EPiC Series in Computing

Volume 97, 2024, Pages 91–100

Proceedings of 36th International Conference on Computer Applications in Industry and Engineering

Establishing Trust using Zero Knowledge Succinct Proof in Peer-to-peer Data Transfer*

Sai Kiran Deverasetti[1], Anjila Neupane[1], Indranil Roy[1], Reshmi Mitra[1] and Bidyut Gupta[2]

[1] Southeast Missouri State University, Cape Girardeau, USA
{sdeverasetti1s, aneupane4s, iroy, rmitra}@semo.edu
[2] Southern Illinois University, Carbondale, USA
bidyut@cs.siu.edu

Abstract

This paper presents a cryptographic solution for establishing trust in peer-to-peer (P2P) networks, addressing issues of privacy, performance, and anonymity. Our protocol utilizes Zero-Knowledge Proofs (ZKP) for continuous trust validation during data transfers. This procedure compels each node to continually demonstrate its integrity, significantly decreasing the potential for network attacks. Upon evaluation, the protocol proved to be highly scalable and efficient, expanding network reach without requiring additional control messages. This result validates the protocol's robustness, suggesting its potential use in larger and more intricate P2P network architectures.

1 Introduction

Peer-to-peer (P2P) distributed trust verification is a complex problem [1] because of the lack of centralized authority to manage the frequent joining and leaving of nodes. There are significant trade-off issues due to security, performance, and cost. As the number of nodes increases, there can be a significant burden on the network to mark each node as legitimate. Secondly, P2P networks are susceptible to catastrophic attacks such as sybil, replay, and eclipse attacks where the adversary creates multiple malicious peers. Moreover, these networks have limited resources in terms of processing and bandwidth which makes it difficult to implement a resource-intensive protocol for mitigation and defense. Due to these factors, it becomes extremely challenging to incorporate traditional Public-key Infrastructure (PKI). Overall, the distributed network requires the participants to be responsible for authenticating, each other in a secure, efficient, and scalable manner.

The struggle of balancing *anonymity vs trust* presents a significant challenge in P2P networks [17]. Anonymity, a feature highly valued for privacy reasons, allows users to interact without revealing their identities or sensitive data, thereby protecting them from surveillance, censorship, and targeted attacks. However, this level of anonymity can also provide cover for malicious actors to perform harmful activities without traceability. Trust, conversely, necessitates some form of identity verification or past

*Corresponding author: rmitra@semo.edu

behavior tracking, and thereby, potentially infringes on the users' anonymity. A highly anonymous network complicates the establishment of trust, leading to an escalation in network-level attacks such as spamming or spreading of false information, as holding malicious nodes accountable becomes increasingly challenging. Conversely, an emphasis on trust often requires reputation systems or identity verification, potentially compromising user anonymity. Thus, striking the right balance between these two conflicting requirements poses a complex problem in P2P networks.

Lightweight cryptography using *Zero Knowledge Proofs (ZKP)* offers a compelling solution to address the above challenges. ZKP allows a prover to demonstrate to a verifier that a certain statement such as the legitimacy of a node is true, without revealing any information beyond the truth of the statement itself. This provides a mechanism for establishing trust while preserving user anonymity which is a challenging balancing act in P2P network. Moreover, ZKPs are beneficial in minimizing computational overhead and communication costs, essential for the effective operation of resource-constrained P2P networks. Furthermore, the unique aspect of ZKPs is to provide non-interactive proofs that are succinct and efficient, requiring minimum interaction between prover and verifier, reducing network latency, and speeding up the data transfer process.

Our solution proposes a light, secure, and effective ZKP-based protocol for establishing and maintaining trust among anonymous peers/group heads during the data transfer phase in a P2P network. The sender group head initiates the communication by proposing a one-step challenge (such as calculating SHA of a random number) to the receiver node. On receiving the correct unique solution, the sender uses it as a nonce to encrypt the message and send it to the receiver. Only the receiver node can decrypt the message since it owns the decryption key. Essentially, it implements a symmetric encryption scheme and a safe key transfer process. The sender node can quickly formulate a different version of the compact problem statement by changing the mathematical operation or random number, without compromising any sensitive information. Furthermore, the protocol can be repeated multiple times without any third-party interference till the verifier is convinced about the trustworthiness of the prover.

