PRIVACY PROTECTED PAYMENTS - REALIZATION OF A PROTOCOL THAT GUARANTEES PAYER ANONYMITY

Svein J.Knapskog
Division of Computer Systems and Telematics,
University of Trondheim, The Norwegian Institute of Technology
N-7034 Trondheim

Introduction

There is a growing consern that the total traceability of users in a conventional electronic card based payment system may become a major argument against these new, more convenient and more cost effective systems. To circumvent this problem, electronic card (smart card) based systems can still be used, but in connection with new data communication protocols involving banks, shops and customers (of banks and shops). Some new ideas regarding use of "electronic coins" will have to be accepted, also.

The basic idea for this new way of using known systems and assets is first presented by David Chaum at CWI, Amsterdam <1>. It is based upon the usage of home terminals (personal computers) and POS - terminals in the different shops, much in the same way as we already are exposed to and getting familiar with in our everyday life today. This new concept, however, will be dependent upon a smart card with an order of magnitude more memory available on it than todays technology permits, and in addition it will rely heavily upon online data communication between shops and banks. The remaining prerequisite is that banks, shops and customers can agree upon a public key algorithm that is considered safe and operationally acceptable to carry out the necessary mathematical operations underlying the new protocol. Banks must also build and maintain the necessary data bases to support the system. With these assumptions accepted, it will be demonstrated that a practical, smoothly operating system is feasible.

C.G. Guenther (Ed.): Advances in Cryptology - EUROCRYPT '88, LNCS 330, pp. 107-122, 1988.
© Springer-Verlag Berlin Heidelberg 1988

The Crypto - algorithm

The most common public key crypto-algorithm today is the RSA-algorithm. Given a message M, the encrypted version C is obtained by raising M to the power of e, the publicly known part of the key:

$$C = M^e \text{Mod } m$$

Decryption involves the secret part of the key, d:

$$M = C^d \text{Mod } m$$

All operations are performed on a closed set of integer numbers, less than or equal to the modulus m. The security of the RSA-algorithm rests upon the fact that factoring large numbers are a mathematical difficult (hard) problem.

The RSA-algorithm has one vital property:

$$(M^d)^e = M$$

This property is exploited when users of the system want to authenticate themselves. Authentic users will be another necessity in the system to avoid fraud.

The "Electronic coins" - concept

Money (coins and banknotes) are virtually untraceable. To keep track of an individual note by its number would be an almost impossible task. Therefore, the basic idea in the consept of "electronic coins", is to keep the benefit of untraceability of traditional money, and add the benefits of electronically stored and transmitted data representing specific value. This we can obtain by creating electronic coins and storing them in a smart card. When these coins are used, no one would be able to trace the coin itself, neither the user of the coin. An electronic coin is created by:

$$M = S^{e_c}$$

where S is a random number designated "seed" and e_c is the public part of a RSA-key of which "no one" knows the secret part, so that exponentiation with e_c is a true oneway function.

Before sending this coin to the bank to get it signed (approved) by the bank, it must be covered by an envelope:

$$<1>=M*r^e$$

where r is another random number and e is the public part of the RSA-key for the bank.

The bank signs the coin still covered by its envelope:

$$<2> = <1>^d = (M*r^e)^d = M^d *r$$

The signed coin is returned to the customer, and at the same time the customers account is debited for the amount of money that the coin represent. The customer is now able to remove the envelope and check if the transmission and the banks routines have worked properly:

$$<2>*r^{-1} = (M^d * r)\ r^{-1} = M^d$$
$$(M^d)^e = M \quad ?$$

Equality tells that the coin is ready for use.

Use of electronic coins

The coins created are valid for use in shops which are customers of the same bank as that of the payer, or another bank that has direct data communications with the payers bank. Generally, the latter is the case. When a payer presents his money (electronic coins) in the shop, the shop sends to the bank:

$$M^d$$

The bank searches the database to check if the money has already been used. If not, it request the seed, S, from the shop (stored in the customers smart card and read by the shops terminal). S is used to check the validity of the money:

$$S^{e_c} = (M^d)^e \quad ?$$

If equal, the shop's account is credited the correct amount, and the database for used money in the bank is updated.

