Towards K-Anonymous Payments in Ethereum

Francesco Buccafurri[1,*,†], Vincenzo De Angelis[2,†], Sara Lazzaro[1,†] and Giorgia Sigilli[3,†]

[1]University Mediterranea of Reggio Calabria, Via dell'Università 25,89124, Reggio Calabria, Italy
[2]University of Calabria, Via P.Bucci, 87036 Arcavacata di Rende (CS), Italy
[3]IMT School for Advanced Studies Lucca, Lucca, Italy

Abstract

It is well known that pseudonymity in blockchains such as Bitcoin or Ethereum does not really hide the identity of the author of a payment. A number of attacks have been documented in the literature to deanonymize blockchain transactions. This is also the case of fully anonymous blockchains such as Monero and ZCash for which traffic analysis methods can be used. The goal of this paper is to outline a solution to the above problem in the most general (and difficult) case, namely that of pseudonymous blockchains. We restrict to the case of blockchains supporting smart contracts, with specific reference to Ethereum. Borrowing an approach used in the context of anonymous communication networks, we design a solution supporting k-anonymous payments against every eavesdropper, including the network adversary. Roughly, the idea is to organize users in rings of cover transactions, through which users indistinguishably exchange actual data or random noise and the initiator is hidden within the ring. Importantly, no off-chain communication is required.

Keywords

Blockchain, Anonymous Payments, Traffic Analysis

1. Introduction

Blockchain is a distributed ledger that keeps track of the occurrence of events. An entity can generate a transaction toward another entity to exchange a value. This transaction is validated by peers participating in the network, and thus does not require any third-trusted party to be validated.

A relevant feature offered by the most known blockchains (e.g., Ethereum and Bitcoin) is pseudonymity. Each user is associated with an address (not directly linked to the real identity of the user) that allows them to send and receive cryptocurrency. Nevertheless, all the transactions a user makes with the same address are linked among them. In the literature, several works were proposed concerning the de-anonymization of blockchain addresses [1] also in the case a single user leverages multiple addresses [2].

Different blockchains such as Monero and ZCash offer full anonymity [3] in place of pseudonymity, by making the transactions made by the same user unlinkable to each other. However, as shown in [4], effective de-anonymization attacks based on network analysis can be

ITASEC'24: Italian Conference on Cybersecurity, April 08–12, 2024, Salerno, Italy
*Corresponding author.
†These authors contributed equally.
bucca@unirc.it (F. Buccafurri); vincenzo.deangelis@dimes.unical.it (V. De Angelis); sara.lazzaro@unirc.it (S. Lazzaro); giorgia.sigilli@imtlucca.it (G. Sigilli)
0000-0003-0448-8464 (F. Buccafurri); 0000-0001-9731-3641 (V. De Angelis); 0000-0002-0846-4980 (S. Lazzaro)

performed even against anonymous blockchains. On the other hand, the problem of anonymous payments in pseudonymous blockchains is relevant per se, even if a solution was found in a fully anonymous blockchain. Indeed, pseudonymous blockchains play a dominant role in the blockchain landscape and their cryptocurrencies are often widely preferred.

As a matter of fact, transactions generate traffic in the network. Therefore, no resistance against the network adversary playing as a global eavesdropper can be achieved if this aspect is not taken into account. Obviously, the trivial idea of interposing an anonymous routing protocol (even resistant to the global observer) between the client and the pseudonymous blockchain cannot work, because the goal is not to hide the network address of the client but their blockchain address, which would be anyway public.

However, anonymous communication networks [5] can be a reference from which to draw inspiration, giving the flow of transactions the role that traffic has in networks. This is the way we follow in this paper, in which we address the problem in pseudonymous blockchains supporting smart contracts. Specifically, we refer to Ethereum [6]. In this preliminary work, we propose a solution to achieve anonymity guarantees in pseudonymous blockchains resisting traffic analysis attacks. We aim to hide the sender activity (i.e., the fact that a user generates a transaction) in an anonymity set of k users also against a global adversary observing the entire traffic exchanged in the network (even at the network layer). Specifically, we organize users in rings of cover transactions, through which actual data or random noise are indistinguishably exchanged so that the initiator of a payment is hidden within the ring. Importantly, our approach does not require off-chain communication channels.

