ACCEPTED FROM OPEN CALL

SPB: A Secure Private Blockchain-based Solution for Distributed Energy Trading

Ali Dorri, Fengji Luo, Salil S. Kanhere, Raja Jurdak, and Zhao Yang Dong

Blockchain is increasingly being used to provide a distributed, secure, trusted, and private framework for energy trading in smart grids. However, existing solutions suffer from a lack of privacy, processing and packet overheads, and reliance on TTP to secure the trade. To address these challenges, the authors propose an SPB framework. SPB enables the energy producers and consumers to directly negotiate the energy price.

Abstract

Blockchain is increasingly being used to provide a distributed, secure, trusted, and private framework for energy trading in smart grids. However, existing solutions suffer from a lack of privacy, processing and packet overheads, and reliance on trusted third party (TTP) to secure the trade. To address these challenges, we propose a secure private blockchain (SPB) framework. SPB enables energy producers and consumers to directly negotiate the energy price. To reduce the associated overheads, we propose a routing method which routes packets based on the destination public key (PK). SPB eliminates the reliance on TTP to ensure both energy producer and consumer commit to their obligations by introducing atomic meta-transactions. The latter consists of two transactions: first the consumer generates a CTP transaction, committing to pay the energy price to the producer. On receipt of the energy, the smart meter of the consumer generates an energy receipt confirmation (ERC) which triggers a smart contract to transfer the committed price in CTP to the energy producer. To verify that the ERC is generated by a genuine smart meter, SPB supports authentication of anonymous smart meters to prevent malicious nodes from linking ERC transactions and thus enhance the user privacy. Qualitative security analysis shows the resilience of SPB against a range of attacks. Implementation results demonstrate that SPB reduces monetary cost and delay compared to existing solutions.

Introduction

Human society is facing the grand challenges of climate change and ever-increasing energy demand. These challenges require us to re-shape the operation patterns of our energy generation, transmission, and consumption patterns. Electrical power systems hence need to be adapted to operate in a more efficient and sustainable manner, for example, accommodate more renewable energy. With this background, the concept of "smart grid" was proposed in the early 21st century [1], setting up the strategic goal to develop next-generation power systems.

The power distribution network which is central to smart grids is significantly characterized by high penetration of distributed renewable resources, flexible loads, and advanced sensing infrastructures. The transformation from a centralized to distributed energy generation pattern has led to the emergence of energy prosumers (producers-and-consumers), who are capable of generating and consuming energy simultaneously, for example, a building equipped with solar panels. This naturally raises the need for establishing an energy trading mechanism that is secure, maintains participant privacy, and fosters energy economics.

Recently blockchain [2] has attracted tremendous attention as a means to provide a distributed, secure, and anonymous framework for energy trading. Blockchain employs changeable public keys (PKs) to identify users, thus providing a level of anonymity. Mihaylov *et al*. [3] propose to convert energy to a virtual currency known as NRGcoin which is then traded in a blockchain. A centralized distribution system operator (DSO) monitors the demand and load by collecting information from the smart meters of producers and consumers and defines the NRGcoin price accordingly. *Energy blockchain* proposed by the authors in [4] introduces a new credit-based payment scheme that reduces the associated delay with energy trading payments as a central trusted entity, known as credit bank, transfers and manages energy coins between user accounts. The bank provides energy coins to participants according to their credit. In [5] the participants in the energy market store energy bids and requests in a central repository. To ensure the privacy of the energy producers, a mixing service is employed that gives a random ID to the produced energy by each smart meter in a particular time frame. Powerledger [6] proposes a blockchain-based energy market that requires users to buy powerledger tokens to be able to trade energy. However, the following issues have yet to be addressed by the state-of-the-art:

Lack of Privacy

All transactions of a user are publicly available; thus, critical information about the user such as energy consumption or production patterns can be obtained by linking multiple identities to the user or by examining the pattern of transactions in the same ledger. In most existing works [4, 5], the transactions generated by each energy prosumer can be tracked as they are either generated using the same PK or are linked together in the same ledger.

Reliance on Trusted Third Party (TTP) Brokers

Trading energy requires both sides to fulfil their commitments in the trade, for example, the producer must send the consumer the energy upon receipt of payment. Achieving this level of trust

Digital Object Identifier: 10.1109/MCOM.2019.1800577

Ali Dorri, Salil S. Kanhere, and Zhao Yang Dong are with UNSW Sydney; Fengji Luo is with The University of Sydney; Raja Jurdak is with CSIRO.

