IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON SECURITY AND PRIVACY IN EMERGING DECENTRALIZED COMMUNICATION ENVIRONMENTS

Received September 26, 2019, accepted October 18, 2019, date of publication November 7, 2019,
date of current version December 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952213

DT-CP: A Double-TTPs-Based Contract-Signing Protocol With Lower Computational Cost

GUANGQUAN XU[1, 2], YAO ZHANG[2], LITAO JIAO[1], EMMANOUIL PANAOUSIS[3], KAITAI LIANG[4], HAO WANG[5], AND XIAOTONG LI[2]

[1]Big Data School, Qingdao Huanghai University, Qingdao 266555, China
[2]Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, Tianjin 300354, China
[3]Department of Computing and Information Systems, University of Greenwich, London SE10 9LS, U.K.
[4]Surrey Centre of Cyber Security, University of Surrey, Guildford GU2 7XH, U.K.
[5]Department of Computer Science, Norwegian University of Science and Technology, 7491 Trondheim, Norway

Corresponding author: Yao Zhang (zzyy@tju.edu.cn)

This work was supported by the National Science Foundation of China under Grant 2018YFB0804402.

ABSTRACT This paper characterizes a contract signing protocol with high efficiency in Internet of Things. Recent studies show that existing contract-signing protocols can achieve abuse-freeness and resist inference attack, but cannot meet the high-efficiency and convenience requirement of the future Internet of things applications. To solve this problem, we propose a novel contract-signing protocol. Our proposed protocol includes two main parts: 1) we use the partial public key of the sender, instead of the zero-knowledge protocol, to verify the intermediate result; 2) we employ two independent Trusted Third Parties (TTPs) to prevent the honest-but-curious TTP. Our analysis shows that our double TTP protocol can not only result in lower computational cost, but also can achieve abuse-freeness with trapdoor commitment scheme. In a word, our proposed scheme performs better than the state of the art in terms of four metrics: encryption time, number of exponentiations, data to be exchanged and exchange steps in one round contract-signing.

INDEX TERMS Contract-signing protocol, abuse-freeness, double trusted third parties, trapdoor commitment scheme.

I. INTRODUCTION

Internet of Things (IoT) has been called ''the next Industrial Revolution'' because of the rapid growth and how it's changing the way people live, and interact. However, we might not be ready for this technology improvement because of the trust issue. A recent study had found that less than 4 out of 10 business owners are confident that their organization have implemented perfect user privacy protection infrastructure. Therefore, to protect users' privacy when purchasing such IoT applications, online contract signing protocols are necessary. For example, if a customer needs to subscribe an E-journal, an online contract is required to be signed by the customer and the IoT supporter, and thus, these smart device help you connect to the journal company. However the involved parties may not trust each other if they are strangers to each other. Hence a fair contract-signing protocol is necessary. The protocol's main property is fairness. Intuitively, it means at the end of the protocol, both parties either have valid signatures or none, even if one of them tries to cheat or the protocol interrupts for some reasons. The second property is timely termination, which is to avoid infinite waiting. Both of the two properties are also crucial in fair exchange, certified e-mail, and fair nonrepudiation protocols. The last property is the abuse-freeness, which was introduced by Garay *et al.* [1]. Abuse-free means if the protocol is not executed successfully, none of the involved parties have the ability to show the validity of intermediate results to a third one.

As mentioned above, the core of contract-signing protocol is how to make distrusted parties fairly exchange digital items over the public network [2], [3]. Till now, there are many related researches. Asokan *et al.* [4] proposed a security model for the problem of exchanging digital signatures fairly. Bao *et al.* [5], [6] described a protocol for fair exchange of electronic data between two parties with off-line TTP. Many relevant work about fair exchange is proposed, such as certified e-mail protocols [11]–[13], nonrepudiation protocols [14], authentication protocols [15],and e-payment protocols on the internet [11], [16].

Contract-signing protocols can be classified into three types: 1) protocols without TTP, 2) protocols with online TTP,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaojiang Du.

3) protocols with offline TTP. In [8], [17], authors described the protocols without TTP, in which both parties exchange their signatures "bit by bit". If one side terminates the protocol, both of them have the same fractions of the peer's commitments. Free from TTP is a strong advantage of these protocols, since TTP is a bottleneck of contract-signing protocols. Nevertheless, their shortcomings are also evident. Firstly, the two involved parties should have the same computation capacity. Otherwise the participant who has stronger computing power has more benefit than the other one. Secondly, the efficiency of computation and the exchange during the protocol's execution is low, so that is far from to meet the requirement of real situation. For instance, if Alice and Bob (for convenient, we call the sender Alice and the receiver Bob) are resigning a protocol by TTP-free, whenever any of the two participants terminates prematurely, both of them can still complete the exchange offline by exhaustively searching the remaining bits of the signatures. Therefore, it is impractical for most real-word applications although this approach enjoys the great advantage of being TTP-free. Moreover, both the computation and communication costs of such protocols are high.

The protocols with online TTP [7], [8] are much simpler and more efficient, since a TTP facilitates the execution of the signing process. Under the online TTP setting, a TTP acts as a mediator between two involved parties. However, The main issue of the online TTP protocol is that TTP is likely to become a security and performance bottleneck. Furthermore, the TTP could be rather expensive with its frequent involvement in each step. Thus, the efficient of online TTP protocol is low, especially in the system where there is a large number of participants.

Compared with the previous two schemes, researchers are more prone to the contract-signing protocols with an off-line TTP [16], [18], which are much more practical in most applications. The reason is that the TTP is not invoked in the execution of the protocol unless one of the parties misbehaved or the protocol was interrupted for some reasons. Park *et al.* [16] proposed an approach for constructing fair-exchange protocols. They employed RSA-based multi-signatures to construct efficient optimistic protocols. In this protocol, the signer splits a RSA private key into two parts and keeps both parts while the TTP keeps just one part. Although the protocol is fair and optimistic, the biggest problem is the vulnerability of inference attack. That is, the honest-but-curious [19] TTP can easily figure out the signer's secret key. Dodis and Reyzin [20] showed that the informal connection between non-interactive fair exchange and secure two-signature schemes can be formalized, and provably secure fair exchange protocols can be created. They employed the gap Diffie-Hellman (GDH) groups. However, in their scheme, the pairing computation consumes too much time and performance could be hindered, even though a number of papers have investigated the pairing computation improvement [9], [21].

Moreover, the two schemes in [20] and [16] mentioned above have a common shortcoming, i.e. they are not abuse-free. If a signer Alice can prove to an external party that she is able to get verifiable intermediate results when protocol is unsuccessfully, we say the protocol is not abuse-free. For instance, Alice wants to buy a house from Charlie. At the same time, Bob is selling another house. Alice could first initiate a contract with Bob. Then she terminates the execution of the contract-signing protocol after obtained the intermediate results generated by Bob. As a result, Alice could make a deal with Charlie in a lower price by showing such universally-verifiable proofs to Charlie. We say it is not abuse-free. To solve this problem, Wang [22] proposed a contract-signing protocol for two mutually distrusted parties based on RSA multi-signature. The author integrated an interactive zero-knowledge protocol [23], which is proposed for confirming RSA undeniable signatures to prove the validity of the intermediate results. To realize the abuse-freeness property, he exploited Symmetric Encryption [24] and trapdoor commitment scheme [25] to enhance the zero-knowledge protocol. However, due to the above strategy the exchange requires four more steps and the frequently exchange will reduce the protocol's efficiency although this scheme is abuse-free and fair.