It is worth noting that our solution has completely ethical goals. Indeed, we are aware that anonymity in payments is often pursued by the crime. However, this is not the case of our solution, in which anonymous payments are allowed only to registered services and, then, potentially subject to verification.

This paper is a work-in-progress paper. Therefore, the proposal is only sketched and some components (also regarding the security analysis) still need to be processed. Moreover, even though the design of the solution is done on the basis of the features offered by Ethereum and its smart contracts [7], no implementation is included in this preliminary paper.

The structure of the paper is the following. In section 2, we investigate the related literature. We describe the proposed solution in Section 3. Its security is analyzed in Section 4. Finally, in Section 5, we draw our conclusion.

2. Related work

In this section, we provide a brief overview of the literature related to our work.

Our goal is to implement payments on Ethereum ensuring k-anonymity for the payment initiator, making them undetectable even by network adversaries with uncertainty less than $1/k$, where k represents a parameter for the desired degree of anonymity.

Transaction anonymity could certainly be achieved using anonymous blockchains like ZCash [8] and Monero [9]. However, even in these cases, the transaction initiator can be detected through traffic analysis [4]. Various studies have been presented in the literature regarding de-anonymization of blockchain addresses, as highlighted in [1], even when a single user utilizes

multiple addresses, as discussed in [2].

In pseudonymous blockchains, several approaches have been proposed, typically based on transaction mixing, to make transactions untraceable, such as [3, 10, 11, 12].

In particular, [3] discusses the challenge of achieving anonymity in blockchain transactions, examines threats, surveys anonymity-provisioning methods, and proposes guidelines for effective blockchain-based e-cash systems and future research directions.

[10] delves into resolving privacy concerns stemming from Bitcoin transparent nature by introducing concepts of "micropayment channels", which enable secure, immediate, and confidential transactions. It addresses storage limitations, facilitates transactions involving untrusted intermediaries, and showcases its viability in currencies.

[11] explores the balance between the trustless aspect of decentralized currencies and the compromised privacy of transactions. It proposes enhancing regulatory compliance and tracking of tainted coins by introducing privacy-preserving mechanisms.

[12] proposes an Ethereum solution to provide anonymous services with accountability guarantees.

However, no one of the above proposals considers the threat model of a global (network) adversary.

Other works that combine blockchain and k-anonymity do not address the execution of anonymous payments, namely [13] and [14]. These solutions utilize blockchain as a framework for data management and leverage k-anonymity for anonymous data sending and exchanging. Despite this, these works do not achieve the goal of our work.

As highlighted in the introduction, we borrow from the context of Anonymous Communication Networks (ACNs) [5] the idea of using cover traffic in P2P overlay routing methods to obtain unobservability of the sender. In our case, the counterpart of the traffic is the flow of transactions.

As observed in [15], the existing routing methods for P2P overlay networks, which resist the global passive adversary [16], mostly require the inclusion of cover traffic [17, 5, 18, 19] to conceal their operation. In alternative to secure-multi-party-based protocols, like DC-nets [20], the inclusion of cover traffic is needed to obtain protection against the global adversary [21].

There are two primary strategies that utilize cover traffic. The first approach, known as "buses" [22, 23, 24], involves the sender following a predefined route to anonymously communicate with the destination. These works introduce innovative protocols for anonymous communication, drawing inspiration from public transportation systems. They aim to conceal traffic patterns and ensure message anonymity, offering either deterministic or randomized approaches. The second approach, known as "mixnets" [25, 26], generally offers lower latency but requires more cover traffic. In particular, the approach presented in [26] is burdened by substantial communication overhead. [27] discusses challenges in onion mixnets within anonymous communication networks, to obtain sender anonymity against the global adversary. Also [**?**] deals with anonymous communications, emphasizing privacy in public networks. The study also examines verifiable mixnets and their applicability to electronic voting, concluding with a brief mention of other anonymity-based systems.

The idea of our work is to use cover transactions instead of cover traffic. However, a mixnet-based solution would not be feasible due to the large amount of cover traffic required [19]. For these reasons, our work is based on the buses approach.

