

Timeliness Optimistic Fair Exchange Protocol Based on Key-Exposure-Free Chameleon Hashing Scheme

Yanbin Sun^{1,2}, Lize Gu^{1,2}, Sihan Qing^{1,3}, Shihui Zheng^{1,2}, Bin Sun^{1,2}, Yixian Yang^{1,2}, Yan Sun⁴

¹Information Security Center, State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China

²Key Laboratory of Network and Information Attack and Defence Technology of MOE,
Beijing University of Posts and Telecommunications, Beijing 100876, China

³Institute of Software, Chinese Academy of Sciences, Beijing 100080, China

⁴Operations Research Center, Shijiazhuang Army Command Academy, Shijiazhuang 050084, China
ybsun@foxmail.com

Abstract—In ESS2008, Yang et al. suggested a key-exposure-free chameleon hashing scheme, and it can be used to design signature and some other cryptography mechanism. In this paper, we construct a new timeliness optimistic fair exchange protocol based on this key-exposure-free chameleon hashing scheme. The new scheme does not require the use of interactive zero-knowledge proofs in the exchange phase. In our scheme, both parties can contact the trusted third party and settle the argument before the deadline. Moreover, the new scheme achieves fairness and timeliness.

Keywords—fair exchange; key-exposure-free chameleon hash; abortion; security; timeliness

I. INTRODUCTION

Fair exchange of electronic information (contract signing, e-cash payment, etc.) is one of the main security problems in e-commerce transactions. A fair exchange protocol (FEP) is defined to ensure no party in an electronic commerce transaction can gain an advantage over the other party by misbehaving, misrepresenting or by prematurely aborting the protocol. In other words, an exchange protocol achieves fairness if it guarantees that either both parties obtain each other's data or none of them get anything useful. Fair exchange protocols can be broadly categorized into three types: (I) gradual exchange protocols [1, 2]: This kind of scheme requires many steps of interactions for exchanging message; (II) protocols requiring an online trusted third party (TTP) [3]: An online TTP who acts as a mediator receives the message from both parties in each transaction and then forwards them to the accurate receiver; (III) protocols requiring an offline TTP (or optimistic fair exchange protocols) [4-6]: The offline TTP does not participate in the actual exchange protocol in normal cases, and is invoked only in abnormal cases to dispute the arguments between the two parties to ensure fairness.

Asokan et al. [4, 5] first proposed the optimistic fair exchange protocol (OFEP) which is based on the concept of verifiably encrypted signature (VES). In that scheme, Alice encrypts her signature with the TTP's public key, and

proves to Bob that she indeed encrypted her valid signature. After receiving her expected item from Bob, Alice proceeds to open the encryption. The approach of [4, 5] was later generalized by [6], but all those schemes involved expensive and highly interactive zero-knowledge proofs (ZKP) in the exchange phase which greatly reduces the efficiency.

To improve efficiency, Park et al. [7] proposed a non-interactive OFEP based on regular RSA signatures. The protocol uses the ZKP only in the setup phase—this is a one-time cost. Later, Dodis et al. [8] shown that the fair exchange protocol presented by Park et al. [7] is insecure. Recently, the notion of non-interactive OFEP has attracted a great extension [9-14].

In ESS2008, Yang et al. [14] designed a key-exposure-free chameleon hashing (KEFCH) scheme based on discrete logarithm assumption. Their KEFCH scheme is secure which satisfies: collision resistant, semantic security, message hiding and key exposure freeness. Yang et al. also proposed an optimistic fair exchange protocol based on KEFCH scheme. Unfortunately, the YYK scheme has two weaknesses. These weaknesses may lead to an unfair situation that Bob can obtain Alice's signature, but Alice can not obtain Bob's signature.

In [15, 16], the authors proposed some discussions of the most common timeliness-related mistakes and some informal guidelines for the OFEP design. In this paper, followed [15, 16], we construct a new timeliness optimistic fair exchange protocol based on the KEFCH scheme [14]. The new scheme does not use VES and does not require ZKP in the exchange phase, which will greatly reduce the communication overhead and managing cost. In our scheme, both parties can contact the TTP and solve the exchange whenever they want before the deadline. Furthermore, the new OFEP achieves fairness and timeliness.

The rest of the paper is organized as follows. In Section 2, we introduce the concept of the CH scheme which is proposed by Yang et al. In Section 3, a new timeliness optimistic fair exchange protocol is presented. We analyze

the security of our proposed protocol in Section 4. Finally, Section 5 concludes the paper.

