

This is the author-created version of the following work:

Banaeian Far, Saeed, Chalak Qazani, Mohamadreza, and Imani Rad, Azadeh
(2025) *Exploring blockchain-based open finance: The role of advanced*
cryptographic tools, state-of-the-art achievements, and background challenges.
Peer-to-Peer Networking and Applications, 18 .

Access to this file is available from:

<https://researchonline.jcu.edu.au/86690/>

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC,
part of Springer Nature 2025

Please refer to the original source for the final version of this work:

<https://doi.org/10.1007/s12083%2D025%2D01979%2Dw>

Exploring blockchain-based open finance: The role of advanced cryptographic tools, state-of-the-art achievements, and background challenges

Saeed Banaeian Far · Mohammad Reza Chalak Qazani · Azadeh Imani Rad

Received: date / Accepted: date

Abstract The rapid advancement of financial technologies has catalyzed the evolution of Blockchain-Based Open Finance (BB-OpenFi), an emerging paradigm that integrates the decentralized, trustless nature of Decentralized Finance (DeFi) with the regulatory compliance and structured oversight of Open Finance (OpenFi). This paper explores how advanced cryptographic techniques, including Zero-Knowledge Proofs (ZKPs), Secure Multi-Party Computation (MPC), and Homomorphic Encryption (HE), address critical challenges in security, privacy, scalability, and regulatory compliance in the BB-OpenFi framework. We propose a comprehensive cryptographic framework that ensures confidentiality, integrity, and verifiability in financial transactions, overcoming limitations such as key escrow vulnerabilities, consensus-driven parameter determination, and secret parameter protection. By providing formal cryptographic models, implementation guidelines, and comparative analyses, our findings demonstrate that the BB-OpenFi framework is a highly scalable and secure next-generation financial paradigm. This research not only bridges the gap between traditional finance, OpenFi, and DeFi, but also sets a new standard for trust and efficiency in global financial systems, making it a valuable reference for academia, industry, and regulatory bodies.

S. Banaeian Far (Corresponding author)
Blockchain and Metaverse research lab, Tehran, Iran
E-mail: saeed.banaeian.far@gmail.com

M. R. Chalak Qazani
Faculty of Computing and Information Technology, Sohar University, Sohar, Oman
E-mail: mqazani@su.edu.om

A. Imani Rad
Blockchain and Metaverse research lab, Tehran, Iran
E-mail: azi.imani.rad@gmail.com

Keywords DAO; DeFi; Formal models; Key escrow; Quantum-safe systems; Verifiable computation

Highlights

- BB-OpenFi is a secure, privacy-preserving, and scalable evolution of DeFi.
- Advanced cryptographic tools, including ZKPs and HE, are used to enhance confidentiality, integrity, and verifiable computing.
- We offer solutions to critical challenges, such as key escrow, scalability, throughput, and parameter determination in BB-OpenFi systems.
- BB-OpenFi's achievements in addressing the blockchain trilemma, regulatory compliance, and user trust, distinguish it from traditional OpenFi and DeFi.

1 Introduction

Traditional banking systems act as custodians of users' sensitive financial data, requiring individuals to trust banks with access to their private information [1]. This centralized model often gives financial institutions significant control, raising concerns about data misuse, unauthorized sharing, and breaches of confidentiality [2, 3]. While crucial for providing financial services, this access raises inherent concerns regarding privacy and security, particularly in an era marked by escalating cyber threats and data breaches [4, 5]. As digital transactions increase, the risks associated with this centralized control continue to grow.

Open banking emerged as a transformative concept, enabling banks to share user data securely with third-party financial service providers (FSPs) through Application Programming Interfaces (APIs) [6, 7]. Under the

PSD2¹ (Payment Services Directive 2) standard, regulated institutions like FinTechs gain access to banking data, fostering innovation and enhancing financial services. Open APIs have become pivotal in creating a collaborative financial ecosystem where banks' partners can provide tailored services while ensuring compliance with established regulations.

Open finance (OpenFi) extends the principles of open banking by incorporating a wider range of financial services, including insurance, capital markets, and tax management [8,9,10]. This includes insurance, capital markets, taxation, and other related domains, creating a more comprehensive view of an individual's financial life. With the introduction of the PSD3² standard and frameworks such as Financial Data Access³ (FIDA), OpenFi offers users a more integrated experience. By broadening access to financial data, it aims to deliver diverse, user-centric solutions while emphasizing interoperability across financial ecosystems.

The expansion of OpenFi necessitates careful consideration of privacy implications, particularly about the General Data Protection Regulation⁴ (GDPR). The regulation ensures users have control over their sensitive information through consent management practices, emphasizing transparency and accountability. OAuth 2.0⁵, a widely adopted token-based authorization standard, supports secure data sharing while maintaining user consent. It enables OpenFi solutions to align with GDPR requirements and build trust among users.

DeFi leverages blockchain technology to create transparent, trustless, and permissionless financial services, revolutionizing traditional systems [11,12,13]. Built on Web3 and decentralized applications (DApps) [14], DeFi contrasts sharply with centralized, web-based, and app-based services. Unlike OpenFi, which relies on centralized intermediaries, DeFi provides enhanced transparency, resilience to censorship, and user autonomy, addressing many limitations of traditional OpenFi models [15,16]. This disintermediation provides users with greater autonomy and control over their assets and data, contrasting significantly with the centralized nature of OpenFi and the often-opaque operations of traditional financial institutions.

In the digital era, data is recognized as a paramount asset, fueling innovation and economic growth across various sectors [17,18]. For example, data is a critical asset, underpinning sectors like healthcare, telecommunications, and finance. Open data initiatives have driven the emergence of the Open Economy, where accessible and interoperable data fuels innovation and collaboration [19]. Standards such as OpenID⁶, developed by consortiums, facilitate secure identity verification, demonstrating how shared frameworks can promote privacy and trust in an increasingly interconnected world.

DeFi applications face unique challenges in aligning with GDPR requirements for data ownership, usage, and processing [20]. The decentralized nature of DeFi conflicts with GDPR's emphasis on centralized data accountability, creating tensions around compliance. The pseudonymity inherent in blockchain also raises concerns about identifying data subjects, thus creating a need for novel solutions to reconcile the decentralized nature of DeFi with the regulatory requirements of data privacy frameworks like GDPR [21,22]. While DeFi promotes user control and anonymity, reconciling these features with GDPR's demand for transparency and consent remains a critical area of exploration.

1.1 Motivation contribution

The inherent security and transparency of blockchain, coupled with its capacity for disintermediation, can address many of the privacy and security concerns associated with traditional OpenFi paradigms [15,16]. DeFi has the potential to reshape OpenFi by addressing security and privacy challenges through advanced cryptographic tools [13]. These tools can enhance trust, safeguard sensitive data, and ensure regulatory compliance without sacrificing decentralization. By integrating these technologies, DeFi can offer a robust alternative to centralized financial systems, fostering an inclusive and secure financial ecosystem.

Advanced cryptographic tools such as Multi-Party Computation (MPC) [23], Homomorphic Encryption (HE) [24,25], Searchable Encryption (SE) [26], and Zero-Knowledge Proofs (ZKPs) [27] offer promising solutions to address the privacy challenges inherent in OpenFi and DeFi. These technologies enable secure data sharing, private computations, and verifiable interactions, ensuring that DeFi applications meet stringent privacy and regulatory requirements. Moreover, they enable data processing and sharing without revealing the underlying sensitive information, allowing for the implementation of various GDPR aspects, such as data minimiza-

¹ https://ec.europa.eu/commission/presscorner/detail/pl/MEMO_17_4961

² [https://www.europarl.europa.eu/RegData/docs_autres_institutions/commission_europeenne/com/2023/0366/COM_COM\(2023\)0366_EN.pdf](https://www.europarl.europa.eu/RegData/docs_autres_institutions/commission_europeenne/com/2023/0366/COM_COM(2023)0366_EN.pdf)

³ https://finance.ec.europa.eu/document/download/d8c27557-05cd-4d03-9db7-d195baa18cbc_en?filename=finance-events-230905-presentation_en.pdf

⁴ <https://gdpr-info.eu/>

⁵ <https://oauth.net/2/>

⁶ <https://openid.net/developers/specs/>

tion, consent management, and secure data analysis [20, 21, 22]. These tools represent a crucial area of research and innovation for developing privacy-preserving financial technologies. By leveraging these tools, DeFi can offer solutions that uphold user privacy while promoting transparency and accountability.

This study introduces a novel cryptographic framework tailored for the BB-OpenFi framework, addressing key security, privacy, and scalability challenges that hinder the adoption of blockchain-based financial ecosystems. Unlike existing DeFi and OpenFi models that struggle with either security vulnerabilities or regulatory misalignment, our proposed framework ensures compliance, efficiency, and decentralization through state-of-the-art cryptographic primitives. This work makes several notable contributions to the field of blockchain and cryptographic research:

1. *A comprehensive cryptographic framework:* We formally define and integrate ZKPs, MPC, HE, and threshold cryptography, demonstrating how these tools can mitigate risks such as unauthorized data exposure, key escrow issues, and non-compliant financial interactions.
2. *Enhanced security and privacy measures:* By leveraging zk-SNARKs, zk-STARKs, and post-quantum cryptographic schemes, our approach ensures robust user anonymity, data protection, and transaction confidentiality, setting a higher security standard compared to traditional OpenFi models.
3. *Scalability without sacrificing security:* We introduce Layer-2 solutions, off-chain verifiable computing, and efficient consensus mechanisms, significantly improving throughput and cost-effectiveness while maintaining a decentralized trust model.
4. *Regulatory compliance and interoperability:* Our framework aligns with GDPR, PSD2, and FIDA while preserving decentralization, enabling real-world adoption and regulatory approval.
5. *Real-world applications and industry adoption:* The research findings pave the way for secure digital asset management, privacy-preserving financial transactions, decentralized identity verification, and scalable cross-border payments, reinforcing the BB-OpenFi framework's viability in both academic research and industry deployment.

These contributions collectively position the BB-OpenFi framework as a robust, scalable, and regulation-aware financial ecosystem, enabling financial institutions, fintech innovators, and policymakers to embrace decentralized financial models without compromising security or compliance. Key motivators include:

- *Future-Proof Design:* Its compliance-ready, secure, and scalable architecture ensures long-term viability.
- *Competitive Edge:* Early adopters gain an advantage by offering superior financial services.
- *Global Impact:* The democratization and accessibility of the BB-OpenFi framework align with ethical and inclusive economic goals.

We believe the BB-OpenFi framework represents the pinnacle of blockchain innovation and is an opportunity to lead the next generation of financial systems.

1.2 Outlines

The rest of the paper is organized as follows. Section 2 outlines the paper's preliminaries and background technologies. Section 3 addresses several advanced cryptographic tools that can be applied in the background of the BB-OpenFi framework. Section 4 identifies the methods to reach security and privacy features using the aforementioned tools in the BB-OpenFi framework. Section 5 proposes the achievements of the BB-OpenFi framework. Section 6 addresses several important challenges of developing the BB-OpenFi framework and offers possible solutions. Section 7 provides a general comparison between outlined cryptographic tools if they have been applied in the BB-OpenFi framework, and illustrates the output of experimental results. Ultimately, Section 8 concludes this study and determines future directions for both academia and industry sectors.

Table 1 shows the acronyms used in this study.

2 Preliminaries and background technologies

The rapid evolution of financial technology (FinTech), coupled with advancements in blockchain technology, has reshaped the global financial landscape. This section delves into the foundational paradigms that underpin these advancements, focusing on OpenFi, Blockchain, and the innovative integration of the two, namely BB-OpenFi. A comprehensive comparison of these paradigms is provided in Table 2.

2.1 Open banking

Open Banking represents a transformative shift in the financial services landscape, emphasizing transparency, democratization, and the responsible use of customer data [28, 29]. By enabling secure access to banking and financial information through APIs, Open Banking fosters innovation and competition. At its core, this paradigm

Table 1 The list of acronyms and notations

Acronym and Notation	Description
$\tilde{\sim}$	A computational indistinguishability
$\mathcal{A}$	A PPT adversary
$\mathcal{A}$	An access mechanisms to APIs
AML	Anti-Money Laundering
API	Application Programming Interface
AVISPA	Automated Validation of Internet Security Protocols and Applications
BB-OpenFi	Blockchain-based Open Finance
$\mathcal{C}$	A commitment
CeFi	Centralized Finance
CeDeFi	Centralized-Decentralized Finance
CT	Confidential Transaction
DA	Data Anonymity
DApp	Decentralized Application
DAO	Decentralized Autonomous Organization
$\mathcal{DB}$	A database
$\mathcal{Dec}$	A decryption function
DeFi	Decentralized Finance
DC	Data Confidentiality
DGK	Distributed Key Generation
DP	Data Protection
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EdDSA	Edwards-Curve Digital Signature Algorithm
$\mathcal{Enc}$	An encryption function
$\mathcal{F}$	An ideal computable functionality
FHE	Fully Homomorphic Encryption
FIDA	Financial Data Access
FSP	Financial Service Provider
GDPR	General Data Protection Regulation
$\mathcal{H}$	A random oracle (an ideal hash function)
HE	Homomorphic Encryption
HLPSP	High-Level Protocol Specification Language
ILP	Interledger Protocols
IOP	Interactive Oracle Proof
IND-CPA	Indistinguishability under chosen plaintext attack
λ	A security parameter
L1	Layer 1 of blockchain
L2	Layer 2 of blockchain
LWE	Learning With Errors
M	secret lattice matrix
MPC	Multi-Party Computation
NIZK	Non-Interactive Zero-Knowledge Proof
OFMC	On-the-Fly Model Checker
OpenFi	Open Finance
$\mathcal{P}$	Prover
$P_{baseline}$	The success probability under random guessing
PK	Public key
PKI	Public-Key Infrastructure
PoS	Proof-of-Stake
PoW	Proof-of-Work
PSD	Payment Services Directive
PQC	Post-Quantum Cryptography
$R(\cdot, \cdot)$	A relation function
RC	Regulatory Compliance
RO	Random Oracle
$\mathcal{S}$	A real-word simulator
SE	Searchable Encryption
SC	Smart Contract
SDP	Sensitive Data Protection
SK	Private key
SRS	Structured Reference String
SSI	Self-Sovereign Identity
SVP	Shortest Vector Problem
SWOT	Strengths - Weaknesses - Opportunities - Threats
T	Threshold of corruptions tolerated
$\mathcal{T}$	The set of transactions in a rollup mechanism
TSS	Threshold Signature Scheme
$\mathcal{U}$	The set of users
UC	Universal Composability
UP	User Privacy
$\mathcal{V}$	Verifier
$View_A$	All information seen by adversarial parties
YOSO	You Only Speak Once
w	A private witness
x	A public input or a plaintext/message
x_i	Private input of party P_i (the i -th party)
y	Output of the computation $f(x_1, x_2, \dots, x_n)$
zk-STARK	Zero-Knowledge Scalable Transparent Argument of Knowledge
zk-SNARK	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge
ZKP	Zero-Knowledge Proof

embodies the principles of OpenFi, open data, and global financial impact, driven by the FinTech sector.

