IEEE 6th CAS Symp. on Emerging Technologies: Mobile and Wireless Comm.
Shanghai, China, May 31~June 2, 2004

# Achieving Fairness in Wireless Environment

Hao Wang, Heqing Guo
Computer Science &Engineering Dept., South China University of Tech.,
Guangzhou, Guangdong, China

***Abstract*-In this paper, we present a fair non-repudiation protocol as an early effort to achieve fairness in the wireless environment. We propose a pseudo-resilient channel to satisfy the common assumption of fair protocols. And we use RSA-based convertible signature schemes and non-interactive zero-knowledge proofs to reduce bandwidth and computation consumptions.**

***Keyword*-Fairness, Non-repudiation, Wireless Environment**

## I. Introduction

### *1.1 Fairness in Electronic Transactions*

The goal of fair services is to guarantee fairness in an electronic transaction, that is, no party can falsely deny involvement in the transaction or having sent/received the specific message. Fairness issue has been studied in different scenarios: fair exchange [1][3][10], contract signing [4], certified e-mail [6], non-repudiable message transmission [13][9], and so on. And these protocols are inter-transformable, e.g., a fair exchange protocol can be easily transformed to be a contract signing protocol. To focus our attention on those critical parts, we choose the relatively simple form of fair protocols, i.e. the non-repudiable protocol, to describe our contribution to the fairness issue.

Main requirements for fair protocols are effectiveness, fairness, timeliness, non-repudiability, and the newly introduced publicity [9]. The generic protocol proposed by [2][8] (AK protocol) and used by [9][10] is the now-existing best basic model to construct a fair protocol.

To achieve fairness, a trusted third party (TTP) will be involved when an error occurs. And to achieve good publicity, transparent TTP [2][5][10] is introduced. Transparent TTP generates exactly the same evidences as the sender or the recipient. In this way, judging the outcome evidence cannot decide whether the TTP has intervened the protocol run. An efficient approach to achieve transparent TTP is using convertible signatures. Boyd and Foo [5] have proposed a fair payment protocol that can generates standard RSA signature as final evidence. But it is not efficient and practical enough because it involves an interactive verification process. Markowitch and Kremer [9][10] have proposed quite effective fair protocols. But they cannot generate standard signatures as final evidences, which make it necessary to adopt new algorithms. And thus make it hard to fit into the general electronic transaction practices.

### *1.2 Fair Protocol in Wireless Environment*

Electronic transactions have been widely used in wired network. And it begins to attract attention from the wireless world. But many efficient applications in wired network meet unexpected problems in wireless network. Wireless communication is characterized by lower bandwidths, higher error rates, and more frequent spurious disconnections [7]. These constraints make the wireless channel unreliable.

Fair protocols can work properly when the channel between sender and recipient is unreliable, but they assume resilient channels between sender/recipient and TTP, that is, the message in the channel will finally arrive at the recipient. So we need to build resilient channel in wireless network either by improving the reliability of protocols in lower levels or revising the fair protocols themselves.

### *1.3 Our Work*

To achieve resilient channel, we revise the fair protocol by proposing a pseudo-resilient channel using cyclic resending method. We present a practical fair non-repudiation protocol with transparent TTP using an adaptation of the convertible signature scheme proposed by Mao et al. [11].

To reduce communication and computation consumptions, we make several improvements. First, The original signature scheme uses an interactive verification protocol that is not efficient for wireless connections. So we replace it by a non-interactive verifying approach developed by ourselves. Second, our protocol generates standard RSA signatures as non-repudiation evidences. We believe that using standard signatures is more efficient because it has been well applied and it can help our protocol easily be integrated with other security systems.

### *1.4 Paper Organization*

In section 2, we list preliminaries terms and notations. Section 3 presents the improved convertible signature scheme. The non-repudiation protocol is presented in section 4. Section 6 gives some concluding remarks.

## II. Preliminaries

### *2.1 Pseudo-resilient Channel*

We propose pseudo-resilient channel using cyclic resending method. In our channel, the sender will cyclically resend the message until the recipient's affirmative arrives. The size of the affirmative message is small enough, so we ignore the probability of losing it. In this way, we claim our channel is pseudo-resilient. (Formal definition is omitted).

### *2.2 Notation*

We use these notations through out the paper.

