1-1-2005

An efficient certified e-mail scheme suitable for wireless mobile environments

Guilin Wang
Institute for Infocomm Research, Singapore, guilin@uow.edu.au

Feng Bao
Institute for Infocomm Research, Singapore

Jianying Zhou
Institute for Infocomm Research, Singapore

Robert H. Deng
Singapore Management University

An efficient certified e-mail scheme suitable for wireless mobile environments

Abstract

As security enhanced systems for standard e-mail, certified e-mail schemes guarantee the fair exchange of a digital message with the corresponding receipt between two mistrusted parties. That is, the intended receiver gets the e-mail content if and only if the e-mail sender obtains an irrefutable receipt issued by the receiver, which could be used to prove that the message has been delivered to the receiver. A number of such protocols have been proposed in recent years. However, most of them are not suitable for mobile networks, since many intricate cryptographic primitives are involved so that considerable overheads are introduced. In this paper, we present a novel simple protocol for certified e-mail delivery. Technical discussions are provided to show that our new solution is both secure and very efficient so that it is truly suitable for wireless mobile users, where the available devices usually have limited resources on computation, communication, storage, and power supply.

Keywords

mail, environments, mobile, scheme, efficient, wireless, certified, suitable, e

Disciplines

Physical Sciences and Mathematics

Publication Details

Wang, G., Bao, F., Zhou, J. & Deng, R. H. (2005). An efficient certified e-mail scheme suitable for wireless mobile environments. 2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2005 (pp. 1994-1998). USA: IEEE.

An Efficient Certified E-Mail Scheme Suitable for Wireless Mobile Environments

Guilin Wang*, Feng Bao*, Jianying Zhou*, and Robert H. Deng†

*Infocomm Security Department, Institute for Infocomm Research (I^2R)
21 Heng Mui Keng Terrace, Singapore 119613
{glwang, baofeng, jyzhou}@i2r.a-star.edu.sg
†School of Information Systems, Singapore Management University
469 Bukit Timah Road, Singapore 259756
robertdeng@smu.edu.sg

Abstract— **As security enhanced systems for standard e-mail, certified e-mail schemes guarantee the *fair* exchange of a digital message with the corresponding receipt between two mistrusted parties. That is, the intended receiver gets the e-mail content *if and only if* the e-mail sender obtains an irrefutable receipt issued by the receiver, which could be used to prove that the message has been delivered to the receiver. A number of such protocols have been proposed in recent years. However, most of them are not suitable for mobile networks, since many intricate cryptographic primitives are involved so that considerable overheads are introduced. In this paper, we present a novel simple protocol for certified e-mail delivery. Technical discussions are provided to show that our new solution is both secure and very efficient so that it is truly suitable for wireless mobile users, where the available devices usually have limited resources on computation, communication, storage, and power supply.**

Keywords: Certified e-mail, fair-exchange, digital signature, wireless mobile environment, information security.

I. Introduction

As a value-added service for standard e-mail systems, a certified e-mail scheme allows a sender Alice to deliver a digital message to a receiver Bob over the Internet in a *fair* way, i.e., either the sender Alice obtains an irrefutable receipt from the receiver and the receiver Bob can access the content of the e-mail simultaneously, or neither party gets the expected item. In other words, the main purpose of certified e-mail delivery is the fair-exchange of a message and a receipt between two potentially mistrusted parties. This property of fairness guarantees that a dishonest party cannot obtain his/her expected item from a honest party in any cheating way such that the honest party is unable to get the corresponding item.

After the completion of exchange, the sender Alice will hold undeniable evidence of receipt (EOR). Therefore, if the receiver Bob denies having received a specific message from Alice, Alice can provide the publicly verifiable receipt to an arbitrator to show that this claim is untrue. Similar security services are provided by non-repudiation protocols [8], [15].