2.1.1 Open finance

OpenFi extends the concept of Open Banking to a broader financial ecosystem, encompassing insurance, investments, pensions, and beyond [6,7]. Formally, let $\mathcal{U}$ be the set of users, and $\mathcal{F}$ the set of financial services, and $\mathcal{A}$ the access mechanisms to APIs. OpenFi ensures:

- $\forall u \in \mathcal{U}, \exists f \in \mathcal{F}$ such that $A(u, f)$ is governed by transparent policies.
- Democratic access via $P(f)$, where P represents platform-agnostic operations.

As a brief distinguish of OpenFi from Open banking, 1) OpenFi also includes non-banking services, while Open Banking is bank-centric, 2) OpenFi demands broader API standards, and 3) OpenFi enhances user empowerment through broader data control.

2.1.2 Centralized Finance (CeFi)

CeFi refers to traditional, bank-led financial systems [30]. While CeFi boasts reliability, scalability, and regulatory compliance, it suffers from concerns over trust, data misuse, and limited user autonomy [31,32].

2.1.3 Key escrow (in OpenFi and CeFi)

Key escrow mechanisms [33,34], such as $Enc(k, m) = c$ with a trusted k -recovery authority, enable key recovery but raise privacy concerns. For examples, 1) Public-key infrastructure (PKI) supports key escrow by default, and 2) MPC and zero-knowledge proofs avoid key escrow, ensuring user autonomy.

2.1.4 Security and privacy

In the OpenFi services, several security and privacy key aspects exist [31,32,35,36]; The most important ones are listed in the following.

- *Data Confidentiality (DC)*: Achieved via HE schemes [24,25].
- *User Privacy (UP)*: Enabled by zk-SNARKs [37,38] and MPC [23].
- *Regulatory Compliance (RC)*: Governed by GDPR [21,22], ensuring lawful access $\mathcal{L}(D)$.

Advanced cryptographic tools like zk-SNARK [37,38], zk-STARK [39,40], HE [24,25], and SE [26] (for details see Section 3) fortify OpenFi by addressing key challenges in user privacy, data protection, and compliance.

Table 2 The summary of addressed paradigms in Section 2

Item	OpenFi	Blockchain	A trade-off
Provider(s) of infrastructure/platform	Banks and financial organizations	Decentralized Autonomous Organizations (DAOs)	Government and leading blockchain companies
Emergence as.	CeFi (Centralized Finance)	DeFi (Decentralized Finance)	CeDeFi (Centralized-Decentralized Finance)
Key escrow aspect(s)	High reliance on key escrow for access recovery	Lacks key escrow, prioritizing user sovereignty	Hybrid, with optional escrow mechanisms
Expected security and privacy features	Transparent data sharing, compliance with regulations	Data confidentiality, anonymity, and decentralization	A balance between transparency and privacy
Provided security and privacy features	Encryption and basic access control mechanisms	Advanced cryptographic tools: zk-SNARK, MPC, and secret sharing	Integration of blockchain and CeFi security models
Important weakness(es) and challenge(s)	Centralized trust and potential abuse by intermediaries	Scalability and regulatory uncertainty	Complex integration and governance issues
Scalability	High due to centralized systems	Limited in current blockchain solutions	Intermediate, leveraging hybrid architectures
Throughput	High due to optimized infrastructures	Limited by consensus mechanisms	Balanced, with dynamic optimization
User control and sovereignty	Limited due to intermediaries	High due to decentralization	Conditional, depending on design
Implementation challenges	Regulatory adherence	Scalability and security optimization	Combining the best of both paradigms

2.2 Blockchain

Blockchain technology represents a paradigm shift in data integrity [41], transparency [42], and decentralized trust. Applications span financial services, insurance, healthcare, and data storage.

2.2.1 Decentralized Autonomous Organization (DAO)

A DAO is a blockchain-based entity governed by smart contracts (SC) with operations O defined as $O(SC) = \text{Autonomous} - \text{Governance}$ [43,44]. In this regard, advanced cryptographic tools can play critical roles. For example, secret-sharing protocols $s = s_1 + s_2 + \dots + s_n$ ensure distributed control [45,46], and in threshold cryptography, k -out-of- n signature is an exciting tool for crowd decision-making [48,49].

2.2.2 Decentralized Finance (DeFi)

DeFi platforms leverage blockchain to provide services like lending, borrowing, and asset management [11,12,13]. To enhance the DeFi application, cryptography can provide many benefits, such as making confidential Transactions (CT) via zk-SNARKs [37,38], supporting Data Anonymity (DA) via MPC schemes [23], and enabling Sensitive Data Protection (SDP) via HEs [24,25].

2.2.3 Key escrow (in DAO and DeFi)

Most blockchain-based services, including DeFi [13] and DAOs [50], inherently reject key escrow mechanisms. This is due to the decentralized nature of blockchain, which prioritizes user sovereignty and eliminates reliance on trusted third parties. Non-escrow properties of blockchains and the role of advanced cryptographic tools enable this user-friendly property.

2.2.4 Security and Privacy

Security and privacy in blockchain-based paradigms like DeFi and DAO are critical for user trust and adoption [51,52,53]. The three important ones are now listed below.

1. *Confidentiality*: Provided via zk-SNARKs [54] or zk-STARKs [55], ensuring transaction details remain hidden while maintaining verifiability $Proof \pi: \{0,1\} \leftarrow Verify(x,\pi)$.
2. *Anonymity*: Achieved through Mixers [56] or MPC [23], ensuring user identities cannot be linked to transactions.
3. *Integrity*: Guaranteed via blockchain's immutable ledger [57], defined as $H(L) = H(L_1 \| L_2 \| \dots \| L_n)$, where H is a random oracle (or an ideal cryptographic hash function).

2.3 Blockchain-based open finance (BB-OpenFi)

BB-OpenFi merges the transparency and accessibility of OpenFi with the decentralized and cryptographically secured infrastructure of blockchain. It offers a novel approach to democratizing global finance while addressing critical challenges in security, privacy, and scalability.

2.3.1 Entities and players

Below are the key entities within BB-OpenFi and their roles:

- *Blockchain*: Provides the foundational infrastructure for secure, decentralized operations.
- *SC*: Automated scripts ensuring tamper-proof and transparent execution of agreements.
- *DAO*: Acts as the governing body for decentralized decision-making.

- *DeFi*: Delivers blockchain-based financial services like lending, borrowing, and trading.
- *Miners/Validators*: Validate transactions and secure the network.
- *Service Owners and Developers*: Design and deploy DApps.
- *Users u_i* : Participate in financial activities and governance, benefiting from transparency and autonomy.
- *DApps*: Enable user interaction with blockchain-based services.
- *Regulatory Nodes*: Facilitate compliance with legal frameworks while preserving decentralization.

2.3.2 Applications

As a newly-proposed paradigm, the BB-OpenFi framework can provide many state-of-the-art applications and enhance DeFi applications. Four important with the most famous projects (not limited to) are listed in the following.

- *Lending and Borrowing*: Platforms like Aave⁷ utilize DeFi to connect lenders and borrowers seamlessly.
- *Insurance*: Nexus Mutual⁸ provides decentralized insurance solutions.
- *Healthcare*: MediBloc⁹ leverages blockchain for secure patient data sharing.
- *Data Storage*: Filecoin¹⁰ offers decentralized and secure data storage solutions.

2.3.3 Expected features

The BB-OpenFi framework is expected to meet advanced security and performance requirements. Below is a formal definition of its key features:

- *Data Protection (DP)*: Ensures sensitive data remains secure using HE schemes.
- *UP*: Enabled via zk-SNARKs, ensuring private yet verifiable transactions.
- *CT*: Uses zk-STARKs to keep transaction details hidden.
- *Scalability*: Achieved via Layer 2 (L2) solutions like rollups.
- *Throughput*: Enhanced using sharding mechanisms.
- *Interoperability*: Facilitates seamless integration with traditional finance.

Regarding several advanced cryptographic tools outlined in Section 3, features identified in Table 3 are also expected from the BB-OpenFi framework.

Table 3 Advanced cryptographic tools to improve BB-OpenFi

Tool	Purpose	Enhancement via PQC	Enhancement via zk-STARK
zk-SNARKs	Confidential Transactions	Quantum-safe proof variants	Transparent setup and scalability
Homomorphic Encryption	Secure Computations	Lattice-based HE for efficiency	Verifiable and scalable evaluations
Threshold Cryptography	Shared Control	Quantum-safe secret-sharing	Fault-tolerant decentralized security

Table 4 A comparison between OpenFi and BB-OpenFi

Feature	OpenFi	BB-OpenFi
Service Provider Type	Banks/FSIs	Blockchain/DAOs
Preliminaries	Centralized infrastructure	Decentralized blockchain
Advanced Tools	Basic encryption, APIs	zk-SNARKs, MPC, HE
Expected Features	Transparency, data sharing	Privacy, scalability, anonymity
Implementation Challenges	Data centralization risks	Scalability, regulatory compliance
Highlighted Benefits	Easy adoption, trust in FSIs	User sovereignty, secure transactions
Government Access	Mandatory	Optional, based on design
User Control	Limited	High

2.3.4 Comparison (OpenFi v.s. BB-OpenFi)

This section provides a brief comparison between two innovative services, OpenFi and BB-OpenFi. Generally, OpenFi is considered as bank-in-the-background provided by FSIs as App-based or Web-based services; On the other hand, the BB-OpenFi framework services are assumed as blockchain-based paradigms provided by DeFi developers as DApp-based or Web3-based services. This comparison is shown in Table 4

2.4 A good trade-off

A balanced approach combining the strengths of OpenFi and BB-OpenFi can address limitations inherent to both paradigms. This trade-off, termed Centralized-Decentralized Finance¹¹ (CeDeFi) [13,58], represents a hybrid model enabling scalability, compliance, and decentralization.

- *Government and leading companies*: Governments and leading firms like Chainalysis¹² and ConsenSys¹³ play pivotal roles in CeDeFi, facilitating regulatory frameworks and interoperability standards.
- *CeDeFi*: CeDeFi bridges centralized trust with decentralized innovation, leveraging advanced cryptographic tools such as MPC and zk-SNARKs to ensure both compliance and privacy.
- *Key escrow*: CeDeFi supports optional key escrow mechanisms for regulated services while preserving user sovereignty for non-regulated ones.

⁷ <https://aave.com/>

⁸ <https://nexusmutual.io/>

⁹ <https://medibloc.com/>

¹⁰ <https://filecoin.io/>

¹¹ <https://www.cedefi.org/>

¹² <https://www.chainalysis.com/>

¹³ <https://consensys.io/>

- *Security and Privacy*: Advanced tools ensure confidentiality (HE and ZKPs) and integrity (using cryptographic hash functions) while addressing compliance with frameworks like GDPR.

3 Advanced cryptographic tools/schemes

This section systematically provides formal definitions, protocol details, and security features of important cryptographic schemes that are more popular to apply in the BB-OpenFi framework background.

3.1 Multi-party computing

An MPC protocol allows n parties $P_1, P_2, \dots, P_n$ to compute a function $f(x_1, x_2, \dots, x_n)$ over their private inputs $x_1, x_2, \dots, x_n$ such that no party learns anything about other inputs beyond what is implied by $f(x_1, x_2, \dots, x_n)$ [23].

Each MPC scheme involves three main algorithms in addition to the *Setup* and *KeyGen*.

1. *Input Sharing* ($\{s_{i,1}, \dots, s_{i,n}\} \leftarrow \text{Share}(x_i)$): Each party P_i secret-shares x_i using a secret sharing scheme $\text{Share}(x_i) = \{s_{i,1}, \dots, s_{i,n}\}$, where $s_{i,j}$ is the share sent to P_j .
2. *Distributed Computation*: Parties collaboratively compute the shares of y without reconstructing intermediate secrets. This is achieved using arithmetic on the shared values $f(x_1, x_2, \dots, x_n) = g(s_{1,1}, \dots, s_{n,n})$, where g is the function represented as a circuit.
3. *Reconstruction* ($y \leftarrow \text{Reconstruct}(\{y_1, \dots, y_n\})$): Each party P_1 sends its output share y_i to all others, and the final result y is reconstructed using $\text{Reconstruct}(\{y_1, \dots, y_n\}) = y$.

The security features of MPC schemes are formalized as follows.

- *Correctness*: An MPC protocol is correct if the output of the protocol is equal to the value of the function $f(x_1, x_2, \dots, x_n)$ computed on the private inputs of the parties. Formally $\forall x_1, x_2, \dots, x_n, \Pr[f(x_1, x_2, \dots, x_n) = \text{Reconstruct}(y)] = 1$, where y is the result of the computation on shared inputs, and the probability is over the randomness of the protocol.
- *Privacy*: Privacy guarantees that no subset of adversarial parties P_A (of size $|A| \leq \tau$) can learn anything about honest parties' inputs beyond what is inferable from the output. Formally, there exists a simulator $\mathcal{S}$ such that $\text{View}_A(P_A, x_A, y) \stackrel{c}{\approx} \mathcal{A}(x_A, y)$, where $\stackrel{c}{\approx}$ denotes computational indistinguishability, and View_A includes all information seen by adversarial parties.

- *Robustness*: Robustness ensures that the protocol functions correctly even if up to T parties act maliciously. This is achieved by verifying shares via error-detection techniques such as polynomial consistency checks $\Pr[\text{Reconstruct fails} \mid |A| \leq T] = 0$.

If the protocol relies on cryptographic primitives modeled in ROM (e.g., hash functions), the adversary's view includes oracle queries. Privacy is guaranteed as long as the adversary cannot invert or find collisions in the oracle.

3.2 Threshold cryptography

A threshold cryptosystem enables a group of n parties to jointly perform cryptographic operations (e.g., encryption or signing) such that a threshold $T < n$ of parties is required to complete the operation and less than T parties learn no information about the secret [48, 49, 59, 60]. As a generic example, a threshold signature scheme is designed through the four below algorithms (SK is a secret key distributed across n parties as shares $\{sk_1, \dots, sk_n\}$, and PK is the public key known to all parties).

1. *Setup* ($\{sk_1, \dots, sk_n\} \leftarrow \text{Share}(SK)$): A dealer distributes shares of SK to all n parties.
2. *Sign* ($\sigma_i \leftarrow \text{Sign}(sk_i, m)$): A subset of at least T parties computes a partial signature $\sigma_i \text{Sign}(sk_i, m)$.
3. *Aggregation* ($\sigma \leftarrow \text{Combine}(\{\sigma_1, \dots, \sigma_n\})$): Partial signatures are combined into a full signature.
4. *Verification* ($\{0, 1\} \leftarrow \text{Verify}(\sigma_i, PK)$): The signature σ is verified using PK .

Regarding the threshold signature scheme discussed above, the security features are generally defined in the following.

