Contract Signatures in E-Commerce Applications

Lien Harn
Department of Computer Science
and Electrical Engineering
University of Missouri-Kansas City
Kansas City, MO 64110, USA
harnl@umkc.edu

Chu-Hsing Lin
Department of Computer Science
Tunghai University
Taichung 407, Taiwan
chlin@thu.edu.tw

Che-Wei Hu
Department of Computer Science
Tunghai University
Taichung 407, Taiwan
g98350032@thu.edu.tw

***Abstract*—In this paper, we propose a notion of contract signatures used in e-commerce applications. This scheme adopts digital multi-signature scheme in public-key cryptography to facilitate fair signature exchange over network. Security proof under the random oracle model of this modified signature scheme is include. This proposed solution allows two parties to produce and exchange two ambiguous signatures which are fully ambiguous for any third party (i.e. 1 out ∞ ambiguity). The combination of these two ambiguous signatures forms the contract signature. There is no "keystone" (i.e. a secret key used in the concurrent signature) of the signature. In case anyone releases the contract signature to a verifier, both signers bind to the contract signature.**

Keywords- Signature; E-commerce; Contract signing; Deniability; Security.

I. INTRODUCTION

As we are rapidly heading into an era of computer communications, cryptographic protocols are pressingly needed to facilitate secure communication. Cryptographic digital signatures have been used to provide non-repudiation services, such as signing business contract. One category of multi-party protocols in the future can be the exchange of digital signatures. One of the most important concerns in exchange signatures is the unfair exchange, which occurs when one party obtains the signature of another party without giving out his own.

The notion of fairness is well established in a traditional secret exchange, in which all parties involved obtain secrets simultaneously. However, in network environment, secrets are exchanged over a network that does not provide simultaneously exchange of messages, and, therefore, adequate cryptographic protocols are needed to facilitate secrets exchange and guarantee fairness to all parties involved.

Fair secret exchange protocols go as early as 1980s [1][2]. Earlier protocols are based on gradually exchange their secret bit-by-bit in an interleaving manner. This process continues until both of them have received and released all the bits; but only at the end of the protocol, both parties can discover that the received bits are real secrets and are not gibberish.

In this paper, we propose a fair protocol to let two parties to interact with each other over network to construct a contract signature (i.e. we will define this term in section 3). Our scheme adopts digital multi-signatures [3] in public-key cryptography to facilitate fair signature exchange. A contract signature is a special form of digital multi-signature that involves two signers, the seller and the buyer. A protocol allows two parties to produce and exchange two ambiguous signatures which are fully ambiguous (i.e. 1 out ∞ ambiguity). The combination of these two ambiguous signatures forms the contract signature. There is no "keystone" (i.e. a secret key used in the concurrent signature). In case anyone releases the contract signature to a verifier, both signers bind to the contract signature.

The paper is organized as follows. Next section gives an overview of related works. We define the contract signature and present a modified ElGamal signature scheme in section 3. We use this modified signature scheme to construct contract signature. Security proof under the random oracle model of this modified signature scheme is included. An interactive protocol to let two signers to construct a contract signature is introduced in section 4. Security discussion of the protocol is presented in section 5. We draw a conclusion in section 6.

II. RELATED WORKS

A number of protocols have been proposed up to date to achieve fair secrets exchange. Historically, the protocol design has been evolved from two-party approach, without the involvement of any trusted third party (TTP), to the TTP-based approach, where fairness is guaranteed with the involvement of a TTP.

The two-party protocols go as early as 1980s. Protocols [1][2] and, more recently, [4][5] are based on gradually exchange of small parts of the items to ensure that the exchange of the items occurs pseudo-simultaneously. This is achieved by having the parties release their secret items bit-by-bit in an interleaving manner- one party releases a bit of his item (together with the proof of correctness of the bit), and in return, receives a bit of his counterpart's item (and its proof of correctness). This process continues until both of them have received and released all the bits. Obviously, to achieve fairness, both secret items must be of the same length.

