CUKUROVA UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Analysis and Comparison of the Popular Zero-Knowledge Proof Algorithms

Sami HEYBET

Department of Computer Engineering

July, 2025

CUKUROVA UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS APPROVAL

Analysis and Comparison of the Popular Zero-Knowledge Proof Algorithms

---

Sami HEYBET

Department of Computer Engineering

This Master's Thesis was evaluated by the following Jury Members on 18/07/2025 and was approved by majority of votes.

Jury : Assoc. Prof. Fatih ABUT (Advisor) ……

: Asst. Prof. Barış ATA ……

: Assoc. Prof. Ali İNAN ……

This Thesis was written in the Department of Computer Engineering, Institute of Natural and Applied Sciences.

Thesis Number:

Prof. Dr. Sadık DİNÇER

Director

Institute of Natural and Applied Sciences

CONTENTS

CUKUROVA UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Analysis and Comparison of the Popular Zero-Knowledge Proof Algorithms

---

Sami HEYBET

Advisor: Assoc. Prof. Fatih ABUT

Department of Computer Engineering

ABSTRACT

The objective of this thesis is to examine zero-knowledge proof-of-knowledge schemes that seek to substantiate the veracity of information without divulging any confidential information. Additionally, the study aims to propose schemes that are well-suited for these scenarios by conducting a comparative analysis of various application scenarios and the information obtained. Additionally, real-world applications in finance, health, business and other general-purpose real-world applications as presented in the literature are analysed and the schemes used in these applications are investigated. Zero-knowledge proof schemes are grouped according to criteria parallel to those used in the literature. Applications are developed with an algorithm from the current schemes representing these groups and compared in the light of basic criteria such as proof size, proof generation time, proof verification time and whether pre-installation is required. Various suggestions are made for certain usage scenarios. It was determined that all of the compared schemes are capable of meeting the security and privacy needs of decentralized systems, such as blockchains. Zk-SNARK schemes were found to have a relatively short proof size, but a longer proof time than other schemes, require a significant amount of memory usage and necessitate pre-installation. However, Bulletproof schemes are found to have the shortest proof size, although this increases the proof time. Zk-STARK schemes are relatively new, but they are highly secure, even for the post-quantum era and offer the promised proof size and proof time.

Keywords: ZKP, zk-SNARK, zk-STARK, Bulletproofs, Cyber Security.

ÇUKUROVA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

Popüler Sıfır-Bilgi Kanıt Algoritmalarının Analizi ve Karşılaştırılması

Sami HEYBET

Danışman: Doç.Dr. Fatih ABUT

Bilgisayar Mühendisliği Anabilim Dalı

ÖZ

Bu çalışma, her hangi bir bilgi ifşa etmeden bilginin doğruluğunu ispat etmeyi amaçlayan sıfır bilgi ispatı şemalarının incelenmesi, elde edilen bilgiler ışığında çeşitli uygulama senaryoları için karşılaştırma ve analiz yapılarak bu senaryolara uygun şemaların önerilmesini amaçlamaktadır. Literatürdeki finans, sağlık, kurumsal ve genel amaçlı gerçek dünya uygulamaları da incelenmiş, bu uygulamalarda hangi şemaların kullanıldığı araştırılmıştır. Sıfır bilgi ispatı şemaları literatüre paralel kriterlere göre guruplandırılmış ve bu grupları temsil eden güncel şemalardan birer algoritma ile uygulamalar geliştirilerek kanıt boyutu, kanıt üretim süresi, ispat süresi, ön kurulumla ihtiyaç duyup duymadığı gibi temel kriterler ışığında kıyaslanarak belli kullanım senaryoları için çeşitli önermelerde bulunulmuştur. Kıyaslanan tüm şemaların özellikle blok zinciri gibi merkeziyetsiz sistemlerin güvenlik ve gizlilik ihtiyaçlarını karşılamak konusunda kullanılabileceği görülmüştür. Zk-SNARK şemalarının görece kısa kanıt boyutuna rağmen ispat süresinin diğer şemalara kıyasla daha uzun olduğu, büyük miktarda bellek kullanımı ihtiyacı olduğu ve ön kuruluma ihtiyaç duyduğu tespit edilmiştir. Buna rağmen Bulletproofs şemalarının en kısa kanıt boyutuna sahip olan şema olduğu fakat bunun ispat süresini uzattığı görülmüştür. Zk-STARK şemalarının görece oldukça yeni olduğu buna rağmen vaad ettiği kanıt boyutu ve ispat süresi yanı sıra quantum sonrası dönem için bile güvenli olması nedeniyle oldukça gelecek vaad eden bir şema olduğu tespit edilmiştir.

Keywords: ZKP, zk-SNARK, zk-STARK, Bulletproofs, Siber Güvenlik.

EXTENDED ABSTRACT

Since the dawn of communication technologies, the possibility of messages being intercepted by third parties has emerged as a fundamental threat to privacy and data security. Throughout history, various encryption methods have been developed to ensure information security, but these methods have faced various limitations in meeting the nature of communication and security requirements. Encryption methods often base their reliability on the difficulty of solving complex mathematical problems; as the level of complexity increases, the security of these systems is considered stronger. Modern encryption algorithms in use today possess levels of complexity that even the most powerful classical computers would struggle to solve. However, the development and ever-increasing potential of quantum computers have become a significant threat to the security of existing encryption methods. Quantum computers have much higher processing speeds than classical computers and are theoretically capable of solving even today's most secure encryption solutions in seconds. Therefore, the advancement of quantum computing has led us to question the reliability of traditional encryption methods and necessitated the development of new approaches that can meet security requirements.

This paradigm shift has called into question the fundamental assumptions of cryptographic security and necessitated the development of new methods and approaches capable of meeting future information security needs. This new threat environment presented by quantum computing has accelerated research into alternative solutions to ensure the secure and confidential transmission of data. In this context, zero-knowledge proofs (Zero-Knowledge Proof, or ZKP), developed in addition to traditional encryption methods, stand out as a significant innovation in data security. Zero-knowledge proofs allow one party to prove to another that they possess certain information without revealing it, simply by proving the "correctness of the information." This technique minimizes the risk of information being intercepted by third parties and offers an innovative solution in many areas requiring confidentiality, such as identity verification, financial transactions, and secure data transfer. ZKP technology was first proposed in 1985 by Goldwasser, Micali, and Rackoff, and provides particularly strong protection against the risk of disclosure of confidential information. Essentially preserving privacy, ZKP offers a unique method for information security that can address various security needs.

ZKP protocols traditionally have an interactive structure, requiring the exchange of information between the verifier and the prover. This exchange of information necessitates a fresh start for the verification process. This can limit their use in many scenarios because each verification requires a different communication process. To overcome this limitation, Feige, Fiat, and Shamir applied the ZKP protocol to identity verification in 1988, introducing the concept of non-interactive zero-knowledge proofs (NIZKP). Non-interactive zero-knowledge proofs allow

verification to be completed in a single transaction. This feature offers a significant advantage, especially in systems requiring continuous verification, such as blockchain. The rapid spread of blockchain technology has further increased the importance of NIZKP for improving the security and efficiency of verification processes.

Today, ZKP algorithms can be categorized into three main categories: zk-SNARK, zk-STARK, and Bulletproofs, based on their security, efficiency, and application areas. These algorithms differ significantly in terms of their level of security, efficiency, mathematical basis, and application areas. Zk-SNARK algorithms are considered a popular option for zero-knowledge proofs. The key features of zk-SNARKs are their short proof lengths, fast verification processes, and wide application areas. These algorithms produce very short proofs compared to other zero-knowledge proofs, reducing data storage costs and providing advantages in applications requiring the verification of large amounts of data. However, a key disadvantage of zk-SNARK algorithms is their requirement for a trusted setup. This trusted setup process makes the system dependent on a central trust element for security. If the private key used during setup is disclosed, the security of the system can be threatened. Mathematically, zk-SNARK algorithms use advanced mathematical structures such as elliptic curve cryptography to generate proofs and accelerate verification processes. These features make them widely used, particularly in blockchain platforms and smart contracts (such as Ethereum). However, the high computational cost of zk-SNARKs may cause performance degradation when processing large data, which may be limiting for applications in some scenarios.

Zk-SNARK algorithms are considered a popular option for zero-knowledge proofs. The key features of zk-SNARKs are their short proof lengths, fast verification processes, and wide application areas. These algorithms produce very short proofs compared to other zero-knowledge proofs, reducing data storage costs and providing advantages in applications requiring the verification of large amounts of data. However, a key disadvantage of zk-SNARK algorithms is their requirement for a trusted setup. This trusted setup process makes the system dependent on a central trust element for security. If the private key used during setup is disclosed, the security of the system can be threatened. Mathematically, zk-SNARK algorithms use advanced mathematical structures such as elliptic curve cryptography to generate proofs and accelerate verification processes. These features make them widely used, particularly in blockchain platforms and smart contracts (such as Ethereum). However, the high computational cost of zk-SNARKs may cause performance degradation when processing large data, which may be limiting for applications in some scenarios.

Zk-STARK algorithms were developed to eliminate the trusted installation requirement of zk-SNARK algorithms and offer a structure that does not require a trusted installation. This lack of

a trusted installation makes zk-STARK algorithms more reliable, especially in public and decentralized systems. Zk-STARK algorithms offer the ability to verify large data sets with high scalability. This feature makes them highly advantageous for applications operating with high data volumes. Zk-STARK algorithms use a structure based on hash functions and polynomial commitments and are considered more resilient to the threats of quantum computers. However, zk-STARK proofs can be longer than zk-SNARKs. This can lead to increased storage costs in some systems. However, zk-STARK algorithms have become a significant choice in financial systems, public infrastructures, and identity verification systems requiring security and transparency.

Unlike other ZKP algorithms, the Bulletproofs algorithm focuses specifically on secret information verification and range proofs. Bulletproofs, unlike zk-STARK, offers significant advantages in systems prioritizing privacy protection by not requiring a secure installation. In Bulletproofs algorithms, proof lengths increase logarithmically, resulting in data storage efficiency and an advantageous structure for systems with limited storage capacity. Mathematically, Bulletproofs is compatible with general arithmetic operations and, because it does not require a secure initial setup, is also applicable to standalone systems. While this algorithm has a lower verification speed compared to other ZKP algorithms, it can provide a cost-effective solution in systems with low transaction volumes. The Bulletproofs algorithm is frequently preferred, particularly in financial transactions, for confidential payment verification, and for areas requiring high data security.

In addition to its technological advantages, the application of ZKP methods offers profound implications for legal, ethical, and regulatory contexts. For instance, in the realm of digital identity, ZKPs can provide compliance with privacy laws such as the GDPR and CCPA by enabling data minimization—allowing entities to verify identity or credentials without actually accessing or storing personal data. This principle is especially crucial in cross-border data flows, where jurisdictional issues often complicate compliance. Furthermore, ZKP's ability to support selective disclosure is highly relevant for applications such as decentralized identity (DID) systems and verifiable credentials, where users can control what information they share and with whom.

The integration of zero-knowledge protocols into real-world systems is also facilitating the rise of privacy-preserving decentralized finance (DeFi) applications. With the growing adoption of blockchain technologies in financial markets, there is a parallel increase in the demand for confidentiality, particularly for institutional users. ZKPs are now being incorporated into layer-2 scaling solutions and privacy-focused networks to enable anonymous transactions, auditability, and secure compliance simultaneously. Examples include protocols like Tornado Cash and Aztec, which leverage ZKPs to anonymize Ethereum transactions while still maintaining verifiability and preventing double-spending.

Despite their many advantages, ZKP implementations also face technical and operational challenges. Computational overhead, especially in large-scale systems, can impact latency and scalability. Additionally, the complexity of ZKP protocol design requires a high level of cryptographic expertise, and any implementation errors can lead to critical vulnerabilities. Research into hardware acceleration, optimized proving systems, and universal trusted setups continues to evolve to address these limitations. Furthermore, standardization efforts, such as those led by the ZKProof community, aim to ensure interoperability and robust security practices across various ZKP applications.

As quantum computing continues to evolve, the necessity of quantum-resistant cryptographic solutions becomes even more pronounced. In this landscape, zero-knowledge proof systems—particularly those like zk-STARKs that utilize quantum-resistant hash functions—are poised to be central to the future of secure communications. By offering both theoretical and practical resistance to quantum attacks, ZKPs are not only complementary to post-quantum encryption methods but may also serve as standalone mechanisms for privacy-preserving computation. These properties highlight the strategic importance of ZKP research and development in future-proofing digital infrastructure.

Moreover, zero-knowledge proofs have also found a crucial role in enhancing trust and privacy within multi-party computation (MPC) systems. In MPC, multiple parties collaborate to compute a function over their private inputs without revealing them to each other. ZKP can serve as a verification layer that ensures each party behaves honestly during the computation, thus preventing data leakage or manipulation. This synergy between ZKP and MPC has opened new possibilities for privacy-preserving data analytics, collaborative AI training, and secure voting systems. By leveraging ZKPs in these contexts, it becomes possible to scale secure computations across untrusted environments, including edge devices and cloud infrastructures.

In addition, advancements in hardware and software tooling have started to make ZKP more accessible for real-world developers. Libraries such as ZoKrates, SnarkJS, and tools like Circom allow developers to write circuits and generate zero-knowledge proofs more efficiently. The emergence of zk-EVM (zero-knowledge Ethereum Virtual Machine) architectures is another leap forward, enabling Ethereum-compatible smart contracts to run with zero-knowledge verification layers natively. These developments significantly lower the entry barrier for developers and increase the potential for mass adoption of ZKP in mainstream applications. As tooling matures and becomes more performant, it is expected that zero-knowledge technologies will become integral to the design of privacy-aware and regulation-compliant digital systems.

In conclusion, zero-knowledge proof technologies are reshaping the foundations of digital trust, enabling secure and private computation in an increasingly interconnected and data-driven world. The unique ability of ZKPs to verify information without disclosure provides a compelling solution to the growing concerns of data privacy, quantum vulnerability, and system transparency. As advancements in quantum computing, blockchain technology, and cryptographic research converge, ZKP algorithms are expected to become indispensable components of next-generation information security frameworks, empowering users, enterprises, and governments to communicate and transact with confidence.

As a result, ZKP technologies play a significant role in information security by enabling the verification of confidential information without exposing it. It is anticipated that the innovative solutions offered by ZKP algorithms will become even more critical in meeting security and privacy requirements in the future, particularly in the face of the threats posed by quantum computers to cryptographic security. Given the continuous development of blockchain technologies, financial systems, and digital identity verification systems, ZKP algorithms are expected to offer promising solutions for ensuring security in these areas.

GENİŞLETİLMİŞ ÖZET

İletişim teknolojilerinin ilk ortaya çıkışından itibaren, mesajların üçüncü şahıslar tarafından ele geçirilme olasılığı, gizliliği ve veri güvenliğini tehdit eden temel bir sorun olarak ortaya çıkmıştır. Tarih boyunca bilgi güvenliğini sağlamak amacıyla çeşitli şifreleme yöntemleri geliştirilmiş, ancak bu yöntemler, iletişimin doğasını ve güvenlik gereksinimlerini karşılamada çeşitli sınırlamalarla karşı karşıya kalmıştır. Şifreleme yöntemleri, güvenilirliğini çoğunlukla karmaşık matematiksel problemlerin çözülme zorluklarına dayandırır; karmaşıklık seviyesi arttıkça, bu sistemlerin güvenliği de o ölçüde güçlü kabul edilir. Günümüzde kullanılan modern şifreleme algoritmaları, en güçlü klasik bilgisayarların dahi çözmekte zorlanacağı seviyelerde karmaşıklık barındırır. Ancak, kuantum bilgisayarların geliştirilmesi ve potansiyellerinin gün geçtikçe artması, mevcut şifreleme yöntemlerinin güvenliğini tehdit eden önemli bir etken haline gelmiştir. Kuantum bilgisayarlar, klasik bilgisayarlara kıyasla çok daha yüksek işlem hızlarına sahiptir ve teorik olarak, günümüzün en güvenilir kabul edilen şifreleme çözümlerini bile saniyeler içinde çözme kapasitesine ulaşabileceği öngörülmektedir. Bu nedenle, kuantum hesaplamanın güçlenmesi, geleneksel şifreleme yöntemlerinin güvenilirliğini sorgulamamıza yol açmış ve güvenlik gereksinimlerini karşılayabilecek yeni yaklaşımların geliştirilmesini zorunlu hale getirmiş durumdadır.

Bu paradigma değişimi, kriptografik güvenliğin temel kabullerini sorgulanabilir hale getirmiş ve gelecekteki bilgi güvenliği ihtiyaçlarını karşılayabilecek yeni yöntem ve yaklaşımların geliştirilmesini zorunlu kılmıştır. Kuantum hesaplamanın sunduğu bu yeni tehdit ortamı, verilerin güvenli ve gizli şekilde iletilmesini sağlamak üzere alternatif çözümlere yönelik araştırmaları hızlandırmıştır. Bu bağlamda, geleneksel şifreleme yöntemlerine ek olarak geliştirilen sıfır bilgi kanıtları (Zero-Knowledge Proof, ZKP), veri güvenliği için önemli bir yenilik olarak öne çıkmaktadır. Sıfır bilgi kanıtları, bir tarafın belirli bir bilgiye sahip olduğunu, ancak bu bilgiyi açığa çıkarmadan, yalnızca "bilginin doğruluğunu" kanıtlayarak karşı tarafa ispatlamasına olanak tanımaktadır. Bu teknik, üçüncü taraflarca bilginin ele geçirilme riskini en aza indirmekte ve kimlik doğrulama, finansal işlemler ve güvenli veri aktarımı gibi gizlilik gerektiren pek çok alanda yenilikçi bir çözüm sunmaktadır. ZKP teknolojisi ilk kez 1985 yılında Goldwasser, Micali ve Rackoff tarafından önerilmiş olup, özellikle gizli bilginin ifşa edilmesi riskine karşı güçlü bir koruma sağlamaktadır. Temelinde gizliliği koruyan ZKP, bilgi güvenliğine yönelik benzersiz bir yöntem sunarak, çeşitli güvenlik ihtiyaçlarına yanıt verebilmektedir.

ZKP protokolleri geleneksel olarak etkileşimli bir yapıya sahiptir; doğrulayıcı (verifier) ve ispatlayan (prover) arasında bilgi alışverişi gerektirmektedir. Bu bilgi alışverişi, doğrulamanın her seferinde yeniden başlatılmasını zorunlu kılmaktadır. Bu durum, her doğrulamanın farklı bir

iletişim süreci gerektirmesi nedeniyle pek çok senaryo için kullanım alanlarını sınırlayabilmektedir. Bu kısıtlamayı aşmak amacıyla 1988 yılında Feige, Fiat ve Shamir, ZKP protokolünü kimlik doğrulama amacıyla uygulayarak, etkileşimsiz sıfır bilgi kanıtları (NIZKP) kavramını ortaya koymuşlardır. Etkileşimsiz sıfır bilgi kanıtları, doğrulamanın yalnızca tek bir işlemde yapılabilmesini sağlamaktadır. Bu özelliği, özellikle blok zinciri gibi sürekli doğrulama gerektiren sistemlerde büyük bir avantaj sunmaktadır. Blok zinciri teknolojisinin hızlı yayılımı, güvenlik ve doğrulama süreçlerinin verimliliğini artırmaya yönelik olarak NIZKP'nin önemini daha da artırmıştır.

Günümüzde ZKP algoritmaları; güvenlik, verimlilik ve uygulama alanlarına göre üç ana kategoriye ayrılmaktadır: zk-SNARK, zk-STARK ve Bulletproofs. Bu algoritmalar, güvenlik düzeyleri, matematiksel temelleri ve uygulama alanları açısından önemli farklılıklar gösterir. Zk-SNARK algoritmaları, sıfır bilgi kanıtları için yaygın olarak tercih edilen bir seçenektir. Zk-SNARK'ların temel özellikleri arasında kısa ispat uzunlukları, hızlı doğrulama süreçleri ve geniş uygulama alanları yer alır. Bu algoritmalar, diğer sıfır bilgi kanıtlarına kıyasla oldukça kısa ispatlar üreterek veri depolama maliyetlerini azaltır ve büyük miktarda verinin doğrulanmasını gerektiren uygulamalarda avantaj sağlar. Ancak, zk-SNARK algoritmalarının temel dezavantajlarından biri, güvenilir bir kurulum süreci gerektirmeleridir. Bu güvenilir kurulum süreci, sistemin güvenliği için merkezi bir güven unsuruna bağlı olmasına neden olur. Kurulum sırasında kullanılan özel anahtarın açığa çıkması durumunda sistemin güvenliği tehdit altına girebilir. Matematiksel olarak, zk-SNARK algoritmaları, ispat üretimini ve doğrulama süreçlerini hızlandırmak amacıyla eliptik eğri kriptografisi gibi ileri düzey matematiksel yapılar kullanır. Bu özellikleri sayesinde özellikle Ethereum gibi blokzincir platformlarında ve akıllı sözleşmelerde yaygın olarak kullanılmaktadırlar. Ancak, zk-SNARK'ların yüksek hesaplama maliyeti, büyük veri işleme durumlarında performans düşüşüne neden olabilir ve bu durum bazı senaryolarda kullanımını sınırlayabilir.

Zk-SNARK algoritmaları, sıfır bilgi kanıtları arasında popüler bir seçenek olarak kabul edilmektedir. Zk-SNARK'ların temel özelliklerini kısa kanıt uzunlukları, hızlı doğrulama süreçleri ve geniş bir uygulama alanına sahip olmaları şeklinde sıralayabilmek mümkündür. Bu algoritmalar, diğer sıfır bilgi kanıtlarına göre oldukça kısa kanıtlar üreterek veri depolama maliyetlerini düşürmekte ve büyük veri miktarlarının doğrulanması gereken uygulamalarda avantaj sağlamaktadır. Ancak zk-SNARK algoritmalarının temel bir dezavantajını, güvenilir bir başlangıç (trusted setup) gerektirmesi olarak belirtebiliriz. Bu güvenilir kurulum süreci, sistemin güvenliği açısından merkezi bir güven unsuruna bağımlı hale gelmesine neden olmaktadır. Kurulum sırasında kullanılan gizli anahtarın açığa çıkması durumunda, sistemin güvenliği tehdit altında kalabilmektedir. Matematiksel olarak, zk-SNARK algoritmaları eliptik eğri kriptografisi gibi ileri düzey matematiksel yapıları kullanarak kanıt oluşturabilmekte ve doğrulama süreçlerini hızlandırmaktadırlar. Bu özellikleri, özellikle blok zincir platformları ve akıllı sözleşmelerde

(örneğin Ethereum gibi) yaygın olarak kullanılmasını sağlamaktadır. Ancak, zk-SNARK'ların yüksek hesaplama maliyeti büyük veriler üzerinde işlem yaparken performansın düşmesine neden olabilmekte ve bu durum bazı senaryolarda uygulamalar için sınırlayıcı olabilmektedir.

Zk-STARK algoritmaları, zk-SNARK algoritmalarının güvenilir kurulum gereksinimini ortadan kaldırmak amacıyla geliştirilmiştir ve güvenilir kurulum ihtiyacı olmayan bir yapı sunmaktadırlar. Güvenilir kurulum gerektirmemesi, zk-STARK algoritmalarını özellikle kamuya açık ve merkeziyetsiz sistemlerde daha güvenilir hale getirmektedir. Zk-STARK algoritmaları büyük veri setlerini yüksek ölçeklenebilirlikle doğrulama imkânı sunmaktadırlar. Bu özellikleri ile, yüksek veri hacmi ile çalışan uygulamalar için oldukça avantajlı bir duruma gelmektedirler. Zk-STARK algoritmaları hash fonksiyonları ve polinom taahhütlerine dayalı bir yapı kullanmakta ve kuantum bilgisayarların tehditlerine karşı daha dayanıklı olarak kabul edilmektedir. Buna rağmen, zk-STARK kanıtları, zk-SNARK'lara kıyasla daha uzun olabilmektedir. Bu durum bazı sistemlerde depolama maliyetlerinin artmasına neden olabilmektedir. Bununla birlikte, güvenlik ve şeffaflık gerektiren finansal sistemlerde, kamu altyapılarında ve kimlik doğrulama sistemlerinde zk-STARK algoritmaları önemli bir tercih sebebi haline gelmiş durumdadır.

Bulletproofs algoritması, diğer ZKP algoritmalarından farklı olarak, özellikle gizli bilgi doğrulama ve aralık kanıtları (range proof) üzerinde yoğunlaşan bir algoritma türü olarak karşımıza çıkmaktadır. Bulletproofs, zk-STARK gibi güvenilir kurulum gerektirmemesiyle gizliliğin korunmasını ön planda tutan sistemlerde önemli avantajlar sağlamaktadır. Bulletproofs algoritmalarında kanıt uzunlukları logaritmik olarak artış göstermektedir, Bu da veri depolama açısından verimlilik sağlamakta ve sınırlı depolama kapasitesine sahip sistemler için avantajlı bir yapının ortaya çıkmasına imkan vermektedir. Matematiksel olarak, Bulletproofs genel aritmetik işlemlerle uyumludur ve güvenilir bir başlangıç kurulumuna ihtiyaç duymadığından bağımsız sistemlerede uygulanabilir yapıdadır. Bu algoritma, diğer ZKP algoritmalarına kıyasla daha düşük doğrulama hızına sahip olmakla birlikte düşük işlem hacmine sahip sistemlerde maliyet etkin bir çözüm sunabilmektedir. Bulletproofs algoritması, özellikle finansal işlemlerde, gizli ödeme doğrulama ve yüksek veri güvenliği gerektiren alanlarda sıklıkla tercih edilmektedir.

Teknolojik avantajlarının ötesinde, ZKP yöntemlerinin uygulanması, hukuki, etik ve düzenleyici bağlamlarda da derin etkiler yaratmaktadır. Örneğin dijital kimlik alanında ZKP'ler, kimliği veya kimlik bilgilerini doğrulamak için verinin doğrudan erişilmesine ya da saklanmasına gerek kalmadan doğrulama yapılmasına olanak tanıyarak GDPR ve CCPA gibi gizlilik yasalarına uyumu destekleyebilir. Bu ilke, özellikle yargı alanlarının veri akışlarını zorlaştırdığı sınır ötesi veri aktarımlarında kritik öneme sahiptir. Ayrıca ZKP'nin seçici açıklama (selective disclosure) yeteneği, kullanıcıların hangi bilgileri kiminle paylaşacakları üzerinde tam denetim sağladığı

merkeziyetsiz kimlik sistemleri (DID) ve doğrulanabilir kimlik bilgileri gibi uygulamalar için büyük önem taşımaktadır.

Sıfır bilgi protokollerinin gerçek dünya sistemlerine entegrasyonu, gizliliği koruyan merkeziyetsiz finans (DeFi) uygulamalarının yükselişini de kolaylaştırmaktadır. Blokzincir teknolojilerinin finans piyasalarında giderek daha fazla benimsenmesiyle, özellikle kurumsal kullanıcılar açısından gizlilik talepleri artmaktadır. ZKP'ler artık katman-2 ölçeklenebilirlik çözümlerine ve gizlilik odaklı ağlara entegre edilerek anonim işlemler, denetlenebilirlik ve uyumun aynı anda sağlanmasını mümkün kılmaktadır. Tornado Cash ve Aztec gibi protokoller, Ethereum işlemlerini anonimleştirirken aynı zamanda doğrulanabilirlik ve çifte harcamayı engelleme gibi önemli işlevleri sürdürmek için ZKP teknolojisinden faydalanmaktadır.

Pek çok avantajına rağmen, ZKP uygulamaları teknik ve operasyonel zorluklarla da karşı karşıyadır. Özellikle büyük ölçekli sistemlerdeki hesaplama yükü, gecikme sürelerini ve ölçeklenebilirliği olumsuz etkileyebilir. Ayrıca, ZKP protokollerinin tasarımındaki karmaşıklık, yüksek düzeyde kriptografik uzmanlık gerektirir ve herhangi bir uygulama hatası, ciddi güvenlik açıklarına yol açabilir. Donanım hızlandırma, optimize edilmiş ispat sistemleri ve evrensel güvenilir kurulumlar üzerine yapılan araştırmalar bu sınırlamaları aşmayı amaçlamaktadır. Ayrıca ZKProof topluluğunun öncülüğünde yürütülen standartlaştırma çalışmaları, farklı ZKP uygulamaları arasında birlikte çalışabilirlik ve sağlam güvenlik uygulamalarının sağlanmasını hedeflemektedir.

Kuantum bilişimin gelişimi devam ettikçe, kuantuma dayanıklı kriptografik çözümlere olan ihtiyaç daha da belirgin hale gelmektedir. Bu bağlamda, özellikle kuantum saldırılarına dayanıklı hash fonksiyonları kullanan zk-STARK gibi sistemler başta olmak üzere, sıfır bilgi kanıt sistemleri geleceğin güvenli iletişim altyapısında merkezi bir rol oynamaya adaydır. Hem teorik hem de pratik anlamda kuantum tehditlerine karşı direnç sunan ZKP'ler, sadece kuantum sonrası şifreleme yöntemlerini tamamlayıcı nitelikte değil, aynı zamanda gizliliği koruyan hesaplamalar için bağımsız mekanizmalar olarak da kullanılabilir. Bu özellikler, ZKP araştırma ve geliştirme çalışmalarını dijital altyapının geleceğe hazırlanmasında stratejik olarak önemli bir konuma getirmektedir.

Ayrıca, sıfır bilgi kanıtları çok taraflı hesaplama (multi-party computation, MPC) sistemlerinde güven ve gizliliği artırma açısından da kritik bir rol oynamaktadır. MPC'de birden fazla taraf, özel verilerini birbirine açıklamadan ortak bir hesaplama gerçekleştirir. ZKP, her bir tarafın hesaplamaya dürüst şekilde katıldığını doğrulamak için bir doğrulama katmanı işlevi görerek veri sızıntısı veya manipülasyon riskini önleyebilir. ZKP ile MPC'nin birleşimi, gizliliği koruyan veri analitiği, ortak yapay zekâ eğitimi ve güvenli oylama sistemleri gibi yeni uygulama alanlarının önünü açmıştır. Bu bağlamda, ZKP'lerin uç cihazlar ve bulut altyapıları gibi

güvenilmeyen ortamlarda da güvenli hesaplamaları ölçeklendirmek için kullanılması mümkün hâle gelmiştir.

Ayrıca, donanım ve yazılım araçlarındaki ilerlemeler, ZKP teknolojisini gerçek dünya geliştiricileri için daha erişilebilir hale getirmeye başlamıştır. ZoKrates, SnarkJS gibi kütüphaneler ve Circom gibi araçlar, geliştiricilerin devreler yazmasını ve sıfır bilgi kanıtları üretmesini daha verimli hale getirmektedir. ZK-EVM (Zero-Knowledge Ethereum Virtual Machine) gibi yapılar da Ethereum ile uyumlu akıllı sözleşmelerin doğrudan sıfır bilgi doğrulama katmanlarıyla çalışmasına olanak tanıyarak önemli bir sıçrama yaratmaktadır. Bu gelişmeler, geliştirici bariyerlerini önemli ölçüde azaltmakta ve ZKP'nin ana akım uygulamalarda yaygınlaşma potansiyelini artırmaktadır. Araçlar olgunlaştıkça ve performansları arttıkça, sıfır bilgi teknolojilerinin gizlilik odaklı ve yasal düzenlemelere uyumlu dijital sistemlerin tasarımında temel bir bileşen haline gelmesi beklenmektedir.

Sonuç olarak, ZKP teknolojileri, gizli bilgilerin ifşa edilmeden doğrulanabilmesi sayesinde bilgi güvenliği alanında önemli bir rol oynamaktadır. Özellikle kuantum bilgisayarların şifreleme güvenliğine yönelik tehditleri karşısında ZKP algoritmalarının önerdiği yenilikçi çözümlerin, gelecekte güvenlik ve gizlilik gereksinimlerini karşılamada çok daha kritik bir önem kazanacağı öngörülmektedir. Blok zincir teknolojilerinin, finansal sistemlerin ve dijital kimlik doğrulama sistemlerinin sürekli geliştiği günümüzde, ZKP algoritmalarının sayılan alanlarda güvenlik sağlamak adına umut verici çözümler sunacağı düşünülmektedir.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, Assoc. Prof. Dr. Fatih ABUT, for his continuous guidance, invaluable advice, patience and constant encouragement throughout the course of this study. His insightful feedback and unwavering support have been instrumental in the successful completion of this thesis.

I would also like to extend my sincere thanks to Assoc. Prof. Dr. Fatih ABUT, Asst. Prof. Dr. Barış ATA and Assoc. Prof. Dr. Ali İNAN for their valuable suggestions, constructive criticism and contributions which have significantly enriched this thesis.

My special thanks go to the authorities of Adana Çukurova University Balcalı Hospital for their support in the procurement process of various hardware components necessary for the implementation of the algorithms. I am also grateful to the supervisors and staff of the IT Unit for their valuable assistance in the selection and procurement processes.

In addition, I would like to express my sincere appreciation to the supervisors and staff of the IT Unit at Adana Alparslan Türkeş Science and Technology University for their contributions during the installation and updating processes of the virtual machines required for executing the algorithms.

Finally, I would like to thank my family, especially my daughter and my wife for their never-ending support, encouragement and patience throughout my life and of course my dear friends and colleagues with whom I had the pleasure of working at Balcalı Hospital for their support and patience.

LIST OF TABLES

LIST OF FIGURES

SYMBOLS AND ABBREVIATIONS

ZKP	:	Zero Knowledge Proof
MA / AM	:	Merlin-Arthur or Arthur-Merlin
SNARK	:	Succinct Non-Interactive Argument of Knowledge
SNARG	:	Succinct Non-Interactive Argument
STARK	:	Scalable Transparent Arguments of Knowledge
NIZKP	:	Non-Interactive Zero Knowledge Proof
R1CS	:	Rank-1 Constraint Satisfiability
AHP	:	Algebraic Holographic Proof
AVX	:	Advanced Vector Extensions
CRS	:	Common Reference String
CRHF	:	Collision Resistant Hash Function

1. INTRODUCTION

1.1. Motivation

Since the advent of verbal or written communication, one of the most critical issues in communication has been the potential interception of messages by third parties. The issue of eavesdropping or interception of transmitted data represents the most significant challenge to the confidentiality of communication or more specifically, its security. The solution employed since ancient times for this predicament is the encryption of communication or more precisely, the encryption of transmitted data (Mulder et al., 2023). In conclusion, the unfeasibility of finding a solution within an acceptable time serves to safeguard the encryption method. The encryption methods currently in use are considered reliable because, even for the most powerful computing machines currently available, it would take several years of dedicated study to crack them. Nevertheless, this acceptance may become invalidated by the advent of quantum computing systems (Ambainis et al., 2014).

In the historical context, the first known ZKP protocol was introduced by Goldwasser, Micali and Rackoff in 1985. Three years later, in 1988, Feige, Fiat and Shamir conducted a study on the customized application of the existing protocol for authentication. In this study, algorithms were introduced to enable the party seeking to prove their identity to verify it without disclosing confidential information to the other party (Beydemir, 2018).

Zero-knowledge proofs represent a relatively new cryptographic innovation originally introduced by Goldwasser et al. They allow a prover to prove certain information to a verifier without revealing the details of that information. However, traditional ZKPs are interactive and require multiple exchanges between the prover and the verifier to establish or deny trust. Additionally, it is not possible for this evidence to be verified by other parties without initiating new interactions, which limits practical applications. To address this limitation, Blum et al. proposed non-interactive zero-knowledge proofs (NIZKPs). NIZKPs enable a validator to validate a claim in a single interaction, allowing subsequent validators to verify the same claim at a later point in time (Oude Roelink et al., 2024).

In a ZKP protocol, a prover provides a verifier with assurance regarding the veracity of a statement without disclosing any information other than its validity. It is guaranteed that a malicious prover cannot deceive the verifier into accepting a false statement (soundness). A more relaxed concept, zero-knowledge arguments, connects the capabilities of the malicious prover computationally to polynomial strategies (computational soundness). An evidentiary may also demonstrate that he knows a witness who meets the testimony. The zero-knowledge feature ensures that the evidence does not reveal or compromise the confidentiality of the witness. For general

expressions, NIZKP protocols are possible only under the assumption of a common reference string (CRS), which must be known by both the prover and the verifier (Panait & Olimid, 2021).

The advent of quantum computing systems promises to revolutionize the landscape of computational speed, potentially offering calculation speeds that are thousands of times faster than the methods of computation currently in use (Ambainis et al., 2014). Consequently, the solution to an encryption method estimated to take decades to crack today can potentially be found in seconds. The immense calculation speeds offered by quantum computing systems render many encryption methods currently considered secure due to their complexity vulnerable. At present, quantum computing systems are not yet positioned to serve as an alternative to traditional computing systems. Nevertheless, preliminary studies on these systems indicate their estimated potential, suggesting the possibility of applications far beyond those of traditional systems (Mulder et al., 2023).

In light of the initial findings obtained, it can be asserted that no existing encryption system is secure against quantum computing systems. Consequently, it can be posited that the development of significantly more robust encryption systems is imperative to address this issue. Nevertheless, this approach is not dissimilar to engaging in a war where victory is unlikely. This is because it can be predicted that the rate of development of any of the traditional computing tools at our disposal will not be able to compete with the rate of development of quantum computing systems. What are the potential solutions to this situation? If there were a method of demonstrating that we possessed knowledge without disclosing it, then the interception or capture of the evidence would not be a concern. The primary concern is that confidential information is not shared in a mutually transparent manner. Consequently, there is no concern about its security. Fortunately, there is a method that can meet this need. ZKP methods can be employed to meet this need. This method was first described by Goldwasser, Micali and Rackoff in 1985 (Goldwasser et al., 1985).

1.2. Problem Statement

With the increasing importance of privacy in digital interactions, ensuring secure and privacy-preserving authentication has become a critical challenge. NIZKPs provide a promising solution by allowing one party to prove knowledge of certain information without revealing the information itself. However, despite their theoretical potential, the practical implementation and efficiency of different NIZKP schemes remain an open issue.