Giving change to electronic coins

The motivation for implementing new payment systems has up till now been strongest for the banks and possibly large shops or chains of shops. The new systems, based on plastic cards of different kinds has raised the effectivity and lowered the risk involved with physically handling large amounts of money. The protocols suggested by D. Chaum, further elaborated by our work, have also taken into account the need for protection of individuals, and in that respect this payment system should be more acceptable to the general public. However, the protocols as described till now are too simple to be seen as equal to or better than existing systems from the users point of view. One facility that quite obviously must be taken care of, is how to give change in the system. As long as the user (payer) has a positivly balanced bankaccount, he (or she) must be able to use his card for whatever amount of money necessary. An extension of the protocol, showed in the following, sketches a non-trivial solution to this trivial problem.

The smart card must be able to perform some mathematical calculations, namely encryption of an envelope r with a publicly known key for that specific date and coin type, and multiply the unsigned coins with it:

$$M*r^e$$

The bank performs checking of the coins in the previously described way, and returns them to the card via the shop. The card must then strip off the envelope:

$$(M^d*r)r^{-1}= M^d$$

The money generated as change is stored in the card and can later be used in the same way as the ordinary money in the card.

The protocols

In the following paragraph is given a description of the protocols used for the data communication between the customers, shops and banks. Six different sequence diagrams pictures the messages between the communicating parties for different cases. Sequence no. 1 shows how the card is filled with money for the first time. Sequence no.2 shows an ordinary transaction without complications. The last four sequences picture events where some check or other fails, and how the systems deals with this kind of anomalies.

Sequence 1.

a) the card is empty (new) and is filled with money for the first time.

b) A used card is refilled with "fresh" money, discarding earlier loaded coins that are getting old or having impractical values. These coins will be returned to the bank and the account balanced accordingly.

Sequence 2.

This is the protocol for the normal use of the card. The transaction is completed without any malfunction or error. Two different banks may be involved in the data-communication, and change will be given if appropriate.

Sequence 3.

In a real world system there will always be users that are tempted to take advantage of any weakness that can be exploited. Some user could for instance try to obtain goods or services even if he knows that there isn´t enough money in his card to pay for this.(An absent minded person could also trigger this sequence without any harm intended.) The money is checked (as always) against a "used money"-list in the customers bank, and this time the check gives a positive answer. As the card doesn´t contain valid money to pay for the goods or services requested, the transaction is terminated and an "unable-to-pay" -message issued to the customer.

Sequence 4.

One can imagine that the check for used money could give a positive answer even if there where no intention of fraud from the user, for instance some kind of off-line transaction that has taken place without properly updating the card. In this case, there will probably also be valid money in the card that can be correctly used after the first attempt has failed.

Sequence 5.

In addition to the check for used money, the money offered as payment are always tested for validness by requesting the seed used in creating the particular coins offered for the payment. If the test fails, no attempt is made to discover what are the reason for the failing test. The bank is simply stating the fact that this money is not valid, and the offered money is returned to the customer. If the card does not contain any other money, the transaction is terminated with the "unable-to-pay" response. It will be the customers own responsibility to clear this discreapancy with his bank, so that money that doesn´t comply with the "valid-seed" - check is removed from the card.

Sequence 6.

In many cases it will be appropriate to try other coins from the same card if the "seed-check" is negative. This is shown in the last protocol sequence.

Protocol operations

In the sequence diagrams, the following notation is used:

e - encryption key for a particular class of coins

d - decryption key for a particular class of coins

e_c - public encryption key for seed

S - seed

M - unsigned coin

r - envelope

r^{-1} - r - inverse defined for the particular class of coins and its modulus

The operations that the actors in the protocol will be executing, are the following:

A1 - The user activates his home terminal and decides what amount of money he wants in his card by typing it on his terminal. If the card already contains money, he will have to give his PIN - code to get access to the card.

A2 - The user is notified that his card is filled and ready for use.

A3 - The customer types his secret number on the shop´s terminal.