978-0-7695-4236-2/10 $26.00 © 2010 IEEE
DOI 10.1109/BWCCA.2010.101

A major disadvantage associated with two-party approach is that a large number of exchange rounds are needed to ensure an acceptable level of fairness, thus the overhead of communication is very high. In addition, gradual secret release protocols require that participating parties have approximately equal computational power in order to guarantee fairness. Otherwise, the party with larger computational capabilities can launch a brute-force attack after receiving the first several bits, and work out the remaining bits of his counterpart's secret. Although reasonably convincing in theory, this approach is too impractical for real life applications. Additionally, this kind of protocols provide no guarantees of the quality of secrets exchanged, i.e. parties can fairly exchange bits of their own, but only at the end of the protocol to discover that the received bits are gibberish.

In order to overcome these weaknesses, a TTP is introduced in the protocols to assist the participants with the exchange and to achieve fairness. In early TTP-based protocols, the TTP is on-line, i.e. it mediates every exchange process, even when participating parties are not attempting to cheat [6-9]. Basically, both parties send their items to the TTP, which verifies the correctness of the items and forwards them to the rightful recipients. The TTP is also responsible for generating and storing security sensitive transactional records, making it a focal point of security attacks. In on-line TTP-based approach, a protocol cannot be performed without the TTP, that makes it a potential communicational and performance bottleneck and susceptible to denial-of-service attacks. In addition, as all the exchanged data is exposed to the on-line TTP, it has to be fully and unconditionally trusted. Clearly, it is desirable to design protocols where the involvement of and security/storage requirements placed on the TTP are reduced. In off-line TTP-based protocols [10-19], the TTP is not involved in the protocol run under normal circumstances, i.e. when the participants do not attempt to cheat or are willing to resolve possible disputes themselves. Only when the exchange process fails to complete due to a network failure or a party's misbehavior, the TTP is invoked to assist the exchange finally come to a fair completion.

Recently, a new category of off-line TTP-based contract signing protocols, capable of making the role of the TTP transparent, have been proposed based on a cryptographic primitive called as Verifiable and Recoverable Encrypted Signature (VRES) [10-15][17][19]. The special primitive, e.g. VRES, makes the fair exchange protocols more complicate and restricts the employment of these protocols.

Misbehavior penalization can motivate all participants to behave honest so as to deter any party from misbehaving. Zhang, Shi, Nenadic, Merabti and Askwith [1] propose a fair signature exchange protocol with such property. However, in their protocol, TTP remains off-line during secrets exchange between two parties. TTP will be called upon whenever some party misbehaved. Since TTP cannot determine who the misbehaved party is and the misbehavior penalization can only be applied to one of the parties, the dishonest party can frame the honest one to be punished.

Chen et al. [20] proposed the concept of concurrent signature, CS for short. Such signature scheme allows two parties, without TTP, to produce and exchange two ambiguous signatures which are ambiguous for any third party until an extra piece of information (called keystone) is released by one of the parties. Concurrent signature is very efficient and requires neither a TTP nor a high degree of interaction between parties. Later in this section, we will address two problems associated with the concurrent signature. In order to strength the ambiguity of the signature before keystone is released, there are papers [21][22] proposed a strong notion, called perfect concurrent signatures. Later, asymmetric concurrent signature [23], tripartite concurrent signature [24], are proposed.

Problems associated with concurrent signature:

A. The keystone is in the hand of only one signature signer (i.e. the initiator). The owner can determine whether to release the keystone or not.