Existing studies (Capko et al., 2022; T. Chen et al., 2022; El-Hajj & Oude Roelink, 2024; Gong et al., 2022; Khovratovich, 2018; Nitulescu, 2020; Oude Roelink et al., 2024; Panait & Olimid, 2021) often focus on specific NIZKP variants such as zk-SNARKs, zk-STARKs or Bulletproofs without systematically comparing their strengths and limitations across various real-world applications. Moreover, while performance metrics such as proof generation time,

verification time and proof size are critical for practical deployment, there is a lack of comprehensive studies that empirically evaluate these aspects across different NIZKP algorithms. This research aims to address these gaps by:

- Conducting a systematic literature review to explore the evolution and applications of major NIZKP schemes.
- Implementing representative NIZKP schemes from each category (zk-SNARKs, zk-STARKs and Bulletproofs) and analyzing their performance.
- Evaluating key performance metrics such as proof size, proof generation time and verification time to provide insights into their suitability for different use cases.
- Offering a comparative assessment of real-world applications to guide researchers and practitioners in selecting the most appropriate NIZKP scheme for their specific needs.

1.3. The Goal of the Study

The goal of this thesis is to analyze and compare different NIZKP schemes in terms of their efficiency, usability and real-world applications. Specifically, the study aims to evaluate zk-SNARKs, zk-STARKs and Bulletproofs based on key performance metrics such as proof size, proof generation time and verification time.

The study seeks to:

- Provide a comprehensive review of the evolution and application areas of major NIZKP schemes.
- Implement representative NIZKP schemes from each category and analyze their efficiency.
- Compare the performance of these schemes using relevant metrics such as computational overhead, security guarantees and scalability.
- Offer insights into the suitability of different NIZKP approaches for various privacy-preserving applications.

To achieve the stated goal, this thesis addresses the following research questions:

- What are the fundamental differences between zk-SNARKs, zk-STARKs and Bulletproofs in terms of cryptographic structure and operational efficiency?
- How do these NIZKP schemes perform in real-world applications when evaluated based on proof size, proof generation time and verification time?

- Which NIZKP scheme provides the best trade-off between efficiency and security for different privacy-preserving use cases?
- What are the practical implications of these findings for researchers and developers working on secure authentication and privacy-preserving technologies?

By addressing these questions, this study aims to provide a detailed understanding of the strengths and limitations of different ZKP approaches, thereby contributing to the field of cryptography and privacy-preserving authentication.

1.4. Structure of the Thesis

The rest of this thesis is structured as follows. Chapter 2 presents a comprehensive literature review on ZKPs organized under specific headings. The classification of NIZKP schemes is carried out according to basic criteria: whether they require a trusted setup, the nature of the setup (public or a fully trusted third party), the mathematical submodels used and their resistance to quantum attacks. By structuring the literature review in this way, this chapter provides a systematic understanding of the evolution, strengths and limitations of different ZKP approaches.

Chapter 3 explores the theoretical foundations of the modern NIZKP algorithms analyzed in this work. A comprehensive description of the mathematical foundations, proof-building mechanisms and security properties of zk-SNARKs, Bulletproofs and zk-STARKs is provided. First, NIZKP schemes that require pre-setup are reviewed and the evolution of zk-SNARKs, their dependence on a trusted third party and their verification processes are explained in detail. The explanation then moves on to Bulletproofs, which eliminates the need for a trusted setup, highlighting their compactness and advantages in blockchain applications. Finally, zk-STARKs are analyzed in terms of their insecure design and resistance to quantum attacks. These explanations form the technical framework for the cryptographic protocols examined in this study.

Chapter 4 aims to explain the selection process, research methodologies and analytical approaches of the software and hardware platforms used in the study. Initially, the theoretical framework of the study is determined and the location of the study is clarified by reviewing existing studies in the literature. In the selection of the software platform, open source and commercial software were evaluated in order to increase the performance and flexibility of the system. Components such as programming languages, operating systems and virtualization solutions were discussed. The features of the selected software were explained and the reasons for these choices were highlighted. It was explained that the selection of the hardware platform was determined by a number of criteria, including but not limited to processing power, memory capacity and the capacity of other system resources. These criteria were selected in accordance with the requirements of the study and the rationale behind the choice for different hardware

alternatives was explained appropriately. The next section provides a comprehensive overview of the data collection methods and analysis processes used in the study and a detailed description of the methodological approach undertaken. The computational models, test cases and measurement techniques used are explained and a general framework is provided on how the obtained results are evaluated and interpreted. The holistic progress of the study from both theoretical and practical perspectives is thus explained.

Chapter 5 focuses on the execution of the selected algorithms and presents the results obtained from running these schemes under various scenarios. Performance metrics such as proof size, proof generation time and verification time are analyzed for each scheme. Different test cases are considered to evaluate the applicability of zk-SNARKs, zk-STARKs and Bulletproofs in real-world scenarios. A scenario-based analysis is conducted to evaluate their effectiveness under different conditions including varying system resources and cryptographic parameters. Chapter 5 also presents a comparative analysis of the obtained results and highlights the strengths and weaknesses of each NIZKP scheme. The chapter presents a general comparison of the schemes based on computational efficiency, security properties, scalability and practical usability. The results are systematically compared with findings from previous studies, providing a deeper understanding of how different NIZKP approaches perform in practice.

Chapter 6 addresses the research gap identified throughout this work and suggests potential directions for future work. Open challenges in ZKP research are discussed, particularly in terms of scalability, post-quantum security and real-world deployment. The chapter concludes by outlining possible improvements to existing schemes and examining emerging trends in privacy-preserving cryptographic techniques.

2. LITERATURE REVIEW

A review of the literature reveals that studies on ZKP schemes can be classified into two main categories based on the necessity of a preliminary setup phase. This is illustrated in Figure 2.1. It can be observed that the group designated as zk-SNARK, which necessitates a preliminary setup phase, can be classified into three subcategories: publicly accessible but no reference string required (transparent), publicly accessible (universal) and reliable (trusted). Schemes that do not necessitate a preliminary setup phase can be examined under three subheadings: These schemes are Bulletproofs, zk-STARK and zk-VPD. In the final section of the literature review, studies examining the practical applications of the theoretically analyzed schemes are grouped and analyzed under four subheadings: finance, medical, business and general applications.

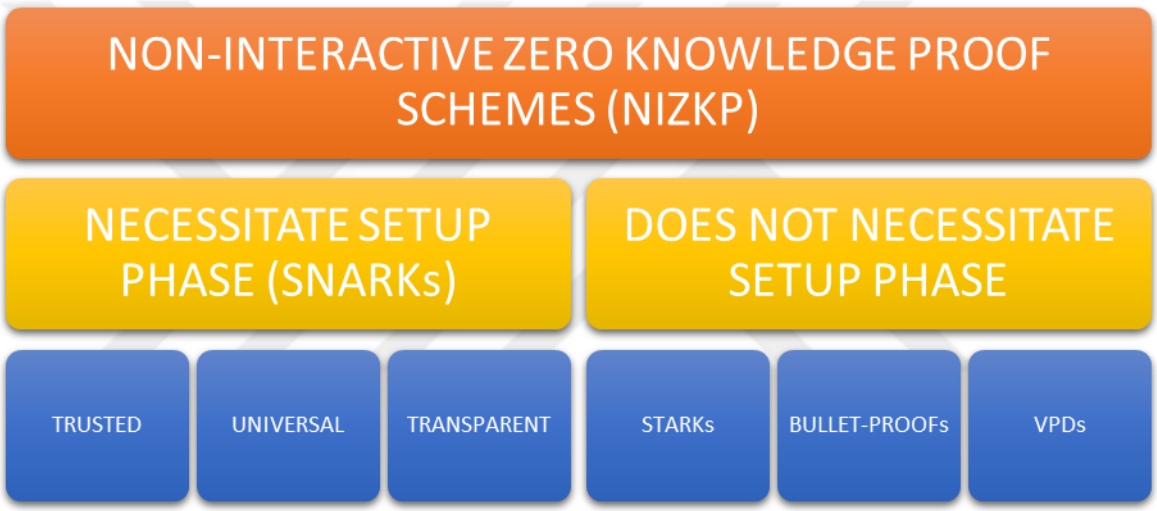


Figure 2.1. Classification of Zero Knowledge Proof Schemes.

2.1. Setup-Dependent ZKP Schemes (zk-SNARKs)

In the field of cryptography, a proof system enables a prover to persuade a verifier of the veracity of a given statement. Non-interactive proofs, such as SNARKs, permit the efficient, concise and secure verification of claims in a single step, thereby protecting against spoofing by computationally limited adversaries. This section presents a review of the existing work on the various types of zk-SNARK schemes.

2.1.1. Trusted Setup-Based Schemes

(Chase et al., 2017) put forth a novel approach for generating signatures in the post-quantum era. Their work focuses on enhancing ZKP systems within both classical and post-quantum cryptographic frameworks. The principal objective of their research is to enhance the efficiency of interactive honest verifier ZKP systems (Σ-protocols) by employing symmetric key primitives for post-quantum security. This study makes a contribution to the existing literature by

introducing ZKB++, an optimised Sigma protocol based on ZKBoo, which validates statements over general circuits with greater efficiency. In particular, ZKBoo++ reduces the size of the proof transcript by half without increasing the computational costs, resulting in more compact ZKPs. Furthermore, the paper presents a method for implementing Unruh's general transformation within the Quantum Random Oracle Model (QROM), marking the first known application of this transform in an efficient signature scheme. The authors put forth two post-quantum signature schemes, designated "Fish" and "Picnic." The Fish scheme employs the Fiat-Shamir transform with ZKBoo and is secure in the Random Oracle Model (ROM), while the Picnic scheme uses an optimised Unruh transform with ZKBoo++ and is secure in QROM. Overall, ZKBoo++ not only halves the proof size relative to ZKBoo but also enhances the feasibility of post-quantum signature schemes by minimising the overhead traditionally associated with Unruh's transformation.

(Parno et al., 2016) introduced Pinocchio, a system designed to simplify the verification of general computations based on cryptographic assumptions. To use Pinocchio, the client generated an evaluation key, proportional to the target computation, which the worker then used to perform the computation and produce a 288-byte proof of correctness. This proof maintained a constant size, regardless of the computation's complexity, enabling public verification through a dedicated verification key. Pinocchio's computational complexity was well-defined, with key setup and proof generation time scaling linearly with the computation size and verification time scaling linearly with input and output sizes. The system also supported zero-knowledge verifiable computation, allowing the prover to validate private inputs without disclosing them. Compared to prior methods, Pinocchio reduced verification time by five to seven times and proof generation time by 19 to 60 times. It maintained a compact proof size similar to an RSA-2048 signature. The inclusion of zero-knowledge features resulted in a minimal increase in overhead, with key generation taking just 213 μs and proof generation time increasing by only 0.1%. By integrating quadratic programming and an efficient cryptographic protocol, Pinocchio achieved both practical and asymptotic efficiency improvements, reducing key and proof generation costs by 60% and advancing computation signing protocols.

(Costello et al., 2015) introduced Geppetto, a system designed to improve prover efficiency and flexibility in verifiable computation. It employed novel techniques, including MultiQAPs for state-sharing, efficient cryptographic embedding’s, limited bootstrapping and energy-saving circuits. Geppetto reduced prover overhead by up to two orders of magnitude when sharing state across MultiQAPs or within single computations, such as in MapReduce and also supported the verification of proprietary algorithms. Building on Pinocchio's QAP-based architecture, Geppetto addressed performance issues related to key size and generation times by optimizing state linking with MultiQAPs, conserving time and memory. Geppetto also incorporated fixed-size proofs achieved through limited bootstrapping, consolidating multiple proofs efficiently. Energy-saving

circuits aligned prover costs with execution time, reducing worst-case costs. Geppetto's performance showed a 34 to 77 times improvement in key and proof generation compared to traditional methods. Further optimizations using a limited bootstrapping approach based on the Barreto-Naehrig curve led to significant performance gains, making verifiable computing more practical. The Geppetto compiler offered developers greater flexibility and performance, facilitating the integration of verifiable computation into real-world applications.

2.1.2. Universal or Transparent Setup-Based Schemes

In this section, schemes that have universal setup or transparent setup will be examined. Universal setups are characterized by their use of a CRS, which, unlike trusted setups, can be reused across multiple applications or communications once established. Transparent setups, on the other hand, eliminate the need for a CRS and instead rely on alternative mechanisms such as block chain technology or cryptographic hash functions.

(Chen et al., 2023) introduced HyperPlonk in their study, an enhanced version of the Plonk proof system, optimized for a Boolean hypercube using multiple linear polynomial commitments. Plonk, a widely utilized non-interactive proof system, supports both low-order special gates and search gates but faces performance bottlenecks when generating proofs for large circuits. These bottlenecks stem from the substantial Fast Fourier Transform (FFT) required, which becomes problematic for circuits larger than size 2^{20}. HyperPlonk overcame this limitation by eliminating the need for FFT during proof generation, allowing it to efficiently handle a greater number of specialized high-order gates without increasing prover runtime. This modification led to a significant reduction in proof generation time. HyperPlonk further improved proof efficiency by refining two advanced constructions: Orion and Virgo. Orion reduced the opening proof size to under 10KB, achieving nearly a 1000-fold improvement compared to previous methods. Virgo, on the other hand, simplified and shortened the FRI-based opening proof, making it more efficient. These innovations preserved the original flexibility of the Plonk system while significantly enhancing its performance for larger circuits. As a result, HyperPlonk provided an optimized solution for generating proofs for complex circuits, offering a practical improvement over the standard Plonk system.

(Xie et al., 2022) concentrate on enhancing the efficacy of ZKP by addressing pivotal performance metrics, namely prover time, proof size and verifier time. The study identifies the challenge of high overhead in proof generation time, particularly in schemes that use arithmetic circuits and seeks to develop a ZKP system with a proof time of O(N) and a proof size of polylog(N). The paper introduces Orion, a novel ZKP scheme that achieves O(N) proof time and $O(\log^2 N)$ proof size for field operations and hash functions. The authors enhance efficiency through the introduction of a new structure for linear-time encodable codes and a "code-switching"

reduction strategy that minimises proof size. The efficacy of Orion was evaluated through experimentation on a 2^{20}-gate circuit, wherein a proof time of 3.09 seconds, a proof size of 1.5 MB and a verification time of 70 ms were achieved. Orion is demonstrably more efficient than existing schemes such as Brakedown, Aurora and Ligero, offering significantly faster proving times and smaller proof sizes. It is noteworthy that Orion's proof size is 6.5 times smaller than that of Brakedown and 12.5 times smaller than that of Ligero for $N = 2^{20}$. Furthermore, it is post-quantum secure and non-interactive through the Fiat-Shamir heuristic. This advancement demonstrates the effectiveness of the proposed code substitution method in improving ZKP performance.

(Weng et al., 2021) put forth a novel protocol for fixed-round interactive ZKPs, addressing the challenges of high prover overhead and communication complexity inherent to existing systems. It is noted that a considerable number of existing ZK solutions are afflicted by either high memory usage or excessive communication complexity, particularly in the case of circuits that involve random logic or arithmetic. The new protocol, designated Wolverine, offers a substantial enhancement by providing an efficient prover with optimal memory usage and diminished communication overhead, in comparison to other protocols with analogous memory efficiency. The runtime and memory requirements of Wolverine are linear, thereby ensuring scalability for large circuits. The protocol reduces communication to approximately 9 bits per gate for Boolean circuits and 2-4 field elements per gate for arithmetic circuits, thereby conferring substantial efficiency benefits over prior systems. To illustrate, in Boolean terms, Wolverine necessitates 15 times less communication and has a 11-second runtime, in comparison to 400 seconds for ZKGC (Heath & Kolesnikov, 2020). In arithmetic circuits, the performance of Wolverine is approximately five times slower than that of Virgo (Zhang et al., 2020). However, it utilises only 3% of the memory required by Virgo, making it an efficient choice for larger circuits. Furthermore, the authors introduce an enhanced subdomain vector oblivious linear evaluation (sVOLE) protocol and demonstrate the scalability of Wolverine for larger circuits. In conclusion, Wolverine represents a substantial advancement in memory-efficient ZK protocols, particularly for applications that necessitate low-bandwidth communication.

(Yang et al., 2021) investigated the challenges of scalability in NIZKPs, with a particular focus on the memory overhead required to generate proofs for circuits comprising a large number of gates. It is noted that previous studies have concentrated on reducing the size of the proofs in question, but have frequently encountered issues of scalability due to the fact that the memory requirement is proportional to the size of the circuit in question. In response, they put forth new fixed-round ZK protocols, designed to enhance efficiency under memory constraints while also enabling sublinear amortized communication for arithmetic circuits. The aforementioned protocols permit a sub-linear communication complexity that is independent of the number of multiplications. Instead, it is dependent on the number of variables and the distinct links in the

highest-degree polynomials. The enhanced ZKP protocol guarantees that both communication size and proof generation are proportional to the input size, rather than the number of multiplications. Authors introduced QuickSilver, their proposed ZK protocol and demonstrated its superior performance in comparison to previous protocols. In the Boolean setting, QuickSilver demonstrated a six-fold reduction in computation time and a seven-fold reduction in communication overhead compared to Wolverine. In the context of arithmetic circuits, QuickSilver demonstrated superior performance compared to Wolverine and Mac'n'Cheese (Baum et al., 2021). It offered at least a seven-fold improvement in computation and a three-to-four-fold improvement in communication.

(Boneh et al., 2021) concentrated on Polynomial Commitment Schemes (PCS) and their pivotal function in the construction of SNARKs. The objective of this thesis is to illustrate how efficient PCS can facilitate sublinear proof verification and be employed in the construction of SNARKs with analogous security and efficiency properties. This paper made a significant contribution to the existing literature by presenting the Halo protocol, which introduces the first proof-carrying data (PCD) construction derived from an inefficient proof system. This is achieved by merging Sonic's PIOP and Bulletproofs' PCS. Authors presented a concrete instantiation of the Fiat-Shamir transformation in a non-black-box manner, demonstrating how PCDs can be constructed from any homomorphic PCS, even when the evaluation proofs are neither concise nor efficient. The results demonstrated that Halo is capable of extending to any PCS with the ability to aggregate linear combinations of commitments into a concise commitment. This novel methodology opened up new avenues for SNARK and PCD configurations that were previously unexplored in the literature, providing a foundation for future developments in the field.

(Setty, 2020) introduced Spartan, a new family of zk-SNARKs designed for proving the satisfiability of NP expressions expressed in rank-1 constraint satisfiability (R1CS). The focus is on achieving sublinear verification costs for NP expressions of arbitrary complexity and on providing time-optimal provers. The primary contribution is the development of Spartan as the inaugural transparent zk-SNARK to attain these sublinear verification costs, utilising a pioneering Arthur-Merlin interactive knowledge argument. Additionally, the paper presented three novel techniques for enhancing the efficiency of the sum-check protocol, which serves as the foundation for Spartan's design. The approach taken by Spartan, which utilises the SpartanKE polynomial commitment scheme, differs from previous schemes such as Libra in that it supports randomised R1CS instances and achieves sublinear verification costs without increasing circuit depth. Empirical evidence indicated that the prover in Spartan is 36 to 152 times faster, with proof sizes 1.2 to 416 times shorter than the baseline and verification costs 3.6 to 1326 times lower. Although Bulletproofs produces shorter proofs, Spartan is more efficient in terms of verification speed.

(Kosba et al., 2020) addressed the challenges associated with the deployment of zk-SNARKs, with a particular focus on the generation of concise proofs and the verification process. The study identified a significant challenge in the current approach to ZKP systems, which relies on a dependable key generation phase for varied computations. This reliance can have hindered the practicality of existing ZKP systems. Although universal circuit generators have been proposed as a means of eliminating the need for per-computation pre-processing, they were not a practical solution for real-world applications due to their detrimental impact on prover performance. To addressed this challenge, the authors proposed a novel zk-SNARK system tailored for randomized algorithms. The new system circumvented the encoding of random number generation within the arithmetic circuit, thereby facilitated more efficient prover times. They put forth a universal circuit that is capable of accepting any arithmetic circuit with a finite number of operations and performing randomized checks for consistency. This circuit demonstrated a linear scaling in circuit size with the number of operations, in contrast to the quasi-linear behaviour observed in other universal circuits. Authors presented MIRAGE, a universal zk-SNARK with highly concise proofs, which introduces a single additional element compared to Groth's per-circuit pre-processing zk-SNARK. Authors demonstrated that universal circuits can be made practical through a modified zk-SNARK protocol, random controls and efficient circuit engineering, thereby overcoming the limitations of previous methodologies.

(Chiesa, Hu, et al., 2020) presented a methodology for the construction of pre-processing zk-SNARKs with a universal and updatable structured reference string (SRS). The approach employed a novel application of the holographic method proposed by Babai et al. (STOC 1991), which enables rapid verification when the expression to be verified is provided in coded form. The proposed framework, designated MARLIN, attained a pre-processing zk-SNARK with a linear dimension for the SRS and a fixed dimension for the arguments. In comparison to previous state-of-the-art methods, such as Sonic (Maller et al., CCS 2019), MARLIN demonstrated enhanced efficiency across all parameters. Proofs are at least twice as fast and verifications three times faster, despite the reduction in SRS and argument sizes. MARLIN is most efficient when instantiated in the algebraic group model and can also be realised under concrete knowledge assumptions. The methodology's fundamental component is an algebraic holographic proof (AHP) for R1CS, which attains linear proof length and constant query complexity. This work addressed the discrepancy in efficiency between universal SRS and circuit-specific SRS, providing a general methodology for constructing concise interactive arguments that can be made non-interactive through the Fiat-Shamir transformation. The enhancements introduced by MARLIN not only reduce the argument size and the verifier's time complexity by over three times, but also significantly enhanced the prover's efficiency, improving its time complexity by over a factor of 10 compared to previous methods, thereby ensuring more practical results.

(Bhadauria et al., 2020) introduced Ligero++, a novel transparent zero-knowledge argument system that combines the most advantageous features of the Ligero and Aurora systems to achieve optimal proof efficiency and conciseness. The objective of this work is to addressed the limitations of transparent sublinear zero-knowledge interactive oracle proofs (IOPs) while ensuring that the system remains resistant to quantum attacks and does not require a trusted setup. Authors built on prior research on IOPs and probabilistically checkable proofs (PCPs) to design a system that enhances the speed and conciseness of proofs in comparison to existing schemes. Ligero++ achieved a significantly smaller proof size than both Aurora and Ligero, maintaining efficiency even for large circuits. Furthermore, it exhibited a proof time that is just two times slower than Ligero, while still outperformed Aurora in both size and generation time. The proposed system demonstrated a significant reduction in proof size for larger circuits and a more rapid proof generation process, offered a more practical and efficient solution for real-world applications.

(Chiesa, Dev, et al., 2020) introduced a novel methodology for efficiently realizing non-interactive, succinct and recursive compositional knowledge arguments. Previously, the only known approach relied on matching-based SNARKs instantiated over cycles of matching-friendly elliptic curves, which are computationally expensive algebraic structures. However, the proposed methodology eliminated the need for such specialized algebraic objects. The study also demonstrated the feasibility of achieving new, desirable post-quantum properties with a secure and transparent setup, where the setup is publicly available information. The primary aim was to develop a post-quantum and transparent pre-processed zk-SNARK for R1CS, given that recursive composition is more straightforward with pre-processed SNARKs. The authors conducted a comparative analysis of FRACTAL against state-of-the-art zk-SNARK schemes for R1CS. These included Aurora (Ben-Sasson et al., 2019), a pre-processor-less zk-SNARK operating in the quantum random oracle model; Groth16 (Jens Groth, 2016), a pre-processed zk-SNARK leveraging circuit-specific SRS and Marlin (Chiesa et al., 2020), a pre-processed zk-SNARK utilizing a universal SRS.

(Bünz et al., 2020) introduced a novel approach for constructing proofs of logarithmic dimension evaluation for polynomials over univariate and multivariate finite fields. This approach included a polynomial dependence scheme with verification time proportional to the number of polynomial coefficients. The key innovation was the Diophantine Argument of Knowledge (DARK), which utilized integer representations of polynomials and groups of unknown order. The authors applied this cryptographic compiler to a specific class of algebraic linear IOPs, termed Polynomial IOPs, aiming to achieve doubly efficient public coin interactive information arguments for NP with succinct communication. They provided a generic compilation of any Polynomial IOP using the DARK polynomial commitment scheme, surpassing frameworks like PLONK and Sonic. This led to the creation of the Supersonic scheme, a SNARK with logarithmic communication,

proving time and online verification time, all while maintaining transparency and removing the need for a trusted setup. The study demonstrated that Supersonic effectively proved circuits containing up to one million gates, producing proofs of only 7.8 KB and achieving verification times under 100 ms. A comparison with systems such as PLONK, Groth16, Bulletproofs, STARK and Virgo showed that Supersonic offered the smallest proof sizes, fastest verification times and a straightforward verification process. The authors presented Supersonic as the first complete zk-SNARK to combine practical proof time with logarithmic proof size and verification time, marking a significant advance in transparent zk-SNARKs.

(Maller et al., 2019) introduced Sonic, a novel ZKP system designed to produce compact and efficient proofs that are inexpensive to verify. The study underscored the need for efficient zk-SNARKs, especially for general arithmetic circuits and pointed out the limitations of existing systems, such as those requiring a trusted setup or having verification times proportional to the complexity of the relation, like Bulletproofs. Sonic advanced beyond previous protocols by introducing a universally structured and updatable reference string, with its size scaling linearly with the circuit size it supports. This approach contrasted with traditional zk-SNARK systems, where the reference string size typically grew quadratically with the relation size. Sonic's fixed proof size and verification costs, comparable to the most efficient SNARKs, made it particularly suitable for "assisted" collective verification. The system also enabled support for universal circuits without relying on bespoke, pre-processed reference strings, ensuring scalability and flexibility. Its verification process was efficient, requiring only a fixed number of checks and allowing simultaneous validation of multiple proofs. The study demonstrated Sonic's efficiency, achieving a proof size of just 256 bytes and a verification time of 0.7 milliseconds for circuits with limited samples. These improvements positioned Sonic as a highly effective solution for large distributed systems.

(Gabizon et al., 2019) argued that zk-SNARKs utilizing updatable universal SRSs remove a major barrier to the widespread adoption of zk-SNARKs (Groth et al., 2018). Maller et al. (2019) made a significant contribution by demonstrating that such an SRS enables Sonic, the first practical zk-SNARK with fully succinct verification for general arithmetic circuits. However, the fully succinct version of Sonic still incurred a relatively high proof generation cost. The authors proposed a universal SNARK framework offering fully succinct and self-verifiable proofs with significantly reduced proof generation times. This framework achieved a reduction in group exponentials by approximately 7.5–20 times compared to Sonic, depending on the circuit's structure. Similar to Sonic, the approach relied on the permutation argument developed by Bayer and Groth. The authors emphasized using evaluations on subgroups instead of monomial coefficients, simplifying both the permutation argument and the arithmetization step. This study introduced a fully succinct, universal and efficient zk-SNARK with substantially improved proof

runtime compared to Sonic. For circuits with over a million gates, PlonK proofs could be generated in under 23 seconds on consumer-grade hardware. This advancement significantly enhanced the efficiency of universal SNARKs, enabling their broader application in real-world scenarios.

(Bowe et al., 2019) introduced Halo, a novel recursive proof composition system designed to eliminate the reliance on a trusted setup, a longstanding challenge in zk-SNARK systems. Halo leverages the discrete logarithm assumption on elliptic curve cycles, enabling recursive proof composition without requiring expensive matching-friendly elliptic curve cycles or reliable configurations. Unlike other recursive proof systems such as STARKs, which produce relatively large proofs (hundreds of kilobytes), Halo achieved a constant proof size and verification time regardless of the recursion depth. The authors employed a technique called "nested amortisation" to optimise the verification circuit, ensuring it avoids linear-time operations. Instead, the circuit used public inputs and outputs to facilitate efficient proof verification. Halo demonstrated exceptional performance, producing recursive proofs as small as 3.5 KB at 128-bit security, verified using consumer-grade hardware. The study also showed that Halo outperformed prior systems like Bulletproofs and Sonic's assisted mode, offering logarithmic verification time and circuit size without requiring pairings or a trusted setup. This made Halo a significant advancement in recursive proof systems, combining efficiency and practicality.

(Ben-Sasson et al., 2019) introduced Aurora, a zero-knowledge SNARG designed for arithmetic circuit satisfiability, with argument dimensions that scale polylogarithmically with the circuit size. The primary goal of this research was to develop a publicly verifiable, concise, non-interactive argument system that is transparent, secure against quantum attacks and efficient in both proof generation and verification times. Aurora addressed the challenge of creating compact proofs for arithmetic circuits, offering exponential improvements in proof size compared to previous systems like Ligero. For instance, proofs for circuits with one million gates were 30 times smaller. The underlying IOP for Aurora used logarithmic query complexity, improving efficiency over earlier protocols that relied on multivariate summation. The paper highlighted several key benefits of Aurora, including post-quantum security, practical efficiency and a transparent setup that eliminated the need for trusted setups, making it well-suited for applications such as Zcash. The authors also compared Aurora's performance with other systems, showing that its argument size grows much more slowly than Ligero and Stark. Moreover, Aurora produced smaller proofs and verified them more quickly for larger circuits. This work contributed an efficient zk-SNARK with significantly reduced argument size, making it highly competitive for real-world use in post-quantum environments.

(Wahby et al., 2018) introduced a novel zero-knowledge argument for NP, addressing several limitations in existing state-of-the-art systems. These limitations included the need for

linear or super-linear proof dimensions, high verification costs and dependence on trusted setups or non-standard cryptographic assumptions. The authors' approach used a standards-based cryptographic model with low communication complexity, significantly reducing proof size and computational costs while eliminating the need for trusted setups. They introduced a commitment scheme for multilinear polynomials, which minimized witness-related communication at the cost of slightly increased verifier runtime. The authors applied the Fiat-Shamir heuristic to construct a succinct, non-interactive zero-knowledge argument within the Hyrax model. This approach produced smaller proofs than all baseline systems, except Bulletproof. While Bulletproof incurred higher computational costs, Hyrax demonstrated superior efficiency for large problem sizes, outperforming other systems in verification time. A comparison with systems like Ligero, libSTARK and ZKBoo++ showed that Hyrax struck an optimal balance between proof size and verification cost, making it particularly effective for practical, large-scale applications.

(Ames et al., 2017) introduced a zero-knowledge argument protocol for NP problems, characterized by communication complexity that grows with the square root of the verification circuit size. The protocol, which can be based on any collision-resistant hash function, is designed to be efficient for both large and medium-sized circuits encountered in practical applications. Additionally, it can be made non-interactive in the random oracle model, allowing for the construction of efficient zk-SNARKs or public key encryption without requiring a trusted setup. The communication cost to verify a SHA-256 preimage is approximately 44 KB, with proving and verification times of 140 ms and 62 ms, respectively. This makes the protocol about four times more efficient than comparable proofs like ZKBoo++ and ZKBoo. Derived from a variant of secure multiparty computation protocols, the protocol is optimized using general transformations. The authors aimed to combine the most effective elements from previous work to create a straightforward, efficient and configuration-free zero-knowledge argument system, reducing communication costs to kilobytes. The system achieves a 20% reduction in communication costs in its stronger variant, Ligero-Strong, compared to the base Ligero protocol. For circuits with over three million gates, the communication cost remains less than the circuit size. The protocol's computational complexity is mainly influenced by polynomial evaluations and interpolations, which can be improved with efficient FFT implementations for verification. The authors suggested that further optimization of polynomial evaluation could enhance verification efficiency.

2.1.3. Summary of Setup-Dependent Schemes

Table 2.1 summarizes the studies on zk-SNARK algorithms examined in detail in this thesis. The first three entries correspond to cryptographic schemes that assume a trusted setup phase. The subsequent seven studies related to schemes operating under a universal setup phase, followed by ten studies addressing schemes that rely on a transparent setup. Within each setup

category, the studies are systematically ordered in reverse chronological order based on their publication dates.

The classification of zk-SNARK systems by setup type highlights the evolution of trust assumptions in ZKP constructions over the past decade. Early zk-SNARK systems such as Geppetto (2014), Pinocchio (2016) and ZKBoo++ (2017) rely on a trusted setup, which requires the generation and secure disposal of toxic waste to ensure soundness and zero-knowledge. Although effective, this setup model introduces a critical point of failure, as any compromise in the setup phase could jeopardize the system's security. In contrast, more recent systems increasingly move towards universal and transparent setups, reflecting a broader trend in the cryptographic community toward minimizing trust assumptions and improving auditability and reproducibility in protocol design.

Table 2.1. Overview of setup-dependent studies

Study	Setup Type	Year	ZKP System	Protocol	Cryptographic Assumptions	Post-Quantum Secure
Chase et al.	Trusted	2017	ZKBoo++	zk-SNARK	KoE	yes
Parno et al.	Trusted	2016	Pinocchio	zk-SNARK	KoE	no
Costello et al.	Trusted	2014	Geppetto	zk-SNARK	KoE	no
Chen et al.	Universal	2023	HyperPLONK	zk-SNARK	AGM	yes
Weng et al.	Universal	2021	Wolverine	zk-SNARK	sVOLE	no
Yang et al.	Universal	2021	QuickSilver	zk-SNARK	sVOLE	no
Kosba et al.	Universal	2020	Mirage	zk-SNARK	GGM	no
Chiesa et al.	Universal	2020	Marlin	zk-SNARK	KoE,AGM	no
Maller et al.	Universal	2019	Sonic	zk-SNARK	AGM	no
Gabizon et al.	Universal	2019	Plonk	zk-SNARK	AGM	no
Xie et al.	Transparent	2022	Orion	zk-SNARK	CRHF	yes
Boneh et al.	Transparent	2021	Halo Infinite	zk-SNARK	DL	no
Setty	Transparent	2020	Spartan	zk-SNARK	KoE	yes
Bhadauria	Transparent	2020	Ligero++	zk-SNARK	CRHF	yes
Chiesa et al.	Transparent	2020	Fractal	zk-SNARK	KoE	yes
Bünz et al.	Transparent	2020	SuperSonic	zk-SNARK	AGM	no
Bowe et al.	Transparent	2019	Halo	zk-SNARK	DL	no
Ben-Sasson et al.	Transparent	2019	Aurora	zk-SNARK	CRHF	yes
Wahby et al.	Transparent	2018	Hyrax	zk-SNARK	DL	no
Ames et al.	Transparent	2018	Ligero	zk-SNARK	CRHF	no

SNARK: Succinct Non-interactive ARgument of Knowledge, KoE: Knowledge of Exponent, AGM: Algebraic Group Model, GGM: Generic Group Model, sVOLE: subfield Vector Oblivious Linear Evaluation, ARA: Adaptive Root Assumption, CRHF: Collision-Resistant Hash Functions.

A notable pattern across the examined zk-SNARK systems is the relationship between cryptographic assumptions and post-quantum security guarantees. Systems based on standard assumptions such as CRHF, as seen in Orion, Ligero++ and Aurora, are generally considered to provide post-quantum security, owing to the resistance of hash-based constructions to quantum

attacks. Conversely, protocols built upon assumptions like KoE or DL — including Spartan, Halo and Fractal — typically do not offer such resilience. This distinction is particularly critical in the transparent setup category, where the tension between efficiency and quantum robustness is most pronounced. Overall, the data suggest a growing emphasis on achieving cryptographic transparency and quantum resistance in modern zk-SNARK design.

2.2. Setup-Free ZKP Schemes

This section will examine studies on zk-STARK, Bulletproofs and zk-VPD schemes that do not require any pre-processing setup phase.

2.2.1. Scalable, Transparent Arguments of Knowledge (zk-STARKs)

(Hak et al., 2024) introduced a novel authentication scheme called "LSAC: A Lightweight, Scalable and Anonymous Cross-Domain Authentication Scheme in IoMT." The LSAC framework was developed to address key challenges of privacy, security and scalability in the Internet of Medical Things (IoMT). Its goal is to establish a lightweight, transparent cross-domain authentication system leveraging Self-Sovereign Identity (SSI) integrated with advanced ZKP protocols, such as zk-STARK and PLONK. This integration enables efficient identity verification without relying on a trusted setup. The study's primary contribution lies in merging SSI with ZKP technologies to create a highly secure and scalable authentication mechanism capable of managing large user volumes in cross-domain IoMT environments while ensuring strict privacy protections. Additionally, the framework employs automated smart contracts using the Hyperledger platform, which enables secure and auditable interactions across domains. Experimental results highlight that LSAC provides fast and secure access to IoMT resources with significant performance enhancements, establishing it as a practical solution for secure data sharing in healthcare applications.

(Quan, 2022) presented a study titled "Improving Bitcoin's Post-Quantum Transaction Efficiency With a Novel Lattice-Based Aggregate Signature Scheme Based on CRYSTALS-Dilithium and a STARK Protocol," aiming to enhance Bitcoin's transaction efficiency in a post-quantum environment. This is achieved through the introduction of a lattice-based aggregate signature (LAS) scheme that integrates CRYSTALS-Dilithium with a STARK protocol. The study addressed the challenge of large post-quantum signature sizes, which negatively impact Bitcoin's transaction throughput. The proposed LAS scheme effectively aggregated signatures, optimising block space while ensuring robust security, all within Bitcoin’s 1MB block size constraint. Empirical findings revealed that LAS reduces transaction efficiency by only a factor of three compared to ECDSA, a substantial improvement over previous post-quantum methods, which caused efficiency reductions of up to 17 times. Consequently, the LAS scheme emerged as a promising solution for meeting Bitcoin's post-quantum scalability and security demands.

(Mouris & Tsoutsos, 2021) introduced Zilch, a framework designed to simplify and expedite the deployment of verifiable computation (VC) and ZKPK across various applications without the need for a trusted setup. Zilch addressed the common reliance on trusted third-party parameters in many VC and zero-knowledge systems by offering a transparent approach. Utilizing the zk-STARK library, Zilch eliminated potential "toxic waste" vulnerabilities while ensuring quantum resistance. The framework featured a high-level programming model, ZeroJava, which translated algorithms into mathematical constraints, enhancing usability and performance. In comparative analyses with transparent zero-knowledge systems like Hyrax and Bulletproof, Zilch successfully demonstrated proof of correctness in computations. However, it incurred higher costs in proof generation and verification time compared to Bulletproof. This research marked a notable advancement in the development of secure, transparent and efficient zero-knowledge systems that operate independently of external trust dependencies.