[28, 29] present some similarities with our approach since they deal with the idea of flow security. The focus is the examination of security attributes within computer systems, in particular enduring properties. These approaches denote that if a system maintains security at any given instance, subsequent states also retain security. Information flow security properties have emerged as a fundamental mechanism for upholding the confidentiality of classified information. These properties serve to manage the transfer of information among distinct groups of entities characterized by differing security levels.

Their goal differs from ours, as they may prioritize different objectives. Consequently, the methodologies they employ may not align with our specific needs, so their methods are not directly applicable in our context.

3. The Proposed Approach

Our approach enables users to transfer money in an anonymous way in a pseudonymous blockchain. In addition, our solution does not require any off-chain interaction, thus all the operations are performed through the blockchain.

3.1. Overview

The core concept of this solution involves grouping users into sets named *rings* (as illustrated in Figure 1). A ring serves as an anonymity set, ensuring that when a user within a ring initiates a payment, their identity remains indistinguishable from other ring users. The solution incorporates an information-hiding mechanism within the ring, achieved through the continuous circulation of *cover transactions*.

These cover transactions are blockchain transactions moving sequentially between adjacent ring users. They may carry either actual or dummy data. These data are probabilistically encrypted by a user with the Ethereum public key of their next in the ring. This way, no external party can tell whether a given transaction is carrying dummy or actual data.

At a broader level, the solution utilizes a smart contract where users initially deposit an equal amount of currency. This deposited sum can be spent later in an anonymous way. Each ring user is linked to a pseudonym, distinct from their Ethereum address. Users maintain private ledgers that associate pseudonyms with balances. However, the mapping between pseudonyms and Ethereum addresses is unknown to the users.

To execute an anonymous payment, a user awaits a cover transaction containing dummy data, which is then replaced with actual data detailing the intended payment (e.g., pseudonym of the sender, recipient, amount). This transaction is forwarded in the ring hop by hop so that, at the end, its content is shared with all the users. Given the pseudonymous, each user in the ring updates their private ledger (i.e., decrements the balance of the pseudonymous of a given quantity). In this manner, each user in the ring is aware that a particular pseudonym has initiated a payment but does not know the corresponding Ethereum address.

Ultimately, the smart contract executes the actual payment after receiving a specified number of confirmations from ring users.

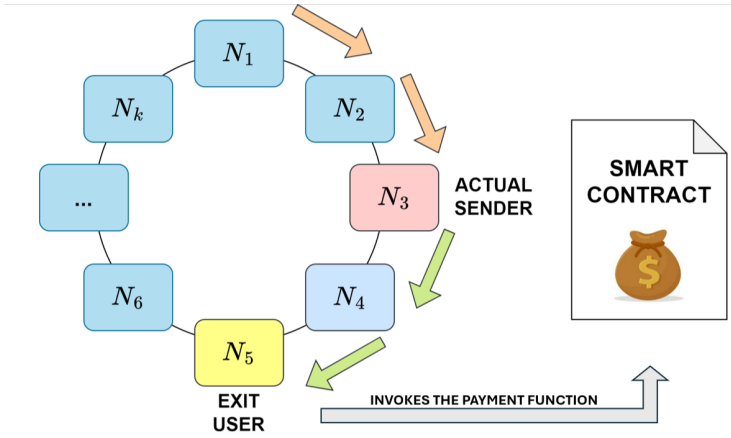

Figure 1: Ring constructed by N users.

3.2. Realm Construction

The main idea of our solution revolves around arranging a collection of users into a cyclically ordered set, named ring, as depicted in Figure 1. A set of rings is called *realm*. The realm is managed through a smart contract, say SC_{root}. SC_{root} has three main goals: (1) management of the users willing to join the system (potential senders of anonymous payments), (2) building of the rings from the realm, and (3) management of the service providers (potential recipients of anonymous payments).

We now describe how SC_{root} reaches goal (1). To enter a realm, each user i should deposit a sum of cryptocurrency $S_i = w_i + d_i + f_i$ in the smart contract SC_{root}. w_i is the amount that can be later on spent in an anonymous way. d_i is a security deposit that serves as an incentive for users to collaborate in the ring. f_i stands for the fee necessary to cover the management of each ring.

Each ring is composed of $k + \alpha$ users. k is a parameter defining the anonymity degree of the system, and α is the assumed maximum number of non-collaborating users in a ring. The meaning of α and k is that, when at most α users do not collaborate, we can obtain that the initiator of a payment is still indistinguishable among k users against a network eavesdropper (thus, obtaining k-anonymous payments).