- $X \rightarrow Y$: transmission from entity $X$ to $Y$
- $X \rightleftarrows Y$: pseudo-resilient transmission from $X$ to $Y$
- $h()$: a collision resistant one-way hash function
- $E_k()$: a symmetric-key encryption function under key $k$
- $D_k()$: a symmetric-key decryption function under key $k$
- $E_X()$: a public-key encryption function under $pk_X$

- $D_X()$: a public-key decryption function under $sk_X$
- $S_X()$: ordinary signature function of $X$
- $PS_X()$: phaseal signature function of $X$
- $FS_X()$: the final signature function of entity $X$
- $m$: the message sent from $A$ to $B$
- $k$: the session key $A$ uses to cipher $m$
- $pk_X$: public key of $X$
- $sk_X$: secret key of $X$
- $cipher = E_k(m)$: the cipher of $m$ under $k$
- $l = h(m,k)$: a label that in conjunction with $(A,B)$ uniquely identifies a protocol run
- $f$: a flag indicating the purpose of a message

## 2.3 Evidences

During the protocol, following evidence are generated:

- The evidence of origin: **EOO** = $PS_A(f_{\mathbf{EOO}}, B, \text{TTP}, l, h(cipher), h(k))$
- The evidence of receipt: **EOR** = $PS_B(f_{\mathbf{EOR}}, A, \text{TTP}, l, h(cipher), h(k))$
- The non-repudiation evidence of origin: **NRO** = $FS_A(f_{\mathbf{NRO}}, B, \text{TTP}, l, h(cipher), h(k))$
- The non-repudiation evidence of receipt: **NRR** = $FS_B(f_{\mathbf{NRR}}, A, \text{TTP}, l, h(cipher), h(k))$
- The evidence of submission key $k$: **Sub** = $S_A(f_{\mathbf{Sub}}, B, l, E_{TTP}(k))$
- The recovery request: **Rec**$_X$ = $S_X(f_{\mathbf{RecX}}, Y, l)$
- The abort request: **Abort** = $S_A(f_{\mathbf{Abort}}, TTP, l)$
- The abort confirmation: **Con**$_a$ = $S_{TTP}(f_{\mathbf{cona}}, A, B, l)$

## III. An Improved RSA-Based Convertible Signature Scheme

In this section, we describe an improved version of the RSA-based convertible signature scheme presented by Mao et al. [11]. The obvious drawback of the original scheme is the interactive verification protocol. As discussed in our former sections, it will cause communication and computation bottleneck. So we replaced it with a non-interactive verification protocol developed by ourselves. Let Alice be the signature signer and Bob be the signature verifier.

Let n be the Alice's RSA modulus. $n$ is a composite integer relatively prime to $\phi$ $(n)$. Alice chooses three integers de-noted by $c$, d and e satisfying:

$$cde \equiv 1(\bmod\ \phi(n)n)$$

and

$$de \neq 1(\bmod(\phi(n)))$$

Her public key is the pair $(e, n)$ and private key is $d$. $c$ is the secret key shared between Alice and TTP, and will be used to convert the phaseal signature to a final one. $c, d, e$ also satisfy:

$$\forall m < n^2 : m^{cde} \equiv m(\bmod\ n^2)$$

and

$$\forall m < n : m^{cde} \equiv m(\bmod\ n)$$

### 3.1 Registration

Before signing any signature, Alice needs to register with the TTP to get her certificate and share some common parameters. The registration protocol needs to be run only once and the resulting certificate can be used for any number of message transmissions.

Alice requests for key registration by sending her public key pair $(e, n)$ and $c$ to the TTP (for security, $c$ should be encrypted by $pk_{TTP}$). TTP checks the validity of $n$ (using the function denoted by $checkn()$), if passes, he sends a random number $\omega < n$ as the reference message. $\omega$ satisfies $gcd(\omega, n) = 1$ and $gcd(\omega \pm 1, n) = 1$. Alice then computes $PS(\omega) = \omega^d$ and send it to TTP. After TTP checks (using the function denoted by $check\,\omega\,()$) whether

$$\omega \equiv PS(\omega)^{ce} (\bmod\ n^2)$$

If it holds, he will send a certificate $cert_A = S_{TTP}(A, c, n, \omega, PS(\omega))$ to Alice.

Registration Protocol

1) $A \rightleftarrows TTP$: $f_{\mathbf{Reg}}, pk_X, E_{TTP}(c)$
 **if not** $checkn(n)$ **then** stop
2) $TTP \rightleftarrows A$: $f_{\mathbf{Ref}}, A, \omega$
3) $A \rightleftarrows TTP$: $f_{\mathbf{Ref}}, PS(\omega)$
 **if not** $check\,\omega\,()$ **then** stop
4) $TTP \rightleftarrows A$: $f_{\mathbf{cert}}, A, cert_A$

With the certificate, Bob can be convinced that TTP can convert the phaseal signatures once they are signed by the same $d$ as $PS(\omega)$.

