Nillion: A Secure Processing Layer for Web3

Miguel de Vega, Andrew Masanto, Rob Leslie,
Andrew Yeoh, Alex Page, Tristan Litre

www.nillion.com

February 2022

Abstract

First popularized by Bitcoin, blockchain-based decentralized architecture paved the way for a next-generation financial system. However, decentralized nodes can have utility beyond blockchain. This paper presents a new cryptographic primitive, Nil Message Compute (NMC), that enables a non-blockchain, decentralized network which similarly paves the way for a generational leap in decentralized processing, storage, and data, both for existing blockchains and as a native public utility. NMC takes arbitrary data and particalizes/shreds/"horcruxes" it, then distributes those particles across a network of permissionless nodes. While the particles enjoy the security of being held in a decentralized, unrecognizable/transformed, fragmented, and Information-Theoretic Secure (ITS) (i.e. post-encryption and post-quantum) manner, the underlying data can still be processed/computed on by nodes at commercially viable speeds, without the need for data reconstruction or inter-node messaging. Nillion therefore intends to become the first fully decentralized, trustless, and permissionless NMC network for web3. The success of such a network could: (1) render both computational encryption and centralized storage of private data obsolete; (2) fundamentally change the way information is stored and processed; (3) offer alternative non-blockchain ways of using decentralized nodes; (4) serve as a Meta Layer that provides new and additional functionality to existing blockchains; and (5) enable a host of new real-world applications on its native public network (the "Nillion Network").

Contents

1 Introduction

While blockchain enables a common ledger to be shared across decentralized networks, the utility of decentralized nodes can be significantly expanded. The original purpose of the blockchain per the Bitcoin whitepaper was to create an "electronic payment system based on cryptographic proof instead of trust."[1] Ethereum and similar programmable blockchain systems (e.g. Solana, Avalanche) leveraged the distributed consensus Bitcoin pioneered to enable complex systems and decentralized applications, but the blockchain was never designed to serve as an efficient mechanism for decentralized general-purpose computation. Vitalik Buterin, co-creator of Ethereum, has suggested that one solution to this mismatch could involve moving "the entire computation off the blockchain... the code still goes on the blockchain, and gets recorded there, but by default the computation is decided by oracles which run the code off-chain in a private EVM and suppl[ies] the answer."[2]

Around the same time that blockchain was conceptualized, a sibling technology purpose-built for processing and computation within a trustless environment was being developed, called Secure Multi-Party Computation (commonly abbreviated as SMPC or MPC) [1]. This technology holds the promise of Information-Theoretic Security (ITS) (a level of security that is cryptoanalytically unbreakable) [2] and distributed computation between trustless nodes, but it has seen limited practical use due to performance constraints caused by inter-node messaging. There are no fully decentralized, permissionless, and fault tolerant SMPC-based networks in existence; current SMPC implementations are permissioned instances limited to the secure storage and retrieval of secrets, simple calculations, or problems for which long processing times are acceptable.[3] However, similar to how Proof of Work facilitated a decentralized immutable ledger for transactions, Nillion's Nil Message Compute (NMC) addresses the shortcomings of SMPC and facilitates a novel (non-blockchain, non-traditional SMPC) network capable of providing decentralized computing to blockchains and developers at near client-server speed.

Decentralization holds the promise of disruption far beyond just the financial sector. Recent examples of this trend include the effect of tokenization on real-world products and services,[4] the impact of Non-Fungible Tokens (NFTs) on the art industry,[5] and the advent of Play-To-Earn (P2E) in the gaming sector.[6] These effects on multivarious industries

[1]Nakamoto Satoshi, 'Bitcoin: A Peer-to-Peer Electronic Cash System' (Bitcoin Whitepaper, 31 October 2008) <https://bitcoin.org/> accessed 31 January 2022

[2]Buterin Vitalik, 'Scalability, Part 1: Building on Top' (Ethereum Foundation Blog, 17 September 2014) <https://blog.ethereum.org/2014/09/17/> accessed 31 January 2022

[3]Existing commercial MPC implementations (e.g. Fireblocks, Synapse, Partisia) use a limited form of MPC where parties are either centralized, coupled with a blockchain, or not used as a core component of the network.

