World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 177-180, 2014
ISSN 1818-4952
© IDOSI Publications, 2014
DOI: 10.5829/idosi.wasj.2014.29.csea.2254

A Novel Ruin Gratis Fair Digital Contract Signing Protocol Based on Rsa Signature

K.P. Thooyamani, R. Udayakumar and V. Khanaa

School of Computing Science, Bharath University, Chennai-73, India

Abstract: A fair contract-signing protocol allows two potentially mistrusted parities to exchange their commitments to an agreed contract over the Internet in a fair way, so that either each of them obtains the other's signature or neither party does. Based on the RSA signature scheme, a new digital contract-signing protocol is proposed in this paper. Like the existing RSA-based solutions for the same problem, our protocol is not only fair, but also optimistic, since the trusted third party is involved only in the situations where one party is cheating or the communication channel is interrupted. Furthermore, the proposed protocol satisfies new property abuse freeness. That is, if the protocol is executed unsuccessfully, none of the two parties can show the validity of intermediate results to others. This abuse-free fair contract signing protocol based on the RSA *signature provides both security and efficiency*.

Key words: Contract signing · Cryptographic protocols · Digital signatures · e-commerce · Fair-exchange · RSA · Security

INTRODUCTION

Contract signing plays a very important role in any business transaction, in particular in situations where the involved parties do not trust each other to some extent already. In the paper-based scenario, contract signing is truly simple due to the existence of simultaneity. That is, both parties generally sign two hard copies of the same contract at the same place and at the same time. After that, each party keeps one copy as a legal document that shows both of them have committed to the contract. If one party does not abide by the contract, the other party could provide the signed contract to a judge in court. As electronic commerce is becoming more and more important and popular in the world, it is desirable to have a mechanism that allows two parties to sign a digital contract via the Internet. However, the problem of contract signing becomes difficult in this setting, since there is no simultaneity any more in the scenario of computer networks. In other words, the simultaneity has to be mimicked in order to design a digital contract-signing protocol. This requirement is essentially captured by the concept of fairness [1]. At the end of the protocol, either both parties have valid signatures for a contract or neither does, even if one of them tries to cheat or the communication channel is out of order. In fact it is proved that impossible to achieve fairness in a deterministic two party contract-signing protocol. The intuitive reason could be explained as follows. The purpose of such a protocol is to go from the initial fair state, in which no party has what he/she expects, to the desired fair state in which both obtain what they want. However, information is exchanged in computer networks non-simultaneously, so at least an unfair state must be passed through [2].

Existing System: Based on the involvement of the trusted third party (TTP), the contract signing protocol is divided into three types. 1) Gradual exchange without any TTP; 2) Exchange with off-line TTP; 3) Exchange with on-line TTP. Early efforts [3] mainly focused on first type of protocols to meet computational fairness. Exchanges take place without TTP. But the drawback is, the party having more computing power will force the other to get the information. In the second type [1], TTP is not invoked unless one of the two parties misbehaves. But in this type dictionary errors are there. In the third type [2] TTP will be in online always. Even though it is expensive it has more advantages compared to other two types. There is no need for maintaining a database to remember the state information.

Corresponding Author: R. Udayakumar, School of Computing Science, Bharath University, Chennai-73, India.

Park *et al*'s scheme is the existing scheme [8] for exchanging the signatures. But in this scheme TTP can find the private key of the parties. In cryptosystem, private key should not be revealed to any party including a partially trusted party. This becomes the main drawback.

Work Done: This paper proposes a new contract-signing protocol for two mutually distrusted parties. This protocol is based on an RSA multi-signature, which is formally proved to be secure. This protocol is fair and optimistic. Furthermore, different from the existing schemes, this protocol is abuse-free. The reason is that an interactive zero-knowledge protocol is integrated and proposed for confirming RSA undeniable signatures, in order to prove the validity of the intermediate results. Moreover, trapdoor commitment schemes are exploited to enhance this zero-knowledge protocol so that the abuse-freeness property can be fully achieved [7]. More specifically, the new protocol satisfies the following desirable properties.

Fairness: This protocol guarantees the two parties involved to obtain or not obtain the other's signature simultaneously. This property implies that even a dishonest party who tries to cheat cannot get an advantage over the other party.

Optimism: The TTP is involved only in the situation where one party is cheating or the communication channel is interrupted. So it could be expected that the TTP is only involved in settling disputes between users rarely, due to the fact that fairness is always satisfied, that is, cheating is not beneficial to the cheater.

Abuse-Freeness: If the whole protocol is not finished successfully, any of the two parties cannot show the validity of the intermediate results generated by the other to an outsider, either during or after the procedure where those intermediate results are produced. As mentioned before, the unique known abuse-free contract signing protocol is based on the discrete logarithm problem, instead of the RSA cryptosystem.