### 3.2 Converting A Phaseal Signature

In this scheme, the phaseal signature is defined as

$$PS(m) = m^d (\bmod\ n^2)$$

and it is converted to be the final signature using

$$FS(m) = PS(m)^c (\bmod\ n)$$

It works because

$$FS(m)^e \equiv PS(m)^{ce} \equiv m^{dce} \equiv m(\bmod\ n)$$

holds.

### 3.3 Generating and Verifying Proofs

1) *Generating proofs:* Alice selects a random number $u < n^2$ and calculates

$$\begin{cases} \Omega = \omega^{u} (\bmod n^2) \\ M = m^{u} (\bmod n^2) \\ v = h(\Omega, M) \\ r = u + vd (\bmod \frac{\phi(n)n}{2}) \end{cases}$$

In this way, the proof of the $PS(m)$, denoted by $pf(PS(m))$, is $(r, \Omega, M)$.

2) *Verifying proofs:* When Bob gets the $PS(m)$ and $pf(PS(m))$, he calculates

$$v = h(\Omega, M)$$

and verifies

$$\begin{cases} \Omega PS(\omega)^{v} = \omega^{r} (\bmod n^2) \\ MPS(m)^{v} = m^{r} (\bmod n^2) \end{cases}$$

We denote the verifying operations as the function $verify(pf(PS(m)), m, PS(m), \omega, PS(\omega))$. If the verification fails, the function returns false.

## IV. A Fair Non-Repudiation Protocol

### *4.1 Registration Protocol*

Before the message transmission, Alice and Bob need to involve the registration protocol with TTP to get their own certificate $cert_A$ / $cert_B$.

Note that TTP can send the same reference message to Alice and Bob, which won't affect the security of the verification protocol.

### *4.2 Main Protocol*

The message to be sent is divided into two parts: the cipher and the key. The main protocol contains 4 moves. We de-note the content to be signed for **EOO** as $a$=( $f_{\mathbf{NRO}}$, $B$, $l$, $h(k)$, $cipher$, $E_{TTP}(k)$), then the **EOO** = $PS_A(a)$ plus $pf(PS_A(a))$ and **NRO** = $FS_A(a)$. Similarly, let $b$=( $f_{\mathbf{NRR}}$, $A$, $l$), then **EOR** = $PS_B(b)$ plus $pf(PS_A(b))$ and **NRR** = $FS_B(b)$.

Main Protocol

```
1) $A \rightarrow B$: $f_{\mathbf{EOO}}, B, l, h(k), cipher, E_{TTP}(k), PS_A(a), pf(PS_A(a))$, Sub
     if not $verify(pf(PS_A(a)), a, PS_A(a), \omega, PS_A(\omega))$ then stop
2) $B \rightarrow A$: $f_{\mathbf{EOR}}, A, l, PS_B(b), pf(PS_A(b))$
     if $A$ times out then abort
     if not $verify(pf(PS_B(b)), b, PS_B(b), \omega, PS_B(\omega))$ then abort
3) $A \rightarrow B$: $f_{\mathbf{NRO}}, B, l, k, FS_A(a)$
     if $B$ times out then recovery[$X$:=$B$,$Y$:=$A$]
4) $B \rightarrow A$: $f_{\mathbf{NRR}}, A, l, FS_B(b)$
     if $A$ times out then recovery[$X$:=$A$,$Y$:=$B$]
```

After the first move, Bob needs to verify the Alice's phaseal signature as **EOO**. Bob will quit the protocol when the verification fails. Note that the verification can be either interactive or non-interactive according the scheme adopted.

When Alice gets the **EOR**, she does the same thing. But instead of simply quitting the protocol, she needs to involve the abort protocol. This is important because it will prevent Bob's later recovery that may result in unfair situation for Alice.

### *4.3 Recovery Protocol*

The recovery protocol and the abort protocol are similar with those in AK protocol. Recovery protocol is executed when an error happens, one party needs TTP's help to decrypt the key $k$ and generate the final evidences for him/her.

Recovery Protocol

```
1) $X \rightleftarrows TTP$: $f_{\mathbf{RecX}}, f_{\mathbf{Sub}}, Y, l, h(cipher)$, h(k), $E_{TTP}(k)$, Rec$_{\mathbf{X}}$, Sub,
     $PS_A(a), PS_B(b)$
     if $h(k) \neq h(D_{TTP}(E_{TTP}(k)))$ or $aborted$ or $recovered$ then stop
     else $recovered$=true
     calculates
       $FS_A(a) = PS_A(a)^{c_A} (\bmod n)$ and $FS_B(b) = PS_B(b)^{c_B} (\bmod n)$
2) $TTP \rightleftarrows A$: $f_{\mathbf{NRR}}, A, l, FS_A(a)$
3) $TTP \rightleftarrows B$: $f_{\mathbf{NRO}}, B, l, k, FS_B(b)$
```

### *4.4 Abort Protocol*

Alice submits an abort request using abort protocol, preventing Bob may recover in a future time which she will not wait.

