Cybersecurity

RESEARCH Open Access

Privacy preserving divisible double auction with a hybridized TEE-blockchain system

Bingyu Liu, Shangyu Xie, Yuanzhou Yang, Rujia Wang and Yuan Hong*

Abstract

Double auction mechanisms have been designed to trade a variety of divisible resources (e.g., electricity, mobile data, and cloud resources) among distributed agents. In such divisible double auction, all the agents (both buyers and sellers) are expected to submit their bid profiles, and dynamically achieve the best responses. In practice, these agents may not trust each other without a market mediator. Fortunately, smart contract is extensively used to ensure digital agreement among mutually distrustful agents. The consensus protocol helps the smart contract execution on the blockchain to ensure strong integrity and availability. However, severe privacy risks would emerge in the divisible double auction since all the agents should disclose their sensitive data such as the bid profiles (i.e., bid amount and prices in different iterations) to other agents for resource allocation and such data are replicated on all the nodes in the network. Furthermore, the consensus requirements will bring a huge burden for the blockchain, which impacts the overall performance. To address these concerns, we propose a hybridized TEE-Blockchain system (system and auction mechanism co-design) to privately execute the divisible double auction. The designed hybridized system ensures privacy, honesty and high efficiency among distributed agents. The bid profiles are sealed for optimally allocating divisible resources while ensuring truthfulness with a Nash Equilibrium. Finally, we conduct experiments and empirical studies to validate the system and auction performance using two real-world applications.

Keywords: Privacy, Truthfulness, Blockchain, Smart contract, TEE, Auction design

Introduction

Divisible resources (e.g., electricity, mobile data, and computation and storage resources in the cloud) have been frequently traded or allocated in a peer-to-peer mode. All the agents can purchase or sell any amount of the resources in such markets. Since all the agents generally compete with each other to maximize their payoffs, divisible double auction mechanisms (Zou et al. 2017) are designed to allow both buyers and sellers to dynamically submit their prices until convergence (e.g., achieving the Nash Equilibrium (Maheswaran and Basar 2003; Johari and Tsitsiklis 2004)) and then complete the transaction with resource allocation. Recently, smart contracts (as decentralized and self-enforcing contracts) can be designed for distributed agents to trade divisible resources with digital agreements. The blockchain-based platform supports the execution of smart contracts for strong integrity and availability, which maintain the transparency, traceable and consensus properties.

However, severe privacy concerns may arise in both double auction (Brandt et al. 2007) and blockchain-based systems (Wüst et al. 2019). For instance, during the auction, all the agents report their bidding profiles, including sensitive data such as their bidding amount and bidding prices. As rival agents, they may want to win competitive advantages in the market (more payoffs) by reporting untruthful bids if they know the others' bid profiles. Then, the market (Krishna 2009) would be deviated. Even worse, such private data might be collected and resold (Brandt et al. 2007) to other untrusted parties.

To this end, it is desirable to propose a truthful divisible double auction mechanism while preserving all the

*Correspondence: yuan.hong@iit.edu
Illinois Institute of Technology, 10 West 35th Street, Chicago, IL 60616, USA

agents' privacy (at least sealing all the bid profiles). Specifically, smart contracts on the blockchain system can be designed for the divisible double auction. However, the blockchain system has limitations on preserving privacy for sensitive data and high performance execution. To complement the blockchain system, the Trusted Execution Environment (TEE) (Hoekstra et al. 2013) could address such limitations by executing the core functionality (e.g., computation for the smart contract) in the *enclave*, which protects the data against malicious attacks. Compared with other types of secure and private solutions (e.g., Secure Multiparty Computation (SMC) (Paillier 1999; Okamoto and Uchiyama 1998; Naccache et al. 1998)), TEE achieves stronger security and high efficiency for blockchain execution (Das et al. 2019). Thus, in this paper, we propose an efficient and privacy preserving divisible double auction with the TEE-Blockchain hybridized system (e.g., on the Intel SGX, which is a TEE supported by an architecture extension of Intel (Hoekstra et al. 2013)). Then, the hybridized system is co-designed in three aspects.

- First, the blockchain-based platform is expected to ensure integrity and availability while it interacts with other components (i.e., TEE) for the transaction, which helps data/state recovery if the execution/protocol is broken or interrupted by accidents.
- Second, the smart contract can be loaded and executed within a protected environment in Intel SGX, (namely *enclave*) (Tsai et al. 2017). All the agents' sensitive data can be protected during the computation.
- Third, we propose an efficient, individually rational and weakly budget balanced double auction based on the Progressive Second Price (PSP) (Lazar and Semret 2001) auction, derived from the Vickrey-Clarke-Groves (VCG) (Tuffin 2002) auction. The proposed divisible double auction ensures truthfulness for all the agents by achieving a Nash Equilibrium.

Furthermore, we conduct experiments for both *off-chain* procedures (executing the TEE program computation) and *on-chain* procedures (the interaction between the blockchain and TEE) in the hybridized system to evaluate the system and auction performance using two real-world applications: (1) energy trading, and (2) wireless bandwidth allocation. The remainder of the paper is organized as follows. We first present the background to briefly introduce the divisible double auction, TEE and smart contract in "Background" section. Then, "Overview of hybridized system" section gives an overview for the proposed hybridized system, and more details of the procedures. It includes how to execute the smart contract, how to trigger the TEE, and how to interact with blockchain to perform the validation. In "Auction mechanism design" section shows the designed divisible double auction mechanism with a truthfulness guarantee. In "Discussions" section analyzes the security of the system, and discusses some real-world applications, which are supported by the proposed hybridized system. We evaluate the performance of the hybridized system in "Experimental evaluations" section. Finally, "Related work" section reviews some relevant literature, and "Conclusion" section concludes the paper.

Background

Divisible double auction

In a divisible double auction, let $\mathcal{B}$ and $\mathcal{S}$ be the sets of buyers and sellers, respectively. The bidding information includes two-dimensional bid profiles, denoted as b_m for buyers and s_n for sellers. During the auction, the bid profiles are submitted as follows: (1) buyer $m \in \mathcal{B}$: $b_m = (\alpha_m, d_m)$ with bid price α_m and amount d_m to buy, and (2) seller $n \in \mathcal{S}$: $s_n = (\beta_n, h_n)$ with bid price β_n and amount h_n to sell. $b = (b_m, m \in \mathcal{B})$ denotes the bid profiles of all the buyers while $s = (s_n, n \in \mathcal{S})$ denotes the bid profiles of all the sellers. In addition, $r = (b, s)$ is defined as a strategy profile, which represents the bid profiles for all the agents. These are private information to be sealed amongst all the agents in the auction. A strategy profile without agent i is denoted as $r_{-i} = (r_1, ...r_{i-1}, r_{i+1}, ..., r_{|m+n|})$, then $r = (r_i; r_{-i})$.

From the global viewpoint, the main goal of the divisible double auction mechanism is to seek the maximum social welfare for optimal allocation. We use A_m and A_n to denote the allocation of buyer m and seller n, respectively. In the current iteration (k-th iteration) of the double auction, $A_m^{(k)}$ and $A_n^{(k)}$ represent the allocation for buyer m (*amount to purchase*) and seller n (*amount to sell*), respectively. The details for our divisible double auction machanism are given in "Auction mechanism design" section.

Trusted execution environment (TEE)

TEE provides a fully isolated environment to prevent others (e.g., software, OS, and hosts) from tampering with or learning the state of applications running in it.

Intel SGX (Costan and Devadas 2016) is an instance of TEE that enables process execution in a protected address space *enclave*. The *enclave* ensures confidentiality and integrity for the process against attacks. An *enclave* is not allowed to make system calls, but can read/write memory outside the *enclave* region. Thus, the isolated execution can be viewed as an ideal model which guarantees to be correctly executed with confidentiality. We

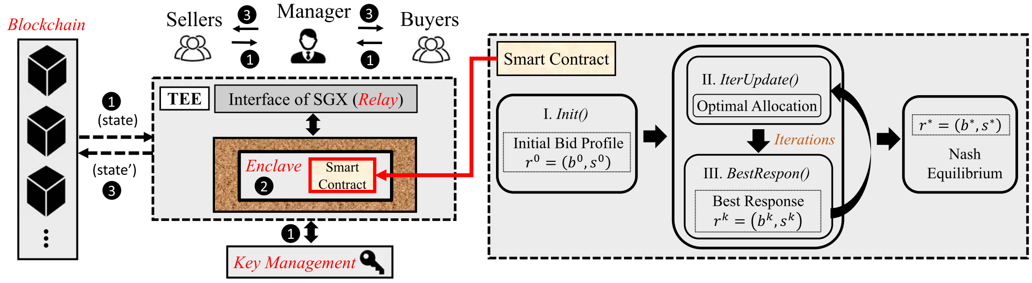


Fig. 1 Hybridized TEE-blockchain system

denote the double auction program inside the *enclave* as Prog_x.

