PAPER *Special Section on Blockchain Systems and Applications*

Multi-Party Electronic Contract Signing Protocol Based on Blockchain

Tong ZHANG[†], Yujue WANG[††], Yong DING[†,†††a)], Qianhong WU[††††], Hai LIANG[†],
and* Huiyong WANG[†††††], *Nonmembers

SUMMARY With the development of Internet technology, the demand for signing electronic contracts has been greatly increased. The electronic contract generated by the participants in an online way enjoys the same legal effect as paper contract. The fairness is the key issue in jointly signing electronic contracts by the involved participants, so that all participants can either get the same copy of the contract or nothing. Most existing solutions only focus on the fairness of electronic contract generation between two participants, where the digital signature can effectively guarantee the fairness of the exchange of electronic contracts and becomes the conventional technology in designing the contract signing protocol. In this paper, an efficient blockchain-based multi-party electronic contract signing (MECS) protocol is presented, which not only offers the fairness of electronic contract generation for multiple participants, but also allows each participant to aggregate validate the signed copy of others. Security analysis shows that the proposed MECS protocol enjoys unforgeability, non-repudiation and fairness of electronic contracts, and performance analysis demonstrates the high efficiency of our construction.

key words: *blockchain, contract signing protocol, aggregate verification*

1. Introduction

With the rapid development of information technology, today's business cooperation can sign electronic contracts through the Internet. Compared with paper contracts, electronic contracts do not need paper as the carrier, and there is no storage cost of contract documents. In addition, electronic contracts do not cost manual storage, and there is no risk of loss and damage. In the process of electronic contract signing, the transportation cost, contract express fee and long waiting time of all parties of signing the paper contract are removed, which improves the efficiency of contract management and facilitates the inquiry and review at any time. However, for the use of electronic contracts, the security and reliability of data storage has always been a problem in performing e-commerce [1]. Under the traditional centralized storage mode, even if the backup is carried out in multiple cloud spaces, the risk of data loss and tampering due to hacker attack, system failure and other reasons cannot be eliminated.

Blockchain is a technology based on consensus algorithms, which can be seen as a distributed database ledger technology. With a linked list structure of data blocks, it can maintain immutable and continuously growing data records. Therefore, when using the blockchain technology, data loss, forgery and tampering on a single node will not affect the authenticity and integrity of the data on the whole blockchain. Thus, the security and reliability of electronic contracts can be guaranteed.

Fair contract signing over the Internet is becoming more and more popular. The essence of fair contract signing protocol is that the participants exchange digital signatures of the contract fairly, which is based on the practical application of fair exchange protocol. It is a major challenge to overcome the trust frontier without the necessity of trusted third parties (TTP) in realizing fair exchange [2]. Notice that the blockchain can functionally provide a decentralized TTP. At the bottom of the blockchain is a dynamically growing block table, in which the original data and timestamp of all transactions are recorded. The timestamp provides a higher level of data recording dimension, and any node can restore the application data state at every time since the creation block.

Based on the characteristics of blockchain, such as decentralization, tamper-proof, traceability, openness and transparency, many blockchain-based electronic contract signing protocols have been proposed [3]–[5]. Ferrer-Gomila et al. [1] used blockchain technology, without the need for a TTP arbitrator, to propose a fair contract signing protocol between two parties. Draper-Gil et al. [6] and Guo et al. [7] discussed fair contract signing protocols based on blockchain in the case of multi-party signing. However, there is not yet a multi-party electronic contract signing protocol based on blockchain with higher efficiency and better security.

1.1 Our Contributions

To address the above mentioned issues, this paper proposes a multi-party electronic contract signing (MECS) protocol

Manuscript received May 20, 2021.
Manuscript revised August 2, 2021.
Manuscript publicized December 7, 2021.

[†] The authors are with the Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China.

[††] The author is with the Hangzhou Innovation Institute, Beihang University, Hangzhou, China.

[†††] The author is with the Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China.

[††††] The author is with the School of Cyber Science and Technology, Beihang University, Beijing, China.

[†††††] The author is with the School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin, China.

a) E-mail: stone_dingy@126.com

DOI: 10.1587/transinf.2021BCP0011

based on the blockchain from the legally fair contract signing technology without keystones [8]. In our MECS protocol, each contract participant needs to upload his or her own identity information, public key and signature to the blockchain, and use the aggregation verification algorithm to verify the signatures of other participants. After all participants are successful in verification, the contract initiator will upload the co-signature to the blockchain, so that others can verify whether the contract is valid. Through analysis and comparison with existing multi-party contract signing protocols, it is shown that the proposed MECS protocol is unforgeable, undeniable, fair and efficient.