is challenging due to the distributed nature of the blockchain. To address this challenge most existing methods rely on a TTP [3–6], which partially centralizes trust in the network. However, the TTP is susceptible to typical issues arising from centralization including a single point of failure, bottlenecks, and so on. Additionally, the privacy of the participating entities may be compromised by the TTP as it has a complete view of all actions performed in the network. In [7] the authors proposed a new concept known as *atomic swap* which enables exchanging different cryptocurrencies in multiple blockchains without requiring exchange servers (TTP). Instead, both the buyer and seller pay coin (in their respective blockchain network) to a smart contract. The contract transfers the exchanged coins to the buyer and seller's account. However, this method incurs processing overhead and delay as four transactions must be mined, that is, stored, in the blockchain for one exchange. A similar method is used in [8] for trading goods. For each trade, the seller creates a smart contract. The buyer pays the price of the goods to the contract. Once the buyer confirms the receipt of goods, the contract pays the seller. Although this method enables trading goods without a TTP, each trade requires three transactions. This incurs processing and packet overheads and increases delay, thus it might not be scalable for large-scale smart grid networks.

Blockchain Overhead

Despite its benefits, blockchain consumes a significant amount of computational, energy, and bandwidth resources, as appending (i.e., mining) a new block often involves executing resource consuming consensus algorithms, and all communications are broadcast, which increases overheads for direct negotiation between nodes. Ethereum [9] employs the Whisper routing protocol that enables direct communications between multiple parties by adding destination and source fields to the transactions [10]. The transactions are broadcast and only the node whose ID matches with the destination ID accepts the transaction.

This article proposes a Secure Private Blockchain-based platform (SPB) to address the aforementioned challenges. SPB can be used by pure consumers, pure producers, and prosumers. The key novel features of SPB are:

- An anonymous routing method overlaid on top of the blockchain for enabling energy price negotiation between the producer and consumer.
- A purely distributed trading method that introduces the notion of atomic meta-transactions.
- A private authentication method to verify smart meters.

SPB uses a new private routing algorithm that enables direct messaging in blockchain. SPB runs over a public blockchain, where anyone can join and participate, and does not require any participating node to buy assets to trade energy. SPB ensures that all involved participants in the energy trade commit to their obligations and reduces the associated processing overhead and delay using "atomic meta-transactions," which are an adaptation of the concept of atomic swaps. In an atomic meta-transaction, a constituent transaction is considered to be valid if and only if it is coupled with at least one other transaction. Once a price is agreed, the consumer generates a commit to pay (CTP) transaction, committing to pay a specific amount of money to the producer. The generation of CTP places a hold on the committed money, so that the consumer can no longer pay this amount to any other node. However, the money is not yet transferred to the producer account until the producer has transferred energy to the consumer. Once energy is transferred, the consumer's smart meter confirms receipt of energy by generating an energy receipt confirmation (ERC) transaction. The atomic meta-transaction, that consists of CTP and ERC is then mined in the blockchain and the committed money is paid out to the energy producer. The blockchain participants must be able to ascertain that an ERC transaction is signed by a genuine smart meter. Additionally, the ERC must remain unlinked to the previous ERC transactions to prevent a malicious node from tracking the energy consumption or generation pattern of a particular user. To address this challenge, SPB introduces a certificate of existence (CoE). Each smart meter constructs a Merkle tree of a number of PKs and sends the root of the tree to a smart meter to sign, which serves as the CoE. The smart meter generates each ERC transaction using a distinct PK stored in the Merkle tree, which prevents a malicious node from linking its transactions.

SPB: Secure Private Blockchain-Based Platform

The participating nodes in the smart grid including energy producers, consumers, prosumers, and distribution companies manage blockchain by storing and verifying transactions and blocks. We assume that smart meters are tamper-resistant. To reduce the blockchain processing overhead, a lightweight consensus algorithm, for example, Distributed Time-based Consensus (DTC) proposed in our previous work [2], is used. In DTC, each miner can only mine one block within a designated *consensus_period*.

The ledger of transactions created by a producer serves as its *energy account*. To establish an energy account, the producer must create a genesis transaction by [2]:

- Burning a specific amount of coin, for example, Bitcoin [11], meaning paying a specific amount of money to an unknown address.
- Receiving a certificate from trusted entities such as energy distributors.

This prevents malicious nodes from creating fake energy accounts intending to flood the network with fake energy trading requests. Users can employ multiple energy accounts to protect against malicious nodes that attempt to de-anonymize the user by analyzing the pattern of transactions in a ledger. However, creating multiple accounts requires the user to pay for either each additional genesis transaction or certificate. Thus, there is a tradeoff between increasing privacy and cost. We quantitively evaluate this tradeoff by measuring privacy and cost as a function of the number of PKs, that is, accounts, that a user employs. We use the methodology employed in [12], which measures the success rate in tracking

Despite its benefits, blockchain consumes a significant amount of computational, energy, and bandwidth resources, as appending (i.e., mining) a new block often involves executing resource consuming consensus algorithms, and all communications are broadcast, which increases overheads for direct negotiation between nodes.