Remote Attestation allows to remotely verify if the pieces of code or program are running within the TEE or not. In Intel SGX, CPU can measure the trusted memory, cryptographically sign the computed results, and generate the signatures for the attesting party. The private key is only known to the hardware over the program. Group signatures (EPID) (Brickell and Li 2009) are used for setting up a secure channel for *remote attestation*.

Smart contract

Cryptocurrencies are traded on the decentralized network of peers which stores all the transactions via a public ledger. Through the consensus protocol, the ledger is stored as a chain of blocks with the agreement state. Smart contract is a machinery built on top of cryptocurrencies, and it defines and executes the contract on the blockchain. In other words, the smart contracts work as a program digitally among distributed agents (Miller et al. 2000). Based on the decentralized cryptocurrencies, the integrity and availability can be guaranteed. In our work, privacy will be ensured by TEE.

Overview of hybridized system

In this section, we provide an overview of the Hybridized TEE-Blockchain System (including the procedures). Figure 1 illustrates the main components of our hybridized system: all the agents ($\mathcal{P}$), TEE ($\mathcal{T}$), *Blockchain* ($\mathcal{BC}$) and *Key Management* ($\mathcal{KM}$)).

Hybridized system architectures

- *All the agents* $\mathcal{P}$ (buyers and sellers) are the end users of the smart contract. Th manager $\mathcal{P}_M$ is the delegation to compute all the incoming private agents' input and deliver results as the administrator. $\mathcal{P}_M$ further leverages *Relay* to trigger the *enclave* to be initialized for computation (will be explained as the following). Note that the manager $\mathcal{P}_M$ is considered to be malicious, which may collude with other agents or interrupt the computation.
- TEE ($\mathcal{T}$) is responsible to run the smart contact to processes the double auction computation among the agents (requested by the manager $\mathcal{P}_M$) in the *enclave* $\mathcal{E}$, which protects the privacy and integrity of computations. It also generates remote attestations (computation correctness) for state updates. To further improve the functionality and security of our system, we design the only interface component *Relay* $\mathcal{R}$ to provide indirect access to *enclave*. *Relay* can also provide the message passing with the *Blockchain*.
- *Blockchain* ($\mathcal{BC}$) maintains a distributed append-only ledger via running a consensus protocol. The state of $\mathcal{BC}$ and attestations are stored on the chain. Moreover, the validity of state update are checked by the blockchain with the TEE attestations.
- *Key Management* ($\mathcal{KM}$) generates keys for both private agents' inputs and state encryption. All the agents and TEE can directly interact with the $\mathcal{KM}$ for the key pairs via a key distribution protocol.

Enclave functionality model

Enclave ($\mathcal{E}$) protects the code of program and data during the computation for the auction. Specifically, the program running inside the *enclave* is completely isolated from an adversarial OS as well as other processes on the host. We formalize and integrate the Intel SGX (Shi et al. 2015) as TEE in our hybridized sytem.

In order to model the ideal functionality channel with some proprieties such as privacy and authenticity, we utilize a global universal composability (UC) framework functionality (Canetti 2001) to instantiate the SGX Functions. More formally, we denote the program X which runs inside the SGX enclave as Prog_x, which can be Prog_{da} for double auction. The SGX function can be expressed as $\mathcal{F}_{SGX}(\sum_{sgx})[\mathsf{Prog}_\mathsf{x}, \mathcal{R}]$, where $\sum_{sgx}$ is a group signature scheme and $\mathcal{R}$ is *Relay*. As shown in Fig. 2, the program Prog_x is loaded into *enclave* via the

```
SGX Functionality Model: $\mathcal{F}_{SGX}[\mathsf{Prog_x}, \mathcal{R}]$
$Initialize$ :
    Upon receiving ($\mathsf{Init}$) from $\mathcal{R}$ :
        Set outp := $\mathsf{Prog_x}.initalize()$
            // model EPID signature
        $\psi_{att} := \sum_{sgx} \cdot \mathsf{Sig}(\mathsf{sk}_{sgx}, (\mathsf{Prog_x}, \mathsf{outp}))$
        Output $(\mathsf{outp}, \psi_{att})$
$Resume$ :
    Upon receiving $\mathsf{Authen}(\mathsf{tx})$ from $\mathcal{R}$ :
        Set outp := $\mathsf{Prog_x}.resume()$
        Output $\mathsf{outp}$
```

Fig. 2 Enclave functionality model

"init" call from *Relay*. When *Relay* calls "resume", the program is executed based on the incoming requests or inputs, denoted as inp, and computes the output with an attestation $\psi_{att} := \sum_{sgx} \cdot \mathsf{Sig}(\mathsf{sk}_{sgx}, (\mathsf{Prog_x}, \mathsf{outp}))$. The signature under TEE hardware key sk_{sgx} and pk_{sgx} could be obtained from the SGX Functions ($\mathcal{F}_{SGX}$).

Procedures

In this section, we now sketch the procedures for the execution of divisible double auction with the smart contract in the hybridized system (more details are given in Fig. 3). It depicts that the designed system is executed with three phases: (1) Initialization, (2) ExecProg, and (3) Finalization. We denote the input and output for the TEE as inp and outp, respectively. Also, regarding the deposit, we use $\widetilde{\xi_{b_m}}$, $\widetilde{\xi_{s_n}}$ and $\widetilde{\xi_{\mathcal{P}_M}}$ for all buyers, sellers and managers.

(1) Initialization. Prior to the auction phase, all the agents (buyers and sellers) are supposed to prepare for their deposits $\widetilde{\xi_{b_m}}$, $\widetilde{\xi_{s_n}}$. Besides, the manager also needs to deposit $\widetilde{\xi_{\mathcal{P}_M}}$ (if the manager or any agent is identified to deviate the computation, then the deposit will be charged as penalty). Then the TEE will set $\mathsf{state} := init$ as confirming that the deposits in the blockchain. Otherwise, the TEE will set $\mathsf{state} := abort$ for preparing next auction and refund the deposit to the agents. For the auction computation, the TEE will fetch the key pair $(\mathsf{pk_{sgx}}, \mathsf{sk_{sgx}})$ from *Key Management* for attestation, where the key $(\mathsf{pk_{sgx}})$ is bundled to the executing prog_x instance (auction) for checking the correctness of computation. Besides, the attestation with current state $[\mathsf{state}, \psi_{att}]$ are posted on the blockchain $\mathcal{BC}$ (as described in "Enclave functionality model" section).

```
Execution Procedure $(\mathcal{P}, \mathcal{BC}, \mathcal{T})$
$InitRequest()$ :
    deposit $\widetilde{\xi_{b_m}}, \widetilde{\xi_{s_n}}$ and $\widetilde{\xi_{\mathcal{P}_M}}$
    invoke procedure $initEnclave()$ via $request$
        // generate key pairs for inputs encryption
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow\$ \mathsf{KGen}(1^n)$
    fetch the key pairs from$\mathcal{KM}$
    publish pk to all agents for $\mathsf{Enc_{pk}}(\mathsf{inp})$
    call $Initagent()$
$InitEnclave()$ :
    $\mathsf{receive}(\mathsf{init}(), request)$ to load $\mathsf{Prog_x}$ inside $E$
    boost $enclave$ with $\mathsf{Prog}_x.\mathsf{initialize}()$
    distribute $(\mathsf{pk}, \psi_{sgx})$ for attestation
$InitAgent()$ :
    $\mathsf{tx} := (\mathsf{Enc_{pk}}(\mathsf{inp}), l_{id}, \widetilde{\xi_{b_m}}, \widetilde{\xi_{s_n}})$ are sent to $\mathcal{P}_M$
    if receive $\mathsf{Authen}(\mathsf{tx})$ then
        $\mathcal{BC}.post(\mathsf{state}, \psi_{att}, l_{id})$
    else
        set state := fail
        refund the deposits $\widetilde{\xi_{b_m}}, \widetilde{\xi_{s_n}}$ and $\widetilde{\xi_{\mathcal{P}_M}}$
$ExecProg()$ :
    boost $enclave$ with $\mathsf{resume}()$ to load and run $\mathsf{Prog}_x$
    retrieve and decrypt the previous state from $\mathcal{BC}$
    decrypt encrypted inputs$(\mathsf{Enc_{pk}}(\mathsf{inp}), l_{id})$
    load and compute $\mathsf{Prog}_x$
    outp := $TEE(\mathsf{Prog_x})$
$Finalization()$ :
    // After generating the outputs, verify the correctness of output
    Vf the correctness of outp with $\psi_{att}$
    $\psi_{sgx} := \sum_{sgx} \cdot \mathsf{Sig}(\mathsf{sk}_{sgx}, (\mathsf{Prog_x}, \mathsf{outp}))$
    $\mathcal{BC}.post(\mathsf{state}', \psi_{att}, l_{id})$
    delivery outp to all agents $\mathcal{P}$
```

Fig. 3 Hybridized system procedure

Next, to tackle the large inputs of agents, the manager $\mathcal{P}_M$ will handle $\mathsf{tx} := [\mathsf{Enc_{pk}}(\mathsf{inp}), l_{id}, \widetilde{\xi_{b_m}}, \widetilde{\xi_{s_n}}]$ from all the agents where inp denotes the inputs of all the agents, and l_{id} represents a unique identifier (ID). Then, $\mathcal{P}_M$ will send tx to the *Relay* for executing the auction computation. Note that all the agents send the transactions through secure communication channels among all the agents and TEE. The tx is a transaction to deliver the input and output data among different system components.