II. OVERVIEW THE KEFCH SCHEME

Yang et al. [14] proposed a key-exposure-free chameleon hashing scheme based on the discrete logarithm assumption. The KEFCH scheme satisfies: collision resistant, semantic security, message hiding and key exposure freeness. Formally, the KEFCH scheme consists of the following algorithms:

GenKey: Let t be a prime power, and $E(F_t)$ is an elliptic curve over finite field F_t . $\#E(F_t)$ denotes the number of points of $E(F_t)$, and P is a point of $E(F_t)$ with prime order q , where $q \mid \#E(F_t)$. G is a subgroup generated by P . Hash function $H : Z_q^* \times \{0, 1\}^* \rightarrow Z_q^*$ is cryptographic secure. Choose two random elements $x, k \in Z_q^*$, and compute $Y = xP$, $K = kP$. The private trapdoor key is $TK = (x, k)$, and the public hash key is $HK = (T, K)$.

Hash: Given a message $m \in Z_q^*$, the label $L \in \{0, 1\}^*$, randomly select $r \in Z_q^*$. The chameleon hash function is defined as follows:

$$h = \text{Hash}(L, m, r) = e(P + K) + rY,$$

where $e = H(m, L)$.

UForge (Universal forge): Given the private trapdoor key TK , the original hashed value (m, r) and the message m' to be forged, a collision (m', r') can be found by computing

$$r' = r + (e - e')(x^{-1} + kx^{-1}) \bmod q,$$

where $e' = H(m', L)$.

Note that

$$\begin{aligned} & \text{Hash}(L, m', r') \\ &= e'(P + K) + r'Y \\ &= e'(P + K) + rY + (e - e')(x^{-1} + kx^{-1})Y \\ &= e'(P + K) + rY + (e - e')(P + K) \\ &= e(P + K) + rY \\ &= \text{Hash}(L, m, r). \end{aligned}$$

SForge (Instance forge): Let (m, r) and (m', r') be a pair of collision, i.e. $\text{Hash}(L, m, r) = \text{Hash}(L, m', r')$, another collision pair (m'', r'') can be computed by

$$r'' = r' + (e' - e'')(e - e')^{-1}(r' - r),$$

where $e'' = H(m'', L)$.

Indeed

$$\begin{aligned} & \text{Hash}(L, m'', r'') \\ &= e''(P + K) + r''Y \\ &= e''(P + K) + r'Y + (e' - e'')(e - e')^{-1}(r' - r)Y \\ &= e''(P + K) + r'Y + (e' - e'')(P + K) \\ &= e'(P + K) + r'Y \\ &= \text{Hash}(L, m', r') \end{aligned}$$

III. NEW OPTIMISTIC FAIR EXCHANGE PROTOCOL

In this section, we present a new optimistic fair exchange protocol based on key-exposure-free chameleon hashing scheme [14]. Before describing the new scheme, we introduce some notations. ID_A , ID_B and ID_T represent Alice's, Bob's and TTP's identities, respectively. Label L is used to identify a specific protocol, and links all messages generated in this protocol. We also assume that the communication channel is secure. In practice, the secure channel can be realized through cryptographic techniques. Our proposed scheme consists of four procedures: **Initialization**, **Main exchange protocol (MEP)**, **Abortion sub-protocol (ASP)** and **Dispute resolution protocol (DRP)**. In the following, we describe the new OFEP as a sequence of rounds, and each round consists of multiple messages.

Initialization: Let t be a prime power, and $E(F_t)$ is an elliptic curve over finite field F_t . $\#E(F_t)$ denotes the number of points of $E(F_t)$, and P is a point of $E(F_t)$ with prime order q , where $q \mid \#E(F_t)$. G is a subgroup generated by P . Hash function $H : Z_q^* \times \{0, 1\}^* \rightarrow Z_q^*$ is cryptographic secure. Let $(\text{Sign}, \text{Ver})$ be any provably secure signature scheme. The TTP chooses two random elements $x, k \in Z_q^*$, and computes $Y = xP$, $K = kP$. The private trapdoor key is $TK = (x, k)$, and the public hash key is $HK = (Y, K)$. The TTP publishes $\text{params} = \{E(F_t), q, G, P, H, (\text{Sign}, \text{Ver}), HK\}$, while keeps TK secretly. Alice has key pair $(x_A, Y_A) = (x_A, x_AP)$, Bob has key pair $(x_B, Y_B) = (x_B, x_BP)$, where $x_A, x_B \in Z_q^*$.

