Protocol for irreversible off-line transactions in anonymous electronic currency exchange

Marek R. Ogiela · Piotr Sułkowski

Published online: 5 September 2014

© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract This paper presents an improvement to the well-known protocol by David Chaum for anonymous currency exchange. We show its vulnerability to serious frauds by both the client and the seller, after an electronic coin is spent at least twice. In this case, the system cannot successfully determine how many times the client spent the coin and how many times the seller faked the transaction. Therefore, the bank is not able to charge the real abuser. This limitation leads to the conclusion that the original system cannot be securely used for irreversible off-line transactions. In this paper, we show the gist of the problem and propose an improved system based on its original off-line version that allows this vulnerability to be overcome.

Keywords Electronic cash · Anonymous transaction · Protocols for exchanging electronic coins

1 Introduction

Today, regardless of the existence of very well developed online banking and payment card infrastructure, there is no way to completely freely send cash in a way guaranteeing one's anonymity. This means that no electronic transaction can be made in secret, as is frequently done when regular cash is used. However, for many reasons, the exchange of regular cash is not a convenient method, so 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 (Hao et al. 2005). Bitcoins are an example of such a system (Bitcoin 2014). The second way is to create an electronic cash system. In this solution, a bank mediates the transfer of cash, but the customers only exchange anonymous cheques of a specified value. Creating such a protocol is far from trivial, also due to the ease with which electronically stored data can be copied.

It should be noted that the first method that enabled cheques to be anonymously exchanged is David Chaum's protocol described in paper (Chaum et al. 1990). Even though it was developed more than 20 years ago and has been improved and modified many times, it has so far been the basic solution and a large proportion of protocols developed later were founded on it (Brands 1994, 1995; Deng et al. 1997; Ferguson 1994; Kim et al. 2002; Menezes et al. 1996). The main subject of this publication is to present an attack on David Chaum's protocol and propose a modification which would allow fraud attempts in this protocol to be eliminated. Firstly, this study will characterise the features of electronic cash exchange systems, next describe the operating mechanism of the basic version of David Chaum's protocol followed by analysis of its vulnerabilities and then show the authors' proposal for enhancing this protocol in a way which prohibits the client or the sellers from committing fraud during multiple offline transactions.

2 Properties of systems processing anonymous transactions

In an electronic cash exchange system, the payment occurs in three stages.

Communicated by V. Loia.

M. R. Ogiela (✉) · P. Sułkowski
Cryptography and Cognitive Informatics Research Group,
AGH University of Science and Technology, Al. Mickiewicza 30,
30-059 Krakow, Poland
e-mail: mogiela@agh.edu.pl

P. Sułkowski
e-mail: piotr.sulkowski@o2.pl

At the first stage, 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.

If transaction protocols are executed like this, the main problem is that the bank must not be able to associate the transaction certificates presented to it by the seller with the cheque which it had issued to the client.

The key requirement for all electronic money systems is that they must operate securely. This applies to the clients, the sellers and the bank itself. The absolute condition is that every party must have a guarantee that it will not be cheated and that this guarantee is not based only on confidence in other participants of the system. Such requirements apply to all systems and their implementations, regardless of other properties which generally boil down to functionality enhancements. In this case, security is mainly understood as the lack of ability to steal funds from other system users, but it also represents the guarantee that other properties will be maintained, such as the transaction anonymity.

The main features of electronic cash transfer systems are as follows:

- Anonymity—the system must guarantee that the bank cannot trace the transactions. Some systems support the use of a single high denomination banknote at many sellers', by transferring only a certain part of this banknote corresponding to the value of the transaction to each seller. However, for such systems, a frequent problem should be noted that the bank could link all the transactions in which fragments of one banknote were used.
- Transaction processing may occur when the banknotes are spent, or may take place later. Systems in which sellers must cash banknotes when they receive them from clients work online. By analogy, systems in which execut-

ing the transaction does not require contacting the bank execute payments off-line. As a rule, they are harder to implement and require additional mechanisms preventing banknotes from being copied. Eliminating the need to stay in constant touch with the bank makes the entire system more complex, usually posing a significant burden on its capacity. At the same time, the possibility to make the transactions off-line is a very desirable property.