As practice-oriented protocols, the implementation of certified e-mail delivery usually needs the help of a trusted third party (TTP) in the fashion of on-line or off-line. Previous schemes [6], [16], [1] focused on the use of on-line TTP, i.e., the TTP is needed for every message delivering. Consequently, those schemes are expensive and inefficient in practice, since the TTP is supposed to offer high quality services, and the TTP is likely to become the system bottleneck if numerous certified emails are exchanged via the same TTP every day.

A remarkable idea is to exploit the TTP in an off-line fashion, i.e., the TTP is involved *only* in abnormal situations, where one of the two parties misbehaves or the communication channel is out of order. That is, in normal situations the TTP is not introduced to the protocol execution at all. Furthermore, once the TTP is applied, it can help the victim to achieve fair results. Therefore, we could expect that the TTP will be involved *rarely* in a real system, since cheating is not beneficial to the cheater. Based on this observation, those schemes with off-line TTPs are called as *optimistic* [2]. Actually, most of researches in this area have focused on optimistic certified e-mail protocols [4], [9], [10], [11], [12].

However, all of the schemes mentioned above are not suitable for wireless mobile environments due to two reasons: efficiency and security. First of all, the wireless mobile environments impose certain restrictions on both computation and communication, due to the fact that mobile devices usually have limited resources of computation, communication, storage, and power supply. But those schemes are not optimized to reduce the number of asymmetric cryptographic operations that are required to complete a message delivery. Actually, most of those schemes [4], [10], [8], [12] need ten or more modular exponentiations, the most expensive operations in cryptography. The other two schemes proposed in [9], [11] are relatively simple, but they are vulnerable to an attack that allows the sender to cheat the receiver by mixing identities of different TTPs [15]. This attack is meaningful in practice since it seems *unreasonable* to assume that the receiver is aware of the existence of all TTPs and may contact each TTP individually for help.

Actually, to our knowledge, the PRCS scheme [12] is the unique protocol designed for users in mobile networks, but it is both insecure and inefficient. More specifically, we can

978-3-8007-2909-8/05/$20.00 ©2005 IEEE

mount an attack that enables the sender to get a valid receipt without delivering a message to the receiver. The reason is that the sender could derive a partial private key from several partial signatures, though it is supposed that this partial private key is known only by the receiver and the TTP. In addition, in the PRCS scheme eleven (11) asymmetric cryptographic operations are needed to complete a message delivery.

In this paper, we present a novel simple protocol for certified e-mail delivery suitable for wireless mobile environments. Technical discussions are provided to show that our new solution is both secure and very efficient so that it is truly suitable for wireless mobile users. Compared with the existing schemes, our protocol actually has a number of appealing features (check Table 1 for details).

The rest of this paper is organized as follows. In Section II, we present desirable efficiency and security requirements for certified e-mail schemes in wireless mobile networks. Then, such a new scheme is proposed in Section III. After that, we analyze the efficiency and security of this new protocol, and compare it with existing solutions in Section IV. Finally, Section V concludes the paper.

II. Requirements for Certified E-Mail Schemes

Different certified e-mail schemes may focus on different requirements on efficiency and security. In this section, we first set up our requirements for certified e-mail scheme suitable for wireless mobile environments, and then argue why those requirements are desirable in this scenario. Naturally, those requirements should reflect the essential characteristics of both certified e-mail and wireless mobile networks. Specifically, we aim to present a certified e-mail schemes to satisfy all of the following requirements.

- R1 **Off-line TTP**: The TTP is required to be involved in the protocol execution *only* in *abnormal cases*, i.e., one party is trying to cheat or the communication channel fails to work.
- R2 **Transparent TTP**: The generated non-repudiable receipt is the same regardless of whether the TTP is involved or not in the protocol execution.
- R3 **Stateless TTP**: To deal with potentially unfair situations, the TTP is not required to maintain and search a database that remembers the state information for each protocol instance.
- R4 **Generic Construction**: The receiver could exploit *any* secure standard digital signature algorithm to generate the irrefutable receipt.
- R5 **High Performance**: To execute the protocol, both overheads of computation and communication should be reduced to as low as possible.
- R6 **Fairness**: After the completion of a protocol run, either each party obtains the expected item or neither party gets any useful information about the other's item.
- R7 **Timely Termination**: Each involved party should be able to terminate the protocol *unilaterally* in a given finite time without losing fairness. This requirement is especially important if the message being delivered is time-sensitive.
- R8 **Confidentiality**: Except the sender Alice and the receiver Bob, the content of the delivered message cannot be accessed by anybody else, including the TTP.