Receiver nodes that either provide an incorrect response to the sender's challenge or fail to respond within the predetermined timeout period are designated as 'corrupt'. Once marked as corrupt, these nodes are effectively disallowed from engaging in future data transfer phases, serving as a penalty and a deterrent for non-compliance with the protocol. This barring mechanism aids in the preservation of the network's integrity, protecting it from nodes that could potentially compromise the system's functionality or security.

This protocol addresses the complex issue of distributed trust verification in P2P networks, striking a balance between user anonymity and trustworthiness of group head nodes. It allows for minimal interaction, reducing computational overhead and communication costs, hence making it ideal for resource-constrained P2P networks. Additionally, this approach proposes *continuous trust verification*, eliminating assumptions about inherent trustworthiness and subsequently reducing the likelihood of attacks. This makes our solution a robust and scalable method for trust verification, enhancing network security while preserving data privacy.

The main contributions of this paper are listed below:

- *Lightweight Cryptographic Solution*: designed a novel ZKP-based protocol for trust verification in P2P networks that preserves user anonymity.

- *Balancing Anonymity and Trust*: a one-step challenge and response mechanism during data transfer, striking a balance between user anonymity and trustworthiness.

- *Scalability and Resource Efficiency*: suitable for resource-constrained P2P networks, this protocol ensures scalable and efficient trust verification, minimizing computational overhead and communication costs.

This paper has four main sections. In Section 2, the summary of state-of-art trust verification in P2P network is presented. Section 4 is about zero-knowledge proof protocol design. Model evaluation and discussion is part of Section 5. We are concluding with the highlights of our work in Section 6.

2 Related Work

2.1 Trust Verification in P2P network

Trust management is a critical aspect of P2P networks, especially in the context of secure data transfer. Trust-based mechanisms play a significant role in ensuring data integrity, confidentiality, and availability in P2P networks. Balfe *et el.* proposed a trusted computing framework to enhance security in P2P networks. Their approach focuses on leveraging trusted computing technologies to establish a secure foundation for P2P interactions. By integrating trusted platform modules and secure boot mechanisms, the authors demonstrate the potential for providing secure communication and data exchange in P2P networks [1]. Selcuk *et el.* proposed a reputation-based trust management system specifically designed for P2P networks. Their approach utilizes the reputation of participating nodes as a trust metric to assess the reliability of peers. The authors emphasize the importance of reputation management in P2P networks and present a comprehensive framework that incorporates trust calculation and decision-making processes to enable effective trust evaluation [15].

Zhao *et el.* addressed the issue of result verification and trust-based scheduling in P2P grids. Their research focused on grid computing environments where resource sharing and collaboration occur among geographically distributed nodes. The authors proposed a trust-based scheduling algorithm that considers both the reputation of participating nodes and the accuracy of their reported results [19]. Frahat *et el.* presented a secure and scalable trust management model specifically tailored for IoT P2P networks. The authors identified the unique characteristics and security challenges of IoT environments and proposed a trust management framework that ensures secure communication and collaboration among IoT devices. Their model incorporates trust evaluation, reputation management, and access control mechanisms to establish a secure and trustworthy IoT P2P network [4]. Hao *et el.* focused on enhancing the trustworthiness of P2P networks through the integration of blockchain technology. The authors proposed a trust-enhanced blockchain P2P topology that enables fast and reliable broadcast of information. The integration of trust mechanisms further enhances the reliability and trustworthiness of the P2P network[14].

2.2 Preliminaries on Zero Knowledge Proofs

ZKPs have played a vital role in ensuring privacy and security P2P networks. Various research has been done to implement ZKP in P2P networks for secure communication. Danezis and Diaz introduced SybilInfer, a system that utilizes social network analysis to identify malicious entities. ZKPs are employed to enhance the trustworthiness of the detection process and improve the security of P2P networks [3]. Lu *et el.* explored the application of ZPKs for authentication in anonymous P2P networks. The paper focuses on the design and implementation of a pseudo-trust system using zero-knowledge authentication [10]. Pop *et el.* investigate the use of ZPKs to enhance privacy in energy transactions on the blockchain. The authors propose a scheme that ensures privacy while maintaining the integrity of energy-related transactions [13]. X Sun *et el.* provides an overview and analysis of the use of ZKPs in blockchain applications. The survey covers various aspects, including the different types of ZKPs used and their potential applications and challenges in the blockchain context [16].