A4 - The customer is notified whether the payment was successful or not.

B1 - Reading the PIN-code.

B2 - Information about sum total to be paid, and transfer of encryption keys for all classes of coins for that particular day.

B3 - Transfer of signed coins and unsigned change from shop to bank.

B4 - Negative respons on the "used-money" - check and request for seed from shop to card.

B5 - Transfer of seed from shop to bank.

B6 - Transfer of change from shop to card.

B7 - Change acknowledgement, payment session terminated.

B8 - Status of used coins from shop to card. Request for money needed to fulfill the payment.

B9 - Break signal from shop to bank. "Unable-to-pay" to customer.

B10- Status of false/unvalid coins from shop to card.

B11- Transfer of unsigned coins to shop´s bank.

C1 - PIN - code check.

C2 - Transfer of old/unvalid coins from card.

C3 - Transfer of seed.

C4 - Storing signed and unsigned coins in the card.

C5 - Payment with signed coins. If change is needed, also unsigned coins must be transferred to the shop.

C6 - Storage of signed change. The card generates r^{-1} for each envelope and modulus.

C7 - Termination of payment session due to low balance.

C8 - Transfer of more coins after alarm due to "money-not-valid".

D1 - Generating new coins.

D2 - Check of PIN - code.

D3 - Request for clearing the card for old or impractical coins.

D4 - Transfer of old or impractical coins from home terminal to bank.

D5 - Request for seed.

D6 - Transfer of seed to bank.

D7 - Removal of envelope and signature check. Seed and unsigned coins to fill the card´s memory are generated after loading with signed coins.

D8 - Acknowledgement of filling session.

S1 - Transfer of coins from shop´s bank to customer´s bank.

S2 - "valid-money".

S3 - Transfer of seed.

S4 - Transfer of change for signing. Account updating.

S5 - Receiving signed change.

S6 - Information transfer regarding used coins.

S7 - Payment session terminated.

S8 - Information transfer regarding false/unvalid coins.

T1 - Check for used coins.

T2 - Check for "money-valid"

T3 - Signing coins. Balancing account. Transfer of signed coins.

T4 - Signing of change. Balancing account.

T5 - Request for terminating checking session.

Implementation

The protocols described in this paper have been developed during a diploma thesis work by Audun Jøsang <2>. They are implemented on a small Token Ring network at the premises of University of Trondheim,Norwegian Institute of Technology, using IBM -AT personal computers as home terminal, POS - terminal and banks. The personal computers have extra cards installed for the Token Ring communication and for the aritmetic functions needed to do the RSA-calculations with reasonable speed. All programs are written in the C programming language.The implementation has shown, allthough in a small scale, that it is quite feasible to realize this kind of payment system with todays technology. The only assumption resting upon further development, is that the smart card will have more memory and the ability to do some straigtforward arithmetic operations. This assumption is believed to be met in the near future.

<1> David Chaum: Privacy Protected Payments. Unconditional payer and/or payer untraceability.
Offprint.

<2> Audun Jøsang: Transaksjonssystemer som skjuler identitet.
NTH Diploma thesis 1987. (in norwegian)