Now, we describe how the rings of a realm are built, i.e., goal (2). The ring construction should happen in a fully decentralized way, such that no off-chain interaction is required. To reach this goal, we propose a mechanism based on a *Distributed Hash Table* (DHT) that allows the users to find the ring to which they belong, through their Ethereum address. However, if this computation depended only on the Ethereum address of each user, another problem would arise. Indeed, for an anonymous payment to be executed, our solution requires at least t confirmations among the $k + \alpha$ users in a ring. Then, an adversary could perform a sibyl attack by generating a lot of Ethereum addresses to find at least t addresses that would belong to the same ring. This way, they would be able to control the ring and spend the cryptocurrencies of the other users.

We recall that each user joining the system should pay an initial sum of cryptocurrency. This

also works as a disincentive for an adversary to try joining the system with multiple Ethereum addresses. Moreover, if users were not able to precompute the ring in which they would fall, the adversary would have to generate (and pay for) a greater number of addresses to increase their chance of having at least t addresses in the same ring. Then, the economic effort required from the adversary would be high.

To achieve this, we propose the following mechanism. Suppose the realm includes n users being split into rings of size $k + \alpha$. As above mentioned, each user deposits a given amount of cryptocurrency in the smart contract SC_{root} to become part of a ring. To do this, the users simply invoke a function of SC_{root} that collects their deposits and stores their Ethereum addresses. However, the rings are not formed until the n^{th} user asks to join a ring. When the n^{th} user joins the realm, the invoked function performs in a slightly different way. After collecting the deposit and storing the Ethereum address of the n^{th} user, the function retrieves and stores the height of the last block included in the blockchain. The idea is to use the hash of a future block whose distance is a fixed number from the stored one. Thus its hash serves as an unpredictable value to implement the DHT and prevent any adversary from precomputing the rings to which the users belong.

Specifically, when the block including the last transaction is mined and added to the blockchain, any entity can invoke another function that retrieves the digest of such a block, say B. Then, for each user u with Ethereum address Eth_u, a value $r_u = H(Eth_u||B)$ is computed and associated with Eth_u, where H denotes a cryptographic hash function. Then, these values are ordered in an increasing way to form the rings. Specifically, the first $k + \alpha$ values form the first ring, the next $k + \alpha$ values form the second ring, and so on. Observe that, since B is not known before the last user joins the system, the users cannot precompute in advance the rings they fall.

Ultimately, each ring is created by deploying a smart contract (per ring), say SC_R, in the blockchain. SC_R is responsible for the execution of anonymous payments. A thorough explanation of how these payments can be issued is given in Section 3.3.

We now explain how SC_{root} handles the service providers willing to be potential recipients of the anonymous payments, i.e., goal (3). These providers are the sole entities that can receive anonymous payments. In the system, there are a number of Attribute Authorities that can dynamically add to the system Service Providers. Attribute Authorities are responsible for the verification of the requirements of the Service Providers to belong to the system.

3.3. Anonymous payments

In this section, we describe the anonymous payment mechanism in a ring. This mechanism is handled via the smart contract SC_R associated with the ring.

Consider a user u willing to initiate an anonymous payment of s Ethers to a service provider SP. First, u should wait for a cover transaction, containing dummy data, from their previous user in the ring. Then, u should replace the dummy data with the actual data $p = \langle PK_u, SP, s, ID_p, Eth_{u'}, \sigma_u \rangle$, wherein ID_p is a random identifier for data p, and $Eth_{u'}$ represents the Ethereum address of another randomly chosen user u' in the ring, referred to as the *exit-user*. σ_u is a signature computed over $\langle PK_u, SP, s, ID_p, Eth_{u'} \rangle$ verifiable using PK_u, i.e., a public key serving as pseudonymous for u.

Then, p is encrypted with the Ethereum public key of the user succeeding u in the ring and transmitted via a cover transaction. We recall that no external party should be able to state if a cover transaction is carrying dummy data or actual data.

The cover transaction completes a full loop in the ring, with all users locally storing its data and verifying the signature σ_u using PK_u. Upon reaching u again, the same cover transaction is sent to the next user in the ring. This process repeats for a second loop until it reaches the exit-user, say u'. At this point, u' empties the cover transaction, i.e., replaces the actual data within such a transaction with dummy data.