Yang and Li present a digital identity management scheme using ZKPs in a blockchain setting. The paper proposes a secure and efficient method for managing digital identities while preserving privacy

[18]. Harikrishnan and Lakshmy explore the use of ZKPs for secure payments in distributed networks. The authors propose a scheme that ensures the confidentiality and integrity of digital service payments while maintaining anonymity [5]. Major *et el.* introduce an authentication protocol based on chaos theory and ZKPs. The paper presents a novel approach to enhance security and privacy in authentication protocols [12]. Kosba *et el.* present Hawk, a blockchain model that incorporates ZKPs for privacy-preserving smart contracts. The paper introduces a framework that enables efficient and secure execution of smart contracts while preserving user privacy [9]. Ben-Sasson *et el.* focus on the development of scalable ZKPs with no trusted setup. The paper introduces a construction called zk-STARKs, which provides a highly efficient and scalable approach to ZKPs [2].

These papers collectively contribute to the understanding and advancement of ZKPs in various domains, such as authentication, privacy in blockchain, digital identity management, secure payments, and smart contracts. They address different challenges and propose innovative solutions, demonstrating the versatility and potential applications of ZKPs in enhancing security and privacy in different contexts.

3 Preliminaries

Here, we have taken into consideration some of the first results of an RC-based low diameter two level hierarchical structured P2P network [6, 7, 11]. We provide a structured design for an interest-based peer-to-peer system in this section. We will use the following notations and their meanings to define the architecture.

Definition 1. We define a resource as a tuple $<Res_i, \mathrm{V}>$, where Res_i denotes the type of a resource and V is the value of the resource. Note that a resource can have many values.

Definition 2. Let S be the set of all peers in a peer-to-peer system. Then $S = \{P^{Ri}\}$, $0 \leq i \leq n-1$, where P^{Ri} denotes the subset consisting of all peers with the same resource type Res_i. and the number of distinct resource types present in the system is n. Also, for each subset P^{Ri}, we assume that H_i is the first peer among the peers in P^{Ri} to join the system. We call H_i as the group-head of group G_i formed by the peers in the subset P^{Ri}.

We now describe our proposed architecture suitable for interest-based peer-to-peer system. Generalization of the architecture is considered in [7]. We use the following notations along with their interpretations while we define the architecture.

3.1 Two Level Hierarchy

It is a two-level overlay architecture and at each level structured networks of peers exist. It is explained in detail below.

1. At level-1, we have a ring network consisting of the peers H_i $(0 \leq i \leq n-1)$. The number of peers on the ring is n which is also the number of distinct resource types. This ring network is used for efficient data lookup and so we name it as transit ring network.

2. At level-2, there are n numbers of completely connected networks (groups) of peers. Each such group, say G_i is formed by the peers of the subset P^{Ri}, $(0 \leq i \leq n-1)$, such that all peers $(\in P^{Ri})$ are directly connected (logically) to each other, resulting in the network diameter of 1. Each G_i is connected to the transit ring network via its group-head H_i.

3. Each peer in the network maintains a Information Resource Table (IRT) that consists of n number of tuples.

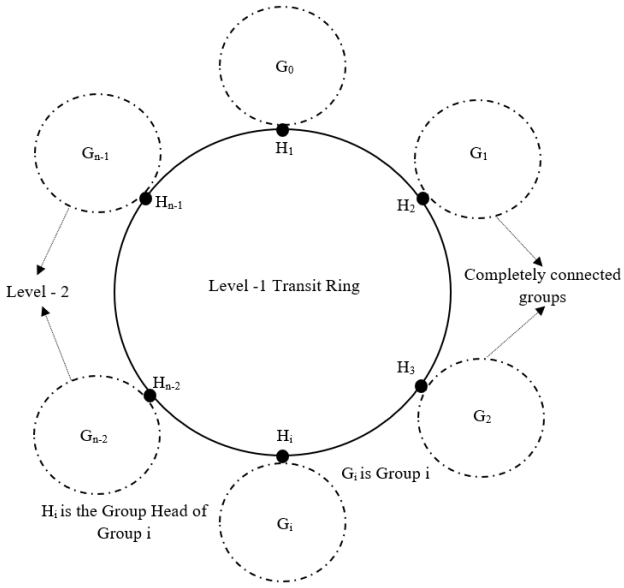

Figure 1: A two-level RC based structured P2P architecture with n distinct resource types

