On the Design of Communication and Transaction Anonymity in Blockchain-Based Transactive Microgrids

Jonatan Bergquist
Datarella GmbH

Aron Laszka
Vanderbilt University

Monika Sturm
Siemens Corporate Technology

Abhishek Dubey
Vanderbilt University

ABSTRACT

Transactive microgrids are emerging as a transformative solution for the problems faced by distribution system operators due to an increase in the use of distributed energy resources and a rapid acceleration in renewable energy generation, such as wind and solar power. Distributed ledgers have recently found widespread interest in this domain due to their ability to provide transactional integrity across decentralized computing nodes. However, the existing state of the art has not focused on the privacy preservation requirement of these energy systems – the transaction level data can provide much greater insights into a prosumer's behavior compared to smart meter data. There are specific safety requirements in transactive microgrids to ensure the stability of the grid and to control the load. To fulfil these requirements, the distribution system operator needs transaction information from the grid, which poses a further challenge to the privacy goals. In this paper, we extend a recently developed trading workflow called PETra and describe our solution for communication and transactional anonymity.

CCS CONCEPTS

• **General and reference → Design**; • **Security and privacy → Pseudonymity, anonymity and untraceability**; *Security protocols*; Domain-specific security and privacy architectures; • **Computer systems organization →** *Peer-to-peer architectures*;

KEYWORDS

transactive energy platforms, blockchain, distributed ledger, privacy, anonymity, onion routing, zero-knowledge proofs

ACM Reference format:
Jonatan Bergquist, Aron Laszka, Monika Sturm, and Abhishek Dubey. 2017. On the Design of Communication and Transaction Anonymity in Blockchain-Based Transactive Microgrids. In *Proceedings of SERIAL'17: ScalablE and Resilient InfrAstructures for distributed Ledgers, Las Vegas, NV, USA, December 11–15, 2017 (SERIAL'17),* 6 pages.
DOI: 10.1145/3152824.3152827

1 INTRODUCTION

Transactive energy models have been proposed as a set of market based mechanisms for balancing the demand and generation of energy in communities [8, 15, 20]. In this approach, customers on the same feeder (i.e. sharing a power line link) can operate in an open market, trading and exchanging generated energy locally. Distribution System Operators (DSOs) can be the custodians of this market, while still meeting the net demand [9]. Blockchains have recently emerged as a foundation for enabling the transactional service in the microgrids. For example, the Brooklyn Microgrid (brooklynmicrogrid.com) is a peer-to-peer market for locally generated renewable energy, which was developed by LO3 Energy as a pilot project. Similarly, RWE, and Grid Singularity have developed blockchain based solutions for incentivizing neighbors to sell excess energy to the grid and payments for electric car charging. However, those solutions do not address the requirements for off-blockchain communication network and the requirements for privacy.

Specifically, while blockchains provide the necessary ledger services, we still need a communication network for sending control commands from the DSO to the prosumers as well as initiating the trade matching mechanisms. Additionally, this communication network and the blockchain itself must preserve the privacy of the prosumers. Energy usage patterns (actual or predicted) are sensitive, personally identifiable data. Legal requirements and security considerations make it mandatory to provide a mechanism to hide the identities and transaction patterns of trade partners. Additionally, solutions must also satisfy safety requirements, which often conflict with privacy goals. For example, to prevent a prosumer from destabilizing the system through careless or malicious energy trading, a transactive grid must check all of the prosumer's transactions. In a decentralized system, these checks require disseminating information, which could be used to infer the prosumer's future energy consumption.

In [16], we introduced *Privacy-preserving Energy Transactions (PETra)*, which is our distributed-ledger based solution that (1) enables trading energy futures in a secure and verifiable manner, (2) preserves prosumer privacy, and (3) enables distribution system operators to regulate trading and enforce the safety rules. In this paper, we extend the communication and transaction anonymity mechanisms. The key contributions of this paper are (a) a survey of the key concepts required for implementing the anonymity across the two dimensions, (b) a discussion on the threats that must be considered when we implement the anonymization mechanisms, and lastly (c) a discussion on implementing the anonymization extensions in PETra.

SERIAL'17, Las Vegas, NV, USA
© 2017 ACM. 978-1-4503-5173-7/17/12. . . $15.00
DOI: 10.1145/3152824.3152827

The outline of this paper is as follows. We first present an overview of the PETra workflow described in [16] in Section 2. We then discuss the communication anonymity extensions in Section 3.1 and transaction anonymity in Section 3.2. Section 3.1.2 discusses the threat vectors for the communication anonymity approach. Section 3.2.3 describes the transaction anonymity threats. Finally, we provide concluding remarks in Section 4.