(2) ExecProg. To execute the auction requested from $\mathcal{P}_M$, the *Relay* will retrieve the state information from the *blockchain* and *Relay* will trigger TEE to execute the requested service (auction) with the "resume" call if the state can be verified. Then TEE first decrypts the

input data (from the *Manager*) with the private key *sk* obtained from the *Key Management* and launch the auction smart contract code as $\mathsf{Prog_x}$ in the *enclave* (a sandboxed environment). Thus, an adversary cannot interrupt the execution or monitor data inside the *enclave* considering the natural merit of *enclave*. The final results output of the program (auction smart contract) $\mathsf{Prog_x}$ will be securely returned to the manager.

(3) $\mathsf{Finalization}$. Once *manager* receives the final result outp from TEE and check the correctness with the *Blockchain*. If the result outp is accepted by the *Blockchain* by checking and verifying the new state state', the auction result $(\mathsf{outp}, \psi_{sgx}, l_{id}, \widetilde{\xi_{b_m}}, \widetilde{\xi_{s_n}})$ will be delivered to all the agents via *Manager* and *Blockchain* will store the new state'.

Threat model and properties

To ensure data privacy and integrity for the auction computation, we use the TEE's attestation (Yuan et al. 2018), where the computation is executed inside the *enclave* trusted by all the agents. However, the remaining software stack outside the enclave and the hardware are not trusted. The adversary may corrupt any number of agents, assuming that honest agents will trust their own codes and platform (leakage resulted from its software bugs are out of the scope). Furthermore, we assume that all the agents do not trust each other in the auction while being potentially malicious, such as stealing the bid profiles information. During the execution, each agent may send, drop, modify and record arbitrary transactions. Note that the side-channel attacks against *enclave* and DoS attacks are not considered in this paper.

In our proposed hybridized TEE-Blockchain system, the TEE compensates for the privacy issue with respect to the smart contract, i.e., our system can address the privacy issue for the double auction by utilizing the TEE for isolating the contract (auction process) execution inside the *enclave*, shielding it from potential malicious agents. From the system aspect, the following properties are addressed:

- **Correctness**. The correctness of computation in the TEE can be guaranteed and verified by the remote attestation based on the given state and inputs.
- **Privacy** and **Security**. Our system protect and verify the sensitive inputs (e.g., bid profiles) and outputs of all the agents.

Auction mechanism design

Problem formulation

We represent the strategy of each agent with a non-negative valuation function $\widehat{V}_m(\cdot)$ for buyers, which indicates the willingness to pay, or value for buyers to obtain the amount of divisible item. Similarly, we have negative cost function $\widehat{C}_n(\cdot)$ for sellers. In the auction design, we adopt generic assumptions (Lazar and Semret 2001; Tuffin 2002) for the valuation function $\widehat{V}_m(\cdot)$: (1) $\widehat{V}_m$ is differentiable, concave and $\widehat{V}_m(\emptyset) = 0$, and (2) $\widehat{V}'_m(\cdot)$ is non-increasing and continuous; for the cost function $\widehat{C}_n(\cdot)$: (1) $\widehat{C}_n$ is differentiable, convex and $\widehat{C}_n(\emptyset) = 0$, and (2) $\widehat{C}'_n(\cdot)$ is increasing and continuous;

In our settings, buyers have diminishing marginal utility while sellers have increasing marginal cost. This indicates that $\widehat{V}_m(A^k_m) > \widehat{V}_m(A^{k+1}_m)$ $(\forall m \in \mathcal{B})$ where $A^k_m < A^{k+1}_m$ while $\widehat{C}_n(A^k_n) < \widehat{C}_n(A^{k+1}_n)$ $(\forall n \in \mathcal{S})$ where $A^k_n < A^{k+1}_n$.

Assuming that each agent is selfish with the goal to maximize their own payoff. Therefore, they may untruthfully modify their bids in the auction. With the blockchain-based system to realize the smart contract for the auction, untruthful responses could be detected, and thus penalty will be applied to the cheating agent.

Thus, valuation function will be converted to $\widehat{V}_m(\cdot) - \mu_p(\cdot)$ where $\mu_p(\cdot)$ is a anti-monotonic function for measuring the penalty applied to the cheated amount for the buyers (Li and Marden 2014). Note that $\mu_p(0) = 0$ means if the valuation is submitted and penalty will be exempted. Similarly, the cost function will be updated as $\widehat{C}_n(y_n) + \mu_p(\cdot)$ where $\mu_p(\cdot)$ is a monotonic function (and increasing derivative) for measuring the penalty applied to the sellers (Li and Marden 2014) and $\mu_p(0) = 0$ (exempting the penalty for truthful response of the sellers).

Then, the payoff functions are defined for buyer m and seller n as $f_m(r)$ and $f_n(r)$, to represent their payoffs w.r.t. the bid profiles of all the agents r. Specifically, ρ_m is the payment made by buyer m while ρ_n is the payment received by seller n. Moreover, $\rho(r_i, r_{-i})$ is defined as the difference between all the buyers' aggregated valuation if any buyer i is absent in the auction minus the aggregated valuation if i is included the auction (Lazar and Semret 2001; Zou et al. 2017; Kojima and Yamashita 2017). Similarly, $\rho(r_j, r_{-j})$ is defined as the difference between all the sellers' aggregated cost if any seller j is absent minus the aggregated cost if j is included. Thus, we have:

$$
\begin{aligned}
\rho(r_i, r_{-i}) &= \sum_{m \neq i} \alpha_m [A_m(0; r_{-i}) - A_m(r_i; r_{-i})] \\
\rho(r_j, r_{-j}) &= \sum_{n \neq j} \beta_n [A_n(0; r_{-j}) - A_n(r_j; r_{-j})]
\end{aligned}
\tag{1}
$$

Then, given the optimal allocation profile for buyer $m \in \mathcal{B}$ and seller $n \in \mathcal{S}$ as A^*_m and A^*_n, we can define the payoffs for the buyer m and seller n as:

$$\begin{aligned} f_m(r) &= \widehat{V}_m(A^*_m) - \rho(r_i, r_{-i}), \forall m \in \mathcal{B} \\ f_n(r) &= \rho(r_j, r_{-j}) - \widehat{C}_n(A^*_n), \forall n \in \mathcal{S} \end{aligned} \qquad (2)$$

Definition 1 (*Individual Rationality*) The divisible double auction mechanism achieves individual rationality if the following holds: $f_m(r) \geq 0$ and $f_n(r) \geq 0$.

It ensures that the all the agents obtain non-negative payoff while participating in the auction mechanism.

Definition 2 (*Incentive Compatibility*) The divisible double auction mechanism achieves incentive compatibility if the following holds: $f_m(r) \geq f_m(\overline{r})$ and $f_n(r) \geq f_n(\overline{r})$ where r and $\overline{r}$ are denoted as the true bid profile and false bid profile.

It ensures that all the agents in the auction will obtain the maximum payoff if they report the truthful bid.

Definition 3 (*Weak Budget Balance*) In the divisible double auction, for $\forall\ m \in B$ and $\forall\ n \in S$, if there exists: $\sum_{\forall m \in B}(\alpha_m \cdot d_m) \geq \sum_{\forall n \in S}(\beta_n \cdot h_n)$, then the auction mechanism satisfies weak budget balance.

It ensures "no budget deficit" in the auction.

Definition 4 (*Clearing Price*) The price θ is defined as the *clearing price* for an optimal allocation $A^*(\cdot)$, if there exists a feasible and efficient allocation, such that, the best response is achieved for the maximum social welfare, denoted as $F(\cdot) = \sum_{m \in \mathcal{B}} \widehat{V}_m(A_m) - \sum_{n \in \mathcal{S}} \widehat{C}_n(A_n)$.

We say that the clearing price θ (Brero et al. 2019) supports the optimal allocation $A^*(\cdot)$ with the maximum social welfare.

Definition 5 (*Nash Equilibrium*) In the divisible double auction, Nash Equilibrium holds if given the bid profile r^* such that:

$$\begin{aligned} f_m(b^*_m, r^*_{-m}) &\geq f_m(b_m, r^*_{-m}), \forall m \in \mathcal{B} \\ f_n(s^*_n, r^*_{-n}) &\geq f_n(s_n, r^*_{-n}), \forall n \in \mathcal{S} \end{aligned} \qquad (3)$$

where $r_{-m} = \{r\} \setminus \{b_m\}$ is a bid profile for all the buyers excluding buyer m from $\mathcal{B}$ and $r_{-n} = \{r\} \setminus \{s_n\}$ is a bid profile for all the sellers except seller n from $\mathcal{S}$.