Main exchange protocol (MEP): Alice and Bob want to exchange their signatures on the same message m . Let Alice be the initiator. The main process of the MEP is shown as follows:

- Step 1. Alice $\rightarrow$ Bob : $\{L, T, ID_A, ID_B, ID_T, (r_A, \sigma_A^*)\}$
- Step 2. Bob $\rightarrow$ Alice : $\{ID_A, ID_B, (r_B, \sigma_B^*)\}$
- Step 3. Alice $\rightarrow$ Bob : (r_A, σ_A)
- Step 4. Bob $\rightarrow$ Alice : (r_B, σ_B)

In detail, as a initiator, Alice chooses a deadline T which refers to the TTP's clock, selects a random number $r_A \in Z_q^*$, and computes $H_A^* = r_A Y$. Alice runs the signing algorithm Sign with the signing key x_A to sign the message H_A^* , denotes $\sigma_A^* = \text{Sign}_{x_A}(H_A^*)$. Then, Alice sends $\{L, T, ID_A, ID_B, ID_T, (r_A, \sigma_A^*)\}$ to Bob.

Upon receiving $\{L, T, ID_A, ID_B, ID_T, (r_A, \sigma_A^*)\}$, if Bob does not agree with the deadline T , he can stop the protocol. Otherwise, he uses Alice's public key Y_A to check the validity of (r_A, σ_A^*) . If $\text{Ver}_{Y_A}(\sigma_A^*) \neq r_A Y$, he does nothing, the exchange protocol ends without disputes. Otherwise, he selects a random number $r_B \in Z_q^*$, computes $H_B^* = r_B Y$, and runs the signing algorithm Sign with his private key x_B to sign the message H_B^* , denotes $\sigma_B^* = \text{Sign}_{x_B}(H_B^*)$. Then Bob sends $\{ID_A, ID_B, (r_B, \sigma_B^*)\}$ to Alice.

Upon receiving $\{ID_A, ID_B, (r_B, \sigma_B^*)\}$, Alice checks the validity of (r_B, σ_B^*) . If $Ver_{Y_B}(\sigma_B^*) = r_B Y$, Alice computes the chameleon hash value $H_A = Hash(L, m, r_A) = e(P + K) + r_A Y$ and its signature $\sigma_A = Sign_{x_A}(H_A)$ for label L , where $e = H(m, L || ID_A || ID_B || ID_T)$. Then, Alice sends (r_A, σ_A) to Bob. Otherwise, Alice invokes the ASP before the deadline.

Upon receiving (r_A, σ_A) , Bob checks the validity of (r_A, σ_A) using Alice's public key Y_A . If $Ver_{Y_A}(\sigma_A) = e(P + K) + r_A Y$, he computes the chameleon hash value $H_B = Hash(L, m, r_B) = e(P + K) + r_B Y$, and its signature $\sigma_B = Sign_{x_B}(H_B)$ for the label L , where $e = H(m, L || ID_A || ID_B || ID_T)$. Then, Bob sends (r_B, σ_B) to Alice. If Bob does not receive (r_A, σ_A) or only receives an invalid (r_A, σ_A) , he executes the DRP before the deadline.

Upon receiving (r_B, σ_B) , Alice checks the validity of (r_B, σ_B) using Bob's public key Y_B , if $Ver_{Y_B}(\sigma_B) = e(P + K) + r_B Y$, the protocol ends with success. If Alice does not receive (r_B, σ_B) or only receives an invalid (r_B, σ_B) , she executes the DRP before the deadline.

Abortion sub-protocol (ASP): If Alice claims that she does not receive the message or receives an invalid message from Bob after Step 1 of the MEP, she can execute the ASP. The main process of the ASP is shown as follows:

Step a. Alice $\rightarrow$ TTP : $\{L, T, ID_A, ID_B, ID_T, (r_A, \sigma_A^*), Sign_{x_A}("Abort the exchange")\}$

Step b.

IF (Request is invalid)

TTP $\rightarrow$ Alice : $Sign_x("Request is invalid")$

ELSE IF (State=DRP)

TTP $\rightarrow$ Alice : (r_B, σ_B) and

TTP $\rightarrow$ Bob : (r_A^*, σ_A^*)

ELSE

TTP $\rightarrow$ Alice and Bob : $Sign_x(L, ID_A, ID_B, ID_T, "Exchange aborted")$

In detail, if Alice wants to abort the exchange, she must send the message $\{L, T, ID_A, ID_B, ID_T, (r_A, \sigma_A^*), Sign_{x_A}("Abort the exchange")\}$ to the TTP. The TTP checks its validity, if any validation fails, the TTP sends the message "Request is invalid" to Alice. If the TTP is at the state of the dispute resolution, the TTP sends their signatures (r_A^*, σ_A^*) and (r_B, σ_B) to each other. Otherwise, the TTP sends the message "Exchange aborted" to Alice and Bob.