- Transaction reversibility. If a seller is able to reverse a transaction and incur no cost or loss, such transactions are referred to as 'soft'. Conversely, if transactions are irreversible, they are called 'hard'. When the seller concludes a hard transaction, it must be certain that it will receive the payment for it. This problem is to some extent connected with the previous criterion. Any soft online system can be turned into a soft off-line one. If a banknote is not processed correctly, it is enough to reverse the transaction. However, it should be noted that supporting only soft transactions significantly limits the functionality of the system.
- System requirements concerning access to data. There are systems, such as those described in Brands (1994), which require the users to hold banknotes stored only on specially secured tamper-proof cards. Such a card stores data about banknotes but does not allow unauthorised persons to read it. It only allows transactions to be executed in a way the card supervises. The majority of data stored on it is used exclusively for executing the calculations (e.g., executing digital signatures) but is never directly read. The use of this technology simplifies the majority of operations greatly, but introduces many restrictions on the use of the system. An additional drawback is the fact that system security is based on purely mechanical security measures preventing cards from being read.

In the light of the above properties, one can see that the most functional system would be one that would support concluding anonymous, hard off-line 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. The seller would then be unable to protect itself from receiving, in several instances, a banknote that has already been used, because the assumption of this system is that there is no need to verify banknotes immediately. Another condition is that transactions must be irreversible. If these two assumptions exist in parallel, the seller is powerless facing dishonest users who duplicate banknotes. To take away the client's ability to commit such fraud, a way is needed of detecting dishonest users after the transaction has been made. 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.

3 Basic Chaum's protocol for electronic cash exchange

The first and best known method of implementing an electronic cash system that meets all the assumptions presented above was firstly outlined in Chaum (1983) and then proposed in Chaum et al. (1990) by David Chaum. This scheme has become the reference example for implementing an electronic cash exchange, presented, e.g., in Goldwasser and Bellare (2008), Menezes et al. (1996), Schneier (1996) and Schneier (2004). In this solution, the banknote is composed of a signed n element sequence of pairs $P_i (i \in 1, 2, \dots, n)$ having the following structure:

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

where u is a unique client ID also known to the bank, a_i is a number randomly selected by the client to hide the value of u , c_i , d_i are 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). To obtain such an electronic banknote, the client communicates with the bank using the following protocol:

1. The client randomly chooses $2 * n$ pairs P_i and their corresponding obfuscating coefficients r_i .
2. The client sends all the obfuscated hashes of pairs $r_i^e * h(P_i)$ to the bank (the number e is the public part of the bank's key).
3. The bank selects n of the pairs sent and asks the client to send their corresponding values of r_i , a_i , c_i and d_i .
4. The client sends the requested numbers to the bank.
5. The bank checks if they comply with the obfuscated hashes sent before.
6. The bank signs all the remaining pairs, multiplies them one by the other and sends them back to the client. It also debits the value of the banknote to the client's account balance.
7. The client receives the signed pairs and then removes the obfuscation from them.

At step 6, the client receives the product of multiplying the signatures of all selected pairs:

$$I = \prod (r_i^e * h(P_i))^d \bmod n, \quad i \in L$$

where e is a the public exponent of the bank key, d is a the private exponent of the bank key, n is a the bank signature module, L is a set of indexes of banknotes selected for signing by the bank. The value I from above formula contain

obfuscated all remaining pairs signed by bank private key. This value allows further to reveal the signed banknote by dividing this value by subsequent obfuscating factors r_i . In such a way, the client obtains the signed banknote, which is represented by following formula:

$$Z = I * \prod r_i^{-1} \bmod n, \quad i \in L,$$

where I is the product of multiplied obfuscated signatures, r_i is a obfuscating coefficients, n is a the bank signature module, L is a set of indexes of banknotes selected for signing by the bank. Finally, the signed banknote is described by following formula:

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

At this stage, the client holds a signed banknote composed of n pairs P_i and the signature Z certifying these pairs. In addition, all the values of coefficients a_i , b_i , c_i and d_i must be stored.

At the next stage, to make the payment, the client presents all pairs P_i and the bank's signature Z 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:

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

The client's response to the challenge is illustrated in Fig. 1. On a current basis, 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. 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

Fig. 1 Client's response to the challenge of the seller

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 $a_x \oplus u$ available to it. The value a_i contains no information identifying the client. Neither does the value $a_x \oplus u$ contain any information, because if we assume that a_i is completely random, this number also becomes completely random.