Ben-Sasson et al. (Ben-Sasson et al., 2018) introduced zk-STARK, a transparent ZKP system developed to improve verification speed and scalability in computational integrity (CI) without relying on a trusted setup. By utilizing interactive oracle proofs and advanced error correction techniques like the fast Reed-Solomon IOP of proximity (FRI), the system achieved exponential verification speedup relative to computation size. Zk-STARK was designed to support transparency, post-quantum security and confidentiality while enabling scalable verification for large-scale computations. The authors demonstrated that zk-STARK significantly outperformed existing systems, particularly in large-scale sequential computations, offering faster proof generation and verification compared to other transparent protocols. While smaller or less complex circuit-specific systems may appear more efficient in certain contexts, zk-STARK delivered superior long-term verification and communication efficiency. This positioned it as a robust solution for scalability-focused applications, such as decentralized cryptocurrencies.

2.2.2. Bulletproofs

(Wang et al., 2024) introduced "SwiftRange: A Short and Efficient Zero-Knowledge Range Argument for Confidential Transactions and More," aiming to develop an optimized zero-knowledge range proof system tailored for block chain-based confidential transactions. Building on the limitations of existing logarithmic-sized range proofs, such as Bulletproofs, they proposed SwiftRange, which features a transparent setup and operates in the discrete logarithm setting. SwiftRange was notable for its efficient computational performance and minimal communication overhead. Its key contributions included a logarithmic-size argument with enhanced verification efficiency and support for aggregation, enabling more compact and scalable confidential transactions. A comparative analysis demonstrated that SwiftRange achieved a 1.73x improvement in proving efficiency and nearly doubled verification efficiency for typical block chain ranges. This

framework offered a promising alternative to existing range arguments, especially in high-throughput transaction environments where verification speed and communication size are crucial.

(Eagen et al., 2024) introduced Bulletproofs++ (BP++), an enhanced version of the Bulletproofs (BP) cryptographic system designed to enable privacy-preserving transactions, such as Confidential Transactions (CT) in cryptocurrencies. Compared to BP, BP++ offers significant improvements in efficiency and compactness, including reductions in proof size and verifier runtime. BP++ relies on the discrete logarithm problem in the random oracle model for security. Unlike BP, which requires O(n) group scalar products, BP++ only requires O(n/log n), making it more efficient, especially for large proofs. The paper presented new techniques for permutation and set membership, allowing for the efficient generation of proofs for expressions encoded as arithmetic circuits. BP++'s interval proofs, such as those verifying a 64-bit value, are 38% smaller and five times faster to prove than those generated by BP. Additionally, BP++ supports a transparent setup, multiparty proofs and batch verification, resulting in substantial reductions in proof and verification times in real-world applications. The work also introduced the Reciprocal Argument, a novel interactive oracle proof that underpins BP++ and generalizes permutation and set membership arguments. Through empirical evidence, the authors demonstrated the clear advantages of BP++ over existing systems, particularly in scalability and performance.

(Chung et al., 2022) introduced Bulletproofs+, an enhanced version of the Bulletproofs cryptographic system, designed to provide shorter, zero-knowledge arguments for arithmetic circuits and range proofs without requiring a trusted setup. The proposed scheme achieves the shortest proof size among systems that do not rely on a trusted setup, with a proof size of 576 bytes for integers under 64 bits. This represented an 85.7% improvement over previous shortest proofs, such as those by Bünz et al. (2018). The key innovation was the zero-knowledge weighted inner product argument (zk-WIP), which reduces the length of both range and arithmetic circuit proofs while maintaining minimal communication overhead. Bulletproofs+ eliminated the need for random numbers used in conventional Bulletproofs by transforming committed values into degree-1 random polynomials, reducing the number of proof elements from five to one. Experimental results showed that Bulletproofs+ slightly increased processing speed compared to Bulletproofs, particularly as the number of accumulated proofs grew. For example, the proposed protocol demonstrated a 9.7% reduction in execution time for a single 32-bit range proof and a 26.8% improvement for 64 × 32-bit secrets, highlighting its effectiveness in practical applications.

(Bünz et al., 2018) introduced Bulletproofs, a novel NIZKP protocol that offers extremely short proofs, with proof sizes logarithmic to the witness dimension and does not require a trusted setup. Bulletproofs were especially effective for generating interval proofs over committed values, using only $2 \log_2(n) + 9$ group and field elements to prove that a committed value lies within a

specified range, where n is the bit length of the range. The proof generation and verification times were linear in n. The authors demonstrated Bulletproofs' efficiency for secret transactions in Bitcoin and other cryptocurrencies, showed that it significantly improves the efficiency of interval proofs compared to previous proposals. Furthermore, Bulletproofs provided a more compact alternative to earlier schemes, such as that of Bootle et al. (2016), while still allowing the use of Pedersen commitments for inputs. The paper explored Bulletproofs' applications in various contexts, including confidential transactions in sidechains, private block chains and privacy-focused cryptocurrencies like Monero. Notably, Bulletproofs could reduce the size of proof-of-range data for Bitcoin UTXOs from 160 GB to under 17 GB, represented a tenfold reduction. The protocol supported multiple interval proofs in a single operation, with proof size growing logarithmically instead of multiplicatively, as seen in other solutions. The evaluation showed that Bulletproofs use less than 100 MB of memory and improve verification efficiency. A 64-bit interval proof occupies only 688 bytes and batch verification further reduces the cost of verifying multiple proofs, made it more time-efficient than traditional ECDSA signature verification.

2.2.3. Zero Knowledge Verifiable Polynomial Delegation (zk-VPD)

(Ghosal et al., 2023) introduced a fully non-interactive, publicly verifiable delegation scheme for committed programs. This allowed a trusted program author (Alice) to delegate computation to an untrusted worker while enabling any verifier to confirm the correctness of the outputs without knowing the specifics of the program. The approach addressed scenarios where Alice's program contains sensitive information or requires significant computational resources, which she may prefer not to disclose or execute personally. Unlike previous solutions that relied on SNARGs or SNARKs with random oracle assumptions, this scheme achieved public verifiability and non-interactivity under the standard Learning with Errors (LWE) assumption. In this model, Alice published a succinct hash (HP) of her program, which the verifier can use to authenticate the worker's outputs. The scheme introduced a semi-trusted SNARG (stSNARG) concept, allowing Alice to create a digest for verification, without requiring access to the program itself. The method preserved zero-knowledge, ensured the confidentiality of Alice's program and enhanced efficiency through hash trees and batch arguments, minimized communication for verification. This framework was particularly beneficial in cloud computing and data privacy applications, where users seek assurances of computational integrity and privacy without disclosing their underlying data or models.

(González & Zacharakis, 2021), focused on developing a publicly verifiable, non-interactive delegation scheme that allows verifiers with limited computational power to delegate polynomial computations to more capable but untrusted provers. The proposed scheme enables verification with proof sizes and verification complexities similar to pairing-based zk-SNARKs but relied on simpler, falsifiable cryptographic assumptions instead of complex, non-falsifiable ones.

Specifically, the authors used k-Matrix Diffie-Hellman (k-MDDH) assumptions to ensure soundness and introduced innovative techniques, such as no-signalling and somewhere statistically binding commitment schemes, to guarantee both extractability and efficient delegation for arithmetic circuits. The study contributed by addressing the limitations of zk-SNARK-based delegation, which typically depends on stronger assumptions. The authors' approach integrated quasi-arguments, enabling partial witness extraction without revealing sensitive data and pairing-based quasi-arguments for linear and quadratic constraints, making the system more efficient for these types of relations. This delegation scheme was fully succinct, with proof size linearity relative to the security parameter and verification complexity proportional to the input size. Experimental results showed practical efficiency gains, particularly in proof size and verification complexity, making it a promising advancement for applications requiring secure delegation without compromising computational practicality.

(Zhang et al., 2021) introduced a novel interactive proof protocol for general arithmetic circuits that is twice as efficient as existing protocols. This protocol generalized the interactive proof method proposed by Goldwasser, Kalai and Rothblum for layered circuits, extending it to arbitrary circuits while maintaining optimal prover complexity, which is linear in the size of the circuit. The proof size remained compact for low-depth circuits and the verification time was sublinear for structured circuits. Additionally, the authors presented a zero-knowledge argument for general arithmetic circuits, combining their interactive proof protocol with polynomial commitments. The key innovation was a sum-checking equation that reduces assertions about circuit outputs to assertions about inputs, eliminating the overhead typically associated with circuit depth. The authors successfully extended the GKR protocol to arbitrary circuits without adding additional prover overhead, ensuring that proof time remained linear in the circuit size O(C). The new protocols, implemented in a system called Virgo++, were 9-13 times faster than the state-of-the-art GKR protocol for random circuits. They could handle circuits with over 600,000 gates in just 0.3 seconds, which is 13 times faster than the original GKR. The proof size was 208 kilobytes and the verifier time was 66 milliseconds. Compared to existing ZKP systems like Virgo, Spartan, Aurora and STARK, Virgo++ exhibited the fastest proving time, with performance up to 268 times faster than Aurora. While the evidence size was slightly larger than that of Virgo++ in some cases, Virgo++ still outperformed other systems in terms of proving and verification speed, being 5.7 times faster than Virgo and 46.8 times faster than Aurora. The authors concluded that their system significantly improves both efficiency and scalability for proving computations on very large circuits.

(Zhang et al., 2020) introduced a novel zero-knowledge argument scheme for layered arithmetic circuits that does not require a trusted setup. This scheme utilized lightweight cryptographic primitives, including collision-resistant hash functions, to ensure post-quantum

security. The proof time was of the order of (C + n log n), while the verification time was of the order of (D log C + $\log_2$ n), where (C) represented the number of gates, (n) was the number of inputs and (D) was the circuit depth. The scheme was implemented in the Virgo system, which combines a new zk-VPD scheme with the GKR protocol to create a transparent ZKP system. Virgo outperformed existing schemes, achieving proof generation 8-10 times faster and significantly reducing verification time—up to 10-30 times faster than Libra. Although the proof size was larger (by a factor of 30-40) than some alternatives, it could be reduced using vector commitment schemes. The authors highlighted that Virgo eliminates the need for a trusted setup phase, which is typically required in traditional systems. The verification time for a circuit with 226 gates is just 50 ms, positioned it on par with systems like STARK. Moreover, its proof size was comparable to other systems, including Bulletproofs and Hyrax, made Virgo highly efficient for practical applications.

2.2.4. Summary of Setup-Free Schemes

Table 2.2 provides a summary of ZKP schemes that do not require a pre-processing setup phase, accompanied by a comparison of their key characteristics, including the underlying cryptographic assumptions and their capacity to provide security in the post-quantum era.

Table 2.2. Overview of setup-free studies

Study	Protocol	Year	ZKP System	Cryptographic Assumptions	Post-Quantum Secure
Hak et al.	zk-STARK	2024	LSAC	CRHF	yes
Quan	zk-STARK	2022	LAS	CRHF	yes
Mouris and Tsoutsos	zk-STARK	2021	Zilch	CRHF	yes
Ben-Sasson et al.	zk-STARK	2018	Stark	CRHF	yes
Wang et al.	sh-SNARK	2024	SwiftRange	DL	no
Eagen et al.	sh-SNARK	2023	Bulletproofs++	DL	no
Chung et al.	sh-SNARK	2022	Bulletproofs+	DL	no
Bünz et al.	sh-SNARK	2018	Bulletproofs	DL	no
Ghosal et al.	zk-VPD	2022	st-SNARG	CRHF	no
González and Zacharakis	zk-VPD	2021	*k*-MDDH	CRHF	no
Zhang et al.	zk-VPD	2021	Virgo++	CRHF	yes
Zhang et al.	zk-VPD	2020	Virgo	CRHF	yes

Sh-SNARK: “short” Succinct Non-interactive ARgument of Knowledge, STARK: Scalable Transparent ARgument of Knowledge, st-SNARG: semi-trust Succinct Argument of Knowledge, DL: Discrete Logarithm, CRHF: Collision-Resistant Hash Functions.

Table 2.2 highlights recent advancements in setup-free ZKP systems, which eliminate the need for a trusted or universal setup phase. This paradigm shift marks a significant departure from earlier zk-SNARK designs, favoring transparency and ease of deployment. Among these systems,

zk-STARKs—exemplified by protocols such as (Hak et al., 2024), (Quan, 2022) and (Ben-Sasson et al., 2018)—consistently rely on CRHF, positioning them as inherently post-quantum secure. Their reliance on standard cryptographic primitives makes them robust candidates for long-term applications, especially in adversarial settings where the presence of a secure setup cannot be guaranteed. These properties, along with their scalability, make zk-STARKs a popular choice in domains such as blockchain and privacy-preserving computation.

In contrast, sh-SNARK protocols, including various iterations of Bulletproofs (2016–2024), are rooted in the DL assumption and consequently lack post-quantum security. While sh-SNARKs are valued for their succinctness and do not require a trusted setup, their reliance on algebraic assumptions vulnerable to quantum adversaries limits their long-term applicability. Meanwhile, the zk-VPD family introduces st-SNARGs such as Virgo and its successors, offering a hybrid approach that balances efficiency and post-quantum resistance through the use of CRHF. Notably, Virgo++ (Zhang et al., 2021) and similar schemes combine the transparency of zk-STARKs with additional structural optimizations, suggesting an evolving research trajectory aimed at reconciling performance with robust security guarantees. Overall, the table illustrates a growing preference for setup-free designs that emphasize post-quantum readiness and minimal trust assumptions.

2.3. Practical Use Cases of ZKPs

This section presents a review of the existing literature on the real-world applications of the ZKP schemes. These applications are categorised into four primary domains: finance, healthcare, business and general-purpose use.

2.3.1. Financial Use Cases

The scenario entails a global accounting firm utilising a centralised database for the processing of its clients' financial data. However, the increasing prevalence of cyber-attacks on databases and the growing demand for transparency from regulators have created a significant challenge for the firm. Clients require the assurance of confidentiality for their sensitive financial data, while regulators advocate enhanced transparency. Furthermore, the prevalence of manual processes within the system has resulted in a notable increase in error rates, accompanied by a rise in operational costs and an elongation of transaction times.

(Wang & Kogan, 2018) explored the potential of block chain technology to revolutionize accounting and auditing practices as a proposed solution to the aforementioned scenario. They concentrated on its capacity to enhance data integrity, reduce costs and facilitate real-time transaction reconciliation. The authors identified several significant challenges, including the susceptibility of public block chains to privacy breaches. In response, they proposed a block chain-based Transaction Processing System (TPS) that integrates ZKPs (zk-SNARKs) and homomorphic

encryption as a potential solution to the problem illustrated in the scenario. This approach protects the confidentiality of transactions while maintaining verifiability. The authors illustrate the tokenisation and transfer of assets within a block chain ecosystem through the use of examples, thereby demonstrating the creation of a secure yet transparent system. The study highlights the advantages of decentralisation in modern cryptocurrencies such as Bitcoin and Ethereum, while addressing privacy concerns. It advocates for the convergence of block chain and cryptographic protocols in order to support secure and efficient enterprise systems. The paper concludes with a comparative analysis of computational performance, demonstrating the efficacy of the TPS over traditional relational databases for continuous monitoring and fraud prevention in accounting.

The scenario presents an international financial technology company that is under pressure to provide customers with a privacy-focused payment system while ensuring compliance with regulatory requirements. The existing centralised payment systems satisfy the requirements of the regulators due to their high transparency, but this is achieved at the expense of customers' privacy. Furthermore, traditional systems are not scalable and transaction costs increase in proportion to the size of the system. The company is seeking an innovative solution that will simultaneously safeguard customer privacy and provide regulators with the ability to conduct audits.

(Chen et al., 2020) proposed the concept of an auditable decentralised confidential payment (ADCP) system as a means of reconciling the competing demands of privacy and auditability in the context of the aforementioned scenario. Moreover, the authors put forth the proposition that the ADCP system offers two distinct degrees of auditability: regulatory compliance and auditing. The authors present a general structure of the ADCP system, which employs an integrated signature and encryption scheme and NIZKP systems. The authors then proceed to present an illustrative example of the general structure of the proposed system, demonstrating how the careful design of the basic building blocks results in the creation of an independent cryptocurrency, designated as PGC. Moreover, the authors demonstrate that the configuration process in PGC is semi-transparent and that transaction costs remain constant regardless of the system's scale. Additionally, the size of the PGC block chain is presented as approximately 1.4 KB, with a creation time of less than 28 ms and a verification time of less than 9 ms. The proposed system is capable of achieving the objective of enabling auditability while providing the required privacy in the aforementioned scenario.

The following scenario is presented for consideration: a country's central bank utilises an RTGS (Real Time Gross Settlement) system to facilitate financial transactions in real time. Nevertheless, the existing system is confronted with a number of significant challenges. A further issue is the lack of trust. The fully centralised nature of the system engenders a lack of trust between the participants in transactions between banks. The centralised nature of the system creates a vulnerability that can result in the cessation of all transactions in the event of a technical failure or

cyber-attack. Furthermore, the transparency of both the payment amount and the recipient information during transactions raises concerns regarding the confidentiality of financial transactions, which is a significant issue for both financial institutions and individuals. This highlights the necessity for the development of a more secure and privacy-focused alternative.

In their examination of the principal challenges confronting contemporary RTGS systems, (Galal & Youssef, 2020) direct their attention to the issues of trust, central bank privileges and single points of failure, which are intrinsic to centralised systems. These are addressed in the above example scenario. In order to address the aforementioned issues, the authors put forth a decentralised netting protocol with the objective of enhancing payment confidentiality and recipient privacy. The protocol is evaluated on the Ethereum platform and its practical applicability and security are demonstrated. The paper makes two principal contributions: the design of the decentralised netting protocol and an evaluation of its performance on the Ethereum platform. The evaluation is centred on the gas cost associated with elliptic curve operations during zk-SNARK proof verification. The results demonstrate that the protocol is feasible on Ethereum and that the costs represent a negligible proportion of the block gas cap. Furthermore, the authors emphasise the augmented functionality of the protocol on permissioned block chain platforms, such as Corda and Hyperledger, which are preferred by financial institutions due to the shortened mining times and absence of gas limitations. In conclusion, the proposed protocol addresses significant limitations in RTGS systems by protecting privacy and guaranteeing the confidentiality of payment amounts, thus providing a promising solution for financial transactions.

The objective of the fintech company is to facilitate the secure transfer of assets on a block chain-based platform. However, the transparent nature of block chain technology renders users' transaction history and account balances publicly accessible, which jeopardises user privacy and underscores the necessity to guarantee anonymity in financial transactions. The company aims to develop a solution that will simultaneously safeguard user privacy and be more efficient than current systems.

In order to address the privacy concerns raised by the digital asset company in the aforementioned scenario, (Huang et al., 2024) put forth the proposal of zkChain, a privacy-preserving and efficient asset transfer system that is based on zk-SNARKs and a hash chain. Despite its prevalence in domains such as market supervision and digital identity management, block chain technology is confronted with considerable challenges pertaining to privacy. This is due to the fact that it is inherently transparent, thereby providing access to all historical transaction data. The issue of privacy has emerged as a significant barrier to the adoption of block chain technology. The authors have introduced the zkChain model, which employs a one-way hash chain to represent account balances and generate smaller, faster ZKPs in comparison to traditional

approaches such as Zerocoin and Zerocash. The zkChain scheme ensures the privacy of user balances and the anonymity of transaction parties. The paper's evaluation of the computational overhead of zk-SNARKs reveals that, although proof generation is more expensive than verification, it has a negligible impact on overall performance, as it is only executed by the user. The findings demonstrate that the zkChain model produces smaller proof sizes and faster verification times than traditional systems, offering an effective approach to privacy in block chain transactions with a moderate overhead. This solution has the potential to facilitate the broader adoption of block chain technology.

2.3.2. Medical Use Cases

The following scenario presents a number of security and efficiency issues that have been identified in the health insurance process. The characters involved in this scenario are as follows: Emma: A patient who is seeking to apply for insurance. Mark: A representative of a health insurance company. System: A block chain-based insurance management platform. Problem: Emma is interested in applying for a health insurance claim, but the current process is perceived as complex, time-consuming and inefficient. Additionally, she is reluctant to share her personal and medical information due to concerns about data breaches. Mark expresses concerns about the rise in fraudulent claims and the length of verification processes.

(Zheng et al., 2022) address the challenges specific to the health insurance claims process, including low efficiency, complexity, unreliable data and data leakage. These issues are exemplified in the aforementioned scenario. Furthermore, they highlight the necessity to safeguard the confidentiality and accuracy of patients' medical data. In light of the fact that approximately 25 per cent of insurance claims are identified as potentially fraudulent, the authors present a block chain-based solution that combines smart contracts and ZKPs with the objective of improving efficiency and guaranteeing confidentiality. The paper presents a novel scheme that employs NIZKPs, homomorphic encryption and block chain technology to guarantee the legitimacy and confidentiality of transactions between patients and insurance companies. The authors make three significant contributions to the field: (1) ensuring the authenticity and confidentiality of medical data using zk-SNARKs and digital signatures; (2) combining block chain with homomorphic encryption to protect transaction privacy; and (3) requiring patients to demonstrate their legal status for claims while protecting their identity. The study presents analyses of the security, computational cost and communication cost of the proposed scheme, demonstrating that it meets legal requirements and ensures privacy while maintaining an acceptable time burden.

As a scenario, in a country, the implementation of intelligent health systems is becoming increasingly prevalent with the objective of digitising healthcare services. However, the utilisation of these systems has resulted in the emergence of significant concerns pertaining to data privacy

and security, as outlined below: In terms of privacy violations, the sensitive nature of patient data means that it is a breach of privacy for it to be accessible to those who should not have access to it. Insufficient access control measures: In a centralized system, it is not possible to implement effective restrictions on which healthcare professionals can access which data. The issue of scalability represents a significant challenge for these systems. As the quantity of data continues to grow, existing systems are unable to process it in a timely and efficient manner. Deficiencies in emergency management include: In the event of a patient's condition becoming critical, it is frequently not feasible to promptly notify the pertinent healthcare personnel. These issues erode the confidence of both patients and healthcare professionals, thereby impeding the adoption of the system.

(Mohammad Hossein et al., 2021) investigated the potential of block chain technology to address data privacy and security challenges in the context of smart healthcare, a key application of the Internet of Everything (IoE), as a solution to the issues encountered in the aforementioned scenario. The authors introduced BCHealth, an architectural solution that enables data owners to set access policies for privacy-sensitive health data. BCHealth employs two distinct chains: a data chain for the storage of health data and an access control chain for the management of access policies. The system employs a novel clustering approach with the objective of enhancing scalability and network throughput, incorporating a proof-of-authority (PoA) consensus algorithm. Furthermore, the BCHealth system incorporates an alarm system that is configured to alert medical personnel in the event of an emergency, based on the patient's health status. The performance of BCHealth is evaluated through comprehensive experimental analyses, which demonstrate enhanced network latency, cost-effectiveness and resilience against security threats. The research makes a number of significant contributions to the field, including an increase in scalability through the use of clustering, optimisation of access control and the assurance of the protection and secure sharing of health data with healthcare providers.

In this scenario, a healthcare organisation has elected to transition to a cloud-based electronic health (eHealth) system with the objective of digitising patient information and enhancing operational efficiency. However, the process is beset with significant challenges, as outlined below. Integrity issues with data: The transfer of electronic health records (EHRs) to the cloud renders verification of their integrity challenging. A further issue is the lack of privacy and traceability. In instances where patient data is disseminated, there is a potential for misuse of access or disclosure to unauthorised individuals. Furthermore, it is not feasible to ascertain the identity of the individual who accessed the data, the date of the access and the manner in which it was accessed. Cost and Performance Issues: Previous eHealth systems have been observed to exhibit elevated communication costs and suboptimal performance, particularly when processing large

data sets. These issues jeopardise both patient safety and the effectiveness of healthcare services, thereby underscoring the necessity for a reliable solution.

(Huang et al., 2021) investigate the advantages and challenges of migrating to a cloud-based electronic health system (eHealth) as discussed in the above example scenario. While such systems address the shortcomings of traditional medical management, significant concerns remain regarding privacy and security, particularly those related to data storage and sharing. The authors identify two principal challenges: firstly, ensuring the integrity of EHRs throughout the outsourcing process; secondly, maintaining confidentiality and traceability during data sharing. In order to mitigate these risks, the authors put forth the implementation of a block chain-based eHealth system, designated as BCES, which records all legitimate queries, alterations and outsourcing activities as block chain transactions, thereby ensuring persistent traceability. Moreover, the BCES system incorporates attribute-based proxy re-encryption, which permits the implementation of granular access control and facilitates the detection of potential integrity issues by auditors. The paper addresses the issue of communication overhead and states that although this increases with data size, the BCES system provides a more cost-effective solution for patients than previous systems. Notwithstanding the intrinsic difficulties, the authors contend that BCES offers a robust safeguard for EHRs, guaranteeing accountability in the event of data manipulation through its traceable and tamper-proof structure. A comprehensive security analysis and performance evaluation demonstrate that BCES is a secure and efficient system, capable of defending against a range of attacks while also providing granular access control and tamper-proofing.

In this scenario, a healthcare organisation is seeking to implement a system for the sharing of patient medical data with research institutions. The objective of this system is to facilitate the development of novel treatment modalities and solutions that will enhance public health outcomes. Nevertheless, a number of challenges emerge during this process, which are outlined in the following section. The issue of patient privacy is a significant concern in this context. Patients are anxious about the potential misuse of their personal and sensitive health information or its inadvertent disclosure to unauthorised individuals. The issue of data integrity and reliability is of paramount importance. It is of the utmost importance to research institutions that the data they receive is accurate and unaltered. Authorisation issues It is unclear how the requisite secure access to data will be provided, given that only certain institutions should be able to access it. The issue of data sharing and anonymity is a significant concern in the field of medical research. In the context of data sharing, it is of paramount importance to ensure the anonymity of patients. These issues highlight the necessity to safeguard patient confidentiality while facilitating secure data sharing for the advancement of health research.

(Huang et al., 2020) addressed the issues related to sharing patient data with third parties in the above-mentioned scenario. They proposed a block chain-based privacy protection scheme designed to facilitate the secure sharing of medical data among a heterogeneous set of entities including patients, research institutions and semi-trusted cloud servers. This was presented as a solution to the problem addressed in this scenario. In addressing the challenge of reconciling patient privacy with research or commercial needs related to health data in the rapidly developing field of smart medicine, the authors leverage block chain technology to guarantee the confidentiality and integrity of medical data. The use of zk-SNARKs enables the secure decryption of data between patients and research institutions through smart contracts, while ensuring the anonymization of sensitive information including the sender, recipient and transaction amounts. The work makes a number of significant contributions, including the development of a decentralized data sharing system that combines block chain technology with proxy re-encryption, thereby ensuring the confidentiality of data while enabling authorized research institutions to reliably decrypt data. Moreover, the authors show that the proposed scheme satisfies the basic privacy criteria, including confidentiality, availability and integrity. Additionally, it demonstrates superior performance in terms of execution efficiency, block generation speed and privacy protection when compared to existing solutions. The findings demonstrate the potential of the scheme to address critical issues in the secure sharing of medical data, thereby enhancing both privacy and availability of data in the healthcare and research sectors. By meeting the basic privacy criteria, such as anonymity, data integrity and access control, this system holds great potential for both the healthcare sector and research institutions. This solution can shape the future of smart medicine by enabling the trustless sharing of healthcare data.

A pharmaceutical manufacturer is seeking to collate medical data utilised in clinical trials and disseminate this data to a third-party research organisation. The objective is to analyse the data in order to evaluate the efficacy of novel drug treatments. Nevertheless, a number of issues have been identified during this process, which are outlined in the following section. The issue of data privacy is a significant concern in this context. The data pertaining to patients is of a highly sensitive nature, both in terms of medical and personal considerations. Consequently, the privacy of this data is of significant importance to patients. The reliability of data validation is a significant concern. It is of paramount importance for the research organisation to ascertain the veracity of the medical data it receives. Nevertheless, it is challenging to verify the data in question without access to its actual content. A further challenge is the issue of scalability. The sheer volume of data involved in clinical trials can impede the validation process and drive up costs. This situation has prompted the search for a solution that will enable third parties who wish to independently verify the accuracy of medical data to do so without accessing the original data.

(Backes et al., 2015) address a significant challenge in the field of privacy-preserving data verification, particularly in contexts where a third party must validate the authenticity of data from a trusted source without accessing the original data. This issue is particularly pertinent in applications such as wearable computing, smart metering and business-to-business interactions, where data privacy and scalability are of paramount importance. The authors put forth ADSNARK, an innovative cryptographic approach that enables third parties to verify the correctness of computations on authenticated data while ensuring its confidentiality, in accordance with zero-knowledge principles. The necessity for such a solution can be attributed to the increasing demand to harmonise data protection with data validation. In contexts such as clinical trials, where medical data must be shared for research purposes, it is of paramount importance to verify the authenticity of this sensitive information without disclosing patient details. ADSNARK addresses this challenge by enabling the validation of processed data, such as the results of medical studies, without disclosing any underlying confidential data, thereby maintaining both privacy and data integrity. The system utilises a highly efficient cryptographic method that not only ensures the correctness of random computations but also provides substantial improvements in proof computation speed and storage efficiency in comparison to unauthenticated solutions. In experimental evaluations, ADSNARK demonstrated superior performance compared to previous methods, such as those based on Pinocchio, by factors of 25 and 20, respectively. This highlights its enhanced scalability and efficiency. The practical applicability of ADSNARK is of significant importance for privacy-sensitive sectors such as healthcare, where the confidentiality of patient data must be ensured while enabling secure and reliable data verification. By providing a robust and scalable solution for the preservation of data privacy, ADSNARK contributes to the secure sharing and analysis of sensitive data, thereby making it a valuable tool for industries that rely on data integrity, such as the pharmaceutical industry, medical equipment manufacturing and clinical research.

2.3.3. Business Use Cases

The following scenario is presented for consideration: A government department is currently engaged in preparations for a significant auction of government bonds, which will involve a diverse group of institutional bidders. The existing auction system, while operational, faces significant challenges in guaranteeing bidder privacy, particularly with regard to the confidentiality of losing bids. Additionally, there is a concern regarding the verifiability of the auction results, as bidders seek assurance that the auction has been conducted fairly, without manipulation or errors. Furthermore, as the auction grows in size, the current communication protocols result in increased operational costs, limiting the scalability of the system.

In their examination of the aforementioned example scenario, (Zhang et al., 2023) proposed a methodology aimed at enhancing the security of auction processes, which play a pivotal role in contemporary economic transactions involving the allocation of resources such as

government bond issues and procurement. The auction system, based on bidders' prices, operates within a set of defined rules. However, it faces challenges in ensuring the reliability of the results while maintaining the confidentiality of unsuccessful bidders. In order to address this issue, the authors propose the introduction of a block chain-based sealed bid auction protocol, which would guarantee privacy while enabling public verifiability. The study demonstrates that the proposed protocol markedly reduces communication costs due to its logarithmic complexity, thereby making it scalable for large-scale auctions. Furthermore, the protocol can be adapted to accommodate various auction formats, including second-price auctions, thus ensuring confidentiality. The security proofs and analyses demonstrate that the scheme effectively meets the goals of privacy preservation and verifiability. This protocol not only secures the auction process, but also increases trust between bidders and stakeholders, leading to a more effective and equitable allocation of resources.

The following scenario will be employed to exemplify the issue. A multinational company engaged in global supply chain operations is confronted with significant challenges to the protection of personal data as it transitions towards the adoption of block chain technology for the purpose of tracking transactions. Although block chain offers the potential for complete transparency, thereby enabling the company to guarantee the integrity and traceability of each stage in the supply chain, there is an increasing awareness of the need to protect sensitive business data. It is of the utmost importance that the company takes the necessary steps to prevent competitors from gaining insight into pricing, volumes and identity details of its partners. A number of existing privacy techniques, including hash networks and ring signatures, are under consideration. However, these either involve intermediaries or require significant computational resources. Moreover, some solutions that operate outside the block chain, while they may be scalable, tend to compromise traceability and the transparency that block chain technology is typically designed to guarantee.

(Konkin & Zapechnikov, 2021) examine the potential of ZKP methods to enhance privacy in enterprise block chain networks, proposing this as a solution to the example scenario described above. The authors commence their discussion by examining the pervasive adoption of distributed ledger technology or block chain, across numerous industries, particularly for the purpose of storing private transactions. The authors identify a number of techniques that can be employed to safeguard privacy, including hash networks, ring signatures and off-chain protocols. However, the authors highlight that these methods have intrinsic limitations, particularly with regard to decentralised data storage and the immutability of private data. The paper addresses two principal concerns pertaining to privacy: identity privacy and transaction privacy. The paper compares existing solutions, such as hash networks that require intermediaries and ring signatures, which, while robust, are resource-intensive. It has been observed that off-chain protocols, which are

commonly used in enterprise block chains, have the potential to compromise key attributes of block chain technology, such as traceability. The authors posit that NIZKP protocols present a promising avenue for addressing these concerns, offering a means of guaranteeing confidentiality while maintaining the integrity and security of block chain systems.

The following scenario is presented for an example. A global retail company that distributes electronic products in multiple regions encounters considerable challenges in monitoring the movement of its products throughout the supply chain. The company requires a system that offers transparency for stakeholders while also ensuring the confidentiality of sensitive information, such as precise pricing, the identities of suppliers and internal logistics details. Furthermore, the company is concerned about the security of data in the event of unauthorised access or data breaches. The efficacy of existing systems that attempt to provide traceability is often undermined by their inability to maintain the confidentiality required by suppliers. This results in shortcomings in the reliability and security of the system. In light of these concerns, the company sought a system that would guarantee product authenticity and safeguard confidential distribution data.

(Uesugi et al., 2021) explore the potential applications of block chain technology in the field of supply chain management, with a particular focus on ensuring traceability and privacy within the system. This is done in order to address the issues highlighted in the aforementioned example scenario. The authors contend that product traceability represents a significant challenge in the context of contemporary supply chains and they underscore the potential of block chain-based systems to provide a unified, transparent database accessible to all stakeholders. The system guarantees the integrity and reliability of distribution data, including ownership transfers. The authors put forth the proposition that public permission less block chains (PPBC) can serve as a scalable solution to preserve privacy while ensuring traceability. In order to address concerns regarding the protection of privacy, the methodology employs cryptographic techniques to conceal distribution data and utilises ZKPs to ensure the confidentiality of block chain addresses throughout the course of transactions between duly authorised suppliers. The system was implemented on Ethereum smart contracts and its functionality was verified, demonstrating that the proposed method effectively tracks product distribution while preserving confidential data. Furthermore, ZKPs are employed to authenticate the ownership of confidential tokens without disclosing the identity of the owner, thus providing an additional layer of security and privacy. The functionality of the system has been validated and its adherence to the original design goals has been verified.

Let us consider the following scenario. A consortium of financial institutions engaged in the processing of large-scale interbank transactions is facing mounting pressure to guarantee the

confidentiality of payment amounts and the identities of the payer and the payee. Given the nature of their transactions, financial institutions require a system that not only ensures confidentiality but also facilitates transparent auditing and transaction verification in accordance with regulatory standards. Nevertheless, the extant block chain solutions are insufficient in addressing these requirements, as they render sensitive data visible while purporting to provide security and transparency. There is a concern among organisations that the use of traditional block chain solutions could result in the compromise of the confidentiality of payments, which could in turn lead to potential breaches of customer data and regulatory violations.

(Kang et al., 2019) examine the pivotal characteristics and privacy concerns associated with block chain technology. They concentrate on the creation of FabZK, a system designed to safeguard the confidentiality of transactions while ensuring data immutability, which is a significant concern in the aforementioned scenario. The principal objective of FabZK is to conceal transactional details within a distributed ledger, encrypt sensitive data (such as payment amounts) and obfuscate transactional relationships (such as those between payer and receiver) within a block chain network. The incorporation of verifiable Pedersen commitments and ZKPs enables FabZK to simultaneously achieve both privacy and auditability. The system was developed as an extension of the open-source Hyperledger Fabric and offers application programming interfaces (APIs) that facilitate the implementation of privacy measures in both client-side and chain code modules. Furthermore, it facilitates the automation of audit procedures based on encrypted data. Hyperledger Fabric, a permissioned block chain platform, was selected for its access control mechanisms, which permit transactions to occur in private channels between recognised entities. The FabZK evaluation demonstrates that it effectively safeguards privacy and performs adequately in data encryption and proof verification, outperforming libSNARK in these areas. However, the study demonstrates that the time required for proof generation increases in proportion to the number of entities involved. Nevertheless, FabZK demonstrates superior performance in terms of throughput, ranging from 5 to 189 times higher than zkLedger. This is attributed to its utilisation of parallel execution in contrast to zkLedger's sequential verification methodology.

In a country where concerns about the integrity of public records and the potential for corruption are growing, a government agency responsible for managing public spending and regulatory compliance is experiencing difficulties in maintaining transparency and accountability in its operations. The organisation handles sensitive data pertaining to contracts, government expenditure and legislative amendments, all of which necessitate the utmost confidentiality. However, the extant mechanisms for public audit and compliance verification are inadequate.

(Goldwasser & Park, 2017) introduced a cryptographic protocol designed to enhance government accountability and transparency, with a particular emphasis on ZKPs as a means of

addressing the concerns outlined in the aforementioned example scenario. They present an efficient record-keeping infrastructure that ensures data confidentiality, facilitates public audits and confirms the compliance of records without compromising the integrity of sensitive information. The system employs block chain and cryptographic techniques, including commitments and zk-SNARKs, to achieve a number of key properties, including indelibility, confidentiality, public controllability of confidential data, accountable deletion and data conciseness. The proposed solution is designed to facilitate auditable record-keeping without the necessity for a trusted third party, in accordance with the interests of governments, corporations and other organisations. The system is designed to minimise data storage while providing irrevocable recording, protection of confidential legislation and efficient public audits. The authors present evidence that the scheme can be implemented on the Ethereum block chain and also propose a simplified version that can be implemented in Bitcoin with less robust guarantees.