- The group heads will have a tuple of the form <Resource Type, Resource Code, Group Head Logical Address, Group Head public Key>for other group heads and <Resource Type, Resource Code, Peer Logical Address, Peer public Key>for the other peers present in the network. The Group Head Logical Address are assigned according to the proposed logical address assignment algorithm proposed in section 2.3 and the public key of the group heads or the peers are exchanged when they are joining the network and the IRT is updated and broadcasted in the network. Also, Resource Code is the same as the group head logical address.
- The peers P_i, who are not group heads but belongs to a group G_i ($P_i \in G_i$) will have the tuple of the form <Resource Type, Resource Code, Group Head Logical Address, Group Head public Key>for group head of G_i and <Resource Type, Resource Code, Peer Logical Address, Peer public Key>for the other peers present in G_i.

4. Any communication between a peer $G_{x,i} \in$ group G_x and $G_{y,j} \in$ group G_y takes place only through the corresponding group heads H_x and H_y.

The proposed architecture is shown in Figure 1. The assignment of the logical addresses is described in [6].

3.2 Salient Features of Overlay Architecture

We summarize the salient features of this architecture.

1. It is a hierarchical overlay network architecture consisting of two levels; at each level the network is a structured one.

2. Use of modular arithmetic allows a group-head address to be identical to the resource type owned by the group. We will show in the following section the benefit of this idea from the viewpoint of achieving reasonably very low search latency.

3. Number of peers on the ring is equal to the number of distinct resource types, unlike in existing distributed hash table-based works some of which use a ring network at the heart of their proposed architecture [8].

4. The transit ring network has the diameter of n/2. Note that in general in any P2P network, the total number of peers N >>n.

5. Each overlay network at level 2 is completely connected. That is, in graph theoretic term it is a complete graph consisting of the peers in the group. So, its diameter is just 1. Because of this smallest possible diameter (in terms of number of overlay hops) the architecture offers minimum search latency inside a group.

4 Protocol based on Zero Knowledge Proof

Let us consider a scenario where $P_i \in G_i$ with group-head G_i^h has requested for a resource Res_j. The data-lookup protocol defined in [6] is used to find the destination peer $P_j \in G_j$ with group-head G_j^h who has the required resource Res_j. After this the data transfer process between P_j and P_i takes place through the group-heads G_j^h and G_i^h. A potential vulnerability in the proposed network is the disruption of the group head, which can result in a single point of failure during a man-in-the-middle attack. The group head plays a crucial role in coordinating and facilitating communication within a specific group or subgroup of peers. If the group head is disrupted or compromised, it can severely impact inter-group communication and hinder the overall functionality of the network.

The designed protocol is for establishing and maintaining trust among anonymous peers/group heads in the RC-based P2P architecture during the *data transfer* phase is presented in algorithm 1. It is based on the cryptographic paradigm of *Zero Knowledge Proofs (ZKP)*. It allows a ***prover*** to demonstrate to a ***verifier*** that a specific statement is true without revealing anything about the statement itself or any other sensitive information. Furthermore, the protocol can be repeated multiple times without any third-party interference till the verifier is convinced about the trustworthiness of the prover as shown in Figure 2. Overall, it can provide a strong privacy guarantee and enables P2P network to scale to a large number of nodes without any decline in security and privacy.

Determining the eligibility of a node for participation within the P2P network is a systematic process and is handled by each group head. It meticulously scans through its *list of untrusted peers* Un_{list}. This list serves as a critical resource, containing all peers whose trustworthiness has not been established or is in question. Thus, the group head must regularly carry out this monitoring and update, because it determines the overall trustworthiness of the network. The dynamism of the network is reflected in the regular updates made to the list of untrusted peers. With each data transfer phase in the transit ring, there may be modifications to the list. This could involve the addition of new peers whose trustworthiness is yet to be proven, or the removal of those who have either demonstrated their trustworthiness or been found corrupt. When a message travels from the sender to the receiver, it typically traverses through several intermediate nodes. These nodes, during the course of the message transfer, are obligated to demonstrate their trustworthiness to their neighboring nodes using the ZKP protocol.

