A Comparative Analysis of Cryptographic Techniques for Privacy Preservation in Blockchain-Based Dispute Resolution

Yassine Obeid[1], Christiana Zaraket[2] and Layth Sliman[1]
[1]*Efrei Paris, Paris Pantheon-Assas University, Paris, France*
[2]*Saint Joseph University of Beirut, ESIB, Beirut, Lebanon*

Keywords: Confidentiality, Untraceable Transactions, Cryptography, Transparency, Stealth Addresses, Blockchain Privacy.

Abstract: Digital innovations have profoundly altered the landscape of remote collaboration, yet these innovations frequently lead to conflicts that necessitate robust and fair decentralized decision-making systems.These conflicts occur either because they are not yet regulated by law, or because they arise from disputes associated with onchain applications, or simply because they are easier to resolve online. Public blockchains, characterized by their transparency and immutability, present viable solutions for these systems. However, the very transparency that makes blockchains appealing also introduces significant privacy concerns, as the traceability of transactions can jeopardize the anonymity of participants. Although users can maintain a degree of pseudoanonymity through cryptographic addresses, their activities remain publicly accessible, which poses considerable risks if their identities are uncovered. To mitigate these challenges, various cryptographic methods, including stealth addresses and zero-knowledge proofs (zk-SNARKs), have been developed to bolster transaction privacy. This paper distinguishes itself by offering an in-depth examination of these privacy-enhancing techniques, emphasizing their integration within blockchain environments, as well as their scalability and programmability. Additionally, we address key limitations, such as the balance between privacy and computational complexity, along with the interoperability issues that arise among privacy-centric protocols. By providing a comparative analysis and investigating future research avenues, this paper contributes valuable perspectives on reconciling privacy and transparency in decentralized collaboration frameworks.

1 INTRODUCTION

In the past few decades, the advent of digital transformation has markedly improved remote collaboration, fostering interactions across diverse industries and geographical boundaries. This evolution has enabled individuals and organizations to engage in effective teamwork from afar. However, these collaborations are not without challenges; disputes may arise, necessitating the establishment of decentralized decisionmaking frameworks that are both resilient and fair. Public blockchains, with their transparency and immutability, appear promising for facilitating these decentralized solutions.Yet, this very transparency raises major privacy concerns, as the traceability of transactions can compromise the anonymity of participants.

On public blockchains like Bitcoin and Ethereum, users operate under a pseudo-anonymous model, identified by cryptographic addresses. However, since the details of transactions remain publicly accessible, if a user's identity linked to an address is revealed, their transaction history becomes traceable. This exposure poses serious privacy risks, particularly for active participants who may be vulnerable to tracing and profiling attempts. Thus, finding a balance between transaction privacy and transparency is a critical challenge in these decentralized systems.

Before addressing the issue of privacy, it is important to consider why public blockchain makes transactions visible. This transparency is essential for preventing double-spending, ensuring that each transaction is verified and unique, thus preventing the same asset from being used multiple times. One of the most significant challenges arises from this: making transactions untraceable and unlinkable on the blockchain. Untraceability and unlinkability are two

Obeid, Y., Zaraket, C. and Sliman, L.
A Comparative Analysis of Cryptographic Techniques for Privacy Preservation in Blockchain-Based Dispute Resolution.
DOI: 10.5220/0013415500003929

In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 1*, pages 1089-1096
ISBN: 978-989-758-749-8; ISSN: 2184-4992

distinct but crucial concepts. Untraceability involves the impossibility of determining the sender of a transaction among a group of potential senders, thus ensuring a high level of privacy. Unlinkability, on the other hand, implies the impossibility of verifying that two outgoing transactions are intended for the same recipient, further protecting identity and connections between transactions (Bernabe et al., 2019).

In response to these challenges, researchers have explored advanced cryptographic methods to render transactions untraceable while maintaining the transparency necessary to secure decentralized processes. Stealth addresses and zk-SNARKs help obscure transaction origins and destinations. These techniques enhance user privacy while preserving system integrity. These methods enable public blockchains to support secure and privacy-focused collaborative processes while adhering to transparency requirements.