Dispute resolution protocol (DRP): If Alice or Bob claims that she/he does not receive the message or receives an invalid message from the other party, she/he can execute the DRP. The main process of the DRP is shown as follows:

Case 1: Alice executes the DRP.

Step a. Alice $\rightarrow$ TTP : $\{L, T, ID_A, ID_B, ID_T, (r_B, \sigma_B^*), (r_A, \sigma_A), Sign_{x_A}("Recovery")\}$

Step b.

IF (Request is invalid)

TTP $\rightarrow$ Alice : $Sign_x("Request is invalid")$

ELSE IF (State=Aborted)

TTP $\rightarrow$ Alice : $Sign_x(L, ID_A, ID_B, ID_T, "Exchange aborted")$

ELSE

TTP $\rightarrow$ Alice : (r_B^*, σ_B^*) and

TTP $\rightarrow$ Bob : (r_A, σ_A)

In detail, if Alice executes the dispute resolution protocol, Alice must send the message $\{L, T, ID_A, ID_B, ID_T, (r_B, \sigma_B^*), (r_A, \sigma_A), Sign_{x_A}("Recovery")\}$ to the TTP. The TTP checks its validity, if the time exceeds the deadline T , or $Ver_{Y_A}(\sigma_A) \neq e(P + K) + r_A Y$, or $Ver_{Y_B}(\sigma_B^*) \neq r_B Y$, the TTP sends the message "Request is invalid" to Alice. If the TTP is at the state of the ASP, the TTP sends the message "Exchange aborted" to Alice. Otherwise, the TTP computes $r_B^* = r_B - e(x^{-1} + kx^{-1}) \bmod q$, then, sends (r_B^*, σ_B^*) and (r_A, σ_A) to Alice and Bob, respectively.

Notice that

$$\begin{aligned} & Hash(L, m, r_B^*) \\ &= e(P + K) + r_B^* Y \\ &= e(P + K) + r_B Y - e(x^{-1} + kx^{-1}) Y \\ &= e(P + K) + r_B Y - e(P + K) \\ &= r_B Y \\ &= H_B^* \end{aligned}$$

Case 2: Bob executes the DRP.

Step a. Bob $\rightarrow$ TTP : $\{L, T, ID_A, ID_B, ID_T, (r_A, \sigma_A^*), (r_B, \sigma_B), Sign_{x_B}("Recovery")\}$

Step b.

IF (Request is invalid)

TTP $\rightarrow$ Bob : $Sign_x("Request is invalid")$

ELSE IF (State=Aborted)

TTP $\rightarrow$ Alice : $Sign_x(L, ID_A, ID_B, ID_T, "Exchange aborted")$

ELSE

TTP $\rightarrow$ Alice : (r_B, σ_B) and

TTP $\rightarrow$ Bob : (r_A^*, σ_A^*)

In detail, if Bob executes the dispute resolution protocol, Bob must send the message $\{L, T, ID_A, ID_B, ID_T, (r_A, \sigma_A^*), (r_B, \sigma_B), Sign_{x_B}("Recovery")\}$ to the TTP. The TTP checks its validity, if the time exceeds the deadline T , $Ver_{Y_B}(\sigma_B) \neq e(P + K) + r_B Y$, or $Ver_{Y_A}(\sigma_A^*) \neq r_A Y$, the TTP sends the message "Request is invalid" to Alice. If the TTP is at the state of the ASP, the TTP sends the message "Exchange aborted" to Bob. Otherwise, the TTP computes $r_A^* = r_A - e(x^{-1} + kx^{-1}) \bmod q$, then, sends (r_B, σ_B) and (r_A^*, σ_A^*) to Alice and Bob, respectively.

Notice that

$$\begin{aligned}
& \text{Hash}(L, m, r_A^*) \\
&= e(P + K) + r_A^* Y \\
&= e(P + K) + r_A Y - e(x^{-1} + kx^{-1})Y \\
&= e(P + K) + r_A Y - e(P + K) \\
&= r_A Y \\
&= H_A^*
\end{aligned}$$

IV. SECURITY DISCUSSIONS

The kernel of the new optimistic fair exchange protocol is the KEFCH scheme. So, the security of the new OFEP follows directly from Theorem 1.