In publication (Chaum et al. 1990), David Chaum mentions the possibility of introducing an additional modification to the protocol that would enable the bank to prove that the client has used the same banknote more than once. In the scheme shown above, the bank is able to generate any certificate on behalf of the client but without its knowledge and permission. The easiest way of stripping the bank of this ability is to force the client to sign the banknotes it receives. The bank will then be unable to create a banknote without the client's involvement, so once the identity of the cheating client is ascertained, this fraud can be proven.

To support this functionality, the unique customer ID u is replaced with an additional number z_i randomly selected by

the client. The banknote is then composed of n pairs P_i of the following form:

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

where the operator $\parallel$ represents such a combination of the numbers that each one can be read separately (e.g., by writing two numbers of a specified bit length one next to the other).

The process of spending the banknote will then additionally start with the client sending the signed list of abbreviations of all numbers z_i . The bank will thus receive the following number:

$$Z = h(z_1), h(z_2), \dots, h(z_n)$$

and the signature: $C_K(Z)$. At the next stage, the client sends the obfuscated banknotes as such. The bank chooses half of them and checks if they are correctly structured. It then receives the appropriate numbers z_i from the client as well. After the banknote has been spent, the bank holds half of the z_i liabilities and the hashes of all these liabilities including the signed ones. If the client uses a banknote more than once, not only will the client's identity be revealed, but so will at least one new z_i coefficient. If the bank presents more than exactly half of the signed liabilities z_i , this is considered to prove that the client has used a banknote several times, because the bank cannot independently prepare such liabilities.

4 Properties and limitations of David Chaum's system

The basic properties of the presented system are as follows:

1. A transaction can be concluded without the need to contact the bank at the time of its conclusion.
2. No need to use any special tamper-proof devices or cards.
3. Transactions are anonymous if the client is behaving honestly.
4. The security of every party is guaranteed even if the remaining two parties are acting in collusion against it.

Apart from these properties, certain risks to the parties using this system may also arise. The main ones are: the same banknote being spent many times, counterfeit banknotes being made and the loss of the client's anonymity.

Let us then 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:

- A. 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.
- B. 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 strategy B 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 strategy B are a reason to adopt strategy A. 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, strategy A gives rise to a greater risk of the bank and the client. Under the B strategy, the risk is mainly borne by the seller. True enough, strategy B restricts the functionality. If a system fulfils all the conditions for strategy A to be adopted, it can also operate according to strategy B. 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 A 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 emphasised 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 Chaum's system is only capable of executing fully reversible transactions without a guarantee that damages will be received if a fraud is committed.

5 A proposed enhancement to the protocol supporting multiple transaction detection

To enhance Chaum's system so that irreversible off-line transactions can be concluded, it is necessary to introduce the ability to prove all transactions concluded. 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 (the structure of the banknote is presented in Fig. 2). 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 is a random numbers chosen by the client, u is the unique ID of the client, K_i — is a public part of a one-time RSA key generated by the client. This can be written, e.g., as $(e \parallel n)$, where e is the public exponent of this key, while n is its module, C is the client's certificate employed to sign all K_i keys. This can be, e.g., the number $h(K)^d \bmod m$, 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 (Mao et al. 1996). It is best if clients use keys certified by a certain certification

Fig. 2 Modified digital note

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 $2 \cdot n$ 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, which represents modified banknote:

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

where d is the private exponent of the bank key, n is the bank signature module, L is 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 (presented in Fig. 3) 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 :

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

where $K_i(x)$ is the signature of the value x with the use of the key K_i .

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.

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

Fig. 3 Customer response to the challenge of the seller in the protocol proposed by the authors

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 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 ran-

domly 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 off-line transaction conclusion possible.

What still remains is the problem of banknote theft. If a banknote ends up in the hands of an unauthorised 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 that accepts signatures off-line—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.