This article provides a comprehensive analysis of cryptographic techniques designed to enhance transaction privacy and security within decentralized frameworks. By synthesizing existing literature and evaluating contemporary methodologies, it lays the foundation for novel approaches to strengthening privacy in public blockchain environments, while addressing trade-offs between anonymity, efficiency, and usability. We analyze the limitations of current methods and discuss key challenges in achieving an effective balance between privacy and transparency. The remainder of this article is organized as follows: Section II provides an overview of stealth addresses, zk-SNARKs, and associated cryptographic techniques. Section III presents state-of-the-art methods for untraceable transactions and their limitations. Section IV identifies key challenges, and Section V proposes potential research avenues. Finally, Section VI concludes by offering our perspective on privacy in public blockchains.

2 BACKGROUND

This section aims to offer a concise summary of stealth addresses, zk-SNARKs, and various services designed to render transactions untraceable.

2.1 Stealth Addresses

Stealth addresses illustrated in Figure 1 allow transactions to be concealed by creating a new address for each operation. This prevents an observer from tracking transactions intended for a specific address. This concept is particularly important in the context of public blockchains, where the traceability of transactions poses a significant risk to user privacy.Stealth addresses are already integrated into Monero (Monero, 2013), a privacy-focused cryptocurrency, and there are ongoing efforts to implement them within Ethereum (Wahrstätter et al., 2024).

In public blockchains, every transaction is recorded on a public ledger, meaning that all addresses and their associated transactions can be observed. Stealth addresses provide a means to protect user privacy by generating unique addresses for each transaction.

When a sender wishes to send an amount to a recipient, they create a stealth address that does not exist beforehand. This process involves several steps:

1. **Generation of a Stealth Address:** The sender uses their ephemeral public key and the recipient's stealth public key to generate a unique stealth address. This address is not published in the ledger, making it difficult for an observer to associate it with the recipient's identity.
2. **Sending the Transaction:** Once the stealth address is generated, the sender can send the amount to this address. Therefore, the sender is sending funds to an address that, in the eyes of the public, is not linked to a specific user, as it was created specifically for this transaction.
3. **Accessing Funds by the Recipient:** To access the funds sent to the stealth address, the recipient must possess the corresponding private key. To do this, they use their spending private key (which is linked to their identity) and a cryptographic method to recover the funds. By combining their private key and the ephemeral public key generated by the sender, they can prove they are the rightful owner of the stealth address.
4. **Spending the Funds:** Once the recipient has access to the stealth address, they can spend the funds as they wish, again using a stealth address for their own transactions, thus ensuring that each new operation remains private.

This ability to create and use stealth addresses ensures that even if an observer can see the transaction on the blockchain, they cannot associate the funds with the recipient's identity, thereby enhancing user privacy.

2.2 zk-SNARKs

A zk-SNARK is a cryptographic protocol (Mayer, 2016) that allows a prover to demonstrate to a verifier

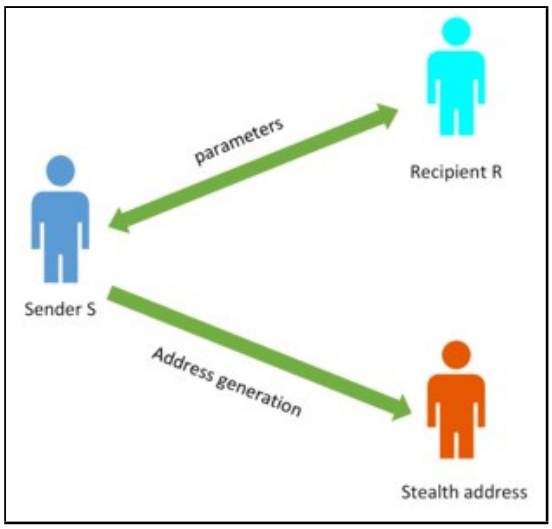

Figure 1: Stealth Address Process.

that he knows a solution to a problem or that he has performed a computation correctly without revealing any information about the solution itself. This type of proof has several key characteristics:

- **Zero-Knowledge:** The verifier learns that the proof is correct but gains no further information about the underlying data or computation.
- **Succinct:** The proof size is very small (often only a few hundred bytes), regardless of the size of the computation being proven.
- **Non-Interactive:** No extensive interaction is needed between the prover and verifier. The proof is unique and can be verified directly.
- **Argument of Knowledge:** The proof guarantees that the prover genuinely possesses the correct information or solution.