The next step is the *problem statement formulation* that enables the sender to identify clean versus corrupt intermediary receiver nodes. We are using *succinct proof* that are very compact and can be verified quickly, regardless of the complexity of the statement being proven or the amount of data involved. Hence, they require lower processing complexity w.r.t. classic NP verification. They are intentionally

Algorithm 1 ZKP Data Transfer Protocol between P_j and P_i

flag = **false** ▷ assume by default G_j^h is untrusted
P_j sends Res_j to it's group head G_j^h
for $k \leftarrow 1$ to 3 **do** ▷ we have considered k = 1, 2, 3 for this example
    if $G_i^h \in Un_{list}$ **then**
        G_j^h drops the packet
        break
    else
        G_j^h sends Puz_k to G_i^h
        G_j^h wait for TP_{out}
        G_i^h solves Puz_k and sends the response Re_k to G_j^h
        if ((Re_k is true) & (verified by G_j^h) & (within TP_{out})) **then** ▷ trusted operation
            G_j^h encrypts E(Res_j, Re_k)
            G_j^h sends E(Res_j, Re_k) to G_i^h
            G_i^h decrypts D(E(Res_j, Re_k), Re_k)
            G_i^h sends Res_j to P_i
            flag = **true**
            break
        else if ((Re_k is false) & (verified by G_j^h) & (within TP_{out})) **then** ▷ potentially untrusted
            continue
        else ▷ potentially untrusted
            TP_{out} has expired
            continue
        end if
    end if
end for
if flag == false **then** ▷ untrusted operation
    Add G_i^h to Un_{list}
    Broadcast the list to other group heads on Level 1
end if

chosen to reduce computational overhead and communication costs. Moreover, the designed protocol ensures minimal interactions to reduce communication between the verifier and prover. This makes the problem formulation suitable for the resource-constrained environment of P2P trust verification.

For proof generation and establishing trust, the sender group head node *proposes a one-step challenge* Puz_k (where k = 1, 2, 3 ...., it is set by the network administrator) such as calculating the SHA-256 hash of a random number. This random number can be changed easily to propose a new challenge for each hop and generate a new verification key. The receiver node (also a group head) responds to this challenge with a unique solution that can act as a nonce - a number used only once to ensure that old communications cannot be reused in replay attacks.

There are two potential outcomes: (1) If the receiver's response correctly solves the challenge proposed by the sender within the predetermined timeout period TP_{out} set by the network designer, the sender then uses this nonce as a symmetric key to encrypt the message and sends it to the recipient. This is shown in Figure 3. (2) If the response is incorrect or received after the expiry period (Figure 4), the receiver node is marked as potentially malicious. In this way, through correct responses to the sender's challenges, the receiver proves it meets the authentication requirements without revealing any sensitive

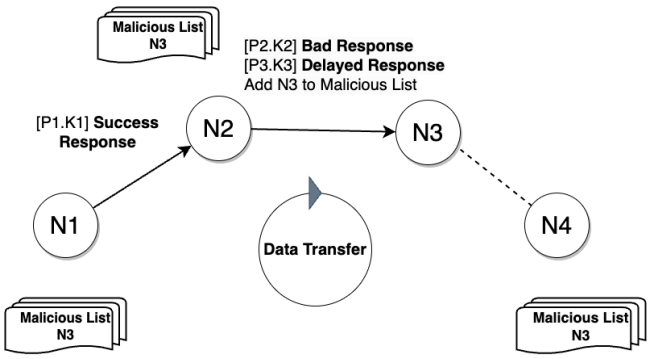

Figure 2: Group head level flow diagram showing trusted operation and untrusted peers being added as malicious.