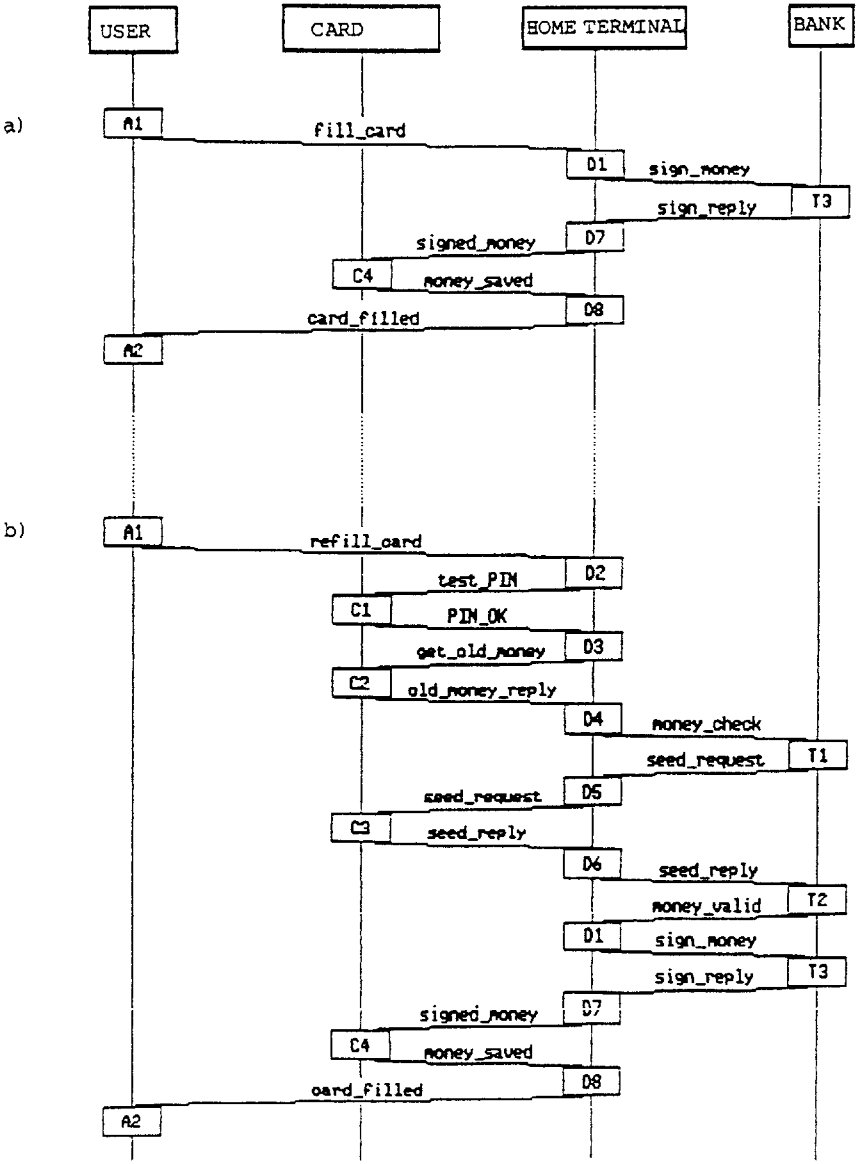

Fig. 1. a) Filling an empty card b) Refilling a used card

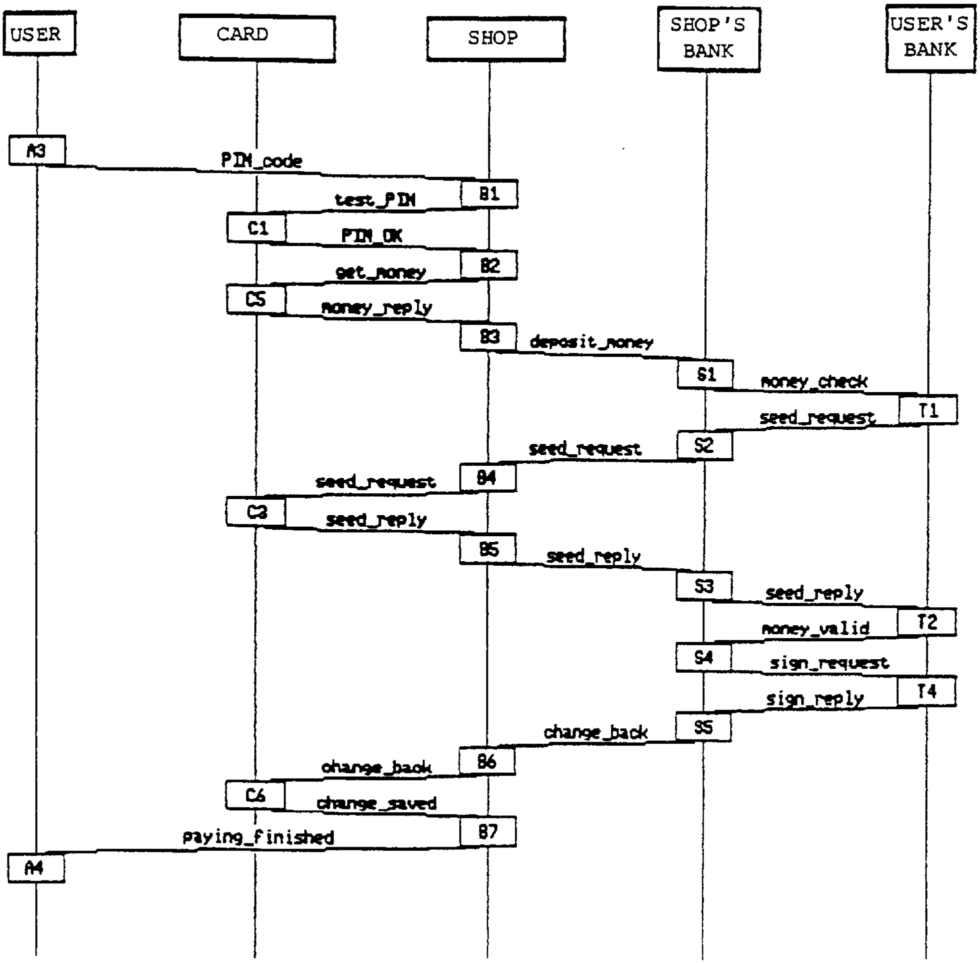

Fig. 2. Ordinary payment (with change)