1.2 Related Works

The main attribute that the contract signing protocol must achieve is the fairness of the transaction. The solutions proposed so far fall into two broad categories. One is the TTP-free protocol [9], where participants exchange signatures on the contract "bit by bit". In this case, the parties in the protocol gradually reveal their secrets over multiple rounds. It has the advantage of eliminating TTP, but it is not suitable for real-world applications because of its high communication cost, and it requires equal computing power between communication parties. The other is to introduce a TTP to achieve fairness, which can be subdivided into two types, that is, online and offline protocols.

In a contract signing protocol, a TTP can make the signing process easier and more efficient. In the online TTP protocol [10], the main problem is that the TTP becomes a system bottleneck when there are multiple participants in the system. A contract signing protocol using an offline TTP [11], [12] is more practical, where the offline TTP is only involved in disputed cases and is called an optimistic contract signing protocol. However, when TTP is employed to ensure fair trading, it is difficult for signatories to agree on a TTP that both parties trust. Zhai et al. [13] proposed an electronic contract signing protocol between two parties with a low-storage-TTP, but the existence of the TTP makes the contract signing subject to many restrictions and has trust issues. In addition, the existence of the TTP can be a bottleneck, as it can be a single point of failure or subject to external or internal attacks.

Chen et al. proposed the concept of concurrent signature [14] for the first time, which allows two entities to generate two signatures in such a way that, from any third party's point of view, the identity of the signer is not clear until the party has published additional information (keystone). Once keystone is released, both signatures will be binding on its true signers. Concurrent signature is highly efficient, requires neither a trusted arbitrator nor a high degree of interaction between parties, and does not rely on a computational balance between parties. Chen et al. proposed a concurrent signature scheme [14] based on the ring signatures [15], [16] and designated verifier signatures [17], which can realize the signature exchange without relying on a TTP. Assuming that calculating the discrete logarithm is difficult, the scheme is proved secure in the random oracle model. The scheme is an improvement based on Schnorr's signature, but the construction of the scheme is not abuse-free.

Susilo et al. further proposed a perfect concurrent signature scheme [18] to strengthen the fuzziness of concurrent signatures. That is, even if it is known that both signers have signed one of the two ambiguous signatures, no third party can infer who signed which signature. However, Wang et al. [19] pointed out that the concurrent signature scheme proposed by Susilo et al. [18] was not actually concurrent, and proposed an attack that enables the original signer to release a carefully prepared keystone. Also, an effective method resisting this attack was proposed by Wang et al. [19] so that the improved scheme was truly perfect for concurrent signature. The legal and fair contract signature scheme (without keystone) [8] proposed by Ferradi et al. can complete the exchange of signatures between two parties without keystone, which guarantees legal fairness and no abuse. However, this scheme requires multiple interactions between two parties, which poses security risks.

In recent years, the blockchain technology has received widespread attention, and its decentralized characteristic has been used to solve problems in multi-user setting. Zhang et al. [20] proposed a two-party fair contract signing scheme based on Ethernet smart contract technology, which used automatic smart contracts instead of TTP during the contract signing process. Huang et al. [3] proposed a fair three-party contract signing protocol that uses verifiable encrypted signature and blockchain to achieve fair exchange. Ferrer-Gomila et al. [1] proposed the first contract signing solution based on Bitcoin. The solution has made improvements in cost, efficiency and compliance with security requirements, but it also has limitations. The first point is that a user must have cryptocurrency from the blockchain to support the protocol. The second limitation refers to the specified time limit. In order to ensure timeliness and fairness, a time limit is specified in the contract signing protocol, but the signatories have no control over how long the miners may take to publish their transactions on the blockchain. This means they may have to restart during periods of high traffic.

2. Review of Legally Fair Signature without Keystones

In [8], Ferradi et al. put forward a contract signing paradigm based on the Schnorr signature, which does not need keystones to achieve legal fairness. As recalled in Fig. 1, this contract signing paradigm produces a co-signature binding a pair of users, and can be adapted to many signature schemes based on Discrete Logarithm Problem (DLP). The contract signing protocol is proved to be secure in the random oracle model under the assumption of the hardness of solving DLP.

3. System Model and Security Requirements

3.1 System Model

As shown in Fig. 2, a MECS system consists of three types of entities, namely, a notary office (NTO), many contract participants (COP) and blockchain.