In the meanwhile, each user in the ring tosses a biased coin, and then with probability $\frac{t}{k}$, each user invokes a function of SC_R to authorize the payment passing $\langle ID_p, SP, s\rangle$ as input.

When the smart contract SC_R receives at least t authorizations, it transfers the amount s from its local balance to SP. Here, t ($\frac{k}{2} + 1 \leq t \leq k$) is a system parameter serving as a safeguard against malicious or non-cooperating users attempting to obstruct a legitimate payment.

Finally, all users in the ring update the balance associated with PK_u in their internal ledger.

Figure 1 shows a graphical representation of this procedure.

3.4. Participant incentives

Our solution requires that each sender of payment should be at least k-anonymous, i.e., they should be disguised among at least k other users in the ring. Thus, given that the ring is composed of $k + \alpha$ users, in a ring there may be at most α non-collaborating users, otherwise the anonymity threshold k cannot be satisfied. We recall that the amount w_i of cryptocurrency (initially deposited) by each ring user i is fixed in advance. So if i depletes the amount w_i before other users in the ring, they will not be able to make additional anonymous payments but must continue to collaborate to facilitate the anonymous payments of other users within the ring. Thus this user may be disincentivised to further collaborate in the ring, since collaborating implies paying for making the cover transaction circulating in the ring and sending confirmations to the smart contract. To prevent this, we introduce a *security deposit mechanism*. Each user i, in joining the system, must deposit an amount d_i. This deposit will be fully refunded to the users who collaborate until the end of the ring. A huge deposit d_i would certainly make users collaborate till the end. However, this is little realistic. Similarly, a small deposit d_i may not be enough to incentivize users. To estimate a plausible value for d_i, we first need to estimate the ring lifetime and thus the maximum amount of expenses necessary to carry out the anonymous payments until the end of the ring. To do so, we split the ring lifetime into epochs. We denote by e an epoch of the ring and by D^e the sum of all the deposits from ring users left in the epoch e. In the first epoch (say $\hat{e}$), $D^{\hat{e}} = \sum_{i=1}^{(k+\alpha)} d_i$. Then, we set a minimum threshold p_{min} for each payment that, as we will see, does not prevent the user from paying less than p_{min}. Since we aim to guarantee service continuity for all the users, an epoch ends once a user i, after payments, terminates the amount w_i^e. Thus, the maximum number of payments in an epoch e is given by $\frac{\sum_{i=1}^{(k+\alpha)} w_i^e}{p_{min}}$, where w_i^e is the amount of cryptocurrency for user i (initially set to w_i) left for the anonymous payments at the epoch e.

Observe that, since in general the amount of cryptocurrency that a user is willing to pay to a

service provider may not be an integer multiple of p_{min}, each payment may generate a change. We say c the entire amount of changes. At the end of each epoch, the changes will be returned to their owner in the ring (i.e., each user in the ring updates their private ledger accordingly). Observe that, an epoch e ends once at least a user i terminates the amount w_i^e but still owns a fraction of the changes c. Once e is ended, we compute the amount of deposit necessary to advance the next epoch, say e'. The idea is that, at the end of the epoch e, we can compute the amount of deposit $D^{e'}$ necessary to cover (in the worst case) the expenses in the subsequent epoch e'. Thus it will depend on the maximum number of payments that will be done during e'. Again, this number can be estimated according to the following formula: $\frac{\sum_{i=1}^{(k+\alpha)} w_i^{e'}}{p_{min}}$.

Then the differences between the deposit in the current epoch (D^e) and the deposit necessary for the next epoch ($D^{e'}$) will be given to each user ring. To incentivize collaboration among users in the ring, a portion of the deposit will be returned to ring users according to a score associated with how collaborative they were during each epoch. Specifically, given a user i, the amount of deposit they will receive back is given by the following formula: $\frac{g_{i,e} \cdot (D^e - D^{e'})}{\sum_{t=1}^{k+\alpha} g_{t,e}}$, where $g_{i,e}$ is the score associated with i during epoch e. The computation of the score is left as future work.

Observe that, since the end of an epoch happens when at least one user makes a payment, it is always guaranteed that $D^{e'} < D^e$. So the process converges.