Provable Security: Under the standard assumption that the RSA problem is intractable, the protocol is provably secure in the random hash function model [4-6].

Timely Termination: The execution of a protocol instance will be terminated in a predetermined time. This property is implemented by adding a reasonable deadline in a contract. If one party does not send his/her signature to the other party after the deadline, both of them are free of liability to their partial commitments to the contract and do not need to wait any more.

Compatibility: In this protocol, each party's commitment to a contract is a standard digital signature. This means that to use the protocol in existing systems, there is no need to modify the signature scheme or message format at all. Thus, it will be very convenient to integrate the contract- signing protocol into existing software for electronic transactions [9].

TTP's Statelessness: To settle potential disputes between users, the TTP is not required to maintain a database to searching or remembering the state information for each protocol instance, so the overhead on the side of the TTP is reduced greatly, compared with the previous schemes.

Trapdoor Commitment Schemes: As using standard zero-knowledge is not enough to guarantee the abuse-freeness in our protocol, we need another cryptographic primitive, called trapdoor commitment schemes. As a two-phase protocol running between a sender and a receiver, a commitment scheme allows the sender to first hide a value by computing a commitment and then reveals the hidden value together with some related information to open the commitment so that the receiver can check whether the commitment is decommitted correctly. Informally, a secure commitment scheme should satisfy the hiding property and the binding property. The former means that given a commitment, the receiver is unable to know which value is committed, while the latter requires that once a commitment have been made, the sender cannot change his mind to cheat the receiver by revealing a different value, which is not the value committed initially. In a trapdoor commitment (TC) scheme, there is one trapdoor that would allow the owner of this trapdoor to open a commitment in different ways. Due to this amazing additional property, a valid answer to a commitment can only be accepted by the owner of the trapdoor, usually the commitment receiver [10, 11]. The reason is that once getting such a valid answer, an outsider cannot distinguish whether this answer is revealed by the sender or forged by the receiver using the trapdoor. Actually, this is why trapdoor commitment schemes can help us to achieve the abuse-freeness property in the contract-signing scenario. A trapdoor commitment scheme consists of four algorithms. They are TCgen, TCcom, TCver, Tcsim.

Tcgen is the key generation algorithm. In this receiver runs TCgen to get a commitment public key Pk and the corresponding trapdoor td. TCcom is the commitment algorithm. This algorithm outputs a pair(com,dec) where com is commitment to the value r and dec is the related information used to decommit com. TCver is the commitment verification algorithm. This algorithm is used to check whether an answer(r,dec) is valid to a given commitment com with respect to public key pk. TCsim is the simulation algorithm. This allows the receiver to simulate a new answer (r',dec') for a commitment com using trapdoor td, when one answer (r,dec) for com is given.

The Proposed Scheme: It is assumed that the communication channel between User A and User B is unreliable, that is messages inserted into such a channel may be lost due to the failure of computer network or attacks from adversaries. However, the TTP is linked with User A and User B by reliable communication channels, that is, messages inserted into such a channel will be delivered to the recipient after a finite delay. Proposed work consists of three protocols. 1)Registration Protocol, 2)Signature Exchange Protocol and 3)Dispute Resolution Protocol.

Table 1: Notations Used

Notations	Definitions
d	Private Key
e	Public Key
C_A	Certificate of User A
V_A	Voucher of User A
σ_1 σ_2	Partial Signature
σ_A, σ_B	Signature of User A, User B respectively
(w, σ_w)	Sample Message Signature Pair

Registration Protocol: To use our protocol for exchanging digital signatures, only the initiator Alice needs to register with the TTP. That is, Alice is required to get a long-term voucher V_A from the TTP besides obtaining a certificate C_A from a CA. To this end, the following procedures are executed.

Step 1: User A sets p,q where pand q are two k-bit safe primes. Find p',q' such that p = 2p'+1 and q = 2q'+1. Calculate n = pq and $\varphi(n) = (p-1)(q-1)$. User A selects public key e such that e $\in_R Z^* \varphi(n)$. User A calculates private key d where $d = e^{-1} \bmod \varphi(n)$. User A registers the public key with a CA to get the certificate C_A. C_A binds (n,e) together.

Step 2: Then User A calculates e_1 such that $\gcd(e,\varphi(n)) = 1$. User A splits the private key d as d_1,d_2 where $d_1 = e^{-1} \bmod \varphi(n)$, $d_2 = d-d_1$. User A generates a sample message-signature pair (w,σ_w) where w $\in$ Z*n. Calculate $\sigma_w = w^{d1} \bmod n$. User A sends ($C_A$,w,$\sigma_w$,$d_2$) to the TTP but keeps (d,d_1,d_2,e_1) secret.