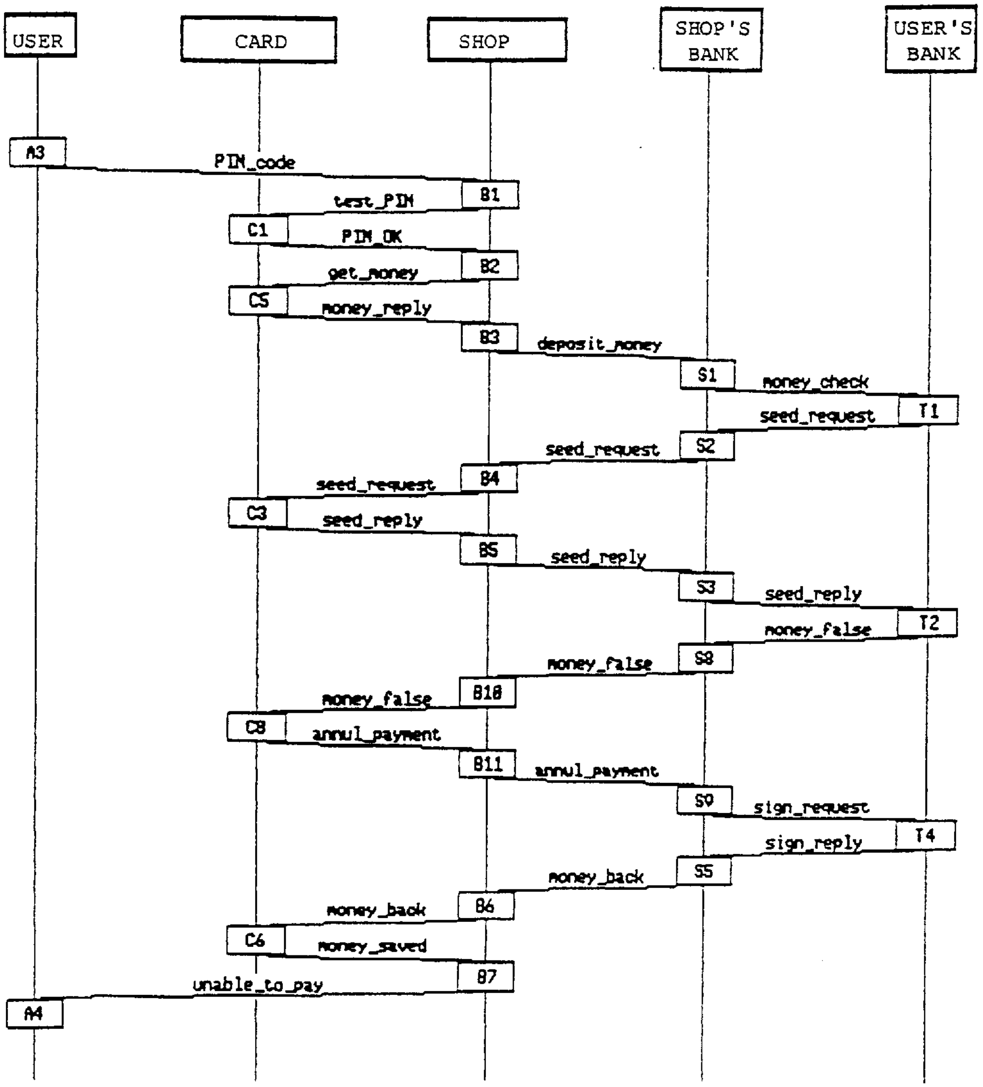

Fig. 3. Attempted payment with false or unvalid money