4. Security Analysis

In this section, we analyze the security of our solution. Our analysis is founded on a basic assumption, which we call **A1**, that at most α users are not collaborative in each ring. It is justified by the proposed incentive mechanism. Indeed, when users do not collaborate, they lose part of all of their security deposit. In addition, also the wallet amount is lost when a user does not collaborate. Indeed, we recall, that the payments are intended only to registered service providers. Then, users cannot transfer their remaining amount to another wallet under their control.

Adversary Model. We consider a global adversary with the capability of capturing all the network traffic exchanged by the users.

Observe that, in the blockchain network, any participant can observe the transactions performed by all users. However, all participants but those directly connected to the sender, do not see the IP address of the originator of a transaction. Our adversary is then much more powerful than standard users. In practice, it might correspond with one or more Internet Service Providers.

Security Goal. Our solution achieves *sender anonymity* [30]: the adversary cannot identify the initiator of a payment (sender) with probability greater than $\frac{1}{k}$.

To show this, we follow an approach similar to [19].

By Assumption **A1**, we consider the worst-case scenario in which in the ring of the sender there are k collaborative users.

When a cover transaction is forwarded hop-by-hop, thanks to probabilistic encryption, the adversary cannot distinguish if it contains dummy data or the details of a payment. Then, the

adversary cannot identify the sender when they fill the cover transaction with actual data.

Therefore, the only way for the adversary to identify the sender is to detect a possible transition from actual/dummy data or dummy/actual data of a cover transaction at another point of the ring and try to draw some information starting from this observation.

The only point of the ring from which the adversary can draw such information is the exit user.

We have to consider two cases.

The first case is when the cover transaction including the details of the payment reaches the exit user. In this case, they send the first confirmation of the payment towards the smart contract. The adversary can observe this confirmation to infer that a payment is performed. However, since before reaching the exit user, the cover transaction containing the details of the payment, performs at least a complete loop of the ring, these details may be inserted by any of the k collaborative users in the ring.

Thus, in this case, the sender cannot be identified with a probability greater than $\frac{1}{k}$

The second case occurs after all the confirmations are sent to the smart contract. In this case, the exit user inserts dummy data in the cover transaction so that it can be used for the next payment. This can be observed by the adversary.

However, the next observation of the adversary is again when another exit user sends the first confirmation to the smart contract. This cannot happen before a complete loop of the ring is performed. Then, we are again in the first case and the sender cannot be identified with probability greater than $\frac{1}{k}$.

Another possible exploitable information for the attacker could be the confirmations. Indeed, the attacker could guess that the initiator of the payment is within (or without) the set of users sending the confirmation. This, in principle, could reduce the size of the anonymity set below the value k. However, this is not the case because every user sends the confirmation with probability $\frac{t}{k}$. Therefore, no bias occurs and no guess is possible for the attacker.

5. Conclusion

In this preliminary paper, we present a solution to achieve k-anonymity in Ether payments to allowed registered services. Anonymity is reached in the most severe threat model, namely a global observer also capable of analyzing network traffic. The solution needs to be further processed (in some components) and implemented, also to test its feasibility in terms of costs. Indeed, a critical aspect to be analyzed is the cost of transactions and execution of smart contracts. However, from a first rough analysis not included in this paper for its incompleteness, costs appear unproblematic even for appreciable anonymity degrees.

A possible criticism of our solution could regard its plausibility in terms of customer acceptance and expected benefits. About this, we observe that our solution is not different from existing closed-loop prepaid payment systems in the domain of standard currencies. Moreover, prepaid systems in which money withdrawal is not allowed also exist, based on cards, phone credit or other kinds of electronic wallets. The benefits of our solution can be found in the ever-increasing demand for privacy that, sometimes, intersects with censorship resistance.

As future work, we plan to explore all the above aspects in depth, addressing both technical

and business issues. Another possible future work is to study the migration of our approach, thought for payments in cryptocurrencies, to permissioned blockchains such as Hyperledger Fabric [31]. This issue is not trivial, because payments should be implemented by fungible tokens and, thus, we should investigate how to preserve the anonymity features of our solution when using tokens.