The main steps in a zk-SNARK are as follows:

1. Setup: this step is used to create public parameters (often denoted pp) that are utilized by both the prover and the verifier. Depending on the type of zk-SNARK, this may require a *trusted setup*, where a trusted entity generates these parameters and then destroys certain secret information. There are also zk-SNARKs with a universal or transparent setup, which do not require this trusted setup phase.
2. **Proof Generation:** the prover uses the public parameters and their solution or computation to generate a succinct proof. The proof is generally much smaller than the underlying computation or data.
3. **Verification:** the verifier uses the proof and public parameters to verify that the proof is correct. This is done in constant time (very quickly) without the verifier needing to know the prover's computation or data.

There are different types of zk-SNARK setups, each designed to meet distinct security and performance requirements. For instance, some zk-SNARKs require a trusted setup, where a reliable entity generates public parameters and removes certain confidential information to ensure system integrity. In contrast, zkSNARKs with a universal or transparent setup eliminate this initial phase by employing protocols or algorithms that allow any participant to independently verify the process, avoiding reliance on a third party. These methodologies cater to a range of applications, from highly confidential systems with minimal external involvement to more adaptable solutions that emphasize transparency in verification.

2.3 Cryptographic Commitments

Cryptographic commitments are protocols that allow an individual (the committer) to commit to a secret value in a way that prevents any alteration afterward while keeping it hidden from other parties (the verifiers) until a designated moment. This technique is crucial for many cryptographic and security protocols, as it ensures both the integrity and confidentiality of the data without requiring immediate disclosure. A cryptographic commitment is often compared to a "locked box": the committer can seal their value inside (the commitment) but cannot modify it once the box is closed, with the "key" remaining secret until the reveal phase. This commitment process generally unfolds in three main steps:

1. **Commitment Phase:** the committer chooses a secret value m (message) and a random factor r. Using a commitment function Com, they create a commitment C =Com(m,r), which hides m and is computationally infeasible to reverse without knowing both m and r. The committer then sends the commitment C to the verifiers, ensuring that m remains immutable and hidden until the reveal phase.
2. **Holding Phase:** the commitment C is retained without revealing the values of m or r. The verifier only knows that C represents a specific fixed m, but they have no access to the actual values of m or r.
3. **Reveal Phase:** the committer reveals m and r to the verifiers. Each verifier can compute Com(m,r)

and verify that it matches the initially received C, thereby ensuring the integrity of m. If the equation C =Com(m,r) holds, the verifier accepts m as the committed value.

Different types of cryptographic commitment setups exist to suit varying security and application needs. For example, some commitments require a trusted setup where an entity securely generates and later destroys certain secret values involved in the setup process. Other types of commitments, such as those with a transparent setup, do not rely on a trusted party, instead using protocols that allow any party to verify the setup independently. These approaches offer flexibility across applications that prioritize either trust minimization or transparency in the commitment process.

2.4 Ring Signatures

Ring signatures are a cryptographic technique used to obscure the sender in a transaction, ensuring anonymity. Ring signatures allow a user's transaction to blend with others, making it difficult to determine the actual signer. This process involves several steps:

1. **Formation of the Ring:** the sender's public key is combined with a selection of other randomly chosen public keys from the blockchain, creating a "ring" of possible signers. This ring provides plausible deniability, as any of these keys could feasibly be the signer.
2. **Signing Process:** the sender generates a unique ring signature using their private key along with the set of public keys. Importantly, this signature proves that one of the keys in the ring signed the transaction, without revealing which one.
3. **Verification:** the network nodes can verify the signature's validity without identifying the actual signer, ensuring the integrity of the transaction. This verification ensures that no double-spending occurs while maintaining the anonymity of the sender.