[4]Croft Jane, 'Which real-world assets are being tokenised?' (Financial Times, 30 November 2021) <https://www.ft.com/content/ac33fb51-53a4-49a0-a4c4-fb92dc6ee241> accessed 3 February 2022

[5]Waelder Paul, 'Why We Keep Talking About NFTs' (*CCCB LAB*, 14 December 2021) <https://lab.cccb.org/en/why-we-keep-talking-about-nfts> accessed 1 February 2022

[6]Binance Academy, 'What Is Play-to-Earn and How to Cash Out?' (Binance Academy, 14 January

are indicative of the fact that as decentralized computing evolves, so will its impact on both established and emerging sectors. Commonly cited candidates to be impacted by decentralized technologies include voting and governance (both in real-world[7] and Decentralized Autonomous Organization (DAO)[8] contexts), payments,[9] identity,[10] social media,[11] biometrics,[12] Internet-of-Things (IOT),[13] among many others. Even players in the decentralized ecosystem find themselves in a constant state of disruption, with new innovations in scaling, cross-chain interoperability, privacy, data storage, and more becoming increasingly relevant every day. The evolution of decentralized computation clearly holds great promise for future disruption.

Nillion intends to become the first fully decentralized, trustless, and permissionless NMC network for web3. The network's native token will initially be implemented using an Ethereum ERC-20, powering a public utility providing fast (near client-server speed), secure (post-quantum), private (ITS), chain-agnostic (universally available base layer infrastructure), and leaderless (fully decentralized) processing, with the objective of both supercharging existing blockchains and offering native network services with properties unique to a new cryptographic primitive. Such a network will enable blockchains, users, and developers to push decentralized technology forward both in the industries mentioned above and in other ways that are yet to be imagined.

2022) <https://academy.binance.com/en/articles/what-is-play-to-earn-and-how-to-cash-out> accessed 1 February 2022

[7]Boucher Philip, 'What if blockchain technology revolutionised voting?' (EPRS — European Parliamentary Research Service, September 2016) <https://www.europarl.europa.eu/RegData/etudes/ATAG/2016/581918/EPRS_ATA(2016)581918_EN.pdf> accessed 1 February 2022

[8]Wright Aaron, 'The Rise of Decentralized Autonomous Organizations: Opportunities and Challenges' (Stanford Journal of Blockchain Law & Policy, 30 June 2021) <https://stanford-jblp.pubpub.org/pub/rise-of-daos/release/1> accessed 1 February 2022

[9]Dixon Denelle, 'How blockchain technology is fixing payments today and what comes next' (World Economic Forum, 29 April 2021) <https://www.weforum.org/agenda/2021/04/how-blockchain-technology-is-fixing-payments-today-what-comes-next/> accessed 1 February 2022

[10]Huillet Marie, 'Blockchain identity market to grow $3.58B by 2025, report claims' (Cointelegraph, 2 August 2021) <https://cointelegraph.com/news/blockchain-identity-market-to-grow-3-58b-by-2025-report-claims> accessed 1 February 2022

[11]Goertzel Ben, 'Social Networks Are the Next Big Decentralization Opportunity' (Coindesk, 22 January 2021) <https://www.coindesk.com/tech/2021/01/22/social-networks-are-the-next-big-decentralization-opportunity/> accessed 1 February 2022

[12]Hersey Frank, 'Integration of blockchain and biometrics to redefine digital identity by 2030: report' (Biometricupdate, 12 November 2021) <https://www.biometricupdate.com/202111/integration-of-blockchain-and-biometrics-to-redefine-digital-identity-by-2030-report> accessed 1 February 2022

[13]Dickson Ben, 'Decentralizing IoT networks through blockchain' (Techcrunch, 28 June 2016) <https://techcrunch.com/2016/06/28/decentralizing-iot-networks-through-blockchain/> accessed 1 February 2022

2 History and Context

2.1 Blockchain, Secure Multi-Party Computation, and Nil Message Compute

First developed in the 1980s, SMPC is a cryptographic protocol that distributes an algorithm across multiple parties where no individual party can see the other parties' data. By contrast to blockchain, which was developed with the primary objective of immutable storage and facilitating fault tolerant processing of transactions, SMPC was developed with the primary objective of achieving secure computation between distributed nodes (with each node's inputs remaining completely private).