Step 3: TTP first checks that User A's certificate C_A is valid. TTP checks (w,σ_w,d_2). TTP validates that w is an element of order at least p'q' by checking that w $\in$ Z*n and that both gcd(w-1,n) and gcd(w+1,n) are not prime factors of n. TTP checks whether $w = (\sigma_w w^{d2})^e \bmod n$. If everything is in order, TTP stores d_2 securely and creates a voucher V_A by computing $V_A = \text{sign}_{TTP}(C_A,w,\sigma_w)$.

Signature Exchange Protocol: In the proposed protocol it is assumed that a contract m has been agreed between User A and User B before they begin to sign it. In addition, it is supposed that the contract explicitly contains the following information. A predetermined but reasonble deadline t and the identities of User A, User B and the TTP. Following are the steps involved in this protocol.

Step 1: User A computes partial signature $\sigma_1 = h(m)^{d1} \bmod$ n.User A sends (C_A,V_A,σ_1) to User B.

Step 2: User B verifies C_A,V_A and checks the identities of User A, User B and TTP.

Step 3: If all those validations hold, User B initiates the following interactive zero-knowledge protocol with User A to check σ_1. a) User B picks 2 numbers i,j $\in_R$ [1,n] at random. b) Sends c to User A where $c = (\sigma_1^{2i})\times(\sigma_w^{j}) \bmod n$ c) User A calculates the respondence $r = c^{e1} \bmod n$ and returns the commitment $r^{-} = TCcom(r,t)$ to User B by selecting t. d) When User B receives r^{-}, B sends A the pair (i,j). User A checks c that is, $c \equiv (\sigma_1^{2i})\times(\sigma_w^{j}) \bmod n$ e) If the answer is positive, User A decommits the commitment r^{-}.

Step 4: User B sends signature σ_B on contract m to User A.

Step 5: User A checks User B's signature σ_B is valid.

- If true User A sends the partial signature σ_2 to User B. b) User B calculates $\sigma_A^{-} = \sigma_1\sigma_2 \bmod n$. c) User B accepts σ_2 as valid if and only if $h(m)^2 = (\sigma_A^{-})^{2e} \bmod n$. d) If σ_2 is invalid User B applies help from TTP.

Dispute Resolution Protocol: If User B has sent his signature σ_B to User A but does not receive the value of σ_2 or only receives an invalid σ_2 from User A before the deadline t, then User B claims TTP to apply dispute resolution. Following are the steps involved in this protocol.

Step 1: If dispute occurs, User B sends $(C_A,V_A,m,\sigma_1,\sigma_B)$ to TTP. TTP checks C_A,V_A,σ_B and deadline t.

Step 2: TTP computes $\sigma_2 = h(m)^{d2} \bmod n$ and checks $h(m)^2 = (\sigma_1\sigma_2)^{2e} \bmod n$.

Step 3: If this equality holds TTP sends (m,σ_2) to User B and forwards (m,σ_A) to user A.

Step 4: If inequality occurs TTP sends an error message to User B.

CONCLUSION

Based on the standard RSA signature scheme, a new digital contract signing protocol is proposed which allows two potentially mistrusted parties to exchange their digital signatures on a contract in an efficient and secure way. The new protocol is fair and optimistic that is two parties get or do not get the other's digital signature simultaneously and the TTP is only needed in abnormal cases that occur occasionally. The proposed protocol is abuse free. This is an important security property for contract signing, especially in the situations where partial commitments to a contract may be beneficial to a dishonest party or an outsider.

REFERENCES

1. Ateniese, G., 1999. Efficient verifiable encryption of digital signature, in 1999.
2. Ben-Or, M., O. Goldreich, S. Micali and R.L. Rivest, 1990. A fair protocol for signing contracts, pp: 36.
3. Even, S., O. Goldreich and A. Lempel, 1985. A randomized protocol for signing contracts, pp: 28.
4. Fischlin, M., 2001. Trapdoor Commitment Schemes and Their Applications.
5. Garay, J., M. Jakobsson and P. MacKenzie, 1999. Abuse-free optimistic contract signing, pp: 1666.
6. Gurgens, S., C. Rudolph and H. Vogt, 2003. On the security of fair nonrepudiation protocols, pp: 2851.
7. Kremer, S., O. Markowitch and J. Zhou, 2002. An intensive survey of fair non-repudiation protocols, pp: 25.
8. Park, J.M., E. Chong, H.J. Siegel and I. Ray, 2003. Constructing fair exchange protocols for e- commerce via distributed computation of RSA signatures.
9. Saravanan, T. and R. Udayakumar, 2013. Optimization of Machining Hybrid Metal matrix Composites using desirability analysis, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1691-1697.
10. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Detection of Material hardness using tactile sensor, Middle-East Journal of Scientific Research, ISSN:1990-9233 15(12): 1713-1718.
11. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Cross Layer Optimization For Wireless Network (Wimax), Middle-East Journal of Scientific Research, ISSN:1990-9233, 16(12): 1786-1789.