- **Scenario A-** Alice wants to sell an item over Internet and Bob is willing to pay it for certain price. Then, Alice and Bob agree to use the concurrent signature protocol to exchange their signatures. Alice is the initiator of the protocol and Alice selects a "keystone" and sends her concurrent signature to Bob. After verifying Alice's signature, Bob sends his concurrent signature to Alice. After verifying Bob's signature, Alice can show the keystone and Bob's signature to another potential buyer, Charley, privately. Charley can verify that Bob has made an offer to buy the item. Thus, Charley makes another higher offer to Alice. Alice can decide not to release the earlier signatures and keystone between Alice and Bob, and makes another deal with Charley.In your paper title, if the words "that uses" can accurately replace the word "using", capitalize the "u"; if not, keep using lower-cased.

B. Without releasing the keystone, the earlier exchanged signatures do not bind to any entity. However, the verifier can assure that any exchanged signature is signed by one out two signers. These signatures are not ambiguous enough. This type of ambiguity, 1 out 2 ambiguity, can only provide partially ambiguous.

- **Scenario B-** There is one potential problem in the ambiguity of concurrent signatures. When both parties deny generating an ambiguous signature (without revealing the keystone), a dispute may occur between two parties. Since both parties can generate the ambiguous signature, this dispute cannot be resolved cryptographically. However, the dispute is often resolved by general public, sometimes unfairly, based on personal records, credit history, personal social

status, etc. of involved parties. For example, when a dispute is occurred between two parties, general publics often make a judgment against the party with a criminal record. One way to improve the fairness of ambiguous signature is to increase the number of possible signers from two to ∞ (i.e. 1 out ∞ ambiguity).

In this paper, we propose a new type of digital signatures, called *contract signature*, to facilitate fair signature exchange over network. Since contract signature contains no keystone, problem presented in Scenario A can be avoided. Also, in our proposed protocol, the exchanged partial signatures are *fully ambiguous*. The problem presented in Scenario B can be avoided. Contract signature scheme is very efficient in terms of signing, verification and transmission.

III. CONTRACT SIGNATURE AND THE MODIFIED DIGITAL SIGNATURE SCHEME

In most business applications, two parties, the seller and the buyer, need to bind to the terms in a paper contract by signing the contract using hand-written signatures. In digital world, a public-key digital signature is used to represent a non-repudiation evidence of the electronic content. We call this type of digital signature the contract signature. Here, we give a formal definition of a digital contract signature.

Definition *Contract signature- a contract signature is a digital signature generated by two signers, A and B, jointly with their private keys, x_A and x_B. The contract signature can be verified by a verifier using signers' public keys, y_A and y_B.*

An efficient digital multi-signature is proposed in [3]. This multi-signature allows any number of signers to work together to generate a digital multi-signature corresponding to a message. The length of digital multi-signature is equivalent to the length of each individual signature and the public key of multi-signature is just the multiplication of all public keys of signers. A contract signature is a special form of digital multi-signature that only involves two signers.

In this section, we first introduce the modified ElGamal signature scheme used to construct contract signature. We present a formal security proof of this modified scheme. The original ElGamal signature scheme [25] was proposed in 1985; but the security was never proved equivalent to the discrete logarithm problem. In 1996, Pointcheval and Stern [26] used the Forking lemma to prove the security of a slight variant of the original ElGamal signature scheme.

1) The modified ElGamal signature scheme used to construct contract signatures consists of 3 steps as follows:

a) Let p be a large prime and g be a generator of Z_p, then the public key is $y = g^x \bmod p$ and the private key is x.

b) Picks $k \in Z_{p-1}$ randomly and a cryptographic hash function h*, the signature of message* m *is (r,s), where $r = g^k \bmod p$ and $s_A = x_A h(\mathrm{m},\mathrm{r}) - kr \bmod p - 1$.*

c) The verification of the signature checks the equation $y^{h(m_A r)} = r^r g^s \bmod p$.

We assume that hash function h behaves like a random oracle, and hence we follow the established cryptographic techniques, i.e., the Oracle Replay Attack and the Forking Lemma as proposed in [26], to prove the security of modified ElGamal signature scheme.