To improve the efficiency of former researches, a new contract-signing protocol is proposed for two mutually distrusted parties in our work. The key contribution of this paper are:

- Our contract-signing protocol employs double offline TTPs. The two offline TTPs are invoked only when the dispute arise. By this offline TTPs structure, the protocol can meet the optimistic property requirement.
- Our protocol is both abuse-free and efficient, because it employs the trapdoor commitment scheme. In more detail, previous research [22] employed the zero-knowledge, which needs more steps so that is not efficient enough. Therefore, double TTPs are applied to substitute zero knowledge protocol and hence improve the efficiency of our proposed contract-signing protocol.
- In order to avoid inference attack, we also use partial private/public keys to make the two TTPs independent. By this method, the protocol is able to prevent the honest-but-curious TTP from acquiring the private key of the users.
- Unlike protocols with single TTP, our double TTPs scheme can prevent an attacker from stealing the partial private key d_2 saved by TTP. In our scheme, every TTP just have a part of d_2, i.e. $d_2$or d_{22}. Therefore even though the attacker obtains one of d_{21} and d_{22}, he cannot get valid d_2.

The rest of the paper is organized as follows. Section 2 describes the related work about the contract signing protocol such as Park *et al.*'s, Wang's schemes, as well as the trapdoor commitment scheme. Section 3 introduces our trapdoor commitment scheme based on the RSA signature, which is significant for attaining abuse-freeness. Then, the security and

efficiency is analyzed in Section 4. Finally, Section 5 gives the conclusion.

II. RELATED WORK

In this section, Park *et al.*'s [16] and Wang's [22] protocols are introduced in detail. After that, we discuss the trapdoor commitment scheme. Park *et al.*'s scheme is based on RSA signature. In this scheme, the private key is split into two parts. First, Alice generates her RSA public key. She sets two primes p and q which are two k-bit safe primes. Then she can get an RSA modulus $n = p \cdot q$ and calculates her private key $d = e^{-1} mod\ \phi(n)$, where $\phi(n) = (p-1)(q-1)$ is Euler's totient function. After that, Alice registers her public key by the certification authority (CA), and the certification authority return a certificate. Alice needs to split the private key d into d_1 and d_2 randomly. Note that d_1 and d_2 satisfy $d = d_1 + d_2 mod\ \phi(n)$, where $d_1 \in_R \mathbb{Z}^*_{\phi(n)}$. Afterwards Alice computes $e_1 = d_1^{-1} mod\ n$ and $e_2 = d_2^{-1} mod\ n$. In order to get voucher V_A from the TTP, Alice needs to send (C_A, e_1, d_2) to the TTP. After that, the TTP checks C_A, e_1, and d_2. If all those verification go through, the TTP returns V_A to Alice. In the exchange phase, Alice computes $\sigma_1 = h(m)_{d_1} mod\ n$, and sends (C_A, V_A, σ_1) to Bob. After verifying the C_A and V_A, Bob returns σ_B. In last step, Alice sends σ_2 to Bob, then Bob obtains $\sigma_A = \sigma_1 + \sigma_2$.

Unfortunately, the protocol is not safe enough because the inference attack that an honest-but-curious TTP can derive Alice's private key. More specifically, the TTP knows that the integer $e - (1 - ed_2)e_1$ is a nonzero multiple of ϕ_n. We call this problem inference attack in the following section. In that case, Alice's RSA modulus n can be easily factored so that the TTP can get Alice's private key d by the extended Euclidean algorithm. Besides, the protocol is not abuse-free though it is fair and optimistic.

Wang [22] improved the Park *et al.*'s protocol. He proposed an abuse-free contract-signing protocol with zero-knowledge protocol. In Wang's protocol, the initiator Alice also randomly splits the private key d into d_1 and d_2, i.e., $d = d_1 + d_2 mod\ \phi(n)$. Then, she sets $p = 2p' + 1$, $q = 2q' + 1$, $n = pq$. At the same time, she generates a sample message-signature pair (ω, σ_ω), where $\omega \in \mathbb{Z}^*_n \backslash 1. - 1$. And then, she sends $(C_A, \sigma_\omega, \omega, d_2)$ to the TTP but keeps (d, d_1, d_2, e_1) as secrets. The TTP first checks if the triple $(C_A, \omega, \sigma_\omega)$ is prepared correctly. After that, TTP stores d_2 securely. Then TTP creates a voucher V_A, and sends it to Alice. In the signature exchange protocol, Alice firstly sends (C_A, V_A, σ_1) to BOb. After Bob receives the secret, he sets $c = \sigma_1^{2i}\sigma_\omega^\omega mod\ n$ c is a challenge of the trapdoor commitment and Bob can verify the availability of the σ_1 by zero knowledge protocol without e_1. Due to zero knowledge protocol, neither Bob nor TTP can obtain e_1. So the honest-but-curious TTP cannot derive the private key of Alice by $e - (1 - ed_2)e_1$, i.e. the reference attack is avoided. Although Wang's protocol is fair, optimistic, abuse-free, secure and of anti-inference attack, it is not efficient enough, since the trapdoor commitment and zero knowledge protocol require four more exchange steps in the signature exchange phase.

In our scheme, double TTPs are used to prevent the inference attack. But double TTPs are not enough to ensure the abuse-freeness of the protocol. Therefore, the trapdoor commitment scheme is also adopted, but this method is employed in a different way. Now we introduce the concept of trapdoor commitment in advance.

A commitment scheme is a basic cryptographic model in the field of cryptography. This model can be considered to be a sealed envelope. If one party, for example Alice wants to commit a message M, she can put M into a sealed envelope. Whenever Alice would like to publish M, she just needs to open the envelope. Generally speaking, commitment scheme should satisfy the "hiding property" and the "binding property". The former one means that no one can get any information about M from the envelope except for Alice. The later one means that Alice cannot change M after she has made the commitment, and she cannot cheat the receiver since the receiver can verify whether Alice has committed M. Trapdoor commitment [26]–[28] is a commitment scheme with special properties, that is, one with the trapdoor key can open his commitment in different ways. In other word, given the trapdoor key, this trapdoor commitment does not satisfy the binding property as traditional commitment schemes. Hence, if the receiver is the owner of the trapdoor, he can change the message which has been committed. Due to this property, a third party cannot make sure that the answer is the real one revealed by the sender. Actually, thanks for the property that a third party cannot verify the availability of intermediate results, our protocol can satisfy the abuse-freeness. A trapdoor commitment (TC) consists of four parts, TC = (TCgen, TCcom, TCvrfy, TCsim). TCgen is the generation algorithm, which is responsible for generate a key pk and a corresponding commitment for the receiver. Actually, the receiver is the owner of the TC. TCcom can output a pair (C, t) for the input (pk, r), where value C is the commutment and value s is the related information to decommit C. Given an input C, related information t is the related information to decommit C. Given an input pk, commitment C, related information t and message M, verification algorithm TCvrfy can check whether a pair (r, t) is valid to the given commitment C. The simulate algorithm TCsim allows the receiver to simulate a "fake" answer (r', t) for the given commitment C while one answer (r, t) has been given. Fortunately, the trapdoor commitment used in this protocol can be find easily. In [18], [27] the author reported how to construct trapdoor commitment scheme from RSA assumption, while [18], [29] is a discrete log(DL)-based instantiation using a free trapdoor commitment scheme, which can be used in the Schnorr signature, EIGamal signature, or DSA, etc. So, no matter which kind of public key published by the receiver, at least one trapdoor commitment scheme can be used. From what we mentioned, we know how to prevent TTP from deriving Alice's private key with the knowledge of (n, e, e_1, d_2). In Wang's protocol, Alice hides the partial

public key e_1 from the TTP and Bob, replaced with the zero-knowledge protocol to prove the availability of σ_1 . In this paper, we propose a more efficient scheme without losing abuse-freeness and fairness.