2.3.4. General Use Cases

The following case study will present a detailed account of a particular situation or issue. A biotechnology company has expressed its intention to utilise a community of experts (crowdsourcing) in order to conduct a comprehensive, data-driven analysis on a new drug development project. However, two significant challenges emerge in this process. Firstly, there is a potential security risk in the form of the possibility of malicious actors uploading fake or malicious data to the platform. This could result in the corruption of the results of the analysis, thus jeopardising the company's reputation. Secondly, the issue of anonymity is a significant concern in this context. The experts contributing to the project have requested that their work remain anonymous and that it be accessible only to authorised parties for analysis.

(Chen et al., 2021) put forth SeCrowd, a secure and efficient interactive crowdsourcing framework designed to safeguard against external attacks, as a potential solution to the aforementioned concerns. Crowdsourcing is regarded as a valuable service delivery paradigm, whereby assistance is sought from a group of contributors. However, the paper identifies potential security risks in the context of interactive crowdsourcing, where contributors collaborate directly. A significant concern is the possibility of external actors introducing malicious data into the platform. To address this issue, SeCrowd employs a novel cryptographic primitive, namely a permission-based signature, which ensures that the identity of the signer is disclosed solely to authorised parties, thus preventing unauthorised access to sensitive information. The framework comprises four distinct entities: a server (or a distributed set of servers), an administrator, contributors and users. The process requires users to submit their work to the server and subsequently distribute tasks to contributors. The administrator is tasked with verifying the identities of contributors and assigning the requisite skills. In order to facilitate the management of contributions, SeCrowd defines two distinct roles: The initial role is that of the "production" role,

wherein contributors complete their assigned tasks. The subsequent role is that of the "inspection" role, wherein existing contributions are reviewed. The findings of the study demonstrate that SeCrowd is invulnerable under a random oracle model and has been proven to be secure and efficient in terms of computational and communication costs.

The following scenario is presented for example: A government in Europe is seeking to develop a block chain-based system with the objective of modernising the digital identity verification process of its citizens. The objective is to safeguard the privacy of citizens while verifying their identity. To illustrate, when an individual is required to demonstrate their status as a citizen of the EU, they are obliged to divulge a substantial amount of personal data, including their place of birth, address and comprehensive citizenship information. Such practices give rise to infringements of privacy and an elevated risk of exploitation. Furthermore, the elevated transaction costs and lack of security in existing systems erode citizens' trust in the system.

(Morais et al., 2019) examine the growing prevalence of Distributed Ledger Technology (DLT), particularly block chain and the significance of privacy mechanisms within these systems, particularly in the context of the aforementioned scenario. The authors concentrate on ZKP, a cryptographic technique that ensures data authentication while maintaining privacy. The paper outlines a number of strategies for constructing ZKRP, including approaches developed by Boudot (2001), Camenisch et al. (2008) and Bulletproofs (2017). The authors describe the Zero-Knowledge Set Membership (ZKSM) technique as a privacy-preserving method analogous to ZKRP, but applicable to arbitrary sets in lieu of numerical intervals. The authors illustrate the application of this technique in the context of proving that secret data belongs to a specific set. This could, for instance, be used to verify that an individual is a citizen of a particular country. The paper compares and contrasts various methods, with a special focus on Bulletproofs, which stands out due to its efficiency and ability to eliminate the need for a trusted setup. This final point represents a significant advantage for DLT and block chain systems. Bulletproofs are highlighted due to their broader applicability in generating ZKPs beyond range proofs, making them the most suitable approach for ensuring privacy and confidentiality in DLT applications. Furthermore, the paper discusses the implementation of Fiat-Shamir, which is used to transform protocols into non-interactive ones. It provides a comprehensive description of the algorithms required for the efficient deployment of ZKRP.

The following example scenario is provided for illustrative purposes. A multinational pharmaceutical and medical device manufacturer requires the implementation of a decentralised system for the management of secure identities and the sharing of data. The company is encountering considerable difficulties in monitoring patient data, administering volunteers in

clinical trials and safeguarding the dissemination of information with business partners in its global supply chain.

(Lundkvist et al., 2016) provide a detailed examination of the core challenges inherent to contemporary identity management. These include the fragmentation of digital identities across multiple service providers or multi-site organisational structures, as illustrated by the aforementioned scenario. This results in the necessity for repetitive login procedures, which in turn gives rise to the adoption of weak security practices, such as password reuse. Centralised identity providers such as Google and Facebook collect valuable data, rendering them susceptible to unauthorised access. Furthermore, they control billions of IoT devices, increasing their vulnerability to breaches that could disrupt both digital and physical infrastructures. In order to address these issues, the authors put forth the argument that decentralised technologies such as block chain and public-key cryptography facilitate a transition towards a model of self-sovereign identity, wherein individuals retain control over their data. They introduce uPort, an Ethereum-based identity system comprising smart contracts, developer libraries and a mobile application that enables users to securely own and manage their personal information, selectively share it and access services without passwords. uPort also benefits organisations by facilitating customer on boarding and improving Know Your Customer (KYC) processes. In contrast to the prevailing approach of abstracting cryptography from users, uPort provides a recovery mechanism that relies on a majority of trusted individuals to restore a user's identity. This approach improves both security and usability in line with the principles of decentralised technology.

The scenario can be described as follows: A health technology company provides personalised health services to users via a mobile health application designed for elderly individuals. The objective of this application is to facilitate the transmission of notifications to relatives and healthcare teams in emergency situations by monitoring the movements of elderly individuals who are particularly susceptible to falls. Nevertheless, for the application to function as intended, it is essential that the user's location is consistently validated. The company had intended to utilise a central authority to verify the location of users through the application of traditional methods. However, this approach was ultimately abandoned due to concerns that it could potentially violate the privacy of users and compromise the security of sensitive health data. Furthermore, it was anticipated that a centralized system could be susceptible to cyber-attacks due to inherent security vulnerabilities, potentially leading to the misuse of users' private information. The company sought an alternative solution that would provide reliable location verification while protecting user privacy.

(Gambs et al., 2014) address the critical challenge of verifying users' locations in location-based services (LBS) without relying on a central authority, proposing a solution to the problem

mentioned in the example scenario. The reliance on trusted third parties to verify user locations is a common practice that gives rise to concerns pertaining to privacy and trust. In light of the growing computational capabilities of smartphones, the authors put forth PROPS (Privacy-Preserving Location Proof System), a pioneering solution that empowers users to generate verifiable, privacy-preserving location proofs with the aid of nearby devices serving as witnesses. The system effectively counters traditional localisation attacks and ensures security through several key features. Firstly, users are able to retain control over location proofs without exposing personal information. Secondly, the system guarantees anonymity even when the proof is presented to a verifier. Thirdly, the system protects against intermediary attacks. Finally, the system provides mechanisms to detect false location proofs. Furthermore, PROPS ensures context-appropriate privacy by enabling users to selectively share spatial and temporal details. The authors posit that PROPS's decentralised, collaborative approach represents a secure and effective means of supporting location verification in LBS, while prioritising user privacy and autonomy.

The real-world application examples analysed thus far, classified according to various categories, demonstrate that ZKP methods are effectively employed in a multitude of real-world applications. It can be concluded from the aforementioned examples that the majority of real-world applications employ block chain-based structures to fulfill the need for a shared database. In light of the fact that block chain is, in essence, a decentralised structure, it is of paramount importance to guarantee the privacy and security of users. It is therefore unsurprising that zero-knowledge proof methods have emerged as an optimal and efficacious solution to the inherent security and privacy challenges associated with block chain-based structures and are preferred for use in the majority of applications.

3. THEORETICAL FOUNDATIONS

This section presents a rigorous theoretical analysis of the fundamental distinctions among the state-of-the-art NIZKP schemes examined in this study. Particular emphasis is placed on the underlying mathematical structures and the essential operational mechanisms by which each protocol realizes zero-knowledge properties.

3.1. Setup-Based NIZKP Schemes

This section will delve into NIZKP schemes that necessitate with a trusted, universal or transparent setup phase.

3.1.1. Trusted Setup-Based Schemes

ZKP protocols underpin the development of zk-SNARKs. Initially, Goldwasser, Micali and Rackoff introduced interactive ZKPs, followed by Blum, Feldman and Micali's NIZK proofs using a CRS. However, revealing the secret used in CRS generation may enable forgery, prompting the use of trusted setups or secure MPC. Unlike interactive protocols, NIZKPs require only one-way communication, making them particularly suited to blockchain applications. Later contributions focused on minimizing proof size, with Kilian and Micali proposing sublinear and fixed-dimensional proofs. Groth's work on mapping-based NIZKPs enabled efficient verification of polynomial relations without revealing underlying messages, a foundational concept in modern zk-SNARK constructions (Chen et al., 2022).

A zk-SNARK argument involves three steps: setup, proof and verification. In the setup phase, a trusted third party generates the public and verification keys, ensuring the integrity of parameters and the confidentiality of randomness to prevent security breaches. During the proof phase, the prover submits a special input x as proof of knowledge about both x and w. Zk-SNARKs are non-interactive, meaning no direct communication occurs between the prover and verifier, which prevents data leakage and falsified proofs. In the final verification phase, the verifier uses the verification key and proof to evaluate the polynomial, ensuring its correctness and maintaining the system's security (Gong et al., 2022).

One of the schemes examined in this thesis ZKBoo (Giacomelli et al., 2016) is grounded in the "MPC-in-the-head" technique, a conceptual framework for constructing ZKPs based on secure MPC. The essential idea is to simulate a secure three-party computation protocol locally within the prover's mind and then generate a proof by selectively revealing parts of this simulation to the verifier. This approach eliminates the need for actual interaction among physical parties and forms the core of ZKBoo's efficiency and simplicity.

The mathematical structure of ZKBoo can be summarized as follows:

$$C: \{0,1\}^n \rightarrow \{0,1\}^m \tag{3.1}$$

Let (3.1) be a Boolean circuit representing a publicly known relation and let

$$X \in \{0,1\}^n \tag{3.2}$$

(3.2) be a private witness such that

$$C(x) = y, \quad \text{where } y \text{ is the public output.} \tag{3.3}$$

The protocol proceeds by having the prover simulate an MPC protocol among three virtual parties P_1, P_2, P_3 that jointly evaluate $C(x)$. Each party holds a *share* of the input and carries out a share of the computation. The input x is secret-shared among the three parties using a random linear secret-sharing scheme (e.g., additive modulo-2 sharing), such that:

$$x = x_1 \oplus x_2 \oplus x_3 \tag{3.4}$$

Where x_i is the share for party P_i and $\oplus$ denotes bitwise XOR. The computation proceeds gate by gate, maintaining this secret-sharing across all intermediate wires. At each gate, the prover computes the local views of all three parties and records these views.

To produce a ZKP, the prover then commits to all three views using a commitment scheme Com. A random challenge $e \in \{1,2,3\}$ is selected (via Fiat–Shamir if non-interactive) and the prover reveals the two views not indexed by e.

This partial revelation allows the verifier to:

Check consistency between the revealed shares and commitments,

Recompute the computation trace and

Verify correctness of the output against y.

Because only two views are revealed and the third remains hidden, the protocol retains zero-knowledge properties.

The soundness error per round is:

$$\varepsilon = \frac{2}{3} \tag{3.5}$$

because a cheating prover cannot anticipate which of the three views will be requested for opening. To achieve a soundness error of at most 2^{-k} , the protocol is repeated

$$t >= \frac{k}{log_2(\frac{3}{2})} \tag{3.6}$$

times, independently. For instance, achieving 128-bit soundness security requires at least 219 rounds. ZKBoo achieves zero-knowledge through simulation: for any verifier, a simulator can construct convincing transcripts (commitments and openings) without knowledge of the witness, due to the fact that only two views are opened and the third can be arbitrarily set by the simulator.

3.1.2. Universal or Transparent Setup-Based Schemes

This section will delve into systems with a universal or transparent setup. Universal setups are typically characterised by the use of a CRS, which is something that, once established, can be reused in multiple applications or communications. In contrast, systems with a transparent setup do not require a CRS; they have alternative mechanisms such as blockchain technology or cryptographic hash functions (Chen et al., 2023; Chiesa, Hu, et al., 2020; Setty, 2020).

Transparent zk-SNARK is zk-SNARK, which is meant to demonstrate the satisfiability of NP expressions as expressed in R1CS. The schemes are mainly aimed at achieving sublinear verification costs for NP expressions of arbitrary complexity while providing time-optimal provers. The first contribution of these works is to achieve these sublinear verification costs using a pioneering Arthur-Merlin interactive information-theoretic argument. Also, these works propose novel techniques to improve the efficiency of the total check protocol. A defining characteristic of these works is their ability to accommodate any R1CS sample while attaining the same sublinear verification cost, a goal never before attained without increasing the depth of the circuit (Setty, 2020).

Spartan is a modern ZKP system designed for efficient verification of computations expressed as R1CS. Its design achieves scalability without relying on trusted setup ceremonies, making it suitable for decentralized and trust-minimized environments. Spartan leverages the concept of *polynomial commitments* and builds upon two key components: a verifiable computation scheme (specifically, a SNARK-like structure) and an *inner product argument* for scalable proof sizes and verification costs (Setty, 2020).

$$C(x, w) = 0 \tag{3.7}$$

Let 3.7 be an R1CS instance, where x is the public input and w is the witness. The goal is to prove that the prover knows a witness w such that the relation holds. Spartan encodes this constraint system as multivariate polynomials, enabling algebraic manipulation suitable for succinct proof generation. The R1CS is transformed into an equivalent representation of a *Quadratic Arithmetic Program* (QAP), where the constraints are represented as polynomials $A(X), B(X), C(X)$, such that:

$$A(X) \cdot B(X) - C(X) = 0 \tag{3.8}$$

holds at all evaluation points corresponding to the computation gates. The prover then constructs low-degree polynomials from the wiring of the R1CS and commits to these using a polynomial commitment scheme.

Spartan employs a variant of the Dark polynomial commitment scheme, which is transparent and based on cryptographic hash functions (rather than elliptic curves or pairings). The commitment scheme enables the prover to commit to polynomial evaluations and prove consistency between committed polynomials and evaluation claims. An essential innovation in Spartan is the use of an inner product argument for verifying polynomial evaluations efficiently. This allows the verifier to check that the committed polynomials satisfy the QAP relation without evaluating the entire polynomial. The inner product argument reduces communication and verification complexity to logarithmic size in the number of constraints. Spartan transforms its interactive proof into a NIZKP using the Fiat–Shamir heuristic in the random oracle model. The prover generates challenges by hashing the protocol transcript, thus removing the need for interaction and enabling applications in blockchain environments and verifiable computation (Setty, 2020).

In place of a CRS, researchers propose a method for building preprocessed zk-SNARKs with a universal and updatable structure. This comes as a new implementation of the holographic method talked about by Babai et al. in STOC 1991 for speedy verification based on the input of the expression in some coded form. The schemes proposed obtain a preprocessed zk-SNARK with a linear size for the SRS and constant size for the arguments. The methodology is based on the algebraic holographic proof (AHP) for R1CS, which has been shown to yield linear proof lengths and constant query complexities. The proposed schemes resolve this disparity of efficiency between universal SRS and circuit-specific SRS constructions, thereby providing a general approach to building succinct interactive arguments which can be made non-interactive by the Fiat-Shamir paradigm (Chiesa, Hu, et al., 2020).

Marlin is a universal and updatable transparent zk-SNARK designed for efficient proof generation and verification without relying on trusted setup ceremonies. It builds upon the Polynomial Commitment-based framework, specifically improving upon the Sonic protocol to achieve better prover time and verifier succinctness. Marlin operates over arithmetic circuits represented as R1CS and enables scalable and reusable CRS, a key feature for applications requiring universal setup, such as general-purpose blockchains and privacy-preserving systems.

Let (3.7) be a statement in the R1CS form, where x is the public input and w is the witness. Marlin expresses this computation as:

$$A \cdot z \circ B \cdot z = C \cdot z \tag{3.9}$$

where $z = (x, w, 1)$ is the extended witness vector (including a constant term) and A, B, C are sparse matrices encoding the circuit constraints. In 3.9 ° means that hadamard multiplexing (element based multiplexing) operation. This system is then encoded into polynomials, transforming the R1CS into an algebraic form suitable for probabilistic proof generation via polynomial identity testing.

Marlin uses the Kate, Zaverucha and Goldberg (KZG) polynomial commitment scheme, which provides succinct and binding commitments to polynomials. This scheme allows a prover to commit to a polynomial $f(x)$ and later prove that it evaluates to a particular value y at a challenge point α, without revealing the entire polynomial. This commitment scheme supports *evaluation proofs* that are succinct and can be verified efficiently by checking pairing equations over elliptic curves. The core of Marlin is an IOP system that checks consistency and satisfiability of the polynomialized R1CS. This protocol is converted into a NIZK using the Fiat–Shamir transform, which replaces interactive challenges with hash-derived randomness in the ROM. To check that the committed witness polynomials satisfy the R1CS constraints, the verifier performs a low-degree test and checks a set of random linear combinations of the polynomials using the sumcheck protocol and polynomial evaluations at random points (Chiesa, Hu, et al., 2020).

3.2. Setup-Free NIZKP Schemes

The present section shall review the zk-STARK and Bulletproofs studies, which are setups not requiring any pre-processing.

3.2.1. Bulletproofs

Bulletproofs algorithms were developed as an alternative to zk-SNARK algorithms, which have certain disadvantages, such as requiring pre-installation or reliance on a trusted third party—factors that limit their applicability in real-world scenarios. Bulletproofs are NIZKP protocols designed to provide concise proofs without the need for a trusted setup. First introduced by Bootle et al. in December 2017, Bulletproofs use advanced cryptographic techniques to improve efficiency and scalability. Unlike traditional ZKP systems that rely on a trusted setup, bulletproofs eliminate this requirement while maintaining strong security guarantees. A key advantage of Bulletproofs is their spatial efficiency, particularly in range proofs. Compared to traditional privacy-preserving protocols, Bulletproofs significantly reduce proof sizes and improve verification speed, making them highly practical for blockchain-based applications. Proof size grows logarithmically with witness size, offering significant improvements over traditional linear-sized range proofs. This makes Bulletproofs particularly effective for verifying committed values within a given range, requiring only $2\log_2(n)+9$ group and field elements, where n represents the bit length of the range. In addition, both proof generation and verification have linear complexity in n, ensuring computational efficiency. By relying solely on the discrete logarithm assumption, bulletproofs

enable efficient ZKPs for general arithmetic circuits without the need for a trusted setup. Their compact nature makes them particularly suitable for privacy-enhancing mechanisms in cryptocurrencies, such as confidential transactions in bitcoin and other blockchain systems. The decentralised and trustless design of Bulletproofs aligns well with the principles of distributed ledger technologies, providing a scalable and secure approach to privacy-preserving computation (Bünz et al., 2018; Gong et al., 2022).

Bulletproofs are particularly efficient for range proofs and general arithmetic circuit proofs, with logarithmic-size proofs and verifier time. The protocol relies on the Discrete Logarithm (DL) assumption and is designed to be compatible with any group where the DL problem is hard, such as secp256k1 used in Bitcoin. Bulletproofs are constructed within the inner-product argument framework, which enables proof sizes and verification time that grow logarithmically with the circuit size. At the core of Bulletproofs are a protocol for proving that the inner product of two committed vectors $\mathbf{a,b} \in \mathbb{Z}_q^n$ is equal to a publicly known scalar *c*, i.e.,

$$\langle a, b \rangle = c \tag{3.10}$$

while keeping $a\ and\ b$ hidden. The prover commits to the vectors using Pedersen commitments:

$$\mathrm{P} = \sum_{i=1}^{n} a_i \cdot G_i + b_i \cdot H_i + c \cdot U \tag{3.11}$$

Where G_i, H_i, U are publicly known generators of the group $\mathbb{G}$. The proof uses a recursive logarithmic protocol that reduces the problem to scalar operations over $\log_2(n)$ rounds. Bulletproofs utilize additively homomorphic Pedersen commitments:

$$Com\,(v, r) = v \cdot G + r \cdot H \tag{3.12}$$

which hide a value v using a blinding factor r. These commitments are binding and hiding under the discrete logarithm assumption. This allows proving that a committed value lies within a certain range (e.g., $[0, 2^{64})$) without revealing the value. To prove that a committed value $v \in [0, 2^{64})$, Bulletproofs represent v in its binary decomposition and use inner-product arguments to prove that each bit is 0 or 1. The protocol creates an arithmetic circuit for this constraint system and uses the inner-product argument to prove correctness of all constraints. The size of the range proof grows as:

$$Proof\ Size = 2 \cdot log_2\,(n) + 9\ group\ elements \tag{3.13}$$

which is highly efficient compared to traditional range proof protocols. Bulletproofs are secure under: The Discrete Logarithm (DL) assumption, The ROM (for non-interactive zero-knowledge

via the Fiat–Shamir heuristic). Algorithm provide soundness, completeness and zero-knowledge properties, with no need for a trusted setup (Bunz et al., 2018).

3.2.2. STARKs

In 2018, Eli Ben-Sasson, Iddo Bentov, Yinon Horesh and Michael Riabzev introduced zk-STARKs, a new ZKP system designed to enhance zk-SNARKs. Zk-STARKs offers a more efficient and scalable approach to cryptographic proofs while eliminating the need for a trusted setup. The 'T' in zk-STARKs signifies transparency, as they are based on publicly verifiable cryptographic primitives as opposed to a CRS created by a trusted party. From a technical perspective, zk-STARKs achieve this transparency by employing collision-resistant hash function-based symmetric verification mechanisms that eliminate the reliance on complex cryptographic assumptions associated with zk-SNARKs. In contrast to zk-SNARKs, which are reliant on computationally expensive cryptographic arguments (represented by "AR"), zk-STARKs circumvent these constraints, a property that lends them enhanced resilience to potential threats posed by quantum computing. A notable benefit of zk-STARKs is their enhanced efficiency in proof generation and verification. The protocol demonstrates effective scalability in the face of increasing computational complexity, adeptly managing the growing volume of data exchanged between the prover and verifier. However, a notable drawback of zk-STARKs is the relatively large proof size compared to zk-SNARKs. This results in longer verification times and higher computational costs, especially in blockchain applications where proof verification directly affects transaction fees. Notwithstanding this trade-off, zk-STARKs present a compelling alternative for privacy-preserving cryptographic proofs, particularly in contexts where trust minimisation and quantum resilience are paramount (Chen et al., 2022; Gong et al., 2022).

STARKs are designed to generate succinct, NIZKP for computations represented as arithmetic traces and rely on CRHF rather than elliptic curve cryptography. Zk-STARKs build on IOPs and employ probabilistic polynomial arguments to prove that a computation was executed correctly, without revealing the witness.

STARKs model computation as a sequence of state transitions over a deterministic finite automaton. Each step in the computation corresponds to a row in a table—called the execution trace—that records the state of the system at that point. Let the execution trace be represented as a matrix $T \in \mathbb{F}^{nxw}$, where : n: number of steps in the computation, w: number of state registers per step, $\mathbb{F}$: a large prime field. The prover claims that the execution trace satisfies a set of algebraic constraints that capture the computation's correctness. These constraints are encoded as polynomials over $\mathbb{F}$ that must evaluate to zero over a certain domain. The verifier checks that:

$$C_1(f_1(X), f_2(X), \dots f_w(X)) = 0 \qquad \forall j \in [n] \tag{3.14}$$

for all constraint polynomials C_i. These are often enforced via a quotient polynomial test, ensuring that the constraint polynomial is divisible by a vanishing polynomial $Z(X)$ over the evaluation domain:

$$\frac{C(X)}{Z(X)} \in \mathbb{F}[X] \tag{3.15}$$

To ensure that all committed polynomials are of low degree, zk-STARKs employ a Reed-Solomon code-based proximity test, often implemented via Fast Reed-Solomon Interactive Oracle Proofs of Proximity (FRI). The prover provides commitments to the evaluations of the polynomials and the verifier performs randomized checks to confirm that:

$$\deg(f_i) \leq d \tag{3.16}$$

This test ensures soundness with high probability and is sublinear in complexity. The protocol is made non-interactive through the Fiat–Shamir heuristic, which derives verifier challenges via hash functions applied to the transcript. Zero-knowledge is achieved by masking the witness with a random codeword in the Reed-Solomon space, ensuring that the committed polynomials do not reveal private information (Ben-Sasson et al., 2018).

3.3. Summary of State-of-the-art NIZKP Schemes

To summarise, the contemporary NIZKP protocols evaluated in this study have been systematically compared and analysed based on a set of fundamental performance characteristics and cryptographic properties. As illustrated in Table 3.1, this comparative framework—adapted from Oude Roelink et al. (2024)—encompasses key attributes such as proof size, computational complexity in proof generation and verification, the necessity of a trusted setup, post-quantum security assumptions and the underlying cryptographic hardness assumptions associated with each protocol.

Table 3.1. Comparison of state-of-the-art NIZKP protocols (Oude Roelink et al., 2024)

Features	**zk-SNARK**	**zk-STARK**	**Bulletproofs**
Proof size	Constant	Polylogarithmic	Logarithmic
Proof generation	Linear	Quasilinear	Linear
Proof verification	Linear	Polylogarithmic	Linear
Trusted Setup	Yes	No	No
Quantum Secure	No	Assumed	No
Cryptographic Assumptions	EC-DLP, B-DHP	Cryptographic Hashes	EC-DLP

EC-DLP: Elliptic Curve Discrete Logarithm Problem, B-DHP: Bilinear Diffie-Hellman Problem

4. MATERIALS AND METHODOLOGY

This section provides a detailed information of the hardware and software specifications of the experimental platforms employed in this thesis, along with the criteria guiding their selection. It further detailed on the cryptographic algorithms under evaluation and detailed outlines the methodologies adopted for their testing and performance analysis.

4.1. Materials

In the following sections, all components that constitute the basic technical infrastructure of the study and the functions and selection criteria of each of them are discussed in more detail. The general structure of the benchmarking platform consisting of software and hardware infrastructure and the algorithms used is shown in detail in Figure 4.1.

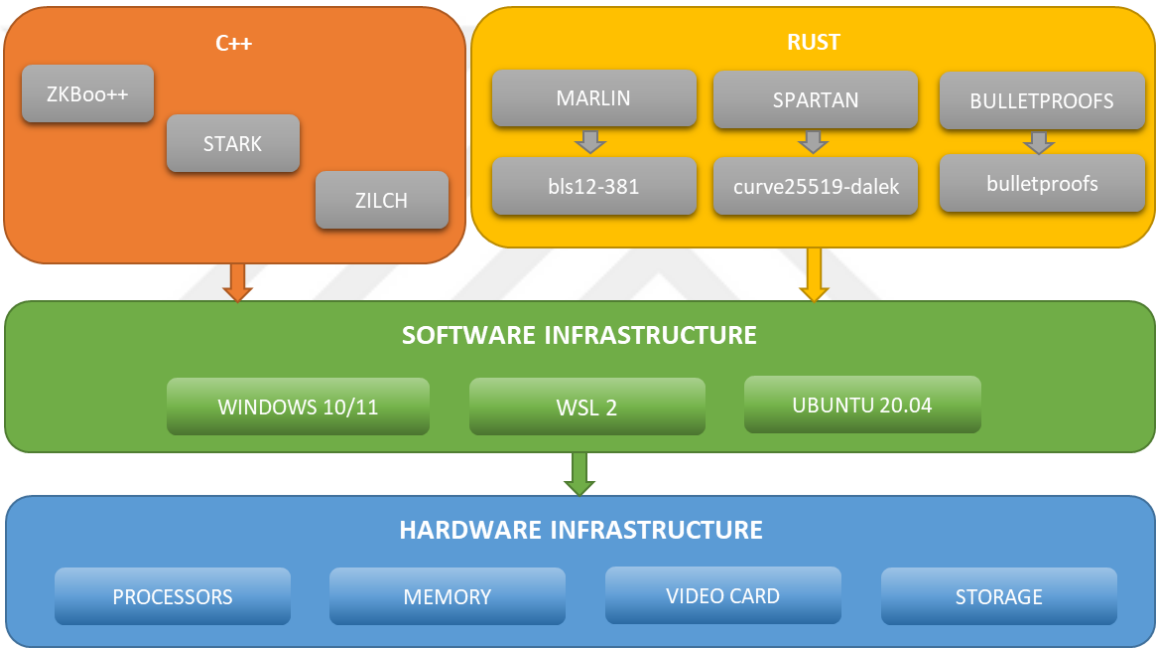


Figure 4.1. Overview of the Benchmark Infrastructure

Figure 4.1 illustrates the benchmark infrastructure employed in this study, encompassing both software and hardware components. The software infrastructure utilizes Rust and C++ for implementing six NIZKP algorithms, with Rust selected for its memory safety, high performance and cryptographic libraries. The development environment is based on Ubuntu 20.04.6 LTS within Windows Subsystem for Linux (WSL) 2, ensuring efficient Linux-based execution on a Windows platform. The hardware infrastructure consists of four platforms, ranging from high-performance systems to portable configurations, enabling an assessment of hardware variability on algorithm performance.

4.1.1. Software Infrastructure

This section presents the software components used throughout the benchmarking phase of the thesis, including the programming language, development environment and virtualization sub system. Each component is discussed with respect to its selection rationale, version information, update status and specific role in the experimental setup.

It has been determined that 3 of the 6 NIZKP algorithms evaluated in this thesis were developed in Rust programming language and the other 3 in C++ programming language. C++ is a proven development environment and has its compiler and dependencies built into almost all Linux distributions including Ubuntu. *C++ 9.4.0* version was used in the study. Although Rust is a newer development environment compared to C++, it was selected based on the following advantages:

- **Memory safety:** Rust provides compile-time guarantees for memory safety without relying on garbage collection.
- **High performance:** Rust compiles to native binaries with performance comparable to C/C++.
- **Extensive ecosystem:** Rust provides a wide range of libraries for cryptography, ZKPs and system programming (e.g. curve25519-dalek, bulletproofs, arkworks).

Rust version *1.86.0* was used throughout the study. This version was selected due to its stability and compatibility with the WSL-based Ubuntu environment. The official rustup tool was used to install and manage the Rust toolchain and updates were applied using the rustup update command. Dependency management was performed using Cargo, Rust's official package manager, which facilitates consistent builds and testing across different configurations. Rust was actively used in all phases of the experiments, including algorithm implementation, compilation, execution and performance measurement.

The guest operating system used during the experimental studies was Ubuntu Linux. Ubuntu is a widely adopted, open-source Linux distribution known for its stability, active community support and suitability for academic and research-focused projects. The key reasons for selecting Ubuntu were:

- **Balance between stability and up-to-dateness:** Long Term Support (LTS) versions offer five years of guaranteed updates, ensuring long-term reliability.
- **Comprehensive software support:** Ubuntu provides an extensive package repository and robust dependency management, essential for cryptographic research.
- **Compatibility with Rust and related libraries:** The Rust compiler and ZKP-related dependencies are seamlessly installable and executable on Ubuntu.

Throughout benchmarking process, *Ubuntu 20.04.6 LTS* was used. This version is fully compatible with the latest versions of C, C++ and the Rust compiler and provides long-term support. Before starting the experiments, the system was upgraded to the latest known update level using the standard apt update && apt upgrade commands. Basic build tools and dependencies such as build-essential, libssl-dev, pkg-config, cmake and clang were installed to support the Rust and C++ development environment. Ubuntu served as the primary environment for compiling experimental algorithms, managing dependencies and performing benchmark evaluations. No system-level incompatibilities or compilation issues were encountered during the experiments.

Since the physical testbed machine ran on the Windows operating system, a Linux-based development environment was established using the WSL. WSL allows native execution of Linux distributions within Windows, eliminating the need for virtual machines and offering a lightweight, high-performance Linux experience. The version used in this study was *WSL 2*. WSL 2 offers enhanced performance through full system call compatibility and improved I/O operations via an integrated virtualization architecture. Ubuntu 20.04.6 LTS was installed and executed within WSL 2 without issues. The rationale for choosing WSL includes:

- **Improved hardware access and performance:** Unlike virtual machines, WSL directly interfaces with Windows kernel resources, enhancing performance.
- **Development environment integration:** Tools such as Visual Studio Code, Git and the Rust toolchain integrate seamlessly in a hybrid Windows–Linux workflow.
- **Full system call support:** Essential for building and running low-level Rust cryptographic libraries, many of which rely on advanced system features.

Although all experimental procedures were executed within the Ubuntu environment, the underlying platform was WSL on Windows. This approach allowed efficient Linux-based development while preserving compatibility with the broader Windows-based development toolset.

4.1.2. Hardware Infrastructure

To ensure the reliability and reproducibility of results, this study is designed to benchmark the same algorithm on diverse hardware configurations within an identical software environment, thereby isolating hardware as the sole variable. This approach will allow us to assess how hardware variations impact algorithm performance. To achieve this, four distinct hardware platforms were selected, each varying in terms of processor power, memory configurations and support for external graphics cards. All algorithms were adapted to function on these platforms and each algorithm was executed individually on each system.

When selecting the hardware platforms, factors such as the processor model, memory type and amount and the graphics card were considered. Notably, to evaluate the performance of the

algorithms on portable platforms with generally limited processing power, one of the platforms chosen was a portable system. The specifications of the aforementioned hardware platforms can be found in Table 4.1.

Table 4.1. Overview of hardware platforms' features

Features	System 1	System 2	System 3	Portable
Processor	Intel i7-12700F	Intel i7-12700	Intel i7-7700	Intel i5-8265U
P Cores	8 @ 4.80 Ghz	8 @ 4.80 Ghz	4 @ 4.20 Ghz	4 @ 3.90 Ghz
E Cores	4 @ 3.60 Ghz	4 @ 3.60 Ghz	n.a.	n.a.
Threads	20	20	8	8
RAM	32 GB	16 GB	8 GB	8 GB
Video Card	NVIDIA RTX 4060	Intel UHD Graphics 770	Intel HD Graphics 630	Intel UHD Graphics 620
Storage	1024 GB NVME SSD	512 GB NVME SSD	256 GB NVME SSD	256 GB M2 SATA SSD

4.2. Methodology

In order to enable a fair and systematic comparison of selected NIZKP algorithms, a consistent benchmarking methodology was adopted throughout this study. The following section outlines the criteria used in selecting representative algorithms, as well as the procedures implemented to ensure software uniformity, compatibility and repeatability of results across different hardware platforms. Particular attention was given to controlling software dependencies and execution environments to isolate hardware as the primary variable under investigation.

4.2.1. Algorithm Selection Methodology

The present study undertakes a systematic comparative analysis of practical implementations derived from multiple NIZKP protocols, with the aim of assessing their relative performance across experimental or research-oriented applications. Through this comparative analysis, the study aims to systematically assess and differentiate the performance characteristics of selected NIZKP protocols, enabling a comprehensive evaluation of their output metrics. To address an existing gap in the literature, one or two representative algorithms were chosen from each of the three principal NIZKP categories—zk-SNARK, zk-STARK and Bulletproofs—prioritising those that have received limited attention in previous empirical research. The objective is to provide meaningful comparisons not only between the protocols themselves but also by testing them on different hardware platforms. Furthermore, an intentional and systematic approach was adopted to compile and execute the protocols within a consistent programming environment, to the extent permitted by their respective technical constraints. This methodological approach serves to

eliminate the challenge of comparing performance across varying development environments, which has been identified as a significant limitation in previous protocol comparisons.

A review of the extant literature (Capko et al., 2022; T. Chen et al., 2021, 2022; El-Hajj & Oude Roelink, 2024; Gong et al., 2022; Khovratovich, 2018; Nitulescu, 2020; Oude Roelink et al., 2024; Panait & Olimid, 2021) reveals that algorithms from the three primary categories have been developed and optimised in significantly different programming environments. This discrepancy gives rise to performance variations, even among algorithms within the same category. Consequently, the initial categorization of the algorithms was undertaken, alongside an analysis of their potential commonalities. A salient challenge encountered during this process pertained to the incompatibility of algorithms developed within the same group and utilising the same development environment, which gave rise to issues when attempting to work collaboratively or compile them with different compiler versions. It is noteworthy that certain algorithms developed in Rust have demonstrated dependency conflicts or mismatched versions, resulting in significant compilation challenges.

In selecting the algorithms, considerable effort was made to include at least one algorithm from each of the subclasses shown in Figure 2.1. Table 4.2 shows the programming languages used to develop the algorithms that will be compared and analyzed in this study.

Table 4.2. Overview of algorithms and development platforms

ZKP System	**Protocol**	**Setup Type**	**Post-Quantum Secure**	**Development Platform**
ZKBoo++	zk-SNARK	Trusted	yes	C++
Spartan	zk-SNARK	Transparent	yes	Rust
Marlin	zk-SNARK	Universal	no	Rust
Bulletproofs+	sh-SNARK	Transparent	no	Rust
Zilch	zk-STARK	Transparent	yes	C++
Stark	zk-STARK	Transparent	yes	C++

Of particular note is the selection of an application to represent each of the three subgroups of the zk-SNARK group: trusted, universal and transparent. This necessitated the identification of a common programming framework and the resolution of dependency issues. The library dependencies of 3 of the 6 algorithms used in our study developed with the Rust programming language are shown in Table 4.3.

A well-documented and optimised Rust implementation was selected for the Bulletproofs application, which served as a best practice example. Subsequent exploration then focused on the potential of Rust implementations for other algorithms within the same category. However, the

superior performance of C++ implementations for STARK algorithms posed a significant challenge to the identification of Rust implementations for all applications. Table 4.4 shows the working versions of the programming languages and compilers used in this study.