- **NTO**: A notary office is a trusted entity for generating public parameters that need to be uploaded to the blockchain.
- **COP**: Contract participants include one contract initiator and multiple contract signatories. Contract participants can upload and download data from blockchain, and the contract initiator is responsible for uploading the contract information and co-signature to the blockchain after all the participants have signed.
- **Blockchain**: Blockchain is a decentralized distributed ledger database with unforgeability, traceability, openness, transparency, and other characteristics, and supports information verification and transmission.

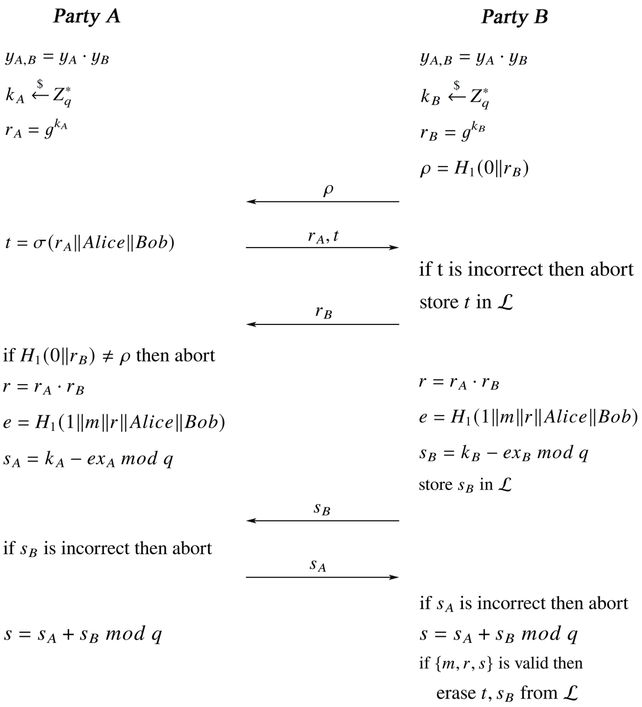
Party A
Party B
$y_{A,B} = y_A \cdot y_B$
$k_A \xleftarrow{\$} Z_q^*$
$r_A = g^{k_A}$
$y_{A,B} = y_A \cdot y_B$
$k_B \xleftarrow{\$} Z_q^*$
$r_B = g^{k_B}$
$\rho = H_1(0\|r_B)$
ρ
$t = \sigma(r_A\|Alice\|Bob)$
r_A, t
if t is incorrect then abort
store t in $\mathcal{L}$
r_B
if $H_1(0\|r_B) \neq \rho$ then abort
$r = r_A \cdot r_B$
$e = H_1(1\|m\|r\|Alice\|Bob)$
$s_A = k_A - ex_A \ mod \ q$
$r = r_A \cdot r_B$
$e = H_1(1\|m\|r\|Alice\|Bob)$
$s_B = k_B - ex_B \ mod \ q$
store s_B in $\mathcal{L}$
s_B
if s_B is incorrect then abort
s_A
if s_A is incorrect then abort
$s = s_A + s_B \ mod \ q$
$s = s_A + s_B \ mod \ q$
if $\{m, r, s\}$ is valid then
erase t, s_B from $\mathcal{L}$

Fig. 1 The legally fair signature without keystones [8].

3.2 Security Requirements

A secure MECS system needs to satisfy the following requirements.

- *Unforgeability*: The signature of the parties to the contract is not forgeable. If one party does not sign the contract after negotiation, it is impossible for other parties to forge a legal contract.
- *Non-repudiation*: During the signing of the contract, the information sent by COPs cannot be denied. Given a signed contract, each COP cannot deny having signed it. Non-repudiation is a prerequisite for fairness.
- *Fairness*: When a contract is signed by multiple parties, it shall be relatively fair for each party involved in the contract. The contract can only come into force after all the parties have signed the contract.
- *Effiency*: In the process of contract signing, the interaction between COPs can be reduced and the efficiency of signature and verification can be improved.