Acknowledgments

This work is partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

[1] H. H. Sun Yin, K. Langenheldt, M. Harlev, R. R. Mukkamala, R. Vatrapu, Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the bitcoin blockchain, Journal of Management Information Systems 36 (2019) 37–73. doi:10.1080/07421222.2018.1550550.

[2] D. Ermilov, M. Panov, Y. Yanovich, Automatic bitcoin address clustering, in: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017, pp. 461–466. doi:10.1109/ICMLA.2017.0-118.

[3] N. Andola, Raghav, V. K. Yadav, S. Venkatesan, S. Verma, Anonymity on blockchain based e-cash protocols—a survey, Computer Science Review 40 (2021) 100394. URL: https://www.sciencedirect.com/science/article/pii/S1574013721000344. doi:https://doi.org/10.1016/j.cosrev.2021.100394.

[4] A. Biryukov, S. Tikhomirov, Deanonymization and linkability of cryptocurrency transactions based on network analysis, in: 2019 IEEE European Symposium on Security and Privacy (EuroS&P), 2019, pp. 172–184. doi:10.1109/EuroSP.2019.00022.

[5] F. Shirazi, M. Simeonovski, M. R. Asghar, M. Backes, C. Diaz, A survey on routing in anonymous communication protocols, ACM Comput. Surv. 51 (2018). URL: https://doi.org/10.1145/3182658. doi:10.1145/3182658.

[6] S. Tikhomirov, Ethereum: State of knowledge and research perspectives, in: A. Imine, J. M. Fernandez, J.-Y. Marion, L. Logrippo, J. Garcia-Alfaro (Eds.), Foundations and Practice of Security, Springer International Publishing, Cham, 2018, pp. 206–221. doi:https://doi.org/10.1007/978-3-319-75650-9_14.

[7] F. Buccafurri, V. De Angelis, G. Lax, L. Musarella, A. Russo, An attribute-based privacy-preserving ethereum solution for service delivery with accountability requirements, in: Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES '19, Association for Computing Machinery, New York, NY, USA, 2019. URL: https://doi.org/10.1145/3339252.3339279. doi:10.1145/3339252.3339279.

[8] G. Kappos, H. Yousaf, M. Maller, S. Meiklejohn, An empirical analysis of anonymity in zcash, in: 27th USENIX Security Symposium (USENIX Security 18), 2018, pp. 463–477.

[9] Y. Li, G. Yang, W. Susilo, Y. Yu, M. H. Au, D. Liu, Traceable monero: Anonymous cryp-

tocurrency with enhanced accountability, IEEE Transactions on Dependable and Secure Computing 18 (2021) 679–691. doi:10.1109/TDSC.2019.2910058.

[10] M. Green, I. Miers, Bolt: Anonymous payment channels for decentralized currencies, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 473–489. URL: https://doi.org/10.1145/3133956.3134093. doi:10.1145/3133956.3134093.

[11] C. Garman, M. Green, I. Miers, Accountable privacy for decentralized anonymous payments, in: Financial Cryptography and Data Security: 20th International Conference, FC 2016, Christ Church, Barbados, February 22–26, 2016, Revised Selected Papers 20, Springer, 2017, pp. 81–98. doi:https://doi.org/10.1007/978-3-662-54970-4_5.

[12] F. Buccafurri, V. De Angelis, S. Lazzaro, A blockchain-based framework to enhance anonymous services with accountability guarantees, Future Internet 14 (2022). URL: https://www.mdpi.com/1999-5903/14/8/243. doi:10.3390/fi14080243.

[13] Y. Long, Y. Chen, W. Ren, H. Dou, N. N. Xiong, Depet: A decentralized privacy-preserving energy trading scheme for vehicular energy network via blockchain and k - anonymity, IEEE Access 8 (2020) 192587–192596. doi:10.1109/ACCESS.2020.3030241.

[14] B. Sowmiya, E. Poovammal, A heuristic k-anonymity based privacy preserving for student management hyperledger fabric blockchain, Wireless Personal Communications 127 (2022) 1359–1376. doi:https://doi.org/10.1007/s11277-021-08582-1.

[15] F. Buccafurri, V. de Angelis, S. Lazzaro, Mqtt-a: A broker-bridging p2p architecture to achieve anonymity in mqtt, IEEE Internet of Things Journal 10 (2023) 15443–15463. doi:10.1109/JIOT.2023.3264019.