Ring signatures significantly enhance privacy by allowing the source of funds to remain untraceable, providing users with both security and confidentiality.

3 STATE OF THE ART

In this section, we will examine a range of pertinent methodologies, emphasizing their advantages while also addressing their shortcomings.

BaseSAP (Wahrstätter et al., 2024) proposes a modular protocol to enable untraceable transactions on public blockchains. This protocol operates as a layer above foundational protocols like DKSAP, providing a platform to integrate stealth addresses through various cryptographic algorithms. In this approach, senders publish announcements of ephemeral public keys via an Announcer contract, allowing recipients to identify transactions intended for them while ensuring the privacy of involved parties. To enhance protocol efficiency and reduce operational costs, mechanisms like "view tags" and toll or staking systems are included to protect against Denial of Service (DoS) attacks.

While the protocol initially builds on DKSAP, advanced versions like PDKSAP (Feng et al., 2020) and EDKSAP (Feng et al., 2021) offer improvements in computation, notably through bilinear pairing to reduce the number of operations required and processing time. The main differences between these protocols are summarized in Table 1, highlighting how each variation optimizes for specific aspects like privacy, computational efficiency, and parsing performance.

This modular framework enables BaseSAP to be adapted to different implementations, addressing specific privacy and anonymity needs.However, a limitation of BaseSAP is that it does not conceal the sender's address, which may reduce the overall anonymity in certain scenarios.

Whereas BaseSAP adds a privacy layer on Ethereum, which does not natively use stealth addresses, blockchains like Monero have integrated stealth addresses from the start and combine them with other cryptographic techniques to further enhance privacy (Monero,).

Key privacy technologies include Ring Confidential Transactions (RingCT) (Noether, 2015), which conceal transaction amounts using cryptographic commitments (specifically Pedersen commitments) to ensure both confidentiality and verifiability of transaction amounts. Other methods include stealth addresses, generating a unique address for each transaction, and ring signatures (Vijayakumaran, 2018), which obscure the sender's identity by mixing their signature with others. Additionally, transactions can be routed through Tor/I2P to mask IP addresses, while Dandelion++ (Fanti et al., 2018) protects user anonymity by obscuring transaction origins before they are publicly relayed.

Zcash (Ben-Sasson et al., 2014), one of the first independent blockchains to incorporate zk-SNARKs, focuses primarily on financial transaction privacy. It

Table 1: Comparison of DKSAP, PDKSAP, and EDKSAP Protocols.

Criterion	DKSAP	PDKSAP	EDKSAP
Utilization of Analysis Key	Yes	Yes	Yes
Spending Key	Yes, separate for transaction security	Yes, separate	Yes, separate
Main Goals	Basic privacy and sender anonymity	Reduced computations in the parsing process	Performance improvement by reducing calculations
Parsing Optimization	No	Yes, by reducing computations with a "view tag"	Yes, through the use of bilinear pairings
Cryptographic Algorithms	Based on standard elliptic curve operations	Based on elliptic curves and "view tags"	Based on bilinear pairings for optimized computation

allows users to choose between transparent and shielded transactions, with shielded transactions fully masking addresses and amounts. However, Zcash remains limited to private financial transactions and does not support decentralized applications (dApps), restricting its ecosystem to use cases centered on transaction privacy.

Aleo (Aleo, 2024) represents a cutting-edge blockchain initiative aimed at enhancing privacy beyond mere transactional confidentiality by facilitating the development of confidential decentralized applications (dApps) through its innovative zeroknowledge virtual machine (zkVM) (Liu et al., 2024). The zkVM empowers Aleo with a high degree of programmability, enabling developers to construct intricate dApps while ensuring the protection of sensitive data. Nevertheless, this level of flexibility incurs certain drawbacks; specifically, the computational demands associated with confidential applications can be quite substantial, leading to increased costs for proof generation. Furthermore, as a nascent blockchain, Aleo encounters obstacles in achieving interoperability with more established blockchain networks, which are often characterized by their closed systems, distinct consensus mechanisms, and varying privacy protocols. This situation hinders Aleo's capacity to integrate smoothly into the broader blockchain ecosystem. The relative immaturity of Aleo may also limit the options available to developers in search of interoperable solutions, thereby affecting the proliferation and acceptance of dApps built on the Aleo platform.