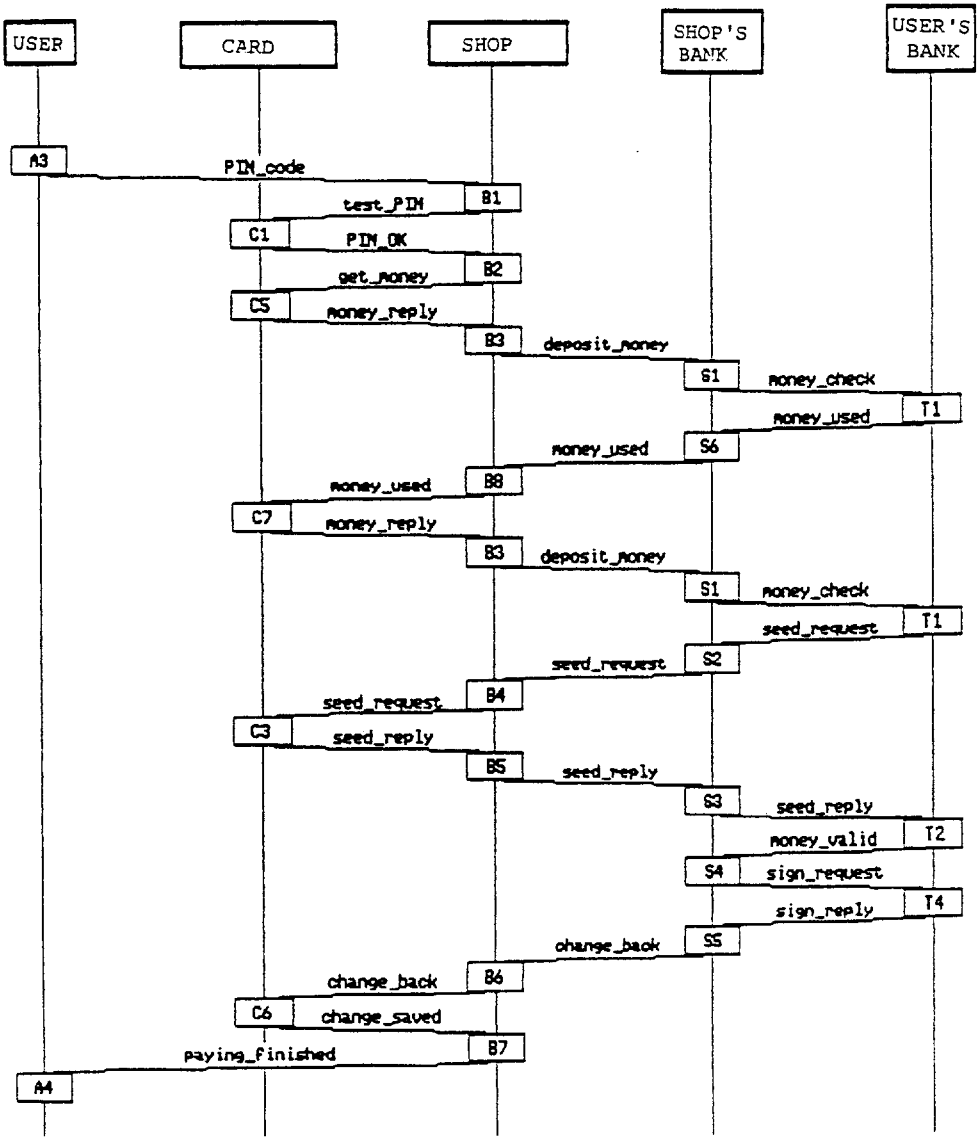

Fig. 4. Money used. Card able to pay

