will actually pay if the ticket wins.

The electronic lottery ticket is signed or otherwise authenticated by the ticket issuer.

We now discuss and exemplify each of these items for our standard application wherein a user wants to pay a vendor a penny for downloading a web page from the vendor's web site.

The *ticket issuer* in our scenario would typically be the user. In an alternative formulation, the issuer and the payer might be the user's bank, and the user would have purchased the lottery ticket from his bank.

The *buyer* is the user who is surfing the web and and wants to pay for some information or service.

The *recipient* is the vendor who receives the lottery ticket as payment for some information or service provided by the vendor to the buyer.

The *ticket number* might be a short number of, say, three decimal digits. This number is determined by the ticket issuer, and is typically chosen randomly.

The *winning number indicator* can be "external" or "internal":

- An "external indicator" refers to some source or authority who will announce a winning number, such as "the last three digits of the forthcoming Massachusetts state lottery number for (specified date)."
- An "internal indicator," contains no such external reference; an example of an internal indicator is "the last three digits of the 30-digit decimal number w whose MD5 hash value is $h(w) =$ `0x3f...13`." (The value of w is "unique": we don't ever expect to see two w values with the same hash value.) Here the issuer would *not* know w when he issues the ticket; he would know only the hash of w. Someone else would have created the value w and supplied $h(w)$ to the issuer. The recipient would then have to know w in order to determine if he has a winning ticket; this is arranged most easily by having the recipient generate w himself at random in the first place and give the issuer $h(w)$ before the issuer creates the ticket. The issuer, when he sees $h(w)$, should not learn anything useful about the last three digits of w. (The "hash-function" h could be an information-theoretically secure commitment function.)

In any case, the chance of the ticket being a winning ticket must be clear to both the issuer and the recipient, so that they can calculate the expected value of the ticket.

The *face value* (or "prize value") of the lottery ticket would, for our micropayment application, be a modest amount like ten dollars. It should be large enough so that the cost of processing the payment by the bank is small compared to the payment itself. The *expected value* of the lottery ticket is the face value of the ticket times the probability that the ticket will turn out to be a winning ticket.

The *payer* would, in our example application, be the user's bank or credit card company. When the recipient presents a winning lottery ticket to the bank, the bank pays the recipient the face value of the ticket, and charges the user's account that much. In some cases, a bank might be both payer and issuer, and the buyer would purchase the lottery tickets from his bank.

The *ticket credential* might be a signed statement from the payer (bank) that the issuer has an electronic lottery micropayment account in good standing with the bank, and that the bank will pay for winning lottery tickets issued by that issuer during the month of (specify month). This credential might give other terms and conditions, or limitations, on the payer's liability for such lottery tickets, but the net effect is to provide evidence to the recipient that, if the ticket wins, he is likely to receive payment from the payer.

The basic ideas sketched above can be woven into a number of existing payment protocols. In some cases it may add little, in other cases it may greatly reduce processing costs. These ideas are best suited for micropayments, since it is difficult to achieve low-value payments if each and every payment must be separately processed by the user, the vendor, and the bank. Electronic lottery tickets greatly reduce the bank's processing costs, since it sees only the winning tickets. The computational costs to the user and the vendor are comparable to the costs of other payment protocols—they still have to do a little work for each payment.

When a sequence of micropayments is made using electronic lottery tickets, there is a risk to the issuer that too many of them will turn out to be winners, and a risk to the recipient(s) that too few of them will turn out to be winners. But the law of large numbers takes over quickly; it takes many micropayments before you are into "real money." The *expected* value of the payments is correct, and the variance is not large. For example, if an issuer makes 10,000 micropayments with an expected worth of \$100 by issuing lottery tickets with a face value of \$10 and a 1/1000 chance of winning, then the probability that he actually pays less than \$50 is less than 3%, and the probability that he pays more than \$200 is less than 0.4%. Also, when internal indicators of the winning number are used, the protocol may have the vendor notify the issuer immediately whenever a ticket wins, so the issuer can track his micropayment obligations.