In the above list, security-related requirements are enumerated as the last three items (i.e., from R6 to R8). We believe those are the most important security requirements for all certified e-mail schemes [4], [9], [10]. Therefore, we retain them as part of requirements for certified e-mail schemes in wireless mobile networks.

Here, we want to stress that other five properties (i.e., from R1 to R5) are very meaningful for certified e-mail schemes aimed for wireless mobile networks. First of all, the TTP's involvement and workload are minimal if a certified e-mail scheme supports off-line, transparent, and stateless TTP. This implies not only that the running cost of the TTP could be reduced accordingly, but also that the TTP's performance can be improved considerably. Because the TTP is only involved into the protocol execution rarely, and even in this case the TTP only needs to perform a few simple operations. Secondly, generic certified e-mail schemes are absolutely important since users in the real world almost inevitably exploit different digital signature algorithms. Finally, high performance is definitely desirable due to the resource limitations on wireless mobile devices. That is, we should design a certified email schemes such that both the computation and communication overheads are as less as possible, while the essential security properties are still satisfied.

III. The Proposed Certified E-Mail Scheme

Like most of existing solutions, our new certified e-mail scheme consists of an *exchange protocol*, a *recovery protocol*, and a *dispute resolution policy*. The dispute resolution policy exactly defines the receipt format and the procedures how a judge settles the potential dispute on repudiation of receipt. The exchange protocol is the main protocol that specifies the procedures for the sender Alice and the receiver Bob to fellow in normal situation. However, once the receiver Bob sent his receipt to Alice but does not get the session key from Alice correctly or timely, he could initiate the recovery protocol to get the key from the TTP directly (and then derive the message). This way assures that the sender Alice cannot cheat the receiver Bob. Before the descriptions of those components, we first introduce some notations and assumptions.

A. Notations and Assumptions

In this paper, we use A, B, and T to denote unique identifiers of a sender Alice, a receiver Bob, and a TTP, respectively. m is a message Alice wants to deliver to Bob. $c = E_k(m)$ is the ciphertext of message m encrypted with a symmetric encryption algorithm $E_k(\cdot)$, where k is a session key selected by the sender Alice. The corresponding symmetric decryption algorithm is denoted by $D_k(\cdot)$, i.e., we have $m = D_k(E_k(m))$. $E_T(\cdot)$ denotes the TTP's secure asymmetric encryption algorithm in the sense that the resulting ciphertexts

can only be decrypted by the TTP using its private key. $S_B(\cdot)$ denotes the receiver Bob's signing algorithm, which can be *any* secure digital signature algorithm in the sense that an attacker cannot forge a valid signature. In addition, $H(\cdot)$ stands for a cryptographic hash function such that it is infeasible to find two inputs with the same output. Naturally, all those algorithms are assumed to be publicly known, and all public keys of all parties are publicly available.

As usual, the communication channel between Alice and Bob is assumed to be *unreliable*, i.e., messages inserted into such a channel may be lost. However, the TTP is linked with Alice and Bob by *reliable* communication channels, i.e., messages inserted into such a channel will be delivered to the recipient after a finite delay. In additon, all of those channels are supposed to be confidential and authenticated. Actually, this service could be provided by the SSL (Secure Sockets Layer) or TLS (Transport layer Security) protocols.

B. Exchange Protocol

When a sender Alice wants to deliver a message m to the receiver Bob with a guaranteed receipt, they collectively execute the following exchange protocol.