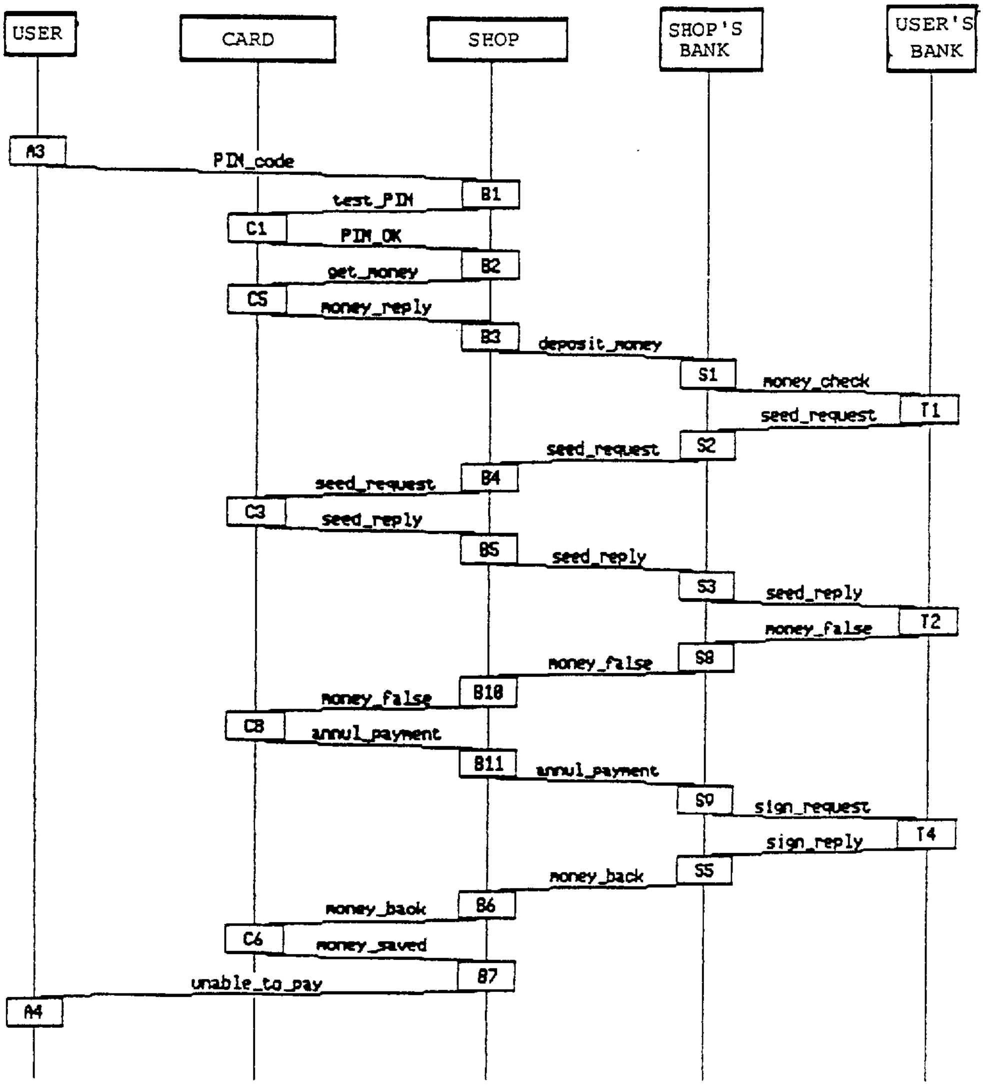

Fig. 5. False or unvalid coins.Transaction terminated.