Our divisible double auction mechanism will find the optimal allocation for all the agents to achieve the maximum social welfare. Moreover, the truthfulness of bids will be ensured in the smart contract via *individual rationality* and *incentive compatibility*. To preserve privacy, all the agents' bid prices and amounts (bid profiles), as well as the valuation/cost functions can be protected in the auction. The clearing price and trading amount will only be disclosed to every pair of potential sellers/buyers at the end of the auction (after convergence).

```
Double Auction Mechanism $\mathrm{Prog}_{da}(\mathcal{B}, \mathcal{S}, r)$
Initialize :$\forall m \in \mathcal{B}, \forall n \in \mathcal{S}, r = (b, s)$
Require :$(\alpha_i)_{\max} \geq (\beta_j)_{\min}$ and $C < \min\{\sum_{i \in B} d_i, \sum_{j \in S} h_j\}$
set iteration $k := 1$
while true do
    $A^*_m(b, C) := \min\{d_m, \{[C - \sum_{i \in \mathcal{B}_m(b)} d_i], 0\}_{\max}\}$
    $A^*_n(s, C) := \min\{h_n, \{[C - \sum_{i \in \mathcal{S}_n(s)} h_j], 0\}_{\max}\}$
    $Q(r, C) := \min\{\sum_{i \in \mathcal{B}} A^*_i(r, C), \sum_{j \in \mathcal{S}} A^*_j(r, C)\}$
    $\widehat{\mathcal{P}} := \frac{p_b(r, C) - p_s(r, C)}{\omega_{\max} + \sigma_{\max}}$
    $\widetilde{C}(r, C) := Q(r, C) + \widehat{\mathcal{P}}$
    $b^*_m = \arg\max\{f_m(b_m, b_{-m})\},\ m \in \mathcal{B}$
    $s^*_n = \arg\max\{f_n(s_n, s_{-n})\},\ n \in \mathcal{S}$
    repeatuntil convergence
    set iteration $k := k + 1$
endwhile
return$b^*_m(\forall m \in \mathcal{B}), s^*_n(\forall n \in \mathcal{S})$
```

Fig. 4 Divisible double auction

Divisible double auction mechanism

We now design the divisible double auction mechanism (DA), which will be executed as a smart contract inside the TEE. The procedures are detailed as below:

(1) Initialization. Denoting the double auction program as Prog$_{\mathrm{da}}$, while executing Prog$_{\mathrm{da}}$ in the *enclave*, the decrypted bid profiles of all the agents will be checked if they satisfy the initial condition (i.e, $(\alpha_i)_{\max} \geq (\beta_j)_{\min}$). Otherwise, the state of auction will be turned from "active" into "fail". Then, it requires all the agents to update their bid profiles. Meanwhile, the potential amount of the resources C should be smaller than the overall demand/supply.[1] The auction will be active if and only if satisfying the above conditions (Fig. 4).

(2) Iteration. Once the iteration starts, the potential amount $\widehat{C}(r, C)$ is updated as below:

[1] The potential amount is used for computing and updating the allocation buyers and sellers in each iteration).

$$\widetilde{C}(r, C) = Q(r, C) + \frac{p_b(r, C) - p_s(r, C)}{\omega_{\max} + \sigma_{\max}} \tag{4}$$

where $Q(r,\ c) = \min\{\sum_{m \in \mathcal{B}} A_m^*,\ \sum_{n \in \mathcal{S}} A_n^*\}$, $p_b(r, C) = \min\ \{\alpha_i, A_i \geq 0\}$, $p_s(r, C) = \max\{\beta_j, A_j \geq 0\}$ and $\widehat{\mathcal{P}} = \frac{p_b(r,C) - p_s(r,C)}{\omega_{\max} + \sigma_{\max}}$.

We denote $Q(r,\ C)$ as the minimum value of total demand and total supply; A coefficient $\widehat{\mathcal{P}}$ is used for gradients of marginal valuations or costs; Two variables $p_b(r, C)$ and $p_s(r, C)$ are defined to stimulate the much faster coverage in each iteration with the updated potential amount. We use $\omega_{\max}$ and $\sigma_{\max}$ to denote the upper bound for buyers' valuations ($\omega_{\max} \geq \max \sup_{A_m} \{|\frac{\partial \widehat{V}_m(A_m)}{\partial A_m}|\}$) and the upper bound for sellers' costs ($\sigma_{\max} \geq \max \sup_{A_n} \{|\frac{\partial \widehat{C}_n(A_n)}{\partial A_n}|\}$). Note that the valuation function $\widehat{V}_m(A_m)$ and cost function $\widehat{C}_n(A_n)$ are updated with the penalty functions in the smart contract. The potential amount is expected to achieve a Nash Equilibrium quickly with the gradients of marginal valuations and costs.

The optimal allocation A_m^* and A_n^* are updated in each iteration, and agent derive their best responses. Given $(r,\ C)$, the optimal allocations (for buyers/sellers) are

$$\begin{aligned} A_m^* &= \min\{d_m, \{[C - \sum_{i \in B_m(b)} d_i], 0\}_{\max}\} \\ A_n^* &= \min\{h_n, \{0, [C - \sum_{j \in \mathcal{S}_n(s)} h_j]\}_{\max}\} \end{aligned} \tag{5}$$

where d_m and h_n are the updated amount for buyer m to purchase and for seller n to sell, respectively; $B_m(b) = \{i \in \mathcal{B} | \alpha_i > \alpha_m\} \cup \{\alpha_i = \alpha_n \text{ and } i < m\}$ and $S_n(s) = \{j \in \mathcal{S} | \beta_j > \beta_n\} \cup \{\beta_i = \beta_m \text{ and } j < n\}$.

The updated potential amount $\widetilde{C}(r, C)$ can be iteratively derived based on the given potential amount C.

(3) Best Response. We use b_m^* and s_n^* to represent the best response of buyer $m \in B$ and seller $n \in S$. With the bid profiles $r = (b, s)$ and a pair of potential amounts $(C, \widehat{C})$, the best response can be derived as below:

$$\begin{aligned} b_m^*(r, C, \widehat{C}) &= \arg\max\{f_m(b_m, b_{-m})\} \\ s_n^*(r, C, \widehat{C}) &= \arg\max\{f_n(s_n, s_{-n})\} \end{aligned} \tag{6}$$

In the divisible double auction program $\mathsf{Prog}_{\mathsf{da}}$, the best response will be computed in each iteration and finally converge to a Nash Equilibrium. Notice that, in different applications (e.g., different divisible resources), the valuation and cost functions would be different. In this dynamic auction game, all the agents recompute their best response to the current strategies (bid profiles) of other agents.

Theorem 1 *The divisible double auction (as program* $\mathsf{Prog}_{\mathsf{da}}$*) achieves individual rationality and incentive compatibility.*

Proof

First, suppose that the truthful bid profile provided by buyer $m \in B$, then we could obtain the non-negative payoff function $f_m(r) = \widehat{V}_m(A_m^*) - \rho(r_i, r_{-i})$. Correspondingly, given the truthful bid profile provided by seller $n \in S$, $f_n(r) = \rho(r_j, r_{-j}) - \widehat{C}_n(A_n^*)$, $\forall n \in \mathcal{S}$. Thus, the truthful bid profiles show that the non-negative payoffs are guaranteed for all the agents in the auction (individual rationality is proven).