(e1). $A \longrightarrow B: \quad A, B, T, H(k), c, t, EK = E_T(\ell, k)$
(e2). $B \longrightarrow A: \quad S_B(\ell, EK)$
(e3). $A \longrightarrow B: \quad k$

That is, Alice first chooses a session key k, and then calculates the following values: ciphertext $c = E_k(m)$, label $\ell = H(A, B, T, H(c), H(k), t)$, encrypted key $EK = E_T(\ell, k)$. After that, Alice sends message flow (e1) to Bob. Note that label ℓ means that message m is supposed to be delivered from the sender Alice to the receiver Bob (with/without the help of the trusted third party T), where m is determined by a ciphertext c and a symmetric key k such that $c = E_k(m)$, $\ell = H(A, B, T, H(c), H(k), t)$. Here, t denotes a deadline with the meaning that after the expiration of t, the TTP does not accept a resolution request related to this t anymore. Therefore, label ℓ can be used as a unique identifier to recognize a specific protocol instance, and link all messages generated in this instance.

Upon receiving message flow (e1), the receiver Bob first determines whether the included deadline t is sufficient for him to get the TTP's help (in case abnormal situations occur later). If the answer is negative, he could simply reject this email or require Alice re-execute the protocol by setting a new deadline. Otherwise, Bob recovers label ℓ, then generates his signature $S_B(\ell, EK)$ and sends it to Alice as message flow (e2). Note that in the above procedure, Bob cannot tell whether the encrypted key EK is correctly prepared. However, our protocol is designed to guarantee that if EK is inconsistent with the content determined by unique label ℓ, Bob's signature $S_B(\ell, EK)$ is useless for anybody (including the sender Alice). In such a condition, this particularly implies that $S_B(\ell, EK)$ cannot be interpreted as a valid receipt.

When message flow (e2) is received, Alice checks whether it is indeed Bob's valid signature on message (ℓ, EK). If this is true, Alice reveals the session key k to Bob. Finally, Bob checks whether $EK \equiv E_T(\ell, k)$. If this true, Bob could derive the message m by decrypting cipthertext c with session key k, i.e., $m = D_k(c)$. However, if Bob does not get correct k from Alice timely (due to Alice's malicious behavior or communication failure), he can execute the recovery protocol with the TTP (see below).

C. Recovery Protocol

Whenever before the expiration of deadline t, Bob could initiate the following recovery protocol to get the session key k from the TTP directly.

(r1). $B \longrightarrow T: \quad A, B, T, H(c), H(k), EK, t, S_B(\ell, EK)$
(r2). $T \longrightarrow B: \quad \ell, k$
$T \longrightarrow A: \quad \ell, S_B(\ell, EK)$

We now explain the above recovery protocol in detail as follows. First, Bob sends the TTP message flow (r1) as a recovery request. The TTP then recovers label ℓ, checks the validity of deadline t, and whether $S_B(\ell, EK)$ is Bob's valid signature on message (ℓ, EK). If any of the above verifications fails, the TTP rejects Bob's application. Otherwise, it decrypts EK with its private key. If the result is the expected pair (ℓ, k), i.e., the expected label ℓ concatenated by a random number k interpreted as a session key, the TTP forwards (ℓ, k) and $(\ell, S_B(\ell, EK))$ to Bob and Alice, respectively. However, if EK cannot be decrypted successfully or the decrypted result is incorrect, the TTP informs Bob that this EK is invalid. With this acknowledgement, Bob is free from taking any responsibility on the signature $S_B(\ell, EK)$ signed by himself, since this signature is actually an invalid receipt.

Remark 1. Note that in the above recovery protocol, the hash value $H(c)$, instead of the whole ciphertext c, is delivered to the TTP. This approach not only reduces the communication overhead between Bob and the TTP since c may be a huge digital file, but also prevents the TTP from deriving the message m from c and k. Because the TTP can just know the session key k, but cannot get the cipertext c, which is confidentially transferred from Alice to Bob.