Aztec (Williamson, 2018), in contrast, operates as a Layer 2 on Ethereum, bringing privacy features to the Ethereum ecosystem. Through Aztec Connect, private transfers of ERC-20 tokens and other confidential operations are possible within the DeFi space on Ethereum. While its programmability is more limited than Aleo's, Aztec offers valuable privacy for DeFi dApps, effectively masking transaction amounts and addresses. The solution uses zk-rollups (Lavaur et al., 2023) to enhance scalability by grouping multiple transactions into a single proof.However, Aztec's application scope is relatively limited: while it supports DeFi protocols and private payments. Additionally, developing dApps on Aztec can be complex, as developers must adapt their applications to align with Aztec's privacy protocols, requiring specific adjustments for compatibility.

Mimblewimble (Silveira et al., 2024) is a blockchain protocol focused on privacy and scalability, proposed in 2016 by an anonymous developer. Using advanced techniques such as Pedersen commitments and confidential transactions to mask transaction amounts and temporary identifiers for involved parties, Mimblewimble achieves a high degree of privacy. However, this protocol has several significant limitations: it does not support complex smart contracts, restricting its applications to financial transactions only. Additionally, its implementation is complex due to the advanced cryptographic mechanisms it employs, requiring specific validations and making integration more challenging than with conventional blockchains. Despite these limitations, Mimblewimble serves as the foundation for several existing blockchains, including Grin and Beam, which use it to enable confidential and anonymous transactions, as well as Litecoin, which has integrated it through the Mimblewimble Extension Blocks (MWEB) for enhanced privacy options. Table 1 presents a comparative overview of existing approaches in terms of privacy and programmability of blockchains and protocols.

Table 2: Comparison of Existing Approaches in Terms of Privacy and Programmability.

Protocol/Blockchain	Type	Use Case	Programmability
Zcash (van Saberhagen, 2013)	Independent blockchain	Financial transactions	Non-programmable
Monero (Vijayakumaran, 2018)	Independent blockchain	Financial transactions	Non-programmable
Aleo (Bernabe et al., 2019)	Independent blockchain	Private transactions and dApps	Programmable (via zkEVM)
Aztec (Silveira et al., 2024)	Layer 2 on Ethereum	Privacy for Ethereum transactions	Programmable (via Aztec Connect)
Beam	Independent blockchain	Financial transactions	Non-programmable
Grin	Independent blockchain	Financial transactions	Non-programmable
Litecoin MWEB	Extension for existing blockchain	Optional private transactions on Litecoin	Non-programmable

A comparative overview of the discussed protocols and blockchains is presented in Table 2, highlighting key differences in types,use cases and programmability.

4 DISCUSSION

The subsequent discussion expands upon the examination of existing privacy-preserving blockchain solutions and protocols, focusing on the challenges and trade-offs that arise in relation to their integration, scalability, and adherence to regulatory requirements.

Balancing Privacy and Programmability: the examined blockchains demonstrate diverse strategies for incorporating privacy; however, a consistent tradeoff becomes apparent: enhanced privacy frequently restricts programmability, as illustrated by Zcash and Monero. The expansion of private features to accommodate more intricate applications, akin to those facilitated by Aleo or Aztec, results in increased computational expenses and integration challenges. This situation prompts an inquiry into the ideal equilibrium between flexibility and performance. Building upon the trade-off between privacy and programmability, the integration of zk-proof algorithms into blockchain transactions introduces another layer of complexity, as the choice of protocol must balance performance, security, and verification costs.