Second, we define the A_m and A_n as allocation of buyer $m \in \mathcal{B}$ and seller $n \in \mathcal{S}$, separately. And A_m^k and A_n^k represent the allocation for k-iteration. We will verify the incentive compatibility for all buyers $m \in \mathcal{B}$ first for incentive compatibility. Suppose there is truthful bid profile $b_m = (\alpha_m, d_m^k)$ where $\alpha_m = \frac{\partial \widehat{V}_m(d_m^k)}{\partial d_m^k}$, which can make $f_m(b_m^k, r_{-m}) \geq f_m(b_m, r_{-m}), \forall m \in \mathcal{B}$, there are two cases involved: Case (A): assuming that $\alpha_m < \frac{\partial \widehat{V}_m(d_m)}{\partial d_m}$, if there is bid b_m^k, which makes $d_m^k = A_m \leq d_m$. Then, we could have $\alpha_m^k \geq \frac{\partial \widehat{V}_m(d_m)}{\partial d_m} > \alpha_m$, due to the diminishing marginal utility of the valuation function. Thus, we have $f_m(b_m^k, r_{-m}) \geq f_m(b_m, r_{-m}), \forall m \in \mathcal{B}$, since we have obtained the maximum social welfare; Case (B): suppose that $\alpha_m > \frac{\partial \widehat{V}_m(d_m)}{\partial d_m}$. If there is bid b_m^k, then we have $d_m^k = \frac{\partial \widehat{V}_m(d_m^k)}{\partial d_m^k} = d_m$, then we get $\alpha_m > \frac{\partial \widehat{V}_m(d_m)}{\partial d_m} = \alpha_m^k$. It is known that $A_m^k \leq A_m$ for the maximum social welfare. If $A_m^k = A_m$, then we get $f_m(b_m^k, r_{-m}) = f_m(b_m, r_{-m}), \forall m \in \mathcal{B}$. If we have $A_m^k < A_m$, the below holds:

$$\begin{aligned} & f_m(b_m, r_{-m}) - f_m(b_m^k, r_{-m}) \\ & = \widehat{V}_m(A_m) - \widehat{V}_m(A_m^k) + \rho(A_m^k, r_{-m}) - \rho(A_m, r_{-m}) \\ & \leq \alpha_m^k(A_m - A_m^k) + F(r) - \alpha_m A_m - F(r^k) + \alpha_m^k A_m^k \\ & \leq \alpha_m^k(A_m - A_m^k) - \alpha_m^k(A_m - A_m^k) = 0 \end{aligned} \tag{7}$$

Thus, we could have $f_m(b_m^k, r_{-m}) \geq f_m(b_m, r_{-m}), \forall m \in \mathcal{B}$ with Case (A) and (B). Similar, incentive compatibility can be proven for all the sellers $\forall n \in \mathcal{S}$. □

Discussions

In this section, we analyze the security of the proposed hybridized TEE-Blockchain system and illustrate some real-world applications supported by the system.

Security

Based on the key feature of isolation in *enclave*, Intel SGX enables the program (data) to be executed inside the secure container (*enclave*) for confidentiality and integrity. The adversary cannot interrupt the computation executed in a sandboxed environment (*enclave*). Note that *enclave* is created in its virtual address space by an untrusted hosting application with OS support. Once *enclave* starts initialization, data and codes inside it will be isolated from the rest of the system. Note that the encrypted data are sent from agents to *enclave* through secure channels. However, other malicious servers cannot eavesdrop on the encrypted data and even tamper with the communication.

During the execution, if any agents abort/skip this step or behave dishonestly during the initialization, the execution will be terminated and refunds to the honest agents within the time threshold T_1. Afterwards the computation starts, all agents send the encrypted inputs to the interface of SGX. In this phase, if no malicious behaviors are detected by the manager, the Relay $\mathcal{R}$ will forward the encrypted inputs to the *enclave* $\mathcal{E}$. However, it is hard to determine if the agents behave dishonest (i.e., fail to send message) or the Relay behave malicious (i.e, dropping message) during the execution if the *enclave* $\mathcal{E}$ does not receive any incoming requests. Thus, all agents $\mathcal{P}$ and Relay $\mathcal{R}$ both receive the challenge request (we denote as $\mathsf{request}_{chal}$). Within the certain time threshold T_2, if agents response with inputs and procedure will move to the next steps. Otherwise, the agents are proved to be malicious. Similarly, if the Relay $\mathcal{R}$ is proven to be the malicious one, the protocol is terminate and set up state is fail. In terms of the last phase Finalization, the TEE return the final results to all the agents and publish the states on the blockchain. Note that during all the data flow, the deposits of malicious agents are not refunded for punishment.

Real-world applications

In practice, divisible resources which could be privately traded using our system, e.g., electricity (Wang et al. 2014), cloud resources (Jin et al. 2018; Fujiwara et al. 2010), and wireless spectrum (Kebriaei et al. 2016). We now discuss two of them as representative applications. Note that different valuation/cost functions will be defined and implemented in different applications.

Energy Trading. There is demand from power grid for trading the excessive locally generated energy, e.g., the renewable energy resources (Aliabadi et al. 2017; Faqiry and Das 2016). The proposed hybridized system is able to implement the privacy preserving divisible double auction for energy trading, due to the divisible of electricity resource. The valuation/cost functions are defined as $\widehat{V}_m(x_m) = \zeta_m \log(x_m + 1)$ and $\widehat{C}_n(y_n) = a_n y_n^2 + b_n y_n$ (Bompard et al. 2007), where ζ_m is a parameter leveraged by the behavior preference of buyer. The parameter of a_n and b_n are used for leveraging how much the sellers incline to sell. The valuation/cost functions follow the general assumption illustrated in "Auction mechanism design" section. Eventually, hybridized system only generate the *clearing price* for the auction to all the agents. The energy amount of each pair of buyer and seller will only obtain the amount traded between them.

Wireless Bandwidth Allocation. We can model the wireless bandwidth allocation (Feng et al. 2015; Zhang et al. 2016b) based on our proposed hybridized system for network traffic and services. In terms of a MVNO (Mobile Virtual Network Operator), the valuation function for buyer m is defined as $\widehat{V}_m(x_m) = \zeta_m \ln(x_m)$ where ζ_m defined as a positive-valued parameter. This indicates that buyers willing to pay for the bandwidth. Meanwhile, the cost function of seller n, an InP (Infrastructure Provider) is denoted as $\widehat{C}_n(y_n) = \alpha_n e^{y_n}$, where y_n presents the bandwidth it can supply and α_n as another positive-valued parameter (bandwidth) for the seller n. As expected, the valuation/cost functions are also follow the general assumptions, and execute privately and truthfully via hybridized system for such divisible double auction.

Experimental evaluations

In this section, the system performance of both *off-chain* and *on-chain* procedures in the hybridized system will be evaluated in the following.

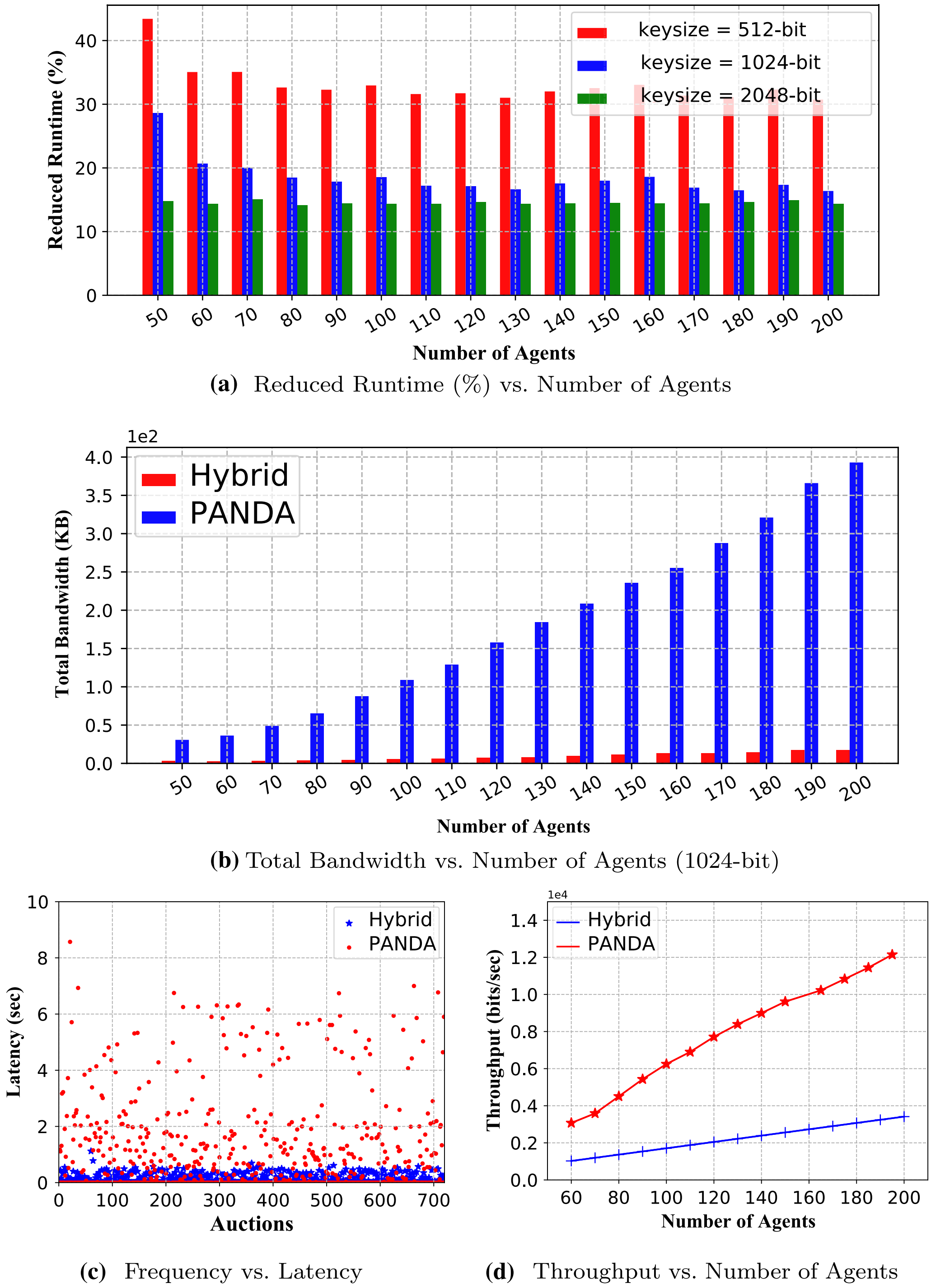
Setting. To support the smart contacts execution within the *enclaves*, we use Graphene[2] on the Microsoft Azure.[3] A *manifest* is adopted to support the *enclave* initialization, and it protects the smart contract execution in the host process. For the *on-chain* implementation, we use the Hyperledger Fabric[4], which is designed for distributed ledger technologies with multiple modules for the blockchain platforms. As a distributed ledger platform, it includes a highly modular and configurable architecture, which supports the smart contract execution. Note that the Hyperledger Fabric is deployed on the VM with Ubuntu 18.04 Standard (2 vcpus, 8 GiB memory) on the Microsoft Azure for the *on-chain* procedures.