D. Dispute Resolution Policy

Someday later, if the receiver Bob denies having received message m from Alice, then the sender Alice could provide Bob's receipt together with other relative information to show that Bob's claim is untrue. Namely, Alice can prove that Bob has already received message m. Specifically, Alice could provide $(A, B, T, m, k, t, S_B(\ell, EK))$ to a judge (or any verifier). Then, the judge performs as follows:

- Compute $c = E_k(m)$, $\ell = H(A, B, T, H(c), H(k), t)$, and $EK = E_T(\ell, k)$.
- Check whether $S_B(\ell, EK)$ is Bob's valid signature on message (ℓ, EK). If yes, accept Alice's claim. Otherwise, reject Alice's claim.

Remark 2. In the above protocol description, we treat the TTP's public encryption algorithm $E_T(\cdot)$ as a determined

TABLE I
COMPARISON OF EFFICIENCY AND SECURITY

	Off-line TTP	Transparent TTP	Stateless TTP	Generic Scheme	No. of Messages	No. of Operations	Fairness	Timely Termination	Confidentiality
AN [4]	Yes	Yes	Yes	No	4	17	Yes	No	Yes
KM [10]	Yes	No	No	Yes	4	10	Yes	Yes	Yes
GRV [8]	Yes	No	No	Yes	4	10	Yes	Yes	Yes
PRCS [12]	Yes	Yes	Yes	No	4	11	Yes*	Yes	Yes
IS [9]	Yes	Yes	Yes	Yes	3	8	Yes*	No	Yes
Micali [11]	Yes	Yes	Yes	Yes	3	8	Yes*	Yes	Yes
Ours	**Yes**	**Yes**	**Yes**	**Yes**	**3**	**4**	**Yes**	**Yes**	**Yes**

algorithm. Actually, $E_T(\cdot)$ could also be chosen as a randomized algorithm with *randomness recoverability* [15]. That is, a random number r will be picked to encrypt message (ℓ, k), denoted by $EK = E_T^r(\ell, k)$; and furthermore, the TTP can recover message (ℓ, k) as well as randomness r from EK by using its private key. As pointed out in [11], this is generally true for most provably secure public key cryptosystems, such as the OAEP series of encryption schemes [14]. Naturally, if such a randomized algorithm is exploited, we suppose that Alice and the TTP will reveal randomness r (together with the session key k) to Bob in the exchange protocol and recovery protocol, respectively. Similarly, in this case Alice is also assumed to provide randomness r to the judge when she requests for dispute resolution.

IV. EVALUATION AND COMPARISON

Now, we argue that the proposed certified e-mail protocol meets all the desirable requirements listed in Section II. At the same time, we compare our protocol with other existing solutions with *off-line* TTPs. The comparison result is summarized in Table 1.

R1 Off-line TTP. From the previous protocol specification, it is obvious that in the normal case, i.e., both involved parties are honest and the communication channel is in order, Alice can get a valid receipt $S_B(\ell, EK)$ from Bob and Bob could access the message m by computing $m = D_k(c)$. Namely, in the normal situation, to deliver a message only the exchange protocol will be executed and the TTP is not involved at all. So, our protocol supports off-line TTP. In other words, it is an optimistic protocol.

R2 Transparent TTP. In our protocol, the TTP's responsibility is to check the validity of a recovery request. If such a request is valid, it further decrypts EK and then sends the session key k to the applicant. Therefore, the format of receipt is the same regardless of whether the TTP is asked to deal with a recovery request. That is, our protocol meets the property of transparent TTP.

R3 Stateless TTP. In our recovery protocol, the TTP is not required to store any information about a specific recovery request. Actually, what the TTP needs to remember is just its decryption private key. So, as we claimed before, in our protocol the TTP is stateless.

R4 Generic Construction. Clearly, the proposed protocol is generic construction, since the receiver Bob could exploit *any* secure standard signature algorithm to generate his receipt $S_B(\ell, EK)$. The AN scheme and PRCS scheme, however, require a receiver to use a specific signature algorithm, i.e., the RSA signature [13] and the GQ signature [7], respectively.