Feasibility of zk-Proof Algorithms for Blockchain Integration: based on the technical characteristics of each zk-Proof protocol as shown in the table 3, integrating zero-knowledge proofs into blockchain transactions requires a balance between performance, verification costs, and security. For instance, protocols with compact proof sizes and fast verification times (such as Groth'16) are particularly suitable for high-transaction throughput blockchains, despite the constraints imposed by a trusted setup. On the other hand, protocols without a trusted setup, such as Bulletproofs and STARK, are preferable for applications where security is paramount, although they come with larger proof sizes and potentially longer verification times. This technical evaluation raises questions about the adaptability of each algorithm depending on the specific constraints of blockchains and use cases, guiding the choice of the most suitable protocol for confidential and programmable implementations while ensuring long-term feasibility.

Scalability and Adoption within Existing Ecosystems: solutions such as Aztec's zk-rollups and Aleo's zkVM offer mechanisms for enhancing privacy on existing blockchains; however, their implementation necessitates considerable technical modifications by developers, which can hinder swift integration. A crucial consideration for the future is whether these protocols can streamline their operations while upholding privacy assurances, thereby promoting broader adoption within established networks such as Ethereum.

Interoperability and Standardization of Privacy Technologies: the necessity for interoperability among blockchains is becoming increasingly vital for confidential decentralized applications (dApps), as demonstrated by the challenges faced by initiatives

such as Aleo. A major obstacle is the need to promote standardized cryptographic methods to enable the seamless migration of dApps across blockchains. Achieving this interoperability could enhance the adoption of privacy protocols and reduce ecosystem fragmentation.

Resilience to Attacks and System Privacy Durability: a key consideration is protocol resilience to attacks, including Denial of Service (DoS) attacks. The mechanisms presented in BaseSAP are designed to enhance resilience; however, their long-term effectiveness remains uncertain. Future research may explore additional strategies to enhance the resilience of privacy systems, ensuring their long-term viability for end users.

Regulatory Considerations and Compliance Strategies: privacy-focused blockchain solutions, such as zk-SNARKs and confidential smart contracts, encounter major regulatory hurdles. While these technologies strengthen privacy protections, they often clash with regulatory mandates such as AntiMoney Laundering (AML) and Know Your Customer (KYC) policies, as well as data protection laws like the GDPR. The immutable nature of blockchain systems conflicts with the GDPR's right to erasure provision, and privacy features may hinder compliance with AML monitoring and verification requirements.

Table 3: Comparison of Proof Size and Verification Time for ZKP Algorithms.

Algorithm	Proof Size	Verification Time
Groth'16	$O(1)$	$O(1)$
Bulletproofs	$O(\log N)$	$O(N)$
STARK	O(poly-log(N))	O(poly-log(N))

5 RESEARCHDIRECTIONS

Advancements in privacy and programmability in public blockchains open promising research opportunities to address existing challenges and improve decentralized solutions. We highlight key research directions in this domain:

Optimization of Zero-Knowledge Proofs: improving the efficiency of zk-SNARKs and zkSTARKs is crucial for reducing computational costs and verification times while strengthening protocol security. These advancements are essential for highthroughput blockchains and can significantly enhance the usability of privacy-preserving solutions for various blockchain applications.

Interoperability Between Blockchains: enhancing the standardization and interoperability of privacy technologies among different blockchains could drive wider adoption and facilitate the seamless migration of confidential decentralized applications (dApps) across blockchain ecosystems. Exploring secure bridging solutions and inter-chain communication protocols is crucial for achieving this goal.
Strengthening the Resilience of Privacy Protocols: defending against attacks, especially DoS attacks, is a critical challenge for privacy protocols. Thorough research is crucial to develop robust selfdefense mechanisms and assess the long-term resilience of privacy systems against emerging threats.

Expanding Programmability for Confidential Applications: improving programmability in privacy-focused blockchains, particularly in executing complex smart contracts, remains a major challenge. While frameworks such as Aleo and Aztec provide a strong foundation, further research is needed to balance privacy with application flexibility.

Developing Economically Viable Solutions: the high cost of private transactions, mainly due to their computational complexity, hinders widespread adoption. Creating cost-effective and efficient solutions could make private transactions more accessible and encourage their adoption.

6 CONCLUSION

In summary, striking the right balance between privacy and transparency is essential to fostering trust in decentralized collaboration frameworks. As cryptographic techniques continue to evolve, it is possible to envision a future where secure, private, and transparent transactions coexist on public blockchains, thus laying the foundation for a resilient and fair digital ecosystem.