Applications. We conduct experimental evaluations for two case studies: (1) energy trading/auction (Aliabadi et al. 2017), and (2) wireless bandwidth allocation (Feng et al. 2015; Zhang et al. 2016b) among up to 200 agents.

[2] Graphene Tsai et al. (2014) is a lightweight guest OS, which replaces the Intel SDK for the *enclave* and host process.

[3] https://azure.microsoft.com/en-us/solutions/confidential-compute/

[4] https://hyperledger-fabric.readthedocs.io/en/latest/index.html



(a) Reduced Runtime (%) vs. Number of Agents

(b) Total Bandwidth vs. Number of Agents (1024-bit)

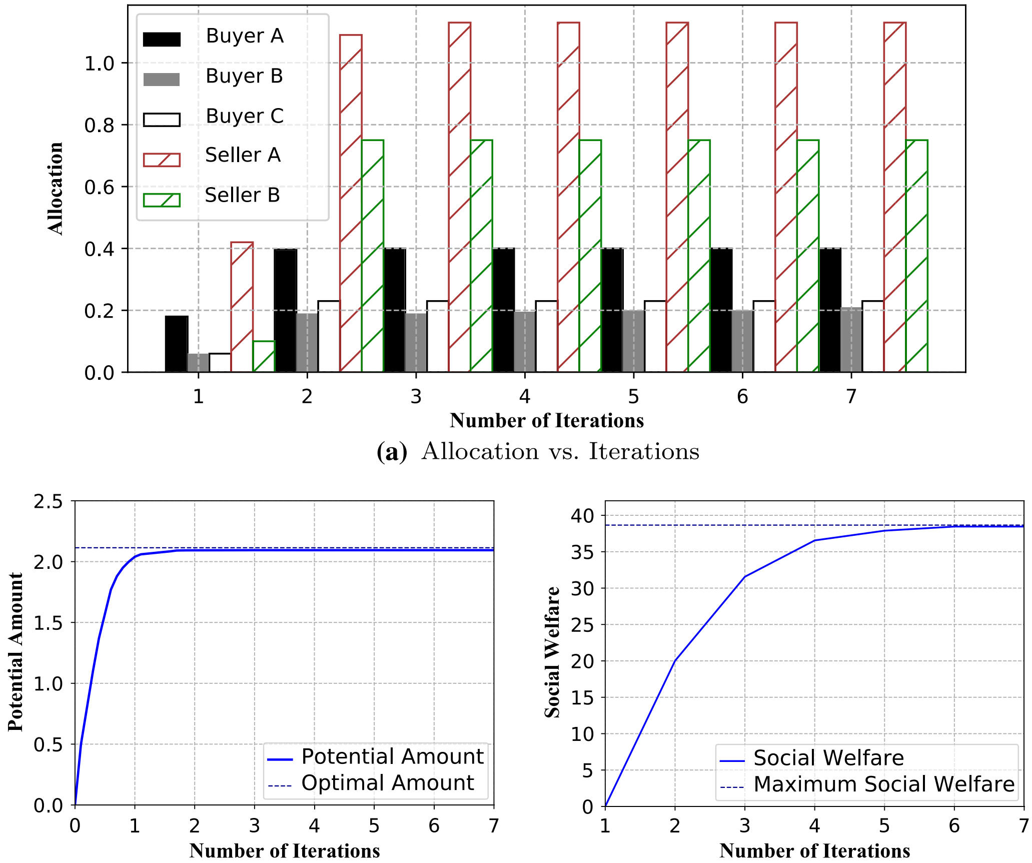
(c) Frequency vs. Latency

(d) Throughput vs. Number of Agents

Fig. 5 *Off-chain* System Performance Evaluation

Each agent can be either a buyer or seller in the auction for both applications.

In the experiments of energy trading/auction, we utilize the valuation function $\widehat{V}_m(x_m) = \zeta_m \log(x_m + 1)$ and cost function $\widehat{C}_n(y_n)=a_n y_n^2 + b_n y_n$, as detailed in Zou et al. (2017). We adopt the same parameters $\zeta_m = 50$, $a_n = 30$ and $b_n = 0$ as Zou et al. (2017). Similarly, wireless bandwidth allocation is implemented with the valuation function $\widehat{V}_m(x_m) = \zeta_m \ln(x_m)$ and cost function $\widehat{C}_n(y_n) = \alpha_n e^{y_n}$, where $\zeta_m = 50$ and $a_n = 2$ Feng



(a) Allocation vs. Iterations

(b) Potential Amount vs. Iterations

(c) Social Welfare vs. Iterations

Fig. 6 Case study (I): energy trading: *off-chain* double auction computation (20 agents)

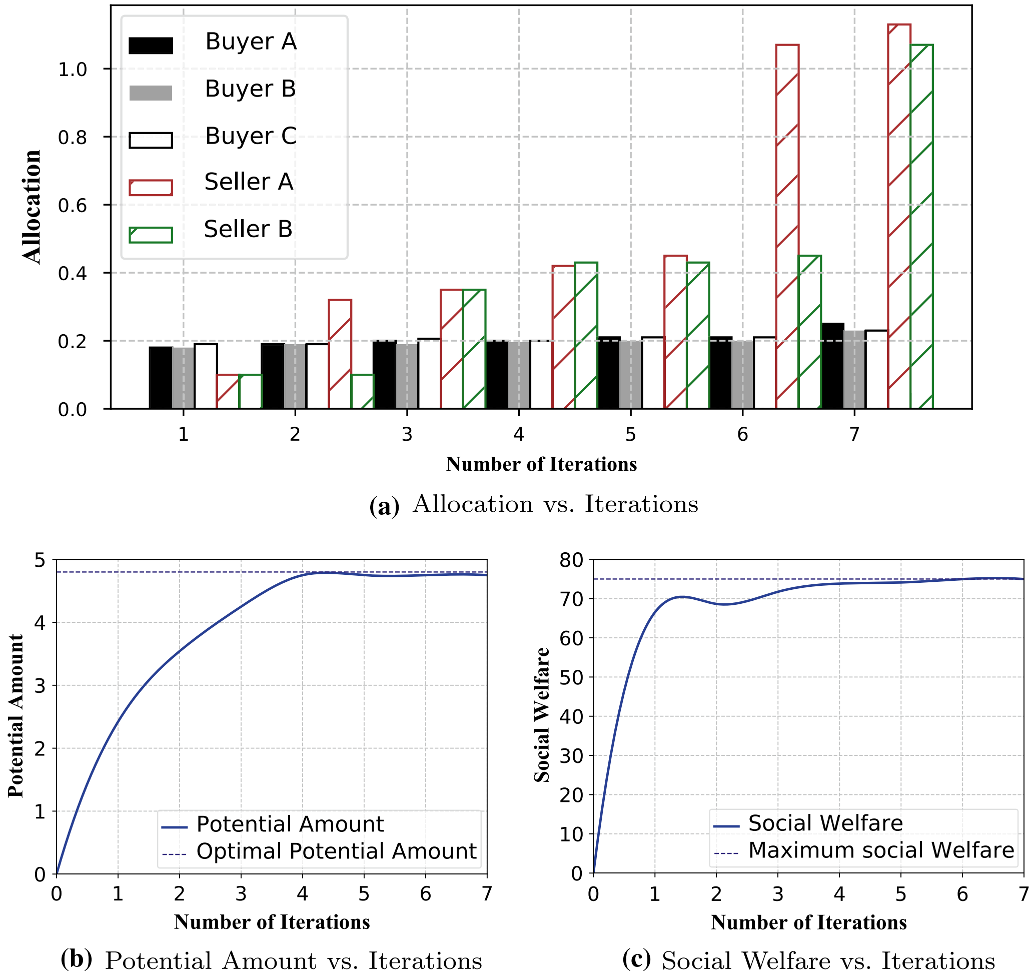
et al. (2015). For the energy trading, real datasets from the UMASS Trace Repository (Barker et al. 2012) are adopted while synthetic datasets are generated per (Feng et al. 2015; Zhang et al. 2016b) for the wireless bandwidth allocation.