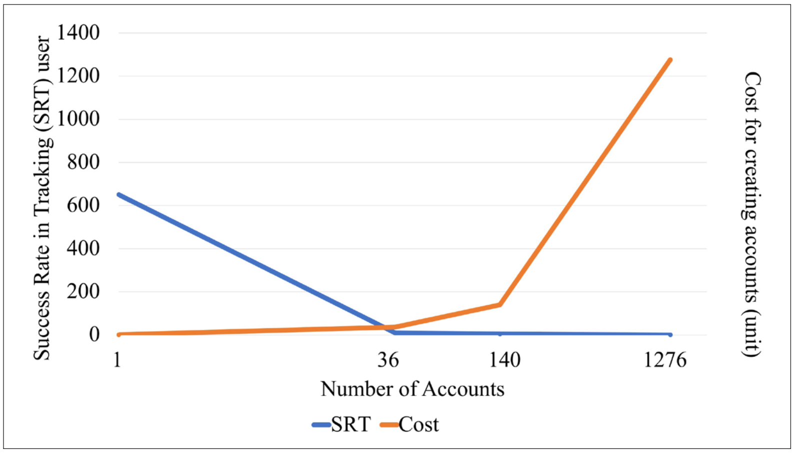


Figure 1. Evaluation on privacy and cost.

(SRT) an anonymous user based on the number of accounts employed by the user. As the cost of creating an account may vary for different certificate providers, we consider a generalized value and assume that the cost of creating an account is one unit. Figure 1 shows the SRT (the left vertical axis) and cost (the right vertical axis) based on results in [12].

Each network participant can add energy to its energy account by generating a *supply_energy* transaction, which includes:

"*T_ID* || *P_T_ID* || *energy_amount* || *energy_price* || *negotiatable* || *PK* || *sign*"

T_ID is the unique identifier of the transaction, which is the hash of the transaction. *P_T_ID* is the identifier of the previous transaction generated by the same node and ensures that the user knows the public/private keys associated with an energy account, and thus has once generated a genesis transaction. The next two fields specify the amount and price of the energy. The *negotiatable* flag status identifies whether the price is negotiable. Finally, the PK and signature of the user are populated in the transaction. The *supply_energy* transaction is sent to a smart contract which adds energy to the account of the energy producer. SPB consists of two main phases, namely negotiation and energy trading, which are discussed below.

Phase 1: Negotiation

The negotiation between the producer and consumer is conducted off-the-chain, meaning that the corresponding transactions are not stored in the blockchain, thus reducing the processing and packet overheads for broadcasting and mining those transactions. The final negotiated price is included in energy trading transactions (see Phase 2). The negotiation is conducted using the negotiation transaction, which includes:

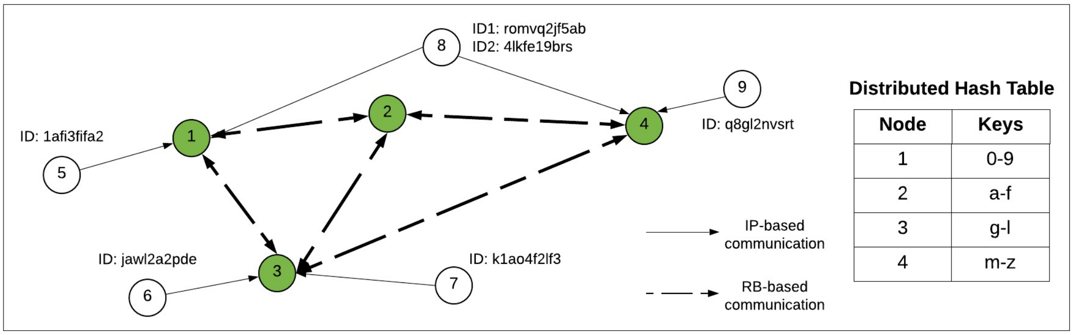
"*T_ID* || *energy_account.PK* || *price* || *status*"

where *T_ID* is the identity of the transaction, *energy_account.PK* is the PK of the energy account with which the consumer wishes to negotiate. The price field is the offered price by the consumer. The *status* field can be either "1" or "0," indicating that the offer is accepted or rejected by the producer, respectively. A producer that receives the negotiation transaction can accept the request by setting status to "1" or offer another price (status set to "0"). Thus, the producer and consumer may directly exchange negotiation transactions till they agree on the price.