- *Unforgeability*: A threshold cryptographic scheme is unforgeable if no $\mathcal{A}$ controlling up to $T - 1$ parties can produce a valid cryptographic output (e.g., signature or ciphertext) without collaboration from honest parties. Formally $\Pr[\text{Forge}(PK) = 1 \mid |A| < T] \leq \text{negl}(\lambda)$, where $\text{negl}(\lambda)$ is a negligible function in the security parameter λ .
- *Correctness*: For any message m , if at least T parties generate valid shares, the protocol correctly combines these shares to produce the cryptographic result R . Formally $\Pr[\text{Verify}(PK, R, m) = 1] = 1$.
- *Privacy*: Privacy guarantees that no $\mathcal{A}$ controlling less than T parties learns anything about SK where $\{sk_1, \dots, sk_n\} \leftarrow \text{Share}(SK)$ so that $|A| < T$. Hence, $\text{View}_A(P_A)$ is computationally indistinguishable from a uniformly random distribution $\text{View}_A \stackrel{c}{\approx} \text{Uniform}$.

Threshold schemes in ROM derive security from the collision resistance and pseudo-randomness of the oracle. Simulation is possible by answering oracle queries consistently.

3.3 Digital Signatures

Digital signatures are a fundamental cryptographic primitive that ensures data authenticity, integrity, and non-repudiation in financial and blockchain-based applications [61, 62]. In the BB-OpenFi framework, digital signatures are critical for validating transactions, enforcing access control, and securing communications.

In addition to the *Setup* algorithm, a generic digital signature scheme consists of the three following algorithms:

1. *Key Generation* $((PK, SK) \leftarrow \text{KeyGen}(\lambda, \text{UserID}))$: A user (or a certificate authority) generates the key pair (PK, SK) , where PK is the public key and SK is the private key.
2. *Signing* $(\sigma \leftarrow \text{Sign}(m, SK))$: Given a message m , the signer computes a signature σ using its private key.
3. *Verification* $(\{0, 1\} \leftarrow \text{Verify}(PK, \sigma, m))$: The verifier checks the validity of σ using m and the public key PK . The *Verify* algorithm returns 1 if σ is valid; Else, it returns 0.

A secure digital signature scheme satisfies the following properties:

- *Unforgeability*: The $\mathcal{A}$ with access to a signing oracle cannot generate a valid signature on a previously unseen message m with a non-negligible probability. The advantage of $\mathcal{A}$ is formalized as $\text{ADV}_{\mathcal{A}}^{\text{Forge}} = \Pr[\mathcal{A}(PK, \text{Sign}(SK, \cdot)) \rightarrow (m, \sigma) \mid \text{Verify}(PK, \sigma, m) = 1] < \text{negl}(\lambda)$.
- *Non-repudiation*: A signer cannot deny having signed a message because only they possess SK .
- *Integrity*: Any modification to m invalidates σ , preventing tampering.

Several digital signature schemes exist, each with different efficiency and security guarantees. Several important ones are now listed below:

- *RSA Signatures* [63]: It designed based on integer factorization hardness, where signing and verification involve modular exponentiation.
- *ECDSA (Elliptic Curve Digital Signature Algorithm)* [64]: The ECDSA signature algorithm provides stronger security per key length compared to RSA, making it efficient for blockchain applications.

- *EdDSA (Edwards-Curve Digital Signature Algorithm)* [65]: This algorithm has been optimized for speed and security, used in privacy-focused applications like Monero¹⁴.

In the BB-OpenFi framework, digital signatures authenticate transactions, secure multi-party computations, and validate smart contract executions. We believe that the combination of threshold signatures with the BB-OpenFi framework ensures decentralized control, preventing single points of failure.

3.4 Homomorphic encryption

Homomorphic encryption allows computation on encrypted data such that $\text{Enc}(f(m_1, m_2)) = f(\text{Enc}(m_1), \text{Enc}(m_2))$, where Enc is the encryption function and f is a computable function [24, 25]. The four below algorithms define a generic HE scheme, where SK and PK are secret and public keys and m is considered as a plain message.

1. *Key Generation* $((PK, SK) \leftarrow \text{KeyGen}(1^\lambda))$: According to the security parameter λ , which determines the security level of the encryption, this algorithm generates the public and secret keys (PK, SK) needed for the encryption and decryption processes.
2. *Encryption* $(c \leftarrow \text{Enc}(PK, x))$: Using PK and x as an input message, the Enc algorithm encrypts the plain message x and returns c .
3. *Homomorphic Evaluation* $(c_f \leftarrow \text{Eval}(PK, f, c_1, c_2, \dots))$: On access to PK and computable function f , which could be arithmetic, logical, etc., the Eval algorithm allows the computation of a function f over encrypted data. The f is applied to encrypted inputs (ciphertexts) $c_1, c_2, \dots$ resulting in an encrypted output c_f .
4. *Decryption* $(m_f \leftarrow \text{Dec}(SK, c_f))$: This algorithm decrypts the ciphertext c_f , which represents the encrypted result of applying f on the encrypted inputs using SK .

Homomorphic encryption thus supports secure computations on encrypted data, allowing privacy-preserving operations without revealing the underlying sensitive information.

HE schemes are considered secure if the following features provided properly.

- *Correctness*: For any plaintexts m_1, m_2 , and a function f the scheme guarantees that decrypting the evaluated ciphertext equals the plaintext result of $f(m_1, m_2)$.

¹⁴ <https://www.getmonero.org/>

Formally, $Pr[Dec(SK, Eval(PK, f, Enc(PK, m_1), Enc(PK, m_2))) = f(m_1, m_2)] = 1$.

- *CPA Security*: Homomorphic encryption is indistinguishability under chosen plaintext attack (IND-CPA) secure if no $\mathcal{A}$ can distinguish encryptions of two chosen plaintexts m_0 and m_1 .

Formally, $Pr[\mathcal{A}(Enc(PK, m_b)) = b] \leq \frac{1}{2} + negl(\lambda)$, where $b \in \{0, 1\}$ is chosen uniformly at random.

- *Homomorphism*: Security ensures that computation on ciphertexts reveals no additional information about the plaintexts.

The $View_{\mathcal{A}} = \{c_1, c_2, Eval(PK, f, c_1, c_2)\}$, is indistinguishable from the simulator's view, which produces random ciphertexts without knowledge of the plaintexts.

The scheme's security relies on the pseudo-randomness of ciphertexts and the hardness of lattice-based or number-theoretic assumptions.

3.5 Secret sharing protocols

A secret s is divided into n shares such that any $\tau \leq n$ shares can reconstruct s and fewer than τ shares reveal nothing about s [45, 46, 47]. Each secret sharing protocols involves two below steps.

1. *Sharing*: Generate random coefficients $a_1, a_2, \dots, a_{\tau-1}$ and compute $s_i = f(i) = s + a_1i + a_2i^2 + \dots + a_{\tau-1}i^{\tau-1}$.
2. *Reconstruction*: Use τ shares to solve for s using Lagrange interpolation.

Each secret-sharing protocol should be supported by the following security properties to be considered secure.

- *Correctness*: Any T -sized subset of shares can reconstruct the secret s . Let $\{s_1, s_2, \dots, s_n\} \leftarrow Share(s)$. Then, for any $S \subset \{1, 2, \dots, n\}, |S| = T$: $Reconstruct(S) = s$.
- *Privacy*: For any $S \subset \{1, 2, \dots, n\}, |S| < T$, the shares reveal no information about s . Formally $Pr[s|S] = Pr[s]$, meaning the distribution of s is independent of the adversary's view of the shares.
- *Robustness*: The scheme ensures that malicious parties providing invalid shares do not disrupt reconstruction. Verification ensures correctness of shares using consistency checks.

The privacy of shares is guaranteed by pseudo-randomness derived from a random oracle. Adversarial attempts to guess s are infeasible due to polynomial hardness.

3.6 Searchable encryption

Searchable encryption enables search over encrypted data. For keyword w , the protocol computes $Trapdoor(w) \rightarrow Search(Enc(DB), Trapdoor(w)) \rightarrow Enc(Result)$, where DB is a database that should be encrypted [26, 67, 68, 69]. According to this notation, the SE schemes are defined in three main algorithms.

1. *Setup* ($\{Enc(d_1), Enc(d_2), \dots, Enc(d_n) \leftarrow Enc(DB)\}$): This algorithm takes as input a database $DB = \{d_1, d_2, \dots, d_n\}$ consisting of n plaintext documents. It outputs the encrypted database $Enc(DB)$, where each document d_i is individually encrypted as $Enc(d_i)$. This ensures that the data remains secure while enabling the search functionality.
2. *Trapdoor Generation* ($td_w \leftarrow Trapdoor(PK, w)$): This algorithm generates a trapdoor for a specific keyword w using a public key PK . The trapdoor td_w serves as a search token that allows the querying entity to perform a search for w within the encrypted database $Enc(DB)$ without revealing the actual keyword or other sensitive information.
3. *Search* ($Result \leftarrow Search(Enc(DB), td_w)$): This algorithm takes the encrypted database $Enc(DB)$ and the trapdoor td_w as input. It performs a search operation to locate all encrypted documents containing the keyword w . The output $Result$ is typically an encrypted representation of the search results, preserving the privacy of the documents and the query.

Each SE scheme should provide security features which query security and IND-CKA security are the most important ones.

- *Correctness*: For any keyword w and encrypted database $Enc(DB)$, the protocol guarantees that searching with trapdoor td_w retrieves all matching documents d_i . Formally, $Pr[Search(Enc(DB), td_w) = Enc(\{d_i | w \in d_i\})] = 1$
- *Query Privacy*: Query privacy ensures that the trapdoor td_w leaks no information about the keyword w beyond search results. Formally, there exists a simulator $\mathcal{S}$ such that $View_{\mathcal{A}}(td_w) \stackrel{c}{\approx} \mathcal{S}(\{Enc(d_i) | w \in d_i\})$.
- *IND-CKA Security*: The scheme is secure against chosen keyword attacks if no adversary can distinguish trapdoors of two chosen keywords w_0 and w_1 : $Pr[\mathcal{A}(td_{w_b}) = b] \leq \frac{1}{2} + negl(\lambda)$.

Query privacy relies on the pseudo-randomness of trapdoors and encrypted database elements. The $\mathcal{A}$'s inability to distinguish trapdoors reduces to oracle collision-resistance assumptions.

3.7 Post-Quantum Cryptography

The advent of quantum computing presents a fundamental challenge to classical cryptographic schemes, necessitating the development and adoption of post-quantum cryptographic (PQC) mechanisms [70]. Traditional asymmetric cryptographic primitives, such as RSA [63] and Elliptic Curve Cryptography (ECC) [64,65], rely on hardness assumptions like integer factorization and the discrete logarithm problem. These assumptions become ineffective in the presence of Shor's algorithm, which enables polynomial-time solutions to these problems on a sufficiently large quantum computer. To ensure the long-term security of the BB-OpenFi framework, cryptographic protocols must be transitioned to quantum-resistant alternatives that rely on computational problems conjectured to be hard even for quantum adversaries.

The following discusses several mathematical structures that serve as the foundation for post-quantum cryptographic schemes.

1. *Lattice-based cryptography*: Lattice-based cryptography, one of the most promising approaches, builds security on problems such as the Learning With Errors (LWE) problem and the Shortest Vector Problem (SVP), which are believed to be intractable even for quantum computers [71,72]. Formalized as follows, the LWE problem assumes that given a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a noise vector $\mathbf{e} \in \mathbb{Z}_q^m$, recovering $\mathbf{s}$ from $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$ is computationally infeasible. This problem forms the basis of schemes such as Kyber for public-key encryption and Dilithium for digital signatures [73].
2. *Code-based cryptography*: Another approach is code-based cryptography, which derives security from the hardness of decoding random linear codes [74,75]. The McEliece cryptosystem, an example of such an approach, employs error-correcting codes and has remained secure for decades despite extensive cryptanalysis.
3. *Multivariate polynomial cryptography*: This type of calculation constructs cryptosystems based on solving nonlinear systems of polynomial equations over finite fields, where security arises from the NP-hard nature of solving these equations [76]. An example is the Rainbow signature scheme, which has been proposed as a quantum-resistant alternative to traditional digital signatures.
4. *Hash-based cryptography*: Hash-based cryptography provides an alternative approach by leveraging the security of cryptographic hash functions [77,78]. Unlike number-theoretic cryptographic primitives, hash functions remain resistant to quantum attacks be-

yond Grover's search algorithm, which only provides a quadratic speedup for brute-force attacks. One example is the SPHINCS+ signature scheme [79], which constructs secure signatures using Merkle trees and one-time signatures, ensuring quantum resistance without relying on computational number-theoretic assumptions.

To integrate post-quantum cryptography into the BB-OpenFi framework, several modifications to existing cryptographic protocols are required.

- Traditional digital signature schemes, such as ECDSA [64], must be replaced with quantum-resistant alternatives like Dilithium [73] or SPHINCS+ [79].
- The adoption of lattice-based zero-knowledge proofs, particularly in zk-STARK constructions [39,40], ensures verifiable and scalable cryptographic proofs while maintaining post-quantum security.
- Hybrid cryptographic frameworks that employ both classical and quantum-resistant schemes in parallel provide a viable transition strategy, allowing interoperability between legacy financial systems and future quantum-secure infrastructures.
- Furthermore, threshold cryptographic mechanisms must be adapted to post-quantum settings by employing lattice-based secret-sharing and signature schemes to maintain the integrity of multi-party financial transactions.

Given that post-quantum cryptographic schemes generally incur greater computational and communication overhead compared to their classical counterparts, optimizing their implementation within the BB-OpenFi framework is crucial. Lattice-based cryptographic primitives, while computationally intensive due to polynomial multiplications in high-dimensional spaces, offer efficient key generation and verification times. In contrast, hash-based signatures such as SPHINCS+ [79] introduce large signature sizes, which can impact storage and bandwidth efficiency. A comparative evaluation of these post-quantum schemes in the context of the BB-OpenFi framework requires a trade-off analysis between security, efficiency, and scalability.

To future-proof the BB-OpenFi framework against quantum adversaries, our framework integrates lattice-based encryption schemes such as NTRUEncrypt [80] and Kyber [81], alongside hash-based signatures like SPHINCS+ [79]. These cryptosystems are resilient against Shor's and Grover's algorithms, ensuring security in a quantum era [82,83]. Specifically, we propose a hybrid approach that utilizes lattice-based zero-knowledge proofs (ZKPs) [84] within zk-STARK constructions to maintain verifiable and scalable cryptographic proofs.

Formally, let $P(x) = \sum_{i=0}^n a_i x^i$, where $P(x)$ represents a polynomial evaluated for zero-knowledge proof commitments. The proof system satisfies transparency and computational efficiency by leveraging Merkle proof structures $C = h(P(x)) + r$. Implementing these quantum-resistant primitives in the BB-OpenFi framework mitigates the risks of future quantum attacks and enhances overall system resilience.

3.8 Zero-knowledge proof

This section addresses a well-known but advanced cryptographic tool called ZKP. This tool involves several subgroups that bring different features to apply in the blockchain-based systems. In this regard, five important subcategories of ZKPs are addressed below, and their applicability in the BB-OpenFi framework is discussed. To complete this review, Tables 5 and 6 provide a comprehensive comparison between the five subcategories of ZKPs and SWOT (Strengths - Weaknesses - Opportunities - Threats) analysis for using each of them in the background of the BB-OpenFi framework.