Theorem 1. *The key-exposure-free chameleon hashing scheme is secure under the assumption of the discrete logarithm problem (DLP) in G is intractable.*

Proof: The scheme satisfies the properties: collision resistance, semantic security, key exposure freeness and message hiding. We refer the reader to [9, Section 3] for the concrete process.

The new optimistic fair exchange protocol satisfies the following features:

Theorem 2. *The new OFEP satisfies the effectiveness requirement.*

Proof: We assume that both Alice and Bob are honest, thus they will follow the MEP properly. They can obtain each other's signature without the involvement of the TTP when the exchange protocol ends. So, the new OFEP satisfies the effectiveness requirement.

Definition 1. The new OFEP is fair if it guarantees that either both parties obtain each other's data or none of them get anything useful when the protocol ends.

Theorem 3. *The new OFEP satisfies the fairness requirement.*

Proof: The proof can be completed by the following steps.

There are three possible cases that Alice can get Bob's signature. Firstly, Alice and Bob perform properly, Alice can obtain Bob's signature at step 4. Secondly, Alice executes the ASP, if the TTP is at the state of the dispute resolution, Alice can obtain Bob's signature from the TTP. Thirdly, Alice executes the DRP, and must send $\{L, T, ID_A, ID_B, ID_T, (r_B, \sigma_B^*), (r_A, \sigma_A)\}$ to the TTP. However, according to the DPR, the TTP will compute (r_B^*, σ_B^*) for Alice only if $\{L, T, ID_A, ID_B, ID_T, (r_B, \sigma_B^*), (r_A, \sigma_A)\}$ are valid. Then, the TTP will forward (r_B^*, σ_B^*) to Alice.

There are two possible cases that Bob can get Alice's signature. Firstly, Bob can obtain Alice's signature after the step 3, while they perform properly. Secondly, Bob executes the DPR, and must send $\{L, T, ID_A, ID_B, ID_T, (r_A, \sigma_A^*), (r_B, \sigma_B)\}$ to the TTP. However, according to the DPR, the TTP will compute (r_A^*, σ_A^*) for Bob only if Bob gives correct $\{L, T, ID_A, ID_B, ID_T, (r_A, \sigma_A^*), (r_B, \sigma_B)\}$ to the TTP. Then, the TTP will forward (r_A^*, σ_A^*) to Bob.

According to the definition 1 and the proof above, the new OFEP is fair.

Definition 2. (Timeliness) Each party can finish the protocol at any time without loss of fairness.

Theorem 4. *The new OFEP satisfies the timeliness requirement.*

Proof: Timeliness refers to the protocol to be ended in a limited time, and does not undermine the fairness. Alice and Bob can contact the TTP and finish the execution of the exchange at any moment. According to the new OFEP, Alice can conclude the protocol in one of the two ways:

- executes the ASP to abort the exchange before step 3.
- executes the DRP to settle the argument before the deadline T .

Bob can conclude the protocol in one of the two ways:

- stops at any time before step 2.
- executes the DRP to settle the argument before the deadline T .

To sum up, the new OFEP satisfies the timeliness requirement.

Theorem 5. *The new OFEP satisfies the non-repudiation requirement.*

Proof: According to the **Initialization** of the new OFEP, we assume that $(\text{Sign}, \text{Ver})$ be any provably secure signature scheme. With this signature scheme, not only any third party can not forge the signatures of Alice and Bob, but also Alice and Bob can not deny their signatures that they really signed. So, the new OFEP satisfies the non-repudiation requirement.

V. CONCLUSION

According to Yang et al.'s OFEP [14], Bob can execute the DRP to finish the exchange, but Alice can not execute the DRP at any moment even in case of not receiving Bob's signature. In fact, the authors stated that Alice does not need to execute the DRP. Unfortunately, this kind of resolution has two weaknesses. Consider the following scenarios: (I) after step 1 of the YYK scheme, Bob has the advantage of carrying on the exchange at a time of his choice (e.g., Bob does not send his signature to Alice, or he delays the exchange intentionally). Alice may have to wait indefinitely. Because, the protocol does not ensure a deadline for Bob to contact the TTP; (II) if Bob sends an incorrect signature to Alice, Alice can detect this, and halts the protocol. Then, Bob may execute the DRP. The TTP would accept the request, and recover Alice's signature to Bob. This may lead to an unfair situation, because Bob can obtain Alice's signature, but Alice can not.