A correct MECS construction should satisfy the following conditions: If all participants faithfully follow the procedures, then

1) Each COP can successfully aggregate and verify the

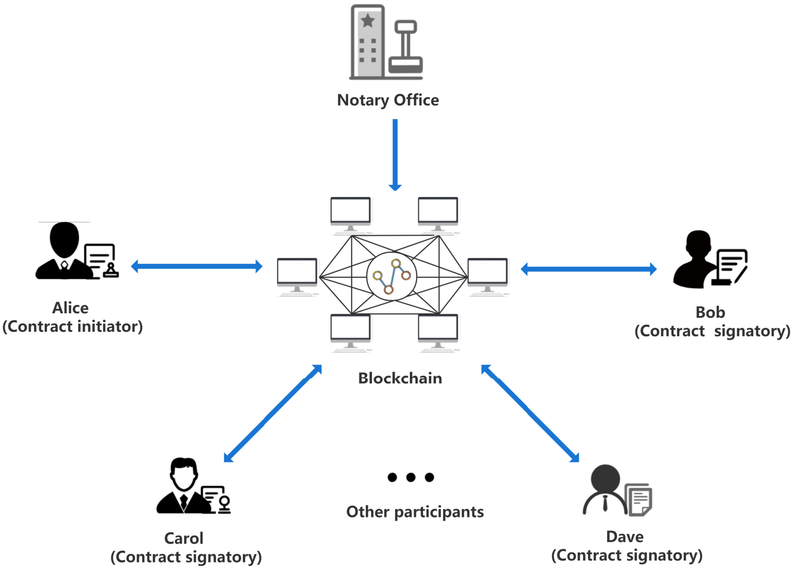

Fig. 2 System model of MECS

signatures of all other COPs;

2) Any user is able to successfully verify the validity of the co-signature on contract information uploaded by the contract initiator to the blockchain.

4. MECS Construction

This section introduces a MECS construction based on blockchain. The frequently used notations are summarized in Table 1. The security of the proposed MECS construction relies on the hardness of solving the following DLP.

Discrete Logarithm Problem. Let $G = \langle g \rangle$ be a cyclic group of prime order q. Given $h \overset{\$}{\leftarrow} G$, find a such that $h = g^a$.

- **Setup:** NTO takes the security parameter 1^γ as input and generates a cyclic group $G = \langle g \rangle$ of prime order q. NTO chooses two collision-resistant hash functions $H_1 : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ and $H_2 : \{0,1\}^* \rightarrow Z_q^*$, and then puts the public parameters (G, g, q, H_1, H_2) on the blockchain.
- **KeyGen:** Each COP u_i takes the system public parameters as input, chooses a random number $x_i \overset{\$}{\leftarrow} Z_q^*$ as the private key, and calculates
$$y_i = g^{x_i}$$
as the corresponding public key. Then he/she puts the identity information u_i and the public key y_i on the blockchain.
- **PaSignGen:** Each COP u_i takes system public parameters and identity information $(u_1, \cdots, u_n)$ as input, selects a random number $k_i \overset{\$}{\leftarrow} Z_q^*$, calculates
$$r_i = g^{k_i}$$
$$\rho_i = H_1(r_i \| u_1 \| \cdots \| u_n \| t_i)$$
where t_i is the timestamp, and writes the partial signature tuple $(r_i, \rho_i, t_i, u_1, \cdots, u_n)$ to the blockchain.
- **Signing:** COP u_i takes system public parameters, identity information $(u_1, \cdots, u_n)$, the contract information m, and his/her secret key x_i as input, downloads all other participants' partial signature tuples from the blockchain and verifies the correctness of all ρ_i. After successful verification, for the contract m to be signed, COP u_i calculates (r, e, s_i) as follows
$$r = \prod_{i=1}^{n} r_i$$
$$e = H_2(1 \| m \| r \| u_1 \| u_2 \| \cdots \| u_n)$$
$$s_i = k_i - e x_i \mod q$$
and uploads s_i to the blockchain.

Table 1 Notations

Notations	Descriptions
U	Contract participant set $U = \{u_1, u_2, \cdots, u_n\}$
u_1	Contract initiator
u_i	Contract signatory for $i = 2, \cdots, n$
γ	Security parameter
λ	A number determined by γ
$x \overset{\$}{\leftarrow} M$	Pick element x uniformly random from M
x_i, y_i	Private key and public key of contract participant u_i
m	Contract to be signed
G	Cyclic group of prime order q
B	A subset of Z_q^*
δ_i	Element in set B
ι_b	Binary length of elements in B
s	Co-signature on contract