3.8.1 Generic ZKP

A generic ZKP system is defined as a tuple $(\mathcal{P}, \mathcal{V})$ where $\mathcal{P}$ (Prover) convinces $\mathcal{V}$ (Verifier) of the truth of a statement without revealing any information beyond its validity [27].

- *Input*: Public input x , private witness w , and a relation function $R(x, w)$ so that $R(x, w) = 1$ if w is a valid witness for x .
- *Protocol steps*:
 1. *Commitment* ($\mathcal{C} \leftarrow \text{Comm}(r)$): The $\mathcal{P}$ generates a commitment $\mathcal{C}$ using a random number r .
 2. *Challenge* ($\mathcal{V} \xrightarrow{e} \mathcal{P}$): The $\mathcal{V}$ sends a random e .
 3. *Response* ($\mathcal{P} \xrightarrow{z=\mathcal{F}(x,w,r,e)} \mathcal{V}$): The $\mathcal{P}$ responds with z , computed from x , w , r , and e .
 4. *Verification* ($\{0, 1\} \leftarrow \text{Verify}(z, (x, c, e))$): The $\mathcal{V}$ checks z using x , c , and e .
- *Output*: The $\mathcal{V}$ accepts (1) or rejects (0) the proof.

Each generic ZKP scheme provide three main security features as follows.

1. *Completeness*: Every honest $\mathcal{P}$ always convinces $\mathcal{V}$ if $R(x, w) = 1$.
2. *Soundness*: No malicious $\mathcal{P}^*$ can convince $\mathcal{V}$ with non-negligible probability if $R(x, w) = 0$.
3. *Zero-Knowledge*: The $\mathcal{P}$'s responses leak no information about w . Formally, there exists a simulator $\mathcal{S}$ that can produce indistinguishable transcripts without access to w .

Generic ZKP systems provide a fundamental building block but are often computationally expensive. They are typically not practical for blockchain applications without optimizations such as non-interactive protocols.

3.8.2 NIZK

NIZK transforms a ZKP into a non-interactive form using the Fiat-Shamir heuristic in the random oracle model [85].

- *Input*: Similar to the generic ZKP; x , w , and $R(x, w)$.
- *Protocol steps*: In the NIZK schemes, two main steps should be passed.
 1. $\pi \leftarrow \text{NIZK.Prove}(x, w)$: The $\mathcal{P}$ generates a proof $\pi = \text{NIZK.Prove}(x, w)$ using a random oracle H .
 2. $\{0, 1\} \leftarrow \text{NIZK.Verify}(x, w)$: The $\mathcal{P}$ checks π using $\text{NIZK.Verify}(x, w)$.
- *Output*: The $\mathcal{V}$ accepts (1) or rejects (0) the proof.

Security features of NIZK schemes are as follows.

1. *Soundness*: Secure against forgery under the random oracle model.
2. *Zero-Knowledge*: The simulator $\mathcal{S}$ exists, producing π^* indistinguishable from valid proofs.

Highly suitable due to the non-interactive nature, reducing communication overhead. Widely used in smart contract verifications and decentralized identity schemes.

3.8.3 zk-SNARK

Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK) involves [37, 38]:

- *Algorithms*: In zk-SNARK schemes three algorithms *Setup*, *Prove*, and *Verify* are defined.
- *Input*: x , w , and $R(x, w)$.
- *Protocol steps*: Three below steps exist in zk-SNARK schemes.
 1. *Setup* ($(pk, vk) \leftarrow \text{Setup}(1^\lambda, R)$): Generate proving and verification keys $((pk, vk))$, using $\text{Setup}(1^\lambda, R)$.
 2. *Proof Generation* ($\pi \leftarrow \text{Prove}(pk, x, w)$): Compute proof $\pi = \text{Prove}(pk, x, w)$.
 3. *Verification* ($\{0, 1\} \leftarrow \text{Verify}(vk, x, \pi)$): Check validity with $\text{Verify}(vk, x, \pi)$.

Important features of zk-SNARK schemes are listed in the following.

1. *Succinctness*: Proof size and verification time are logarithmic in x .
2. *Completeness and Soundness*: Same as NIZK, with structured reference string (SRS) assumptions.

Table 5 A Comparison between the subcategories of ZKPs

Feature	Generic ZKP	NIZK	zk-SNARK	zk-STARK	MPC-in-the-Head
Computational Cost	High	Moderate	Low	Moderate	Moderate
Communication Overhead	High	Low	Low	Moderate	Low
Complexity	Low	Moderate	High	High	Moderate
Limitation	Interactive	Trusted Setup	Trusted Setup	Hash-based	High Computation
Security Highlight	Generality	Non-interactivity	Succinctness	Transparency	Efficiency
Blockchain Usability	Limited	Good	Excellent	Excellent	Moderate
Completeness Level	High	High	High	High	High
Soundness Level	High	High	High	High	High
Perfect Soundness	Moderate	High	High	High	Moderate
Post-Quantum Security	No	No	No	Yes	Yes

Table 6 The SWOT analysis of subcategories of ZKPs if used in the background of BB-OpenFi

	Strengths	Weaknesses	Opportunities	Threats
Generic ZKP	Foundational, flexible	High overhead, interaction required	Fundamental building block for other protocols	Inefficiency in practical applications
NIZK	Non-interactive, efficient for blockchain	Random oracle model reliance	Smart contracts, token verifications	Vulnerabilities in random oracles
zk-SNARK	Succinctness, efficient verification	Trusted setup requirement	Scalable OpenFi	Dependence on SRS
zk-STARK	Transparent setup, post-quantum security	Larger proof size	High-throughput systems	Complexity in implementation
MPC-in-the-Head	Efficiency in privacy-preserving tasks	Computationally intensive	Secure multiparty systems	Less suitable for lightweight blockchain tasks

3. *Zero-Knowledge*: Achieved via cryptographic bilinear pairings.

zk-SNARKs are highly efficient, making them ideal for scalability in OpenFi applications. However, reliance on trusted setup is a potential drawback.

3.8.4 zk-STARK

Zero-Knowledge Scalable Transparent Argument of Knowledge (zk-STARK) [39,40] replaces SRS with transparent setup [86,87]

- *Input*: $x, w, R(x, w)$.
- *Protocol steps*: Similar to previous, three below steps should be passed.
 1. *Setup*: Transparent randomness via public randomness beacons.
 2. *Prove*: Generate proof using polynomial IOPs (Interactive Oracle Proofs).
 3. *Verify*: Efficiently verify the proof using Merkle commitments.

The zk-STARK schemes provide different security features outlined in the following.

1. *Post-Quantum Security*: Uses cryptographic hash functions instead of pairings. For scalability and quantum resistance, zk-STARK formulas are derived from polynomials with commitments like $C_i = H(f(x_i)) + r$.
2. *Transparent Setup*: No trusted third-party assumptions.
3. *Scalability*: Supports large computations.

zk-STARKs are advantageous for transparency and scalability, suitable for DeFi with high throughput requirements.

3.8.5 MPC-in-the-head

MPC-in-the-Head combines secure MPC with ZKP [88, 89,90]

- *Input*: $x, w, R(x, w)$.
- *Protocol steps*: Three steps of MPC-in-the-head schemes are as follows.
 1. Simulate an MPC protocol within the $\mathcal{P}$'s head.
 2. Generate commitments and responses for virtual MPC participants.
 3. $\mathcal{V}$ checks consistency and correctness.

Security features of MPC-in-the-head schemes are in the following.

1. *Efficiency*: Reduces communication costs compared to traditional MPC.
2. *Zero-Knowledge*: Achieved through masking inputs and outputs.

Good for complex privacy-preserving computations but less practical for lightweight applications due to higher computational overhead.

3.8.6 PQC and Quantum-Safe zk-STARKs

The PQC is critical for protecting the BB-OpenFi framework against quantum adversaries. Lattice-based schemes, such as NTRUEncrypt, provide efficient public-key encryption, and hash-based signatures like XMSS ensure tamper-resistant records [61,62]. Lattice-based encryption can be modeled as $H(x) = g(x.M) \bmod p$, where M is a secret lattice matrix. PQC's compatibility with zk-STARKs enables scalable, transparent, and quantum-secure proofs.

zk-STARKs not only eliminate the trusted setup but also exhibit strong quantum resistance due to their reliance on hash-based cryptographic primitives. The proof generation process, formalized using polynomials, is as $P(x) = \sum_{i=0}^n a_i \cdot x^i$, where $P(x)$ represents a polynomial evaluated for zero-knowledge proof commitments. Hash-based Merkle proofs reinforce the transparency and scalability of zk-STARK implementations.

zk-STARKs can be employed for high-throughput transaction proofs without relying on trusted setups. Formal Representation of zk-STARK Proof is $P(x) = \sum_{i=0}^n a_i \cdot x^i \Rightarrow c = H(P(x)) + r$.

3.9 Analyzer models for proving the security of cryptographic schemes/protocols

Cryptographic protocols are central to modern security mechanisms, necessitating rigorous analysis to ensure their robustness under diverse adversarial settings. Over the decades, the cryptographic community has developed a variety of heuristic, formal, and simulation-based methods to analyze these protocols, each catering to different dimensions of security. This article delves deeply into these methods, offering precise definitions, mathematical rigor, and an exhaustive discussion of their applicability and limitations, ensuring a comprehensive perspective suitable for advanced research audiences. A perspective of this section is provided in Table 7.

3.9.1 Heuristic (non-provable category)

The game-based security paradigm formalizes the security of cryptographic schemes through adversarial interaction within structured games [91]. A scheme is secure if $\mathcal{A}$ cannot achieve a specific goal, quantified as an advantage, over a negligible threshold in computational terms.

To define in a mathematical framework, let $\mathcal{S}$ represent a cryptographic scheme, defined by a tuple $(Setup, KeyGen, Encrypt, Decrypt)$. Based on the four-algorithm tuple, a general game is outlined in the four following steps.

1. The challenger initializes the system using $Setup()$ and shares public parameters with $\mathcal{A}$.
2. The adversary queries the system (e.g., encryption oracles) to gather information.
3. The challenger issues a challenge (e.g., ciphertext of a chosen plaintext).
4. The adversary responds, attempting to achieve its goal.

In the generic game defined above, the advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}}^{Game} = |Pr[\mathcal{A} \text{ succeeds}] - P_{baseline}|$, where $P_{baseline}$ is the success probability under random guessing.

3.9.2 Model-based and formal methods

As many formal models proposed to analyze security protocols/schemes, important ones are addressed in the following.

- *BAN logic* [92]: Burrows, Abadi, and Needham introduced BAN logic, a formal logic framework for analyzing authentication protocols. It focuses on communicating parties' beliefs regarding the authenticity and freshness of messages.

These four notations construct BAN-based proofs: (i) $P| \equiv X$ means principal P believes X , (ii) $P| \Rightarrow X$ means P sees X , (iii) $P| \triangleright X$ means P has jurisdiction over X , and (iv) $X \#$ means X is fresh.

Additionally, the two below key inference rules are defined in the BAN logic framework.

1. *Message Meaning Rule*: If $P| \Rightarrow \{X\}_K$ and $P| \equiv K \longleftrightarrow Q$, then $P| \equiv Q| \equiv X$.
2. *Freshness Rule*: If $P| \equiv X \#$ then P assumes X was not replayed.

- *Random Oracle (RO) model* [93]: The RO model abstracts cryptographic hash functions as a theoretical entity, the random oracle H , which provides random and consistent outputs for each input. It bridges the gap between idealized proofs and real-world hash functions.

Two formal structures exist in this model. (1) An ideal hash function which is defined as $H : \{0, 1\}^* \rightarrow \{0, 1\}^n$, and (2) a random oracle queries $H(x)$ which returns uniformly random $y \in \{0, 1\}^n$ such that $H(x_i) = H(x_j)$ iff $x_i = x_j$.

The security condition of the RO model says that if $\mathcal{A}$ interacts with H , its advantage is calculated as $Adv_{\mathcal{A}}^{RO} = |Pr[A(H) \Rightarrow 1] - Pr[Baseline]|$.

- *Universal Composability (UC) framework* [94]: The UC framework ensures that a cryptographic protocol retains its security properties even when composed with arbitrary other protocols in a shared environment.

This model includes two main assumptions, (1) Real-world protocol π interacts with $\mathcal{A}$ and an environment $\mathcal{Z}$, and (2) Ideal-world functionality $\mathcal{F}$ interacts with a simulator $\mathcal{S}$ and $\mathcal{Z}$.

For the security assumption, the π UC-realizes $\mathcal{F}$ if $\forall A, \exists \mathcal{S}: Pr[Real_{\pi, \mathcal{A}, \mathcal{Z}} = 1] = Pr[Ideal_{\pi, \mathcal{A}, \mathcal{Z}} = 1]$.

Table 7 A perspective of security protocol/scheme analyser models/tools

Framework/Tool	Model Type	Strengths	Limitations
Game-Based	Heuristic	Intuitive, widely applicable	Non-compositional, informal
BAN Logic	Formal	Simplicity in reasoning	Limited expressiveness
RO Model	Formal	Facilitates practical designs	Unrealistic abstraction
UC Framework	Formal	Compositional security guarantees	Complex formalism
AVISPA	Simulation	Automated, scalable	Limited protocol coverage
ProVerif	Simulation	Symbolic and scalable	Over-approximations
CryptoVerif	Simulation	Computational rigor	Higher complexity
TAMARIN	Simulation	Comprehensive symbolic-computational model	High expertise required

3.9.3 Simulation-based

Simulation-based tools provide automated, rigorous, and often formal verification methods for analyzing cryptographic protocols. These tools are rooted in well-defined theoretical models, including symbolic representations (e.g., the Dolev-Yao model [95]) and computational settings. They allow researchers to simulate adversarial behaviors, test security properties, and analyze both simple and complex cryptographic protocols under various attack scenarios.

Below is a detailed exploration of four widely used simulation-based tools—AVISPA¹⁵, ProVerif¹⁶, CryptoVerif¹⁷, and TAMARIN Prover¹⁸—highlighting their theoretical underpinnings, functionalities, and capabilities in cryptographic protocol analysis. A general comparison between these simulation-based tools is provided in Table 8

- *AVISPA tool*: The Automated Validation of Internet Security Protocols and Applications (AVISPA) is a symbolic analysis tool for the formal verification of security protocols. It is designed to analyze large-scale and industrially relevant protocols using a symbolic representation of cryptographic operations. This tool operates within the Dolev-Yao symbolic model, where cryptographic primitives are assumed to be perfect (e.g., encryption is unbreakable without the key). Protocols are specified in the High-Level Protocol Specification Language (HLPSSL), which formalizes the interactions among principals, the adversary, and the network. Protocols, in the AVISPA tool, are described in HLPSSL, which defines roles (e.g., initiator, responder), message formats, and security goals (e.g., confidentiality, authenticity). And HLPSSL specifications are translated into intermediate formats understood by back-end solvers. Moreover, four back-end tools perform the analysis:

1. *On-the-Fly Model Checker (OFMC)*: Exhaustive search of execution paths.
 2. *CL-AtSe*: Constraint logic-based symbolic exploration.
 3. *SATMC*: SAT-based model checking.
 4. *TA4SP*: Specialized for timed security protocols.
- *ProVerif*: ProVerif is a state-of-the-art formal analysis tool for verifying cryptographic protocols, focusing on symbolic representations of messages and adversaries. It is particularly well-suited for secrecy, authentication, and other security properties. ProVerif employs the Dolev-Yao model for symbolic analysis. It encodes protocols as a set of Horn clauses (a type of logical formula), which are then analyzed for satisfiability. ProVerif’s symbolic approach enables it to explore all possible protocol executions and adversarial strategies. ProVerif verifies several security properties, such as secrecy (ensures that sensitive data are not leaked), authentication (confirms the identity of protocol participants), and forward secrecy (guarantees that session keys remain secure even if long-term keys are compromised).
 - *CryptoVerif*: CryptoVerif has bridged the gap between symbolic and computational models by providing a tool for analyzing protocols under computational assumptions. Unlike symbolic tools, it has proved security properties in the standard computational model, leveraging the probabilistic nature of cryptographic constructs. This advanced tool works in the asymptotic security model, where the adversary’s advantage is analyzed as a negligible function of security parameters. It employs the Game-Based Framework, defining protocols as sequences of games (this tool can be considered as the formalized form of the game-based model described in Section 3.9.1), and evaluates adversarial success probabilities.