III. IMPROVEMENT ON TRADITIONAL CONTRACT-SIGNING PROTOCOL USING DOUBLE TTPS

In this section, we describe our contract-signing protocol with double TTPs, which satisfies the following cardinal principles of the contract-signing protocols [16], [22].

- Fairness: fairness is the fundamental property of the contract-signing protocol, since the first task of a contract-signing is to ensure the fairness between users. In this point, fairness means both of the two parties have the same advantages. If one of them cheats, he/she cannot acquire any benefit from the other one.
- Optimism: In this scheme, we adopt two off-line TTPs, which are awaked only when one party is cheating or other disputes arise. However, disputes are rare incidents due to the fact that the cheater cannot get any benefit. We call the property of the offline TTP Optimism property.
- Abuse-freeness: in contract-signing protocol, we need to ensure that any of the two parties cannot show the validity of the intermediate results to outside parties when the protocol executes unsuccessfully. In this paper, we realize abuse-freeness by using the trapdoor commitment scheme.
- Timely Termination: This property implies that the protocol defined a deadline T. If one of the parties does not send his signature after T, the other party does not need to wait, and both of them can be free from the protocol.
- TTPs' Statelessness: In this protocol, our double TTPs do not need to maintain a database to search or remember the state information. Because of this property, the burden of the TTPs is reduced and the cost of the TTPs will much lower.
- High Efficiency: In [22], the protocol execution in a normal status requires seven steps. Our scheme is more efficient because just three rounds are required in the exchange protocol. In most instances, the protocol executes in the normal status. Therefore, if we reduce the steps of the exchange protocol, the efficiency can be greatly improved.

The main process of our protocol is showed in Figure 1. The sender Alice first executes the registration protocol to register with the double TTPs. She splits her private key d into d_1 and d_2, where $d = d_1 + d_2 \, mod \, \phi(n)$. Furthermore, Alice sequentially splits d_2 into d_{21} and d_{22} satisfying $d_2 = d_{21} + d_{22} \, mod \, \phi(n)$. After that, d_{21} and d_{22} are respectively delivered to the TTP_1 and TTP_2, while Alice keeps (d, d_1, d_{21}, d_{22}) as secrets. At the same time, Alice sends e_1 to TTP_1 and TTP_2. After that, TTP_1 computes the trapdoor commitment $\overline{r}_1 = TCcom(e_1, t_1)$, and TTP_2 computes $\overline{r}_2 = TCcom(e_1, t_2)$, where t_1 and t_2 are the related information to decommit the commitment $\overline{r}_1$ and $\overline{r}_2$ respectively.

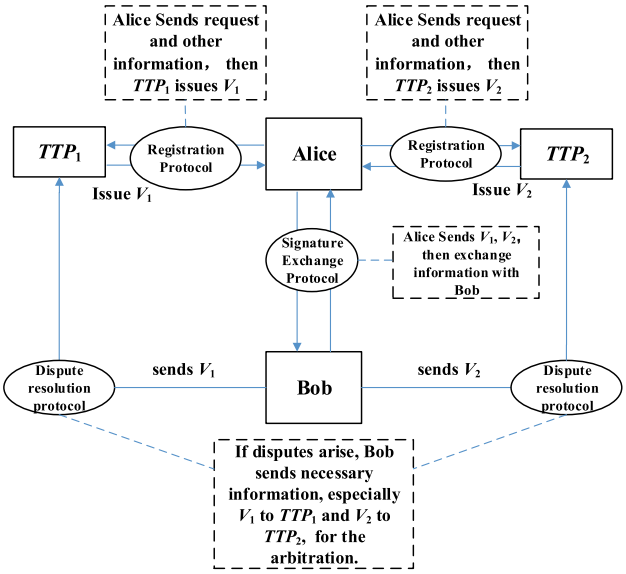


FIGURE 1. The general frame of contract signing protocol with double TTP.

Note that the commitment $\overline{r}_1$ and $\overline{r}_2$ depends on Bob's signature, which can commit to the partial public key e_1. To exchange her signature $\sigma_A = h(m)^d \, mod \, n$ with Bob's signature, Alice needs to execute signature exchange protocol with Bob. In this sub-protocol, Alice first sends her partial signature $\sigma_1 = h(m)^{d_1} \, mod \, n$, $\overline{r}$ and (e_1, t) to Bob. Then, Bob checks e_1 committed by double TTPs so that he can verify whether σ_1 is available. Finally, Bob sends his signature σ_B to Alice because he is sure that he can obtain the second partial signature σ_2 from TTP_1 and TTP_2. Generally speaking, we assume that the communication channel between Alice and Bob is unreliable, i.e., the message transmitting in this channel may be lost due to the failure of computer network or attacks. Nevertheless, the channel between the involved parties and TTP is reliable, i.e., the message will be delivered to the recipient after a finite delay.

A. REGISTRATION

To exchange digital signatures, the initiator Alice is required to register to TTPs, so that she can get the vouchers V_1 and V_2 from double TTPs, and a certificate C_A from CA. The voucher can be regraded as a kind of certificate for the registration steps. The registration protocol is briefly illuminated in Figure 2, and the detailed procedure is elaborated as follows.

1) In the general RSA key generation process, Alice first sets a RSA modulus $n = pq$, where p and q are two k-bit safe primes. For instance, she sets two primes p', q', and the safe prime satisfy $p = 2p' + 1$, $q = 2q' + 1$. After that, Alice randomly selects her public key $e \in_R \mathbb{Z}^*_{\phi(n)}$, and calculates her private key $d = e^{-1} \, mod \, \phi(n)$, where $\phi(n) = (p-1)(q-1)$. At last, Alice registers her public key to a CA for a certificate C_A, which includes her identity and her public key pair (n, e).

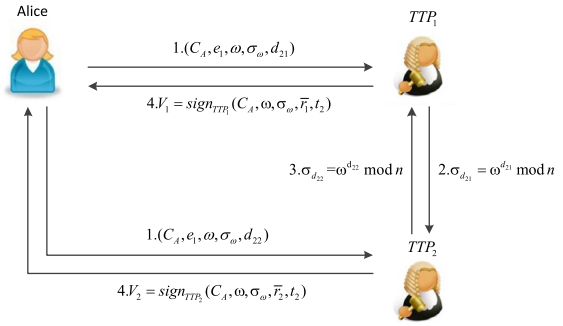


FIGURE 2. Registration protocol.