2 PRIVACY-PRESERVING ENERGY TRANSACTIONS

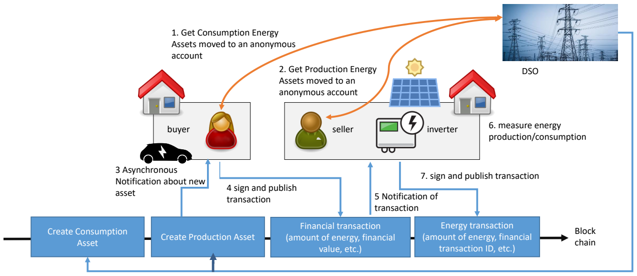
1. Get Consumption Energy Assets moved to an anonymous account
2. Get Production Energy Assets moved to an anonymous account
DSO
buyer
seller
inverter
6. measure energy production/consumption
3 Asynchronous Notification about new asset
4 sign and publish transaction
5 Notification of transaction
7. sign and publish transaction
Create Consumption Asset
Create Production Asset
Financial transaction (amount of energy, financial value, etc.)
Energy transaction (amount of energy, financial transaction ID, etc.)
Block chain

Figure 1: The sequence of activities in PETra. The orange arrows show off-block chain communication and blue arrows show transactions on block-chain. Producers and consumers request the DSO to allocate the energy production and consumption assets to blockchain. The consumers receive asynchronous notifications about offers from producers. Thereafter, they can finalize a transaction. The energy and financial transfer happens at a later time and is also recorded on the chain.

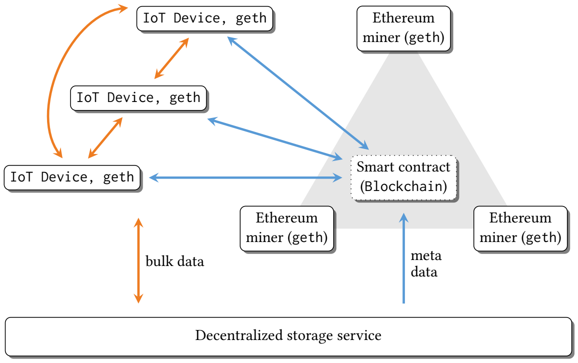

Figure 2: Components of IoT Blockchain pattern. Typically the IoT devices communicate with each other over a messaging middleware (red arrows). They communicate with blockchain and smart contracts (blue arrows) through clients, for example the Ethereum geth client. The miners are entities responsible for validating the events/transactions.

There is a systematic pattern emerging in the domain of Internet of Things which requires transactional capabilities. Examples include transactive ride-share systems [35], transactive health-care systems [2], and transactive energy systems described earlier in this section. As shown in Figure 2, there are three separate layers of these transactions. The first layer is the distributed ledger, which is responsible for keeping track of all log of all events of interest; in the energy domain these events are trades, energy transfers and financial transactions. In the health care domain, the events record the time of access of the health care data. The data is not stored in the blockchain due to the size and privacy issues. Rather, the data is stored in the second layer, which can be implemented by either a cloud or a decentralized storage service like Storj or IPFS. The third layer is the IoT layer, which is responsible for sensing and control. This third layer is typically implemented using messaging middlewares like MQTT, DDS, etc..

A key aspect is the tight integration of distributed messaging patterns between actors and the blockchain-based communication network used for transferring transactional information. In the transactive energy domain, PETra involves the interactions between DSO, prosumer, and a smart contract. The smart contract is keeping track of the energy and financial assets enabling prosumers to post trade offers and exchange assets when a prosumer decides to accept. PETra uses quantised energy asset tokens[1] that can represent the amount of power to be produced or consumed (for example, measured in watts), the time interval in which energy is to be produced (or consumed) and the last time interval in which energy is to be produced (or consumed) (Figure 1 describes the full sequence of activity). These assets are withdrawn and submitted to anonymized accounts on behalf of prosumers by the distribution system operator, which is also responsible for validating that the specific prosumer has the energy capacity for feasible trades given the assets. Once the DSO posts the assets into the blockchain, prosumers can trade between themselves using these quantised assets and anonymized addresses, hiding their identity from each other. The DSO is also responsible for releasing and managing the transfer of currencies, which are represented by financial assets. This is simply an unsigned integer value, denominated in a fiat currency. In this workflow, there are both on- and off-blockchain communications between DSO and prosumer. The off-blockchain communication is required to request the transfer of assets. On-blockchain communication occurs via filters that track the posting of assets. Similarly, prosumers communicate with each other via blockchain to indicate when an offer has been posted and when a transaction has cleared.