R5 High Performance. As the main protocol, our exchange protocol is very efficient on aspects of both computation and communication. More specifically, in our exchange protocol, to complete the message delivery 3 message flows are transferred between Alice and Bob. In most of existing solutions, 4 message flows are needed. On the other hand, to run our exchange protocol, Alice and Bob need to perform 4 main computations: (a) Encrypt and verify the encrypted session key EK under the TTP's public key; and (b) Sign and verify the receipt $S_B(\ell, EK)$. Our exchange protocol hence requires 4 asymmetrically cryptographic operations. However, this number representing computation cost varies from 8 to 17 in other schemes. Note that in this comparison, we do not consider the computation cost of hash function evaluation, symmetric encryption and decryption, since those operations are much faster than asymmetric operations.

In fact, our protocol could become surprisingly efficient if specific algorithms are exploited. For example, let both the TTP and the receiver Bob have RSA cryptosystems with 1200-bit modulus, and their public key be short exponents, e.g., 3, 17 or $625537 = 2^{16} + 1$. In this case, the sender Alice only needs to compute less than 34 modular multiplications, while the receiver Bob has to perform 1834 modular multiplications on average. This setting is especially good for a mobile sender Alice. We remark that 1834 modular multiplications are also not a real burden for a mobile receiver Bob, since this overhead is just equivalent to produce a standard RSA signature.

Note that in the above performance comparison, we just compare our exchange protocol with those of existing schemes, but do not discuss the overheads of recovery protocols. This is because recovery protocols are expected to be run *occasionally* in abnormal cases, as we noticed before.

R6 Fairness. Now, we discuss the fairness, the most important security requirement for all certified e-mail protocols. We need to show that any of the two involved parties cannot cheat the other in dishonest ways. Our discussion is classified into two cases.

Case 1. Alice is honest, but Bob is trying to cheat. Since Alice is honest, the message flow (e1) is correctly prepared. So, dishonest receiver Bob has to figure out a way to get the session key k without issuing the receipt $S_B(\ell, EK)$. However, k is securely encrypted under the TTP's public key, so Bob (maybe colluding with his conspirators other than Alice and the TTP) cannot derive k from the EK directly. Therefore, to get the value of k, Bob has to reveal his signature $S_B(\ell, EK)$ to Bob or the TTP *before* the deadline t. In this situation, Bob could drive message m by calculating $m = D_k(m)$, but Alice also gets valid receipt $S_B(\ell, EK)$ from Bob or the TTP. The result is hence fair for both parties. On the other hand, if Bob does not successfully apply for recovery before the expiration of deadline t, this protocol run is deemed to be cancelled. In this situation, neither Alice nor Bob gets their expected items, so the result is still fair.

Case 2. Bob is honest, but Alice is trying to cheat. In this case, the sender Alice may dishonestly prepare message flow (e1). For example, she could send an incorrect ciphertext c, improperly commit $H(k)$, and/or select a random number for EK, etc. But the honest receiver Bob cannot find such potential inconsistencies in message flow (e1), so he will return his signature $S_B(\ell, EK)$ to Alice after computing label $\ell = H(A, B, T, H(c), H(k), t)$ if t is long enough for him. So Alice will get valid $S_B(\ell, EK)$ but Bob cannot access a valid message. However, the point is that in this situation, $S_B(\ell, EK)$ cannot be interpreted as a valid receipt according to our dispute resolution policy. This means that the result is also fair: neither party gets the expected item. Therefore, to get a valid receipt Alice has to properly prepare and send message flow (e1) to Bob. This implies that the last chance for Alice to cheat Bob is by refusing to reveal Bob the session key k after getting valid $S_B(\ell, EK)$ from Bob. However, as we mentioned before, this noncooperation cannot harm the receiver Bob at all, since he can get the value of k from the TTP directly.

In contrast, the original schemes in [9], [11], [12] are actually unfair, as we reviewed in Introduction. Under the fairness column in Table 1, we mark those schemes with "Yes*" since they could be made fair by some *proper* modifications.

R7 Timely Termination. In addition, our protocol also respects the property of timely termination due to the usage of deadline t. That is, Bob could apply recovery help at any time before the deadline, though Alice may need to wait the receipt $S_B(\ell, EK)$ until the expiration of deadline t.