Abort Protocol

```
1) $X \rightleftarrows TTP$: $f_{\mathbf{Abort}}, l, B$, abort
     if $aborted$ or $recovered$ then stop
     else $aborted$=true
2) $TTP \rightleftarrows A$: $f_{\mathbf{Cona}}, A, B, l$, Con$_a$
3) $TTP \rightleftarrows B$: $f_{\mathbf{Cona}}, A, B, l$, Con$_a$
```

## V. Conclusion

In this paper, we propose a pseudo-resilient channel in the wireless environment. And based on that, we present a fair non-repudiation protocol with transparent TTP for wireless electronic transactions. The protocol uses an improved RSA-based convertible signature scheme. We achieve reduction in bandwidth and computation consumption by using a non-interactive verification protocol and the RSA signature as the non-repudiation evidences.

1) *The pseudo-resilient channel:* Although we realize higher channel reliability by using the cyclic resending method, we believe that there must be better way to achieve a resilient channel.

2) *Effectiveness, fairness, timeliness and non-repudiability:* Thanks to the contributive work of former fair protocols, our protocol satisfies these requirements quite well (readers are referred to [2][8] for detailed discussion).

3) *Security:* The scheme of Mao et al. is secure at a high probability (readers are referred to [11] for detailed discussion) and we can prove our verification protocol generates zero-knowledge proofs (proof is omitted).

4) *Publicity:* We realize the transparency of the TTP, so our protocol achieves good publicity.

5) *Abuse-freeness:* In some specific scenarios like contract signing and fair payment, abuse-freeness is quite important. Abuse-freeness means that before the protocol ends, no party can prove to a third party that he can choosing whether to complete or to abort the transaction. A non-repudiation protocol providing abuse-freeness in [12] can be adapted to the wireless network.

## REFERENCES

[1] N. Asokan, M. Schunter, and M. Waidner, "Optimistic protocols for fair exchange," Proceedings of the fourh ACM Conference on Computer and Communications Security, pages 6, 8-17, Zurich, Switzerland, Apr. 1997.

[2] N. Asokan and V. Shoup, "Optimistic fair exchange of digital signatures," Advances in Cryptology -- EUROCRYPT '98, number to be assigned in Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1998.

[3] F. Bao, R. H. Deng and W. Mao, "Efficient and practical fair exchange protocols with off-line TTP," Proceedings of 1998 IEEE Symposium on Security and Privacy, pages 77-85. Oakland, May 1998.

[4] M. Ben-Or, O. Goldreich, S. Micali and R. Rivest, "A fair protocol for signing contracts," IEEE Transactions on Information Theory, 36(1): 40-46, January 1990.

[5] C. Boyd and E. Foo, "Off-line Fair Payment Protocols using Convertible Signatures," Advances in Cryptology— ASIACRYPT'98, 1998.

[6] R. H. Deng, L. Gong, A. A. Lazar, and W. Wang, "Practical protocols for certified electronic mail," Journal of Network and Systems Management, 4(3), 1996.

[7] G. Forman and J. Zahorjan, "The challenges of mobile computing," IEEE Computer, 27(4), april 1994.

[8] S. Kremer and O. Markowitch, "Optimistic non-repudiable information exchange," 21th Symposium on Information Theory in the Benelux, pages 139-146, may 2000.

[9] O. Markowitch and S. Kremer, "An optimistic non-repudiation protocol with transparent trusted third party," Information Security: ISC 2001, volume 2200 of Lecture Notes in Computer Science, pages 363-378, Malaga, Spain. Springer-Verlag, October 2001.

[10] O. Markowitch and S. Saeednia, "Optimistic fair-exchange with transparent signature recovery," 5th International Conference, Financial Cryptography 2001, Lecture Notes in Computer Science. Springer-Verlag, 2001.

[11] W. Mao and K. Paterson, "Convertible Undeniable Standard RSA Signatures," unpublished.

[12] H. Wang and H. Guo, "Making non-repudiation practicable," unpublished.

[13] J. Zhou and D. Gollmann, "A fair non-repudiation protocol," Proceedings of 1996 IEEE Symposium on Research in Security and Privacy, pages 55-61, May 1996.