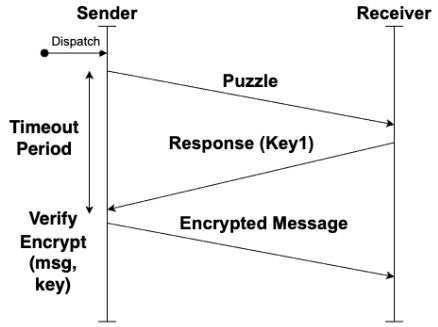

Figure 3: Sequence diagram for an encrypted message sent from sender to receiver after successfully verifying the receiver.

information. The sender's honesty is managed by continuous monitoring through zero trust architecture.

After verifying the trustworthiness of the group head node using ZKP's, they forward the message to the next hop in the network until the message reaches its intended destination. By **continuous trust verification**, it eliminates any assumptions that a peer is inherently trustworthy. Instead, each node has to prove and earn its integrity and authenticity, reducing the attacks from the corrupted nodes through sybil, and replay attacks among others.

5 Evaluation

Our protocol underwent a rigorous evaluation, considering a range of conditions with a focus on ideal scenarios where the number of hops required for efficient data transfer was optimally low (3 hops without ZKP protocol). Through extensive testing, we observed that the number of hops increased from a best-case scenario of 5 to a worst-case scenario represented by $2k+3$, where k represents the maximum number of puzzles set by the network administrator. This expansion in hops signifies an extended network reach, crucial for robust communication in P2P networks. Notably, this increase in network reach was achieved without a corresponding rise in control messages, ensuring efficiency and security. Our ZKP-based protocol maintains operational security and efficiency despite the need

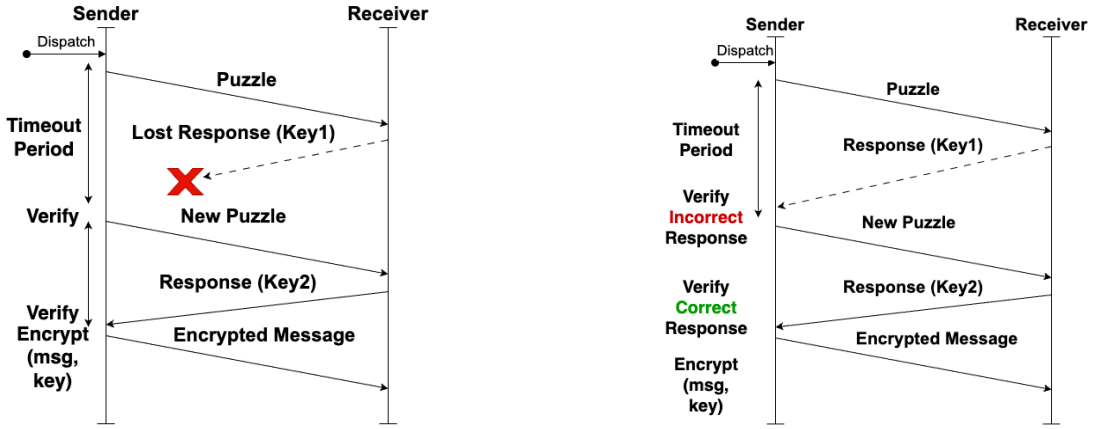

Figure 4: Sequence diagram depicting the protocol's response to a lost/delayed/incorrect solution from a receiver node. In this scenario, the sender node is necessitated to send a new challenge or puzzle during the next communication cycle to reaffirm the receiver's trustworthiness.

for more network navigation, highlighting its advantages for larger and more complex P2P network architectures where resource constraints and efficiency are vital considerations.

6 Conclusion

Our novel solution, leveraging Zero-Knowledge Proof (ZKP), resolves the vital issue of establishing and maintaining trust in peer-to-peer (P2P) networks, ensuring maximum security, privacy, and efficiency in data transfers. The protocol manages trust verification while balancing user anonymity and node legitimacy, enhancing network security and scalability, reducing the chances of malicious attacks. This contributes to increased network resilience, seamless data exchanges, and improved user experiences. Further, our solution meets the challenge of expanding network reach without requiring additional control messages. Even if the number of data transfer hops increases to a worst-case scenario of 2k+3 (where k represents the maximum number of puzzles set by the network administrator), our protocol maintains its operational security and efficiency. Thus, our approach allows network growth without sacrificing performance or security, offering an optimized, scalable solution for larger, more intricate P2P network architectures. Future work plans to address privacy challenges in P2P networks using similar cryptographic paradigms.