This completes the description of the basic scheme. The next sections discuss various details and variations of the scheme.

3 The "standard" version of electronic lottery tickets

We sketch in more detail the "standard" version.

Here the issuer is also the buyer. He has a signed credential from his bank (the payer) stating that his account is in good standing, and that he may issue electronic lottery tickets based on this credential for the month of (specify month). This credential is the ticket credential.

The winning ticket number is determined by an internal indicator, as described above. The recipient makes up a random number w, and gives the issuer $h(w)$ to include in the lottery ticket itself. This hash value is given to the issuer early in the payment protocol. (If the payment protocol must be non-interactive, as for an emailed purchase order, then an external indicator would have to be used.)

The issuer may issue as many electronic lottery tickets as he likes, to whomever he likes, to pay for goods and services. These payments may be viewed as "probabilistic checks" on the user's account with the bank.

There is a risk that the user may issue too many winning tickets, and be unable to pay for the winnings from his account. How should this risk be handled?

In the simplest model, the recipient (vendor) absorbs this risk; the vendor may not get paid if the issuer lacks sufficient funds with the payer. In this case the payer will not re-certify the issuer for the next time period. This model works reasonably well when the items being sold are low-value information items, so that the vendor has lost little. While some fraction of the winning electronic lottery tickets will turn out to be worthless, the system is self-correcting in that bad issuers will get weeded out over time. A vendor may be perfectly happy if he can collect on 90% of his winning tickets. Over time, he can adjust prices to cover such lossage. It is not worthwhile for the vendor to do an on-line verification for each micropayment received. On the other hand, he can determine immediately if ticket wins (assuming it has an internal indicator), and deposit that ticket immediately, receiving notice if the issuer's account has insufficient funds. Or, the vendor could verify on-line the issuer's good standing with the payer when the vendor receives his *first* electronic lottery ticket from an issuer, and thereafter just deposit the winners. (This is reminiscent of the techniques of Jarecki and Odlyzko [4] for "probabilistic polling".)

There is also some risk to the user that he might be issuing too many winning tickets, and so run up a large bill for his "micropayments." As noted above, the chance of this happening are small, and the users of electronic lottery tickets should be aware that there will be some variability in the actual amount they owe, even though the expected amount paid is correct. Furthermore, a user can be notified immediately by the recipient whenever he issues a winning ticket (assuming internal indicators), so he can track his actual expenditures. For every 1000 penny web pages visited, a window could pop up explaining that the user has just been billed $10 for micropayments.

Since the issuer may create as many lottery tickets as he wishes, it is appropriate (and necessary) that he take on the obligation of paying off all winning tickets.

My favorite version of this scheme is based on the micropayment scheme PayWord by Adi Shamir and myself [6]. A user commits to a given vendor a

"payword chain" consisting of values

$$x_0, x_1, x_2, \ldots, x_n$$

where

$$x_i = h(x_{i+1}) \quad \text{for } i = 0, 1, \ldots, n-1,$$

by signing a message containing the root value x_0. Each successive payment is made by releasing the next consecutive value in the sequence, which can be verified by checking that it hashes to the previous element. This ensures that a sequence of payments can be made to a vendor where the buyer has to make only one digital signature (for the commitment).

We now sketch a new micropayment protocol where the payword chains become a sequence of electronic lottery tickets. The word w initially given from the vendor to the buyer (and included in the buyer's commitment message) is now the root $w = w_0$ of another payword chain

$$w_0, w_1, w_2, \ldots, w_n \cdot$$

Then the i-th ticket in the buyer's chain, with value x_i, is a winning ticket if and only if (x_i mod 1000) is the same as (w_i mod 1000). In this way, the vendor can immediately inform the buyer when x_i wins by revealing w_i, without giving the buyer information that would allow him to determine if future tickets in the chain will be winners.