Theorem 1. *The modified ElGamal signature scheme is secure under the random oracle model against no-message attack and against adaptively chosen message attack.*

Proof :

For a formal security proof, the hash function *h=h(m, r)* in modified signature scheme will be treated as a random oracle. Assume to the contrary that without knowing the trapdoor secret (*x, k*), given input (*p, g, y*), the adversary can generate an output (m_1, h_1, r_1, s_1) such that $y^{h(m_1,r_1)} = r_1^{r_1} g^{s_1} \bmod p$, where $h_1 = h(m_1, r_1)$, by making some oracle queries in probabilistic polynomial time. Then, based on the well-known cryptographic techniques, the Oracle Replay Attack and the Forking Lemma as proposed in [7], the adversary uses the oracle replay attack by a polynomial replay of the attack with the same random tape and a different oracle. Readers can refer to the original works in [7] for the detailed description. The adversary obtains two outputs of a special form as (m_1, h_1, r_1, s_1) and (m_2, h_2, r_2, s_2), where $(m_2, r_2) = (m_1, r_1)$ and $h_1 \neq h_2$, $s_1 \neq s_2$. Since (h_1, r_1, s_1) and (h_2, r_2, s_2) are two pairs of signatures for *h* where $h_1 \neq h_2$, we obtain the following two equations: $y^{h(m_1,r_1)} = r_1^{r_1} g^{s_1} \bmod p$, and $y^{h(m_2,r_2)} = r_2^{r_2} g^{s_2} \bmod p$. Thus, we have $y^{h(m_1,r_1)-h(m_2,s_2)} = g^{s_1-s_2} \bmod p$. It is easy to compute the discrete logarithm of *y* as $x = (h(m_1,r_1) - h(m_2,r_2))^{-1}(s_1 - s_2) \bmod p - 1$. This result contradicts the discrete logarithm assumption.

In our proposed contract signature, a contract signature (r,s) of a contract *m* can be verified by a verifier by checking whether $y^{h(m_A r)} = r^r g^s \bmod p$, where $y = y_A \cdot y_B = g^{x_A + x_B} \bmod p$. To digitally generate a valid contract signature (r,s), according to Theorem 1, knowing the private key $x = x_A + x_B$ of the public key is needed. In the next section, we propose an exchange protocol to allow A and B to interact with each other to construct the contract signature.

IV. PROTOCOL FOR CONSTRUCTING A CONTRACT SIGNATURE

System set-up:

Let A and B agree to sign a contract *m* over Internet. We assume that A's private key is x_A, public key

is $y_A = g^{x_A} \bmod p$, and B's private key is x_B, public key is $y_B = g^{x_B} \bmod p$.

Exchange protocol:

a) : A randomly selects a secret k_A, and computes a public value $r_A = g^{k_A} \bmod p$. r_A is sent to B.

b) Similarly, B randomly selects a secret k_B, and computes a public value $r_B = g^{k_B} \bmod p$. r_B is sent to A.

c) A computes $r = r_A \cdot r_B \bmod p$. A solves s_A such that $s_A = x_A h(\mathrm{m}, r) - k_A r \bmod p - 1$. s_A is sent to B.

d) B verifies $y_A^{h(m,r)} = r_A^{r} \cdot g^{s_A} \bmod p$. If it is, then B computes $r = r_A \cdot r_B \bmod p$, and solves s_B such that $s_B = x_B h(\mathrm{m}, r) - k_B r \bmod p - 1$. s_B is sent to A.

e) A verifies $y_B^{h(m,r)} = r_B^{r} \cdot g^{s_B} \bmod p$. Both A and B can compute the contract signature, (r, s), where $s = s_A + s_B \bmod p - 1$, of the contract m. The contract signature (r, s) of the contract m can be verified by a verifier by checking $(y_A \cdot y_B)^{h(m,r)} = r^r \cdot g^s \bmod p$.

