Computers and Electrical Engineering 37 (2011) 169–173

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Contract signature in e-commerce [☆]

Lein Harn [a,*], Chu-Hsing Lin [b]

[a] Department of Computer Science Electrical Engineering, University of Missouri-Kansas City, United States
[b] Department of Computer Science, Tunghai University, Taiwan

ARTICLE INFO

Article history:
Available online 19 February 2011

ABSTRACT

In this paper, we propose a notion of contract signature used in e-commerce applications. We propose a contract signature scheme based on the discrete logarithm assumption. The contract signature scheme adopts a digital multi-signature scheme in public-key cryptography to facilitate fair signature exchange over network. This proposed solution allows multiple signers of a contract signature to exchange their partial signatures which are *fully ambiguous* for any third party (i.e., *1 out of* ∞ *ambiguity*) to construct a valid contract signature. In case any signer releases the partial signature to others, the signer does not bind to the contract.

1. 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 multiple 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 the digital multi-signature [3] in public-key cryptography to facilitate fair signature exchange. A contract signature is a special form of digital multi-signature that involves multiple signers. A protocol allows multiple parties to produce and exchange their ambiguous signatures which are *fully ambiguous* (i.e., *1 out of* ∞ *ambiguity*). The combination of these 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, all signers bind to the contract signature.

☆ Reviews processed and proposed for publication by Associate Editor Pro. N. Sklavos.

* Corresponding author. Address: University of Missouri-Kansas City, Computer Science Electrical Engineering, 5100 Rockhill Rd., Kansas City, MO 64110, United States. Tel.: +1 816 235 2358; fax: +1 816 235 5159.

E-mail address: harnl@umkc.edu (L. Harn).

0045-7906/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2011.02.001

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 multiple 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.

2. 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.

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 provides 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. Wang [20] has proposed an abuse-free fair contract-signing protocol based on the RSA signature very recently. In Wang's scheme, the TTP is involved only in the situations where one party is cheating or the communication channel is interrupted.

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 et al. [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. [21] 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 [22,23] proposed a strong notion, called perfect concurrent signatures. Later, asymmetric concurrent signature [24], tripartite concurrent signature [25], are proposed.

2.1. Problems associated with concurrent signature

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.

2.1.1. 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.

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 of 2 ambiguity*, can only provide *partially ambiguous*.

2.1.2. 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 of* ∞ *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.

3. 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 3.1. A contract signature is a digital signature generated by multiple signers, jointly with their private keys. The contract signature can be verified by a verifier using signers' public keys.

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 [26] was proposed in 1985; but the security was never proved equivalent to the discrete logarithm problem. In 1996, Pointcheval and Stern [27] used the Forking lemma to prove the security of a slight variant of the original ElGamal signature scheme.

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

- 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.
- 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 $s_A = x_A h(m, r) - kr \bmod p - 1$.
- 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 [27], 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 $y^{h(m_{1_A} r_1) - h(m_2,s_2)} = g^{s_1 - s_2} \bmod p$. 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.

4. Protocol for constructing a contract signature

4.1. System set-up

For the rest of paper, we assume that a contract signature is signed by two signers. When there are more than two signers, same procedures can be followed to generate the contract signature. 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 $y_B = g^{x_B} \bmod p$.

4.2. Exchange protocol

Step 1: 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.
Step 2: 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.
Step 3: A computes $r = r_A \cdot r_B \bmod p$. A solves s_A such that $s_A = x_A h(m,r) - k_A r \bmod p-1$. s_A is sent to B.
Step 4: 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 $s_B = x_B h(m,r) - k_B r \bmod p-1$. s_B is sent to A.
Step 5: 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

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

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. □

5. Security discussion

In this proposed scheme, the partial signatures (r_A, s_A) and (r_B, s_B) have no binding to any signer. Verifier cannot tell who is the signer because anyone, without knowing the private key x_A, can first randomly select s_A and solve for r_A to satisfy the equation $y_A^{h(m,r)} = r_A^r \cdot g^{s_A} \bmod p$. This type of ambiguity, 1 out of ∞ 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.

6. Conclusion

We propose the formal definition of a contract signature and a modified ElGamal signature scheme to implement the contract signature. Security proof of the modified signature scheme under the random oracle model is included. The protocol allows multiple signers to exchange their ambiguous signatures interactively. The combination of these ambiguous signatures is the contract signature. Since our proposed contract signature has no keystone, security problems can be avoided. In addition, the length of a contract signature is identical to the length of a single digital signature and the verification time of a contract signature is fixed.