- **AgVerify:** COP u_i takes system public parameters, the partial signature tuples, the public keys and signatures of other participants as input, randomly picks a vector $\vec{\Delta} = (\delta_1, \cdots, \delta_{i-1}, \delta_{i+1}, \cdots, \delta_n)$, where $\delta_i \overset{\$}{\leftarrow} B$ with ι_b-bit length, and B is a subset of Z_q^*. COP u_i checks the following equality
$$\prod_{j=1, j\neq i}^{n} r_j^{\delta_j} = g^{\sum_{j=1, j\neq i}^{n} s_j \delta_j} \cdot \left(\prod_{j=1, j\neq i}^{n} y_j^{\delta_j} \right)^e \tag{1}$$
If it holds, then outputs 1, which means the signatures of other COPs are valid, otherwise 0.
- **CoSign:** COP u_i takes system public parameters and the signatures of all COPs as input, calculates
$$s = \sum_{i=1}^{n} s_i \mod q$$
Then the co-signature tuple (m, r, s) is written to the blockchain by the contract initiator.
- **MuVerify:** Take the co-signature tuple (m, r, s) and public keys $y_1, y_2, \cdots, y_n$ as input, any one can validate whether it satisfies the following equality
$$g^s \cdot \left(\prod_{i=1}^{n} y_i \right)^e = r \tag{2}$$
where $e = H_2(1 \| m \| r \| u_1 \| u_2 \| \cdots \| u_n)$. If it is true, then outputs 1, otherwise 0.

Theorem 1: The proposed MECS construction is correct.

Proof 1: In order to prove the correctness of our MECS construction, it is necessary to prove that two Eqs. (1) and (2) are satisfied.

For the signatures generated by COPs, Eq. (1) holds as follows
$$g^{\sum_{j=1, j\neq i}^{n} s_j \delta_j} \cdot \left(\prod_{j=1, j\neq i}^{n} y_j^{\delta_j} \right)^e$$
$$= \prod_{j=1, j\neq i}^{n} g^{s_j \delta_j} \cdot \prod_{j=1, j\neq i}^{n} y_j^{e\delta_j}$$
$$= \prod_{j=1, j\neq i}^{n} g^{s_j \delta_j} \cdot y_j^{e\delta_j}$$

$$= \prod_{j=1,j\neq i}^{n} (g^{s_j}\cdot {y_j}^{e})^{\delta_j}$$

$$= \prod_{j=1,j\neq i}^{n} \left(g^{k_j - ex_j}\cdot g^{ex_j}\right)^{\delta_j}$$

$$= \prod_{j=1,j\neq i}^{n} \left(g^{k_j}\right)^{\delta_j} = \prod_{j=1,j\neq i}^{n} r_j^{\delta_j}$$

For the co-signature (r, s) on contract m, Eq. (2) satisfies as follows

$$g^{s}\cdot\left(\prod_{i=1}^{n} y_i\right)^{e} = g^{\sum_{i=1}^{n} s_i}\cdot\prod_{i=1}^{n} {y_i}^{e}$$

$$= \prod_{i=1}^{n} (g^{s_i}\cdot {y_i}^{e}) = \prod_{i=1}^{n} r_i = r$$

Thus, the proposed MECS construction is correct.

5. Analysis and Comparison

5.1 Security Analysis

Theorem 2: The proposed MECS construction is unforgeable under the assumption that the DLP is hard.

Proof 2: The signature in the proposed MECS construction is based on the Schnorr signature. Note that the security of Schnorr signature scheme has been formally proved under the assumption of DLP in the random oracle model [21], [22]. Thus, the proposed MECS construction also enjoys unforgeability in the random oracle model under the DLP assumption.

Theorem 3: The proposed MECS construction is undeniable.

Proof 3: Since the signed contract is publicly verifiable, after a participant sends his or her signature s_i, anyone can verify whether he or she generates a valid signature by checking whether $g^{s_i}\cdot {y_i}^{e} = r_i$ is correct. As long as the verification is passed, each COP cannot deny the message of his or her signature, thus achieving non-repudiation. Also, since the blockchain has the characteristics of openness, transparency and traceability, it can act as a distributed database to record every data transmitted on it. When a COP denies that he/she has uploaded data to the blockchain, other COPs can verify it through the blockchain. Therefore, no COPs can deny the sent information, which means the proposed construction is undeniable.

Theorem 4: The contract generation process in the proposed MECS construction is fair.

Proof 4: The blockchain can record every transaction made by honest parties very well. Malicious users cannot lie about their actions, that is, COPs cannot lie about the signatures they did not send or deny data they have sent. In the proposed MECS construction, the fairness is achieved by the co-signature, which can bind all the COPs. If the contract signer who last sent the signature does not upload his/her own signature after obtaining the signatures already uploaded to the blockchain by other COPs, that party may not use the co-signature as evidence of the other COPs' signatures to the contract. Because of the simultaneity of signatures, the fairness of MECS will not be affected if the protocol is terminated at any step. Hence, the proposed MECS construction offers fairness.