4 The bank as issuer

In a variation, the bank would issue the tickets, or authorize a batch of tickets made up by the buyer (say by using Merkle's tree-authentication scheme). The buyer would then transmit individual tickets to a vendor for specific payments.

This scheme doesn't add much to the standard scheme. There is a risk that the buyer will give the same ticket to more than one vendor, and so "double spend." This is not so different from having the buyer issue several tickets in the standard scheme. Furthermore, the buyer still has to sign the ticket over to the recipient.

5 External indicators

If the winning number indicator is external, then the issuer may issue "store and forward" electronic lottery tickets which may be included in email; he does not need the recipient to give him the value $h(w)$ that fixes the winning number.

If the bank creates the winning number, then there is a risk that the bank and the issuer will collude to defraud the vendor. (If the bank creates the winning number early and shares it with the issuer before the issuer creates the tickets, the issuer can choose ticket numbers he knows not to be winning numbers.) This issue arises if the creator of the winning number is anyone that might be

in collaboration with the issuer. Thus, best way to define daily winning numbers externally might be to hash together data from several independent sources (stock market data, state lottery numbers, exchange rates, NYT crossword puzzle solutions).

A problem with external indicators is that vendors must store all electronic lottery tickets they have received, until the daily winning number is revealed.

Another problem is that the buyer does not get immediate feedback on whether the lottery tickets he issues are winning ones or not.

6 Efficiency considerations

From the bank's point of view, electronic lottery tickets are extraordinarily efficient. The bank has only to (a) provide monthly electronic credentials to customers with micropayment accounts in good standing, and (b) pay off winning lottery tickets issued by that customer from that customer's account.

The bank does not need to perform any actions corresponding to the "withdrawal" portion of standard electronic coin schemes. The monthly credential-issuing is all that needs to be performed.

Similarly, the bank does not need to perform any processing to handle each and every micropayment. Instead of handling 1000 micropayments each worth one penny each, the bank handles a single winning lottery ticket worth \$10. This "probabilistic aggregation" of many small payments into a few winning lottery tickets is the essential reason why electronic lottery tickets are so efficient.

From the customer's point of view, his workload is comparable to that of a standard digital coin scheme. There is no "withdrawal" protocol, but an electronic lottery ticket has to be issued for each micropayment. With standard techniques (such as those used by PayWord), the number of public-key operations can be kept very small; the user needs to sign just one message for each vendor he contacts. While the amount he actually pays varies, it is usually close to being right, and the bank fees for micropayment service are likely to be much better than for other micropayment schemes.

From the vendor's point of view, the scheme is also very efficient. He does not store every micropayment received, but only the winners. The computational load is comparable to that of other micropayment schemes. There is a risk that some winners will turn out to be worthless, but by dealing only with issuers with credentials from reputable banks, his losses are likely to be small. (Banks would typically require that a deposit be made, or a credit limit established, before issuing a micropayment credential.)

7 On the legality of electronic lottery tickets as micropayments

The electronic lottery ticket micropayment scheme makes use of "probabilistic payments" to effect micropayments very efficiently.

While I have used the term "electronic lottery tickets" as a pedagogic device in explaining the scheme, I should emphasize that there are some real differences between this scheme and ordinary lotteries, which are usually highly regulated.

First, note that in these schemes the role of buyer and seller are turned around, because a lottery ticket is used as a *payment* rather than as an *item being bought.* The typical user *issues* a lottery ticket and incurs an obligation to pay it off if it wins, rather than *purchasing* a lottery ticket and hoping to receive money if it is wins.

Second, the face values are considerably smaller here (say \$10) than is usual for typical lotteries (which may run into the millions of dollars).

Third, the "purchaser" of lottery tickets here (whom we have been calling the "vendor") does not pay for them with money, but with information. No one is going to make money by running a lottery of our sort if all they are receive in return is access to web pages. The way to circumvent this claim would be to have side payments by the vendor to the issuer for the lottery tickets. Perhaps the simplest approach legally would be to restrict lottery *payments* by individuals to some modest amount (say \$1000 per year) to cover their micropayment needs, without enabling them to run a large-scale lottery.