References

[1] Shane Balfe, Amit D Lakhani, and Kenneth G Paterson. Trusted computing: Providing security for peer-to-peer networks. In *Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05)*, pages 117–124. IEEE, 2005.

[2] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable zero knowledge with no trusted setup. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part III 39*, pages 701–732. Springer, 2019.

[3] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Ndss*, pages 1–15. San Diego, CA, 2009.

[4] Rzan Tarig Frahat, Muhammed Mostafa Monowar, and Seyed M Buhari. Secure and scalable trust management model for iot p2p network. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–6. IEEE, 2019.

[5] M Harikrishnan and KV Lakshmy. Secure digital service payments using zero knowledge proof in distributed network. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pages 307–312. IEEE, 2019.

[6] Swathi Kaluvakuri, Bidyut Gupta, Banafsheh Rekabdar, Koushik Maddali, and Narayan Debnath. Design of rc-based low diameter two-level hierarchical structured p2p network architecture. In Mohammed Serrhini, Carla Silva, and Sultan Aljahdali, editors, *Innovation in Information Systems and Technologies to Support Learning Research*, pages 312–320, Cham, 2020. Springer International Publishing.

[7] Swathi Kaluvakuri, Koushik Maddali, Nick Rahimi, Bidyut Gupta, and Narayan Debnath. Generalization of rc-based low diameter hierarchical structured p2p network architecture. *International Journal of Computer and Their Applications*, page 74, 2020.

[8] Dmitry Korzun and Andrei Gurtov. "Hierarchical Architectures in Structured Peer-to-Peer Overlay Networks". *Peer-to-Peer Networking and Applications, Springer*, 7(4):359-395, 2014.

[9] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE symposium on security and privacy (SP)*, pages 839–858. IEEE, 2016.

[10] Li Lu, Jinsong Han, Yunhao Liu, Lei Hu, Jin-Peng Huai, Lionel Ni, and Jian Ma. Pseudo trust: Zero-knowledge authentication in anonymous p2ps. *IEEE Transactions on Parallel and Distributed Systems*, 19(10):1325–1337, 2008.

[11] Koushik Maddali, Banafsheh Rekabdar, Swathi Kaluvakuri, and Bidyut Gupta. Efficient capacity-constrained multicast in rc-based p2p networks. In *Proceedings of 32nd International Conference on*, volume 63, pages 121–129, 2019.

[12] Will Major, William J Buchanan, and Jawad Ahmad. An authentication protocol based on chaos and zero knowledge proof. *Nonlinear Dynamics*, 99:3065–3087, 2020.

[13] Claudia Daniela Pop, Marcel Antal, Tudor Cioara, Ionut Anghel, and Ioan Salomie. Blockchain and demand response: Zero-knowledge proofs for energy transactions privacy. *Sensors*, 20(19):5678, 2020.

[14] Yingying Ren, Zhiwen Zeng, Tian Wang, Shaobo Zhang, and Guoming Zhi. A trust-based minimum cost and quality aware data collection scheme in p2p network. *Peer-to-Peer Networking and Applications*, 13:2300–2323, 2020.

[15] Ali Aydin Selcuk, Ersin Uzun, and Mark Resat Pariente. A reputation-based trust management system for p2p networks. In *IEEE International Symposium on Cluster Computing and the Grid, 2004. CCGrid 2004.*, pages 251–258. IEEE, 2004.

[16] Xiaoqiang Sun, F Richard Yu, Peng Zhang, Zhiwei Sun, Weixin Xie, and Xiang Peng. A survey on zero-knowledge proof in blockchain. *IEEE network*, 35(4):198–205, 2021.

[17] Patrick P Tsang and Sean W Smith. PPAA: Peer-to-peer anonymous authentication. In *Applied Cryptography and Network Security: 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings 6*, pages 55–74. Springer, 2008.

[18] Xiaohui Yang and Wenjie Li. A zero-knowledge-proof-based digital identity management scheme in blockchain. *Computers & Security*, 99:102050, 2020.

[19] Shanyu Zhao, Virginia Lo, and C Gauthier Dickey. Result verification and trust-based scheduling in peer-to-peer grids. In *Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05)*, pages 31–38. IEEE, 2005.