While all of the transactive IoT systems require communication and transactional anonymity there are domain-specific requirements and challenges that must be considered. These characteristics and requirements guide us in the description of the anonymization architecture that we describe in the rest of this paper. Specifically, these characteristics are as follows: (1) transactions in a microgrid must clear in bounded time and any errors must be detected[2], (2) there is a dedicated communication channel available in a microgrid

[1] There are two kinds of energy tokens: Energy Production Asset and Energy Consumption Asset. Token attributes include power and time interval for which the token is valid.

[2] Energy trades that have an impact on real-time control (e.g., selling energy production for the near future) must be permanently recorded on the ledger *in time* since grid control signals cannot be delayed.

that connects the prosumers and the distribution system operator, (3) the set of participants in the network are fixed and known ahead of time. Thus, a discovery procedure is typically not required. Finally, (4) even though all the transactions are anonymous there is still a need for maintaining associativity of properties like: maximum generation capacity[3], reputation scores to prosumers as they participate in trades while reducing the likelihood of jeopardizing the stability of the microgrid[4]. In the next two sections, we describe the mechanisms for implementing communication and transaction anonymity in this workflow.

3 COMMUNICATION AND TRANSACTIVE INFRASTRUCTURE

3.1 Communication Anonymity

The anonymous communication layer is the infrastructure upon which all other anonymity services in PETra are built. The goal of communication anonymity is to allow smart meters and users to exchange transactions and bids without revealing their IP-addresses or other information which can be used to identify them. In almost all cases, at the very least the Internet Service Provider (ISP) has information about the users' communications and identities. The goal of this section is to maximize the anonymity to such an extent that not even ISPs can identify users. Existing protocols for low-latency communication anonymity include onion routing [25] or the similar garlic routing [18], STAC [14] and the decentralized Matrix protocol.[5] In this section, we present a survey of onion and garlic routing, especially with respect to application in PETra.

3.1.1 Onion and Garlic Routing. Onion routing is based on messages in communication being encapsulated in multiple layers of encryption and sent through a number of nodes in a network, called onion routers. The most widespread implementation of the protocol is called the Tor network. It is anonymous since no single node, except the sender and the receiver, can know the origin and the recipient of the message. In Figure 3, an example shows how smart meter A encrypts a message *m*, with final destination G, through a network of onion routers. A encrypts the message, for example a confirmation of an energy purchase, a certain number of times, along with addresses of members of the onion network. Each subsequent node, selected by the sender and specified in the different layers of encryption, decrypts one layer using its private key, revealing the next node to which the encrypted message is forwarded. Finally, the second to last node reveals the address of smart meter G and sends the still encrypted message to G, who can decrypt it safely. No single node in the network, except for the sender, knows how many times the packages is re-routed, and no node except for the sender and recipient can know their internal position in the chain of routing. A related technique for communication anonymity is called *garlic routing*. It differs from onion routing in that multiple messages are encrypted together to counter tracing attacks. In practice, the deployment of garlic routing in the Invisible Internet Project (I2P) works as follows. Each node in the network operates an I2P router, allowing for anonymous communications. A router is distinct from an endpoint application in that it is not a secret who runs a router. By contrast, an application is the destination for the communications and is anonymous. This disconnect allows for a higher degree of anonymity. To communicate between routers, unidirectional tunnels are set up. The tunnels use layered encryption, meaning that each router in the tunnel can only decrypt one layer. In order to transmit a message between two routers, the sender needs to know where to direct the message, i.e., what the address of the entry point of the receiver is.

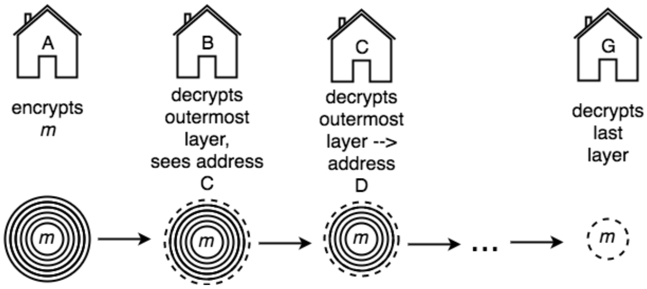
A
B
C
G
encrypts m
decrypts outermost layer, sees address C
decrypts outermost layer --> address D
decrypts last layer
m
m
m
...
m

Figure 3: The principle behind onion and garlic routing. The difference being that in onion routing, *m* is a single message, whereas in garlic routing, *m* is multiple messages packaged together.