6 Summary

The system presented by David Chaum in Chaum et al. (1990) is the first system capable of processing anonymous transactions off-line and represented a ground-breaking discovery in the field of anonymous electronic payments. However, if the same banknote is spent at least twice by the client, the system becomes susceptible to attacks both by cheated sellers and the client itself. This weakness means it cannot be used to conclude irreversible transactions. The same problem also applies to other systems based on a similar protocol. The introduction of the modification proposed by the authors allows such attacks to be prevented and makes it possible to conclude hard transactions as well.

At present there is no widely used electronic cash system based on the electronic cheque scheme, but there are no obstacles to building one (Ma et al. 2011). What is necessary is a certain surcharge of calculations for every transaction as well as the suitably greater storage resources which will allow data about used up banknotes to be stored for a long time. Solutions described in publication (Ferguson 1994) allow the necessary resources to be significantly reduced.

It seems that, sooner or later, electronic cash technology will gain in popularity and will start replacing traditional credit cards and bank transfers (Hao et al. 2005). This is very probable, as otherwise the banks would obtain huge amounts of confidential information about their customers. However, maintaining secrecy is crucial for the security and development of many companies, which will therefore be happy to use this new solution to mitigate the risk of losing data or intrusion detection (Leu et al. 2010; Ogiela and Ogiela 2012a, b, 2014).

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Bitcoin (2014) Website of Bitcoin foundation developing virtual currency with the same name. <http://bitcoin.org>. Accessed 15 May 2014
- Brands S (1994) Untraceable off-line cash in wallets with observers. In: Proceedings of the 13th annual international cryptography conference on advances in cryptology, CRYPTO '93, pp 302–318, Springer-Verlag, New York

- Brands S (1995) Off-line electronic cash based on secret-key certificates. In: Lecture notes in computer science, vol 911, pp 131–166. Springer-Verlag, New York
- Chaum D (1983) Blind signatures for untraceable payments advances in cryptology. In: Proceedings of Crypto 82, pp 199–203. Springer-Verlag, New York
- Chaum D, Fiat A, Naor M (1990) Untraceable electronic cash. In: Proceedings on advances in cryptology, CRYPTO '88, pp 319–327. Springer-Verlag, New York
- Deng RH, Han Y, Jeng AB, Ngair T (1997) A new on-line cash check scheme. In: Proceedings of the 4th ACM conference on computer and communications security, pp 111–116. ACM, London
- Ferguson N (1994) Single term off-line coins. In: Workshop on the theory and application of cryptographic techniques on advances in cryptology, EUROCRYPT '93, pp 318–328. Springer-Verlag, New York
- Goldwasser S, Bellare M (2008) Lecture notes on cryptography. Cambridge University Press, Cambridge
- Hao YY, Havey DM, Tumer DA (2005) An exchange protocol for alternative currencies. In: International conference on information technology—coding and computing location, vol 1, pp 420–424. IEEE Computer Society, Las Vegas, 04–06 April 2005
- Kim S, Oh H (2002) A new electronic check system with reusable refunds. *Int J Inf Secur* 1/3:175–188 (Springer-Verlag)
- Leu F-Y, Yang C-T, Jiang F-C (2010) Improving reliability of a heterogeneous grid-based intrusion detection platform using levels of redundancies. *Future Gener Comput Syst* 26(4):554–568
- Ma W-M, Wang Ke, Liu Z-P (2011) Mining potentially more interesting association rules with fuzzy interest measure. *Soft Comput* 15(6):1173–1182
- Mao W (1996) Blind certification of public keys and off-line electronic cash. Hawlett-Packard Laboratories, Palo Alto
- Menezes AJ, van Oorschot P C, Vanstone SA (1996) Handbook of applied cryptography. CRC Press, London
- Ogiela MR, Ogiela U (2012) Linguistic protocols for secure information management and sharing. *Comput Math Appl* 63(2):564–572
- Ogiela MR, Ogiela U (2012) DNA-like linguistic secret sharing for strategic information systems. *Int J Inf Manag* 32(2):175–181
- Ogiela MR, Ogiela U (2014) Secure information management using linguistic threshold approach. *Adv Inf Knowl Process*. doi:10.1007/978-1-4471-5016-9 (ISSN 1610-3947, ISBN 978-1-4471-5015-2, Springer-Verlag, London)
- Schneier B (1996) Applied cryptography: protocols, algorithms, and source code in C. Wiley, London
- Schneier B (2004) Secrets and lies: digital security in a networked world. Wiley, New York