Off-chain evaluation

Performance Evaluation We first evaluate system performance for securely perform computation for the auction (the off-chain computation for the optimal allocation). Figure 5a presents the percentage of total reduced runtime for the off-chain computation by comparing the our hybridized TEE-blockchain system ("Hybrid") with the cryptographic protocol based double auction system (Liu et al. 2020) ("PANDA"). Note that this evaluation is performed by the varying the number of agents (from 50 to 200) with the different key sizes (from 512-bit to 2048-bit). As shown in Fig. 5a, compared with "PANDA", the runtime of our ("Hybrid") has been significantly reduced for all different key sizes. The hybridized system ("Hybrid") shows the higher efficiency and scalability by reducing more than 15 % runtime (on average) with strong security guarantees (in case of the 2048-bit key size), 18 % average runtime for 1024-bit key size and 35% average runtime for 512-bit key size.

Furthermore, Fig. 5b illustrates the comparison between the total bandwidth for ("PANDA") and ("Hybrid") during the auction. The "PANDA" composes cryptographic primitives for secure computation, which results in heavy burdens for computation. With the TEE-blockchain system, the bandwidth of "Hybrid" has been drastically reduced to make the communications more efficient.

In addition, Fig. 5c presents the latency of 720 different auctions. The latency of our hybridized system is less than 1 s for most auctions, which is also significantly lower than the cryptographic protocols (PANDA). It indicates that the real-time performance of divisible double auction can be achieved via the hybridized TEE-blockchain system. Finally, Fig. 5d illustrates the throughput (bits/sec) of the system on a varying number



(a) Allocation vs. Iterations

(b) Potential Amount vs. Iterations

(c) Social Welfare vs. Iterations

Fig. 7 Case study (II): wireless bandwidth allocation: *off-chain* double auction computation (20 agents))

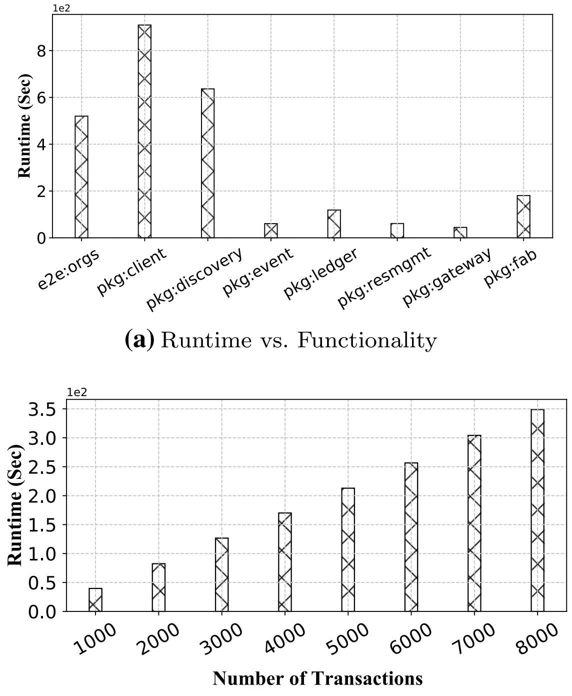
of agents (1024-bit key). Essentially, throughput measures the amount of data transmitted during a specified time period via a network, interface, or channel. In this case, we use the throughput to measure the average size of the encrypted data that are transmitted among different agents per second. It is defined as throughput (bits/sec) = (the average data size of all the communication channels)/(the average time). Note that the average time includes the agents' local computational time. As shown in Fig. 5d, the throughput of "Hybrid" increases slower than "PANDA" as the number of agents increases.

Case study. We also conduct empirical studies for two example applications (energy trading and wireless bandwidth allocation) in case of 20 agents, including 12 buyers and 8 sellers. Figures 6 and 7 demonstrate the detailed results on (1) allocation ($A^*_m(b, C)$ and $A^*_{n=1}(s, C)$), (2) potential amount (C), and (3) social welfare ($F(\cdot)$) in each iteration until achieving the Nash Equilibrium, for both applications.

First, Figs. 6a and 7a show the allocation for five randomly picked agents (three buyers and two sellers) in different iterations. The allocation of both buyers and sellers increase and finally achieve the optimally allocated amount after multiple iterations. Second, in Figs. 6b and 7b, the potential amount of the auction (used for updating the allocation for buyers and sellers in each iteration) grows until convergence while moving to new iterations. Finally, the social welfare ($F(\cdot)$) is derived based on equation $F(\cdot) = \sum_{m\in\mathcal{B}} \widehat{V}_m(A_m) - \sum_{n\in\mathcal{S}} \widehat{C}_n(A_n)$. Figure 6c presents an increasing trend in multiple iterations and the social welfare of energy trading converges to the maximum value \$38 while the social welfare of the wireless bandwidth allocation in Fig. 7c converges to the maximum value \$75.

On-chain performance evaluation

Besides the off-chain computation, we demonstrate the performance of on-chain transactions. Figure 8a shows the runtime for different package functionalities. The functionality of the `pkg:client` takes 909.22 s (most of the on-chain runtime), the processes of functionality include preparing/creating channel and client context, and communicating with the Fabric network via the channel. Compare with the `e2e:orgs`, the `pkg:discovery` takes relatively longer time (636.58 s). It is implemented on the `DiscoveryFilterService` package and the discovery service with filter is returned. Also, `pkg:ledger` takes 118.77 s while `pkg:fab` takes around 180.67 s. Furthermore, `pkg:event` (60.63 s) works for users to receive events such as block, filtered block, contracts, and transaction status events. The `pkg: resmgmt` (61.11 s) enables the creation and update of resources on a Fabric network, and it also allows the administrators to create/update channels, query peer for channels, and perform some operations, i.e., installing, instantiating and upgrading the smart contracts. Finally, `pkg: gateway` (44.49 s) enables users to update the application based on Hyperledger Fabric programming model. Finally, Fig. 8b presents the on-chain runtime on a varying number of transactions. The runtime is only up to 350 s in case of 8000 transactions.



(a) Runtime vs. Functionality

(b) Runtime vs. Transaction

Fig. 8 *On-Chain* performance evaluation

Related work

There were some other auction mechanisms for allocating divisible resources, i.e., spectrum allocation (Wu et al. 2011; Dong et al. 2012). Combinatorial auctions (Dong et al. 2012) was discussed for cognitive radio networks. Strategy-proof mechanism for multi-radio spectrum buyers was proposed by Wu and Vaidya (Wu et al. 2011). A sealed-bid reserve auction was modeled for the radio spectrum allocation problem. Hoefer et al. (Hoefer et al. 2014) investigated the combinatorial auctions with a conflict graph via an approximation algorithm (LP formulation). Other studies related to divisible resources auctions focused on the revenue maximization(Jia et al. 2009) or the efficiency of social maximization(Dong et al. 2012; Gopinathan and Li 2010).

The privacy concerns in auction mechanism for divisible resources have been raised in Chen et al. (2014); Huang et al. (2015). In Suzuki and Yokoo (2003) cryptographic techniques were proposed for achieving the privacy and security in the auction game. A cryptographic scheme for one-side auctions was proposed in Huang Huang et al. (2013). In addition, Cheng et al. (2019) presented the complementary characters for blockchain and TEE, the rigorous security proofs are provided to support the confidentiality of the hybrid system. Also, the Hawk system (Kosba et al. 2016) was designed as a decentralized smart contract framework for running the contracts off-chain while posting zero-knowledge proofs on-chain. Zhang et al. (2016a) proposed a system Town Crier that authenticates data feed using smart contracts supported by the Ethereum platform. It enables data fetching from existing HTTP-enabled data sources, and utilizes TEE to execute its core functionality and protect its data against malicious attackers.

In the context of double auction, a recent scheme was proposed to protect privacy for the bids (Liu et al. 2020). However, it requires a heavy computation burden by composing the cryptographic primitives. Instead, ETA (Liu et al. 2021a) was proposed for an efficient and private system, which securely executes double auction for allocating divisible resources among distributed agents within the Intel SGX. However, TEE cannot guarantee the availability (as the host can terminate TEE). We extend the ETA system to the Hybridized TEE-Blockchain System (Liu et al. 2021b), which enables smart contract execution on the blockchain to ensure strong integrity and availability with high efficiency. Therefore, the proposed hybridized system can securely and efficiently perform secure computation for the double auction.

Conclusion

In this paper, we design a hybridized TEE-Blockchain system to securely execute divisible double auction among distributed agents within the *enclave* in a highly efficient way. Meanwhile, it interacts with the blockchain for validation and storage. The proposed divisible double auction mechanism guarantees *individual rationality*, *incentive compatibility*, *weak budget balance* and *pareto efficiency*. The input private data of all the agents in the divisible double auction can also be protected in the hybridized system. The experimental results have demonstrated both effectiveness and efficiency for the designed hybridized system to privately compute the optimal allocation and execute the divisible double auction.