R8 Confidentiality. In our protocol, each communication channel between any two parties is assumed to be confidential and authenticated. So, neither the TTP nor other outsiders can get the ciphertext c, which is confidentially transferred to Bob by Alice. Consequently, except the sender Alice and the receiver Bob, anybody else (including the TTP) cannot access the content of the delivered message m, though the TTP may have the knowledge of the session key k (if Bob applied for recovery). In other words, our certified e-mail protocol satisfies the property of *confidentiality*.

V. Conclusion

In this paper, we proposed a novel certified e-mail scheme suitable for wireless mobile environments, where the exploited devices usually have limited resources on computation, communication, storage, and energy supply. Technical discussions were provided to show that our new protocol is not only secure and but also very efficient. Compared with existing solutions, our scheme supports a number of desirable properties simultaneously.

References

[1] M. Abadi, N. Glew, B. Horne, and B. Pinkas. Certified email with a light on-line trusted third party: Design and implementation. In: *Proc. of 2002 International World Wide Web Conference (WWW'02)*, pp. 387-395. ACM press, 2002.

[2] N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18 (4): 591-606, 2000.

[3] G. Ateniese. Efficient verifiable encryption (and fair exchange) of digital signature. In: *Proc. of AMC Conference on Computer and Communications Security (CCS'99)*, pp. 138-146. ACM Press, 1999.

[4] G. Ateniese and C. Nita-Rotaru. Stateless-receipient certified E-mail system based on verifiable encryption. In: *CT-RSA'02*, LNCS 2271, pp. 182-199. Springer-Verlag, 2002.

[5] F. Bao, R.H. Deng, and W. Mao. Efficient and practical fair exchange protocols with off-line TTP. In: *Proc. of IEEE Symposium on Security and Privacy*, pp. 77-85, 1998.

[6] R. Deng, L. Gong, A. Lazar, and W. Wang. Practical protocol for certified electronic mail. *Journal of Network and Systems Management*, 1996, 4(3): 279-297.

[7] L.C. Guillou and J.J. Quisquater. A "paradoxical" identity-based signature scheme resulting from zero-knowledge. In: *CRYPTO'88*, pp. 216231. Spinger-Verlag, 1988.

[8] S. Gürgens, C. Rudolph, and H. Vogt. On the security of fair non-repudiation protocols. In: *Information Security Conference (ISC 2003)*, LNCS 2851, pp. 193-207. Springer-Verlag, 2003.

[9] K. Imamoto and K. Sakurai. A cerified e-mail system with receiver's selective usage of delivery authortiy. In: *Indocrypt 2002*, LNCS 2551, pp. 326-338. Springer-Verlag, 2002.

[10] S. Kremer and O. Markowitch. Selective receipt in cerified e-mail. In: *Indocrypt 2001*, LNCS 2247, pp. 136-148. Springer-Verlag, 2001.

[11] S. Micali. Simple and fast optimistic protocols for fair electronic exchange. In: *Proc. of 22th Annual ACM Symp. on Principles of Distributed Computing (PODC'03)*, pp. 12-19. ACM Press, 2003.

[12] J.M. Park, I. Ray, E.K.P. Chong, and H.J. Siegel. A certified e-mail protocol suitable for mobile environments. In: *Proc. of the 2003 IEEE Global Telecommunications Conference (GLOBECOM 2003)*, pp. 1394-1398. IEEE Communication Society, 2003.

[13] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, Feb. 1978, 21(2): 120-126.

[14] V. Shoup. OAEP reconsidered. *Journal of Cryptology*, 15(4): 223-249, 2002.

[15] G. Wang. Generic fair non-repudiation protocols with transparent off-line TTP. In: *Proc. of the 4th International Workshop for Applied PKI (IWAP'05)*. IOS press, 2005 (to appear).

[16] J. Zhou and D. Gollmann. Certified electronic mail. In: *Computer Security - ESORICS'96*, LNCS 1146, pp. 160-171. Springer-Verlag, 1996.