8 Privacy and Anonymity

Compared to ordinary credit-card payments or non-anonymous digital cash, electronic lottery tickets improve a user's privacy tremendously, because very few of his transactions are reported to the bank. Thus, the bank learns about only a few of the vendors that the user contacts.

True anonymity of the user from the vendor is more costly, as it requires an intermediary (the vendor needs to have routing/delivery information for the goods sold). That third party could also intermediate the payments for the user—the user pays the intermediary, and the intermediary pays the vendor. This works for electronic lottery tickets as for any electronic payment scheme. Here it is simplest if the vendor reveals the winning number w after he receives his lottery ticket, so that with slight protocol changes the intermediary's ticket to the vendor can be made winning if and only if the user's ticket to the intermediary is winning.

Another way of achieving such anonymity from the vendor is for the issuer's credential to have no identification of the issuer other than a pseudonym known to the issuer and the bank, and to run the payment protocol as usual, except that an intermediary is used to hide the issuer's network address from the vendor.

9 Conclusions and Discussion

We have presented a micropayment scheme, based on "electronic lottery tickets," which is exceptionally cost-effective and provides enhanced user privacy.

Postcript

The idea that bets, gambles, or lotteries can be equivalent to fixed payment is not new. Indeed, this notion is a basic theme in the foundations of probability, utility theory, and statistical decision theory. See, for example, the 1947 treatment by von Neumann and Morgenstern [10], the 1972 treatise by Savage [9], or Berger's text [2] on statistical decision theory.

A preliminary version of this paper was distributed to several colleagues, who pointed out the following relevant references and prior art. Making payments by gambles has been described by science fiction write Poul Anderson [1]. Making payments by probabilistic methods with the correct expected value is the subject matter of two patents [7,8] by Michael Rossides, and is discussed in a note in the *Economist*[3]. Most recently, David Wheeler [11] has also proposed "transactions using bets." With respect to this prior art, the present note makes contributions by emphasizing the utility of probabilistic payments in the micropayment arena, by carefully distinguishing the roles of issuer and payer, and by extending the PayWord mechanism to enable probabilistic payments.

Acknowledgments

I'd like to thank Ross Anderson, Andrew Odlyzko, Bruce Schneier, and David Wheeler for helpful comments and pointers.

References

1. Poul Anderson. *Inside Straight.* Appears in collection "*7 Conquests*". (Collier, 1970).
2. James O. Berger. *Statistical Decision Theory: Foundations, Concepts, and Methods.* (Springer 1980).
3. Heads I win, tails you lose. *The Economist* (June 13, 1992), 84.
4. Stanislaw Jarecki and Andrew Odlyzko. *An efficient micropayment system based on probabilistic polling.* Proceedings 1997 Financial Cryptography Conference (Springer, 1997).
5. Ronald L. Rivest. The MD5 message-digest algorithm. Internet Request for Comments, April 1992. RFC 1321.
6. Ronald L. Rivest and Adi Shamir. *PayWord and MicroMint: Two simple micropayment schemes* `http://theory.lcs.mit.edu/~rivest/publications.html` (1996).
7. Method of using a random number supplier for the purpose of reducing currency handling. U.S. Patent 5085435. Issued 2/4/1992.
8. Expected value payment method and system for reducing per unit costs of paying and/or receiving a given amount of a commodity. U.S. Patent 5269521. Issued 12/14/93.
9. Leonard J. Savage. *The Foundations of Statistics*, second edition. (Dover, 1972).
10. John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior* (second edition). (Princeton University Press, 1947).
11. David Wheeler. Transactions Using Bets. *Security Protocols* (ed. Mark Lomas), Lecture Notes in Computer Science no. 1189 (Springer, 1996), 89–92. (Also available by ftp from the server `ftp.cl.cam.ac.uk` as `/users/djw3/tub.ps`.)