Acknowledgements
This work is partially supported by the National Science Foundation (NSF) under Grants No. CNS-1745894, CNS-2034870, and CNS-2046335. The authors would like to thank the anonymous reviewers for their constructive comments.

Author contributions
All authors read and approved the final manuscript.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 5 August 2021 Accepted: 18 October 2021
Published online: 01 December 2021

References

Aliabadi DE, Kaya M, Şahin G (2017) An agent-based simulation of power generation company behavior in electricity markets under different market-clearing mechanisms. Energy Policy 100:191–205

Barker S, Mishra A, Irwin D, Shenoy P, Albrecht J (2012) SmartCap: flattening peak electricity demand in smart homes. PerCom

Bompard E, Ma Y, Napoli R, Abrate G (2007) The demand elasticity impacts on the strategic bidding behavior of the electricity producers. IEEE Trans Power Syst 22:188–197

Brandt F, Sandholm T, Shoham Y (2007) Spiteful bidding in sealed-bid auctions. In: Proceedings of IJCAI, pp1207–1214

Brero G, Lahaie S, Seuken S (2019) Fast iterative combinatorial auctions via Bayesian learning. In: AAAI, pp 1820–1828

Brickell E, Li J (2009) Enhanced privacy ID from bilinear pairing. IACR Cryptol. ePrint Arch. 95

Canetti R (2001) Universally composable security: a new paradigm for cryptographic protocols. In: FOCS, pp 136–145

Cheng R, Zhang F, Kos J, He W, Hynes N, Johnson NM, Juels A, Miller A, Song D (2019) Ekiden: a platform for confidentiality-preserving, trustworthy, and performant smart contracts. In: IEEE EuroS&P, pp 185–200

Chen Z, Huang L, Li L, Yang W, Miao H, Tian M, Wang F (2014) PS-TRUST: provably secure solution for truthful double spectrum auctions. In: INFOCOM, pp 1249–1257

Costan V, Devadas S (2016) Intel SGX explained. IACR Cryptol 2016:86

Das P, Eckey L, Frassetto T, Gens D, Hostáková K, Jauernig P, Faust S, Sadeghi A (2019) Fastkitten: practical smart contracts on bitcoin. In: USENIX security symposium, pp 801–818

Dong M, Sun G, Wang X, Zhang Q (2012) Combinatorial auction with time-frequency flexibility in cognitive radio networks. In: Proceedings of the INFOCOM, pp 2282–2290

Faqiry MN, Das S (2016) Double-sided energy auction in microgrid: equilibrium under price anticipation. IEEE Access 4:3794–3805

Feng Z, Qiu C, Feng Z, Wei Z, Li W, Zhang P (2015) An effective approach to 5g: wireless network virtualization. IEEE Commun Mag 53(12):53–59

Fujiwara I, Aida K, Ono I (2010) Applying double-sided combinational auctions to resource allocation in cloud computing. In: Tenth annual international symposium on SAINT, Proceedings, pp 7–14

Gopinathan A, Li Z (2010) Strategyproof wireless spectrum auctions with interference. In: Proceedings of the GLOBECOM

Hoefer M, Kesselheim T, Vöcking B (2014) Approximation algorithms for secondary spectrum auctions. ACM Trans Internet Technol 14(2–3):16–11624

Hoekstra M, Lal R, Pappachan P, Phegade V, del Cuvillo J (2013) Using innovative instructions to create trustworthy software solutions. In: HASP@ISCA, p 11

Huang Q, Tao Y, Wu, F (2013) SPRING: a strategy-proof and privacy preserving spectrum auction mechanism. In: Proceedings of the INFOCOM, pp 827–835

Huang H, Li X, Sun Y, Xu H, Huang L (2015) PPS: privacy-preserving strategy proof social-efficient spectrum auction mechanisms. IEEE Trans PAR Distrib Syst 26:1393–1404

Jia J, Zhang Q, Zhang Q, Liu M (2009) Revenue generation for truthful spectrum auction in dynamic spectrum access. In: Proceedings of the ACM MobiHoc, pp 3–12

Jin A, Song W, Zhuang W (2018) Auction-based resource allocation for sharing cloudlets in mobile cloud computing. IEEE Trans Emerg Top Comput 6(1):45–57

Johari R, Tsitsiklis JN (2004) Efficiency loss in a network resource allocation game. Math Oper Res 29(3):407–435

Kebriaei H, Maham B, Niyato D (2016) Double-sided bandwidth-auction game for cognitive device-to-device communication in cellular networks. IEEE Trans Veh Technol 65:7476–7487

Kojima F, Yamashita T (2017) Double auction with interdependent values: incentives and efficiency. Theor Econ 12(3):1393–1438

Kosba AE, Miller A, Shi E, Wen Z, Papamanthou C (2016) Hawk: the blockchain model of cryptography and privacy-preserving smart contracts. In: IEEE S&P, pp 839–858

Krishna V (2009) Auction theory. Academic Press, London

Lazar AA, Semret N (2001) Design and analysis of the progressive second price auction for network bandwidth sharing. Telecommun Syst 13

Li N, Marden JR (2014) Decoupling coupled constraints through utility design. IEEE Trans Autom Control 59:2289–2294

Liu B, Xie S, Hong Y (2020) PANDA: privacy-aware double auction for divisible resources without a mediator. In: AAMAS, pp 1904–1906

Liu B, Xie S, Hong Y (2021) Efficient and private divisi-ble double auction in trusted execution environment. In: EAI International Conference on Applied Cryptography in Computer and Communications. Springer, pp 75–92

Liu B, Yang Y, Wang R, Hong Y (2021) Poster: privacy preservingdivisible double auction with a hybridized TEE-Blockchain system. In: 41st IEEE International Conference onDistributedComputing Systems, ICDCS 2021, Washington DC, USA, July 7-10, 2021. IEEE, pp 1144–1145

Maheswaran RT, Basar T (2003) Nash equilibrium and decentralized negotiation in auctioning divisible resources. Group Decis Negot 12(5):361–395

Miller MS, Morningstar C, Frantz B (2000) Capability-based financial instruments. In: Proceedings FC 1962, pp 349–378

Naccache, D., Stern, J (1998) A new public key cryptosystem based on higher residues. In: Proceeedings of the ACM CCS, pp 59–66

Okamoto T, Uchiyama S (1998) A new public-key cryptosystem as secure as factoring. In: Advances in cryptology—EUROCRYPT '98, IACR, Proceedings, pp 308–318

Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology—EUROCRYPT '99, IACR, Proceedings, pp 223–238

Shi E, Zhang F, Pass R, Devadas S, Song D, Liu C (2015) Trusted hardware: life, the composable universe, and everything. Manuscript

Suzuki K, Yokoo M (2003) Secure generalized Vickrey auction using homomorphic encryption. In: FC, pp 239–249 (2003)

Tsai C, Arora KS, Bandi N, Jain B, Jannen W, John J, Kalodner HA, Kulkarni V, de Oliveira DAS, Porter DE (2014) Cooperation and security isolation of library oses for multi-process applications. In: EuroSys, pp 9–1914

Tsai C, Porter DE, Vij M (2017) Graphene-sgx: a practical library OS for unmodified applications on SGX. In: USENIX ATC, pp 645–658

Tuffin B (2002) Revisited progressive second price auction for charging telecommunication networks. Telecommun Syst 20:255–263
Wang Y, Saad W, Han Z, Poor HV, Basar T (2014) A game-theoretic approach to energy trading in the smart grid. IEEE Trans Smart Grid 5(3):1439–1450
Wüst K, Diana L, Kostiainen K, Karame G, Matetic S, Capkun S (2019) Bitcontracts: adding expressive smart contracts to legacy cryptocurrencies. IACR Cryptol 2019:857
Wu F, Vaidya NH (2011) SMALL: a strategy-proof mechanism for radio spectrum allocation. In: INFOCOM, joint conference of the IEEE COMSOC, pp 81–85
Yuan R, Xia Y, Chen H, Zang B, Xie J (2018) Shadoweth: private smart contract on public blockchain. J Comput Sci Technol 33(3):542–556
Zhang F, Cecchetti E, Croman K, Juels A, Shi E (2016a) Town crier: an authenticated data feed for smart contracts. In: ACM CCS, pp 270–282
Zhang D, Chang Z, Yu FR, Chen X, Hämäläinen T (2016b) A double auction mechanism for virtual resource allocation in sdn-based cellular network. In: IEEE (PIMRC), pp 1–6
Zou S, Ma Z, Liu X (2017) Resource allocation game under double-sided auction mechanism: efficiency and convergence. IEEE Trans Autom Control 63(5):1273–1287

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.