Protocols, analyzed through in CryptoVerif framework, are written in a dedicated language similar to ProVerif which many probabilistic aspects (e.g., random sampling) are included explicitly.

The CryptoVerif translates the protocol into a sequence of games and each game step approximates

¹⁵ <https://www.avispa-project.org/>

¹⁶ <https://bblanche.gitlabpages.inria.fr/proverif/>

¹⁷ <https://bblanche.gitlabpages.inria.fr/CryptoVerif/>

¹⁸ <https://tamarin-prover.com/>

Table 8 Summary comparison of simulation-based tools

Tool	Model	Security Properties	Strengths	Limitations
AVISPA	Symbolic (Dolev-Yao)	Confidentiality, Authentication	Automated, scalable for industrial protocols	Symbolic-only, no computational guarantees
ProVerif	Symbolic (Dolev-Yao)	Secrecy, Authentication, Forward Secrecy	Automated, supports complex cryptographic primitives	Over-approximations can lead to false positives
CryptoVerif	Computational	Secrecy, Authentication, Reductionist Proofs	Computational guarantees, rigorous proofs	Requires expertise, less automation
TAMARIN Prover	Hybrid	Stateful protocols, Privacy (Trace Equivalence)	Combines symbolic and computational analysis	Steep learning curve, interactive proof complexity

the behavior of the protocol under adversarial strategies in addition to providing computational security guarantees for proof generation is automatic.

- *TAMARIN prover*: TAMARIN Prover is a versatile tool for analyzing cryptographic protocols, combining the symbolic analysis of tools like ProVerif with the ability to model complex stateful protocols and advanced properties such as trace equivalence. This security prover extends the Dolev-Yao model to support reasoning about state transitions and global constraints. It uses rewriting rules to represent protocol operations, where rules capture message generation, key exchanges, and adversarial capabilities. TAMARIN Prover is empowered by several unique features, such as (i) stateful protocols, to model protocols with long-term states (e.g., secure channels, session resumption), (ii) trace equivalence to verify indistinguishability of protocol executions, crucial for privacy properties like anonymity and unlinkability, and (iii) interactive proofs to allow manual intervention to refine proof strategies for complex properties.

4 Reaching security and privacy aspects using cryptography

The proliferation of the BB-OpenFi framework has necessitated advanced security and privacy frameworks to protect users, ensure compliance with regulations, and promote system integrity. Cryptographic tools are foundational to achieving these objectives. They provide a mathematical and protocol-based approach to securing sensitive information, ensuring transactional integrity, and enabling trustless interactions in decentralized systems.

By leveraging tools such as ZKPs, HE, and threshold cryptography, the BB-OpenFi framework achieves unparalleled security and privacy. These tools enable confidentiality, integrity, verifiability, and resilience against traditional attack vectors. A summary of these methods is provided at the end of this section, addressing technical aspects, challenges, and benefits for both users and developers and is presented in Table 9.

4.1 Confidentiality

Confidentiality ensures that sensitive user data, transactional information, and identity remain private, even in an open, decentralized environment. In the BB-OpenFi framework, confidentiality must address:

4.1.1 Data Protection (DP)

Protecting sensitive financial data can be achieved using HE schemes [24,25]. For example, data D is encrypted as $Enc(PK, D)$, allowing computations on encrypted data. Formally, for an operation $\oplus$, $Enc(PK, d_1 \oplus d_2) = Enc(PK, d_1) \circ Enc(PK, d_2)$, where $\circ$ represents the homomorphic equivalent of $\oplus$.

Encrypted user data stored on decentralized nodes allows computation without revealing raw data.

4.1.2 Identity Privacy

Ensuring user anonymity while interacting with the system generally can be provided through ZKP schemes [27]. A $\mathcal{P}$ demonstrates knowledge of a secret x without revealing it. For a statement $S(x)$ it is defined as $1 \leftarrow Verify(\pi, S)$.

This tool enables users to prove eligibility (e.g., creditworthiness), in the BB-OpenFi framework, without revealing their identity.

4.1.3 Confidential Transactions (CT)

The zk-SNARK schemes are generally considered tools for maintaining transactional privacy without compromising transparency [37,38]. This tool enables verifiable transactions as $\exists w: Prove(w, T, R) \wedge Verify(w, T, R)$.

Using zk-SNARK schemes in the BB-OpenFi framework ensures transaction details remain private while proving compliance with rules.

4.2 Integrity

Data integrity ensures that stored or transmitted data is tamper-proof [41]. In the BB-OpenFi framework, integrity is essential for (1) preventing fraudulent activities, (2) building trust in financial systems, and (3)

Table 9 The summary of Section 4

Aspect	Confidentiality	Integrity	Verifiable Computing	Multiserver
Cryptographic Tools	HE, zk-SNARKs	Hashing, Merkle Trees	zk-SNARKs, YOSO Models	Threshold Cryptography, MPC
Challenges	Scalability, key management	Collision resistance	Dynamic allocation	High latency
Benefits	Privacy, data protection	Fraud prevention	Efficient outsourcing	Fault tolerance
GDPR Aspects	Anonymization compliant	Integrity requirements met	Meets transparency mandates	Distributed data minimization

Table 10 Comparison of YOSO Models

Model	Computation Distribution	Collusion Resistance	Challenges
Static YOSO	Predefined node roles	Moderate	Limited scalability
Dynamic YOSO	Roles assigned dynamically	High	Role allocation complexity

ensuring regulatory compliance. In the following, three fundamental cryptographic tools that can provide the mentioned properties are addressed.

1. *Hash Functions*: This tool is generally considered the main tool to provide a secure way to verify data integrity. For a message m , $H(m)$ ensures any modification results in $H(m') \neq H(m)$ [96]. However, finding a collision-resistant hash function in high-throughput environments (an ideal hash function) is a challenge.
2. *Merkle Trees*: Merkle tree is defined as a two-to-one hash function and formally shown as $h : \{0, 1\}^{2l} \rightarrow \{0, 1\}^l$. It efficiently verifies the integrity of large datasets [97]. Leaves are hashed, and parent nodes are recursively computed $Merkle-root = H(H_{L1} || H_{L2})$ which the verification of a leaf requires to calculate only $\log(n)$ hashes.
3. *Digital Signatures*: Digital signature schemes are used for user authentication and content verification [98]. In these cryptographic schemes, message m is signed using SK as $\sigma = Sign(SK, m)$. On the other side, everyone can check the validity using PK through $Verify(PK, \sigma, m)$. Although all types of digital signature schemes are popular tools, key management in decentralized environments is challenging.

4.3 Verifiable computation

A client outsources computation to a server and verifies the result without re-executing the computation [99, 100]. Formally, for a function $f(x)$: $1 \leftarrow Verify(f, x, y)$ iff $y = f(x)$.

Verifiable computing provides many applications in the BB-OpenFi framework, such as efficient contract execution via zk-SNARKs, ensuring correctness in off-chain computations, and privacy-preserving audits of financial operations. Moreover, You Only Speak Once (YOSO) Models [101, 102] in the BB-OpenFi framework distribute computation tasks among multiple nodes, ensuring no single node holds complete sensitive data (see

Table 10 for comparison between YOSO models). For example, YOSO models can enhance security, and resistance to collusion by leveraging secret-sharing and threshold cryptography.

4.4 Multiserver and removing point of failure

Multiserver architectures eliminate single points of failure by distributing responsibilities [101]. In the BB-OpenFi framework, this increases resilience, ensures continuous service availability, and enhances trust via redundancy.

Threshold cryptography ensures distributed control over sensitive operations. For t -out-of- n scheme $f(x) = \sum_{i=1}^t c_i \cdot x^i$, reconstruction from t shares. And MPC enables secure MPC [103] without exposing individual data. Applying these advanced cryptographic tools in the BB-OpenFi framework provides many benefits, such as increasing fault tolerance and enhancing user trust due to decentralized control.

4.5 Cross-layer security mechanisms

A cross-layer security mechanism integrates Layer 1 (for consensus) [116] and Layer 2 (to enhance scalability) [107, 117] frameworks [115]. By leveraging zk-STARK proofs on L2, the BB-OpenFi framework can maintain verifiable computation integrity without overloading the base blockchain. An example of layered execution is $Verify(L1, L2) = ZKP(\sum_{i=0}^n L2_i)$, where $L2_i$ corresponds to off-chain computations aggregated via zero-knowledge proofs. This integration enhances throughput and ensures robust tamper-resistance across layers.

Additionally, integrating AI for automated cryptographic parameter tuning using reinforcement learning can optimize cryptographic protocols dynamically based on transaction throughput and user privacy requirements.

5 Achievements for BB-OpenFi

The BB-OpenFi framework emerges as a revolutionary evolution in the decentralized financial landscape, transcending the limitations of traditional OpenFi and

DeFi. Table 11 provides a brief comparison of BB-OpenFi’s achievements with DeFi and OpenFi. BB-OpenFi’s achievements lie in its seamless integration of advanced cryptographic techniques, enhanced scalability, compliance mechanisms, and robust security frameworks. This section explores these achievements comprehensively, emphasizing the distinct advantages of the BB-OpenFi framework and its motivation for developers and industry leaders.

Table 12 summarizes the provided futures of cryptographic tools in the BB-OpenFi framework. As we predict the BB-OpenFi framework will be used in the future, several quantum-side achievements, relying on the power of the applied quantum cryptographic tools, are briefly proposed in Table 13.

5.1 Enhanced Security and Privacy

Security and privacy are paramount in financial systems and the BB-OpenFi framework leverages cutting-edge cryptographic tools to provide unparalleled protections. Unlike OpenFi and DeFi, which often rely on traditional methods or lack robust privacy features, the BB-OpenFi framework achieves:

1. *Confidentiality*: Advanced cryptographic tools like zk-SNARKs [37,38], HE [24,24], and Ring Signatures protect user identities, transaction details, and sensitive data [104,105,106]. Hence, users can interact with financial platforms without fearing identity theft or unauthorized data exposure.
2. *Data Integrity*: The use of Merkle Trees [97], Hash Functions [96], and Digital Signatures [98] ensures data immutability, tamper resistance, and accountability. As an achievement, the BB-OpenFi framework prevents fraud, promotes transparency, and builds user trust in BB-OpenFi systems.
3. *Resilience Against Attacks*: Threshold Cryptography [59,60] and MPC [23,103] eliminate single points of failure and mitigate risks from cyberattacks. Therefore, high availability and robustness that are essential for mission-critical financial applications are provided by the BB-OpenFi framework.
4. *Quantum security*: The BB-OpenFi framework can adopt XMSS or SPHINCS signatures [66] for tamper-proof data validation. These systems generate cryptographically secure Merkle trees [97] for verification.

5.2 Advanced Scalability and Efficiency

The BB-OpenFi framework addresses scalability challenges that have long hindered OpenFi and DeFi adoption, particularly for global-scale financial operations.

1. *L2 Solutions*: Integration with L2 solutions (e.g., rollups [107], sharding [108]) enables the offloading of computations while maintaining the security guarantees of Layer 1 (L1). Relying on this basis, the BB-OpenFi framework achieves higher throughput and lower transaction costs.
2. *Optimized Verifiable Computing*: By employing state-of-the-art verifiable computing frameworks such as zk-SNARKs [37,38] and YOSO models [101,102], the BB-OpenFi framework ensures efficient and provably correct off-chain computations. This reduces computational overhead on the main blockchain, improving system responsiveness.

5.3 Regulatory Compliance and GDPR Alignment

One of the significant barriers for DeFi platforms has been their inability to align with regulatory frameworks and data protection laws like GDPR. the BB-OpenFi framework overcomes these challenges through the following considerations.

1. *Built-in Anonymization Mechanisms*: Leveraging ZKPs, the BB-OpenFi framework allows systems to demonstrate compliance (e.g., KYC/AML checks) without exposing private user data [56,109,110]. This advanced tool ensures regulatory approval while preserving user privacy.
2. *Data Minimization*: Using distributed architectures and cryptographic primitives, the BB-OpenFi framework achieves GDPR-compliant data storage and processing. Hence, it reduces service providers’ liability and fosters trust with users.
3. *Auditability*: Transparent and immutable records on the blockchain enable verifiable audits while safeguarding sensitive data [111,112]. This property of blockchain enhances trust with regulators and users.

5.4 User-Centric Design and Accessibility

The BB-OpenFi framework empowers users by offering greater control, inclusivity, and fairness compared to OpenFi and DeFi. Therefore, the following achievements increase the user-friendly aspects.

1. *Democratized Access*: Decentralized and permissionless architectures reduce barriers to entry, making financial services accessible globally. This level of

Table 11 Distinguishing BB-OpenFi from OpenFi and DeFi

Aspect	BB-OpenFi	DeFi	OpenFi
Privacy	Advanced tools (zk-SNARKs, HE)	Limited anonymity	Minimal privacy measures
Regulatory Compliance	GDPR alignment, auditability	Limited compliance frameworks	Typically unaligned
Regulatory Alignment	Full GDPR and AML compliance	Limited frameworks	PSD2-compliant
Scalability	L2 and verifiable computing	Scalability challenges	Moderate scalability
Throughput	High (10k TPS with L2)	Low (1k TPS max)	High (>10k TPS)
Security	Threshold cryptography, MPC	Basic cryptographic tools	Centralized controls
Quantum Resistance	zk-STARKs, Lattice Cryptography	Not quantum-safe	Not applicable
Ecosystem Integration	Cross-chain interoperability	Limited composability	Fragmented ecosystems
Energy Efficiency	PoS and zk-Rollups	Energy-intensive PoW	Not optimized for eco-friendliness

Table 12 Futures provided by cryptographic tools in BB-OpenFi

Tool	Use Case	Limitations Addressed	Scalability Potential
zk-STARKs	Verifiable computation	Trusted setup in zk-SNARKs	High
Homomorphic Encryption	Confidential data sharing	Lack of data privacy	Moderate
Lattice Cryptography	Post-quantum security	Vulnerability to quantum attacks	High
Threshold Cryptography	Decentralized control	Single points of failure	Moderate

Table 13 Post-Quantum Impact Analysis

Aspect	Pre-PQC BB-OpenFi	Post-PQC BB-OpenFi
Scalability	Moderate (Rollups)	High (zk-STARK + Lattice HE)
Security	Classical Cryptography	Quantum-Resistant Algorithms
Privacy	zk-SNARKs	zk-STARKs + Lattice Encryption

accessibility promotes financial inclusion, especially in underbanked regions.