Anonymous Routing Backbone: In conventional energy trading using blockchain, all transactions including those used for negotiation are broadcast, which increases the associated packet overhead and delay. To address this challenge, SPB proposes an anonymous routing backbone (ARB). Unlike traditional networks where participating nodes are known by their Internet Protocol (IP) addresses, blockchain participants are known by their PK. Thus, in ARB, packets are routed toward specific PKs. The participating nodes which have sufficient resources, for example, controller centers, serve as the backbone nodes which are responsible for routing negotiation transactions to the destination. The backbone nodes route transactions based on the *X* most significant bytes of the PK of the destination, known as routing bytes (RB). The backbone nodes initialize a distributed hash table (DHT) (see example in Fig. 2), which identifies the backbone node corresponding to each value of RB. The value of *X* is influenced by the total number of backbone nodes and the total load, that is, transactions, that need to be routed. In small backbones, *X* is very small (e.g., *X* = 1 in Fig. 2), while for larger backbones, *X* is increased. If a backbone node is overloaded by transactions, then the backbone nodes reconstruct the hash table and update the RB to a larger value. Increasing the RB does not guarantee balancing the load between backbone nodes as PKs are randomly generated. Participating nodes associate with the backbone nodes based on their PK. The backbone nodes use conventional routing protocols to route transactions toward the destination, for example, OSPF [RFC2328]. To ensure reliability, the destination backbone node sends an ACK back to the sender backbone node.

PK-Based Routing: When the backbone is formed, all nodes in the blockchain, referred to as regular nodes in the rest of the article, are notified of the DHT. Each regular node then associates with the backbone node that handles transactions for its PK by sending a join message which is signed with the private key corresponding to the PK. This protects against malicious nodes that may impersonate other nodes. A regular node may employ multiple PKs and thus be associated with multiple backbone nodes, for example, node 8 in Fig. 2. Regular nodes use the IP address of the backbone nodes to send transactions as shown in Fig. 2. However, the backbone nodes use the destination PK of the received transaction to determine the path to forward the transaction. Compared to traditional IP-based routing, ARB incurs a small processing overhead at the backbone nodes for looking up the destination PK.

To negotiate with a producer, the consumer sends an offer with its suggested price and energy requirement. The negotiation message is sent to one of the backbone nodes and is subsequently routed to the producer. The maximum number of offers and counteroffers that can be generated is limited to a certain threshold, *offer_limit*. This threshold ensures that malicious nodes are unable



Node	Keys
1	0-9
2	a-f
3	g-l
4	m-z

Figure 2. An illustrative example of ARB and the corresponding hash table.

An atomic process is an indivisible operation which appears to the rest of the system to occur at once without being interrupted. SPB defines two transactions that together form an atomic meta-transaction, such that they are valid if and only if both transactions are generated within a specific time period.

to launch a DoS attack by sending a large number of negotiation messages.

Phase 2: Energy Trading

After negotiations (if any), the consumer and the producer commence the energy trading process. SPB eliminates the need for a TTP by introducing *atomic meta-transactions* based on atomic swaps [13].

Atomic Meta-Transactions: An atomic process is an indivisible operation which appears to the rest of the system to occur at once without being interrupted. SPB defines two transactions that together form an atomic meta-transaction, such that they are valid if and only if both transactions are generated within a specific time period.

The first transaction is generated by the consumer to commit payment of the price agreed with the producer, and is thus known as a commit to pay (CTP) transaction. The CTP is a payment transaction that commits a specific amount of money to the producer. However, a CTP payment has *pending* status meaning that the consumer cannot spend the money committed in CTP, and the money is not transferred to the producer's account. Consequently, the CTP is not mined in the blockchain. Each miner maintains a database of pending CTPs, which is used for verification. To ensure the consistency of the CTP database among blockchain participants, a new header field is introduced for blocks named "*CTP hash*." The latter is populated by each miner with the hash of its CTP database copy. Ensuring consistency of the CTP database among participating nodes is similar to the issue of maintaining blockchain consistency. The CTP database reduces delay and processing overheads for trading energy compared to conventional blockchain-based solutions as the CTP transaction does not need to be mined. To secure SPB against double spending, where a malicious node pays the same amount of coin to multiple users, the miners check if the amount of money to be spent in the transaction is less than the sum of the amount of money in the user's account in the blockchain and the CTP database.

The structure of CTP transaction is as follows:

"*T_ID || Time Stamp || Expiry Time || Price || Contract Hash || PK || Sign*"

T_ID is the transaction identifier. *Time Stamp* is the time when the transaction is generated. Each CTP transaction can only be stored by the miners for a specific period of time, which is indicated in the *Expiry Time* field. This ensures that if the producer does not send energy to the consumer, the pending money of the consumer will be returned to their account after the expiry time. *Price* denotes the money that needs to be transferred between the producer and the consumer. *Contract Hash* is the hash of the amount and rate of energy that the producer and consumer have agreed on. The final two fields are the PK and signature of the consumer. The consumer signs the hash of the entire transaction to ensure integrity.