5.2 Theoretical Analysis

As shown in Table 2 and Table 3, our MECS construction is compared theoretically and functionally with existing schemes, where n is the number of signers, P is a bilinear pair operation, S is a scalar multiplication operation in G, E is a point addition operation in G, and H is a hash operation. In order to facilitate comparison, G in our MECS construction is regarded as an elliptic curve addition group.

Xiao et al.'s scheme [23] and Zhang et al.'s scheme [20] deal with the signing of an electronic contract between two parties, whereas our scheme supports multiple signatories. Zhang et al.'s scheme [20] is designed in bilinear groups, while our MECS construction is proposed in general cyclic groups. Under the same conditions, the computation overhead of bilinear pair is much higher than that of exponential operation, so the verification overhead in our scheme is lower than that of Zhang et al.'s scheme [20].

Among these solutions, only our scheme can support applications in multi-user environment and aggregate verification. All three schemes offer fairness for contract signing, where Xiao et al.'s scheme [23] relies on a third party, while Zhang et al.'s scheme [20] and our scheme do not

Table 2 Theoretical comparison

Protocol	No. of signers	KeyGen cost	Signing cost	AgVerify cost
Xiao et al.'s scheme [23]	2	S	$S + 3H$	$2S + H$
Zhang et al.'s scheme [20]	2	$2S + P$	$2S$	$3S + 2P$
Our scheme	n	S	$S + (n - 1)E + H$	$2nS + (2n - 4)E$

Table 3 Functional comparison

Protocol	Multi-user	TTP	Aggregate verification	Fairness
Xiao et al.'s scheme [23]	-	√	-	√
Zhang et al.'s scheme [20]	-	-	-	√
Our scheme	√	-	√	√

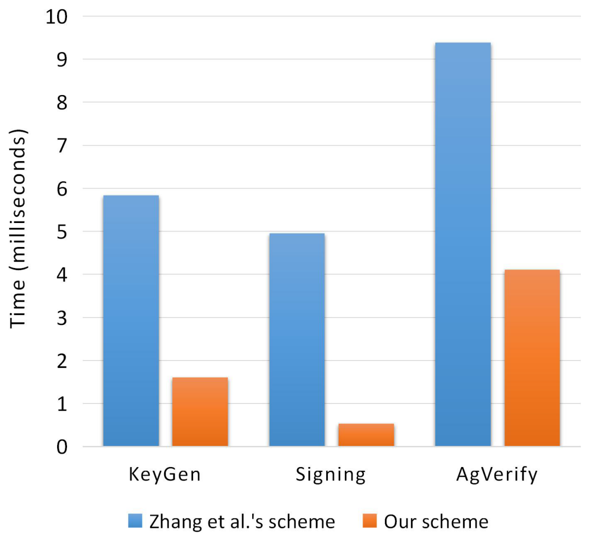

Fig. 3 Time cost of each phase.

need an additional third party. Compared with Xiao et al.'s scheme [23], the signatures of COPs in our construction are exchanged via blockchain, which reduces the interactions between COPs. Also, the information uploaded by COPs cannot be tampered with, thus the security is improved.

5.3 Performance Evaluation

We conducted the experiments of the proposed MECS construction and Zhang et al.'s scheme [20] using Go and Solidity programming languages, on a platform with Ubuntu 16.04 operating system and 4GB memory. The machine is with an AMD Ryzen 5 4600H at 3.00 GHz, and 16 GB memory. The elliptic curve is of Type A ($y^2 = x^3 + x$) such that q is a 256-bit prime and the element size in group G is 512 bits.

We use the same contract document for evaluating two schemes. Figure 3 shows the timing cost comparison of each phase of two schemes, including KeyGen phase, Signing phase and AgVerify phase. The comparison shows that the MECS scheme is more efficient than Zhang et al.'s scheme [20] in all phases, which is due to that Zhang et al.'s scheme [20] relies on the resource-intensive bilinear pairing operation.

In the AgVerify phase, the aggregate verification mechanism is used by the proposed MECS scheme, in which a COP only needs to verify once to validate the signatures of all other COPs. Figure 4 shows the time cost of aggregate verification with 10 through 100 COPs' signatures, respectively. It can be seen that the time of aggregate verification increases linearly with the number of COPs. Figure 5 shows the time cost of writing data to blockchain in five phases of Setup, KeyGen, PaSignGen, Signing and AgVerify, respectively.