Theorem 2. *The combination of two partial signatures, (r_A, s_A) and (r_B, s_B), in above protocol is a valid signature.*

Proof:

The two partial signatures, (r_A, s_A) and (r_B, s_B), satisfy

$$y_A^{h(m,r)} = r_A^{r} \cdot g^{s_A} \bmod p, \text{ and}$$

$$y_B^{h(m,r)} = r_B^{r} \cdot g^{s_B} \bmod p.$$

Multiplying above two equations, we obtain

$$(y_A \cdot y_B)^{h(m,r)} = (r_A \cdot r_B)^{r} \cdot g^{(s_A + s_B)} \bmod p$$
$$= r^r \cdot g^s \bmod p,$$

where $r = r_A \cdot r_B \bmod p$ and $s = s_A + s_B \bmod p - 1$. Thus, the combined signature (*r, s*) is a valid contract signature of the contract *m.*

V. Security Discussion

In this proposed scheme, the partial signatures, (rA, sA) and (rB, sB) have no binding to any signer. Verifier cannot tell who is the signer because anyone, without knowing the private key xA, can first randomly select sA and solve for rA to satisfy the equation $y_A^{h(m,r)} = r_A^{r} \cdot g^{s_A} \bmod p$. This type of ambiguity, 1 out ∞ ambiguity, can provide fully ambiguous. On the other hand, the signature (r, s) binds to both signers since to generate a valid pair of (r, s) to satisfy $(y_A \cdot y_B)^{h(m,r)} = r^r \cdot g^s \bmod p$, according to Theorem 1 in previous section, needs to know the private key, $x = x_A + x_B$, of the public key.

In network environment that does not provide simultaneously exchange of messages, one party must be able to obtain some final-round information earlier than the other party. In order to improve fairness, content of final-round information becomes very important. If the content of final-round information is a secret keystone, it will benefit the receiver of the final-round information. We have addressed the potential problem in ***Scenario A*** in the introduction section. In our proposed solution, the content of final-round information can turn earlier exchanged information into a digital contract signature. Since the contract signature binds to both Alice and Bob, this feature can morally and legally prevent Alice and Charley to make another deal.

VI. Conclusion

We propose a notion of contract signatures to facilitate fair signing contract over Internet. Contract signature is based on a modified ElGamal signature scheme. Security proof of this modified signature scheme under the random oracle model is include. The protocol to construct a contract signature allows two parties to exchange two ambiguous signatures which are fully ambiguous. The combination of these two ambiguous signatures is the contract signature.

VII. Acknowledgement

This work is supported in part by National Science Council under grants NSC99-2221-E029-034-MY3.

References

[1] Blum M., How to exchange (secret) keys, ACM Transactions on Computer Systems Vol. 1, No. 2, pp.175-193, 1983.

[2] Even S., Goldreich O., Lempel A., A randomized protocol for signing contracts, Communications of the ACM 28 (6), pp. 637-647, 1985.

[3] Harn L., Group-oriented (t, n) threshold signature and multisignature, IEE Proceedings-Computers and Digital Techniques, Vol. 141, No. 5, pp. 307-313, Sep. 1994.

[4] Damgard LB., Practical and provably secure release of a secret and exchange of signatures, Advances in Cryptology - EUROCRYE'T '93, LNCS, Springer-Verlag, Berlin, Germany, vol. 765, pp. 200-217, 1994.

[5] Okamoto T., Ohta K., How to simultaneously exchange secrets by general assumptions, Proceedings of ACM Conference on Computer and Communication Security, pp. 184- 192, 1994.

[6] Franklin M. K., Reiter M., Fair exchange with a semi-trusted third party, Proceedings of ACM Conference on Computer and Communications Security, pp. 1-5, 1997.

[7] Bürk H., Pfitzmann A., Value exchange systems enabling security and unobservability, Computers & Security 9, pp. 715- 721, 1990.