The I2P protocol differs from regular network communications in that, for communications to take place between routers, each router needs to know a structure called the *RouterInfo*. It contains the 2048-bit ELGamal encryption key, a signing key, a certificate, timestamp, text field, signature of bundle and the contact addresses where a router can be reached. The RouterInfo is given along with something called a *LeaseSet*, containing a group of tunnel entry points for a particular client destination, when the tunnel will expire, the destination itself, encryption key for end-to-end encryption of garlic messages, revocation key and a signature of the LeaseSet data. The LeaseSet identifies an application on the I2P network. The I2P protocol ensures the anonymity of its users because of the disconnect between the identities of the applications communicating over the network, and the identities of the routers. This metadata is stored in a distributed directory called the netDb, based on the Kademlia P2P-protocol, which describes a provably consistent and fault-tolerant distributed hash table [19]. The RouterInfo and LeaseSet data are stored on the netDb under the key derived from the SHA256 of the destination.

3.1.2 Threat Vectors in Onion and Garlic Routing. Murdoch and Danezis [23] show that a low-cost traffic analysis is possible of the Tor-network, theoretically and experimentally. Traffic analyses are based on tracking the forwarding of the size of a data package between computers, for example, if computer A sends a package of exactly 42 bytes to computer J, who then sends a package of exactly 42 bytes to B, it can be easily deduced that A sent a package of unknown content to computer B. This is possible because of the distribution of metadata to all routers in the Tor-network [13]. In what is called a timing analysis attack, an attacker tries to find a correlation between the timing of messages moving through the

[3] To prevent destabilization of the grid, a producer is not allowed to bid more than its maximum generation capacity.

[4] A prosumer with low reputation score might have a history of not fulfilling the energy transfer obligations

[5] Open-federated protocol for instant messaging, Voice-over-IP and IoT communications (https://matrix.org/).

network to gain information about user identities and their communications. Analyses have shown that these types of attacks can be very effective over a wide range of network parameters when specific defences are not employed [17, 34]. To counter timing analysis attacks, the I2P network bundles multiple messages together (principle of garlic routing) and renders it more difficult to analyse [18]. Schimmer, 2009, showed that the bandwidth opportunistic peer-selection and -profiling algorithm does not prioritize anonymity in favor of performance [30]. Herrmann and Grothoff, 2011, exposed a potential weakness in anonymous HTTP-hosting done over the I2P network [12]. The arguably only practical attack against the I2P network was done against the directory, the netDb, by Egger *et al.* [10]. An improvement of the protocol, aimed at Egger *et al.*'s attack was suggested by Timpanaro *et al.*, 2015 [32].

Another potential weakness of onion routing and garlic routing is that, even though the actual message is encrypted and the destinations are unknown, there is always a trace of the communication at the ISP level. The fact that a connection took place will be logged and is openly visible at the very least to the ISP. This attack can be countered in PETra by each node transacting and participating in the mixing network, regardless of the need for trading at that time. Trading of "zero"-assets can help obfuscate the non-zero-assets of others. Another liability in onion and garlic routing can be that the legitimacy of the sender can not be immediately verified. This can be achieved by the techniques described in the section Transaction Anonymity.

3.1.3 Proposed Solution. Given the survey of the previous paragraphs, performing P2P energy trading in transactive grids over a garlic routing network protocol such as the I2P network provides a high amount of communication anonymity for users. Only part of the energy trading in PETra will be anonymized by garlic routing, namely the internet connections. PETra is no different from other network communications in that aspect. The particularity of the trading being local and thus IP-addresses being close, is a potential weakness that can be countered by creating "fake" IP-addresses. To apply garlic routing to transactive microgrids, the smart meters, prosumers, and DSO can act as onion routers, and distribution of available routers is done over netDb. In practice, this service can be built on the free and open-source I2P software with private Directory Authorities. In this case, anonymous communication identifiers in bids and asks correspond to public-keys that identify I2P applications.

3.2 Transaction Anonymity

Communication anonymity is necessary but not sufficient for anonymous trading, as the cryptographic objectives of authentication and legitimacy are not fulfilled. We suggest using cryptographic techniques from distributed ledgers, *blockchains* and cryptocurrencies. The most adopted one, Bitcoin allows for very simple digital cash spending but has serious privacy and anonymity flaws [1, 3, 26]. Additionally, Biryukov and Pustogarov, 2015, show that using Bitcoin over the Tor network opens a new attack surface [4]. Solutions to the tracing and identification problems identified by these researchers have been proposed and implemented in alternative cryptocurrency protocols: mixing using ring signatures and zero-knowledge proofs [21, 33].