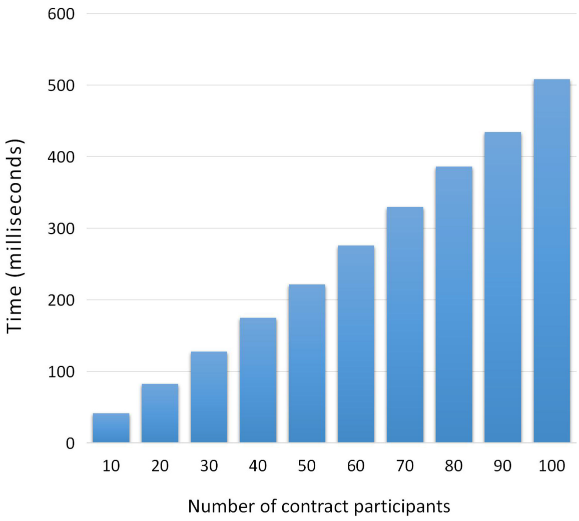

Fig. 4 Time cost of the AgVerify phase.

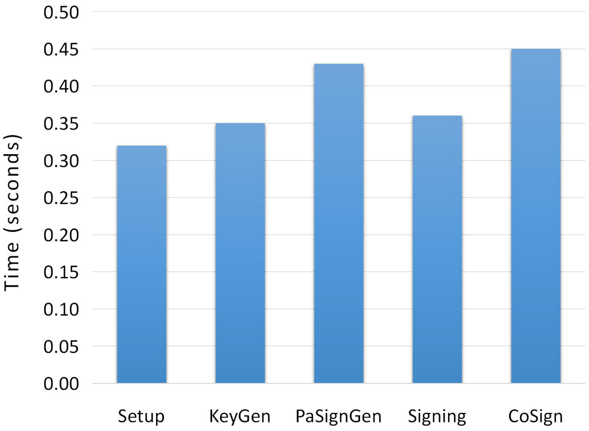

Fig. 5 Time cost of data writing to the blockchain.

6. Conclusion

To address the fairness and efficiency in achieving multi-party contract signing, this paper proposed an electronic contract signing protocol based on blockchain. The proposed MECS construction supports aggregate verification mechanism to validate the signed data from other participants. All information are exchanged via blockchain, which can reduce the amount of interactions between COPs, improve efficiency and provide security guarantee. The analysis showed that the proposed MECS construction is more efficient than existing ones.

Acknowledgements

This article is supported in part by the National Key R&D Program of China under projects 2020YFB1006004, 2020YFB1005601 and 2019QY(Y)0602, the National Natural Science Foundation of China under projects 61962012, 61932011, 61972019 and 61772538, the Guangdong

Key R&D Program under project 2020B0101090002, the Guangxi Natural Science Foundation under grants 2018GXNSFDA281054, 2019GXNSFFA245015 and 2019GXNSFGA245004, and the PCNL Major Key Project under grants PCL2021A09-4 and PCL2021A02-3.

References

[1] J.-L. Ferrer-Gomila, M. Francisca Hinarejos, and A.-P. Isern-Deyà, "A fair contract signing protocol with blockchain support," Electronic Commerce Research and Applications, vol.36, p.100869, 2019.

[2] F. Hawlitschek, B. Notheisen, and T. Teubner, "The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy," Electronic Commerce Research and Applications, vol.29, pp.50–63, 2018.

[3] H. Huang, K.-C. Li, and X. Chen, "A fair three-party contract singing protocol based on blockchain," International Symposium on Cyberspace Safety and Security, pp.72–85, Springer, 2017.

[4] H. Huang, K.-C. Li, and X. Chen, "Blockchain-based fair three-party contract signing protocol for fog computing," Concurrency and Computation: Practice and Experience, vol.31, no.22, p.e4469, 2019.

[5] J.-L. Ferrer-Gomila and M. Francisca Hinarejos, "A 2020 perspective on "a fair contract signing protocol with blockchain support"," Electronic Commerce Research and Applications, vol.42, p.100981, 2020.

[6] G. Draper-Gil, J.-L. Ferrer-Gomila, M. Francisca Hinarejos, and J. Zhou, "On the efficiency of multi-party contract signing protocols," J. Lopez and C.J. Mitchell, eds., Information Security, pp.227–243, Springer International Publishing, Cham, 2015.

[7] L. Guo, X. Li, and J. Gao, "Multi-party fair exchange protocol with smart contract on bitcoin," IJ Network Security, vol.21, no.1, pp.71–82, 2019.

[8] H. Ferradi, R. Géraud, D. Maimuţ, D. Naccache, and D. Pointcheval, "Legally fair contract signing without keystones," International Conference on Applied Cryptography and Network Security, pp.175–190, Springer, 2016.