2) Aice first splits d into d_1 and d_2 randomly, and then splits d_2 into d_{21} and d_{22}. Meanwhile, she generates a sample-signature pair ω, ϕ_ω, where $\omega \in \mathbb{Z}_n^* \setminus 1, -1, ord(\omega) \geq or \geq p'q', \sigma_\omega = \omega^{d_1} \bmod n$. Finally, Alice sends $(C_A, e_1, \omega, \sigma_\omega, d_{21})$ to TTP_1 and $(C_A, e_1, \omega, \sigma_\omega, d_{22})$ to TTP_2.
3) TTP_1 and TTP_2 checks whether Alice's certificate C_A is valid and $\sigma_\omega^{e_1} \equiv \omega(mod\ n)$ or not. If yes, TTP_1 computes $\sigma_{d_{21}} = \omega^{d_{21}} \bmod n$, and sends $\sigma_{d_{22}}$ to TTP_1.
4) TTP_1 and TTP_2 check whether σ_ω is constructed correctly by verifying the following congruence relations: $\omega = \sigma_\omega^{e_1} \bmod n$, $\sigma_\omega^e \cdot (\sigma_{d_{21}} \cdot d_{22})^e \equiv \omega(mod\ n)$. If the verification is correct, both TTP_1 and TTP_2 accept Alice's claim that the congruence relations of $e_1 \cdot d_1 \equiv 1(mod\ \phi(n))$ and $(d_1 + d_{21} + d_{22}) \cdot e \equiv 1(mod\ n)$ hold. Then, TTP_1 stores d_{21} and TTP_2 stores $d_2 2$ securely. Meanwhile, TTP_1 generates a commitment $\overline{r}_1 = TCcom(e_1, t_1)$ and TTP_2 generate a commitment $\overline{r}_2 = TCcom(e_1, t_2)$ by selecting a random number t_1 and t_2, in which t_1 and t_2 are the related information used to decommit the trapdoor commitment $\overline{r}_1$ and $\overline{r}_2$. As the aforementioned trapdoor commitment scheme, the TCcom is the commitment algorithm of a secure trapdoor commitment scheme which depends on Bob's public key. After that, the two TTPs generate their voucher V_1 and V_2 respectively, by computing $V_1 = sign_{TTP_1}(C_A, \omega, \sigma_\omega, \overline{r}_1, t_2)$ and $V_2 = sign_{TTP_2}(C_A, \omega, \phi_\omega, \overline{r}_2, t_2)$. At last, TTP_1 sends V_1 and TTP_2 sends V_2 to Alice.

We give some notes on the registration parts mentioned above. In step (1), Alice gets her certification from a CA so that she can prove that the modulus is the product of two safe primes. In fact, if Alice has obtained C_A from CA, the step (1) can be omitted. After Alice sends $(C_A, e_1, \omega, \sigma_\omega, d_{21})$ to TTP_1 and $(C_A, e_1, \omega, \sigma_\omega, d_{22})$ to TTP_2, the two TTPs verify $\omega \in \mathbb{Z}_n^* \setminus \{1. -1\}$, and ensure both $gcd(\omega - 1, n)$ and $gcd(\omega + 1, n)$ are not prime factors of n. After step (3), Alice proves to TTP_1 and TTP_2 that she knows the discrete logarithm of σ_ω to the base ω without revealing d_1. This can be done via the zero-knowledge protocol in [30]. Finally, once the verification of step (4) is passed, TTP_1 and TTP_2 generate the voucher V_1 and V_2 respectively for the Alice. The registration part above just needs to work only when the first time Alice executes the protocol, and the vouchers generated for Alice can be used to the next contracts. Therefore, Alice can fairly sign contracts for a long period.

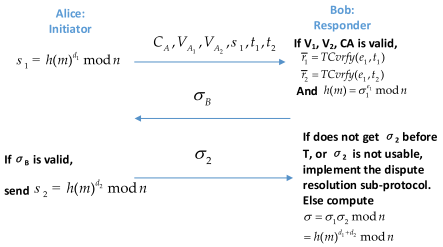


FIGURE 3. Signature exchange protocol.

B. SIGATURE EXCHANGE

When Alice and Bob initiate the protocol, We assume that Alice and Bob have gone through a negotiation on the contract M, which may contain the identity of Alice and Bob, and contain the double TTPs, as well as the deadline T. The signature exchange protocol is depicted in Figure 3.

1) First, the initiator Alice generates her partial signature σ_1 by computing $\sigma_1 = h(m)^{d_1} \bmod n$. Then she sends the seven tuple $(C_A, V_1, V_2, \sigma_1, e_1, t_1, t_2)$ to Bob, which is committed by TTP_1 and TTP_2.
2) After receiving $(C_A, V_1, V_2, \sigma_1, e_1, t_1, t_2)$ from Alice, Bob is required to verify that C_A is the real certificate of Alice. Then, Bob checks if the identity of Alice, Bob, and TTP are correctly descripted corresponding to M.
3) Bob also checks whether V_1 is signed by TTP_1 and V_2 is signed by TTP_2. Then Bob needs to decommit $\overline{r}_1$ and $\overline{r}_2$ via the trapdoor commitment verification algorithm TCvrfy. Bob is required to check $\overline{r}_1 = TCvrfy(e_1, t_1)$ and $\overline{r}_2 = TCvrfy(e_1, t_2)$. If the verification returns true, Bob can confirm that e_1 is the real partial public key of Alice. Then, he verifies the partial signature σ_1 by checking $h(M) \equiv \sigma_1^{e_1}(mod\ n)$.If the verification is correct, Bob computes $\sigma_B = h(M)^{d_B} mod\ n$, and sends his signature σ_B on contract M to Alice, since he is convinced that another partial signature σ_2 can be released by the double TTPs, in case Alice might refuse to do so.
4) Upon receiving σ_B, Alice checks whether it is Bob's valid signature on contract M. If it goes well, Alice sends her partial signature σ_2 to Bob by computing $\sigma_2 = h(m)^{d_2} mod\ n$. While Bob obtains σ_2, he checks whether $h(m) \equiv (\sigma_1\sigma_2)^e mod\ n$. If the equation holds, Bob can get Alice's standard RSA signature σ_A on message M from $\sigma_A = (\sigma_1\sigma_2)^e mod\ n$. On the contrary, if σ_2 is not be successfully received by Bob, or Alice cheats during the protocol, Bob can turn to the double TTPs. We explain more detail on the above signature exchange protocol in the following section.

Firstly, the trapdoor commitment scheme in step (3) is significant in the exchange part. For one thing, the trapdoor commitments committed by TTP_1 and TTP_2 is used to verify

TABLE 1. Property Comparison.

scheme	p1	p2	p3	p4	p5	p6
Park et al.	YES	YES	NO	NO	NO	YES
Bodkhe et al.	YES	NO	NO	NO	NO	NO
Alawi et al.	YES	YES	YES	NO	YES	NO
Wang	YES	YES	YES	YES	YES	NO
ours	YES	YES	YES	YES	YES	YES
p1:fairness; p2:optimism; p3: abuse-freeness; p4:timely terminatin; p5: TTPs' statelessness; p6: High efficiency						

the validity of e_1 in case that Alice might forge a fake e_1. For another, it also enhances the security of the above verification of σ_1. Specifically, using a commitment scheme can prevent Bob from forwarding the intermediate results to convince an outsider of the validity of σ_1. Because of the special property of the trapdoor commitment, Bob cannot collude with one or more outside parties by leaking Alice's partial public key e_1. Actually, the trapdoor commitment scheme relies on Bob's public key, but it requires some extra parameters, i.e. s and u. In [29], the two parameters of trapdoor commitment should be given by receivers. However, in [18], [26] the parameters s and u are set before the protocol's execution. In our protocol, the parameters also can be set as a fixed value, e.g. $s = 2$ and u a fixed 160-bit prime number, for all receivers who employ RSA signatures. In this way, the two TTPs who only know the receiver Bob's RSA public key can run the trapdoor commitment scheme without asking for the parameters from Bob. Last but not least important, although Alice is required to register with the TTPs, Bob does not need to do so due to that the double TTPs do not need to conserve Bob's identity information.