2. *User Sovereignty*: Integration of Self-Sovereign Identity (SSI) systems [113,114], where users control their data and decide what to share. Therefore, SSI redefines the relationship between users and financial platforms, increasing autonomy and satisfaction.
3. *Lower Fees*: Efficiency gains from advanced cryptographic methods and L2 technologies significantly reduce transaction costs [107,108]. So, a cost-efficient service makes financial services affordable for small-scale users.

5.5 Interoperability and Composability

The BB-OpenFi framework is designed to operate in a diverse ecosystem of financial systems and blockchains, fostering collaboration and innovation [115]. The following high-tech technologies can provide much help to this issue.

1. *Cross-Chain Operability*: The BB-OpenFi framework can employ atomic swaps, Cross-Chain Bridges, and

Interledger Protocols (ILP) to facilitate seamless transactions across different blockchains. Relying on these technologies encourages ecosystem synergy and expands the scope of financial applications.

2. *Composable Smart Contracts*: The modular design of smart contracts allows developers to build on existing protocols, enhancing innovation speed and accelerating the development of novel financial products and services.

The BB-OpenFi framework requires scalability solutions beyond traditional blockchain limitations. We introduce a hybrid L2 framework integrating zk-Rollups and Validium to enhance transaction throughput while preserving decentralization. zk-Rollups leverage off-chain computation while maintaining on-chain validity through succinct proofs, where transactions $\mathcal{T}$ are batched and verified via $\pi = \text{Prove}(\mathcal{T}, R)$. Additionally, Validium supports high-volume data storage off-chain while ensuring data availability through commitment schemes $C = h(D)$. Our proposed hybrid approach optimises scalability and security, ensuring the BB-OpenFi framework supports large-scale financial operations efficiently.

5.6 Sustainability and Eco-Friendliness

The BB-OpenFi framework adopts mechanisms to reduce the environmental footprint of blockchain operations, addressing growing concerns about energy-intensive systems. There are two main ideas to address this.

1. *Proof-of-Stake (PoS) and Alternatives*: Transition to PoS and energy-efficient consensus mechanisms reduces energy consumption compared to Proof-of-Work (PoW) [116]. This idea aligns with global sustainability goals, making the BB-OpenFi framework attractive to environmentally-conscious companies.
2. *Optimized Resource Usage*: Cryptographic techniques like zk-Rollups [117] reduce on-chain storage and computational needs and cause increase efficiency and minimize waste.
3. *Decentralized Energy Credits Management*: Incorporating blockchain-based carbon credit trading systems in the BB-OpenFi framework using crypto-

graphic auditing [111,112] can be considered an excellent achievement.

5.7 Economic and Business Advantages

BB-OpenFi's technical advancements translate into significant business opportunities and economic benefits, such as:

1. *Innovative Revenue Models*: The BB-OpenFi framework enables new financial instruments (e.g., privacy-preserving DeFi, tokenized assets) and attracts both traditional and crypto-native investors.
2. *Developer Incentives*: Transparent frameworks for rewarding developers who contribute to protocol improvements and accelerate ecosystem growth.
3. *Enterprise Adoption*: This paradigm enhanced security, scalability, and compliance making the BB-OpenFi framework suitable for large-scale enterprise applications and bridging the gap between traditional finance and decentralized systems.

6 Background challenges and possible solutions

The adoption of the BB-OpenFi framework introduces a multitude of challenges that stem from both the underlying blockchain infrastructure and the integration of financial services into a decentralized ecosystem. This section delves into several critical challenges inherent to the BB-OpenFi framework and proposes advanced cryptographic and technical solutions to address them. The summary of this section is presented in Table 14.

6.1 Key Escrow and Protecting Secret Parameters

We have divided this section into two following subsections.

6.1.1 Key Escrow

Key escrow remains a major challenge in decentralized systems, as the ability of a third party to access private keys undermines user trust and system decentralization. Private keys become a single point of failure and a target for attacks if held in escrow by a centralized authority. Hence, compromising escrowed keys could lead to unauthorized access to sensitive data or funds. To face this challenge, two general solutions are now listed below:

1. *Threshold Cryptography* [59, 60]: Threshold cryptography divides the private key SK into n shares sk_1 ,

$sk_2, \dots, sk_n$ using a secret sharing scheme such as Shamir's Secret Sharing. A predefined threshold t allows reconstruction of the key only if t or more shares are combined.

To do that, SK should be divided into shares as $P(x) = a_0 + a_1x + \dots + a_{t-1}x_{t-1}$, where $P(0) = SK$ and each participant i holds $sk_i = P(i)$.

To reconstruct SK , at least t shares are required, using Lagrange interpolation as $SK = \sum_{i=1}^t sk_i \prod_{j \neq i} \frac{x_j}{x_j - x_i}$.

2. *Distributed Key Generation (DKG)* [118, 119]: DKG eliminates the need for a trusted party during key generation. Using protocols like Pedersen's DKG, all participants jointly compute a public key PK and their private shares without ever exposing SK . To do that, each participant P_i commits to a polynomial $f_i(x)$: $C_{ij} = g^{f_i(j)}$, $\forall j$. As the result, $PK = \prod_i g^{f_i(0)}$, and shares $sk_i = f_i(0)$ remain secret.

6.1.2 Protecting Secret Parameters

In the BB-OpenFi framework, secret parameters such as cryptographic keys or private data must be safeguarded against unauthorized access. Secure computation involving private data is challenging in a decentralized. Hence, storing sensitive parameters on-chain is infeasible due to transparency [120]. To avoid this, we offer possible solutions based on the power of advanced cryptographic tools.

1. *HE* [24, 25]: HE enables computations on encrypted data without decrypting it. For example, given ciphertexts $c_1 = Enc(PK, m_1)$ and $c_2 = Enc(PK, m_2)$, it holds that $Dec(SK, c_1.c_2) = m_1 + m_2$. This tool protects user data during computations in the BB-OpenFi framework while ensuring privacy.
2. *ZKP* [27]: ZKPs allow a $\mathcal{P}$ to demonstrate knowledge of a secret parameter x satisfying $f(x) = y$, without revealing x . zk-SNARKs can prove that a user holds sufficient funds $bal > t$ without disclosing bal . Formally, a zk-SNARK proof π ensures $Verify(PK, \pi, y) = 1$ if $\exists x : f(x) = y$.

6.2 Determining critical and public parameters

Deciding critical and public parameters (e.g., consensus configurations, token supply) requires both domain expertise and adherence to decentralized governance principles. The two below technologies can polish these types of decisions.

1. *DAO* [50]: DAOs use smart contracts to enable governance through token-holder voting. To do this, voting power V_i is proportional to token holdings

Table 14 The summary of Section 6

Challenge	Problem Description	Proposed Solutions	Key Benefits	Implementation Challenges
Key Escrow	Centralized key storage creates a single point of failure.	1. Threshold Cryptography: Split SK into n shares with t threshold reconstruction. 2. Distributed Key Generation (DKG): Joint key generation without exposure.	- Eliminates single point of failure. - Improves security and trust in BB-OpenFi.	- Increased computation and storage overhead. - Complexity in protocol implementation.
Protecting Secret Parameters	Ensuring private data and keys remain confidential during processing and storage.	1. HE: Computation on encrypted data. 2. ZKPs: Privacy-preserving verification of statements.	- Enables privacy in public blockchain settings. - Supports secure computations without exposing data.	- Heavy computational load for HE. - Designing efficient ZKP circuits for specific use cases.
Determining Critical Parameters	Identifying knowledgeable individuals or groups to establish critical parameters in a decentralized way.	1. DAOs: Governance through token-based voting. 2. Multi-Signature Schemes: Collective approval for parameter updates.	- Ensures decentralized decision-making. - Aligns with blockchain transparency and governance principles.	- Token-weighted governance risks centralization. - Multi-signature schemes may introduce latency.
Scalability	Limited transaction capacity and high latency in blockchain systems.	1. L2 Solutions (e.g., zk-Rollups): Offload computations to L2. 2. Sharding: Partition blockchain into parallelizable shards.	- Increases throughput without compromising security. - Reduces transaction costs.	- Designing secure shard communication. - zk-Rollups require efficient proof generation.
Throughput	Inability to handle high transaction volumes required by BB-OpenFi.	1. Parallel Processing: Execute independent transactions concurrently. 2. Optimized Consensus (e.g., Tendermint): Fast finality consensus.	- Supports high-frequency financial transactions. - Reduces transaction latency for end-users.	- Requires high-performance computing infrastructure. - Optimized consensus may reduce decentralization in large networks.
Interoperability	Ensuring compatibility between BB-OpenFi and other blockchains.	Cross-chain solutions like atomic swaps and blockchain bridges.	- Enhances usability by enabling asset movement across chains. - Supports multi-platform integration.	- Risk of bridge vulnerabilities. - High latency for cross-chain transactions.
Energy Efficiency	High energy consumption in proof-of-work-based systems.	Transition to PoS consensus mechanisms.	- Reduces environmental impact. - Lowers operational costs.	- Initial migration effort. - Security concerns in early-stage PoS models.
Usability	Complex interfaces hinder user adoption.	Simplified UI/UX with cryptographic abstraction layers.	- Improves user experience. - Attracts non-technical users to BB-OpenFi.	- Balancing simplicity with the need for transparency. - Designing accessible yet secure interfaces.
Governance Disputes	Resolving conflicts in decentralized decision-making processes.	Hybrid governance models combining on-chain voting and off-chain expert arbitration.	- Improves dispute resolution without compromising decentralization. - Ensures fair representation.	- Defining the arbitration framework. - Risk of governance capture by powerful entities.

T_i as $V_i = \frac{T_i}{\sum_j T_j}$. The consensus on parameter P is achieved if $\sum_i V_i \delta(P_i = P) \geq \text{Threshold}$.

2. *Multi-Signature Schemes* [121, 122]: Multi-signature wallets require multiple approvals before parameter changes are finalized. In this scheme, P is valid if $\text{Signatures} - \text{received} \geq \text{Threshold}$.

6.3 Scalability

The scalability trilemma (security, decentralization, scalability) is a persistent challenge for blockchain systems [115], including the BB-OpenFi framework. The two following ideas are assumed as the most famous solutions to face this challenge.

1. *L2 Solutions* [107]: zk-Rollups are off-chain transactions to L2 while maintaining security [117]. In zk-Rollups, for n transaction $T_1, T_2, \dots, T_n$: $\text{Commit} = H(T_1 || T_2 || \dots || T_n)$, and finally, On-chain only the commitment is stored.
2. *Sharding* [108]: Divide the blockchain into shards, each processing a subset of transactions. Transactions are routed to shard s_k based on $H(T) \bmod k$.

6.4 Throughput

High throughput is critical for financial systems to handle large transaction volumes [115]. There are two innovative ideas to increase throughput.

1. *Parallel Processing*: Use parallel execution engines to process transactions simultaneously. Let T_1 and T_2 be independent; execute $\text{Exec}(T_1 || T_2)$.
2. *Optimized Consensus Algorithms*: Algorithms like Tendermint reduce consensus overhead, achieving sub-second finality.

6.5 Other challenges

Beyond the primary challenges, the BB-OpenFi framework faces marginal yet significant obstacles that require consideration and tailored solutions.

- *Interoperability*: Employing atomic swaps and cross-chain bridges ensures cross-chain compatibility.
- *Energy Efficiency*: Transitioning to PoS causes reduced energy usage.
- *Usability*: Designing simplified UI/UX designs with abstracted cryptographic details improves complex interfaces and deters adoption.
- *Governance Disputes*: Hybrid models combining on-chain voting and experts can resolve conflicts in decentralized governance.

7 Comparison and experimental results

This section provides a comparative analysis of the advanced cryptographic primitives used in the BB-OpenFi framework, evaluates computational costs, and presents experimental results.

- *Computational complexity analysis:* The computational efficiency of cryptographic primitives in the BB-OpenFi framework is critical for practical implementation. Table 15 presents a comparison of key schemes.
- *Communication overhead:* Cryptographic tools impact communication efficiency. Table 16 compares their network requirements.
- *Comparative analysis of BB-OpenFi features:* The BB-OpenFi framework integrates cryptographic primitives to balance security, efficiency, and compliance. Table 17 illustrates and compares these features.
- *Experimental results:* The following experimental results compare the performance of BB-OpenFi’s cryptographic primitives with prior works.

We benchmarked key cryptographic primitives in the BB-OpenFi framework simulation using a dataset of 10,000 transactions. The results are shown in Table 18. Additionally, the BB-OpenFi framework optimizes transaction throughput while maintaining security. The related results are illustrated in Table 19.

All in all, lattice-based PQC (e.g., Dilithium) is more efficient than classical cryptographic primitives while being quantum-safe. The zk-STARKs and HE schemes increase privacy but introduce higher computational costs. The BB-OpenFi framework ensures compliance while preserving decentralization.

8 Conclusion and future directions

This paper has introduced a secure, scalable, and privacy-preserving framework for Blockchain-Based Open Finance (BB-OpenFi), demonstrating how advanced cryptographic techniques can revolutionize decentralized financial systems. By integrating ZKPs, MPC, HE, and threshold cryptography, we have proposed a comprehensive cryptographic infrastructure that successfully addresses critical challenges, including key escrow elimination, confidential transaction execution, scalable consensus mechanisms, and regulatory compliance.

Our analysis highlights that the BB-OpenFi framework is not just an incremental improvement over existing OpenFi and DeFi models, but rather a transformational shift that enables a trustless, regulation-compliant, and high-performance financial ecosystem.

The proposed methodologies ensure that transactions remain tamper-proof, auditable, and quantum-resistant, positioning the BB-OpenFi framework as a future-proof financial architecture.

Future directions: As blockchain adoption accelerates in financial sectors, BB-OpenFi’s ability to bridge traditional finance with decentralized models makes it a compelling framework for financial institutions, fintech startups, and regulators. Future research directions include enhancing quantum resistance, integrating AI-driven fraud detection, and optimizing decentralized financial contracts to further advance the scalability, security, and usability of the BB-OpenFi framework ecosystems. With these advancements, the BB-OpenFi framework is poised to become a cornerstone of next-generation global finance, redefining how users and institutions interact in a trustless and decentralized digital economy. In this regard, we offer several future directions for both academia and industry sectors who want to innovate in this area of technology.

– Academia

1. *Advanced Cryptographic Schemes:* Future research should explore optimized and scalable implementations of cryptographic tools like ZKPs, SNARKs, and fully homomorphic encryption tailored to BB-OpenFi systems.
2. *Formal Verification Frameworks:* Developing formal verification tools to ensure the correctness and security of the BB-OpenFi framework protocols will be a critical research direction.
3. *Blockchain Trilemma:* Investigating innovative consensus mechanisms or hybrid architectures that address the trilemma of scalability, security, and decentralization in the BB-OpenFi framework.
4. *AI in Cryptography for BB-OpenFi:* AI-driven cryptography can detect anomalous patterns in BB-OpenFi operations by integrating machine learning models trained on blockchain transaction data. For instance, reinforcement learning models can optimize gas usage in high-throughput systems, while neural networks classify potential fraud. This fusion of AI and cryptographic tools ensures proactive threat management.