References

[1] Blum M. How to exchange (secret) keys. ACM Trans Comput Syst 1983;1(2):175–93.
[2] Even S, Goldreich O, Lempel A. A randomized protocol for signing contracts. Commun ACM 1985;28(6):637–47.
[3] Harn L. Group-oriented (t, n) threshold signature and multisignature. IEE Proc-Comput Digital Tech 1994;141(5):307–13.
[4] Damgard LB. Practical and provably secure release of a secret and exchange of signatures. In: Proceedings of advances in cryptology – EUROCRYE'T '93, vol. 765. Berlin, Germany: LNCS, Springer-Verlag; 1994. p. 200–17.
[5] Okamoto T, Ohta K. How to simultaneously exchange secrets by general assumptions. In: Proceedings of ACM conference on computer and communication security, 1994. p. 184–92.
[6] Franklin MK, Reiter M. Fair exchange with a semi-trusted third party. In: Proceedings of ACM conference on computer and communications security, 1997. p. 1–5.
[7] Bürk H, Pfitzmann A. Value exchange systems enabling security and unobservability. Comput Security 1990;9:715–21.
[8] Ketchpel S. Transaction protection for information buyers and sellers. In: Proceedings of the Dartmouth institute for advanced graduate studies '95. Boston, USA: Electronic Publishing and the Information Superhighway; 1995.
[9] Zhou J, Gollmann D. Observations on non-repudiation. In: Proceedings of advances in cryptology – ASIACRYPT '96, vol. 1163. LNCS, Springer; 1996. p. 133–44.
[10] Ateniese G. Efficient verifiable encryption (and fair exchange) of digital signatures. In: Proceedings of ACM conference on computer and communications security, 1999. p. 138–46.
[11] Asokan N, Schunter M, Waidner M. Optimistic fair exchange of digital signatures. IEEE J Selected Areas Commun 2000;18:593–610.
[12] Asokan N, Schunter M, Waidner M. Opthistic fair exchange of digital signatures, extended abstract. In: Proceedings of advances in cryptology – EUROCRYPT '98, vol. 1403. Berlin, Germany: LNCS, Springer-Verlag; 1998. p. 591–606.
[13] Gamy JA, Jakobsson M, MacKenzie P. Abuse-free optimistic contract signing. In: Proceedings of advances in cryptology – CRYPTO '99, vol. 1666. Berlin, Germany: LNCS, Springer-Verlag; 1999. p. 449–66.
[14] Asokan N, Schunter M, Waidner M. Asynchronous protocols for optimistic fair exchange. In: Proceedings of the IEEE symposium on security and privacy, 1998. p. 86–100.
[15] Bao F, Deng R, Mao W. Efficient and practical fair exchange protocols with off-line TTP. In: Proceedings of IEEE symposium on security and privacy, 1998. p. 77–85.
[16] Ray I, Ray I. An optimistic fair exchange e-commerce protocol with automated dispute resolution. In: Proceedings of first international conference on electronic commerce and web technologies EC-Web 2000, vol. 1875. Berlin: LNCS, Springer-Verlag; 2000. p. 84–93.
[17] Wu C, Varadharajan V. Fair exchange of digital signatures with offline trusted thud party. In: Proceedings of the international conference on information and communication security, 2001. p. 466–70.
[18] Zhang N, Shi Q. An efficient protocol for anonymous and fair document exchange. Computer Networks Journal 41, Elsevier Science Publisher, 2003. p. 19–28.
[19] Chen L. Efficient fair exchange with' verifiable confirmation of signatures. In: Proceedings ASIACRYPT '98, vol. 1514. Berlin, Germany: LNCS, Springer-Verlag; 1998. p. 286–99.
[20] Wang G. An abuse-free fair contract-signing protocol based on the RSA signature. IEEE Trans Inform Forensics Security 2010;5(1):158–68.
[21] Chen L, Kudla C, Paterson KG. Concurrent signatures. In: Proceedings of advances cryptology – EUROCRYPT '04, vol. 3027. Berlin, Germany: LNCS, Springer-Verlag; 2004. p. 287–305.
[22] Susilo W, Mu Y, Zhang F. Perfect concurrent signature schemes. In: Proceedings of the ICICS '04, vol. 3269. Berlin, Germany: Springer-Verlag; 2004. p. 14–26.
[23] Wang G-l, Bao F, Zhou J-y. The fairness of perfect concurrent signatures. In: Proceedings of the ICICS '06, vol. 4307. Berlin, Germany: Springer-Verlag; 2006. p. 435–51.
[24] Nguyen K. Asymmetric concurrent signatures. In: Proceedings of the ICICS '05, vol. 3783. Berlin, Germany: Springer-Verlag; 2005.
[25] Susilo W, Mu Y. Tripartite concurrent signatures. In: Proceedings of the IFIP/SEC '05, 2005. p. 425–41.
[26] ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans IT 1985;31:469–72.
[27] Pointcheval D, Stern J. Security proofs for signature schemes. In: Proceedings of advances in cryptology – Eurocrypt '96, vol. 1070. Berlin, Germany: Springer-Verlag; 1996. p. 387–98.

Lein Harn received Ph.D. degree in Electrical Engineering from the University of Minnesota in 1984. In 1984, he joined the Department of Electrical and Computer Engineering, University of Missouri-Columbia as an Assistant Professor, and in 1986, he moved to the Department of Computer Science Electrical Engineering, University of Missouri-Kansas City (UMKC). In 1996, he took one-year development leave to work as a research staff at Racal Datacom Group, Florida. His research interests include cryptography, network security, and wireless communication security.

Chu-Hsing Lin joined the faculty of Tunghai University, Taiwan, in 1989. Since then, he has served as the Director of Computer Center and the Department Chair, Department of Computer Science, Tunghai University. He has published motr than 100 papers in academic journals and international conferences. His current research interests include multimedia information security, wireless ad hoc networks, and embedded systems applications.