[8] Ketchpel S., Transaction protection for information buyers and sellers, Proceedings of the Dartmouth Institute for Advanced Graduate Studies '95: Electronic Publishing and the Information Superhighway, Boston, USA, 1995.

[9] Zhou J., Gollmann D., Observations on non-repudiation, Proceedings of Advances in Cryptology - ASIACRYPT '96, LNCS, Springer, vol. 1163, pp. 133-144, 1996.

[10] Ateniese G., Efficient verifiable encryption (and fair exchange) of digital signatures, Proceedings of ACM Conference on Computer and Communications Security, pp. 138-146, 1999.

[11] Asokan N., Schunter M., Waidner M., Optimistic fair exchange of digital signatures, IEEE Journal on Selected Areas in Communications 18, pp. 593-610, 2000.

[12] Asokan N., Schunter M., Waidner M., Opthistic Fair exchange of digital signatures, Extended Abstract, Proceedings of Advances in

Cryptology - EUROCRYPT '98, LNCS, Springer-Verlag, Berlin, Gennany, vol. 1403, pp. 591-606,1998.
[13] Gamy J. A., Jakobsson M., MacKenzie P., Abuse-fiee optimistic contract signing, Proceedings of Advances in Cryptology - CRYPTO '99, LNCS, Springer-Verlag, Berlin, Germany, vol. 1666, pp. 449-466, 1999.
[14] Asokan N., Schunter M., Waidner M., Asynchronous protocols for optimistic fair exchange, Proceedings of the IEEE Symposium on Security and Privacy, pp. 86-100, 1998.
[15] Bao F., Deng R., Mao W., Efficient and practical fair exchange protocols with off-line TTP, Proceedings of IEEE Symposium on Security and Privacy, pp. 77-85, 1998.
[16] Ray I., Ray I., An optimistic fair exchange e-commerce protocol with automated dispute resolution, Proceedings of 1st International Conference on Electronic Commerce and Web Technologies EC-Web 2000, LNCS, Springer-Verlag, Berlin, Vol.1875, pp. 84-93, 2000.
[17] Wu C., Varadharajan V., Fair exchange of digital signatures with offline trusted thud party, International Conference on Information and Communication Security, pp. 466-470, 2001.
[18] Zhang N., Shi Q., An efficient protocol for anonymous and fair document exchange, Computer Networks Journal 41, Elsevier Science Publisher, pp. 19-28, 2003.
[19] Chen L., Efficient fair exchange with' verifiable confirmation of signatures, Proceedings ASIACRYPT '98, LNCS, Springer-Verlag, Berlin, Germany, vol. 1514, pp. 286-299, 1998.
[20] Chen L., Kudla C., Paterson K. G., Concurrent signatures, Proceedings of Advances Cryptology -EUROCRYPT '04, LNCS, Springer-Verlag, Berlin, Germany, vol. 3027, pp. 287-305, 2004.
[21] Susilo W., Mu Y., Zhang F', Perfect concurrent signature schemes, Proceedings of the ICICS '04, Springer-Verlag, Berlin, Germany, vol. 3269, pp. 14-26. 2004.
[22] Wang G.- l., Bao F., Zhou J. – y., The fairness of perfect concurrent signatures, Proceedings of the ICICS '06, Springer-Verlag, Berlin, Germany, vol. 4307, pp. 435-451, 2006.
[23] Nguyen K., Asymmetric concurrent signatures, Proceedings of the ICICS '05, Springer-Verlag, Berlin, Germany, vol. 3783, 2005.
[24] Susilo W., Mu Y., Tripartite concurrent signatures, Proceedings of the IFIP/SEC '05, pp.425-441, 2005.
[25] ElGamal, T., A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. IT, vol. 31, pp. 469-472, 1985.
[26] Pointcheval, D., Stern, J., Security proofs for signature schemes, Proceedings of Advances in Cryptology - Eurocrypt '96, Springer-Verlag, Berlin, Germany, vol. 1070, pp. 387-398, 1996.