[9] O. Goldreich, "A simple protocol for signing contracts," Advances in Cryptology, pp.133–136, Springer, 1984.

[10] 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, 1990.

[11] N. Asokan, M. Schunter, and M. Waidner, "Optimistic protocols for fair exchange," Proc. 4th ACM Conference on Computer and Communications Security, pp.7–17, 1997.

[12] H. Kılınç and A. Küpçü, "Optimally efficient multi-party fair exchange and fair secure multi-party computation," Cryptographers Track at the RSA Conference, pp.330–349, Springer, 2015.

[13] Y. Zhai, G. Xu, and Y. Zhang, "A low-storage-ttp and abuse-free contract signing protocol based on the schnorr signature," 2017 International Conference on Networking and Network Applications (NaNA), pp.279–285, IEEE, 2017.

[14] L. Chen, C. Kudla, and K.G. Paterson, "Concurrent signatures," International Conference on the Theory and Applications of Cryptographic Techniques, pp.287–305, Springer, 2004.

[15] M. Abe, M. Ohkubo, and K. Suzuki, "1-out-of-n signatures from a variety of keys," International Conference on the Theory and Application of Cryptology and Information Security, pp.415–432, Springer, 2002.

[16] R.L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," International Conference on the Theory and Application of Cryptology and Information Security, pp.552–565, Springer, 2001.

[17] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," International Conference on the Theory and Applications of Cryptographic Techniques, pp.143–154, Springer, 1996.

[18] W. Susilo, Y. Mu, and F. Zhang, "Perfect concurrent signature schemes," International Conference on Information and Communications Security, pp.14–26, Springer, 2004.

[19] G. Wang, F. Bao, and J. Zhou, "The fairness of perfect concurrent signatures," International Conference on Information and Communications Security, pp.435–451, Springer, 2006.

[20] L. Zhang, H. Zhang, J. Yu, and H. Xian, "Blockchain-based two-party fair contract signing scheme," Information Sciences, vol.535, pp.142–155, 2020.

[21] D. Pointcheval and J. Stern, "Security proofs for signature schemes," International Conference on the Theory and Applications of Cryptographic Techniques, pp.387–398, Springer, 1996.

[22] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," Journal of cryptology, vol.13, no.3, pp.361–396, 2000.

[23] H. Xiao, L. Wang, and Y. Wei, "A new fair electronic contract signing protocol," L. Barolli, H. Nishino, and H. Miwa, eds., Advances in Intelligent Networking and Collaborative Systems, pp.289–295, Springer International Publishing, Cham, 2020.

Tong Zhang received the B.S. degree in Computer Science and Technology from Nantong University, Nantong, China. She is currently pursuing the master degree in Electronic Information at the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. Her research interests include blockchain, cryptography and information security.

Yujue Wang received the Ph.D. degrees from Wuhan University, Wuhan, China, and City University of Hong Kong, Hong Kong, under the joint Ph.D. program. He is currently with the Hangzhou Innovation Institute, Beihang University, China. His research interests include applied cryptography and cloud computing security.

Yong Ding received the Ph.D. degree in cryptography from the School of Communication Engineering, Xidian University, Xi'an, China, in 2005. From 2008 to 2009, he was a Research Fellow for computer science with the City University of Hong Kong. He is now a professor at the School of Computer and Information Security of Guilin University of Electronic Science and Technology. His current research interests include cryptography and information security.

Qianhong Wu received his Ph.D. in Cryptography from Xidian University in 2004. Since then, he has been with Wollongong University (Australia) as an associate research fellow, with Wuhan University (China) as an associate professor, and with Universitat Rovira i Virgili (Spain) as a research director. He is currently a professor in Beihang University in China. His research interests include cryptography, information security and privacy, VANET security and cloud computing security. He has been a holder/co-holder of 8 China/Australia/Spain funded projects. He has authored more than 20 patents and over 120 publications in leading journals and conferences. He has served in the program committee of several international conferences in information security and privacy. He is a member of IACR and IEEE.

Hai Liang received the M.S. degree in computer science from Guilin University of Electronic Technology, China, in 2007. He is currently a lecturer at Guilin University of Electronic Technology, China. His research interests include Blockchain and Industrial Internet Security.

Huiyong Wang received his Ph.D. degree in software theory and applications from Chinese Academy of Sciences, China, in 2017. He is currently an Associate Professor at the School of Mathematics and Computing Science, Guilin University of Electronic Technology, China. His research interests include privacy-preserving computation, information security, cyber security, multi-party computation and homomorphic encryption.