Once the CTP is broadcast, the producer receives the transaction and compares the contract hash with its own contract hash to ensure the consumer has not changed the amount or rate of the energy. Next, the producer begins delivering energy to the smart meter of the consumer. Once all of the energy is delivered, the consumer's smart meter generates an energy receipt confirmation (ERC) transaction. The network participants need to verify that a node that claims to be a meter is genuine to protect against malicious activities. Conversely, the meter must remain private as it reveals sensitive information including the electricity usage of its owner. To address this challenge, SPB proposes a method to anonymously verify a smart meter without revealing the true identity of the meter using a certificate of existence (CoE).

Certificate of Existence (CoE): The manufacturer of each meter creates a private/public key pair for each smart meter (known as M-PK-/*M-PK*$^+$, respectively) and serves as the certified authority (CA) for M-PK$^+$s. Once the smart meters are deployed, each meter produces a number of public/private keys (Fig. 3 step (1)). The smart meter owner determines the number of keys required depending on the level of anonymity desired. A large pool of key pairs achieves higher anonymity as transactions that are generated using the same key can be linked, thus revealing energy usage patterns. Instead, when multiple keys are used, the energy usage history of the user is split in different transaction ledgers which have no mutual links, thus making the aforementioned inferences difficult. After generating the key pairs, the smart meter constructs a Merkle tree (2) by recursively hashing the PKs which are stored as the leaves of a tree as shown in Fig. 4. The PKs in the Merkle tree and their corresponding private keys are later used to privately create an ERC transaction. A key feature of a Merkle tree is that the existence of a leaf can be proved with small overhead. As an illustrative example, to prove the existence of "A" in the Merkle tree shown in Fig. 4, one must store HB and HCD locally and reveal them to prove

CoE is proposed to protect the privacy of the consumer. There is no link between the VM and the smart meter that is using the CoE as the VM is selected randomly by the smart meter. Each meter may sign multiple CoEs for other smart meters. Consequently, tracking CoEs signed by particular meters will not compromise the privacy.

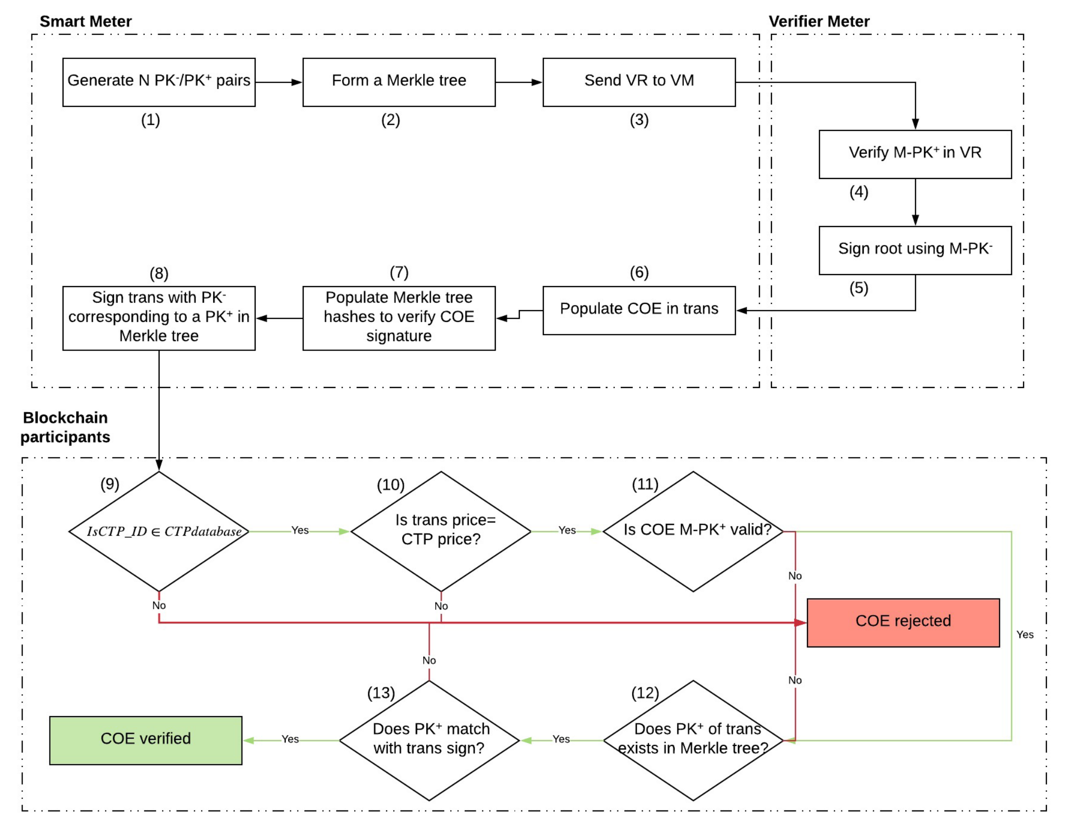


Figure 3. A summary of CoE generation and usage process.