3.2.1 Mixing Through Ring Signatures. A proposed improvement to standard ring signatures is the CryptoNote protocol, which prevents tracing assets back to their original owners by mixing together incoming transactions and outgoing transactions. This service hides the connections between the prosumers and the addresses. Mixing requires the possibility to create new wallets at will and the existence of a sufficient number of participants in the network. Monero is an example of a cryptocurrency that provides built-in mixing services by implementing the CryptoNote protocol [24]. There are however alternative implementations of mixing protocols such as CoinShuffle [28] or Xim [5]. A variant of ring signatures, group signatures, were first introduced by Chaum and van Heyst, 1991, [7] and then built upon by Rivest *et al.*, 2001 [27]. The basis for anonymity in the CryptoNote protocol, however, is a slightly modified version of the *traceable ring signature* algorithm by Fukisaki and Suzuki, 2007 [11]. This allows a member of a group to send a transaction so that it is impossible for a receiver to know any more about the sender than that it came from a group member without the use of a central authority. The CryptoNote protocol achieves two objectives:

(1) Untraceable transactions - *for each incoming transaction all possible senders are equiprobable.*
(2) Unlinkable transactions - *for any two outgoing transactions it is impossible to prove they were sent to the same person* [33].

Unlinkability is achieved by *one-time ring signatures*, making use of four algorithms: **GEN, SIG, VER, LNK**. The general principle of the unconditional unlinkability is that a sender signs a transaction using a public key and a key image generated by **GEN** and produces a one-time ring signature using **SIG** and the public key pair and key image. **SIG** makes use of a non-interactive zero-knowledge proof which the verifier(s) then use to check the signature in **VER**. If the signature is valid, the verifier checks if the key image has been used in previous transactions, which mean that the same secret key was used to produce multiple signatures. She does that by running the algorithm **LNK**. Assuming that the mapping of the secret key to the key image is a one-way injection, it is certain that: **A.** The signer is not identifiable by way of recovering the secret key from the key image. **B.** The signer cannot create another key image with the same secret key without double-spending.

Additionally, if the receiver and sender have randomly generated, unique and new addresses, the Diffie-Hellman protocol can be used to generate a new pair of public-private keys. This is how untraceability of public keys is achieved. The sender should generate ephemeral keys for each transfer, enabling only the receiver to recover the corresponding private key. As an example, in Figure 4, a diagram shows households A, B and C signing a transaction since they are part of the same ring. A group would, in reality, be many more households, not necessarily of the same microgrid. Let's assume that A is the true origin of the transaction. When E receives the transaction, the only thing that E can know with certainty is that one of A, B or C initiated the transaction. To increase the transaction anonymity further, a second, third or n rounds of group signatures can be imposed upon the network. With each round of signing parties, the group of potential origins grows linearly. Notably, the ring signature algorithm by Fujisaki and Suzuki, [11], has been published in a peer-reviewed paper. This can be compared favorably to many cryptocurrency protocols which are simply published as white papers without any formal review-process [33].

It is also possible for household A that it paid prosumer B for energy by either disclosing the random number used in the generation of the one-time public destination key used in that transaction to B. Or she can use any other kind of zero-knowledge protocol to prove she knows the random number. The ring signatures would also allow the auditing of transactions by, for example, the DSO. This would be achieved by prosumer B giving the tracking key or truncated address to the DSO, who would then be able to link all incoming transactions to B.

3.2.2 Mixing through Zero-Knowledge Proofs. Zero-knowledge proofs (ZKP) are ways for a person to prove the knowledge of some specific fact to a verifier, without actually having to disclose the knowledge. Blum *et al.* provided non-interactive ZKPs (NIZK) in 1988 [6], where the prover and verifier don't have to interact or communicate directly with each other. The Zerocoin protocol [21] outlines a way how NIZKs can achieve the untraceability objective of the previous section and it ensures that no double-spending is allowed.[6] Zerocoin is a protocol for the decentralized mixing of coins, so that they can not be traced, or *tainted*. However, senders and destinations can still be identified [21]. Luckily, Zerocash [29] extends the NIZK functionality to allow for anonymous transactions, anonymous balances and coins, improved performance of transactions and sending of assets to a receivers fixed address without action required from the receiver. It makes use of a more efficient version of the NIZK, used in Zerocoin, called zk-*Succinct Non-interactive ARguments of Knowledge* (zk-SNARK).

The Zerocash-scheme could be carried out using a simple messaging board, but would not be safe in practice since information might be manipulated or the owner of said board might collude etc.. Therefore, an immutable, decentralized data storage, governed by the consensus of its peers is required to assure the secure transmission of information. The blockchain provides such a structure.