– Industry

1. *Interoperability Solutions:* Developing robust cross-chain communication frameworks to enable seamless integration of the BB-OpenFi framework with existing financial systems and other blockchain networks.
2. *Regulatory Compliance:* Building GDPR-compliant systems that balance user privacy with trans-

Table 15 The comparison of the computational efficiency of the cryptographic tools if applied in BB-OpenFi

Cryptographic tool/scheme	Key size (bit)	Signature size (byte)	Verification time	Computation cost	Security level
RSA-2048	2048	256	High	$\mathcal{O}(\log n)$ exponentiations	Classical
ECDSA-256	256	64	Moderate	$\mathcal{O}(1)$ scalar multiplications	Classical
Dilithium-III (PQ)	1952	2420	Low	Polynomial multiplications	Post-Quantum
SPHINCS+	256	17 K	Moderate	Hash-based signing	Post-Quantum

Table 16 The comparison of the communication overhead of the cryptographic schemes if applied in BB-OpenFi

Cryptographic tool/scheme	Communication Overhead	Scalability Impact
zk-SNARKs	Low	High
zk-STARKs	Moderate	High
HE	High	Moderate
Threshold Signatures	Moderate	High

Table 17 A comparison between balance security, efficiency, and compliance

Feature	OpenFi	DeFi	BB-OpenFi
Privacy	Low	High	Very High
Scalability	High	Low	High (L2, zk-Rollups)
Compliance	High	Low	High (GDPR, FIDA)
Security	Moderate	Moderate	Very High (Threshold Crypto, PQC)

Table 18 A comparison between computational benchmarks

Method	Avg. signature gen time (ms)	Avg. verification time (ms)	Computational Overhead (%)
ECDSA	1.2	0.8	100
Dilithium	0.9	0.6	75
SPHINCS+	1.5	1.1	130

Table 19 A comparison between computational benchmarks

Framework	TPS	Latency (ms)
OpenFi	10,000	5
DeFi	1000	120
BB-OpenFi	8,500	15

parency, ensuring widespread adoption by regulated industries.

3. *Scalable Infrastructure*: Implementation of L2 solutions and sharding techniques to enhance scalability and throughput for large-scale financial transactions in the BB-OpenFi framework.

We believe both academia and industry must work collaboratively to refine and implement the BB-OpenFi framework technologies, ensuring they meet the demands of future financial ecosystems while upholding the principles of security, privacy, and decentralization.

Declarations and statements

Acknowledgment: We as authors appreciate anonymous reviewers for their insightful comments to improve this work.

Author(s) contribution: Saeed Banaeian Far: Contributor, Wrote the first draft - Mohammad Reza Chalak Qazani: Supervisor, Final comments - Azadeh Imani Rad: Advisor, Final review.

Competing interests: The authors declare that they have no conflict of interest in the publication of this article.

Funding: The authors received no financial support for the research and/or authorship of this article.

Ethical Approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Data availability statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. Brodsky, Laura, and Liz Oakes. "Data sharing and open banking." McKinsey & Company 1105 (2017).
2. Godard, Béatrice, Jörg Schmidtke, Jean-Jacques Cassiman, and Ségolène Aymé. "Data storage and DNA banking for biomedical research: informed consent, confidentiality, quality issues, ownership, return of benefits. A professional perspective." European journal of human genetics 11, no. 2 (2003): S88-S122.
3. Wang, Shuang, Muhammad Asif, Muhammad Farrukh Shahzad, and Muhammad Ashfaq. "Data privacy and cybersecurity challenges in the digital transformation of the banking sector." Computers & security 147 (2024): 104051.
4. Darem, Abdulbasit A., Asma A. Alhashmi, Tareq M. Alkhaldi, Abdullah M. Alashjaee, Sultan M. Alanazi, and Shouki A. Ebad. "Cyber threats classifications and countermeasures in banking and financial sector." IEEE Access 11 (2023): 125138-125158.
5. Alzoubi, Haitham M., Taher M. Ghazal, Mohammad Kamrul Hasan, Asma Alketbi, Rukshanda Kamran, Nidal A. Al-Dmour, and Shayla Islam. "Cyber security threats on digital banking." In 2022 1st International Conference on AI in Cybersecurity (ICAIC), pp. 1-4. IEEE, 2022.
6. Burgess, Trevor, Marcio Deoliveira, and Michael Claffey. "Securely integrating third-party applications with banking systems." U.S. Patent 9,525,690, issued December 20, 2016.
7. Premchand, Anshu, and Anurag Choudhry. "Open banking & APIs for transformation in banking." In 2018 international conference on communication, computing and internet of things (IC3IoT), pp. 25-29. IEEE, 2018.
8. Claessens, Stijn. "Access to financial services: a review of the issues and public policy objectives." (2005).
9. Remolina, Nydia. "Open Finance: Regulatory Challenges of the Evolution of Data Sharing Arrangements in the Financial Sector." SMU Centre for AI & Data Governance Research Paper 2019/05 (2022): 35-66.
10. Beck, Thorsten, and Augusto De La Torre. "The basic analytics of access to financial services." Financial markets, institutions & instruments 16, no. 2 (2007): 79-117.

11. Zetzsche, Dirk A., Douglas W. Arner, and Ross P. Buckley. "Decentralized finance." *Journal of Financial Regulation* 6, no. 2 (2020): 172-203.
12. Chen, Yan, and Cristiano Bellavitis. "Blockchain disruption and decentralized finance: The rise of decentralized business models." *Journal of Business Venturing Insights* 13 (2020): e00151.
13. Far, Saeed Banaeian, Azadeh Imani Rad, and Maryam Rajabzadeh Asaar. "Blockchain and its derived technologies shape the future generation of digital businesses: a focus on decentralized finance and the Metaverse." *Data Science and Management* 6, no. 3 (2023): 183-197.
14. Valiente, María-Cruz, and Juan Pavón. "Web3-DAO: an ontology for decentralized autonomous organizations." *Journal of Web Semantics* (2024): 100830.
15. Singh, Bhupinder, and Christian Kaunert. "Revealing Green Finance Mobilization: Harnessing FinTech and Blockchain Innovations to Surmount Barriers and Foster New Investment Avenues." In *Harnessing Blockchain-Digital Twin Fusion for Sustainable Investments*, pp. 265-286. IGI Global, 2024.
16. Albshaier, Latifa, Seetah Almarri, and M. M. Hafizur Rahman. "A review of blockchain's role in E-Commerce transactions: Open challenges, and future research directions." *Computers* 13, no. 1 (2024): 27.
17. Zhu, Yida, Keke Yu, Ming Wei, Yanli Pu, and Zeyu Wang. "AI-enhanced administrative prosecutorial supervision in financial big data: New concepts and functions for the digital era." *Journal of Advanced Computing Systems* 5, no. 1 (2025): 1-17.
18. Papachristou, Nikolaos, Grigorios Kotronoulas, Nikolaos Dikaos, Sarah J. Allison, Harietta Eleftherochorinou, Taranpreet Rai, Holger Kunz, Payam Barnaghi, Christine Miaskowski, and Panagiotis D. Bamidis. "Digital transformation of cancer care in the era of big data, artificial intelligence and data-driven interventions: navigating the field." In *Seminars in oncology nursing*, vol. 39, no. 3, p. 151433. WB Saunders, 2023.
19. Lake, David A. "Open economy politics: A critical review." *The Review of International Organizations* 4 (2009): 219-244.
20. Uzougbo, Ngozi Samuel, Chinonso Gladys Ikegwu, and Adefolake Olachi Adewusi. "Regulatory frameworks for decentralized finance (DEFI): challenges and opportunities." *GSC Advanced Research and Reviews* 19, no. 2 (2024): 116-129.
21. Kshetri, Nir, and Nir Kshetri. "The Indian blockchain landscape: Regulations and policy measures." *Asian Res. Policy* 9, no. 2 (2018): 56-71.
22. Abuhashim, Alkhansaa, and Chiu C. Tan. "Smart contract designs on blockchain applications." In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1-4. IEEE, 2020.
23. Goldreich, Oded. "Secure multi-party computation." Manuscript. Preliminary version 78, no. 110 (1998): 1-108.
24. Gentry, Craig. "Fully homomorphic encryption using ideal lattices." In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 169-178. 2009.
25. Van Dijk, Marten, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. "Fully homomorphic encryption over the integers." In *Advances in Cryptology-EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, French Riviera, May 30-June 3, 2010. *Proceedings* 29, pp. 24-43. Springer Berlin Heidelberg, 2010.
26. Bellare, Mihir, Alexandra Boldyreva, and Adam O'Neill. "Deterministic and efficiently searchable encryption." In *Advances in Cryptology-CRYPTO 2007: 27th Annual International Cryptology Conference*, Santa Barbara, CA, USA, August 19-23, 2007. *Proceedings* 27, pp. 535-552. Springer Berlin Heidelberg, 2007.
27. Fiege, Uriel, Amos Fiat, and Adi Shamir. "Zero knowledge proofs of identity." In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pp. 210-217. 1987.
28. Mansfield-Devine, Steve. "Open banking: opportunity and danger." *Computer Fraud & Security* 2016, no. 10 (2016): 8-13.
29. Omarini, Anna Eugenia. "Banks and FinTechs: How to develop a digital open banking approach for the bank's future." *International Business Research* 11, no. 9 (2018): 23-36.
30. Klagge, Britta, and Ron Martin. "Decentralized versus centralized financial systems: is there a case for local capital markets?." *Journal of Economic Geography* 5, no. 4 (2005): 387-421.
31. Kennedy, John James. "Finance and rural governance: centralization and local challenges." In *Rural Politics in Contemporary China*, pp. 101-118. Routledge, 2016.
32. Friesendorf, Cordelia, and Alena Blütener. "Centralized Finance." In *Decentralized Finance (DeFi) How Decentralized Applications (dApps) Disrupt Banking*, pp. 11-27. Cham: Springer Nature Switzerland, 2023.
33. Denning, Dorothy E., and Dennis K. Branstad. "A taxonomy for key escrow encryption systems." *Communications of the ACM* 39, no. 3 (1996): 34-40.
34. Lin, Jingqiang, Wen-Tao Zhu, Qiong Xiao Wang, Nan Zhang, Jiwu Jing, and Neng Gao. "RIKE+: Using revocable identities to support key escrow in public key infrastructures with flexibility." *IET Information Security* 9, no. 2 (2015): 136-147.
35. Alhawamdeh, Loai N., Ayman Abdalmajeed Alsmadi, Manaf Al-Okaily, and Abdalmuttaleb Al-Sartawi. "The Rise of Open Banking: Analyzing Consumer Trust and Data Privacy Concerns." In *Artificial Intelligence and Economic Sustainability in the Era of Industrial Revolution 5.0*, pp. 243-257. Cham: Springer Nature Switzerland, 2024.
36. Baum, Carsten, James Hsin-yu Chiang, Bernardo David, and Tore Kasper Frederiksen. "Sok: Privacy-enhancing technologies in finance." *Cryptology ePrint Archive* (2023).
37. Bowe, Sean, Ariel Gabizon, and Matthew D. Green. "A multi-party protocol for constructing the public parameters of the Pinocchio zk-SNARK." In *Financial Cryptography and Data Security: FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers* 22, pp. 64-77. Springer Berlin Heidelberg, 2019.
38. Luong, Duc Anh, and Jong Hwan Park. "Privacy-preserving blockchain-based healthcare system for IoT devices using zk-SNARK." *IEEE Access* 10 (2022): 55739-55752.
39. Ben-Sasson, Eli, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Scalable zero knowledge with no trusted setup." In *Advances in Cryptology-CRYPTO 2019: 39th Annual International Cryptology Conference*, Santa Barbara, CA, USA, August 18-22, 2019, *Proceedings*, Part III 39, pp. 701-732. Springer International Publishing, 2019.

