Peppercoin Micropayments

Ronald L. Rivest

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
rivest@mit.edu

Abstract. We present the "Peppercoin" method for processing micropayments efficiently. With this method, a fraction of the micropayments received are determined, via a procedure known as "cryptographic selection," to qualify for upgrade to a macropayment. The merchant deposits the upgraded micropayments as macropayments, and merely logs locally the non-qualifying micropayments. In this manner, the merchant transforms a large collection of small micropayments into a smaller collection of macropayments, of the same total expected value. The merchant pays much less for processing the resulting macropayments, since there are fewer of them. Consumers are billed for exactly the amount they spend, based on auxiliary information recorded in each micropayment. The method is highly secure, and compatible with existing payment mechanisms such as credit cards.

1 Introduction

The Peppercoin micropayment system is due to Micali and Rivest [MR02]; we refer the reader to their original paper for more details. Here we give only a high-level description of the method.

For this paper, a "micropayment" is any payment that is small enough that processing it is relatively costly, as a percentage of the overall transaction value. Given that typical credit card processing fees may be twenty-five cents per transaction, we may consider a "micropayment" to be any payment under $10.

We view the introduction of efficient micropayments into the world of internet e-commerce as potentially as significant as the invention of metal coins by the Lydians in 640 B.C. Coins turned out to be a signficant market enabler—the first retail markets evolved soon thereafter.

Today, it is clear that small electronic payments will soon become commonplace. Not only to pay for music downloads (note the recent success of Apple's iTunes), but also for other digital downloads and other

digital goods and services. Small electronic payments will begin to replace metal coins and small paper bills for real-world purchases as well.

Efficient processing is essential for a successful micropayment method; it makes no sense to charge twenty-five cents to process a ten-cent payment!

Other important factors include ease-of-use, security, and compatibility with the existing payment infrastructure.

2 Aggregation Methods

The key to efficient processing of micropayments is, of course, the *aggregation* of many small micropayments into a few larger macropayments. We distinguish four levels of potential aggregation, of increasing efficiency.

2.1 No Aggregation

When no aggregation is done, each payment, no matter how small, makes its way around the entire payment cycle, from purchaser to merchant to merchant (acquiring) bank to consumer (issuing) bank to consumer (for billing). [Or the equivalent, depending on whether this is a debit or credit system.] Having every party touch every payment in this manner is extremely inefficient and costly. Chaum's Digicash system [Cha83] is an example of a payment system with no aggregation. (To be fair, the emphasis of his design was on anonymity rather than efficiency for micropayments.)

2.2 Session-level Aggregation

With session-level (also known as merchant-level) aggregation, the merchant collects several small payments from a given consumer—say all of those spent by that consumer during one day—and submits a macropayment representing the aggregate amount due at the end of the day or session. This method only works sometimes: when the consumer makes repeated small purchases at the same merchant within a short time period; it doesn't work in general. PayWord [RS97] is an example of a method based on session-level aggregation.

2.3 Aggregation by Intermediation

Another approach to provide aggregation is to create a new intermediary that all consumers and merchants must interact with in order to process

micropayments. This intermediary attempts to keep track of all micropayments made by each consumer at any participating merchant, and then submit for processing by the ordinary banking system only payments that represent the entire amount spent by that consumer during the given time period, or the entire amount to be received by a given merchant during that time period. This approach still requires handling of each payment by the intermediary, who is tasked with replicating the functionality of the entire existing banking system, at lower cost! Clearly, the way towards success should be by *reducing* the amount of mechanism and processing involved, not by *increasing* it!

2.4 Universal aggregation

The "Peppercoin" method uses *universal aggregation*, which we sometimes also call *cryptographic selection* or *many/many/many* aggregation, since it aggregates smoothly across many consumers, many merchants, and many payment service providers.

With universal aggregation, each participating merchant processes micropayments directly, using special cryptographic software.

The software first checks the validity of the micropayment—for example, by checking the consumer's digital signature on that micropayment. If the micropayment is valid, the merchant then completes the transaction and delivers the purchased goods to the consumer.

The software then checks whether the micropayment "qualifies for an upgrade" (to a macropayment). If the micropayment does not qualify for an upgrade, the merchant merely logs the micropayment, and need do no further processing. If the micropayment does qualify for an upgrade to a macropayment, then the merchant will deposit that micropayment *as a macropayment* with his bank.

The size of the resulting macropayment will be a fixed system parameter, such as $10 or $20.

The fraction of micropayments that qualify for such an upgrade depends on the size of the micropayments. For example, for ten-cent micropayments and a $10 macropayment size, approximately one in every 100 micropayments will qualify for such an upgrade to a $10 macropayment.

It is easy to see that the merchant expects to receive the same amount on the average, since one hundred ten-cent micropayments has the same net value as one ten-dollar macropayment.

Indeed, the merchant should be very happy with such a procedure, since he is now only paying a single transaction processing fee (for the

ten-dollar macropayment) instead of one hundred transaction processing fees (for each of the micropayments). Universal aggregation turns what was probably a money-losing proposition into a profitable operation for the merchant!

The qualification procedure is cryptographic in nature, so that neither the consumer nor the merchant can affect the decision as to whether a particular micropayment will qualify for upgrade. The qualification procedure depends upon the merchant's digital signature on data derived from the micropayment, so that other parties, such as the merchant's and consumer's banks, can check that a given micropayment did indeed qualify for upgrade.

While one may loosely think of the qualification procedure as selecting a given micropayment for upgrade "with a certain probability," the qualification procedure is in fact deterministic and not randomized—the merchant's digital signature method will be a deterministic signature method.

It is important to note that each micropayment is tested for qualification for upgrade *independently* of each other micropayment. The merchant does not need to keep any sort of records of previous transactions, cumulative amount spent by each consumer, or the like; this simplicity permits very elegant and clean implementations for the merchant.

When a particular micropayment qualifies for an upgrade to a macropayment, and is turned in for a $10 deposit by the merchant, who pays the merchant the $10? It may be awkward to bill the consumer, since it may be his very first micropayment.

The Peppercoin system ensures that a consumer is never billed for more than he has spent. This is accomplished by having each micropayment indicate the total cumulative value of the consumer's expenditures to date. The consumer's bank sees these values every so often, when it sees micropayments from the consumer that have been upgraded to macropayments, and can thus incrementally bill the consumer appropriately. The consumer's bank thus acts as a financial "buffer" between the cash outlays to merchants and the receipts from the consumer, in a manner very similar, but not identical, to what happens with standard credit card processing. The cryptographic nature of the qualification process ensures that the cash flows of the consumer's bank will (almost exactly) balance, as they grow in value. The use of cryptography—based on digital signatures by consumers and merchants—also prevents various forms of fraud of one or two parties against the other(s).

Universal aggregation excels for processing micropayments, since the micropayments exist as such only in the hands of the consumers and

merchants, who are in any case involved with other transaction details as well. Deposits made by the merchant are solely in the form of macropayments. Consumers are only billed for the amount they have spent at *all* merchants over the billing period (e.g. one month), which will not be a micropayment (for most consumers). There is no intermediary involved who has to handle every payment. Thus, universal aggregation provides a clean and simple way to extend an existing payment system, such as a credit-card system, to the realm of micropayments.

3 Other issues

3.1 Ease of use

The basic Peppercoin method can be implemented in a variety of ways, to maximize ease-of-use for the consumer in a given situation. For example, while the basic Peppercoin method requires that each consumer have digital signature capability, one can easily eliminate this requirement by having a party trusted by the consumer sign the payments for him as a proxy; this might be a natural approach in a web-services environment.

The Peppercoin method can also be implemented so that it feels to the consumer as a natural extension of his existing credit-card processing procedure, further increasing consumer acceptance and ease-of-use.

3.2 Scalability

The Peppercoin micropayment system scales easily to very large implementations, since all of the "real work" involving micropayments is handled by the consumer and merchant directly, and since the system works naturally with a variety of financial institutions representing the consumer and the merchant.

3.3 Non-interactivity

The Peppercoin method is *non-interactive* in the sense that a Peppercoin micropayment can be emailed or transmitted directly from consumer to merchant. There is no need for the merchant to interact with the consumer during the payment process, or even to be on-line at the time of the payment.

This non-interactivity means, for example, that Peppercoin micropayments could conceivably be used in applications such as spam-prevention, where it has often been proposed that spam could be reduced by requiring a micropayment with each email sent.

3.4 Low-cost qualification procedures

The simple universal aggregation method described above requires the merchant to compute a digital signature for every micropayment received. For some merchants, who are processing a very high volume of very low-priced goods, this may be a bit of a burden.

It is possible to reduce this computation cost considerably, by modifying the qualification procedure slightly. For example, it may depend only on the merchant's signature on the time of the micropayment, measured to the nearest minute. Then the merchant need only compute one digital signature per minute. A different approach to reducing computation time can be based on having the server compute a "Merkle tree" [Mer79] to hash together many micropayments, and then compute a digital signature on the root.

3.5 Variable-sized payments

Although this point may already be clear, we emphasize that the Peppercoin micropayment system handles micropayments of varying sizes in a smooth and efficient manner. The only relevant factor is the ratio between the macropayment size and the micropayment size. For example, if macropayments are ten dollars and a micropayment is ten cents, this ratio is one-hundred; in this case the qualification procedure ensure that one out of every one hundred ten-cent micropayments, on the average, qualifies for an upgrade to a macropayment. Thus, as an additional example, one-dollar micropayments would qualify for upgrade one out of every ten times, on the average, to a ten-dollar macropayment.

3.6 Revenue variance

The merchant will see a dramatic reduction in his costs for processing transactions, since he is requesting processing for a small number of macropayments instead of a large number of micropayments.

But the merchant may worry that the qualification procedure might leave him nonetheless somehow at a disadvantage, since during a given period a unusually small number of micropayments might qualify for upgrade.

Fortunately, this worry is easily determined, with a little analysis, to be a non-issue. The cost-savings provided by Peppercoin, which provide a benefit to the merchant on *each and every* transaction, and which grow cumulatively in value as he processes more transactions, are going

to overwhelm any "jitter" in the qualification decisions, which are unbiased. The following theorem is one example of such analysis, comparing a Peppercoin implementation which charges a fee of pT for processing each transaction of value T, versus another system that charges qT for processing each transaction of value T.

Theorem 1. *If a Peppercoin implementation which charges a fee of pT for processing each transaction of value T, while another system that charges qT for processing each transaction of value T, then once the total number of macropayments (qualifying micropayments) exceeds*

$$(5/(q-p))^2$$

the probability is 999,999 out of 1,000,000 that the merchant's net total receipts will be higher with Peppercoin than with the other system.

As an example, consider a scenario where a Peppercoin-based system offers to process ten-cent payments for a penny each (i.e., $p = 0.1$; quite feasible with Peppercoin), while competitor C offers to process them for three cents each (i.e., $q = 0.3$; very hard to achieve without using a selection procedure such as Peppercoin's). Thus, $q-p = 0.2$, and the merchant will almost surely be ahead with Peppercoin after only $(5/0.2)^2 = 625$ macropayments. This is a rather worst-case estimate, and the merchant is likely to be ahead with Peppercoin from the start.

4 Summary

The Peppercoin universal aggregation method for processing micropayments offers low-cost processing, even for very small payments, with a high-degree of security. It can be implemented in an easy-to-use manner that extends existing payment mechanisms.

More details can be found on the Web [Pep,Riv].

References

[Cha83] David Chaum. Blind signatures for untraceable payments. In R. L. Rivest, A. Sherman, and D. Chaum, editors, *Proc. Crypto '82*, pages 199–203, New York, 1983. Plenum Press.

[Mer79] Ralph Charles Merkle. *Secrecy, Authentication, and Public Key Systems*. PhD thesis, Stanford University, June 1979. Technical Report 1979-1; Information Systems Laboratory.

[MR02] S. Micali and R. L. Rivest. Micropayments revisited. In B. Preneel, editor, *Proc. Cryptography Track at RSA Conference 2002*, pages 149–263. Springer, 2002. Lecture Notes in Computer Science No. 2271.

[Pep] Peppercoin web site. http://www.peppercoin.com.

[Riv] Rivest web site. http://theory.csail.mit.edu/~rivest.

[RS97] Ronald L. Rivest and Adi Shamir. PayWord and MicroMint–two simple micropayment schemes. In Mark Lomas, editor, *Proceedings of 1996 International Workshop on Security Protocols*, volume 1189 of *Lecture Notes in Computer Science*, pages 69–87. Springer, 1997. (Also available in CryptoBytes, volume 2, number 1 (RSA Laboratories, Spring 1996), 7–11, and at http://theory.lcs.mit.edu/~rivest/RivestShamir-mpay.pdf).