Table 4.3. Dependencies used for compilation

Dependency Name	**Dependency Version**
bulletproofs	5.0.0
curve25519-dalek	4.1.3
hex	0.4.3
merlin	3.0.0
rand	0.8.5
bincode	1.3.3
byteorder	1.5.0
digest	0.8.1
flate2	1.0.35
itertools	0.10.5
rand_core	0.6.4
serde	1.0.217
sha3	0.8.2
subtle	2.6.1
criterion	0.3.6
ark-ff	0.3.0
ark-poly	0.3.0
ark-poly-commit	0.3.0
ark-relations	0.3.0
ark-serialize	0.3.0
ark-std	0.3.0
derivative	2.2.0
digest	0.9.0
rayon	1.10.0
ark-bls12-381	0.3.0
ark-mnt4-298	0.3.0
ark-mnt4-753	0.3.0
ark-mnt6-298	0.3.0
ark-mnt6-753	0.3.0
blake2	0.9.2
rand_chacha	0.3.1

Table 4.4. Versions of software used for compilation and benchmark

Compiler	**Version**
g++	9.4.0
rustc	1.86.0-nightly
cargo	1.86.0-nightly

Consequently, a hybrid approach was adopted, utilising Rust implementations for three of the six application algorithms and C++ implementations for the remaining three. This approach was selected to achieve a balance between compatibility, efficiency and performance across the various algorithms.

4.2.2. Benchmarking Methodology

In order to rectify any residual system resources and guarantee a uniform initial state, each hardware platform under evaluation was subjected to a reboot. Subsequent to the aforementioned reboot, two-minutes waiting period was enforcement to ensure that all core initialization processes have been fully initialized. The WSL console application was then initiated and the Ubuntu 20.04 operating system was permitted to stabilise. To execute the benchmarking procedures under controlled conditions, an Ubuntu terminal session was initiated with the root user. The individual algorithms were then switched to their respective directories and the Rust and C++ applications were launched.

The correct default configurations were applied and the number of iteration rounds was set in the benchmark code. Subsequently, the 'cargo bench' command was issued, which compiled the Rust code as a release target for optimal performance. Thereafter, the benchmark was executed for the tested protocol using the aforementioned compiled binary file. Upon completion of the benchmarking for the Rust applications, the benchmark results and other measurement outputs were saved in an Excel sheet for each protocol under the specified number of iteration rounds. Subsequent to the Rust benchmark results being recorded, the relevant directories in the same terminal were accessed for the C++ application and the process was repeated using the suggested compile and run commands for each algorithm. These commands, akin to the 'cargo bench' command for Rust, compiled the C++ applications and executed the benchmark, thereby yielding the desired results. Subsequent to the execution of all benchmarks for the specified number of iteration rounds, the results were meticulously documented in a single Excel file.

A further experiment was conducted in order to compare the performance of the Bulltproofs application for different options. The process for this benchmark was similar to the procedure previously described, with the number of rounds being kept constant. The benchmark was run using the AVX instruction set, with the AVX switch being turned on in the 'Config.toml' file and the 'Cargo Bench' command being started. This process was repeated to obtain the performance difference caused by a single option parameter change. Subsequently, the benchmark was executed without utilising the AVX instruction set by deactivating the AVX switch in the 'Config.toml' file and initiating the 'Cargo Bench' command, thereby yielding the performance difference caused by a single option parameter change. Subsequently, the benchmark results and measurements were saved in an Excel spreadsheet. Thereafter, the option parameter was reset to its

default setting and this process was repeated on all hardware platforms. Having completed the execution of all the aforementioned benchmarks, the measurements obtained from the Excel sheet were then plotted into the result tables and graphs.

5. RESULTS AND DISCUSSION

This section presents and interprets the experimental results obtained from benchmarking selected NIZKP protocols across various hardware platforms. The primary aim is to analyse key performance metrics—namely, proof generation time, proof verification time and proof size—within a consistent software environment. The discussion further examines the broader implications of these findings, contrasts them with results from alternative platforms and underscores the influence of hardware on the overall efficiency and practical viability of each protocol.

5.1. Results of System 1 Hardware Platform

As outlined in Table 4.1, System 1 is distinguished by its high-performance hardware specifications, representing the most advanced configuration among the evaluated platforms. It is hypothesised that the results of the tests performed on this platform will exceed those of the other platforms. The objective of this thesis is to ascertain the impact of processor power, memory capacity and video card on performance, utilising this advanced hardware configuration. The results obtained from these tests will serve as a benchmark for future comparisons with other platforms, facilitating a comprehensive evaluation of the system's capabilities.

As illustrated in Table 5.1, the proof production times of the six algorithms tested on the System 1 hardware platform are presented in ascending order, from the shortest to the longest. It is evident that three of the six algorithms exhibited a proof production time of less than one second. This observation indicates that the majority of the algorithms are well-suited for deployment in real-world scenarios and possess relatively acceptable proof production times.

Table 5.1. Results of proving time of system 1 platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proving Time (ms)
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	204.94
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	205.22
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	893.48
Chung et al.	2022	Bulletproofs+	Bulletproof	Transparent	no	1373.28
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	30503.03
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	43760.00

Upon analysis of the data presented in the table, it is evident that zk-SNARK algorithms, with the exception of ZKBoo++, demonstrate suboptimal performance in terms of proof generation time. In contrast, zk-STARK algorithms have been found to demonstrate a strong performance in this regard. The Bulletproofs algorithm, in particular, has been observed to generate proofs in excess of one second, a duration that is likely to have a detrimental effect on the preference, particularly in real-world applications where bandwidth is limited. It is notable that ZKBoo++ from

the zk-SNARK group and Zilch algorithms from the zk-STARK group generate results with a high degree of similarity and that these algorithms can be regarded as secure even in the period following the quantum revolution, as evidenced by the enhancements in their structures as outlined in the original studies. This observation underscores their status as preferred algorithms for numerous real-world applications. A thorough examination of the original studies reveals that the test results obtained for Spartan and Marlin algorithms are consistent with the existing literature. It is observed that the ZKBoo++ algorithm performs well for the zk-SNARK group and that the improvements made yield positive results.

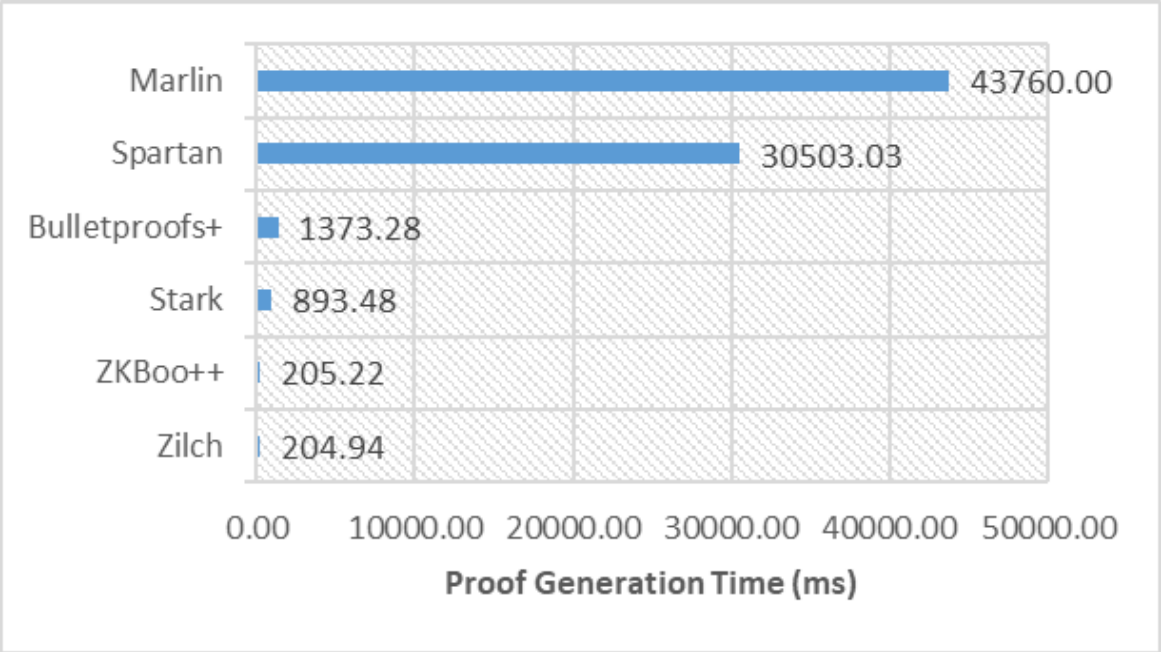


Figure 5.1. Overview of Proof Generation Performance on System 1

To assess the efficiency of each protocol, Figure 5.1 presents a comparative analysis of the proof generation times for all six algorithms under uniform test conditions. It is evident from this analysis that the Marlin and Spartan algorithms demonstrate a clear separation, with the former exhibiting a pronounced tendency to generate proofs more expeditiously. Consequently, it is deduced that these algorithms are optimally suited for server-side applications characterised by high-powered server-client architectures. Conversely, the Zilch and ZKBoo++ algorithms are deemed more suitable for peer-to-peer applications and systems with limited system resources, such as IoT applications, due to their comparatively brief proof generation times. The Bulletproofs+ algorithm is considered suitable for applications with relatively higher bandwidth, such as identity verification, where the relative waiting time is not a problem.

The time required for proof verification by each of the six algorithms tested on the System 1 hardware platform is presented in Table 5.2, with the values listed in increasing order from the shortest to the longest in milliseconds.

Table 5.2. Results of verification time of system 1 platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Verify Time (ms)
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	27.49
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	39.96
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	92.01
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	104.94
Chung et al.	2022	Bulletproofs+	Bulletproof	Transparent	no	149.41
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	5180.98

It is evident that the zk-STARK algorithms demonstrate a marked superiority. It is observed that two algorithms in the zk-SNARK group demonstrate comparable performance, ranking just behind the top performers. It is evident that four of the six algorithms exhibit a proof verification time of near 100 milliseconds or less. This observation indicates that the majority of the algorithms are well-suited for deployment in real-world scenarios and possess a satisfactory level of proof verification efficiency. The Bulletproofs+ algorithm is observed to be closely positioned behind the aforementioned group of four, exhibiting performance that is also within acceptable limits. The Marlin algorithm is hypothesised to generate a proof verification time that is very high, similar to the proof generation process and therefore its use in real-world applications can only be under very special conditions. In addition to these data, the fact that all of the algorithms in the top four are considered safe for the post-quantum era is also promising in terms of long-term use of these algorithms.

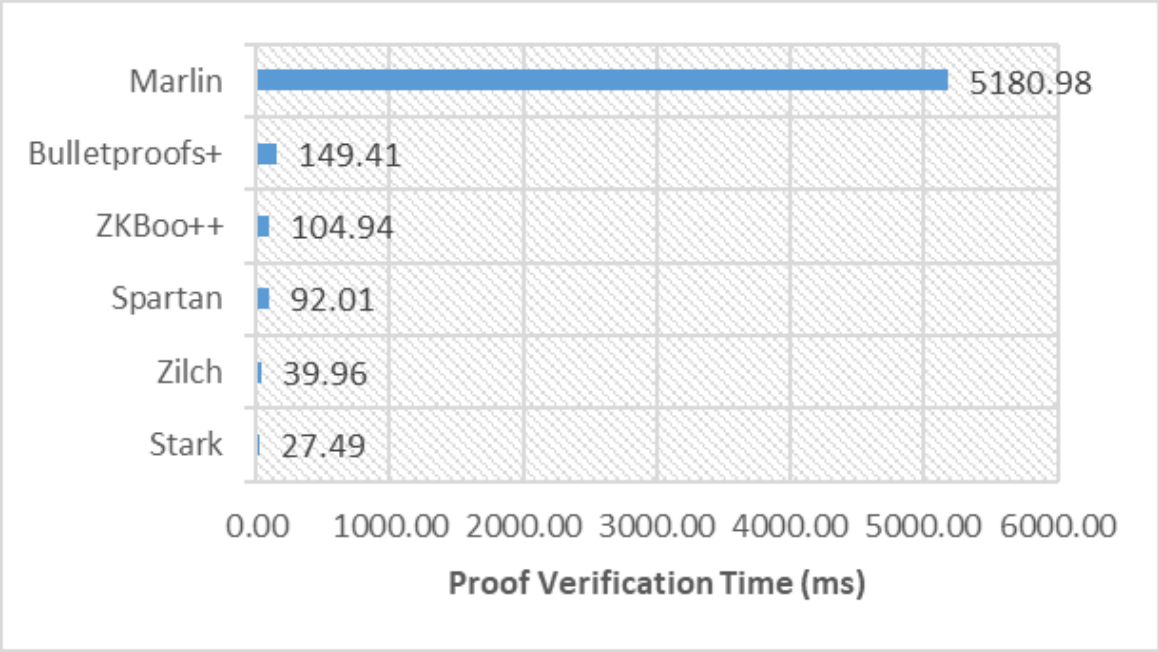


Figure 5.2. Overview of Proof Verification Performance on System 1

The proof verification times of the six algorithms in this study can be observed in Figure 5.2, presented in a comparative manner. A thorough analysis of this figure reveals that the Marlin algorithm exhibits a marked negative deviation from the other algorithms. This finding lends further credence to the hypothesis that the Marlin algorithm is best suited to server-side applications characterised by high-powered server-client architectures. Conversely, the Zilch and Stark algorithms are regarded as more favourable for general applications and those with limited system resources due to their notably brief proof verification times. It is hypothesised that the Bulletproofs+ algorithm will be applicable to many real-world applications, given that its proof verification time is not significantly longer than that of the first four algorithms that demonstrated good performance.

The proof sizes yielded by each of the six algorithms tested on the System 1 hardware platform as a consequence of the proof production process are presented in Table 5.3, with the values listed in increasing order of kilobyte size from smallest to largest.

Table 5.3. Results of proof size of system 1 platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proof Size (KB)
Chung et al.	2022	Bulletproofs+	Bulletproof	Transparent	no	0.67
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	141.77
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	401.64
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	774.28
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	4512.00
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	n.a.

When considering proof size as a criterion, the Bulletproofs+ algorithm demonstrates a clear advantage. However, when executing the application, no data regarding the proof size of the Marlin algorithm could be obtained. The original study reports that for the BLS12-381 elliptic curve, the proof size is 880 bytes. Based on this information, despite the Marlin algorithm's relatively poor performance in proof generation and verification times, it exhibits strong efficiency in terms of proof size. Drawing on the findings from the original study, the Marlin algorithm emerges as significantly superior to the majority of the other algorithms, ranking second only to Bulletproofs+. Among the six algorithms analyzed, Bulletproofs+ produces the smallest proof sizes, as expected. Given this characteristic, Bulletproofs+ appears to be the most suitable option for applications with low bandwidth constraints, heavy network traffic or blockchain-based systems that face challenges related to storage and data transmission over the network. Furthermore, the study indicates that the proof sizes of the zk-SNARK and zk-STARK algorithms, which follow Bulletproofs+, are significantly larger in comparison. As a result, these algorithms may not be ideal for applications requiring efficient storage and low network overhead. Similarly, although the

ZKBoo++ algorithm demonstrates strong performance in proof generation and verification, it suffers from excessively large proof sizes, potentially limiting its practical applications despite its computational efficiency.

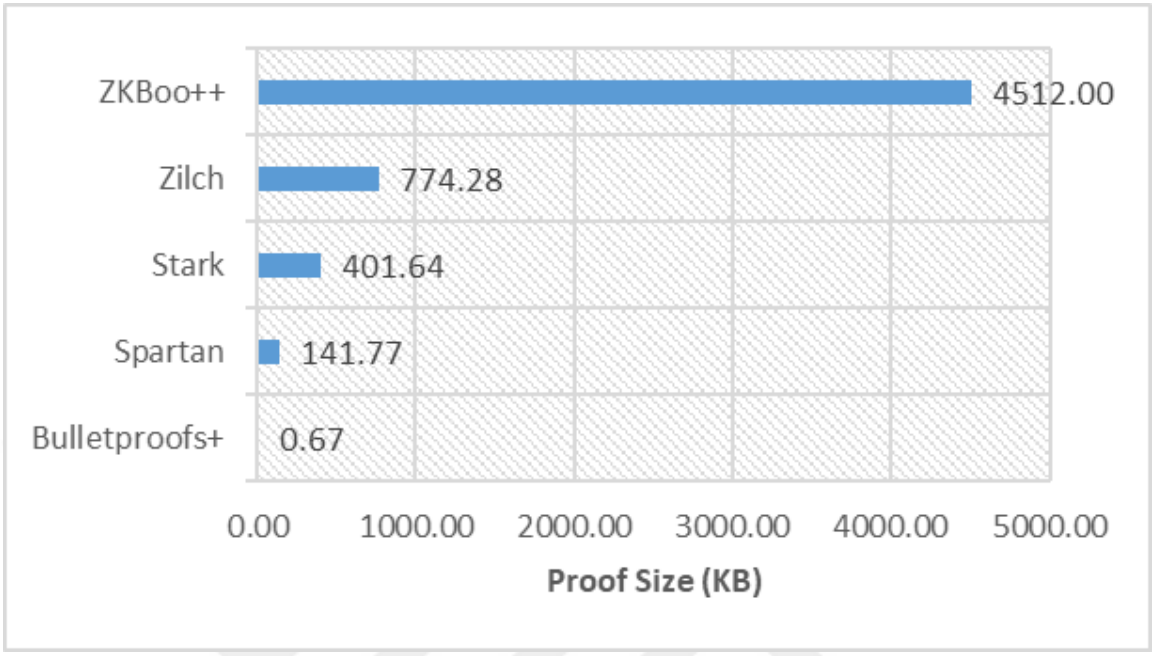


Figure 5.3. Overview of Proof Size on System 1

Figure 5.3 presents a comparative analysis of the proof sizes of the six algorithms examined in this study. A detailed evaluation of this figure highlights that the ZKBoo++ algorithm exhibits a significant negative deviation from the other algorithms in terms of proof size. This suggests that ZKBoo++ is more suitable for high-bandwidth network environments, closed-loop systems and applications without storage constraints. Additionally, the zk-STARK algorithms demonstrate relatively larger proof sizes compared to the other algorithms. A notable observation is that the variance seen in proof generation times is also reflected in proof size comparisons, reinforcing the idea that a clear standard has yet to be established for zk-STARK-based systems. Due to their structural characteristics, zk-SNARK algorithms are expected to have shorter proof sizes. When considering ZKBoo++ as an exception and taking the Marlin algorithm's proof size from the original study as a reference, it can be concluded that this expectation is largely met. Consequently, zk-SNARK algorithms can be widely applied in real-world applications, except for a few exceptional cases. Finally, it is evident that zk-STARK algorithms produce larger proof sizes compared to Bulletproofs+ and zk-SNARK algorithms. This characteristic is likely to negatively affect their adoption in scenarios where proof size efficiency is a critical factor.

Table 5.4. Results of the impact of using the AVX instruction set

Study	Year	ZKP System	AVX Status	Proving Time (ms)	Verify Time (ms)	Proof Size (KB)
Chung et al.	2022	Bulletproofs+	Enable	673.75	73.62	0.67
Chung et al.	2022	Bulletproofs+	Disable	1373.28	149.41	0.67

The data pertaining to the utilisation of the AVX instruction set for the Bulletproofs+ algorithm is delineated in Table 5.4. It is evident from the data that the utilisation of the AVX instruction set results in an approximate 2.03-fold enhancement in both the proof generation time and the proof verification time. The observation that a comparable enhancement is attained in both evaluation metrics indicates that the cryptographic framework of the algorithm is designed to utilise this instruction set. The values obtained in the scenario where the AVX instruction set is activated demonstrate that the Bulletproofs+ algorithm is among the top three algorithms in both proof generation and proof verification. This finding is further substantiated by the data presented in Tables 5.5 and 5.6.

Table 5.5. Results of the proving time at AVX instruction set enable

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proving Time (ms)
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	204.94
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	205.22
Chung et al.	2022	Bulletproofs+	Bulletproof	Transparent	no	673.75
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	893.48
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	30503.03
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	43760.00

Table 5.6. Results of the verifying time at AVX instruction set enable

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Verify Time (ms)
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	27.49
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	39.96
Chung et al.	2022	Bulletproofs+	Bulletproof	Transparent	no	73.62
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	92.01
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	104.94
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	5180.98

5.2. Results of System 2 Hardware Platform

As presented in Table 4.1, the System 2 hardware platform constitutes the second most powerful configuration among those evaluated. This platform serves as a basis for direct comparison with System 1, which is regarded as the reference system in this study. Given the

architectural similarities between the two systems—differing primarily in memory capacity and graphics processing unit—the comparative analysis aims to assess how such hardware variations influence performance outcomes. The results obtained on System 2 will be examined in relation to those of System 1, thereby facilitating an informed evaluation of hardware-dependent performance differentials.

As shown in Table 5.7, the proof generation times of the six evaluated algorithms on the System 2 hardware platform are listed in ascending order, from the shortest to the longest duration. Notably, only two of the six algorithms achieve proof generation times under one second. This result suggests a potential sensitivity of the algorithms to hardware characteristics such as memory capacity or GPU performance, particularly when contrasted with System 1, which shares identical CPU specifications but exhibits superior memory and graphics capabilities.

Table 5.7. Results of proving time of system 2 platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proving Time (ms)
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	207.86
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	220.27
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	1406.19
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	1975.53
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	31047.81
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	53454.00

The analysis of the data presented in the table indicates that zk-SNARK algorithms, with the exception of ZKBoo++, demonstrate suboptimal performance in terms of proof generation time. Conversely, Zilch from the zk-STARK algorithms demonstrates notable efficacy in this domain. It is observed that Bulletproofs and Stark algorithms in particular generate proofs in excess of one second; this time is likely to have a negative impact on the preference, especially in real-world applications where bandwidth is limited. While this outcome is unanticipated in view of its characteristics, it is noteworthy that the Bulletproofs+ algorithm exhibits superior performance in comparison to the Stark algorithm. It is also noteworthy that ZKBoo++ from the zk-SNARK group and Zilch from the zk-STARK group algorithms produce highly similar results and these algorithms can be considered secure even in the period following the quantum revolution, as evidenced by the improvements in their structures stated in the original studies. It is also evident that these algorithms exhibit a substantial and positive margin of superiority over competing algorithms. This observation underscores the prominence of these algorithms as preferred solutions for a substantial number of real-world applications. A thorough review of the original studies indicates that the test results obtained for the Spartan and Marlin algorithms are consistent with the existing literature.

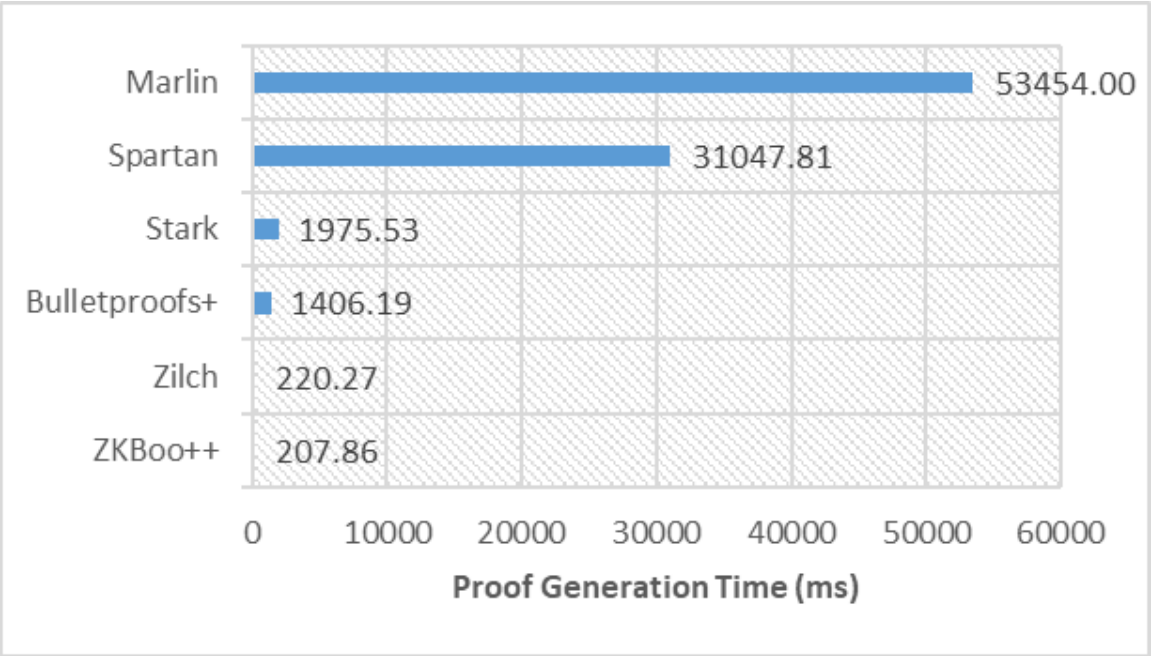


Figure 5.4. Overview of Proof Generation Performance on System 2

As illustrated in Figure 5.4, a comparative analysis of the proof generation times for the six algorithms is presented. The results indicate a marked divergence in performance, particularly between the Marlin and Spartan algorithms. Among these, Spartan consistently demonstrates a relatively faster proof generation time, highlighting its efficiency advantage within the evaluated set.As a result, it is seen that there is no significant change when compared to the performance values obtained by these algorithms on the System 1 hardware platform. This can be interpreted as these algorithms are more sensitive to CPU power and the effect of other system resources such as memory and video card power on the performance is very limited or not. In contrast, the Zilch and ZKBoo++ algorithms are considered more suitable for systems with limited system resources such as peer-to-peer applications and IoT applications due to their relatively short proof generation times. The superior performance of the Bulletproofs+ algorithm in comparison to the Stark algorithm on the System 1 hardware platform can be attributed to the beneficial impact of the video card or amount of memory used by Stark algorithm. The fact that the Bulletproofs+ algorithm obtains similar values to the performance values obtained on the System 1 platform can be interpreted as the performance of this algorithm depends mainly on CPU power and amount of memory available.

As demonstrated in Table 5.8, the proof verification times for the six algorithms evaluated on the System 2 hardware platform are listed in an ascending order of milliseconds from the shortest to the longest.

Table 5.8. Results of verification time of system 2 platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Verify Time (ms)
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	34.49
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	40.76
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	91.64
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	105.70
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	152.65
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	5267.69

It is seen that the values obtained with zk-STARK applications on the System 2 hardware platform show a significant superiority compared to other algorithms. It is observed that the two algorithms in the zk-SNARK group show comparable performance and are just behind the best performers. It is clear that four of the six algorithms show a proof verification time of approximately 100 milliseconds or less. This observation shows that most of the algorithms are quite suitable for deployment in real-world scenarios and have a satisfactory level of proof verification efficiency. In addition, it is seen that these values obtained are quite close to the value obtained on the System 1 hardware platform. This situation shows that the algorithms are heavily dependent on CPU power during the proof verification process. It is observed that the Bulletproofs+ algorithm is just behind the group of four mentioned above and shows a performance within acceptable limits. It is assumed that the Marlin algorithm produces a very high proof verification time similar to the proof generation process and therefore its use in real-world applications is considered to be quite limited. In addition to this data, the fact that all of the algorithms in the first four are considered safe for the post-quantum age also makes them promising for future use.

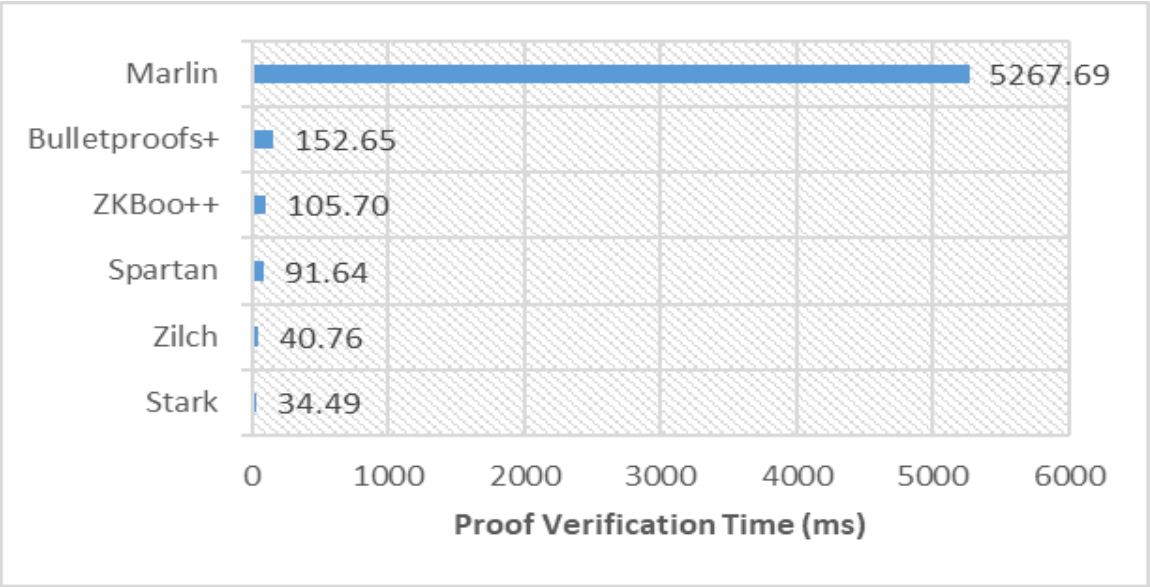


Figure 5.5. Overview of Proof Verification Performance on System 2

The proof verification times of the six algorithms in this study are presented in Figure 5.5, which is arranged for comparative purposes. A thorough examination of this figure reveals that the Marlin algorithm demonstrates a substantial negative deviation from the other algorithms. This finding indicates that the Marlin algorithm is not sufficiently useful for real-world applications. Conversely, the Zilch and Stark algorithms are regarded as more suitable for general and resource-constrained applications, as evidenced by their demonstration of the shortest proof verification times. The Bulletproofs+ algorithm is considered applicable to many real-world applications, given that its proof verification time is not significantly longer than that of the first four well-performing algorithms. It is observed that the performance values seen here are very close to those seen in Figure 5.2. This situation reveals that the CPU power is the most important factor that directly affects the performance in the proof verification time criterion for almost all algorithms.

The proof sizes obtained as a result of the proof generation process for each of the six algorithms tested on the System 2 hardware platform are presented in Table 5.9 and the values are listed in increasing order from smallest to largest according to kilobyte size.

Table 5.9. Results of proof size of system 2 platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proof Size (KB)
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	0.67
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	141.79
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	401.64
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	774.28
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	4512.00
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	n.a.

Upon examination of the table data illustrating the proof sizes, it becomes evident that the Bulletproofs+ algorithm is demonstrably superior. However, it should be noted that no data regarding the proof size of the Marlin algorithm could be obtained during the implementation. The original study reports that the proof size for the BLS12-381 elliptic curve is 880 bytes. This suggests that the Marlin algorithm performs well in terms of proof size, despite its relatively poor performance in terms of proof generation and verification times. In a similar vein, the ZKBoo++ algorithm, despite its strong performance in terms of proof generation and verification, is encumbered by excessively large proof sizes, a limitation that potentially curtails its practical applications despite its computational efficiency. Among the six algorithms analysed, Bulletproofs+ is seen to produce the smallest proof size, as expected. It is determined that the proof sizes are equivalent in System 1 and System 2 platforms and no discrepancy is observed. Consequently, it is deduced that the proof sizes are attributable to the cryptographic models employed by the algorithms and therefore no discernible change can be identified.

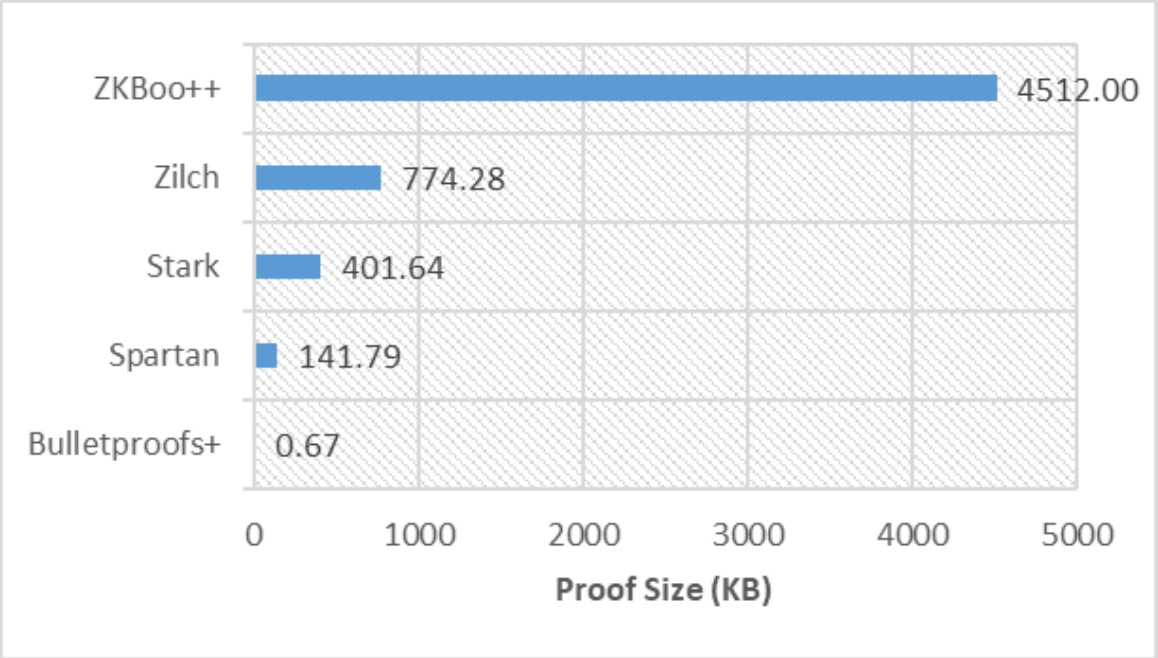


Figure 5.6. Overview of Proof Size on System 2

As illustrated in Figure 5.6, a comparative analysis of the proof sizes of the six algorithms examined in this study is presented. A detailed evaluation of this figure highlights that the ZKBoo++ algorithm exhibits a significant negative deviation from the other algorithms in terms of proof size. Consequently, it can be deduced that ZKBoo++ is more appropriate for applications that do not have storage constraints, such as high-bandwidth network environments, closed-loop systems or has huge storage resources. In addition, it is observed that the zk-STARK algorithms have relatively larger proof sizes compared to the other algorithms. A striking observation is that

the difference in proof generation times is also reflected in the proof size comparisons, reinforcing the idea that a clear standard has not yet been established for zk-STARK-based systems. Due to their structural characteristics, zk-SNARK algorithms are anticipated to exhibit reduced proof sizes. Considering ZKBoo++ as an exception and taking the proof size of the Marlin algorithm from the original study as a reference, it can be concluded that the Marlin and Spartan algorithms largely meet this expectation. Consequently, it is anticipated that zk-SNARK algorithms will be extensively utilised in real-world applications, with the exception of a few exceptional cases. Finally, it is evident that zk-STARK algorithms generate larger proof sizes in comparison to Bulletproofs+ and zk-SNARK algorithms. This attribute is hypothesised to impede the adoption of these algorithms in real-world scenarios where proof size efficiency is a pivotal consideration.

Table 5.10. Results of the impact of using the AVX instruction set

Study	Year	ZKP System	AVX Status	Proving Time (ms)	Verify Time (ms)	Proof Size (KB)
Chung et al.	2022	Bulletproofs+	Enabled	697.18	76.07	0.67
Chung et al.	2022	Bulletproofs+	Disabled	1406.19	152.65	0.67

The utilization of the AVX instruction set in the Bulletproofs+ algorithm is comprehensively analyzed in Table 5.10. The results indicate that incorporating the AVX instruction set leads to an approximately twofold improvement in both proof generation and verification times. A similar enhancement is observed across key evaluation metrics, suggesting that the cryptographic framework of the algorithm is well-optimized to exploit the computational acceleration provided by AVX. Furthermore, the data reveal that when the AVX instruction set is enabled, the Bulletproofs+ algorithm ranks among the top three in terms of both proof generation and verification efficiency. This observation is further corroborated by the results presented in Tables 5.11 and 5.12. Additionally, a comparative analysis between the performance data obtained from System 2 and System 1 indicates a marginal increase of approximately 2-3%. However, this difference is considered statistically insignificant and does not materially impact the overall performance assessment.

Table 5.11. Results of the proving time at AVX instruction set enable

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proving Time (ms)
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	207.86
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	220.27
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	697.18
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	1975.53
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	31047.81
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	53454.00

Table 5.12. Results of the verifying time at AVX instruction set enable

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Verify Time (ms)
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	34.49
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	40.76
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	76.07
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	91.64
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	105.70
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	5267.69

5.3. Results of System 3 Hardware Platform

As detailed in Table 4.1, the System 3 hardware platform—excluding the portable configuration—represents the least powerful configuration among the test systems. Its architectural similarity to the System 2 platform, differing primarily in memory capacity and CPU performance, provides a controlled basis for comparative evaluation. This section aims to assess how these hardware limitations influence performance outcomes by comparing results against those of System 1, which serves as the performance benchmark throughout this study.

As demonstrated in Table 5.13, the proof generation times of the six algorithms tested on the System 3 hardware platform are presented in increasing order from the shortest to the longest. It is evident that merely two of the six algorithms demonstrate a proof generation time of less than one second, indicating that only two of the algorithms can be considered practically usable. This observation can be interpreted as the amount of memory and CPU power directly affecting the algorithm performance when the algorithms are compared with the System 2 platform, which has a similar structure. It was observed that the compiled version of the Spartan algorithm on this platform did not run with an invalid instruction function error. Consequently, the algorithm source code was recompiled on the System 3 platform and subsequently executed once more. Notwithstanding, the compiled code did not run with a runtime error. This observation suggests that the algorithm incorporates optimizations that are contingent on the specific instruction sets supported by certain CPUs.

Table 5.13. Results of proving time of system 3 platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proving Time (ms)
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	325.03
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	360.85
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	1720.43
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	5605.44
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	121890.00
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	Could not run

The analysis of the data presented in the table indicates that, with the exception of ZKBoo++, zk-SNARK algorithms demonstrate unsatisfactory performance with regard to proof generation time. It is evident that the Spartan algorithm is unfeasible, despite all attempts to execute it. This outcome underscores the reliance of these algorithms on hardware-based acceleration technologies to attain satisfactory performance, despite their CPU-based nature. In contrast, Zilch from the zk-STARK algorithms has been found to demonstrate notable efficiency in this domain. It is evident that there are substantial discrepancies between zk-STARK algorithms. Despite their apparent similarities, significant disparities in proof generation times can be attributed to the recent development and lack of standardisation of these algorithms. The analysis of the obtained data indicates that Zilch and ZKBoo++ algorithms are the only ones that can be practically used for real-world applications. It has been observed that Bulletproofs and Stark algorithms in particular generate proofs in excess of one second; this time is likely to have a negative impact on the choice, especially in real-world applications where bandwidth is limited. This outcome is notable given the features of the Bulletproofs+ algorithm, which exhibits superior performance in comparison to the Stark algorithm. This can be interpreted as the Bulletproofs+ algorithm having a short proof size and the Stark algorithm having a relatively long proof size, thus the long proof size directly affects the proof production process in systems with relatively limited system resources. It is further hypothesised that the ZKBoo++ algorithm from the zk-SNARK group and the Zilch algorithm from the zk-STARK group yield analogous outcomes and can be regarded as secure even in the period following the quantum revolution, as evidenced by the enhancements in the structures of these algorithms specified in the original studies.