40. Wei, Li, Liang Peili, and Li Fei. "zk-STARKs based scheme for sealed auctions in chains." *IET Blockchain* (2024).
41. Zikratov, Igor, Alexander Kuzmin, Vladislav Akimenko, Viktor Niculichev, and Lucas Yalansky. "Ensuring data integrity using blockchain technology." In 2017 20th Conference of Open Innovations Association (FRUCT), pp. 534-539. IEEE, 2017.
42. Bertino, Elisa, Ahish Kundu, and Zehra Sura. "Data transparency with blockchain and AI ethics." *Journal of Data and Information Quality (JDIQ)* 11, no. 4 (2019): 1-8.
43. Szabo, Nick. "Formalizing and securing relationships on public networks." *First monday* (1997).
44. Zou, Weiqin, David Lo, Pavneet Singh Kochhar, Xuan-Bach Dinh Le, Xin Xia, Yang Feng, Zhenyu Chen, and Baowen Xu. "Smart contract development: Challenges and opportunities." *IEEE transactions on software engineering* 47, no. 10 (2019): 2084-2106.
45. Feldman, Paul. "A practical scheme for non-interactive verifiable secret sharing." In 28th Annual Symposium on Foundations of Computer Science (sfcs 1987), pp. 427-438. IEEE, 1987.
46. Eskicioglu, Ahmet M., and Edward J. Delp. "A key transport protocol based on secret sharing applications to information security." *IEEE Transactions on Consumer Electronics* 48, no. 4 (2002): 816-824.
47. Li, Chaoyang, Mianxiong Dong, Xiangjun Xin, Jian Li, Xiu-Bo Chen, and Kaoru Ota. "Efficient privacy preserving in IoMT with blockchain and lightweight secret sharing." *IEEE Internet of Things Journal* 10, no. 24 (2023): 22051-22064.
48. Tian, Youliang, Jianfeng Ma, Changgen Peng, and Qi Jiang. "Fair (t, n) threshold secret sharing scheme." *IET Information Security* 7, no. 2 (2013): 106-112.
49. Cascudo, Ignacio, Ronald Cramer, and Chaoping Xing. "Bounds on the threshold gap in secret sharing and its applications." *IEEE Transactions on Information Theory* 59, no. 9 (2013): 5600-5612.
50. Far, Saeed Banaeian, and Seyed Mojtaba Hosseini Bamakan. "Blockchain-based reporting protocols as a collective monitoring mechanism in DAOs." *Data Science and Management* 5, no. 1 (2022): 11-12.
51. Halpin, Harry, and Marta Piekarska. "Introduction to Security and Privacy on the Blockchain." In 2017 IEEE European symposium on security and privacy workshops (Euro S & P W), pp. 1-3. IEEE, 2017.
52. Zhang, Rui, Rui Xue, and Ling Liu. "Security and privacy on blockchain." *ACM Computing Surveys (CSUR)* 52, no. 3 (2019): 1-34.
53. Mohanta, Bhabendu Kumar, Debasish Jena, Soumyashree S. Panda, and Srichandan Sobhanayak. "Blockchain technology: A survey on applications and security privacy challenges." *Internet of Things* 8 (2019): 100107.
54. Konkin, Anatoly, and Sergey Zapechnikov. "Zero knowledge proof and ZK-SNARK for private blockchains." *Journal of Computer Virology and Hacking Techniques* 19, no. 3 (2023): 443-449.
55. Oude Roelink, Bjorn, Mohammed El-Hajj, and Dipti Sarmah. "Systematic review: Comparing zk-SNARK, zk-STARK, and bulletproof protocols for privacy-preserving authentication." *Security and Privacy* 7, no. 5 (2024): e401.
56. Banaeian Far, Saeed, Maryam Rajabzadeh Asaar, and Afrooz Haghbin. "A blockchain-based coin mixing protocol with certificateless signcryption." *Peer-to-Peer Networking and Applications* 16, no. 2 (2023): 1106-1124.
57. Liyanaarachchi, Gajendra, Giampaolo Viglia, and Fidan Kurtaliqi. "Addressing challenges of digital transformation with modified blockchain." *Technological Forecasting and Social Change* 201 (2024): 123254.
58. Scharfman, Jason, and Jason Scharfman. "Decentralized finance (DeFi) compliance and operations." *Cryptocurrency Compliance and Operations: Digital Assets, Blockchain and DeFi* (2022): 171-186.
59. Desmedt, Yvo G. "Threshold cryptography." *European Transactions on Telecommunications* 5, no. 4 (1994): 449-458.
60. Yu, Keping, Liang Tan, Caixia Yang, Kim-Kwang Raymond Choo, Ali Kashif Bashir, Joel JPC Rodrigues, and Takuro Sato. "A blockchain-based shamir's threshold cryptography scheme for data protection in industrial internet of things settings." *IEEE Internet of Things Journal* 9, no. 11 (2021): 8154-8167.
61. Li, Chaoyang, Bohao Jiang, Mianxiong Dong, Yuling Chen, Zhifeng Zhang, Xiangjun Xin, and Kaoru Ota. "Efficient designated verifier signature for secure cross-chain health data sharing in biomed." *IEEE Internet of Things Journal* (2024).
62. Li, Chaoyang, Bohao Jiang, Mianxiong Dong, Xiangjun Xin, and Kaoru Ota. "Privacy preserving for electronic medical record sharing in healthchain with group signature." *IEEE Systems Journal* 17, no. 4 (2023): 6114-6125.
63. Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM* 21, no. 2 (1978): 120-126.
64. Johnson, Don, Alfred Menezes, and Scott Vanstone. "The elliptic curve digital signature algorithm (ECDSA)." *International journal of information security* 1 (2001): 36-63.
65. Bisheh-Niasar, Mojtaba, Reza Azarderakhsh, and Mehran Mozaffari-Kermani. "Cryptographic accelerators for digital signature based on Ed25519." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 29, no. 7 (2021): 1297-1305.
66. Bernstein, Daniel J., Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. "SPHINCS: practical stateless hash-based signatures." In *Annual international conference on the theory and applications of cryptographic techniques*, pp. 368-397. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.
67. Chen, Lanxiang, Wai-Kong Lee, Chin-Chen Chang, Kim-Kwang Raymond Choo, and Nan Zhang. "Blockchain based searchable encryption for electronic health record sharing." *Future generation computer systems* 95 (2019): 420-429.
68. Gupta, Brij B., Kuan-Ching Li, Victor CM Leung, Kostas E. Psannis, and Shingo Yamaguchi. "Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system." *IEEE/CAA Journal of Automatica Sinica* 8, no. 12 (2021): 1877-1890.
69. Li, Chaoyang, Mianxiong Dong, Jian Li, Gang Xu, Xiu-Bo Chen, Wen Liu, and Kaoru Ota. "Efficient medical big data management with keyword-searchable encryption in healthchain." *IEEE Systems Journal* 16, no. 4 (2022): 5521-5532.

70. Chen, Lily, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray A. Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. Vol. 12. Gaithersburg, MD, USA: US Department of Commerce, National Institute of Standards and Technology, 2016.
71. Micciancio, Daniele, and Oded Regev. "Lattice-based cryptography." In *Post-quantum cryptography*, pp. 147-191. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
72. Regev, Oded. "Lattice-based cryptography." In *Annual International Cryptology Conference*, pp. 131-141. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.
73. Ducas, Léo, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "Crystals-dilithium: A lattice-based digital signature scheme." *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2018): 238-268.
74. McEliece, Robert J. "A public-key cryptosystem based on algebraic." *Coding Thv* 4244, no. 1978 (1978): 114-116.
75. Barthe, Gilles, Benjamin Grégoire, and Santiago Zanella Béguelin. "Formal certification of code-based cryptographic proofs." In *Proceedings of the 36th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pp. 90-101. 2009.
76. Banaeian Far, Saeed, and Maryam Rajabzadeh Asaar. "A blockchain-based quantum-secure reporting protocol." *Peer-to-Peer Networking and Applications* 14, no. 5 (2021): 2992-3011.
77. Butin, Denis. "Hash-based signatures: State of play." *IEEE security & privacy* 15, no. 4 (2017): 37-43.
78. Far, Saeed Banaeian, Azadeh Imani Rad, and Maryam Rajabzadeh Asaar. "Goodbye Bitcoin: A general framework for migrating to quantum-secure cryptocurrencies." In *2022 30th International Conference on Electrical Engineering (ICEE)*, pp. 512-517. IEEE, 2022.
79. Bernstein, Daniel J., Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. "The SPHINCS+ signature framework." In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pp. 2129-2146. 2019.
80. Hoffstein, Jeff, Daniel Lieman, Jill Pipher, and Joseph H. Silverman. "NTRU: A public key cryptosystem." *NTRU Cryptosystems, Inc.*(www.ntru.com) (1999).
81. Bos, Joppe, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM." In *2018 IEEE European Symposium on Security and Privacy (EuroS & P)*, pp. 353-367. IEEE, 2018.
82. Shor, Peter W. "Algorithms for quantum computation: discrete logarithms and factoring." In *Proceedings 35th annual symposium on foundations of computer science*, pp. 124-134. Ieee, 1994.
83. Long, Gui-Lu. "Grover algorithm with zero theoretical failure rate." *Physical Review A* 64, no. 2 (2001): 022307.
84. Lyubashevsky, Vadim, Ngoc Khanh Nguyen, and Maxime Plançon. "Lattice-based zero-knowledge proofs and applications: shorter, simpler, and more general." In *Annual International Cryptology Conference*, pp. 71-101. Cham: Springer Nature Switzerland, 2022.
85. Fiat, Amos, and Adi Shamir. "How to prove yourself: Practical solutions to identification and signature problems." In *Conference on the theory and application of cryptographic techniques*, pp. 186-194. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986.
86. Ganesh, Chaya, Hamidreza Khoshakhlagh, Markulf Kohlweiss, Anca Nitulescu, and Michał Zajac. "What Makes Fiat-Shamir zkSNARKs (Updatable SRS) Simulation Extractable?." In *International Conference on Security and Cryptography for Networks*, pp. 735-760. Cham: Springer International Publishing, 2022.
87. Nitulescu, Anca, and Michał Zajac. "What Makes Fiat-Shamir zkSNARKs (Updatable SRS) Simulation Extractable?." *Security and Cryptography for Networks*: 735.
88. Feneuil, Thibault, and Matthieu Rivain. "Threshold linear secret sharing to the rescue of MPC-in-the-head." In *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 441-473. Singapore: Springer Nature Singapore, 2023.
89. Sidorenko, Nikolaj, Sabine Oechsner, and Bas Spitters. "Formal security analysis of MPC-in-the-head zero-knowledge protocols." In *2021 IEEE 34th Computer Security Foundations Symposium (CSF)*, pp. 1-14. IEEE, 2021.
90. Far, Saeed Banaeian, Mohammad Reza Chalak Qazani, Seyed Mojtaba Hosseini Bamakan, Azadeh Imani Rad, and Ahad Zareravasan. "A novel Layer 2 framework for breaking the blockchain trilemma problem using MPC-in-the-Head." *Computer Networks* (2025): 111148.
91. Nowak, David. "A framework for game-based security proofs." In *Information and Communications Security: 9th International Conference, ICICS 2007, Zhengzhou, China, December 12-15, 2007. Proceedings 9*, pp. 319-333. Springer Berlin Heidelberg, 2007.
92. Burrows, Michael, Martin Abadi, and Roger Needham. "A logic of authentication." *ACM Transactions on Computer Systems (TOCS)* 8, no. 1 (1990): 18-36.
93. Canetti, Ran, Oded Goldreich, and Shai Halevi. "The random oracle methodology, revisited." *Journal of the ACM (JACM)* 51, no. 4 (2004): 557-594.
94. Lindell, Yehuda. "General composition and universal composability in secure multiparty computation." *Journal of cryptology* 22 (2009): 395-428.
95. Dolev, Danny, and Andrew Yao. "On the security of public key protocols." *IEEE Transactions on information theory* 29, no. 2 (1983): 198-208.
96. Lamster, Lukas, Martin Unterguggenberger, David Schrammel, and Stefan Mangard. "HashTag: Hash-based Integrity Protection for Tagged Architectures." In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 2797-2814. 2023.
97. Merkle, Ralph C. "A certified digital signature." In *Conference on the Theory and Application of Cryptology*, pp. 218-238. New York, NY: Springer New York, 1989.
98. Guay, Yannick, and Jean-Gregoire Morin. "System and method for digital signature and authentication." U.S. Patent Application 11/181,506, filed January 18, 2007.
99. Fiore, Dario, Rosario Gennaro, and Valerio Pastore. "Efficiently verifiable computation on encrypted data." In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 844-855. 2014.
100. Parno, Bryan, Jon Howell, Craig Gentry, and Mariana Raykova. "Pinocchio: Nearly practical verifiable computation." *Communications of the ACM* 59, no. 2 (2016): 103-112.
101. Benhamouda, Fabrice, Shai Halevi, Hugo Krawczyk, Alex Miao, and Tal Rabin. "Threshold Cryptography as a Service (in the Multiserver and YOSO Models)."

- In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 323-336. 2022.
102. Gentry, Craig, Shai Halevi, Hugo Krawczyk, Bernardo Magri, Jesper Buus Nielsen, Tal Rabin, and Sophia Yakubov. "Yoso: You only speak once: Secure mpc with stateless ephemeral roles." In Annual International Cryptology Conference, pp. 64-93. Cham: Springer International Publishing, 2021.
 103. Goldreich, Oded. "Secure multi-party computation." Manuscript. Preliminary version 78, no. 110 (1998): 1-108.
 104. Alberto Torres, Wilson Abel, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, Veronika Kuchta, Nandita Bhattacharjee, Man Ho Au, and Jacob Cheng. "Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice RingCT v1. 0)." In Information Security and Privacy: 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, July 11-13, 2018, Proceedings 23, pp. 558-576. Springer International Publishing, 2018.
 105. Sun, Shi-Feng, Man Ho Au, Joseph K. Liu, and Tsz Hon Yuen. "Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero." In Computer Security-ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II 22, pp. 456-474. Springer International Publishing, 2017.
 106. Yuen, Tsz Hon, Shi-feng Sun, Joseph K. Liu, Man Ho Au, Muhammed F. Esgin, Qingzhao Zhang, and Dawu Gu. "Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security." In Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10-14, 2020 Revised Selected Papers 24, pp. 464-483. Springer International Publishing, 2020.
 107. Thibault, Louis Tremblay, Tom Sarry, and Abdelhakim Senhaji Hafid. "Blockchain scaling using rollups: A comprehensive survey." IEEE Access 10 (2022): 93039-93054.
 108. Zamani, Mahdi, Mahnush Movahedi, and Mariana Raykova. "Rapidchain: Scaling blockchain via full sharding." In Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp. 931-948. 2018.
 109. Far, Saeed Banaeian, Azadeh Imani Rad, and Maryam Rajabzadeh Asaar. "ACR-MLM: A privacy-preserving framework for anonymous and confidential rewarding in blockchain-based multi-level marketing." Data Science and Management 5, no. 4 (2022): 219-231.
 110. Far, Saeed Banaeian, Maryam Rajabzadeh Asaar, and Afrooz Haghbin. "A privacy-preserving framework for blockchain-based multi-level marketing." Computers & Industrial Engineering 177 (2023): 109095.
 111. Han, Hongdan, Radha K. Shiwakoti, Robin Jarvis, Chima Mordi, and David Botchie. "Accounting and auditing with blockchain technology and artificial Intelligence: A literature review." International Journal of Accounting Information Systems 48 (2023): 100598.
 112. Banaeian Far, Saeed, Maryam Rajabzadeh Asaar, and Afrooz Haghbin. "A generic framework for blockchain-assisted on-chain auditing for off-chain storage." International Journal of Information Security (2024): 1-29.
 113. Ferdous, Md Sadek, Farida Chowdhury, and Madini O. Alassafi. "In search of self-sovereign identity leveraging blockchain technology." IEEE access 7 (2019): 103059-103079.
 114. Shuaib, Mohammed, Noor Hafizah Hassan, Sahnus Usman, Shadab Alam, Surbhi Bhatia, Arwa Mashat, Adarsh Kumar, and Manoj Kumar. "Self-Sovereign Identity Solution for Blockchain-Based Land Registry System: A Comparison." Mobile Information Systems 2022, no. 1 (2022): 8930472.
 115. Far, Saeed Banaeian, and Seyed Mojtaba Hosseini Bamakan. "Third layer blockchains are being rapidly developed: Addressing state-of-the-art paradigms and future horizons." Journal of Network and Computer Applications (2024): 104044.
 116. Bamakan, Seyed Mojtaba Hosseini, Amirhossein Motavali, and Alireza Babaei Bondarti. "A survey of blockchain consensus algorithms performance evaluation criteria." Expert Systems with Applications 154 (2020): 113385.
 117. Lavour, Thomas, Jérôme Lacan, and Caroline PC Chanel. "Enabling blockchain services for ioe with zk-rollups." Sensors 22, no. 17 (2022): 6493.
 118. Groth, Jens, and Victor Shoup. "Fast batched asynchronous distributed key generation." In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 370-400. Cham: Springer Nature Switzerland, 2024.
 119. Abraham, Ittai, Philipp Jovanovic, Mary Maller, Sarah Meiklejohn, Gilad Stern, and Alin Tomescu. "Reaching consensus for asynchronous distributed key generation." In Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing, pp. 363-373. 2021.
 120. Benhamouda, Fabrice, Craig Gentry, Sergey Gorbunov, Shai Halevi, Hugo Krawczyk, Chengyu Lin, Tal Rabin, and Leonid Reyzin. "Can a public blockchain keep a secret?." In Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part I 18, pp. 260-290. Springer International Publishing, 2020.
 121. Ohta, Kazuo, and Tatsuoaki Okamoto. "Multi-signature schemes secure against active insider attacks." IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 82, no. 1 (1999): 21-31.
 122. Yi, Lijang, Guoqiang Bai, and Guozhen Xiao. "Proxy multi-signature scheme: a new type of proxy signature scheme." Electronics Letters 36, no. 6 (2000): 527-528.