Originally published as pp. 303-306 of Coulouris, Dolllimore and Kindberg, Distributed Systems, Edition 3, 2001.

Low-value electronic transactions: the Millicent protocol

A secure Internet payment system is required that is cost-effective and convenient for the purchase of low-priced services and access to electronic resources at prices in the range 0.1–100 cents, for example to charge users for web page access, email transmissions or Internet phone calls. Most existing payment schemes such as credit card transactions are too expensive for these purposes.

Here we describe the solution offered by the Millicent scheme [Glassman *et al.* 1995]. The scheme employs the simple form of secret-key-based digital signature to reduce its computational cost. Encryption is used only where privacy is required.

Existing payment systems and their drawbacks ◊ The authors of the Millicent protocol have summarized the drawbacks of the payment methods currently available as follows. All of them will require cryptographic security, their computing and communication costs vary, but they are significant in all cases.

Credit cards: The problem of using credit cards for small transactions is the high transaction cost resulting from the need to interact with the card company's central system, which is exacerbated by the additional costs incurred to secure electronic transactions and by other features such as the preparation of statements given to customers.

Customers maintain accounts with vendors: Customers need to establish an account with a vendor ahead of the first transaction. Transaction costs are then low, but the initial overhead tends to discourage casual transactions. A vendor must maintain the customer's entry in an account database over a fairly long period.

Aggregation of transactions: When a customer makes several purchases the vendor can keep records of them and bill them at the end of a period. This is similar to the maintenance of accounts but may save some of the setup costs. The vendor must keep the records even for customers who make a single purchase, and the cost of this may exceed the revenue.

Digital cash: Like conventional cash, digital cash – i.e. digital tokens of value, issued by a bank or broker – should offer an efficient means of paying for small transactions, but developers of digital cash must solve the problem of *double spending*. Double spending is a consequence of the fact that the holder of a digital token can make any number of undetectable copies. Therefore, tokens must be uniquely identified and they must be validated as unspent at the time of use. The challenge is to devise a validation scheme that is scalable, reliable and cost-effective in all circumstances.

The Millicent scheme ◊ The Millicent project has developed a scheme for the secure distribution and use of *scrip* – a specialized form of digital cash that is designed for use in low-value transactions. Millicent is of interest here because it employs several of the security techniques described in this chapter with an outcome that is quite different from those of SSL and other security systems.

Scrip is a form of digital cash that is valid only for a specific vendor. It is designed to offer the following features:

- it has value only at a specific vendor;
- it can be spent only once;
- it is tamper-resistant and hard to counterfeit;

- it can be spent only by its rightful owner;
- it can be produced and validated efficiently.

The design of the Millicent scheme is scalable because each vendor’s server is responsible only for validating the scrip that it has issued. Customers can acquire scrip directly from the vendor, or from a broker who holds scrip for many vendors, by performing a conventional transaction.

Scrip, the vendor-specific currency introduced by Millicent, is represented by digital tokens with the following format:

Vendor	Value	Scrip ID	Customer ID	Expiry date	Properties	Certificate

The *properties* field is available for uses determined by the vendor – it might for example include the customer’s country or state of residence, so that the appropriate tax rate can be applied. The *certificate* is a digital signature protecting all the fields in the scrip against modification. The signature is produced by the MAC method described in Section 7.4.2. The purpose of the remaining fields will emerge in the following description.

Scrip is generated and distributed by *Brokers* – brokers are servers that deal in scrip in bulk, relieving customers and vendors of some of the overheads involved. Brokers exchange scrip for real cash, buying scrip (or the right to generate scrip) from vendors at a discount and selling scrip to customers against credit card or other payments. Customers may buy scrip for several vendors from a single broker, aggregating the charges and paying at the end of a period.

Scenario ◊ This is how an electronic purchase transaction proceeds using scrip: a customer Alice is interested in buying a small product or service (for example a phone call or a web page) from a vendor Venetia. Alice may already hold scrip suitable for transactions with Venetia that is left over from previous transactions; if not, Venetia gives her the URL of a broker, Bob, who deals in Venetia’s scrip (we’ll call it *V-scrip*). Alice buys some V-scrip from Bob.

Alice then sends Venetia a purchase request, attaching a piece of V-scrip with a value large enough to cover the purchase that she wants to make. Venetia validates the scrip – checking that its Scrip ID isn’t in her list of scrip that has already been spent and that the customer ID in the scrip is Alice’s ID. If the value is greater than the payment required, Venetia makes a new piece of scrip to cover the difference and sends it to Alice as change.

Brokers can obtain vendor scrip in several ways. In the simplest case, Venetia manufactures scrip and sells it to Bob at a discount in a bulk transaction. Alternatively, Venetia licenses Bob to manufacture scrip on her behalf. In this model, Venetia is unaware of the scrip until she receives it in payment from customers. From time to time she contacts Bob and sends him the identifiers of the scrip that she has received from customers and Bob pays her a discounted price for it.

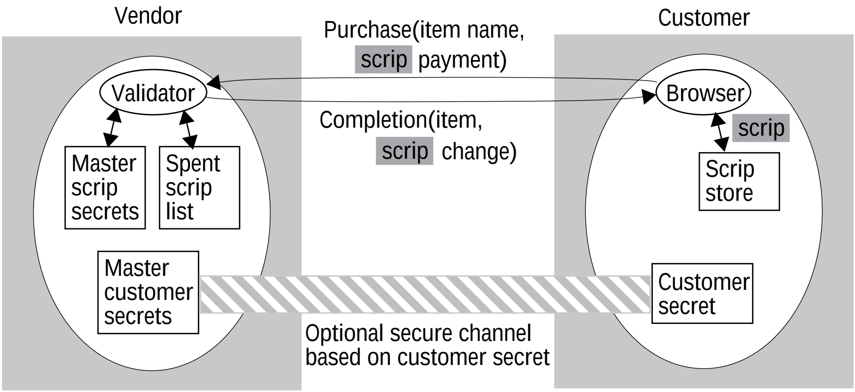
Goals ◊ Millicent was designed as an electronic cash system that avoids the communication overheads and the delays associated with a centralized system, while providing security adequate to prevent fraudulent use. The security of scrip is based on digital signatures; encryption is used for secrecy where necessary. Low-cost cryptography is used because there is no point in making the cost of breaking the security of the system greater than the value of the transactions for which it is used.

The Millicent scheme is free-standing. It can be implemented without support from other services, such as a secure name service, because it does not depend upon an externally authenticated identity for customers or vendors. Instead, Millicent generates and distributes unique secret keys for use by the parties to scrip transactions.

Implementation ◊ Figure 0.1 gives an overview of the architecture. We have mentioned that scrip is signed to protect it against alteration or forgery. The signature contained in the certificate attached to each piece of scrip is made when the scrip is generated, using a *master scrip secret* as the signing key. The signing method is exactly the same as the MAC signature method described in Section 7.4.2. The key is appended to the other fields of the scrip, and a secure digest function such MD5 or SHA is used to produce a 128-bit digest, which becomes the certificate.

Before generating any scrip, a vendor (or a broker licensed by a vendor) will generate a vector of 64-bit master scrip secrets. One such secret would be sufficient, but the availability of

Figure 0.1 Millicent architecture



several enables a new one to be selected from time to time to guard against its disclosure by accident or in a successful attack.

Each piece of scrip has a unique identifier. Its value includes an 8-bit *master secret index,* which is used to select a master secret from the vendor's vector of master secrets. The master secrets are retained by the vendor and one is used to produce a certificate for each piece of scrip generated

Several variations of the Millicent protocol are suggested, offering different levels of security. In all of them the vendor must validate each item of scrip submitted by customers, and we describe that step first.

To validate a piece of scrip:

1. The vendor checks that it is not forged or tampered with by generating a check-signature using the relevant master secret (using the master secret index from the scrip ID to select it from the vector of master secrets) and compares it with the signature in the certificate.
2. The vendor checks that the scrip has not already been spent. To do this, she maintains a list of the IDs and expiry dates of all the pieces of scrip she has issued, with an indication as to whether it has been spent. The expiry date field enables the vendor to delete scrip that has passed its expiry date from the list, thus preventing it from growing everlastingly. Customers must exchange old scrip for new before the expiry date passes.

Transactions such as that described in the scenario above can be performed securely based on validation alone. This protects the vendor against forgery and double spending, but it does not protect the customer against the theft of her scrip, nor does it provide any confidentiality for the customer or the vendor.

Protection against theft requires the vendor or broker selling scrip to maintain a vector of *customer master secrets* (selected using a portion of the customer ID) and to issue each customer with a *customer secret*. The customer secret is constructed by appending the customer master secret to the customer ID and applying a secure hash function such as MD5 to the result. The customer secret must be transmitted from the vendor to the customer through a secure channel. The use of SSL for the initial purchase of scrip is recommended for this purpose.

Once it has been transmitted to the customer, the customer secret can be used as a shared secret key between the customer and vendor. The customer can use it for signing transactions, and both the customer and the vendor can use it as an encryption key when confidentiality is required.

To prevent theft, transactions are signed by the customer, using her customer secret, and the vendor checks that the customer ID in the scrip matches the customer ID associated with the customer secret. If not, then the scrip is being spent by someone other than the customer to whom it was sold – in other words, it was stolen.

To provide confidentiality, a customer sends her customer ID to the vendor and they establish a secure channel using a secret-key encryption algorithm with the customer secret as the encryption key. The secure channel can then be used to complete a transaction in secret.

We have described Millicent in some detail because it provides a real-world example illustrating the application of many of the security techniques described in this chapter. The Millicent system is one of several electronic cash schemes that have been developed for use in electronic commerce. It was originally developed at the Digital Systems Research Center, Palo Alto, California.

References

Glassman et al. 1995 Glassman, S., Manasse, M., Abadi, M., Gauthier, P. and Sobalvarro, P. (1995). The Millicent Protocol for Inexpensive Electronic Commerce. Fourth International WWW Conference, December 1995.