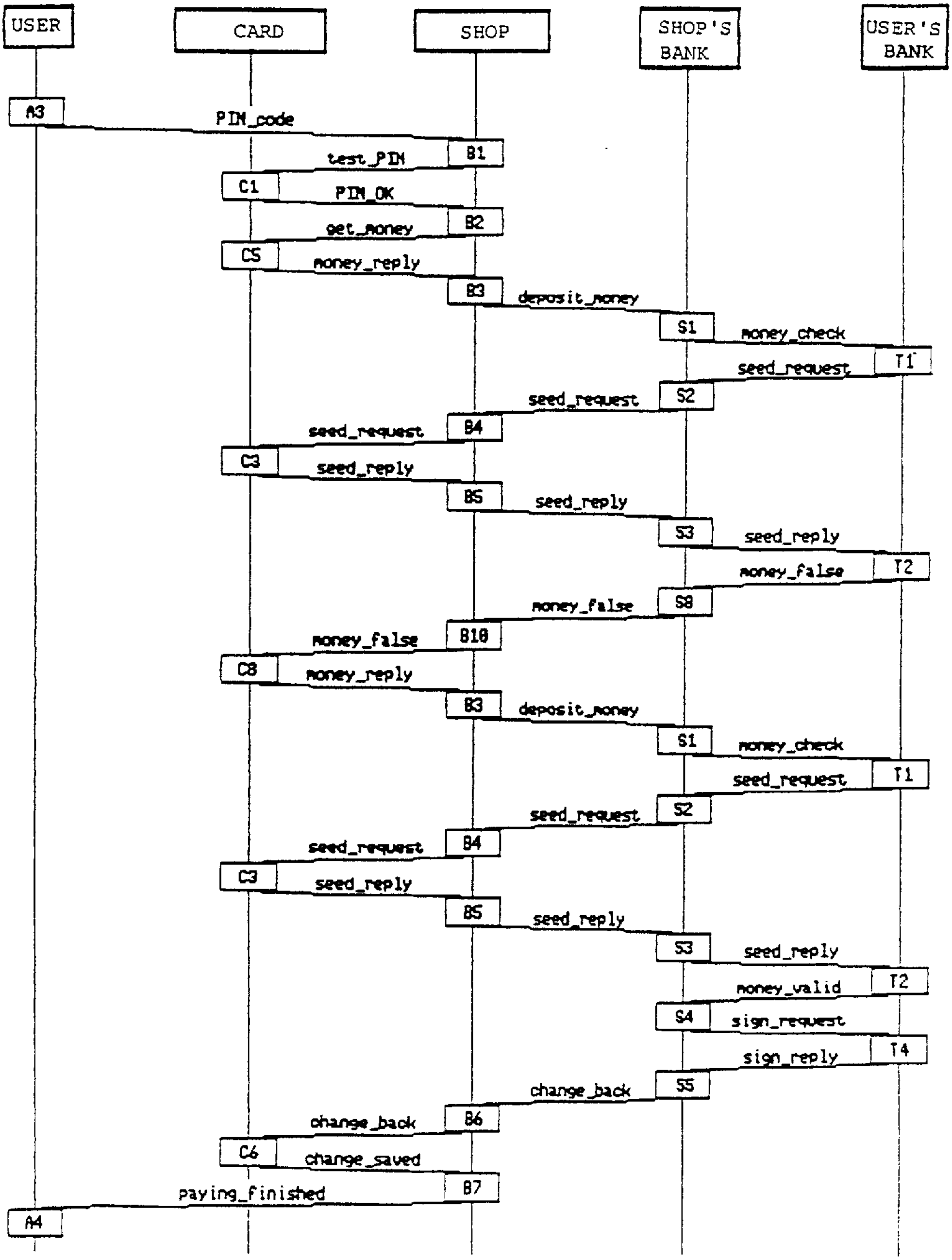

Fig. 6. False or unvalid coins. Card able to pay.