In this paper, we proposed a new optimistic fair exchange protocol with timeliness based on the key-exposure-free chameleon hashing scheme [14]. The new protocol overcomes those weaknesses. So, both parties can contact the

TTP and solve the exchange whenever they want before the deadline. Furthermore, the new OFEP achieves fairness and timeliness.

ACKNOWLEDGMENT

This work is supported by the National Basic Research Program of China (973 Program) (No. 2007CB310704), the National Natural Science Foundation of China (No. 90718001, 60821001), and the National 863 (2008AA011004).

REFERENCES

- [1] S. Even, O. Goldreich and A. Lempel. "A randomized protocol for signing contracts," *Communications of the ACM*, ACM Press, 28(6): 637-647, 1985.
- [2] M. Blum. "How to exchange (secret) keys," *ACM Transactions on Computer Systems*, ACM Press, pp. 175-193, 1983.
- [3] M. K. Franklin and M. K. Reiter. "Fair exchange with a semi-trusted third party (extended abstract)," *Proceedings of the 4th ACM conference on Computer and communications security*, April 1-4, 1997, Zurich, Switz. pp. 1-5, 1997.
- [4] N. Asokan, V. Shoup V and M. Waidner. "Optimistic fair exchange of digital signatures," *IEEE Journal on Selected Areas in communications*, pp. 593-610, 2000.
- [5] N. Asokan, M. Schunter and M. Waidner. "Optimistic protocols for fair exchange," in *Proc. of the 4th ACM Conference on Computer and communications Security*, April 1-4, 1997, Zurich, Switz. ACM Press, pp. 7-17, 1997.
- [6] J. Camenisch and I. B. Damgård. "Verifiable encryption, group encryption, and their applications to group signatures and signature sharing schemes," in *Proc. of ASIACRYPT 2000*, December 3-7, 2000, Kyoto, Japan. Springer-Verlag, LNCS 1976, pp. 331-345, 2000.
- [7] J. M. Park, E. K. P. Chong and P. J. Siegel. "Constructing fair exchange protocols for e-commerce via distributed computation," in *Proc. of 22th Annual ACM Symposium on Principles of Distributed Computing (PODC'03)*, July 13-16, 2003, Boston, MA, United states. ACM Press, pp. 172-181, 2003.
- [8] Y. Dodis and L. Reyzin. "Breaking and repairing optimistic fair exchange from PODC 2003," in *Proc. of the 3rd ACM workshop on Digital Rights Management*, Oct 27, 2003, Washington, DC, United states. ACM Press, pp. 47-54, 2003.
- [9] Y. Okada, Y. Manabe and T. Okamoto. "Optimistic Fair Exchange Protocol for E-Commerce," *The 2006 Symposium on Cryptography and Information Security*. Hiroshima, Japan, 2006.
- [10] I. Ray and H. J. Zhang. "Experiences in developing a fair-exchange e-commerce protocol using common off-the-self components," *Electronic Commerce Research and Applications*, Vol. 7, No. 2, pp. 247-259, 2008.
- [11] Y. Okada, Y. Manabe and T. Okamoto. "An optimistic fair exchange protocol and its security in the universal composability framework," *International Journal of Applied Cryptography*, Vol.1, No. 1, pp. 70-77, 2008.
- [12] Z. H. Shao. "Fair exchange protocol of signatures based on aggregate signatures," *Computer communications*, Vol. 31, No. 10, pp. 1961-1969, 2008.
- [13] W. Gao, F. Li and B. H. Xu. "An Abuse-Free Optimistic Fair Exchange Protocol Based on BLS Signature," in *Proc. of CIS2008*. Vols 1 and 2, pp. 841-845, Dec. 2008.
- [14] X. Yang, Z. P. Yu and B. Kang. "Chameleon-based optimistic fair exchange protocol," in *Proc. of the International Conference on Embedded Software and Systems*, July 29-31, 2008. Chengdu, Sichuan, China. IEEE Computer Society, pp. 298-302, 2008.
- [15] J. L. Hernandez-Ardieta, A. I. Gonzalez-Tablas and B. Ramos Alvarez. "An optimistic fair exchange protocol based on signature policies," *Computers and Security*, pp. 309-322, 2008.
- [16] F. R. Piva, J. R. M. Monteiro and R. Dahab. "Regarding timeliness in the context of fair exchange," *International Conference on Network and Service Security (N2S'09)*, Oct 19-21, 2009, Gold Coast, Australia. IEEE Computer Society Press, pp. 1-6, 2009.