C. DISPUTE RESOLUTION

After Bob submits his signature σ_B to Alice, he may not receive the Alice's signature σ_1 or may receive invalid signature from Alice. If the dispute arises, Bob can turn to TTP for assistance before the deadline T for the Alice's valid partial signature σ_2 via dispute resolve protocol. This protocol is shown in Figure 4 and more detail is depicted as follows.

1) Bob first sends $(C_A, V_{A_1}, M, \sigma_1, e_1, \sigma_B)$ to TTP_1 and $(C_A, V_{A_2}, M, \sigma_1, e_1, \sigma_B)$ to TTP_2 . Later, TTP_1 checks whether C_A and V_{A_1} are Alice's valid certificate and voucher on contract M. Besides, he also verifies the availability of Bob's signature σ_B. Meanwhile, TTP_2 checks the availability of C_A, V_{A_2}, and σ_B like TTP_1 does. If the verification is correct, the two TTPs need to check the contract M to make sure that the deadline T contained in contract M is not expired and to make sure that both the sender Alice, and the receiver Bob are the right parties specified in M. If any of the verifications mentioned above fails, TTP_1 and TTP_2 send errors to Bob.
2) Ater that, TTP_1 computes $\sigma_{21} = h(m)^{d_{21}} \ mod \ n$. Then TTP_1 sends σ_{21} to TTP_2. Upon receiving σ_{21} from TTP_2, TTP_2 sends σ_{22} to TTP_1. Whereafter, TTP_1 and TTP_2 can compute $\sigma_2 = \sigma_{21}\sigma_{22}$ mod n. Accordingly, they can check whether $h(m) \equiv (\sigma_1\sigma_2)^e$ mod n. If this equality holds, TTP_1 and TTP_2 send (M, σ_{21}) and (M, σ_{22}) to Bob respectively. Simultaneously, both of the two TTPs send (M, σ_B) to Alice. However, if the preceding equation does not hold, the two TTPs send errors to Bob.

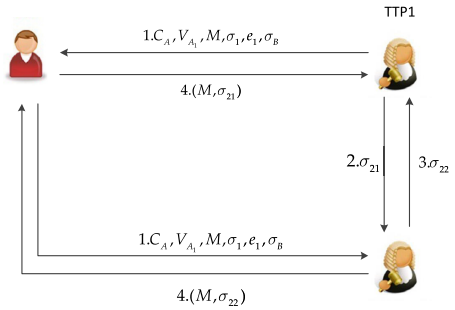


FIGURE 4. Dispute resolution protocol.

IV. DISCUSSION

A. SECURITY DISCUSSION

The proposed scheme satisfies several security properties shown in Table 1. We compare our scheme with recently proposed schemes. Obviously, the table demonstrates that our scheme satisfies the proposed six properties, overperforming the other four previous researches. In the following section, We discuss the critical part of the security properties of our protocol in detail.

1) ABUSE-FREENESS

First, we specify the abuse-freeness. During the signature exchange protocol, after Bob received $(C_A, V_1, V_2, \sigma_1, e_1, t_1, t_2)$ from Alice, he is required to check whether e_1 is valid via the trapdoor commitment verification algorithm $\bar{r}_1 = TCvrfy(e_1, t_1)$ and $\bar{r}_2 = TCvrfy(e_1, t_2)$. However, if Bob wants to sell the partial signature σ_1 to another party Charlie for some benefits, he might hardly success. If Charlie gets σ_1 with the proof $(C_A, V_1, V_2, \sigma_1, t_1, t_2)$, he needs firstly check the availability of e_1 by $\bar{r}_2 = TCvrfy(e_1, t_1)$ and $\bar{r}_2 = TCvrfy(e_1, t_2)$. But actually, Bob could use the trapdoor td to simulate new answers (e_1', t_1') and (e_1', t_2'). On this occasion, Charlie cannot confirm whether e_1 is a valid partial public key of Alice. Therefore, Bob is not able to convince Charlie that σ_1 is a real partial signature of Alice. It means that our contract-signing protocol is also abuse-free even we do not use zero knowledge protocol like Wang [22] did.

2) ANTI-REFERENCE ATTACK

Furthermore, our protocol also avoids inference attack vulnerbility in Park *et al.*'s protocol [16], which is pointed out by Wang. That is, if Alice is honest, an honest but curious TTP can derive Alice's private key d from d_2 due to that $e-(1-ed_2)e_1$ is a nonzero multiple of $\phi(n)$. It is dangerous if anybody except for Alice herself has Alice's private key d. Because an attacker may steal the private key by a variety of approaches from the one who knows d, e.g. TTP, but Alice is ignorant of anything. In our protocol, we assume that the two TTPs are independent and do not collude with each other. Alice divides the partial private key d_2 into d_{21} and d_{22} so that each of the two TTPs just has a part of the d_2. Since we assume mutual collusion does not exist between the two TTPs, neither TTP_1 nor TTP_2 can obtain the complete d_2. Hence, they cannot derive Alice's private key d by inferring from the integer $e-(1-ed_2)e_1$. In addition, the deadline T embedded in the contract M is a reasonable predetermined value. The protocol should not be terminated beyond the deadline T. Therefore, if the dispute arises, the receiver Bob must recall the two TTPs before T. If the protocol expires, each involved party is free of liability. Actually, after the deadline T, TTP_1 and TTP_2 will not accept any dispute resolution request about previous protocol. On this account, Bob has to awaken the double TTPs before deadline T in case that the time expires but the disputes are not solved.

3) FAIRNESS

Except for the above discussion, fairness is a vital portion of a contract-signing protocol. In the following parts, we mainly discuss the fairness property in our contract-signing protocol. Fairness here means that any of the involved two parties cannot take advantage of the other one even though one of them cheats during the exchange protocol. This property can be divided into two points. For one thing, Alice is an honest one, but Bob is a cheater. For another, Bob is an honest one, but Alice is a cheater. We will analysis the fairness from the two contrary points. In the first place, we assume that Alice is honest, but Bob cheats. Note that Bob or any other adversary cannot forge the σ_1, σ_2 and M. Therefore, only Alice herself can generate a valid σ_1 and σ_2. Similarly, nobody can generate σ_{21} and σ_{22} except for Alice and the two TTPs. In the signature exchange protocol, Alice first sends $(C_A, V_1, V_2, \sigma_1, e_1, t_1, t_2)$ to Bob. When Bob receives σ_1, he is required to decide whether send σ_B to Alice. If Bob does not send his signature σ_B or send an invalid signature to Alice, Alice will not return σ_2. In that case, Bob certainly cannot obtain Alice's signature σ_A. Then, if Bob executes the dispute resolution protocol before deadline T, he has to send his signature σ_B to TTPs in order to get σ_2. Eventually, both Alice and Bob can get the other's signature on M. In addition, we assume that Bob just lie to one TTP but honest to the other one, for example, he sends an invalid signature to TTP_1 but sends the valid signature σ_B to TTP_2. In that case, Bob can only get σ_{21}, but cannot get σ_{22}, so that he cannot obtain σ_A. However, TTP_1 will send (σ_B, M) to Alice. Obviously, no matter which TTP Bob cheats, Bob cannot get any benefit.