The rise of digital collaboration has highlighted the need for privacy-preserving mechanisms in public blockchains, particularly as interactions in decentralized systems become more widespread. This study explores sophisticated cryptographic methods, including stealth addresses and zk-SNARKs, designed to enable untraceable and secure transactions. Furthermore, it evaluates their existing applications, limitations, and the challenges of balancing transparency and privacy.

Public blockchains, despite advancements in privacy measures, remain inherently transparent. To address this challenge, it is essential to develop advanced cryptographic techniques to enhance anonymity without compromising security or efficiency. To unlock the full potential of decentralized collaboration, future research should focus on optimizing privacy solutions to reduce computational costs and enhance scalability. Furthermore, achieving interoperability among privacy-enhancing protocols across blockchain ecosystems could empower users by enabling secure cross-platform interactions.

Our forthcoming research will focus on exploring approaches to enhance privacy-preserving solutions and scalability while ensuring smooth interoperability across blockchain ecosystems.

REFERENCES

Aleo (2024). Aleo record model: A secure and efficient blockchain. Online. Available: https://aleo.org/post/aleo-record-model-secureefficient-blockchain/.

Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., and Virza, M. (2014). Zerocash: Decentralized anonymous payments from bitcoin. In *Proceedings of the IEEE Symposium on Security Privacy (Oakland)*, pages 459–474.

Bernabe, J. B., Canovas, J. L., Hernandez-Ramos, J. L., Moreno, R. T., and Skarmeta, A. (2019). Privacypreserving solutions for blockchain: Review and challenges. In *IEEE Access*, volume 7, pages 164908– 164940.

Fanti, G., Venkatakrishnan, S. B., Bakshi, S., Denby, B., Bhargava, S., Miller, A., and Viswanath, P. (2018). Dandelion++: Lightweight cryptocurrency networking with formal anonymity guarantees.

Feng, C., Tan, L., Xiao, H., Qi, X., Wen, Z., and Liu, Y. (2021). Edksap: Efficient double-key stealth address protocol in blockchain. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1196–1201.

Feng, C., Tan, L., Xiao, H., Yu, K., Qi, X., Wen, Z., and Jiang, Y. (2020). Pdksap: Perfected double-key stealth address protocol without temporary key leakage in blockchain. In *2020 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, pages 151–155.

Lavaur, T., Detchart, J., Lacan, J., and Chanel, C. P. C. (2023). Modular zk-rollup on-demand. *Journal of Network and Computer Applications*, 217.

Liu, T., Zhang, Z., Zhang, Y., Hu, W., and Zhang, Y. (2024). Ceno: Non-uniform, segment and parallel zero-knowledge virtual machine. Cryptology ePrint Archive, Paper 2024/387. Available: https://eprint.iacr.org/2024/387.

Mayer, H. (2016). zk-snark explained: Basic principles. CoinFabrik. Dec. 13.

Monero. About monero. Online. Available: https://www.getmonero.org/fr/resources/about/.

Monero (2013). Stealth address. Online. Available: Modular stealth address protocol for programmable blockchains. *IEEE Transactions on Information Forensics and Security*, 19:3539–3553.

Williamson, Z. J. (2018). The aztec protocol. AZTEC, Version 1.0.1. https://www.getmonero.org/fr/moneropedia/stealthaddress.html.

Noether, S. (2015). Ring signature confidential transactions for monero. Cryptology ePrint Archive, Paper 2015/1098. Available: https://eprint.iacr.org/2015/1098.

Silveira, A., Betarte, G., Cristiá, M., and Luna, C. (2024). A formal analysis of the mimblewimble cryptocurrency protocol. Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay.

Van Saberhagen, N. (2013). Cryptonote v 2.0. Online. Available: https://web.archive.org/web/20201028121818/.

Vijayakumaran, S. (2018). Monero ring signatures.

Wahrstätter, A., Solomon, M., DiFrancesco, B., Buterin, V., and Svetinovic, D. (2024). Basesap: