Improved Cryptographic Protocol for Digital Coin Exchange

Marek R. Ogiela, Piotr Sułkowski
AGH University of Science and Technology
Department of Automatics and Biomedical Engineering
30 Mickiewicza Ave.
30-059 Krakow, Poland
e-mail: mogiela@agh.edu.pl, piotr.sulkowski@o2.pl

***Abstract*—In this paper we describe a serious vulnerability of the well-known protocol for exchange of anonymous currency by David Chaum. The weakness of the system consists in the possibility of spending an electronic coin more than once, which can lead to the serious frauds by both the client and the seller. What is more, the system cannot determine the right number of transactions, which were made and the bank is not able to charge the real abuser.**

Keywords—cryptographic algorithms; digital cash; financial systems; electronic transactions

I. INTRODUCTION

Today, regardless of the existence of very well developed online banking and payment card infrastructure, no electronic transaction can be made in secret. However, because of many drawbacks of the exchange of regular cash, a way of anonymously transferring cash over the Internet would be very useful. There are at least two ways in which such a system can be built. One of them is to create a virtual currency, an example of which is Bitcoin [17]. Another one is to create an electronic cash system, in which a bank mediates the transfer of cash and the clients exchange anonymous cheques of a specified value. The first method of anonymous exchange of cheques was invented more than 20 years ago by David Chaum, but still it is the basic solution on which many of later protocols are based [1][2][5][6][8][9]. The main subject of this publication is to present a possible attack on this protocol.

II. PROPERTIES OF SYSTEMS PROCESSING ANONYMOUS TRANSACTIONS

In an electronic cash exchange system, the payment occurs in three stages. At the first, the client contacts the bank and downloads an electronic cheque. Then, the client contacts the seller and spends the cheque downloaded from the bank at the seller's. At the third stage, the seller contacts the bank and presents it with a certificate of concluding a transaction with the client, in return for which it receives its monetary equivalent.

Such an electronic cash exchange system assumes that the three parties involved in it, i.e. the bank, the client and the seller, are completely independent of one another. Thus the system must guarantee to each party that it will not be cheated, even if the other two parties are acting in bad faith and in collusion. In addition, every one of the three stages, i.e. the creation, spending and cashing of the notes can be executed separately. Hence every one of these stages requires a different special protocol. The notes held by the client after it finishes communicating with the bank are called electronic cash due to their guaranteed anonymity, just as paper cash is anonymous. After it is spent at the second stage, the seller holds certificates of the transaction which it presents to the bank any time after finishing its communications with the client.

The most functional system would be one that would support concluding anonymous, irreversible offline transactions without using special tamper-proof devices. However, the problem of banknote copying arises under these assumptions. This is because there is nothing to stop the client from duplicating the banknotes it holds and using them several times in various transactions. This necessitates storing some information about the holder in the banknote. However, we do not want this information to be readable when the user is behaving honestly. Lower down, we present a system based on this approach.

III. CLASSIC PROTOCOL FOR ELECTRONIC CASH EXCHANGE

The first and best known method of implementing an electronic cash system which meets all the assumptions presented above was firstly outlined in [3] and then proposed in [4] by David Chaum. This scheme has become the reference example for implementing an electronic cash exchange, presented e.g. in [7][10][15][16]. In this solution, the banknote is composed of a signed n element sequence of pairs P_i $(i \in 1, 2, \ldots, n)$ having the following structure:

$$P_i = (h(a_i, c_i), h(a_i \oplus u, d_i))$$

where:
u - unique client ID also known to the bank
a_i - number randomly selected by the client to hide the value of u
c_i, d_i - numbers randomly selected by the client to create the hash function with a password

The number n is a certain constant of the system and determines its security (at the cost of its possible efficiency).

978-1-4799-5955-6/14/$31.00 ©2014 IEEE

The client must store all the values of coefficients a_i, c_i and d_i together with the signed sequence of pairs P_i. The details how the client obtains banknotes from the bank can be found in [3][4].

At the next stage, to make the payment, the client presents all pairs P_i and the bank's signature associated with them to the seller. The seller, having checked the regularity of the banknote and its compliance with the signatures, creates the so-called challenge *Y*. This is an *n*-long sequence of zeroes and ones, of which some are constant and assigned to the seller, and some are randomly chosen. The seller then sends them to the client. For every respective element of this sequence, the client returns the following to the seller:

- The values a_i and c_i - if the *i*-th element of the sequence *Y* is 0
- The values $a_i \oplus u$ and d_i - if the *i*-th element of the sequence *Y* is 1

The seller checks the compliance of the values sent with their hashes previously sent by the client in the form of pairs P_i. If everything is correct, the payment is accepted.

At this stage, the seller holds a sequence of pairs P_i, their corresponding signature *Z*, and for each P_i pair also the values (a_i, c_i) or $(a_i \oplus u, d_i)$. This data will be referred to as the transaction certificate. In order to cash the received certificate, the seller contacts the bank and provides it with all the data received from the client as well as the challenge *Y* generated during the sale. The bank checks the regularity of the banknote, the digital signature and also whether the values (a_i, c_i) as well as $(a_i \oplus u, d_i)$ correspond to their hashed values in P_i . If any of these values is incorrect, the fault is on the seller's part, as it was able to independently check the regularity of the banknote when the client was making the payment. If the banknote is correct, the bank checks its database to see whether the same banknote had not been used before. If it has not, the bank credits the appropriate, previously established amount to the seller's account. However, if the same banknote is already kept in the database, the bank tries to establish who is at fault in this situation.

The seller is able to copy the transaction certificate, but all copies will only be correct for one challenge *Y*. This is randomly chosen in each transaction and the seller receives a different certificate every time. Thus if it presents two identical certificates, it can be said with a high likelihood that it is the seller who is trying to commit a fraud. If the certificates differ, this means that it was the client who used the same banknote twice in different transactions.

If the banknote had been used twice, it is highly likely that two challenges randomly chosen in these processes - Y_1 and Y_2, respectively – differ by at least one element. If so, then there is at least one pair P_x for which we know the value of both a_x and $a_x \oplus u$. Thus the bank is able to calculate the value *u*, which is the unique ID of the client. However, this ID cannot be ascertained if the client had paid only once with this banknote. Then the bank always has only either a_i or $a_i \oplus u$ available to it. The value a_i contains no information identifying the client. Neither does the value $a_i \oplus u$ contain any information, because if we assume that a_i is completely random, this number also becomes completely random.

IV. Limitation of this System

Let us consider a case in which the client tries to spend a banknote several times. In this case, the risk to the bank and the seller depends greatly on the strategy of action chosen. There are at least two different options:

- In exchange for every banknote correctly presented by the seller, the bank pays money out, even if the client has spent this banknote several times. The client is then obliged to pay for all transactions concluded.
- If it is detected that a banknote has been used several times, no funds are credited to the seller's account. The seller is only given the personal data of the dishonest client. It is then the seller who must reverse the transaction and possibly claim damages.

Adopting second strategy makes the system a soft one, so all the risk rests with the seller. If the seller incurs any costs of the transaction, it must charge them to the client itself. The limitations resulting from selecting second strategy are a reason to adopt the first strategy. This solution implies that the bank is responsible for prosecuting dishonest clients. However, it is difficult to claim the amount due from clients. One can imagine a situation in which a person with an average income spends one banknote worth one dollar one million times. The bank has to pay a million dollars to sellers. However, it stands no realistic chance to recover this amount due from the client. Another problem is banknote theft. The thief can spend the stolen banknotes many times, each time charging the account of the client it has stolen them from.

Consequently, first strategy gives rise to a greater risk of the bank and the client. Under the second strategy, the risk is mainly borne by the seller. True enough, second strategy restricts the functionality. If a system fulfills all the conditions for first strategy to be adopted, it can also operate according to second strategy. In the opposite case, it is necessary to introduce a mechanism for proving the sale.

Let us thus consider whether the presented system is ready to operate correctly under the first strategy. Let us consider the following scenario. A client, intending to commit a fraud, spends a banknote exactly twice. The injured sellers can now secretly exchange the information received. Assuming that their challenges Y_1 and Y_2 differ in *m* bits, they can jointly generate as many as 2^m different combinations of transaction certificates. It is enough that they combine a part of one certificate with a part of the other. A new transaction certificate is thus produced. In this situation, the sellers have certificates of transactions which have never taken place. They can then approach the bank claiming that they have been cheated many times and the bank cannot establish how many times it was really the client cheating, and how many times the sellers. The bank cannot establish what amount the client has really spent, and therefore cannot demand compensation from the client. This situation makes the bank unable to accept the strategy of guaranteed disbursements in this system. The bank is forced to guarantee only to reveal the identity of dishonest clients. In the

majority of frauds, it will probably be possible to charge double the amount of the transaction to the client, but it must be emphasized that if this method is the only one used, this can never be guaranteed. Hence the seller cannot rely on the regularity of the banknote itself until it cashes it with the bank.

The above example thus shows that the original system is only capable of executing fully reversible transactions without a guarantee that damages will be received if a fraud is committed. There is, however, a possibility to modify this protocol, so that it becomes resistant to this kind of attacks. We propose a brief description of such modification. The details can be found in [11][12][13][14].

V. Proposed Modification

In order to enhance Chaum's system so that irreversible offline transactions can be concluded, it is necessary to introduce the ability to prove all transactions concluded [13][14]. This will make it possible to claim compensation if the same banknote is spent more than once.

To this end, it is necessary to change the certificates issued by the clients in such a way that sellers cannot generate new ones based on any number of those already held.

We therefore propose a modification of the protocol in which clients sign all the challenges received from sellers and send them together with certificates. However, if they used their own key for signing, they would cease to be anonymous. It is therefore necessary to apply a one-time key which should be tied to the real key of the client somehow. One of the possible ways is to attach it to the banknote together with its certificate signed by the client. Instead of pairs P_i, the client then sends triplets:

$$T_i = (h(a_i, c_i), h(a_i \oplus (u \parallel C(K_i)), d_i), K_i)$$

where:

a_i, c_i, d_i - random numbers chosen by the client

u - the unique ID of the client

K_i - public part of a one-time RSA key generated by the client. This can be written e.g. as (e || n), where e is the public exponent of this key, while n is its module

C - the client's certificate employed to sign all K_i keys. This can be e.g. the number $h(K)^d$ mod m, where the numbers (d, m) constitute the private part of the RSA key published by the client (the so-called main key).

Before starting to create banknotes, the clients must register their main public keys with the bank [9]. It is best if clients use keys certified by a certain certification authority. The keys are registered only once for each client, at the time its account is created.

The protocol of creating the banknote, in which the bank and the client are involved, is similar as in the previous version. The difference is the banknote itself, which consists of n triplets T_i, and not pairs P_i as in the previous case. The client sends the bank $2*n$ obfuscated banknotes, from which the bank chooses one half and asks the client to remove the obfuscation from them. Having received all the necessary coefficients, the bank checks the regularity of the banknotes just as before. In addition, it must assure itself that all certificates $C(K_i)$ contained in the sent banknotes are the correct certificates of keys K_i, i.e. they apply to the key K_i contained in the subsequent part of the banknote and they have been signed with the client's main key (which the bank holds in its database). If everything is correct, then the bank sends the signed banknote to the client just as in the previous version of the protocol. Thus the client, having removed the obfuscation, has the following number:

$$Z = \prod h(T_i)^d \bmod n, i \in L$$

where:

d - the private exponent of the bank's key

n - the module of bank's signature

L - a set of indexes of banknotes selected for signing by the bank

The idea of this entire improvement is that every transaction executed by the client should leave a unique trace that cannot be faked. To obtain this functionality of the protocol, the client signs the challenge sent to it by the seller using the K_i keys contained in the banknotes. The protocol for the banknote exchange between the client and the seller thus looks as follows:

1. The client sends the banknote Z signed by the bank.
2. The client also sends the T_i triplets (i.e. the values $h(a_i, c_i)$, $h(a_i \oplus (u \parallel C(K_i)), d_i)$ and K_i).
3. The seller checks whether banknote Z is the correct signature of the signed triplets.
4. The seller sends the challenge Y to the client.
5. The client provides the seller with the value of the challenge Y signed with all one-time keys K_i:

$$R = K_1(K_2(\ldots K_n(Y)\ldots))$$

6. The seller verifies the validity of the signature R.
7. The client provides the seller, respectively, with the values (a_i, c_i) or $(a_i \oplus (u \parallel C(K_i)), d_i)$ depending on the value of the i-th bit of challenge Y (just as in the previous version of the protocol)
8. The seller checks whether the data sent corresponds to the hashes contained in triplets T_i. If everything is correct, the payment is accepted.

In order to cash the banknote, at whatever moment, the seller presents the bank with the signed banknote Z, the sequence of triplets T_i, the generated challenge Y together with the signature R and all the values dependent on this challenge sent by the client. The bank is able to check the regularity of all data in the same way as the seller was able to when it exchanged the banknote with the client.

Just as in the standard version of the protocol, after spending the banknote once, the client reveals only one half of each liability. At the same time, if the same banknote has been spent at least twice, the bank is highly likely to possess two complementary halves, but thanks to the modification it will

learn not only the client's identity, but also at least one of the certificates $C(K_i)$.

Thus the bank becomes able to prove to the client that it has spent a banknote more than once. This is because every transaction is signed by the client with all keys K_i. At the time of the fraud, the bank not only knows the identity of the client, but also holds at least one set containing the liability signed with a certain one-time key and the client's certificate authenticating this key. If the bank is able to provide the client with the signature R of a given challenge Y and to prove to it that the signature belongs to the client, this transaction can be considered proven. This is because no one other than the client can create the certificate for the key K_i used to sign the challenge. Neither can anyone fake this signature as the banknote only contains the public part of it.

If the client spends banknotes only once, then it is impossible to learn either its ID u, or any of the certificates $C(K_i)$. This certificate, together with the key K_i, could also be used to identify the client. It is enough that the bank tries to verify this certificate by using all main keys of clients it holds in its database. One of them would probably be correct. This could be the basis for discovering the identity of the client, so certificates must also be kept secret until a fraud occurs. What is, however, overt is the key K_i itself. It is created randomly by the client and contains no information that could identify it.

After this solution is implemented, the bank is able to prove exactly how much the client has spent. Consequently (assuming that it is able to recover this receivable from the client by way of effective collection), it can adopt the strategy of paying funds to all sellers who present correct transaction certificates. Thus the system makes offline transaction conclusion possible.

What still remains is the problem of banknote theft. If a banknote ends up in the hands of an unauthorized person, it can be used to overdraw the owner's account without any limitation. To prevent this, once the client learns of the theft, it can report it to the bank so that the latter publishes a list of void banknotes. In addition, the bank itself, once it detects a double payment, can publicly report this banknote as stolen. On the other hand, if we assume that transactions are concluded without contacting the bank, we can never eliminate this problem completely. However, the same difficulty arises in the digital signature scheme itself. The problem can be eliminated if, every time before we start receiving a digital signature, we refer to a public database to check if the signature has not been stolen. However, if we decide to build a system which accepts signatures offline - without contacting a public database of stolen signatures – we can never be certain that the signature has not been stolen.

If the risk of theft is considered to be too high, it is always possible to fall back on the strategy of concluding only irreversible transactions. Apart from capacity issues, the proposed modification does not weaken the original system in any way.

VI. SUMMARY

As we can see the protocol for anonymous currency exchange by David Chaum has a serious vulnerability and because of that, cannot be used to make irreversible transactions. The system can be abused by both the client and the seller if an electronic banknote was spent at least twice. It is essential that the same problem also applies to other systems based on a similar protocol. However, the threat of abuse can be eliminated by a modification of the basic protocol.

ACKNOWLEDGMENT

This work has been supported by the AGH University of Science and Technology research Grant No 11.11.120.329

REFERENCES

[1] S. Brands, “Untraceable Off-line Cash in Wallets with Observers,” CRYPTO '93 Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology, pp. 302-318, Springer-Verlag, 1994.

[2] S. Brands, “Off-Line Electronic Cash Based on Secret-Key Certificates,” Lecture Notes in Computer Science, vol. 911, pp. 131-166, 1995.

[3] D. Chaum, “Blind Signatures for Untraceable Payments,” Advances in Cryptology: Proceedings of Crypto 82, pp. 199-203, Springer-Verlag, 1983.

[4] D. Chaum, A. Fiat, M. Naor, “Untraceable Electronic Cash,” CRYPTO '88 Proceedings on Advances in Cryptology, pp. 319-327, Springer-Verlag, 1990.

[5] R. H. Deng, Y. Han, A. B. Jeng, T. Ngair, “A new on-line cash check scheme,” Proceedings of the 4th ACM conference on Computer and communications security, pp. 111-116, ACM, 1997.

[6] N. Ferguson, “Single term off-line coins,” EUROCRYPT '93 Workshop on the theory and application of cryptographic techniques on Advances in cryptology, pp. 318-328, Springer-Verlag, 1994.

[7] S. Goldwasser, M. Bellare, Lecture Notes on Cryptography. Cambridge, 2008.

[8] S. Kim, H. Oh, “A new electronic check system with reusable refunds,” International Journal of Information Security, vol. 1(3), pp. 175-188, 2002.

[9] W. Mao, Blind Certification of Public Keys and Off-line Electronic Cash. Hawlett-Packard Laboratories, 1996.

[10] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, Handbook of Applied Cryptography. CRC Press, 1996

[11] M. R. Ogiela, U. Ogiela, “Linguistic Protocols for Secure Information Management and Sharing,” Computers and Mathematics with Applications, vol. 63(2), pp. 564-572, 2012.

[12] M. R. Ogiela, U. Ogiela, Secure Information Management using Linguistic Threshold Approach. Advanced Information and Knowledge Processing, DOI 10.1007/978-1-4471-5016-9, Springer-Verlag, London 2014.

[13] M. R. Ogiela, P. Sułkowski, “Protocol for irreversible off-line transactions in anonymous electronic currency exchange,” Soft Computing, DOI 10.1007/s00500-014-1442-2, Springer, 2014.

[14] M. R. Ogiela, P. Sułkowski, “Protocol for Detection of Counterfeit Transactions in Electronic Currency Exchange,” in Z. Kotulski et al. (Eds.): CSS 2014, CCIS 448, pp. 145–152, 2014, Springer-Verlag Berlin Heidelberg, 2014.

[15] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley, 1996.

[16] B. Schneier, Secrets and Lies: Digital Security in a Networked World. Wiley, 2004.

[17] Website of *Bitcoin* foundation developing virtual currency with the same name: http://bitcoin.org