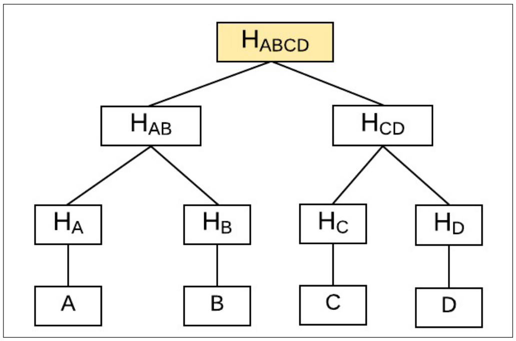


Figure 4. The structure of a Merkle tree.

existence. To verify the existence, HA is hashed by HB, and the result is then hashed by HCD. If the final hash matches with HABCD, then the existence of "A" is confirmed.

The meter generates a verification request (VR) with the following fields:

Root || *M-PK*$^{+}$ || *Sign*

where *Root* is root hash of the Merkle tree encrypted with the M-PK^{+} of a randomly chosen smart meter, known as a verifier meter (VM). The second field is the M-PK^{+} of the meter that generates the VR, followed by *Sign*, the signed hash of the VR by *M-PK*$^{-}$. Once populated, the meter routes VR to the VM using the proposed routing method (3). On receiving the Merkle tree root, the VM first verifies the smart meter, that has generated the root, using its M-PK^{+} and the CA of the manufacturer (4). If verified, it signs the received root using its M-PK^{-} (5) and returns the signed root back to the smart meter of which initially sent the transaction. The signed root of the Merkle tree is used as the CoE.

In order to generate the ERC transaction, the smart meter adopts one of the keys used to construct the Merkle tree in the CoE as the identity of the ERC transaction (6). The structure of the ERC transaction is as follows:

"T_ID || Time Stamp || CTP_ID || Price || CoE || CoE_PK || Merkle hashes || PK || Sign"

The first two fields are the same as the CTP transaction. The *CTP_ID* refers to the ID of the corresponding CTP transaction that is used by the miners to verify the ERC transaction as discussed below. Price denotes the total price of the contract. The *CoE* is as discussed above. The *CoE_PK* contains the PK of the VM used for verification of the CoE using the CA. Merkle hashes are the hashes that are required to prove the existence of the PK used to generate the ERC transaction in the Merkle tree (7). Finally, the last two fields are the PK of the ERC generator and its corresponding signature (8).

Finalizing the Trade: On receiving the ERC transaction, the miners verify the transaction by: a) verifying that the *CTP_ID* exists in the list of pending transactions (9); b) batching the price of the current transaction with the CTP price (10); c) verifying the signature and PK of the VM using the manufacturer CA (11); d) verifying the existence of the PK used to generate the transaction in the Merkle tree (12), and e) matching the PK hash in the Merkle tree with the signature on the transaction (13). If the above steps are successfully performed, then the ERC is validated. The ERC triggers the smart contract to pay the agreed price to the producer. If any of the steps is not successful, the CTP transaction of the consumer is removed from the CTP database after the expiry time.

Evaluation and Discussion

Security and Privacy Analysis

In this section, we discuss the security and privacy of SPB.

Privacy: We study the privacy of SPB from the perspective of the producer, consumer, and the consumer's smart meter. The producer can employ multiple energy accounts which protects them from being tracked. The consumer and their smart meter employ changeable PKs for communicating with multiple users to enhance their anonymity and thus provide a level of privacy.

CoE is proposed to protect the privacy of the consumer. There is no link between the VM and the smart meter that is using the CoE as the VM is selected randomly by the smart meter. Each meter may sign multiple CoEs for other smart meters. Consequently, tracking CoEs signed by particular meters will not compromise the privacy. The smart meter changes the PK used for each ERC transaction, thus protecting against potential tracking.

The Merkle tree in a CoE may be used by a single meter or by multiple meters that protect against malicious nodes who track a particular CoE.

Security: The security of SPB draws on the inherent security features of blockchain. Each transaction contains the hash of either the contract or the whole transaction that ensures integrity.

We discuss below the key attacks possible in SPB:

Malicious Producer: A malicious producer may not deliver energy to the consumer after receiving the CTP. In this case, the smart meter does not generate the corresponding ERC transaction as energy is not received. The CTP transaction is discarded from the CTP database by the expiry time and the consumer's money is released.

Malicious Consumer: The malicious consumer cannot receive energy without paying the producer as the energy transfer can only be triggered by a valid CTP.

CoE Attack: The malicious node pretends to be a smart meter by using the CoE of a genuine smart meter. The ERC transaction is signed by one of the keys in the Merkle tree which ensures that only the genuine node that knows the private key corresponding to one of the keys in the Merkle tree can use the CoE.

Malicious Backbone Node: The backbone node is compromised and does not deliver transactions to the regular nodes, which is similar to the blackhole attack in the literature [14]. The sender reverts to broadcasting the message to the entire network. The message eventually reaches the destination which would acknowledge receipt. Thus, the sender can be assured that the destination is alive. The sender then informs other backbone nodes of suspicious malicious behavior along the path to the destination. The backbone nodes run blackhole attack detection algorithms, such as [14], to detect malicious backbone nodes and eliminate them from the DHT and ARB.

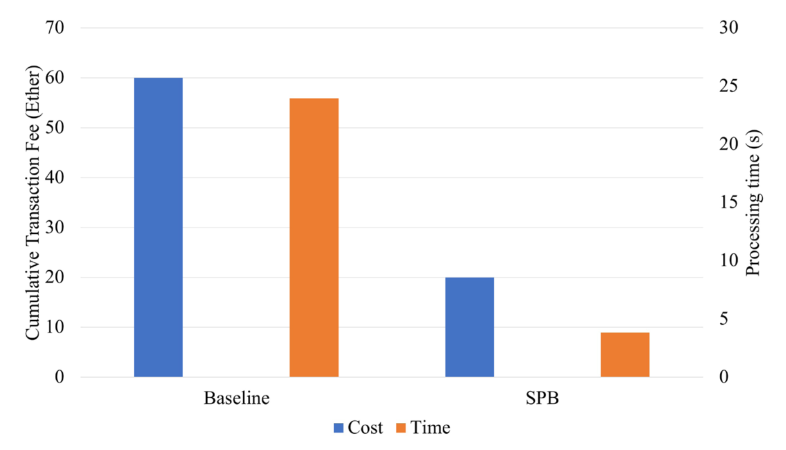


Figure 5. Evaluation of cost and processing time.

Implementation

To demonstrate the use of SPB by end users, we implemented the core SPB functions using the Solidity language that is then deployed in a private Ethereum version (testnet). The private network consists of the following three entities:

- Energy producer
- Energy consumer
- Miner

It is assumed that the energy producer and consumer have agreed on the price of energy off-the-chain. A video demonstration of the aforementioned scenarios is available at [15].

We evaluate the cumulative monetary cost that the end-user has to pay as a transaction fee, and the delay experienced by the energy consumer and producer. The delay incurred is exclusively the latency experienced for mining transactions and does not include the communication delay as the latter is not the focus of SPB. As a benchmark, we used Ethereum-based energy trading as proposed in the Introduction [7, 8], referred to as the *baseline*. Energy trading in the baseline involves mining three transactions: a smart contract, which is mined for each trade; the producer payment to the contract; and the contract payment to the consumer. Similar to SPB, we implement the baseline using Solidity in the testnet. The implementation results are shown in Fig. 5. The left and right vertical axes illustrate the monetary cost and delay, respectively. It is assumed that the cost of mining a transaction is 20 Ether. Since SPB requires mining of a single atomic transaction as compared to three transactions in the baseline, both the latency and cost are significantly lower with the former than the latter.

Conclusion

We proposed a Secure Private Blockchain-based framework (SPB) that enables energy prosumers to negotiate over the energy price and trade energy in a distributed manner. SPB employs a routing method for forwarding negotiation traffic, thus reducing the associated packet overhead. SPB eliminated the need for TTP using atomic

We evaluate the cumulative monetary cost that the end-user has to pay as a transaction fee, and the delay experienced by the energy consumer and producer. The delay incurred is exclusively the latency experienced for mining transactions and does not include the communication delay as the latter is not the focus of SPB.

meta-transactions. Smart meters are verified using a certificate of existence (CoE) without revealing information about the previous transactions of the same meter. Security analysis showed the robustness of SPB against several attacks. In the future, we aim to implement SPB on real devices and benchmark its performance.

References

[1] J. Giri *et al.*, "A More Intelligent Grid," *IEEE Power Energy Mag.*, vol. 7, no. 2, 2009, pp. 34–40.

[2] A. Dorri *et al.*, "Towards an Optimized Blockchain for IoT," *Proc. Second Int'l. Conf. IoTDI. ACM*, 2017.

[3] M. Mihaylov *et al.*, "NRGcoin: Virtual Currency for Trading of Renewable Energy in Smart Grids," *Proc. 11th IEEE Int'l. Conf. EEM*, IEEE, 2014, pp. 1–6.

[4] Z. Li *et al.*, "Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things," *IEEE Trans. Industrial Informatics*, 2017, 14.8, pp. 3690–3700.

[5] A. Laszka *et al.*, "Providing Privacy, Safety, and Security in IoT-Based Transactive Energy Systems Using Distributed Ledgers," *Proc. Seventh Int'l. Conf. Internet of Things*, ACM, 2017.