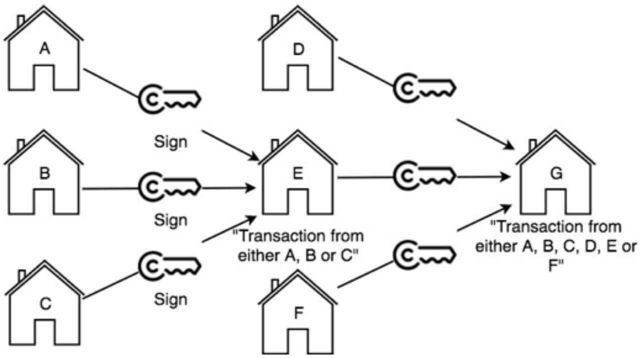
A
B
C
D
E
F
G
Sign
Sign
Sign
"Transaction from either A, B or C"
"Transaction from either A, B, C, D, E or F"

Figure 4: Visualization of untraceability in ring signatures in smart meter-based energy trading and the potential deductions of origin of the transaction by a single household in the chain of signatures.

3.2.3 Threats and Weaknesses in Ring Signature- and Zero-Knowledge Proof Schemes. When applying either ring signatures or zero-knowledge SNARKs to PETra, potential weaknesses or attacks need to be considered. A potential threat to ring signatures is when a large amount of the unspent transactions are owned by an adversary or when insufficient amounts of signatures are included in a ring. When a prosumer A wishes to select a group of signatures to sign her transaction as well, then it is likely that she will select many of the transactions from the adversary. Assuming the adversary spends his outputs without *mixing*[7], then A's transaction is exposed as well [31]. Recent research also show that up to 65% of Monero transactions are trivially traceable using one attack. They also exposes two more attacks that have been amended in the latest versions of the protocol, lowering the amount of transactions traceable to 20% [22, 31]. To protect against a user signing two transactions of the same amount simultaneously in different groups, Noether [24] proposes the RingCT improvement to CryptoNote. It uses *Multi-layered Linkable Spontaneous Anonymous Groups* (MLSAGs) to achieve this, which are built upon the ring signatures of [11].

One of the main weaknesses of the Zerocash-based protocol is that for each private transaction, a costly zk-SNARK needs to be computed. This is not a threat to anonymity, just a practical reason why it might be difficult to run the scheme over a congested public blockchain. In [29], experiments show an average time of 3 min to create the zk-SNARK for a private transaction, verifying it takes only 8.5 ms. Another large practical drawback of Zerocash is the lack of programmability and functionality that would be required in PETra. Zhang *et al.* solve some of the practical flaws and amend security issues [36].

3.2.4 Proposed Solution to Achieve Transaction Anonymity. Applying the CryptoNote protocol to PETra could be done by performing both energy transactions and monetary transactions using ring signatures. They would be securely logged, tamper-proof and anonymous through the usage of a blockchain. Even though some security flaws exists, as seen in the previous paragraph, the risk of identification, linking or tracing of transactions can be minimized by imposing a high minimum number of signatures per transaction. We also propose to connect the global transaction networks to augment the number of transactions and thereby limit the chance of deduction by elimination. In order to not reveal the denominations of the transactions, a scheme proposed in [24] should be used. Given a public key P and an amount a, the pair (P, a) can be used as input to a transaction (P, aH) where H is a masking point. All the input amounts will not be masked, but the outputs from this transaction will be hidden, and the necessary relations from [see 24, Section 4.] will hold.

Applying ZKPs to PETra would require that a smart meter can encrypt and sign a transaction, transmit a proof of it to the blockchain and thus the receiver of the payment, without having to reveal the actual amount of energy or cost incurred to anyone but the receiver. This is achieved by the Zerocash-protocol and is implemented as a fork of the Bitcoin blockchain. Neither the receiver, nor any other participants can gain information about the transactions sent over the blockchain. To provide full functionality for

[6] Each coin in the protocol is identified uniquely by a serial number.

[7] The number of other signatures used in the ring.

PETra, the Zerocash-protocol would need to be implemented for the transmission of bids and asks as well as the already existing monetary transactions. The second implementation would need to be modified to transmit and link bids and asks to the payments ledger. A more straightforward but bloated structure would be to create transactions without monetary value to post a bid or an ask and then directly reference the final bid-/ask-transaction in the payment-transaction.

4 CONCLUSION AND DISCUSSION

Through the use of garlic routing and ring signatures, communication and transaction anonymity under certain weak assumptions can be achieved. A garlic routing network such as I2P can ensure that no usage, bid, ask or identifiable data is leaked from the system. By using ring signatures, transactions cannot be traced, but it can still be proven that a bid or an ask has been responded to and that a transaction has taken place. The design we've proposed anonymizes the whole chain of transactions, both on a network communication layer and on a distributed ledger transaction layer.

As for the DSO, it receives the same information from the smart meter as in a non-transactive smart grid (i.e., amount of energy produced and consumed). In particular, since price policies are recorded on the ledger (which the smart meters may read), each prosumer's smart meter may calculate and send the prosumer's monthly bill to the DSO, without revealing the prosumer's energy consumption or production. The DSO still gets aggregate information regarding load on the grid, but cannot identify individual users and their energy prosumption.

Acknowledgment: This work was funded in part by a grant from Siemens Corporation, CT.

REFERENCES

[1] M. Apostolaki, A. Zohar, and L. Vanbever. 2017. Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. In *2017 IEEE Symposium on Security and Privacy (SP)*. 375–392. DOI: https://doi.org/10.1109/SP.2017.29

[2] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. 2016. Medrec: Using blockchain for medical data access and permission management. In *Open and Big Data (OBD), International Conference on*. IEEE, 25–30.

[3] Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun. 2012. *Bitter to Better — How to Make Bitcoin a Better Currency*. Springer Berlin Heidelberg, Berlin, Heidelberg, 399–414. DOI: https://doi.org/10.1007/978-3-642-32946-3_29

[4] A. Biryukov and I. Pustogarov. 2015. Bitcoin over Tor isn't a Good Idea. In *2015 IEEE Symposium on Security and Privacy*. 122–134. DOI: https://doi.org/10.1109/SP.2015.15

[5] George Bissias, A Pinar Ozisik, Brian N Levine, and Marc Liberatore. 2014. Sybil-resistant mixing for Bitcoin. In *Proc. of 13th Workshop on Privacy in the Electronic Society*. ACM, 149–158.

[6] Manuel Blum, Paul Feldman, and Silvio Micali. 1988. Non-interactive Zero-knowledge and Its Applications. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing (STOC '88)*. ACM, New York, NY, USA, 103–112. DOI: https://doi.org/10.1145/62212.62222

[7] David Chaum and Eugène van Heyst. 1991. *Group Signatures*. Springer Berlin Heidelberg, Berlin, Heidelberg, 257–265. DOI: https://doi.org/10.1007/3-540-46416-6_22

[8] William Cox and Toby Considine. 2013. Structured energy: Microgrids and autonomous transactive operation. In *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES*. IEEE, 1–6.

[9] O. Dag and B. Mirafzal. 2016. On stability of islanded low-inertia microgrids. In *Proc. of 2016 Clemson University Power Systems Conference (PSC)*. 1–7.

[10] Christoph Egger, Johannes Schlumberger, Christopher Kruegel, and Giovanni Vigna. 2013. *Practical Attacks against the I2P Network*. Springer Berlin Heidelberg, Berlin, Heidelberg, 432–451. DOI: https://doi.org/10.1007/978-3-642-41284-4_22

[11] Eiichiro Fujisaki and Koutarou Suzuki. 2007. *Traceable Ring Signature*. Springer Berlin Heidelberg, Berlin, Heidelberg, 181–200. DOI: https://doi.org/10.1007/978-3-540-71677-8_13

[12] Michael Herrmann and Christian Grothoff. 2011. *Privacy-Implications of Performance-Based Peer Selection by Onion-Routers: A Real-World Case Study Using I2P*. Springer Berlin Heidelberg, Berlin, Heidelberg, 155–174. DOI: https://doi.org/10.1007/978-3-642-22263-4_9

[13] Nicholas Hopper, Eugene Y. Vasserman, and Eric Chan-TIN. 2010. How Much Anonymity Does Network Latency Leak? *ACM Trans. Inf. Syst. Secur.* 13, 2, Article 13 (March 2010), 28 pages. DOI: https://doi.org/10.1145/1698750.1698753

[14] S. Jebri, M. Abid, and A. Bouallegue. 2017. STAC-protocol: Secure and Trust Anonymous Communication protocol for IoT. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. 365–370. DOI: https://doi.org/10.1109/IWCMC.2017.7986314

[15] Koen Kok and Steve Widergren. 2016. A society of devices: Integrating intelligent distributed resources with transactive energy. *IEEE Power and Energy Magazine* 14, 3 (2016), 34–45.

[16] A. Laszka, A. Dubey, M. Walker, and D. Schmidt. 2017. Providing privacy, safety and security in IoT-based transactive energy systems using distributed ledgers. In *Proceedings of the 7th International Conference on the Internet of Things*.

[17] Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew Wright. 2004. *Timing Attacks in Low-Latency Mix Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 251–265. DOI: https://doi.org/10.1007/978-3-540-27809-2_25

[18] Peipeng Liu, Lihong Wang, Qingfeng Tan, Quangang Li, Xuebin Wang, and Jinqiao Shi. 2014. Empirical Measurement and Analysis of I2P Routers. *JNW* 9 (2014), 2269–2278.

[19] Petar Maymounkov and David Mazières. 2002. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems (IPTPS '01)*. Springer-Verlag, London, UK, UK, 53–65. http://dl.acm.org/citation.cfm?id=646334.687801

[20] Ronald B Melton. 2013. *Gridwise transactive energy framework (draft version)*. Technical Report. Pacific Northwest National Laboratory, Richland, WA.

[21] Ian Miers, Christina Garman, Matthew Green, and Aviel D Rubin. 2013. Zerocoin: Anonymous distributed e-cash from Bitcoin. In *Proc. of 2013 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 397–411.

[22] Andrew Miller, Malte Möser, Kevin Lee, and Arvind Narayanan. 2017. An Empirical Analysis of Linkability in the Monero Blockchain. *CoRR* abs/1704.04299 (2017). http://arxiv.org/abs/1704.04299

[23] S. J. Murdoch and G. Danezis. 2005. Low-cost traffic analysis of Tor. In *2005 IEEE Symposium on Security and Privacy (S P'05)*. 183–195. DOI: https://doi.org/10.1109/SP.2005.12

[24] Shen Noether. 2015. Ring Signature Confidential Transactions for Monero. Cryptology ePrint Archive, Report 2015/1098. (2015). http://eprint.iacr.org/2015/1098.

[25] Michael G Reed, Paul F Syverson, and David M Goldschlag. 1998. Anonymous connections and onion routing. *IEEE Journal on Selected areas in Communications* 16, 4 (1998), 482–494.

[26] Fergal Reid and Martin Harrigan. 2013. *An Analysis of Anonymity in the Bitcoin System*. Springer New York, New York, NY, 197–223. DOI: https://doi.org/10.1007/978-1-4614-4139-7_10

[27] Ronald L. Rivest, Adi Shamir, and Yael Tauman. 2001. *How to Leak a Secret*. Springer Berlin Heidelberg, Berlin, Heidelberg, 552–565. DOI: https://doi.org/10.1007/3-540-45682-1_32

[28] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. 2014. CoinShuffle: Practical decentralized coin mixing for Bitcoin. In *19th European Symposium on Research in Computer Security (ESORICS)*. Springer, 345–364.

[29] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. 2014. Zerocash: Decentralized Anonymous Payments from Bitcoin. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy (SP '14)*. IEEE Computer Society, Washington, DC, USA, 459–474. DOI: https://doi.org/10.1109/SP.2014.36

[30] Lars Schimmer. 2009. Peer Profiling and Selection in the I2P Anonymous Network. In *PET-CON 2009.1*. TU Dresden, Germany.

[31] Sarang Noether Shen Noether and Adam Mackenzie. 2014. A Note on Chain Reactions in Traceability in CryptoNote 2.0. Cryptology ePrint Archive, Report 2014. (2014). https://lab.getmonero.org/pubs/MRL-0001.pdf.

[32] J. P. Timpanaro, T. Cholez, I. Chrisment, and O. Festor. 2015. Evaluation of the anonymous I2P network's design choices against performance and security. In *2015 International Conference on Information Systems Security and Privacy (ICISSP)*. 1–10.

[33] Nicholas van Saberhagen. 2012. *CryptoNote v 2.0*. Technical Report. 3 pages. https://cryptonote.org/whitepaper_v2.pdf

[34] L. Xin and W. Neng. 2009. Design Improvement for Tor against Low-Cost Traffic Attack and Low-Resource Routing Attack. In *2009 WRI International Conference on Communications and Mobile Computing*, Vol. 3. 549–554. DOI: https://doi.org/10.1109/CMC.2009.18

[35] Yong Yuan and Fei-Yue Wang. 2016. Towards blockchain-based intelligent transportation systems. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2663–2668.

[36] Yuncong Zhang, Yu Long, Zhen Liu, Zhiqiang Liu, and Dawu Gu. 2017. Z-Channel: Scalable and Efficient Scheme in Zerocash. *IACR Cryptology ePrint Archive* 2017 (2017), 684. http://eprint.iacr.org/2017/684