In another case, we assume that Bob is honest, while Alice is a cheater. For example, in step (2) of the registration protocol, Alice sends an invalid partial public key $e_1{}'$ to TTP_1 and TTP_2. The double TTPs need to check whether σ_ω is valid in advance by zero knowledge protocol [30]. If the verification goes through, the two TTPs check whether $\omega \equiv \sigma_\omega{}^{e_1{}'} (\mathrm{mod} n)$. Eventually, the two TTPs cannot send vouchers to Alice because $\omega \neq \sigma_\omega{}^{e_1} (\mathrm{mod} n)$. Besides, if Alice sends an invalid partial private key $d_{21}{}'$ and $d_{22}{}'$ to double TTPs respectively, she cannot pass the verification in steps (4), since $\sigma_\omega^e \cdot (\sigma_{d_{21}{}'} \cdot \sigma_{d_{22}{}'})^e \neq \omega (\mathrm{mod} n)$. In addition, if Alice cheats in the signature exchange protocol, e.g. she does not send her partial signature σ_2, Bob can execute the dispute resolution protocol. From what we discuss before, we can conclude that our contract-signing protocol does not favor to any parties. Namely, our protocol satisfies fairness.

B. PERFORMANCE ANALYSIS AND COMPARISONS

In this section, the perfomance of our protocol is analyzed in the normal case, i.e., the operation of the dispute resolution protocol is not included. Moreover, we add the encryption time and the exchange frequency as new criteria, which is not mentioned in Park's and Wang's work. We evaluate the efficiency of our scheme in terms of four criteria:

- encryption time of every encrption scheme in signature exchange protocol.
- Number of modular exponentiations required for signature exchange protocols by Alice and Bob.
- Size of the data to be exchanged.
- Number of exchange steps in one round contract-signing.

Note that we assume the length of RSA modulus n is 1200 bit, and that the hash function $h(\cdot)$ has 256-bit fixed output. We can easily calculate the size of the data to be exchanged in the signature exchange subprotocol. As in Figure 5 item ''encryption time'', we can see that the data size of our protocol is much smaller than Alawi's scheme, and almost equal to Bodkhe's and Wang's scheme. Whereas, Park's scheme is smaller. It is because Park did not adopt the commitment scheme so that fails to realize the abuse-freeness property. Actually, our protocol need to exchange smaller data than Wang's and the others without losing the abuse-freeness. In order to clearly compare the encryption overide between the others' work and ours, we made a simulation encryption experiment to compute the encryption time of the protocols(shown in Table 2). As the table shows, the TTP involved scheme cause extra computational overhead, i.e. Bodkhe *et al.*'s work since this scheme needs TTP's participation in each signature exchange. For a more precise comparison, we simulate the cryptographic oprations mentioned in the Table 2 using a pycrypto cryptographic library. In addition, we assume that the symmetric encryption are implemented by the SHA-256. As shown

TABLE 2. Encryption time of protocols.

scheme	TTP	Alice	Bob
Park et al.	N/A	$2t_{RSA} + 1t_{hash} + 1t_{dRSA}$	$1t_{RSA} + 1t_{hash} + 2t_{dRSA}$
Wang	N/A	$2t_{RSA} + 2t_{com} + 1t_{dRSA}$	$2t_{RSA} + 1t_{com} + 1t_{dRSA}$
Bodkhe et al.	$2t_{RSA} + 2t_{dRSA}$	$1t_{RSA} + 1t_{hash} + 1t_{dRSA}$	$1t_{RSA} + 1t_{hash} + 2t_{dRSA}$
Alawi et al.	N/A	$2t_{RSA} + 2t_{hash} + 2t_{com} + 2t_{dRSA}$	$2t_{RSA} + 1t_{hash} + 2t_{dRSA} + 2t_{com}$
ours	N/A	$2t_{RSA} + 1t_{hash} + 1t_{dRSA}$	$1t_{RSA} + 1t_{hash} + 2t_{dRSA}$
t_{RSA}:RSA scheme; t_{dRSA}:RSA decryption scheme; t_{com}:commitment scheme; t_{hash}:hash function			

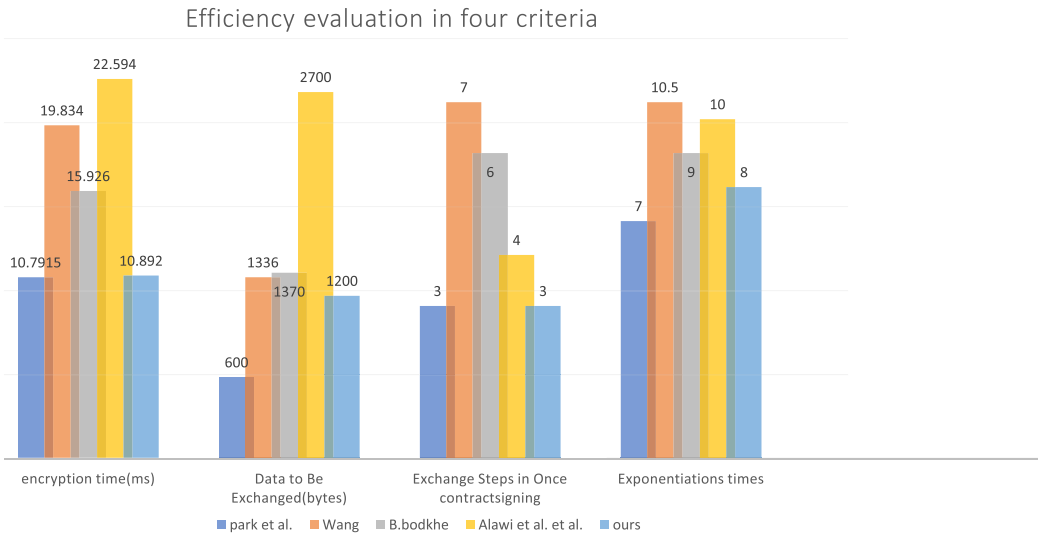


FIGURE 5. Efficiency evaluation in four criteria.

in Figure 5, the Park *et al.*'s scheme takes 10.7915ms and B.Bodkhe's scheme needs 15.926ms. Especially, the Wang's takes 19.834 and Alawi's take 22.594, which are twice of us. It clearly denote that, except the Park's scheme, the proposed scheme causes much more computational overhead than ours.

Note that the most computationally expensive cryptographic operation is modular exponentiation in the finite field $\mathbb{Z}_n$. So, we take the number of modular exponentiations as the computational cost. Similar to Wang's scheme, we assume that σ_B could be generated and verified by one modular exponentiation separately, as in [22] and [16]. Figure 5 item "Exponentions time" compares the number of exponentiations between the proposed scheme and ours. Obviously, our scheme needs less exponentiation than Wang's, Bodkhe's and Alawi's.

Except for the mentiond criterion, the number of exchange steps in one round contract-signing is a fourth criterion, which is different from the others' scheme. Because it can directly influence the efficiency of the protocol, the information translation would consume more time during the protocol execution. Figure5 item "Exchange Steps in Once Contract-signing" shows the comparison of exchange steps between our protocol and other RSA-based protocols. In general, our scheme improves the efficiency of Wang's protocol without losing the excellent property, i.e. abuse-freeness and security.number of exchange steps in one round contract-signing in our scheme is decreases by 57% compared with Wang's scheme. In other word, there are only three exchange steps in our scheme in normal case, which is much less than seven exchange steps in Wang's scheme.

What need to be explained is that Park's scheme seems to be more outstanding than ours in the four criterion. But actually, this scheme has some shortcomings. For example, it did not realize the abuse-freeness in this scheme. That is, the receiver Bob can easily prove that he has the ability to get the intermediate product of sender Alice in this protocol since he did not adopt trapdoor commitment scheme. Moreover, Park *et al.* did not resolve the inference attack mentioned before, while our scheme completely solves the problem via double TTPs. From the above, we can say that our scheme outperform the others' schemes except for park's as shown in Figure 5, but does not have the significant vulnerability in Park's scheme.

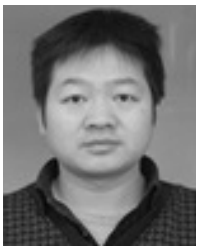
V. CONCLUSION AND FUTURE WORK

In this paper, a novel contract-signing protocol with two TTPs is proposed for Interent of Things in higher efficiency and convenience. Like the existing RSA-based solutions, our new protocol is fair and optimistic. Namely, both of the two involved parties have valid signatures of the other one or neither does, and the two TTPs are only involved when dispute arise occasionally. Furthermore, our protocol employs trapdoor commitment to achieve abuse-freeness and adopts double TTPs to resolve the security risk which we called inference attack. By this way, the interactive zero-knowledge protocol is not required. Therefore, our protocol merely needs three exchange steps in the normal case, rather than seven steps in Wang's scheme, and hence improves the efficiency. In addition, our approach has other advantages. Note that the two TTPs are stateless in our contract-signing protocol, because it does not keep any state information regarding to each protocol instance. Moreover, our protocol can cooperate with coverless information technology [31] and automatic network management [32], [33] in fair payments of e-commerce. Based on this, we can set online bank and the market administrator, e.g. Amazon or eBay, as two TTPs. One customer purchases goods from a merchant via internet by paying digital check. Our proposed protocol can improve the efficiency and convenience in such Iot-transactions.

Finally, our protocol allows two potentially mistrusted parties to exchange their signatures with double TTPs. However, the protocol should be extended for the scenario of multiparty in order to satisfy more requirement in real life. Besides, though the double TTPs is honest, hidden dangers are also existing, i.e. collusion attack, since the two TTPs can collusively get partial private key d_2 of Alice. Therefore, we could try to hide the d_{21} and d_{22} or using the proxy private key to prevent the collusive attack.

REFERENCES

[1] J. A. Garay, M. Jakobsson, and P. MacKenzi, "Abuse-free optimistic contract signing," in *Advances in Cryptology-CRYPTO*, vol. 99. Berlin, Germany: Springer, 1999, pp. 449–466.
[2] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Trans. Commun.*, vol. E98-B, no. 1, pp. 190–200, 2015.
[3] G. Xu, Y. Cao, Y. Ren, X. Li, and Z. Feng, "Network security situation awareness based on semantic ontology and user-defined rules for Internet of Things," *IEEE Access*, vol. 5, pp. 21046–21056, 2017.
[4] N. Asokan, V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 593–610, Apr. 2000.
[5] F. Bao, G. Wang, J. Zhou, and H. Zhu, "Analysis and improvement of Micali's fair contract signing protocol," *Inf. Secur. Privacy*, vol. 31, no. 24, pp. 176–187, 2004.
[6] F. Bao, "Colluding attacks to a payment protocol and two signature exchange schemes," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2004, pp. 417–429.
[7] M. Ben-Or, O. Goldreich, S. Micali, and R. L. Rivest, "A fair protocol for signing contracts," *IEEE Trans. Inf. Theory*, vol. 36, no. 1, pp. 40–46, Jan. 1990.
[8] I. B. Damgård, "Practical and provably secure release of a secret and exchange of signatures," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, 1993, pp. 200–217.
[9] S. D. Galbraith, K. Harrison, and D. Soldera, "Implementing the tate pairing," in *Proc. Int. Algorithmic Number Theory Symp.* vol. 2369. Berlin, Germany: Springer, 2002, pp. 324–337.
[10] S. Micali, "Simple and fast optimistic protocols for fair electronic exchange," in *Proc. 22nd Annu. Symp. Princ. Distrib. Comput.*, 2003, pp. 12–19.
[11] M. Abadi and N. Glew, "Certified email with a light on-line trusted third party: Design and implementation," in *Proc. Int. Conf. World Wide Web*, 2002, pp. 387–395.
[12] G. Ateniese and C. Nita-Rotaru, *Stateless-Recipient Certified E-Mail System Based on Verifiable Encryption*. Berlin, Germany: Springer, 2002.
[13] K. Imamoto and K. Sakurai, "A certified e-mail system with receiver's selective usage of delivery authority," in *Proc. Int. Conf. Cryptol., Progr. Cryptol.*, 2002, pp. 326–338.
[14] S. Gürgens, C. Rudolph, and H. Vogt, "On the security of fair non-repudiation protocols," *Int. J. Inf. Secur.*, vol. 4, no. 4, pp. 253–262, 2005.
[15] X. Zeng, G. Xu, Z. Xi, X. Yang, and W. Zhou, "E-AUA: An efficient anonymous user authentication protocol for mobile IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1506–1519, Apr. 2019.
[16] J. M. Park, E. K. P. Chong, and H. J. Siegel, "Constructing fair-exchange protocols for E-commerce via distributed computation of RSA signatures," in *Proc. 22nd Annu. Symp. Princ. Distrib. Comput.*, 2003, pp. 172–181.
[17] Z. Wan, R. H. Deng, and D. Lee, "Electronic contract signing without using trusted third party," in *Proc. Int. Conf. Netw. Syst. Secur.*, 2015, pp. 386–394.
[18] G. Xu, Y. Zhang, A. K. Sangaiah, X. Li, A. Castiglione, and X. Zheng, "CSP-E^2: An abuse-free contract signing protocol with low-storage TTP for energy-efficient electronic transaction ecosystems," *Inf. Sci.*, vol. 476, pp. 505–515, Feb. 2019.
[19] B. Gilburd, A. Schuster, and R. Wolff, "k-TTP: A new privacy model for large-scale distributed environments," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 563–568.
[20] Y. Dodis and L. Reyzin, "Breaking and repairing optimistic fair exchange from PODC 2003," in *Proc. 3rd ACM Workshop Digit. Rights Manage.*, 2003, pp. 47–54.
[21] P. S. L. M. Barreto, S. D. Galbraith, C. ÓhÉigeartaigh, and M. Scott, "Efficient pairing computation on supersingular abelian varieties," *Des., Codes Cryptogr.*, vol. 42, pp. 239–271, Mar. 2007.
[22] G. Wang, "An abuse-free fair contract-signing protocol based on the RSA signature," *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 1, pp. 158–168, Mar. 2010.
[23] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *J. Cryptol.*, vol. 1, no. 2, pp. 77–94, 1988.
[24] J. Li, Y. Huang, Y. Wei, S. Lv, Z. Liu, C. Dong, and W. Lou, "Searchable symmetric encryption with forward search privacy," *IEEE Trans. Dependable Secure Comput.*, to be published.
[25] M. Fischlin, "Trapdoor commitment schemes and their applications," Ph.D. dissertation, Goethe Univ. Frankfurt, Frankfurt am Main, Germany, 2001.
[26] R. Nishimaki, E. Fujisaki, and K. Tanaka, "A multi-trapdoor commitment scheme from the RSA assumption," in *Proc. Australas. Conf. Inf. Secur. Privacy*, 2010, pp. 182–199.
[27] R. Gennaro, "Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks," in *Proc. Annu. Int. Cryptol. Conf.*, vol. 3152. Berlin, Germany: Springer, 2004, pp. 220–236.
[28] P. Mackenzie and K. Yang, "On simulation-sound trapdoor commitments," in *Proc. Advances in Cryptology—EUROCRYPT 2004* (Lecture Notes in Computer Science), vol. 3027. Berlin, Germany: Springer, 2003, pp. 382–400.
[29] S. Chandrasekhar and M. Singhal, "Multi-trapdoor hash functions and their applications in network security," in *Proc. IEEE Conf. Commun. Netw. Secur.*, Oct. 2014, pp. 463–471.
[30] H. Wang and S. Zhixin, "Research on zero-knowledge proof protocol," *Int. J. Comput. Sci. Issues*, vol. 10, no. 1, pp. 194–200, 2013.
[31] X. Chen, H. Sun, Y. Tobe, Z. Zhou, and X. Sun, "Coverless information hiding method based on the chinese mathematical expression," *J. Internet Technol.*, vol. 18, no. 2, pp. 91–98, 2017.
[32] Z. Qu, J. Keeney, S. Robitzsch, F. Zaman, and X. Wang, "Multilevel pattern mining architecture for automatic network monitoring in heterogeneous wireless communication networks," *China Commun.*, vol. 13, no. 7, pp. 108–116, 2016.
[33] Z. Liu, B. Li, Y. Huang, J. Li, Y. Xiang, and W. Pedrycz, "NewMCOS: Towards a practical multi-cloud oblivious storage scheme," *IEEE Trans. Knowl. Data Eng.*, to be published.

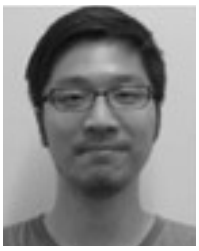


GUANGQUAN XU received the Ph.D. degree from Tianjin University, China, in 2008. He is currently a Full Professor with the Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University. He is also the Director of the Network Security Joint Laboratory and the Network Attack and Defense Joint Laboratory. He has published more than 70 articles in reputable international journals and conferences, including the IEEE IoT J, FGCS, IEEE Access, PUC, JPDC, and the IEEE Multimedia. His current research interests include cyber security and trust management. He served as a TPC Member for IEEE UIC 2018, SPNCE2019, IEEE UIC2015, and IEEE ICECCS 2014, and a Reviewer for journals, such as IEEE Access, ACM TIST, JPDC, IEEE TITS, *Soft Computing*, FGCS, and *Computational Intelligence*. He is a member of the CCF.

YAO ZHANG was born in Handan, Hebei, China. He received the B.S. degree from the Hebei University of Economics and Business. He is currently pursuing the Ph.D. degree with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His current research interests include symbolic execution and program static analysis.

LITAO JIAO received the MBA degree from the Shandong University of Science and Technology, in 2016. He is currently an Associate Professor with Qingdao Huanghai University, China. He participated in five major provincial and municipal research projects and published more than ten articles. His current research interests include HR management and information security. He received the Prize of Provincial Educational Achievement, in 2018.

EMMANOUIL PANAOUSIS received the B.Sc. degree in informatics and telecommunications from the University of Athens, Greece, in 2006, the M.Sc. degree in computer science from the Athens University of Economics and Business, Greece, in 2008, and the Ph.D. degree in mobile communications security from Kingston University London, U.K., in 2012. He was a Senior Lecturer of cyber security and privacy with the University of Brighton, an Invited Researcher with the Imperial College, a Postdoctoral Researcher with the Queen Mary University of London, and a Research and Development Consultant with Ubitech Technologies Ltd., Surrey Research Park. He is currently a Reader (Associate Professor) with the Department of Computing and Information Systems, University of Greenwich, U.K. His current research interests include cyber security, privacy, and algorithmic decision making.



KAITAI LIANG received the Ph.D. degree in computer science (applied cryptography direction) from the City University of Hong Kong, in 2014. He was a Lecturer with the School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University; a Postdoctoral Researcher with the Department of Computer Science, Aalto University, Finland; a Visiting Scholar with the Department of Computer Science, UCL; a Visiting Scholar with KU LEUVEN, Belgium, Sapienza University of Rome, Italy, and the University of Wollongong, Australia; and a Research Intern with the Institute for Infocomm Research, Singapore. He has been involved (as CI and PI) in several European funded projects, such as SPEAR, SECONDO, CUREX, and Academic of Finland. He is currently an Assistant Professor in secure systems with the University of Surrey, U.K., and a member of the Surrey Centre for Cyber Security, a GCHQ recognized UK Academic Centre of Excellence in Cyber Security Research. He has published a series of research works (over 60 publications with more than 1,000 citations), applying secure tools to tackle real-world problems in many high tier international journals, such as the IEEE TIFS, the IEEE Network, and the IEEE Transactions on Industrial Informatics. His current research interests include data security, user privacy, cybersecurity, blockchain security, and privacy-enhancing technology. He has served as a TPC for many renowned international security/privacy conferences, including ESORICS, ACNS, TRUSTCOM, and ASIACCS, and was the PC Chair of the International Workshop on Security in Big Data. He is an Official ISO Member of UK ISO Crypto Sub Committee IST/33/2, an Associate Editor of the *IET Wireless Sensor Systems*, and a Security Consultant of SEMs.

HAO WANG received the B.Eng. and Ph.D. degrees in computer science and engineering. He is currently an Associate Professor with the Norwegian University of Science and Technology, Norway. He has published more than 80 articles in reputable international journals and conferences. His current research interests include big data analytics, industrial Internet of things, high performance computing, safety-critical systems, and communication security. He is a member of the IEEE IES Technical Committee on Industrial Informatics. He served as a TPC Co-Chair for the IEEE DataCom 2015, IEEE CIT 2017, and ES 2017, and a Reviewer for journals, such as the IEEE TKDE, TII, TBD, TETC, T-IFS, the IoTJ, and ACM TOMM.

XIAOTONG LI received the B.S. degree from the School of Mechanical, Electrical, and Information Engineering, Shandong University, China, in 2018. She is currently pursuing the master's degree with the College of Intelligence and Computing, Tianjin University, China. Her current research interests include automated program repair and web application protection technique.