[6] power ledger, www.powerledger.io, accessed Oct. 10, 2018.

[7] M. Herlihy, "Atomic Cross-Chain Swaps," *Proc. 2018 ACM Symposium on Principles of Distributed Computing, PODC '18*, United Kingdom, pp. 245–54.

[8] https://dappsforbeginners.wordpress.com/tutorials/two-party-contracts/, accessed Oct. 10, 2018.

[9] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Ethereum project yellow paper 151, 2014, pp. 1–32.

[10] https://github.com/ethereum/wiki/wiki/Whisper, accessed Oct. 10, 2018.

[11] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[12] S. Han *et al.*, "Expressive Privacy Control with Pseudonyms," *ACM SIGCOMM Computer Commun. Review*, vol. 43, no. 4. ACM, 2013.

[13] D. B. Lomet, "Process Structuring, Synchronization, and Recovery Using Atomic Actions," *ACM Sigplan Notices*, vol. 12, no. 3, ACM, 1977.

[14] A. Dorri *et al.*, "DEBH: detecting and eliminating black holes in mobile ad hoc network," *Wireless Networks*, vol. 24, no. 8, pp. 2943–55.

[15] SPB Demo, available at: https://youtu.be/rX58GO_hQqI, accessed Oct. 10, 2018.

Biographies

Ali Dorri is a Ph.D. student at UNSW and a postgraduate research student at CSIRO, Australia. His research interests include blockchain applications and challenges for large scale networks including the Internet of Things, smart cities, smart grid, and e-health. He has published over 20 peer-reviewed papers. His publications in blockchain are being ranked among most popular and top-cited papers in their respective venues.

Fengji Luo received his bachelor and master degrees in software engineering from Chongqing University, China in 2006 and 2009. He received his Ph.D. degree in electrical engineering in 2014 from The University of Newcastle, Australia. Currently, he is a lecturer in the School of Civil Engineering, The University of Sydney, Australia. His research interests include energy demand side management, smart grid, and energy informatics. He has published over 100 papers in peer referred journals and conferences.

Salil S. Kanhere received his M.S. and Ph.D. degrees, both in electrical engineering, from Drexel University, Philadelphia. He is currently a professor in the School of Computer Science and Engineering at UNSW Sydney, Australia. His research interests include Internet of Things, pervasive computing, blockchains, crowdsourcing, privacy and security. He has published over 200 peer-reviewed articles and delivered over 40 tutorials and keynote talks on these research topics. He is a contributing research staff at Data61, CSIRO. He regularly serves on the organizing committee of a number of IEEE and ACM international conferences (e.g., IEEE PerCom, ACM MobiSys, ACM SenSys, ACM CoNext, IEEE WoWMoM, IEEE LCN). He currently serves as the area editor for Pervasive and Mobile Computing, and Computer Communications. Salil is a Senior Member of both the IEEE and the ACM. He is a recipient of the Humboldt Research Fellowship in 2014 and an ACM Distinguished Speaker.

Raja Jurdak is a senior principal research scientist at CSIRO, where he leads the Distributed Sensing Systems Group. He has an M.Sc. in computer engineering and a Ph.D. in information and computer science, both from the University of California. His current research interests focus on energy-efficiency, mobility, and trust in networks. He has over 145 peer-reviewed journal and conference publications, as well as a book published by Springer in 2007 titled *Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective*. He regularly serves on the organizing and technical program committees of international conferences (DCOSS, ICBC, RTSS, PERCOM, EWSN, ICDCS, Blockchain, MSWIM, WoWMoM). He is a Guest Editor for a Special Issue of *Elsevier IoT Journal* on Distributed Ledger Technology (DLT) for the Internet of Things. He is an honorary professor at UQ, and an adjunct professor at Macquarie University, JCU, and UNSW. He is a Senior Member of the IEEE.

Z.Y. Dong obtained a Ph.D. from the University of Sydney, Australia in 1999. He is now with the University of NSW, Sydney, Australia. He is also Director of the ARC ITRP Research Hub for Integrated Energy Storage Solutions, and Director of the UNSW Digital Grid Futures institute. He was the Ausgrid Chair and Director of the Ausgrid Centre for Intelligent Electricity Networks providing R&D support for the $600M Smart Grid, Smart City national demonstration project. He also worked as a manager for system planning at Transend Networks (now TASNetworks), Australia. His research interest includes smart grid, power system planning, power system security, load modeling, renewable energy systems, and electricity market. He is an editor of *IEEE Transactions on Smart Grid*, *IEEE Transactions on Sustainable Energy*, *IEEE PES Transaction Letters* and *IET Renewable Power Generation*. He is an international advisor for the *Journal of Automation of Electric Power Systems*. He is a Fellow of IEEE.