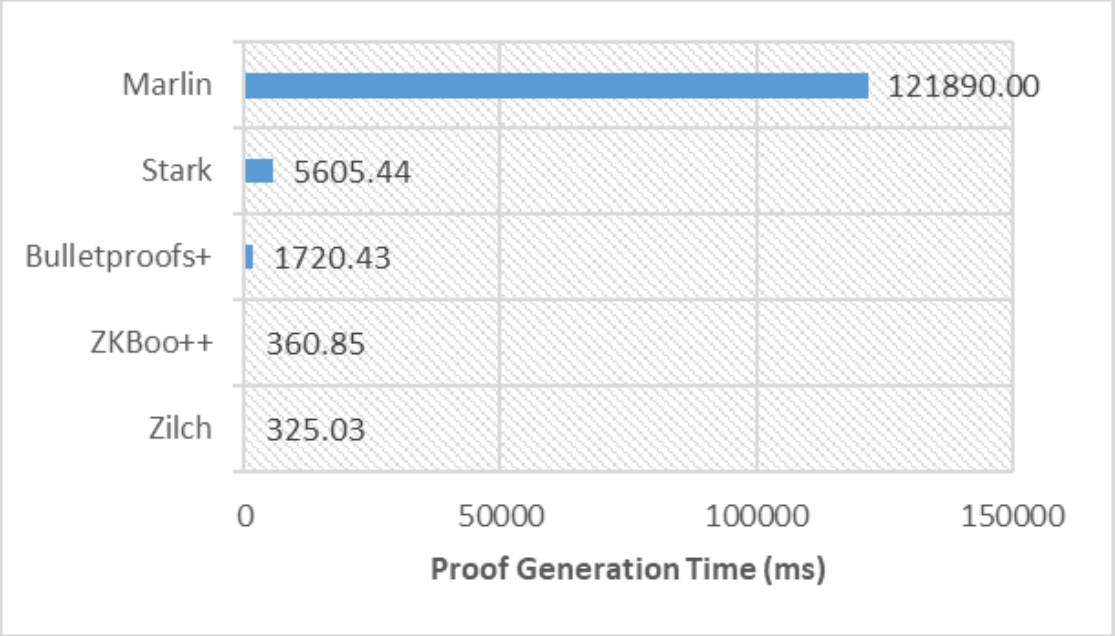


Figure 5.7. Overview of Proof Generation Performance on System 3

There is the comparative analysis of proof generation times for six algorithms in Figure 5.7, wherein it is revealed that there is quite a negative gap when it comes to the proof generation times of the algorithms namely Marlin and Spartan from the view of the comparative analysis. The Spartan is unfit for use on the ground that it cannot work while the result arrived at using the Marlin algorithm is virtually nonusable. Thus, what is evident is that these algorithms considerably take longer in proof generation, which is nearly double the performance values achieved on System 2 hardware platforms. This can be interpreted as indicating that these algorithms are far too sensitive to the CPU power, implying in turn that other resources of the system memory and video card power do little or nothing to contribute to the overall performance. In contrast, the Zilch and ZKBoo++ algorithms can be considered valid even on systems with scarcity of resources because they show rather quick proof generation time. The higher performance of the Bulletproofs+ algorithm compared with the Stark algorithm on the System 2 hardware platform could be attributed to the amount of memory used by the Stark algorithm or the positive effect of the video card. The main disadvantage in the performance of the Stark algorithm is because of the heavy dependency on the acceleration functions provided by modern hardware, resulting in below-average performance in platforms with lesser system resources. On the other hand, the metrics for the Bulletproofs+ algorithm almost match those on the System 2 platform, indicating that the performance of this algorithm is mostly dependent on processing power.

Table 5.14 illustrates the proof verification times of the six algorithms tested on the System 3 hardware platform, namely ordered with respect to time in milliseconds from the shortest time to the longest time.

Table 5.14. Results of verification time of system 3 platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Verify Time (ms)
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	36.31
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	54.34
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	189.32
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	189.80
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	8604.45
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	Could not run

The evidence for the superiority of zk-STARK values obtained through applications running on the hardware platform of System 3 over any other algorithms has been seen. It is understood that only one of these algorithms from the entire zk-SNARK group provides enough performance and it comes in at number four out of six. This only shows that there exist only two out of these six algorithms that return a proof verification time of about 50 milliseconds or lesser. This can simply be interpreted such that most of the algorithms would be good enough for

deployment and have performed strongly in terms of the proof verification efficiency in the real world, but they all have a performance curve directly proportional to the power of the current hardware platforms. One can also see that the values returned with the zk-STARK algorithms are, by and large, closing in on the one returned on the System 2 hardware platform; on the other hand, the performance of the Bulletproofs+ and ZKBoo++ algorithms is poor as well, with the former taking up to twice as much computation work as that of ZKBoo++. This phenomenon can be attributed to the heavy reliance of the algorithms on CPU power during proof verification and not on the much better performing zk-STARK algorithms, which were optimized to yield their best even under less system resources. The collected observations indicate that although Bulletproofs+ and ZKBoo++ demonstrate satisfactory performance, they still fall significantly short compared to zk-STARKs, which exhibit comparable metrics. It is thought that high proof verification time can be generated by the Marlin algorithm as well as by proof generation process, so using it in practice applications can be regarded as very limited. Spartan cannot be utilized because it cannot execute.

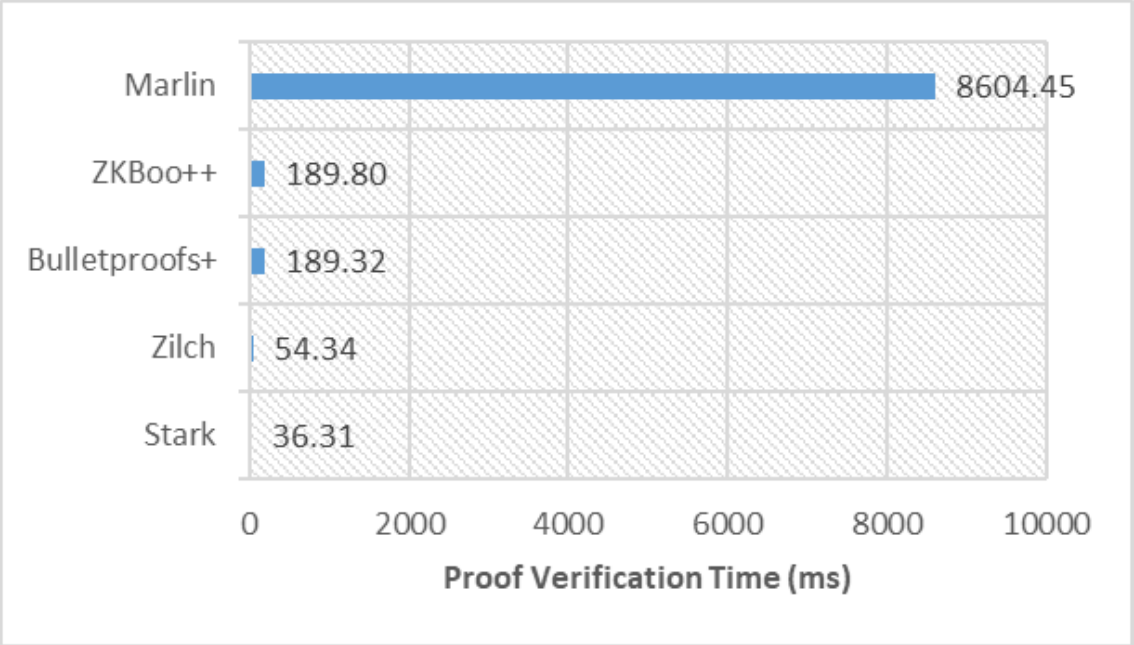


Figure 5.8. Overview of Proof Verification Performance on System 3

The proof verification times of the six algorithms depicted in this study Figure 5.8 have been arranged for the sake of comparison. A detailed study of this figure shows a significant negative deviation, thus asserting that the Marlin algorithm is not particularly useful for real-life scenarios. Since the Spartan implementation has an error, it cannot be executed. Zilch and Stark, however, are considered really good, perhaps even for general and resource-limited applications- thanks to their prompt proof verification times. Bulletproofs+ is in the third position in terms of proof verification time and is generally applicable to most real-world applications. A similar observation can also be made about the ZKBoo++ algorithm. The performance values observed

here are somewhat close to the values shown in Figure 5.6. This finding is crucial, as it emphasizes that CPU prowess determines performance regarding proof verification time for various algorithms.

The proof sizes obtained due to the proof generation process from each of six algorithms tested on the hardware platform System 3 have been included in Table 5.15 and sorted increasingly according to kilobytes.

Table 5.15. Results of proof size of system 3 platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proof Size (KB)
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	0.67
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	269.50
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	774.28
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	4512.00
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	Could not run
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	n.a.

A close analysis of the proof sizes represented in the tables shows conclusively that the Bulletproofs+ algorithm is favored over others. However, it is vital to point out that no information concerning the proof size for the Marlin was collected during the time of running tests. The initial paper mentioned the proof size for the BLS12-381 elliptic curve as 880 bytes. This finding shows that the Marlin performs favorably in proof sizes, whereas it underperforms in the proof generation and verification periods. Conversely, the ZKBoo++ algorithm offers great proof generation and verification performance, but it suffers from a sizable proof size that greatly constrains its utility in practice. Of the six algorithms evaluated, the Bulletproofs+ is expected to offer the least proof sizes. Proof sizes on the setups, System 2 and System 3 are seen to be comparable, except for Stark in which there is no difference noted. This leads to the conclusion that the proof sizes can be traced back to the cryptographic models that the algorithms implement; hence, no deployable difference is observed. It is, however, argued that the variation in the Stark algorithm may have arisen from the CPU that favored using smaller cryptographic starting values within proof production due to instructions not supported by the CPU. This could have implications with respect to the size of the proof.

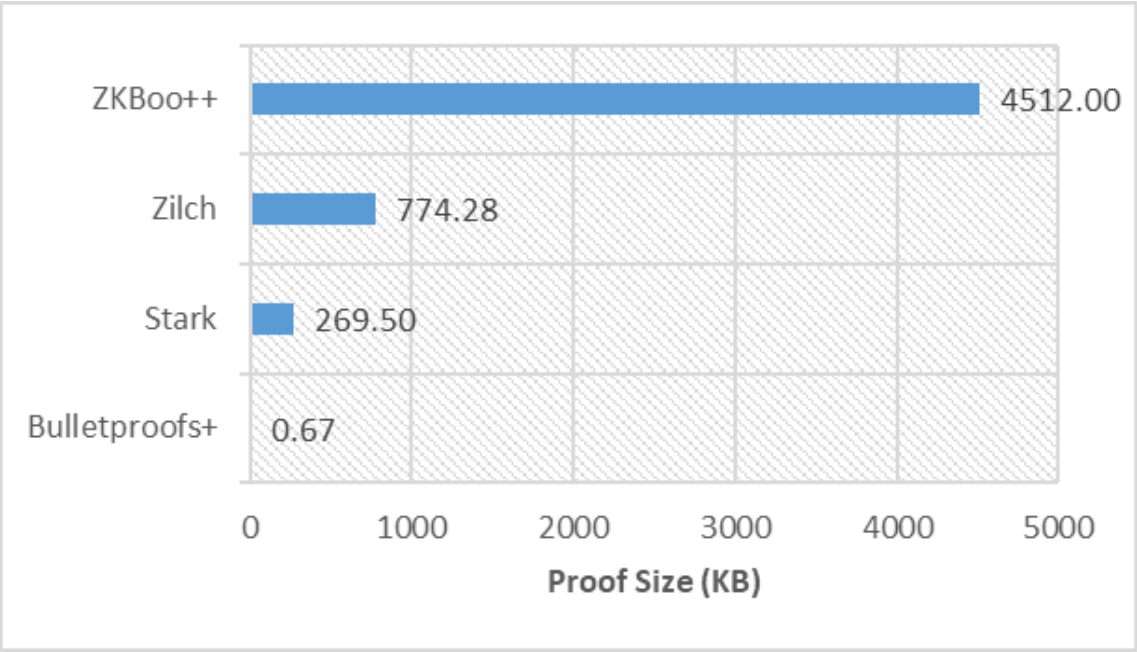


Figure 5.9. Overview of Proof Size on System 3

Figure 5.9 treats the comparative analysis of the proof sizes for the six algorithms studied. The in-depth assessment indicates that the proof size of the ZKBoo++ algorithm, by a considerable margin, lies on the negative side compared to the rest of the algorithms. Therefore, the conclusion can be drawn that ZKBoo++ works particularly well in fast networks or applications that can support high storage demand. Our findings also reveal that the proof sizes associated with zk-STARK algorithms are generally larger compared to those produced by the other schemes. Notably, the behavior of proof size under the zk-STARK algorithm exhibits a markedly distinct pattern on the System 2 platform, diverging from the trends observed in other systems. This reinforces the statement that no standard uniformity has been put in place for zk-STARK-promoting systems and that the algorithms enjoy optimization targeted at CPUs. Given their design, it is expected that zk-SNARKs will attain a lower proof size. ZKBoo++ is the exception and Marlin's proof size from the original study is the benchmark. The performance of Marlin falls within this particular expectation. This therefore implies that, barring a couple of instances, the zk-SNARK will find itself in excessive use in the real world. In the end, this analysis affirms that the zk-STARK algorithms yield bigger proof sizes compared to Bulletproofs+ and the zk-SNARK algorithms. It is foresight that this particular feature will stifle acceptance of these algorithms in practical, real-world space, where proof size efficiency is a crucial consideration.

Table 5.16. Results of the impact of using the AVX instruction set

Study	Year	ZKP System	AVX Status	Proving Time (ms)	Verify Time (ms)	Proof Size (KB)
Chung et al.	2022	Bulletproofs+	Enabled	1139.83	124.86	0.67
Chung et al.	2022	Bulletproofs+	Disabled	1720.43	189.32	0.67

An exhaustive analysis of the utilization of the AVX instruction set within the Bulletproofs+ algorithm is provided in fine detail in Table 5.16. The results demonstrate that when to the AVX instruction set was included, both proof generation and proof verification times improved roughly 1.5 times. On further examination, it is clear that both proof generation and proof verification times were increased by 34%, although that improvement is less pronounced than the 50% improvement afforded by the System 2 platform. This therefore implies that CPU technology advance that ties with enhanced performance is beneficial to newer CPUs for further improvements in speed. The data tends to indicate that the Bulletproofs+ algorithm performs among the top three from proof generation up to proof verification when the AVX instruction set is activated. This observation is further confirmed by the results presented in Tables 5.17 and 5.18. A comparative analysis of the performance traces gathered from Systems 3 and 2 reflects a marginal gain of approximately 15%. This variance is considered statistically significant, thus strengthening the conclusion on the improved efficiency evidence in the overall performance of contemporary CPU architectures.

Table 5.17. Results of the proving time at AVX instruction set enable

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proving Time (ms)
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	325.03
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	360.85
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	1139.83
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	5605.44
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	121890.00
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	Could not run

Table 5.18. Results of the verifying time at AVX instruction set enable

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Verify Time (ms)
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	36.31
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	54.34
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	124.86
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	189.80
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	8604.45
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	Could not run

5.4. Results of Portable System Hardware Platform

The portable testbed is in fact the least powerful among all testbeds (refer Table 4.1). This will hence facilitate a back-to-back comparison with the System 3 hardware platform, being the next similar specifications. The discussion here focuses on how the results would affect

performance, since the portable hardware platform is structurally similar to the System 3 hardware platform but with varying CPU power. Thereafter, comparisons will be made with results obtained from the System 1 hardware platform, which forms a basis for benchmarking.

The proof generation times of the six algorithms tested on the portable hardware platform are arranged in Table 5.19 in increasing order from the shortest to the longest. It can thus be seen that merely two of the six algorithms show a proof generation time of less than one second, meaning that only two of all the algorithms can be considered practically usable. One interpretation of this finding is that the CPU power directly affects algorithm performance when comparing with the System 3 platform with a similar structure. In fact, it is interesting to note that the first two algorithms have poor performance relative to values obtained on the System 3 platform, while the second two have better performance. And the similar observation happened when the Spartan algorithm was compiled and executed under the System 3 hardware platform and, at that point, gave an 'invalid instruction function' error. Hence, the algorithm source codes were recompailed on the portable platform and executed again, but not with the desired result; that is, it did not run with a runtime error, indicating the presence of optimizations in the algorithm dependent on certain instruction sets supported by certain CPUs. It is observed that the running times obtained through the Marlin algorithm were poorer than those of the System 3 platform and this was likewise true for the running times of the first two algorithms.

Table 5.19. Results of proving time of portable platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proving Time (ms)
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	368.00
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	404.87
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	1282.30
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	3866.86
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	186333.00
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	Could not run

Examining data thoroughly reveals that, except for ZKBoo++, zk-SNARK algorithms display poor performance in the measurement of the time taken to generate proof. It is almost obvious that the Spartan algorithm could not be used since it was not achieved despite many attempts to run it. This result gives emphasis on the fact that these algorithms are hardware-based accelerator dependent for attaining reasonable performance, even when they are CPU-based. Zilch on the other hand from the zk-STARK category has shown incredible efficiency in this area. At this point, one can conclude that substantive differences exist among zk-STARK algorithms. Differences in proof generation times, despite obvious similarities among them, are probably due to their recent development and lack of standardization. The analysis of the data acquired suggested

that Zilch and ZKBoo++ are among the very few such algorithms that can really be used practically in real applications. In specific Bulletproofs and Stark algorithms were noted to make proofs in more than one second duration, which in all probability would deter their selection, especially in real-life scenarios where the bandwidth is low. This is all the more interesting considering that Bulletproofs+ is proven to outperform the Stark algorithm. This may imply that Bulletproofs+ carries a smaller proof size in contrast to the Stark algorithm which attains a relatively larger proof size; hence, directly the longer proof size will affect the processing of proof generation in systems which are relatively hardware poor. Moreover, it has been demonstrated that the values obtained through Stark and Bulletproofs+ algorithms exceed those obtained through the System 3 platform. It's interpretable that the algorithms listed are optimized for achieving higher performance values especially with the newer generations of CPUs while other algorithms are tuned for IPC performance merely due to the CPU clock frequency. It is hypothesized further that the ZKBoo++ algorithm from the zk-SNARK group and the Zilch algorithm from the zk-STARK group yield analogous outcomes and can be regarded as secure even in the period following the quantum revolution based on advancements in the algorithms' structures, as described in the original studies.

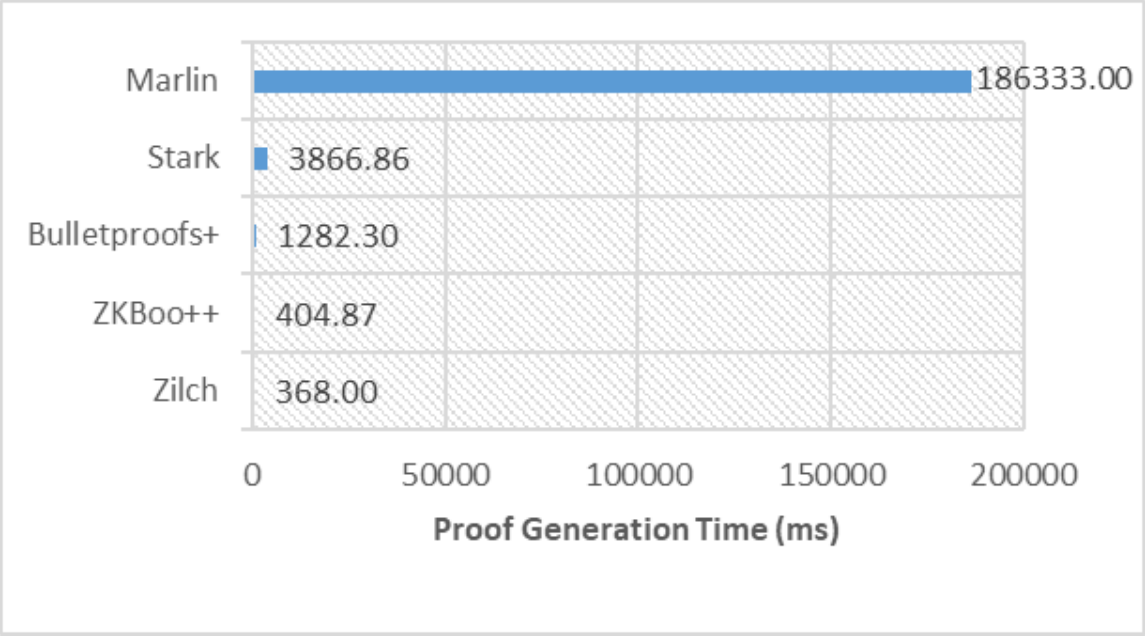


Figure 5.10. Overview of Proof Generation Performance on Portable

Comparative timing for proof generation in six algorithms is seen in figure 5.10. It shows a considerable negative difference between Marlin and Spartan algorithms and all others. Also, the Spartan algorithm failed to generate any proof. The level obtained by using the Marlin algorithm is considered improper and thus unusable. The above analysis shows that it takes these algorithms enormous time to produce proofs, which includes some 50% more processing time with the Marlin

algorithm when compared to its performance values on System 3 hardware platforms. That could be seen as these algorithms being very sensitive to CPU power and with little if anything of other resources, like system memory and video card power, into the performance. In comparison, Zilch and ZKBoo are valid even under very limited resources affording very speedy proof times. Another interesting observation is that the Bulletproofs+ algorithm performs better than the Stark algorithm. The interpretation of this is that proof generation takes longer on systems with relatively few resources due to their large proof sizes. But the outstanding performance of Stark and Bulletproofs+ algorithms with portable hardware can be explained by the availability of a newer generation processor in comparison to that of the System 3 hardaware platform. One serious drawback of the Stark algorithm lies in its overwhelming dependence on acceleration functions offered by current hardware, thus accounting for less than average performance on systems having lower resources.

As illustrated in Table 5.20, the proof verification times of the six algorithms tested on the Portable hardware platform are presented in milliseconds, from the shortest to the longest.

Table 5.20. Results of verification time of portable platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Verify Time (ms)
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	49.58
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	61.57
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	137.89
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	210.52
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	10139.17
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	Could not run

The results derived from the experiments conducted on the portable hardware platform suggest that zk-STARK-based implementations consistently outperform other algorithms in terms of performance metrics. However, a comprehensive evaluation of the zk-SNARK family reveals that only one algorithm within this group achieves a competitive performance level, ranking fourth overall among the six algorithms examined. When all the algorithms are evaluated, it is evident that only three of these six algorithms return a proof verification time of approximately 130 milliseconds or less. This finding suggests that the majority of the algorithms are suitable for deployment and demonstrate robust performance in terms of proof verification efficiency in practical scenarios. However, it is noteworthy that all of these algorithms exhibit a performance curve that is directly proportional to the computing power of the current hardware platforms. It is also evident that the values returned by the zk-STARK algorithms are closely aligned with those returned on the System 3 hardware platform and that they exhibit suboptimal performance in comparison to System 3, in contrast to the proof generation times. Conversely, the performance of

the Bulletproofs+ and ZKBoo++ algorithms is also weak, with ZKBoo++ requiring approximately twice the computational work of the Bulletproofs+ algorithm. This finding indicates that the algorithms are highly dependent on CPU power during proof verification, thereby underscoring the direct impact of IPC performance on verification performance. While Bulletproofs+ and ZKBoo++ demonstrate satisfactory performance, they exhibit significantly lower performance compared to zk-STARKs, which have approximately half the computational requirements. It can be deduced that the Marlin algorithm is improbable to be utilised in practical applications, due to its notably elevated proof verification time, in addition to the exceedingly elevated values in the proof generation process. The Spartan algorithm is not a viable option for implementation due to its incompatibility.

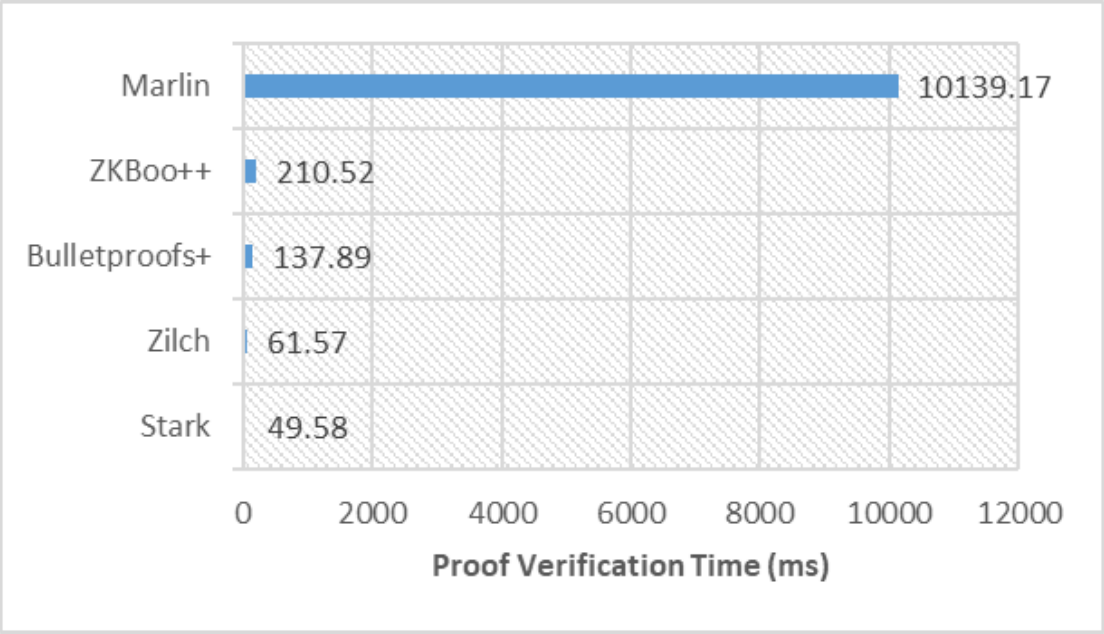


Figure 5.11. Overview of Proof Verification Performance on Portable

The proof verification times of the six algorithms depicted in Figure 5.11, have been arranged for the purpose of comparison. A thorough examination of this figure reveals a substantial negative deviation, suggesting that the Marlin algorithm is not particularly effective in real-world scenarios. The Spartan implementation is not viable, thus rendering it unusable. Conversely, Zilch and Stark are regarded as highly effective, even for general and resource-constrained applications, due to their rapid proof verification times. Bulletproofs+ is ranked third in terms of proof verification time and is generally applicable to most real-world applications. A similar observation can be made about the ZKBoo++ algorithm. The performance values observed here are relatively close to the values obtained on the System 3 platform. This finding is significant because it highlights that the CPU capability determines the performance regarding proof verification time for almost all algorithms.

The proof sizes resulting from the proof generation conducted through each of the six algorithms were tested on the portable hardware platform and are sorted in increasing order in kilobytes and presented in Table 5.21.

Table 5.21. Results of proof size of portable platform

Study	Year	ZKP System	Protocol	Setup Type	Post-Quantum Secure	Proof Size (KB)
Chung et al.	2022	Bulletproofs+	sh-SNARK	Transparent	no	0.67
Ben-Sasson et al.	2018	Stark	zk-STARK	Transparent	yes	401.64
Mouris and Tsoutsos	2021	Zilch	zk-STARK	Transparent	yes	774.28
Chase et al.	2017	ZKBoo++	zk-SNARK	Trusted	yes	4512.00
Setty S.	2020	Spartan	zk-SNARK	Transparent	yes	Could not run
Chiesa et al.	2020	Marlin	zk-SNARK	Universal	no	n.a.

A thorough examination of the proof sizes in the data presented in Table 5.21 unequivocally demonstrates the superiority of the Bulletproofs+ algorithm over its competitors. However, it is important to note that no information on the proof size could be collected for Marlin during the time the tests were run. The original paper for Marlin stated the proof size as 880 bytes for the BLS12-381 elliptic curve. This finding indicates that Marlin performs well in terms of proof sizes, but exhibits poor performance in terms of proof generation and verification periods. In contrast, the ZKBoo++ algorithm offers excellent proof generation and verification performance, but suffers from a significant proof size that greatly limits its use in practice. Of the six algorithms evaluated, Bulletproofs+ is expected to provide the least proof size for the reasons explained in the original paper. With the exception of the deviation in proof size obtained by the Stark algorithm for the System 3 platform, it is evident that the proof sizes for all other algorithms are similar in all setups including all platforms. This observation suggests a direct correlation between proof size and the cryptographic models implemented by the algorithms, thereby indicating that no statistical difference was observed. However, it is hypothesised that the variation observed in the Stark algorithm may be attributed to the CPU's inability to support the instructions, which favours the utilisation of smaller cryptographic initial values in the algorithm employed for proof generation.

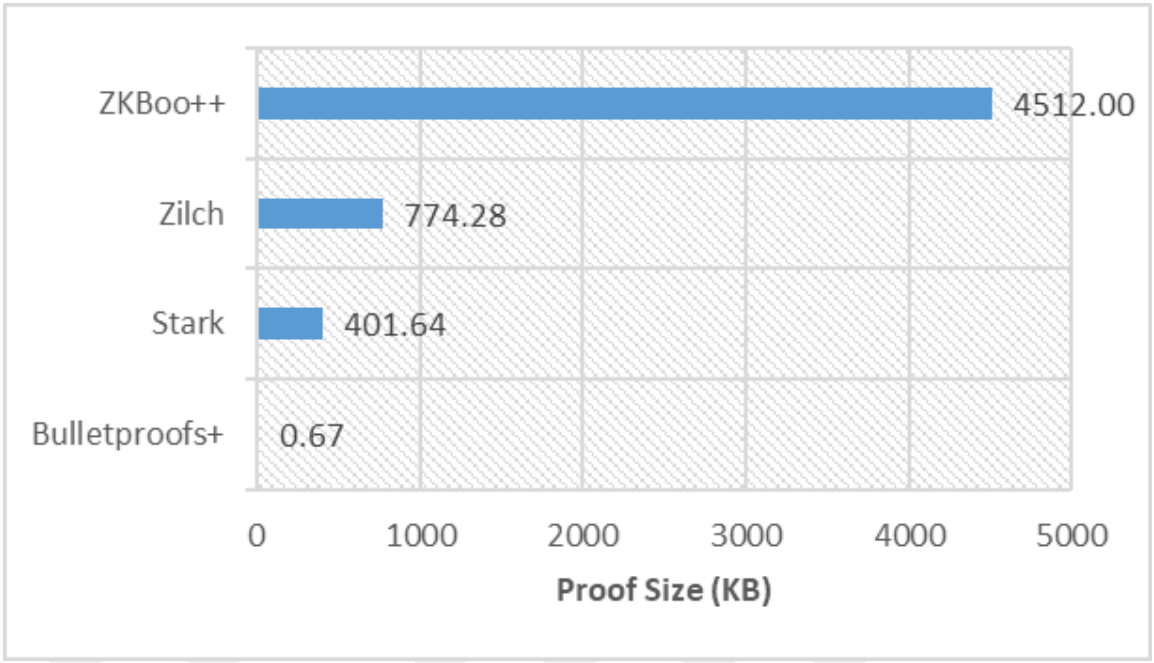


Figure 5.12. Overview of Proof Size on Portable

As indicated in Figure 5.12, proof sizes of algorithms 1-6 were compared. The analysis has shown that the proof size in the analysis of ZKBoo++ is unproportionally large compared to the other algorithms, leading one to conclude that ZKBoo++ is most suited for fast networks or applications with abundant storage resources. It is evident that the proof sizes for zk-STARK algorithms are smaller than those of zk-SNARKs, although larger than those of the others. By design, zk-SNARKs are anticipated to be lower in proof size. ZKBoo++ is one exception and a reasonable reference point for the proof size of Marlin from the original paper. Indeed, Marlin's score is on point concerning this metric. Hence, it appears zk-SNARKs will generally find their use in practice barring some exceptions. To summarize, this platform's evaluation has evidently shown that zk-STARK algorithms incur more extensive proof sizes compared to Bulletproofs and zk-SNARKs, excluding the ZKBoo++ algorithm. This particular trait is anticipated to result in more difficulty in applicability for these algorithms in the real world, where proof size efficiency is a crucial consideration.

The study, which was conducted on the use of the AVX instruction set in the Bulletproofs+ algorithm on the portable platform, revealed no significant differences in the results. The data sheets published by Intel demonstrate that the microprocessor powering the aforementioned platform supports the AVX2 instruction set. However, the tests conducted did not reveal any changes. This observation can be interpreted as indicating that the virtualization subsystem, which was relocated and activated from the reference platform to this particular platform, was not fully compatible with the portable system, thereby restricting certain functions. Nevertheless, a

comparison of the obtained values with those of the system 3 platform does not appear to indicate any inconsistency.

5.5. General Comparison

The independent results of the tests conducted with four different hardware platforms are discussed in the above sections on a platform-by-platform basis. In this section, the objective is twofold: firstly, to compare and analyse the proof generation and verification times obtained for each algorithm on four platforms; and secondly, to compare and analyse the results obtained from the four platforms with those previously mentioned. Subsequently, the values obtained with the System 1 hardware platform will be accepted as the baseline, i.e. the value that corresponds to 100% performance. The percentage of the other hardware platforms compared to the baseline will then be evaluated and analysed.

Table 5.22. Results of proving time of hardware platforms

		Proving Time (ms)			
Study	**ZKP System**	**System 1**	**System 2**	**System 3**	**Portable**
Chase et al.	ZKBoo++	205.22	207.86	360.85	404.87
Setty S.	Spartan	30503.03	31047.81	n.a.	n.a.
Chiesa et al.	Marlin	43763.00	53454.00	121890.00	186333.00
Chung et al.	Bulletproofs+	673.75	697.18	1139.83	1282.30
Mouris and Tsoutsos	Zilch	204.94	220.27	325.03	368.00
Ben-Sasson et al.	Stark	893.48	1975.53	5,605.44	3866.86

As shown in Table 5.22, the obtained values are the results of the proof generation times of the six algorithms examined in this study on four different hardware platforms. The first three algorithms numbered in the table belong to the zk-SNARK group, which is characterized by has trusted, universal and transparent type setup deployments. The proof generation times produced by these algorithms are listed in each row from fastest to slowest on different hardware platforms. The algorithm in the fourth row represents the Bulletproofs algorithm and each platform is listed from fastest to slowest. The last two rows in the table correspond to the zk-STARK group and the values are once again listed from fastest to slowest. A closer look at the data in this table reveals a common feature: the values obtained with the strongest hardware platform, the System 1 platform, increase towards the weakest platform, the portable platform, i.e., the performance decreases. This phenomenon can be interpreted as the algorithms run time performance is directly affected by the proportional to the system resources for all kinds of algorithms.

Table 5.23. Percentage results of proving time of hardware platforms

		Proving Time (%)			
Study	**ZKP System**	**System 1**	**System 2**	**System 3**	**Portable**
Chase et al.	ZKBoo++	100.00	98.73	56.87	50.69
Setty S.	Spartan	100.00	98.25	n.a.	n.a.
Chiesa et al.	Marlin	100.00	81.87	35.90	23.49
Chung et al.	Bulletproofs+	100.00	96.64	59.11	52.54
Mouris and Tsoutsos	Zilch	100.00	93.04	63.05	55.69
Ben-Sasson et al.	Stark	100.00	45.23	15.94	23.11

As presented in Table 5.23, the System 1 hardware platform serves as the reference baseline, with its performance values standardised to 100%. The results obtained from the remaining platforms are expressed as percentages relative to this baseline. When comparing the percentage-based results of the algorithms executed on the System 1 and System 2 platforms, it is observed that the performance variation is minimal for the ZKBoo++ and Spartan algorithms. The variation was found to be approximately 1% for ZKBoo++ and Spartan, approximately 3% for Bulletproofs and approximately 7% for Zilch. This finding indicates that there is negligible variation (1-3%) between platforms with equivalent CPU power. This finding indicates that the algorithms in question demonstrate a high degree of proportionality in their performance relative to CPU power. In addition, a significant disparity emerges between the System 1 and System 2 platforms with respect to RAM memory and GPU power. Notwithstanding this, no substantial discrepancy was observed except the values generated by the Stark and Marlin algorithms. This finding suggests that the ZKBoo++, Spartan, Bulletproofs and Zilch algorithms are not significantly impacted by variations in memory and GPU power, indicating a limited dependence on these resources. The 7% performance deficit observed in the Zilch algorithm can be interpreted as the fundamental performance benchmark of this algorithm being predominantly proportional to the CPU and to a lesser extent, to the available memory. Despite extensive research, no evidence has been found that the algorithms examined to date utilise GPU power. This assertion is based on the negligible loss in performance values generated by the algorithms, despite the substantial disparity in GPU power between the System 1 and System 2 platforms. The 54% performance loss observed in the Stark algorithm can be interpreted as follows: The Stark algorithm exhibits a performance that is directly proportional not only to the CPU power but also to the amount of available memory in the system. However, no definitive conclusion has been reached regarding the utilisation of GPU power by this algorithm. This is due to the significant disparity in GPU power between System 1 and System 2 platforms, yet the absence of a comparable difference between System 2 and System 3 platforms. However, a notable discrepancy in the values obtained by the algorithm is evident between System 2 and System 3 platforms. The dramatic performance drops

observed in Zilch and Stark also support this hypothesis. The lower resources available on the System 3 platform in terms of CPU power and memory when compared to the System 2 platform are reflected in the performance values of the Stark algorithm. Conversely, given the relatively minor disparity in GPU power, it is hypothesised that the observed significant performance variation is not attributable to GPU, but rather to RAM and CPU.

Table 5.24 has been obtained by ranking the proof generation times of the six algorithms compared in our thesis study from the smallest to the largest on each system basis. The objective of this comparison is to ascertain the most efficient algorithm on each system basis. Furthermore, for each algorithm, the mean of the values obtained for the four platforms on which the algorithm operates is determined and the positive or negative deviation of each hardware platform from the mean is evaluated.

Table 5.24. Average results of proving time of hardware platforms

		Proving Time				
Study	**ZKP System**	**System 1**	**System 2**	**System 3**	**Portable**	**Average**
Chase et al.	Zilch	204.94	220.27	325.03	368.00	279.56
Setty S.	ZKBoo++	205.22	207.86	360.85	404.87	294.70
Chiesa et al.	Bulletproofs+	673.75	697.18	1139.83	1282.30	948.26
Chung et al.	Stark	893.48	1975.53	5605.44	3866.86	3085.33
Mouris and Tsoutsos	Spartan	30503.03	31047.81	n.a.	n.a.	30775.42
Ben-Sasson et al.	Marlin	43763.00	53454.00	121890.00	186333.00	101360.00

An examination of Table 5.24 reveals that, with the exception of the value obtained by the ZKBoo++ algorithm on the System 2 platform, the Zilch algorithm from the zk-STARK group obtained the optimal proof production time on all platforms. The observed discrepancy of approximately 5% between these values suggests the possibility of analogous outcomes from both algorithms. It is determined that the situation created by the ranking from the most efficient (shortest proof time) to the least efficient for the System 1 platform is valid for the other three platforms. This finding serves to corroborate the analysis previously conducted for the data presented in Table 5.23. This finding indicates that the performance of all algorithms is directly proportional to their pure computational power. A comparison of the calculated average with the values obtained in the systems reveals that the System 1 and System 2 platforms are positively separated as a cluster, while System 3 and the Portable platform are negatively separated as a cluster. A further examination of the general ranking of the algorithms reveals that there is an acceptable level of difference between the first four algorithms, with all offering a proof production time of less than 1 second. It has been determined that the Spartan algorithm, which has a transparent setup and a zk-SNARK group and the Marlin algorithm, which has a universal setup, are far behind and cannot produce a practically usable proof production time even on the best

platform. The negligible difference between the ZKBoo++ algorithm, which is included in the group of zk-SNARK algorithms preferred due to their short proof sizes in real-world applications and the Zilch algorithm, which is included in the zk-STARK group, especially on the System 1 platform, is seen as promising in terms of the practical use of zk-STARK algorithms. When evaluating the security ranking of the algorithms, it is notable that three of the first four algorithms are considered secure for the post-quantum era. This suggests that almost all NIZKP algorithms can be used in practical applications in the future without any concerns. Finally, an evaluation of the deviations observed in the ZKBoo++ and Stark algorithms is warranted: It is concluded that the value obtained for the ZKBoo++ algorithm is within the measurement error limits and is not statistically significant. The negative deviation observed in the comparison of the Portable and System 3 platforms in the Stark algorithm is not evident in the other algorithms. This suggests that the deviation may be a development problem specific to the Stark algorithm or a lack of optimisation in a library utilised by the algorithm, rather than being caused by the virtualisation platform.

Table 5.25. Results of verifying time of hardware platforms

		Verifying Time (ms)			
Study	**ZKP System**	**System 1**	**System 2**	**System 3**	**Portable**
Chase et al.	ZKBoo++	104.94	105.70	189.80	210.52
Setty S.	Spartan	92.01	91.64	n.a.	n.a.
Chiesa et al.	Marlin	5180.98	5267.69	8604.45	10139.17
Chung et al.	Bulletproofs+	73.62	76.07	124.86	137.89
Mouris and Tsoutsos	Zilch	39.96	40.76	54.34	61.57
Ben-Sasson et al.	Stark	27.49	34.49	36.31	49.58

As demonstrated in Table 5.25, the obtained values are the result of the proof verification times of the six algorithms examined in this study on four different hardware platforms. The initial three algorithms enumerated in the table are part of the zk-SNARK group, which is distinguished by its possession of reliable, universal and transparent deployment characteristics. The proof verification times produced by these algorithms are listed in each row from the fastest to the slowest on different hardware platforms. The algorithm in the fourth row represents the Bulletproofs algorithm and the data is listed from the fastest to the slowest for each platform. The final two rows of the table correspond to the zk-STARK group and the values are once again listed from the fastest to the slowest. A closer inspection of the data reveals a common feature: the values obtained with the strongest hardware platform, the System 1 platform, increase towards the weakest platform, the portable platform, i.e. the performance decreases. This phenomenon can be characterized as the proof verification time performance of the algorithms varies in direct proportion to the system resources for each type of algorithm.

Table 5.26. Percentage results of verifying time of hardware platforms

		Verifying Time (%)			
Study	**ZKP System**	**System 1**	**System 2**	**System 3**	**Portable**
Chase et al.	ZKBoo++	100.00	99.29	55.29	49.85
Setty S.	Spartan	100.00	100.40	n.a.	n.a.
Chiesa et al.	Marlin	100.00	98.35	60.21	51.10
Chung et al.	Bulletproofs+	100.00	96.78	58.96	53.39
Mouris and Tsoutsos	Zilch	100.00	98.05	73.54	64.90
Ben-Sasson et al.	Stark	100.00	79.69	75.69	55.44

As demonstrated in Table 5.26, the System 1 platform is regarded as the reference platform. Consequently, the values obtained with this platform are accepted as 100%. Subsequent comparison of values obtained on alternative platforms is then made with these values expressed as percentages. When the values obtained by the algorithms on System 1 and System 2 platforms are compared in percentages, it is determined that the variation is minimal for ZKBoo++ and Spartan and Marlin algorithms. The variation was found to be approximately 1% for ZKBoo++, Spartan and Marlin, approximately 3% for Bulletproofs and approximately 2% for Zilch. This finding demonstrates that there is negligible variation (1-3%) between platforms with equivalent CPU power. Consequently, it can be deduced that the efficacy of proof verification of the aforementioned algorithms is proportionate to CPU power. In addition, significant discrepancies have been identified between System 1 and System 2 platforms with respect to RAM memory and GPU power, as illustrated in Table 4.1. Despite this, no significant inconsistency is observed between the values produced by other algorithms except Stark. This finding indicates that ZKBoo++, Spartan, Marlin, Bulletproofs and Zilch algorithms are not significantly impacted by variations in memory and GPU power or possess limited reliance on these resources. The observed discrepancy of approximately 20% in the Stark algorithm's performance can be interpreted as the primary performance metric of this algorithm being strongly proportional to the CPU as well as the available memory. Despite extensive research, no evidence has been found to date that the algorithms studied use GPU power. This assertion is founded on the negligible loss in performance values produced by the algorithms, despite the significant disparity in GPU power between System 1 and System 2 platforms. Consequently, a definitive conclusion cannot be drawn concerning the utilisation of GPU power by the algorithm for the Stark algorithm. This is attributable to the substantial disparity in GPU power between System 1 and System 2 platforms, though the discrepancy between System 2 and System 3 platforms is comparatively less pronounced. Despite this, the discrepancy in values obtained by the algorithm is comparatively less between System 2 and System 3 platforms than between System 1 and System 2. The observation of significant performance dips in other algorithms, with the exception of Stark, lends further credence to this

hypothesis. The System 3 platform exhibits lower levels of CPU power and memory in comparison to the System 2 platform. This disparity is evident in the performance metrics of nearly all algorithms, with the exception of Stark. Conversely, given the relatively minor disparity in GPU power, it is hypothesised that the substantial performance variations observed in other algorithms can be ascribed to RAM and CPU, rather than GPU.

Table 5.27 has been obtained by ranking the proof verification times of the six algorithms compared in our thesis study from the smallest to the largest on each system basis. The objective of this comparison is to ascertain the most efficient algorithm on each system basis in the proof verification process. Furthermore, for each algorithm, the mean of the values obtained for the four platforms on which the algorithm operates is determined and the positive or negative deviation of each hardware platform from the mean is evaluated.

Table 5.27. Average results of verifying time of hardware platforms

		Verifying Time (ms)				
Study	**ZKP System**	**System 1**	**System 2**	**System 3**	**Portable**	**Average**
Ben-Sasson et al.	Stark	27.49	34.49	36.31	49.58	36.97
Mouris and Tsoutsos	Zilch	39.96	40.76	54.34	61.57	49.16
Chung et al.	Bulletproofs+	73.62	76.07	124.86	137.89	103.11
Setty S.	Spartan	92.01	91.64	n.a.	n.a.	91.83
Chase et al.	ZKBoo++	104.94	105.70	189.80	210.52	152.74
Chiesa et al.	Marlin	5180.98	5267.69	8604.45	10139.17	7298.07

An examination of Table 5.27 reveals that the zk-STARK group algorithms Stark and Zilch, in that order, achieve the optimum proof verification times on all platforms. It is determined that the situation created by the ordering from the most efficient (shortest proof time) to the least efficient for the System 1 platform is also valid for the other three platforms. This finding serves to confirm the analysis made previously for the data presented in Table 5.26. This finding demonstrates that the performance of all algorithms is directly proportional to their pure computational power. A comparison of the calculated average with the values obtained in the systems shows that the System 1 and System 2 platforms are positively separated as a cluster, while System 3 and the Portable platform are negatively separated as a cluster. A more detailed examination of the overall ranking of the algorithms reveals that there is an acceptable level of difference between the first five algorithms, all of which offer proof verification times less than 200 milliseconds. The Marlin algorithm, which possesses a universal configuration, has been identified as significantly lagging behind and incapable of producing a practically applicable proof verification time, even on the most optimal platform. The disparity observed between the Bulletproofs algorithm, favoured for its compact proof size in practical real-world applications and the Spartan and ZKBoo++ algorithms in the zk-SNARK group, particularly on the System 1

platform, is indicative of the efficacy and practicality of zk-SNARK algorithms. In the course of evaluating the security ranking of the algorithms, it is noteworthy that three of the first four algorithms are considered secure for the post-quantum era. This observation suggests that the vast majority of NIZKP algorithms can be employed in practical applications with minimal concerns in the future. Finally, an evaluation of the linear increase observed in the performance values of all algorithms reveals that they are either not affected at all or are affected at a negligible level by factors other than CPU power, such as memory size, GPU power, etc., during the proof verification process. This outcome is analogous to the comparison results observed in the proof production process. However, it is noted that the data examined in this study exhibit a greater degree of proximity to each other (lacking any data demonstrating significant negative or positive deviations). Consequently, it can be asserted that the proof verification process demands significantly less complex calculations than the proof production process across all platforms. A comprehensive evaluation of the average evidence creation times depicted in Table 5.24 and the evidence verification times outlined in Table 5.27 reveals that the evidence verification times correspond to approximately one-tenth of the evidence production times. This finding serves to reinforce the validity of the proposed hypothesis.

Table 5.28. Results of proof size of hardware platforms

		Proof Size (KB)			
Study	**ZKP System**	**System 1**	**System 2**	**System 3**	**Portable**
Chase et al.	ZKBoo++	4512.00	4512.00	4512.00	4512.00
Setty S.	Spartan	141.79	141.77	n.a.	n.a.
Chiesa et al.	Marlin	0,88*	0,88*	0,88*	0,88*
Chung et al.	Bulletproofs+	0.67	0.67	0.67	0.67
Mouris and Tsoutsos	Zilch	774.28	774.28	774.28	774.28
Ben-Sasson et al.	Stark	401.64	401.64	269.50	401.64

'*' Not tested value, referenced from (Chiesa, Hu, et al., 2020) for BLS12-381 curve

As illustrated in Table 5.28, the proof sizes produced by the six tested algorithms on each hardware platform vary significantly. Upon examination of the table, a significant discrepancy is observed for the Stark algorithm on the System 3 platform and a negligible difference is evident for the Spartan algorithm on the System 2 platform. The proof sizes obtained for the remaining algorithms remain consistent across all platforms. This finding suggests that the proof sizes produced by the algorithms are independent of hardware resources and are instead directly influenced by the structure of the algorithm itself. The significant disparities observed between the algorithms belonging to the same family (ZKBoo++ and Spartan, Zilch and Stark) substantiate this hypothesis. While the Stark algorithm generates notably smaller proof sizes on the System 3 platform compared to other platforms, it exhibits a substantially longer proof production time on the former. This phenomenon can be interpreted as either an optimisation problem during the

algorithm's development phase or a performance issue originating from a library utilised by the algorithm. A collective evaluation of the obtained data indicates that the Bulletproofs algorithm exhibits the smallest evidence size production and comparatively rapid evidence production and verification times, as asserted in the original article. This renders the algorithm the optimal candidate for decentralised structures and systems characterised by high transaction loads, particularly due to its compact evidence size. Zk-STARK algorithms, notably Zilch, have demonstrated their capacity to supersede these algorithms in applications employing zk-SNARK algorithms, exhibiting comparable performance in evidence production and verification time despite their comparatively substantial evidence sizes. Conversely, Marlin and Spartan algorithms have demonstrated the necessity for further development due to their suboptimal performance, particularly in the evidence production process, despite their comparatively smaller evidence sizes.

5.6. Comparison of the Results with the Literature

In order to rigorously contextualize the findings of this study within the broader body of scholarly work, it is essential to conduct a systematic comparative analysis with prior research on NIZKP systems. While numerous studies have explored the performance characteristics of individual NIZKP protocols, as illustrated in Table 5.29, such investigations are often constrained by the uniformity of their hardware configurations or by the limited diversity of the cryptographic schemes under consideration. For instance, (Chiesa, Hu, et al., 2020) conducted an in-depth study on zk-SNARK constructions over the BLS12-381 curve, yet their evaluation primarily focused on CPU-based settings and did not account for alternative platforms such as GPUs or constrained environments. Similarly, the benchmarking performed by (Capko et al., 2022) provided valuable insights into the scalability of Bulletproofs in blockchain applications, but lacked comparative data across distinct hardware architectures. In contrast, this study introduces a platform-diverse benchmarking framework encompassing CPU, GPU, virtualized and mobile environments, thereby offering a broader perspective on protocol behavior under practical constraints. Moreover, the inclusion of protocols such as Spartan and ZKBoo++, which are relatively underrepresented in large-scale empirical studies, further differentiates this work. The head-to-head comparison under uniform experimental conditions not only highlights the algorithmic trade-offs in proof size and verification time but also exposes the extent to which hardware-level characteristics influence cryptographic performance. This section delineates the key similarities and divergences between the results obtained herein and those reported in related works, thereby elucidating the distinct contributions and practical implications of this research.

Table 5.29. Comparison of the Results with the Literature

Study	Hardware	Algoritmhs	Comparison Metrics	Literature Contribution
(El-Hajj & Oude Roelink, 2024)	12 Core CPU, 32 GB RAM	SNARK, STARK, Bulletproof	Setup time, Proof genaration time, Verification time, Proof size, Security level, Witness size	To compare three NIZKP protocols in the same application scenario under equal conditions.
(Oude Roelink et al., 2024)	-	SNARK, STARK, Bulletproof	Proof genaration time, Verification time, Key/Proof size, Setup complexity, scalability, quantum resistance, attack model	SLR between 2013 and 2023, evaluating existing works across diverse application domains
(Chen et al., 2022)	-	SNARK, STARK	CRS size, prover complexity, verifier complexity, trusted setup requirements, quantum resistance	Private Auctions and Decentralized Card Games offered
(Capko et al., 2022)	-	SNARK, STARK, Bulletproof	Trusted setup requirements, Proof size, Verification complexity, Cryptographic assumptions and Post-quantum resistance	Study serves as a bridge between theoretical ZKP frameworks and scalable, real-world blockchain applications.
(Gong et al., 2022)	-	SNARK, STARK, Bulletproof, MPC	Trusted setup, Complexity of algorithms, Application scenario, Quantum threats	Theoretical comparison, setup processes, efficiency and potential vulnerabilities
(Chen et al., 2021)	-	SNARK (Pinocchio, Groth16)	Prover and Verifier complexity, proof size, post-quantum security and setup requirements.	Theoretical comparison, protocol comparison, blockchain and zk-Rollup applications
(Panait & Olimid, 2021)	4 Core CPU, 8 GB RAM	SNARK, STARK	Setup time, proof generation time, verifier transfer cost, verifier execution cost,	Blockchain based identity management systems comparison on SNARKs and STARKs
(Nitulescu, 2020)	-	SNARK	Algortihmic complexity, proof size, verification time, trusted setup requirements, universality, post-quantum security and transparency.	Theoretical comparison of Sonic, Marlin, Plonk and DARK and Halo

(Khovratovich, 2018)	8 core CPU, 16 GB RAM	SNARK, STARK, Bulletproof	Prover and Verifier complexity, proof size, verify time, proof generation time, prover and verifier scalability, post-quantum security and setup requirements.	Comprehensively summarizes the current "state of the art" approaches in verifiable computing. It also provides valuable insight into the practical capabilities of different systems in real-world applications.
Our Study (2025)	4 Different Hardware Platform	SNARK, STARK, Bulletproof	Proof genaration time, Verification time, Proof size, Setup requirements	Comprehensively summarizes the current "state of the art" approaches in NIZKP algorithms. It also provides valuable insights into the practical capabilities of implementations by benchmarking the algorithms on different hardware.

In recent years, NIZKPs received significant attention in both academic and industrial contexts, particularly due to their applicability in privacy-preserving authentication and blockchain-based systems. However, most of the existing studies in the literature evaluated zk-SNARKs, zk-STARKs and Bulletproofs individually within distinct application domains, under varying hardware conditions and using different programming environments. This disparity introduced major obstacles to direct and meaningful comparisons between protocols. The benchmark study conducted by (El-Hajj & Oude Roelink, 2024) addressed this gap by providing a standardized and fair performance comparison of zk-SNARK, zk-STARK and Bulletproof protocols. Unlike earlier works, the authors implemented a uniform hash-based application (MiMC) across all three protocols, using the same hardware platform (AMD Ryzen 9 5900X, 32GB RAM) and the same programming language (Rust) wherever possible. This methodological uniformity enabled a reliable and practical comparison of the protocols under consistent and reproducible conditions. This study contributed to the literature by providing a standardized benchmarking framework that enabled fair comparison of zk-SNARK, zk-STARK and Bulletproof protocols within an identical application context. It evaluated key performance metrics such as proof time, verification time, proof size, setup requirements and security levels. Additionally, it offered practical insights into protocol selection based on security assumptions and application needs. The open-source implementation enhanced reproducibility and the work built directly upon the authors' prior SLR by addressing the identified gap in comparative performance evaluation. In conclusion, this work

represented a significant step forward in the empirical evaluation of NIZKP protocols, contributing a reproducible, detailed and context-aware comparison that complemented theoretical analyses. It made a significant contribution to the literature by enabling implementers to make informed decisions regarding protocol selection by considering both performance and security issues.

The systematic literature review conducted by (Oude Roelink et al., 2024) provided a comprehensive and structured synthesis of the application of three prominent NIZKP protocols—zk-SNARK, zk-STARK and Bulletproof—in privacy-preserving authentication systems. By examining 43 peer-reviewed articles published between 2013 and 2023, the study offered a unique comparative analysis of how these protocols had been utilized across various domains, including finance, healthcare, business and general-purpose applications. A key contribution of this work lay in its methodologically rigorous filtering and classification approach, which ensured that only high-quality, directly relevant studies were included. Unlike previous reviews that had either focused on theoretical aspects or individual case studies, this study bridged the gap between protocol design and real-world implementation by highlighting both performance outcomes and contextual challenges in practical deployments.

The study by (T. Chen et al., 2022) made a multifaceted contribution to the zk-SNARK literature by bridging theoretical constructs with practical implementations. Most notably, it proposed two original application scenarios—privacy-preserving auctions and decentralized card games—each supported by detailed zk-SNARK circuit designs. These implementations demonstrated the practical viability of ZKPs in real-world settings requiring confidentiality, fairness and verifiability. In addition, the study offered a comparative evaluation of two leading zk-SNARK frameworks, Groth16 and Pinocchio, providing empirical insights into their respective trade-offs in proof size, verification time and implementation flexibility. By openly sharing the Circom implementations of the proposed circuits, the study further enriched the field with concrete tools for practitioners. Beyond application-focused contributions, the authors critically examined methodological inconsistencies in performance reporting across the literature, highlighting the urgent need for standardized benchmarking practices. As a forward-looking recommendation, the study proposed the development of a comprehensive, real-world benchmark framework to enable consistent and reproducible evaluations of NIZKP protocols. Collectively, these contributions not only advanced academic understanding but also offered practical guidance for the secure deployment of zk-SNARKs in adversarial and privacy-sensitive digital environments.

The study conducted by (Capko et al., 2022) made a significant contribution to the literature by offering a structured and comparative analysis of ZKP techniques in the context of blockchain scalability, with a particular focus on ZK-rollups as Layer 2 solutions for Ethereum. Study introduced a generational taxonomy of ZKP systems—ranging from circuit-specific trusted

setups (G1) to universal setups (G2) and setup-free protocols (G3)—and classified widely adopted schemes such as Groth16, Plonk, STARKs and Bulletproofs accordingly. Through this classification, the study effectively mapped the trade-offs between efficiency, transparency and post-quantum security. While SNARKs were noted for their compact proofs and low verification costs, their reliance on trusted setups was highlighted as a limitation. Conversely, STARKs were praised for transparency and quantum resistance, albeit with larger proof sizes. Beyond theoretical classification, the study emphasized practical deployment strategies by surveying real-world rollup architectures and ZK-EVM-compatible solutions like zkSync and Polygon Hermez. Additionally, it acknowledged ongoing efforts in hardware acceleration—including FPGA, GPU and distributed systems—to improve proving efficiency. The key contribution of this work lies in its integrative approach, bridging abstract cryptographic properties with blockchain engineering needs. By presenting a clear taxonomy, comparative framework and optimization landscape, the study offers both a foundational reference for researchers and a strategic guide for developers aiming to implement scalable and privacy-preserving blockchain systems. Furthermore, it sets the stage for future research by identifying performance bottlenecks and promoting the integration of cryptographic innovation with hardware-level advancements.

Another publication by (Gong et al., 2022) presented a comprehensive analysis and comparison of existing ZKP schemes in terms of their complexity, application scenarios and resilience against quantum threats. The comparison aimed to provide engineers and researchers with a general understanding of the advantages and disadvantages of these algorithms and to raise awareness of the specific use cases associated with each ZKP scheme. In particular, the study discussed the setup processes, efficiency and potential vulnerabilities—including susceptibility to quantum attacks—of each protocol in detail. It served as a valuable guide to assist researchers and developers in selecting the most appropriate ZKP scheme for their specific application requirements.

In their 2021 publication, (T. Chen et al., 2021) provided a comprehensive and theoretically grounded examination of zk-SNARK protocols, offering both foundational insights and forward-looking perspectives. Through a structured comparison of leading constructions—such as Pinocchio and Groth16—the authors highlighted key performance trade-offs in terms of proof size, verification cost and setup requirements. Groth16 was emphasized as particularly efficient for blockchain use cases due to its succinctness and minimal verifier overhead, while Pinocchio's historical relevance was noted alongside its relatively higher computational demands. The study also contextualized zk-SNARKs within broader zero-knowledge frameworks (PCP, QAP, IOP) and extended the discussion to post-quantum secure alternatives like STARKs and Spartan, thereby enriching the comparative landscape. Although empirical benchmarks were not conducted, the work underscored the computational burden of pairing-based cryptography and suggested the

importance of hardware acceleration for practical deployment. Importantly, the authors proposed zk-SNARK-compatible application scenarios—privacy-preserving auctions and decentralized card games—which demonstrated the practical relevance of their theoretical analysis. Overall, the paper's key contribution lies in bridging abstract cryptographic concepts with real-world system design, offering a pedagogically structured reference that informs both foundational research and application-oriented development in privacy-preserving technologies.

The study by (Panait & Olimid, 2021) offered valuable practical insights into the integration of zk-SNARKs within blockchain-based identity management systems, using the ZoKrates framework. By deploying zk-SNARK-enabled smart contracts on the Ethereum blockchain and testing them on the Kovan testnet, the authors conducted an empirical evaluation of performance, gas costs and real-world feasibility. Their analysis demonstrated that while verification costs remained within acceptable limits, proof complexity and network conditions significantly affected setup times and transaction fees. Beyond technical evaluation, the study also identified critical security challenges—such as replay attacks, malleability risks and long-term confidentiality concerns associated with on-chain certificate storage. These findings highlighted the need for careful cryptographic design and application-specific optimization. As a notable contribution, the study provided a working implementation blueprint for privacy-preserving identity protocols on Ethereum, serving as a practical reference point for further development. It also emphasized the current immaturity of zk-SNARK/zk-STARK tooling and recommended further research into performance optimization, secure off-chain integration and improved framework resilience before production-grade deployment becomes feasible.

In a foundational study, (Nitulescu, 2020) presented a structured and modular introduction to zk-SNARKs, addressing their historical development, foundational definitions, construction methodologies and contemporary application domains. The work systematically categorized zk-SNARK frameworks into several major classes, including PCP-based constructions (e.g., Kilian's protocol, CS proofs), QAP-based schemes (e.g., GGPR13, Groth16) and more recent LIP/IOP-based models (e.g., Marlin, Sonic, Aurora). In addition, the survey highlighted post-quantum candidates such as STARK, Spartan and Halo and offered insights into their cryptographic resilience. Prominent protocols were comparatively evaluated with respect to various criteria, including proof size, verification time, trusted setup requirements, universality, post-quantum security and transparency. Notably, Groth16 stood out for its compact proof size and low verification latency, while newer systems such as Plonk and Marlin offered improved flexibility through universal setup. STARK-based schemes, despite their larger proofs, were particularly notable for their transparency and quantum resistance. Although the study did not provide experimental benchmarks, it drew upon existing literature to highlight empirical observations—particularly regarding zk-SNARKs' suitability for blockchain deployments such as Zcash. This

study made a significant contribution to the literature by offering a structured taxonomy of zk-SNARK frameworks and a thorough examination of key security aspects such as knowledge soundness and simulation extractability, articulated with notable conceptual clarity. Furthermore, it identified several open research directions, including transparent and post-quantum secure SNARK constructions, hardware-optimized proving systems and improvements in prover efficiency. As such, the study served both as an accessible entry point for newcomers and as a foundation for future academic exploration in privacy-preserving cryptographic protocols.

The study conducted by (Khovratovich, 2018) comprehensively summarized the then-current "state-of-the-art" approaches in verifiable computing. It identified the features required for different scenarios (public data, partially private data, multi-party computation) and described the fundamental principles and properties of various proof systems, including homomorphic public-key cryptography, discrete logarithm-based schemes, MPC variants and incremental verifiable computation. In particular, it provided a comparative analysis of these approaches in terms of key characteristics such as prover and verifier complexity, communication overhead, post-quantum security and the necessity of trusted setup. Additionally, it offered valuable insights into the practical capabilities of these systems in real-world applications by classifying existing implementations based on their maturity levels and comparing their performance using a benchmark involving one million multiplication gates. This work served as a comprehensive reference for researchers and developers seeking to determine which proof systems were most appropriate for their specific application requirements.

This thesis makes several important contributions to the literature on NIZKP systems by addressing key methodological and contextual gaps identified in previous research, summarized in Table 5.29.

First, the study systematically reviews the existing literature works and highlights two predominant comparative approaches commonly observed in the literature: intra-class comparisons of algorithms within the same cryptographic family and inter-class comparisons across different algorithmic categories. It further identifies a significant limitation in the methodological landscape—namely, the widespread reliance on single, homogeneous hardware or software environments for evaluation, which tends to obscure the platform-specific factors that influence algorithmic performance. To address these limitations, the present study adopts a robust and multi-layered methodology. It includes a conceptual and empirical evaluation of at least two representative algorithms from each of the three major families of NIZKPs (zk-SNARKs, zk-STARKs and Bulletproofs), followed by a comprehensive cross-class comparison. This two-tiered approach enables the investigation of performance variations both within and between algorithmic classes. The resulting insights contribute not only to the refinement of theoretical models but also

offer practical guidance for system designers and developers, particularly those working in decentralized and privacy-preserving environments.

A further and distinct contribution of this thesis lies in its deliberate inclusion of multiple execution environments. By conducting performance evaluations under two different software configurations and across four heterogeneous hardware platforms, the study systematically examines the effects of GPU presence, RAM capacity, CPU consistency and integrated graphics capabilities on the efficiency of cryptographic proof systems. This comprehensive experimentation enables the isolation and quantification of hardware-level influences, providing a level of granularity rarely observed in comparable studies.

Lastly, the introduction of a reference hardware platform as a benchmarking baseline adds a practical and scalable dimension to the performance evaluation. By comparing all other platforms relative to this standardized baseline, the study facilitates meaningful and reproducible comparisons across varied deployment scenarios. This allows researchers and practitioners to make informed predictions regarding the expected behavior of NIZKP algorithms under different infrastructural conditions, thereby contributing directly to the design of context-aware, resource-efficient and high-performance privacy-preserving systems.

In sum, this thesis extends the current body of knowledge through its comprehensive methodological framework, multidimensional comparative strategy and practical relevance for real-world system implementation. It is expected to serve as a valuable reference for both academic researchers seeking to advance theoretical discourse and practitioners aiming to optimize zero-knowledge systems for operational use.

6. CONCLUSION AND FUTURE WORKS

In the final chapter of this thesis, the overall findings derived from the performance evaluation of six prominent NIZKP algorithms are synthesized and critically assessed. The results are interpreted in light of their practical implications across different hardware configurations and application domains. Emphasis is placed on the interplay between algorithmic design choices and system-level constraints, which collectively influence the efficiency and applicability of each protocol. Additionally, this chapter outlines several directions for future research, aiming to extend the current work with more comprehensive hardware-level experimentation, security assessments and optimization techniques. By highlighting both the strengths and limitations observed, this section provides a conclusive perspective on the suitability of NIZKP schemes for diverse computational environments.

6.1. Conclusion

This thesis presented an extensive performance evaluation of six widely recognized NIZKP algorithms, categorized into three main families based on their cryptographic construction: zk-SNARK algorithms (ZKBoo++, Spartan and Marlin), Bulletproofs and zk-STARK algorithms (Stark and Zilch). The experimental results clearly demonstrated that the performance of these algorithms varies significantly depending on both their internal design and the characteristics of the underlying hardware platform. Therefore, selecting the most suitable NIZKP algorithm for a particular application must consider the computational environment, resource constraints and security requirements.

Among the evaluated algorithms, zk-SNARK-based solutions—especially ZKBoo++, but not Spartan and Marlin (despite their particularly good proof size)—showed relatively good performance in terms of proof generation and proof verification time on devices with limited computational resources. Their relatively small proof sizes and efficient verification steps make them highly applicable for scenarios such as blockchain-based authentication, lightweight authentication protocols in IoT systems and mobile application security. In particular, ZKBoo++ showed the fastest proof generation times in constrained environments among zk-SNARKs, while Spartan and Marlin offered better scalability and efficiency, but only on systems with robust system resources.

Bulletproofs, on the other hand, delivered a balanced performance profile across all tested hardware platforms. Its transparent setup (no trusted setup requirement), logarithmic proof size growth and moderate computational demands position it as a versatile option for privacy-preserving cryptocurrency systems, confidential transactions and decentralized finance (DeFi) applications operating in cloud or desktop environments. Bulletproofs may also serve well in

blockchain applications where medium-scale privacy guarantees are required without incurring excessive computational overhead.

In contrast, zk-STARK-based algorithms — Stark and Zilch — demonstrated significantly higher computational requirements, particularly during proof generation, due to their reliance on advanced cryptographic primitives and post-quantum secure constructions. However, these algorithms provided the strongest security guarantees among all evaluated methods, including resistance to quantum attacks and transparent setup properties. Their performance was considerably improved on high-performance computing platforms equipped with multi-core processors or hardware acceleration. Therefore, Stark and Zilch are most suitable for privacy-preserving applications in large-scale blockchain networks, zero-knowledge rollups, verifiable computation systems and data integrity verification in decentralized environments — especially when deployed on servers or cloud platforms with sufficient computational power.

In summary, the results of this thesis emphasize that the optimal choice of a NIZKP algorithm is highly dependent on the specific use-case and hardware constraints of the target system. While zk-SNARK algorithms such as ZKBoo++ are more appropriate for low-power or resource-limited environments, Bulletproofs serves as a middle-ground solution for applications requiring moderate privacy and efficiency. Conversely, zk-STARK algorithms like Stark and Zilch are better suited for future-proof, highly secure and scalable systems operating on powerful computing infrastructures**.**

6.2. Future Works

While this thesis provides valuable insights into the performance characteristics of various NIZKP algorithms across different hardware platforms, there are still several research directions that could be explored to further advance the understanding and practical applicability of these cryptographic protocols. One important extension of this work would involve conducting experiments on the same NIZKP algorithms, but running them directly on bare-metal physical hardware rather than virtualized or abstracted environments. This approach would allow for a more accurate measurement of the performance impact of different system resources such as memory bandwidth, CPU frequency, storage latency and cache hierarchy. Evaluating these algorithms on physically isolated hardware setups would help to identify resource bottlenecks and optimize algorithm-specific hardware configurations.

Another promising avenue for future research is to analyze the performance benefits of leveraging hardware acceleration technologies, particularly GPUs. Given the parallelizable nature of many cryptographic operations involved in NIZKP constructions — such as polynomial evaluations, FFT and Merkle tree computations — testing the evaluated algorithms on GPU-accelerated platforms could yield substantial performance improvements. GPU-based

implementations of algorithms like Spartan, Marlin or Stark may significantly reduce proof generation times, especially in environments where computational scalability is a critical factor. Moreover, comparative studies between CPU-based and GPU-accelerated deployments would provide valuable guidelines for real-world system designers aiming to select the most suitable deployment strategy.

Furthermore, future studies should consider testing these NIZKP algorithms on specialized hardware platforms such as FPGAs or ASICs. These hardware solutions offer customizable architectures optimized for specific cryptographic operations, potentially providing substantial gains in both energy efficiency and computation speed. An in-depth performance comparison between general-purpose hardware (CPU/GPU) and custom-designed FPGA/ASIC implementations could reveal important trade-offs between flexibility, performance, cost and power consumption. Additionally, this research direction aligns with industry trends, where privacy-preserving cryptographic protocols are increasingly integrated into dedicated hardware modules for secure and high-throughput applications.

Finally, beyond performance optimization, a comprehensive future research direction would involve evaluating the selected NIZKP algorithms from a security and cryptanalysis perspective, particularly in the context of post-quantum cryptography. Since zk-STARK-based algorithms (Stark and Zilch) are designed with quantum-resistant properties, a systematic comparison of their security assumptions against zk-SNARK algorithms (ZKBoo++, Spartan, Marlin) and Bulletproofs in the post-quantum era is of critical importance. This evaluation could include resistance against known quantum attacks, cryptanalytic efforts targeting specific primitives and compliance with emerging post-quantum security standards. Incorporating this security-oriented analysis would provide a holistic view of NIZKP algorithms, balancing both performance and security considerations and guiding their adoption in long-term, quantum-resilient cybersecurity architectures.

REFERENCES

Ambainis, A., Rosmanis, A., & Unruh, D. (2014). Quantum attacks on classical proof systems: The hardness of quantum rewinding. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 474–483. https://doi.org/10.1109/FOCS.2014.57

Ames, S., Hazay, C., Ishai, Y., & Venkitasubramaniam, M. (2017). Ligero: Lightweight sublinear arguments without a trusted setup. *Proceedings of the ACM Conference on Computer and Communications Security*, 2087–2104. https://doi.org/10.1145/3133956.3134104

Backes, M., Barbosa, M., Fiore, D., & Reischuk, R. M. (2015). ADSNARK: Nearly practical and privacy-preserving proofs on authenticated data. *Proceedings - IEEE Symposium on Security and Privacy*, *2015-July*, 271–286. https://doi.org/10.1109/SP.2015.24

Ben-Sasson, E., Bentov, I., Horesh, Y., & Riabzev, M. (2018). Scalable, transparent and post-quantum secure computational integrity. *Eprint.Iacr.Org*, *693423*, 1–83. https://eprint.iacr.org/2018/046.pdf

Ben-Sasson, E., Chiesa, A., Riabzev, M., Spooner, N., Virza, M., & Ward, N. P. (2019). Aurora: Transparent succinct arguments for R1CS. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11476 LNCS*, 103–128. https://doi.org/10.1007/978-3-030-17653-2_4

Beydemir, A. (2018). *Sıfır Bilgi İspatı Tabanlı Kimlik Doğrulama*. Msc Thesis, Gebze Technical University, Gebze, p. 62.

Bhadauria, R., Fang, Z., Hazay, C., Venkitasubramaniam, M., Xie, T., & Zhang, Y. (2020). Ligero++: A New Optimized Sublinear IOP. *Proceedings of the ACM Conference on Computer and Communications Security*, 2025–2038. https://doi.org/10.1145/3372297.3417893

Boneh, D., Drake, J., Fisch, B., & Gabizon, A. (2021). Halo Infinite: Proof-Carrying Data from Additive Polynomial Commitments. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12825 LNCS*, 649–680. https://doi.org/10.1007/978-3-030-84242-0_23

Bowe, S., Grigg, J., & Hopwood, D. (2019). Recursive Proof Composition without a Trusted Setup. *IACR Cryptology EPrint Archive*, *2019/1021*, 1–31. https://electriccoin.co/

Bunz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., & Maxwell, G. (2018). Bulletproofs: Short Proofs for Confidential Transactions and More. *Proceedings - IEEE Symposium on Security and Privacy*, *2018-May*, 315–334. https://doi.org/10.1109/SP.2018.00020

Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., & Maxwell, G. (2018). Bulletproofs: Short Proofs for Confidential Transactions and More. *Proceedings - IEEE Symposium on Security and Privacy*, *2018-May*, 315–334. https://doi.org/10.1109/SP.2018.00020

Bünz, B., Fisch, B., & Szepieniec, A. (2020). Transparent snarks from dark compilers. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12105 LNCS*, 677–706. https://doi.org/10.1007/978-3-030-45721-1_24

Capko, D., Vukmirovic, S., & Nedic, N. (2022). State of the Art of Zero-Knowledge Proofs in Blockchain. *2022 30th Telecommunications Forum, TELFOR 2022 - Proceedings*, 9–12. https://doi.org/10.1109/TELFOR56187.2022.9983760

Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., & Zaverucha, G. (2017). Post-quantum zero-knowledge and signatures from symmetric-key primitives. *Proceedings of the ACM Conference on Computer and Communications Security*, 1825–1842. https://doi.org/10.1145/3133956.3133997

Chen, B., Bünz, B., Boneh, D., & Zhang, Z. (2023). HyperPlonk: Plonk with Linear-Time Prover and High-Degree Custom Gates. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *14005 LNCS*, 499–530. https://doi.org/10.1007/978-3-031-30617-4_17

Chen, J., Chen, J., He, K., & Du, R. (2021). SeCrowd: Efficient secure interactive crowdsourcing via permission-based signatures. *Future Generation Computer Systems*, *115*, 448–458. https://doi.org/10.1016/j.future.2020.09.033

Chen, T., Lu, A., Kunpittaya, J., & Luo, A. (2021). *A Review of Zero Knowledge Proofs*. *December*, 1–32.

Chen, T., Lu, H., Kunpittaya, T., & Luo, A. (2022). A Review of zk-SNARKs. *ArXiv Preprint*, 1–34. http://arxiv.org/abs/2202.06877

Chen, Y., Ma, X., Tang, C., & Au, M. H. (2020). Pgc: Decentralized confidential payment system with auditability. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12308 LNCS*, 591–610. https://doi.org/10.1007/978-3-030-58951-6_29

Chiesa, A., Dev, O., & Spooner, N. (2020). FRACTAL: Post-Quantum and Transparent Recursive Proofs from Holography. *In Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020*, 769–793.

Chiesa, A., Hu, Y., Maller, M., Mishra, P., Vesely, P., & Ward, N. (2020). Marlin: Preprocessing zkSNARKs with universal and updatable SRS. *In Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020*, 738–768.

Chung, H., Han, K., Ju, C., Kim, M., & Seo, J. H. (2022). Bulletproofs+: Shorter Proofs for a Privacy-Enhanced Distributed Ledger. *IEEE Access*, *10*, 42081–42096. https://doi.org/10.1109/ACCESS.2022.3167806

Costello, C., Fournet, C., Howell, J., Kohlweiss, M., Kreuter, B., Naehrig, M., Parno, B., & Zahur, S. (2015). Geppetto: Versatile verifiable computation. *Proceedings - IEEE Symposium on Security and Privacy*, *2015-July*, 253–270. https://doi.org/10.1109/SP.2015.23

Eagen, L., Kanjalkar, S., Ruffing, T., & Nick, J. (2024). Bulletproofs++: Next Generation Confidential Transactions via Reciprocal Set Membership Arguments. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 249–279. https://doi.org/10.1007/978-3-031-58740-5_9

El-Hajj, M., & Oude Roelink, B. (2024). Evaluating the Efficiency of zk-SNARK, zk-STARK and Bulletproof in Real-World Scenarios: A Benchmark Study. *Information (Switzerland)*, *15*(8). https://doi.org/10.3390/info15080463

Gabizon, A., Williamson, Z. J., & Ciobotaru, O. (2019). Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *Cryptology EPrint Archive*.

Galal, H. S., & Youssef, A. M. (2020). Privacy Preserving Netting Protocol for Inter-bank Payments. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12484 LNCS*(December 2020), 319–334. https://doi.org/10.1007/978-3-030-66172-4_21

Gambs, S., Killijian, M.-O., Roy, M., & Traore, M. (2014). PROPS : A PRivacy-Preserving Location Proof System. *33rd IEEE International Symposium on Reliable Distributed Systems*, 1–11.

Ghosal, R., Sahai, A., & Waters, B. (2023). Non-Interactive Publicly-Verifiable Delegation of Committed Programs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 13941 LNCS*. https://doi.org/10.1007/978-3-031-31371-4_20

Giacomelli, I., Madsen, J., & Orlandi, C. (2016). ZKBoo: Faster zero-knowledge for boolean circuits. *Proceedings of the 25th USENIX Security Symposium*, 1069–1083.

Goldwasser, S., Micali, S., & Rackoff, C. (1985). The knowledge complexity of interactive proof-

systems. *In Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing (STOC '85)*, *18*(1), 291–304. https://doi.org/10.1137/0218012

Goldwasser, S., & Park, S. (2017). Public accountability vs. secret laws: Can they coexist? A cryptographic proposal. *WPES 2017 - Proceedings of the 2017 Workshop on Privacy in the Electronic Society, Co-Located with CCS 2017*, *2017-Janua*, 99–110. https://doi.org/10.1145/3139550.3139565

Gong, Y., Jin, Y., Li, Y., Liu, Z., & Zhu, Z. (2022). Analysis and comparison of the main zero-knowledge proof scheme. *Proceedings - 2022 International Conference on Big Data, Information and Computer Network, BDICN 2022*, 366–372. https://doi.org/10.1109/BDICN55575.2022.00074

González, A., & Zacharakis, A. (2021). Fully-Succinct Publicly Verifiable Delegation from Constant-Size Assumptions. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 13042 LNCS* (Issue 713673). https://doi.org/10.1007/978-3-030-90459-3_18

Hak, L., Fugkeaw, S., & Sato, H. (2024). LSAC:A Lightweight, Scalable and Anonymous Cross-Domain Authentication Scheme in IoMT. *Proceedings - 2024 IEEE Intelligent Mobile Computing, MobileCloud 2024*, 8–15. https://doi.org/10.1109/MobileCloud62079.2024.00009

Huang, H., Sun, X., Xiao, F., Zhu, P., & Wang, W. (2021). Blockchain-based eHealth system for auditable EHRs manipulation in cloud environments. *Journal of Parallel and Distributed Computing*, *148*(2021), 46–57. https://doi.org/10.1016/j.jpdc.2020.10.002

Huang, H., Zhu, P., Xiao, F., Sun, X., & Huang, Q. (2020). A blockchain-based scheme for privacy-preserving and secure sharing of medical data. *Computers and Security*, *99*, 102010. https://doi.org/10.1016/j.cose.2020.102010

Huang, J., Huang, T., Wei, H., Zhang, J., Yan, H., Wong, D. S., & Hu, H. (2024). zkChain: A privacy-preserving model based on zk-SNARKs and hash chain for efficient transfer of assets. *Transactions on Emerging Telecommunications Technologies*, *35*(4), 1–11. https://doi.org/10.1002/ett.4709

Kang, H., Dai, T., Jean-Louis, N., Tao, S., & Gu, X. (2019). FabZK: Supporting Privacy-Preserving, Auditable Smart Contracts in Hyperledger Fabric. *Proceedings - 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2019*, 543–555. https://doi.org/10.1109/DSN.2019.00061

Khovratovich, D. (2018). *State of the Art in Verifiable Computation*. *December*, 1–9.

Konkin, A., & Zapechnikov, S. (2021). Privacy methods and zero-knowledge poof for corporate

blockchain. *Procedia Computer Science*, *190*, 471–478. https://doi.org/10.1016/j.procs.2021.06.055

Kosba, A., Papadopoulos, D., Papamanthou, C., & Song, D. (2020). MIRAGE: Succinct arguments for randomized algorithms with applications to universal zk-SNARKs. *Proceedings of the 29th USENIX Security Symposium*, 2129–2146.

Lundkvist, C., Heck, R., Torstensson, J., Mitton, Z., & Michael, S. (2016). *Uport : a Platform for Self - Sovereign Identity*. http://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf

Maller, M., Kohlweiss, M., Bowe, S., & Meiklejohn, S. (2019). Sonic: Zero-knowledge snarks from linear-size universal and updatable structured reference strings. *Proceedings of the ACM Conference on Computer and Communications Security*, 2111–2128. https://doi.org/10.1145/3319535.3339817

Mohammad Hossein, K., Esmaeili, M. E., Dargahi, T., Khonsari, A., & Conti, M. (2021). BCHealth: A Novel Blockchain-based Privacy-Preserving Architecture for IoT Healthcare Applications. *Computer Communications*, *180*(August), 31–47. https://doi.org/10.1016/j.comcom.2021.08.011

Morais, E., Koens, T., van Wijk, C., & Koren, A. (2019). A survey on zero knowledge range proofs and applications. *SN Applied Sciences*, *1*(8), 1–17. https://doi.org/10.1007/s42452-019-0989-z

Mouris, D., & Tsoutsos, N. G. (2021). Zilch: A Framework for Deploying Transparent Zero-Knowledge Proofs. *IEEE Transactions on Information Forensics and Security*, *16*, 3269–3284. https://doi.org/10.1109/TIFS.2021.3074869

Mulder, V., Mermoud, A., Lenders, V., & Tellenbach, B. (2023). Trends in data protection and encryption technologies. In *Trends in Data Protection and Encryption Technologies*. https://doi.org/10.1007/978-3-031-33386-6

Nitulescu, A. (2020). *zk-SNARKs : A Gentle Introduction*. Ecole Normale Superieure. https://www.di.ens.fr/~nitulesc/files/Survey-SNARKs.pdf

Oude Roelink, B., El-Hajj, M., & Sarmah, D. (2024). Systematic review: Comparing zk-SNARK, zk-STARK and bulletproof protocols for privacy-preserving authentication. *Security and Privacy*, *April 2023*. https://doi.org/10.1002/spy2.401

Panait, A. E., & Olimid, R. F. (2021). On Using zk-SNARKs and zk-STARKs in Blockchain-Based Identity Management. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12596 LNCS* (pp. 130–145). https://doi.org/10.1007/978-3-030-69255-1_9

Parno, B., Howell, J., Gentry, C., & Raykova, M. (2016). Pinocchio: Nearly practical verifiable computation. *Communications of the ACM*, *59*(2), 103–112. https://doi.org/10.1145/2856449

Quan, Y. (2022). Improving Bitcoin's Post-Quantum Transaction Efficiency with a Novel Lattice-Based Aggregate Signature Scheme Based on CRYSTALS-Dilithium and a STARK Protocol. *IEEE Access*, *10*(October), 132472–132482. https://doi.org/10.1109/ACCESS.2022.3227394

Setty, S. (2020). Spartan: Efficient and general-purpose zksnarks without trusted setup. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12172 LNCS*, 704–737. https://doi.org/10.1007/978-3-030-56877-1_25

Uesugi, T., Shijo, Y., & Murata, M. (2021). Design and Evaluation of a Privacy-preserving Supply Chain System Based on Public Permissionless Blockchain. *ACM International Conference Proceeding Series*, 312–321. https://doi.org/10.1145/3459104.3459155

Wahby, R. S., Tzialla, I., Shelat, A., Thaler, J., & Walfish, M. (2018). Doubly-Efficient zkSNARKs Without Trusted Setup. *Proceedings - IEEE Symposium on Security and Privacy*, *2018-May*, 926–943. https://doi.org/10.1109/SP.2018.00060

Wang, N., Chau, S. C.-K., & Liu, D. (2024). SwiftRange: A Short and Efficient Zero-Knowledge Range Argument For Confidential Transactions and More. *2024 IEEE Symposium on Security and Privacy (SP)*, 1832–1848. https://doi.org/10.1109/sp54263.2024.00054

Wang, Y., & Kogan, A. (2018). Designing confidentiality-preserving Blockchain-based transaction processing systems. *International Journal of Accounting Information Systems*, *30*(June), 1–18. https://doi.org/10.1016/j.accinf.2018.06.001

Weng, C., Yang, K., Katz, J., & Wang, X. (2021). Wolverine: Fast, scalable and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. *Proceedings - IEEE Symposium on Security and Privacy*, *2021-May*, 1074–1091. https://doi.org/10.1109/SP40001.2021.00056

Xie, T., Zhang, Y., & Song, D. (2022). Orion: Zero Knowledge Proof with Linear Prover Time. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13510 LNCS*, 299–328. https://doi.org/10.1007/978-3-031-15985-5_11

Yang, K., Sarkar, P., Weng, C., & Wang, X. (2021). QuickSilver: Efficient and Affordable Zero-Knowledge Proofs for Circuits and Polynomials over Any Field. *Proceedings of the ACM Conference on Computer and Communications Security*, 2986–3001. https://doi.org/10.1145/3460120.3484556

Zhang, J., Liu, T., Wang, W., Zhang, Y., Song, D., Xie, X., & Zhang, Y. (2021). Doubly Efficient Interactive Proofs for General Arithmetic Circuits with Linear Prover Time. *Proceedings of the ACM Conference on Computer and Communications Security*, 159–177. https://doi.org/10.1145/3460120.3484767

Zhang, J., Xie, T., Zhang, Y., & Song, D. (2020). Transparent polynomial delegation and its applications to zero knowledge proof. *Proceedings - IEEE Symposium on Security and Privacy*, *2020-May*, 859–876. https://doi.org/10.1109/SP40000.2020.00052

Zhang, Q., Yu, Y., Li, H., Yu, J., & Wang, L. (2023). Trustworthy sealed-bid auction with low communication cost atop blockchain. *Information Sciences*, *631*(November 2022), 202–217. https://doi.org/10.1016/j.ins.2023.02.069

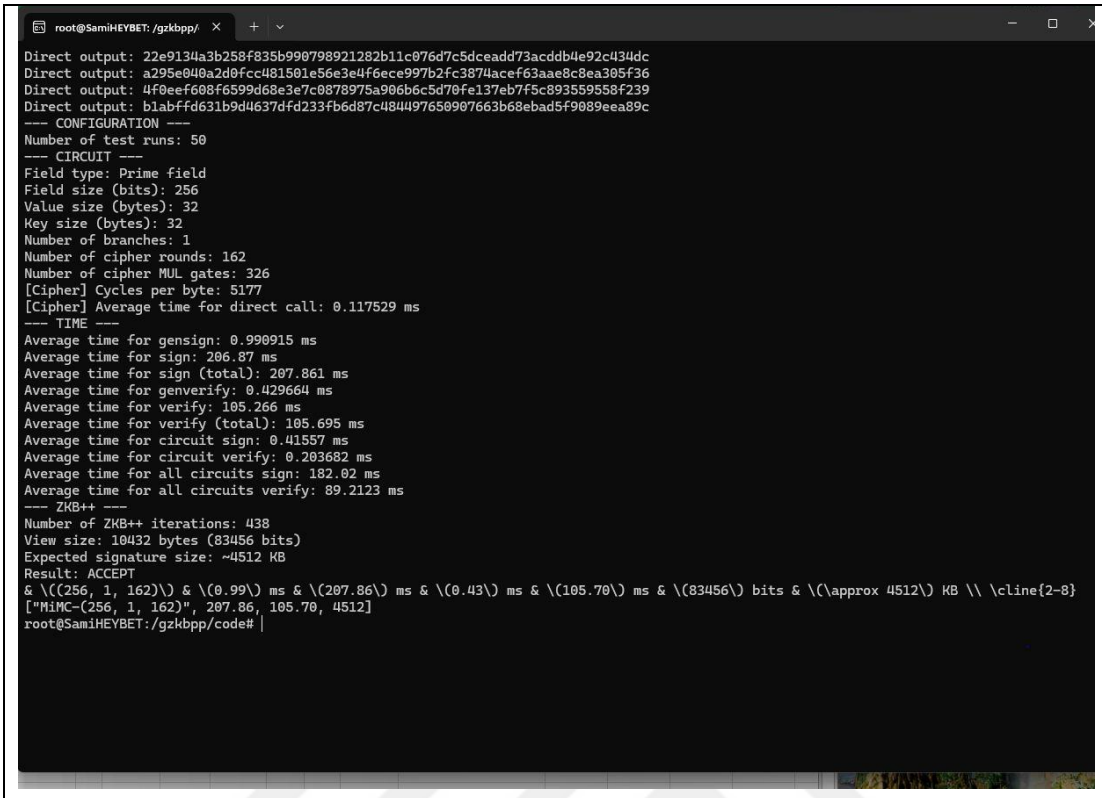
Zheng, H., You, L., & Hu, G. (2022). A novel insurance claim blockchain scheme based on zero-knowledge proof technology. *Computer Communications*, *195*(April), 207–216. https://doi.org/10.1016/j.comcom.2022.08.007

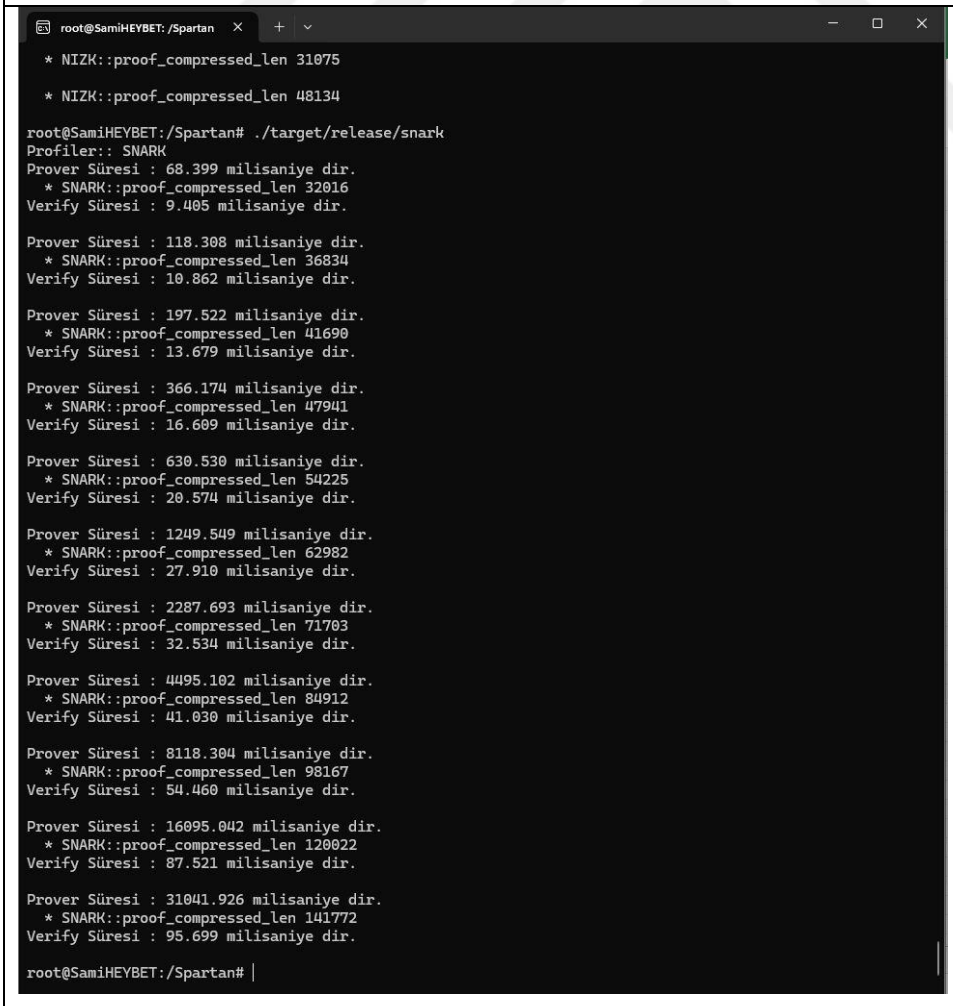
CURRICULUM VITAE

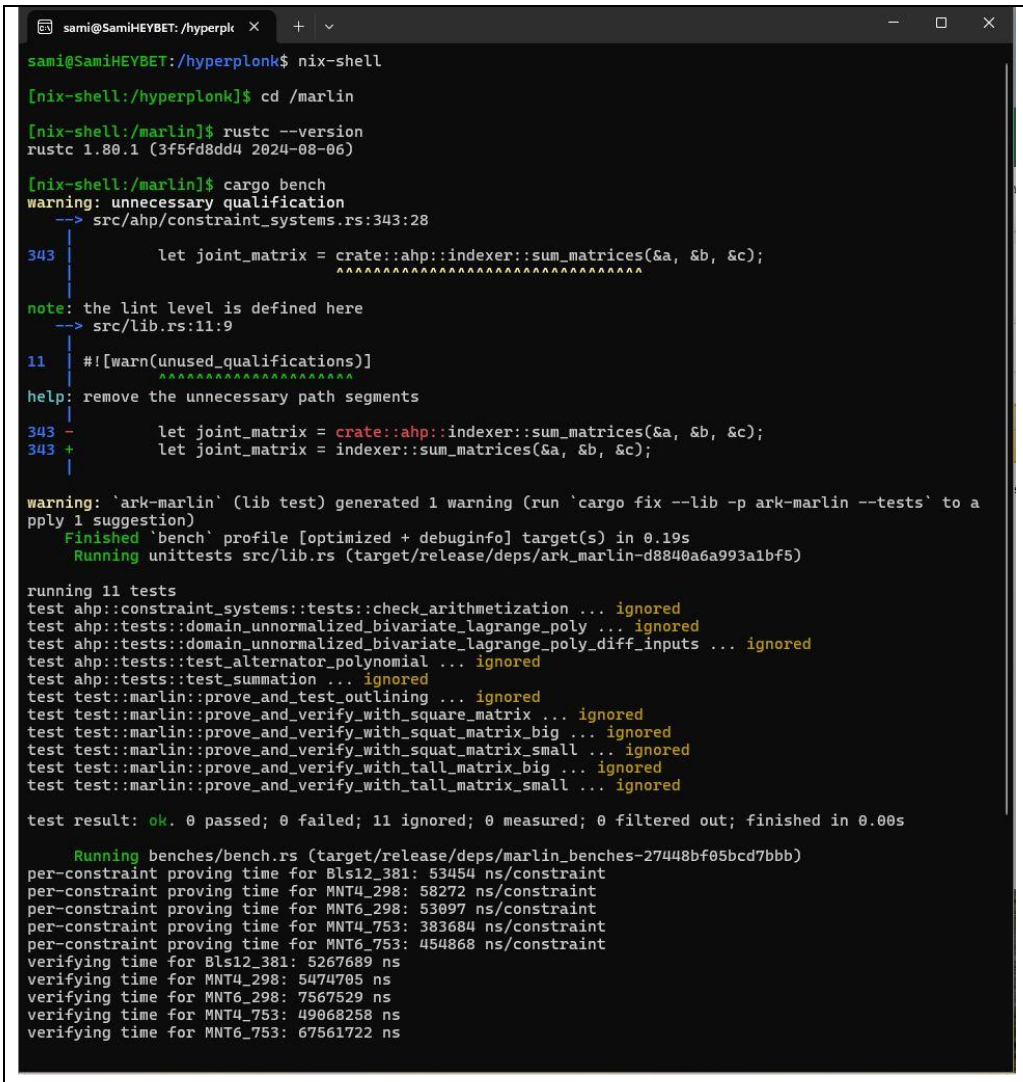
Sami HEYBET completed his high school education at Adana Central Technical High School in 1991. He received his undergraduate degree in Business Administration from the Faculty of Open Education, Anadolu University, in 2014. He later earned a Bachelor of Science (B.Sc.) degree in Computer Engineering from Çukurova University in 2021. He is currently a Lecturer in the Department of Cloud Computing Operations at Adana Alparslan Türkeş Science and Technology University.

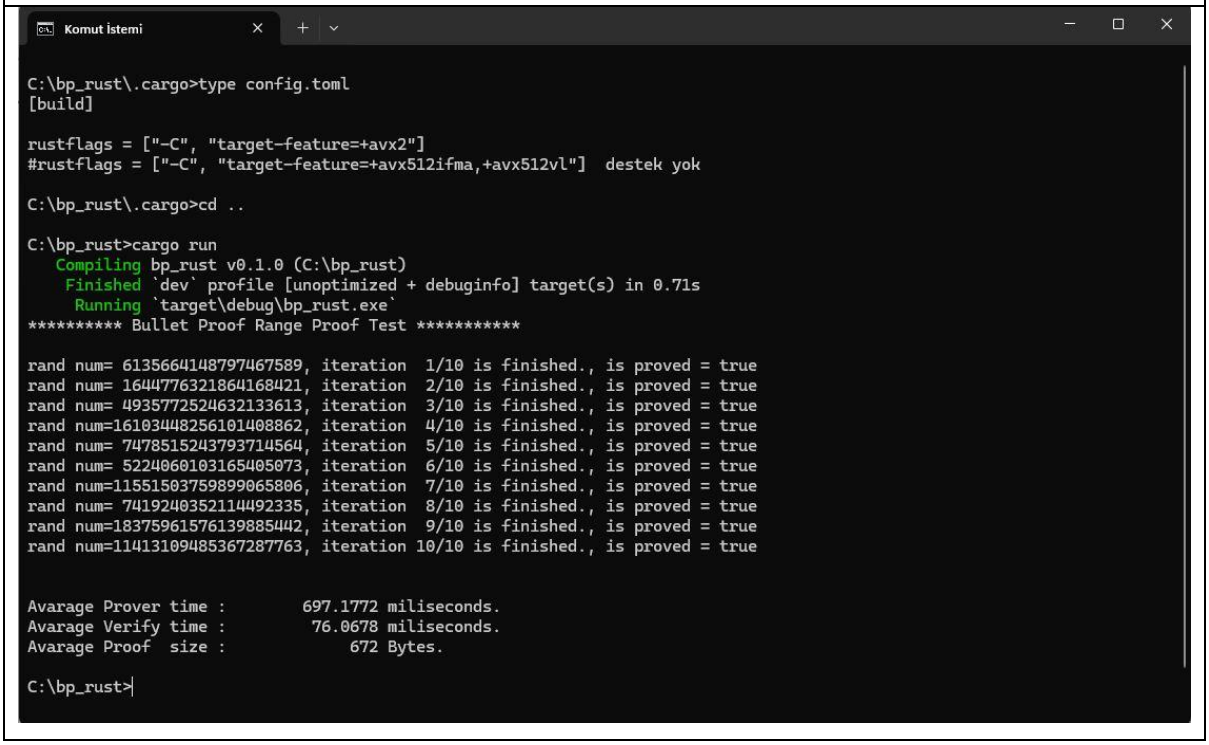
APPENDICES

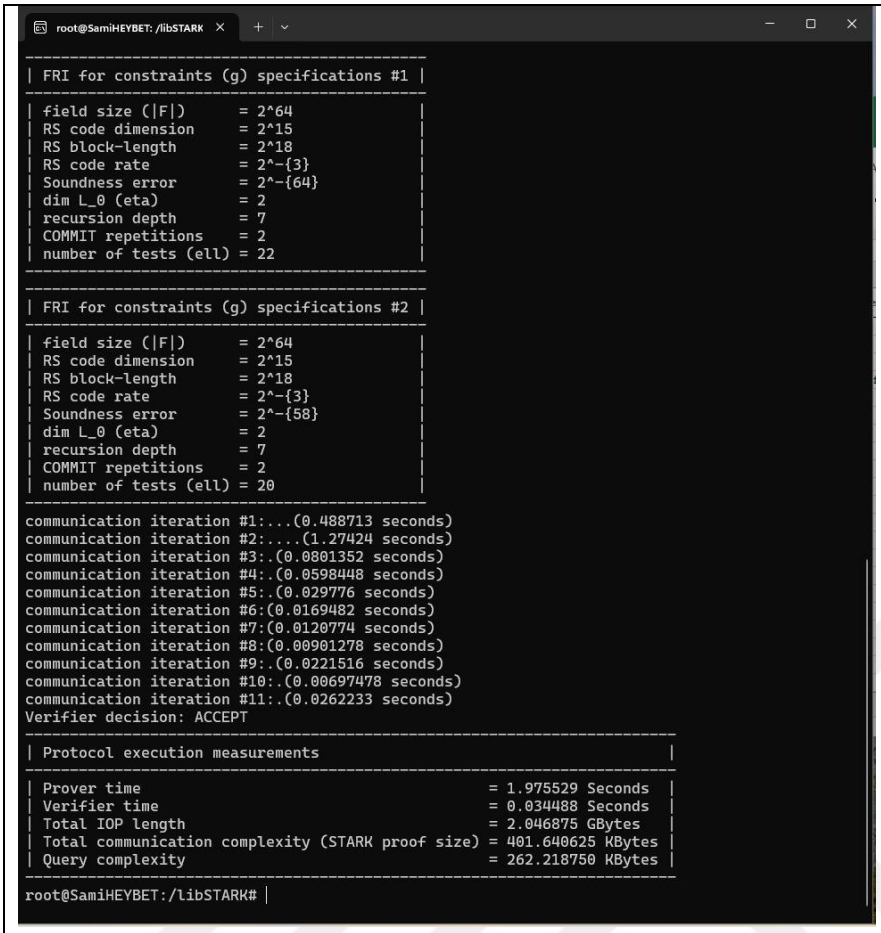
A.1. Screen Shot of ZKP Algorithms on Hardware-1 Platform

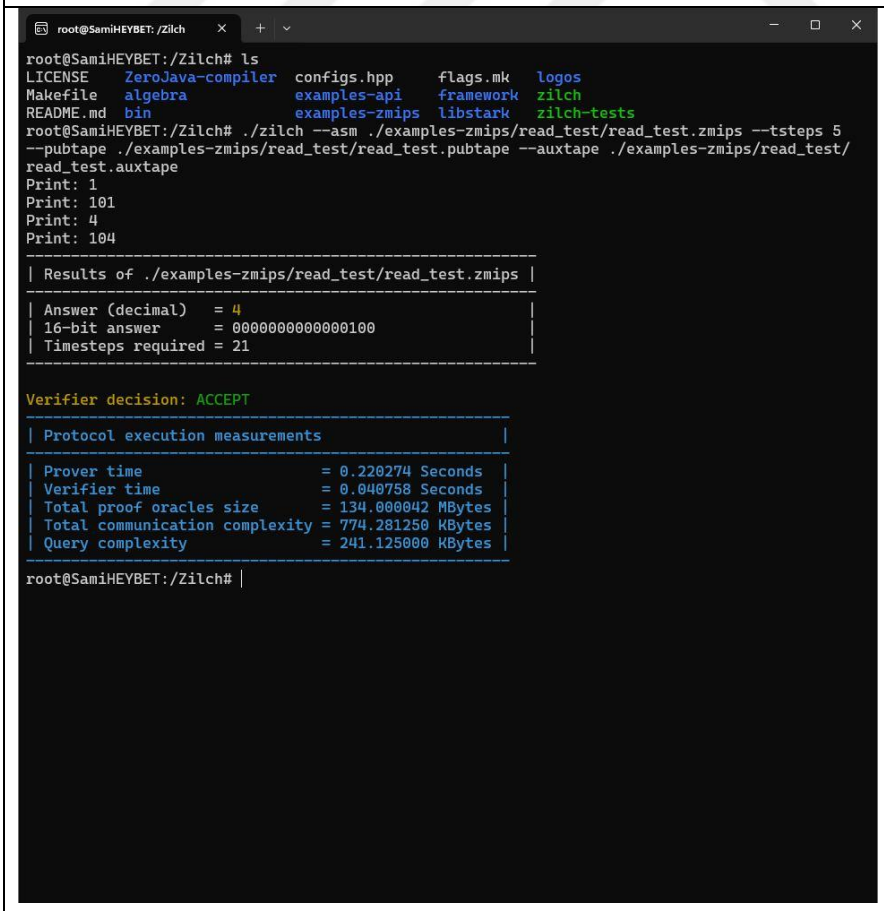





sami@SamiHEYBET: /hyperplk
sami@SamiHEYBET:/hyperplonk$ nix-shell
[nix-shell:/hyperplonk]$ cd /marlin
[nix-shell:/marlin]$ rustc --version
rustc 1.80.1 (3f5fd8dd4 2024-08-06)
[nix-shell:/marlin]$ cargo bench
warning: unnecessary qualification
--> src/ahp/constraint_systems.rs:343:28
let joint_matrix = crate::ahp::indexer::sum_matrices(&a, &b, &c);
note: the lint level is defined here
--> src/lib.rs:11:9
#![warn(unused_qualifications)]
help: remove the unnecessary path segments
343 - let joint_matrix = crate::ahp::indexer::sum_matrices(&a, &b, &c);
343 + let joint_matrix = indexer::sum_matrices(&a, &b, &c);
warning: `ark-marlin` (lib test) generated 1 warning (run `cargo fix --lib -p ark-marlin --tests` to apply 1 suggestion)
Finished `bench` profile [optimized + debuginfo] target(s) in 0.19s
Running unittests src/lib.rs (target/release/deps/ark_marlin-d8840a6a993a1bf5)
running 11 tests
test ahp::constraint_systems::tests::check_arithmetization ... ignored
test ahp::tests::domain_unnormalized_bivariate_lagrange_poly ... ignored
test ahp::tests::domain_unnormalized_bivariate_lagrange_poly_diff_inputs ... ignored
test ahp::tests::test_alternator_polynomial ... ignored
test ahp::tests::test_summation ... ignored
test test::marlin::prove_and_test_outlining ... ignored
test test::marlin::prove_and_verify_with_square_matrix ... ignored
test test::marlin::prove_and_verify_with_squat_matrix_big ... ignored
test test::marlin::prove_and_verify_with_squat_matrix_small ... ignored
test test::marlin::prove_and_verify_with_tall_matrix_big ... ignored
test test::marlin::prove_and_verify_with_tall_matrix_small ... ignored
test result: ok. 0 passed; 0 failed; 11 ignored; 0 measured; 0 filtered out; finished in 0.00s
Running benches/bench.rs (target/release/deps/marlin_benches-27448bf05bcd7bbb)
per-constraint proving time for Bls12_381: 53454 ns/constraint
per-constraint proving time for MNT4_298: 58272 ns/constraint
per-constraint proving time for MNT6_298: 53097 ns/constraint
per-constraint proving time for MNT4_753: 383684 ns/constraint
per-constraint proving time for MNT6_753: 454868 ns/constraint
verifying time for Bls12_381: 5267689 ns
verifying time for MNT4_298: 5474705 ns
verifying time for MNT6_298: 7567529 ns
verifying time for MNT4_753: 49068258 ns
verifying time for MNT6_753: 67561722 ns


Komut İstemi
C:\bp_rust\.cargo>type config.toml
[build]
rustflags = ["-C", "target-feature=+avx2"]
#rustflags = ["-C", "target-feature=+avx512ifma,+avx512vl"] destek yok
C:\bp_rust\.cargo>cd ..
C:\bp_rust>cargo run
Compiling bp_rust v0.1.0 (C:\bp_rust)
Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.71s
Running `target\debug\bp_rust.exe`
********** Bullet Proof Range Proof Test **********
rand num= 61356641487974675 89, iteration 1/10 is finished., is proved = true
rand num= 16447763218641684 21, iteration 2/10 is finished., is proved = true
rand num= 49357725246321336 13, iteration 3/10 is finished., is proved = true
rand num=16103448256101408862, iteration 4/10 is finished., is proved = true
rand num= 74785152437937145 64, iteration 5/10 is finished., is proved = true
rand num= 52240601031654050 73, iteration 6/10 is finished., is proved = true
rand num=11551503759899065806, iteration 7/10 is finished., is proved = true
rand num= 74192403521144923 35, iteration 8/10 is finished., is proved = true
rand num=18375961576139885442, iteration 9/10 is finished., is proved = true
rand num=11413109485367287763, iteration 10/10 is finished., is proved = true
Avarage Prover time : 697.1772 miliseconds.
Avarage Verify time : 76.0678 miliseconds.
Avarage Proof size : 672 Bytes.
C:\bp_rust>


root@SamiHEYBET: /libSTARK
| FRI for constraints (g) specifications #1 |
| field size (|F|) = 2^64
| RS code dimension = 2^15
| RS block-length = 2^18
| RS code rate = 2^-{3}
| Soundness error = 2^-{64}
| dim L_0 (eta) = 2
| recursion depth = 7
| COMMIT repetitions = 2
| number of tests (ell) = 22
| FRI for constraints (g) specifications #2 |
| field size (|F|) = 2^64
| RS code dimension = 2^15
| RS block-length = 2^18
| RS code rate = 2^-{3}
| Soundness error = 2^-{58}
| dim L_0 (eta) = 2
| recursion depth = 7
| COMMIT repetitions = 2
| number of tests (ell) = 20
communication iteration #1:...(0.488713 seconds)
communication iteration #2:....(1.27424 seconds)
communication iteration #3:.(0.0801352 seconds)
communication iteration #4:.(0.0598448 seconds)
communication iteration #5:.(0.029776 seconds)
communication iteration #6:(0.0169482 seconds)
communication iteration #7:(0.0120774 seconds)
communication iteration #8:(0.00901278 seconds)
communication iteration #9:.(0.0221516 seconds)
communication iteration #10:.(0.00697478 seconds)
communication iteration #11:.(0.0262233 seconds)
Verifier decision: ACCEPT
| Protocol execution measurements |
| Prover time = 1.975529 Seconds |
| Verifier time = 0.034488 Seconds |
| Total IOP length = 2.046875 GBytes |
| Total communication complexity (STARK proof size) = 401.640625 KBytes |
| Query complexity = 262.218750 KBytes |
root@SamiHEYBET:/libSTARK#


root@SamiHEYBET: /Zilch
root@SamiHEYBET:/Zilch# ls
LICENSE ZeroJava-compiler configs.hpp flags.mk logos
Makefile algebra examples-api framework zilch
README.md bin examples-zmips libstark zilch-tests
root@SamiHEYBET:/Zilch# ./zilch --asm ./examples-zmips/read_test/read_test.zmips --tsteps 5
--pubtape ./examples-zmips/read_test/read_test.pubtape --auxtape ./examples-zmips/read_test/
read_test.auxtape
Print: 1
Print: 101
Print: 4
Print: 104
| Results of ./examples-zmips/read_test/read_test.zmips |
| Answer (decimal) = 4
| 16-bit answer = 0000000000000100
| Timesteps required = 21
Verifier decision: ACCEPT
| Protocol execution measurements |
| Prover time = 0.220274 Seconds |
| Verifier time = 0.040758 Seconds |
| Total proof oracles size = 134.000042 MBytes |
| Total communication complexity = 774.281250 KBytes |
| Query complexity = 241.125000 KBytes |
root@SamiHEYBET:/Zilch#