[16] F. Buccafurri, V. D. Angelis, M. Francesca Idone, C. Labrini, Wip: An onion-based routing protocol strengthening anonymity, in: 2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2021, pp. 231–235. doi:10.1109/WoWMoM51794.2021.00041.

[17] T. Grube, M. Thummerer, J. Daubert, M. Mühlhäuser, Cover traffic: A trade of anonymity and efficiency, in: Security and Trust Management: 13th International Workshop, STM 2017, Oslo, Norway, September 14–15, 2017, Proceedings 13, Springer, 2017, pp. 213–223. doi:https://doi.org/10.1007/978-3-319-68063-7_15.

[18] F. Buccafurri, V. De Angelis, M. F. Idone, C. Labrini, S. Lazzaro, Achieving sender anonymity in tor against the global passive adversary, Applied Sciences 12 (2022). URL: https://www.mdpi.com/2076-3417/12/1/137. doi:10.3390/app12010137.

[19] F. Buccafurri, V. De Angelis, M. F. Idone, C. Labrini, A protocol for anonymous short communications in social networks and its application to proximity-based services, Online Social Networks and Media 31 (2022) 100221. URL: https://www.sciencedirect.com/science/article/pii/S2468696422000258. doi:https://doi.org/10.1016/j.osnem.2022.100221.

[20] M. R. Nosouhi, S. Yu, K. Sood, M. Grobler, Hsdc–net: Secure anonymous messaging in online social networks, in: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2019, pp. 350–357. doi:10.1109/TrustCom/BigDataSE.2019.00054.

[21] G. Danezis, C. Diaz, A Survey of Anonymous Communication Channels, Technical Report MSR-TR-2008-35, 2008. URL: https://www.microsoft.com/en-us/research/publication/

a-survey-of-anonymous-communication-channels/.

[22] Beimel, Dolev, Buses for anonymous message delivery, Journal of Cryptology 16 (2003) 25–39. doi:https://doi.org/10.1007/s00145-002-0128-6.

[23] A. Hirt, M. Jacobson, C. Williamson, Taxis: Scalable strong anonymous communication, in: 2008 IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008, pp. 1–10. doi:10.1109/MASCOT.2008.4770566.

[24] A. L. Young, M. Yung, The drunk motorcyclist protocol for anonymous communication, in: 2014 IEEE Conference on Communications and Network Security, 2014, pp. 157–165. doi:10.1109/CNS.2014.6997482.

[25] I. Ben Guirat, D. Gosain, C. Diaz, Mixim: Mixnet design decisions and empirical evaluation, in: Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society, WPES '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 33–37. URL: https://doi.org/10.1145/3463676.3485613. doi:10.1145/3463676.3485613.

[26] P. Kotzanikolaou, G. Chatzisofroniou, M. Burmester, Broadcast anonymous routing (bar): scalable real-time anonymous communication, International Journal of Information Security 16 (2017) 313–326. doi:https://doi.org/10.1007/s10207-016-0318-0.

[27] Y. Xia, R. Chen, J. Su, H. Zou, Balancing anonymity and resilience in anonymous communication networks, Computers & Security 101 (2021) 102106. URL: https://www.sciencedirect.com/science/article/pii/S0167404820303795. doi:https://doi.org/10.1016/j.cose.2020.102106.

[28] A. Bossi, R. Focardi, D. Macedonio, C. Piazza, S. Rossi, Unwinding in information flow security, Electron. Notes Theor. Comput. Sci. 99 (2004) 127–154. URL: https://doi.org/10.1016/j.entcs.2004.02.006. doi:10.1016/j.entcs.2004.02.006.

[29] A. Bossi, R. Focardi, C. Piazza, S. Rossi, Verifying persistent security properties, Comput. Lang. Syst. Struct. 30 (2004) 231–258. URL: https://doi.org/10.1016/j.cl.2004.02.005. doi:10.1016/j.cl.2004.02.005.

[30] A. Pfitzmann, M. Hansen, A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, 2010. URL: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, v0.34.

[31] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, J. Yellick, Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, EuroSys '18, Association for Computing Machinery, New York, NY, USA, 2018. URL: https://doi.org/10.1145/3190508.3190538. doi:10.1145/3190508.3190538.