With SMPC, the parties agree on the function to be computed, and then follow the SMPC protocol to collaboratively compute the output without revealing their secret inputs. The idea of secure computation was introduced by Andrew Yao within the context of two-party computations (which came to be known as Yao's Millionaires' problem [3]), and it was further developed by Goldreich, Micali, and Wigderson (GMW) to include multiple parties [4]. These protocols guarantee confidentiality as long as the parties follow the protocol, providing *passive security*. The authors also provided an enhanced version of GMW that is secure against adversaries who might deviate from the SMPC protocol, introducing the notion of *active security*. A rudimentary description of the distinction between SMPC and blockchain computation is that SMPC breaks up the desired computation and distributes it among multiple nodes, where the output of each node can be checked for correctness using Verifiable Secret Sharing techniques [5], while blockchains typically run the entire function on a single node (with verifiers) or the entire function on multiple nodes, using a consensus protocol to verify correctness. Each executed base function, once verified, is then immutably recorded on a blockchain. With SMPC, the primary objective is to securely distribute the processing over a multiplicity of nodes, with computational correctness embedded in the result, rather than the primary objective being the secure execution and immutable storage of transactions, as in the case of blockchain. In other words, the decentralized processing of transactions, with the nodes breaking up and sharing the required processing, is not something blockchains were natively designed to do.

The main issue with SMPC is its performance; SMPC computations can easily take hours, as they require the exchange of a large number of messages between nodes [6]. For this reason, since the late 2000s, SMPC-related efforts have largely been focused on generating efficiency improvements in protocols with particular applications in mind, such as distributed voting, bidding and auctions, the sharing of signatures or decryption functions, and hidden information retrieval. Consequently, SMPC has generally been considered a practical solution to a limited set of real-world problems, including those that require only the secure storage and retrieval of secrets or very simple computations, or for which long processing times are acceptable.

With two very recent public announcements by SMPC companies, namely, Coinbase's

acquisition of Unbound Security, reportedly in excess of $150 million, and Fireblocks' Series E private raise, reportedly at an $8 billion valuation (having risen from $2 billion just four months earlier), it is clear that SMPC is state-of-the-art technology for the secure storage and processing of both data and digital assets. Commenting on the Unbound acquisition, Coinbase stated, "Crypto can't grow without strong cryptography and strong security. Secure Multi-Party Computation is an application of advanced mathematics to enable crypto assets to be stored, transferred, and deployed more securely, easily, and flexibly than ever before. The crypto-economy is growing exponentially with myriad new use cases such as staking, DeFi, DAOs, and NFTs. Unfortunately, so are the threat vectors and complexities for participants to safely manage their crypto private keys. Technologies such as MPC will enable these groundbreaking use cases to come to life safely, securely, and in a way that's user friendly."[14] SMPC is also increasingly becoming the technology of choice for crypto protocols seeking to develop decentralized, interoperable, cross-blockchain capabilities.[15][16] Clearly, SMPC has interesting proven applications, albeit in narrow use cases.

Nillion expands state-of-the-art SMPC and blockchain capabilities using a new cryptographic primitive called Nil Message Compute. NMC takes a generational leap away from SMPC by expanding beyond the use of a Linear Secret Sharing scheme such as Shamir's Secret Sharing [7]. Shamir's Secret Sharing allows for the distribution of a secret into fragmented shares across a network of nodes. The nodes, which are typically permissioned or owned by the same company or group of companies, must then exchange messages in order for the shares to be used in the context of a computation. As previously mentioned, while theoretically interesting, SMPC has been limited in commercial viability as the computation step of data with SMPC can take hours. In contrast, NMC makes it possible to use fragmented, sensitive data without the need for inter-node messaging during computations (see Section 3). Decentralized computations can therefore occur at commercially viable speeds, approximately equivalent to those of centralized client-server interactions in plaintext. This method of decentralized computation enables a new host of decentralized use cases, both for existing blockchains and as an independent public network.

[14]Orbach Meir, 'Crypto Giant Coinbase Acquiring Israel's Unbound Security' (CTECH, 30 November 2021) <https://www.calcalistech.com/ctech/articles/0,7340,L-3923738,00.html> accessed 1 February 2022

[15]Synapse Protocol, 'What Is Synapse?' (Welcome to Synapse - Synapse Protocol) <https://docs.synapseprotocol.com/> accessed 1 February 2022

[16]Bluehelix, 'Bluehelix joins Polygon Ecosystem as an MPC- based Cross-Chain Bridge Solution Provider' (Bluehelix Guide) <https://bhexchain.gitbook.io/bhex-chain/v/english/announcements/bluehelix-joins> accessed 1 February 2022

2.2 The Digital World and the Metaverse Need a Decentralized "Fort Knox"

The Metaverse has been described as the combination of multiple elements of technology that allow users to "live" within a digital universe.[17] As an ever-increasing part of our lives and economic value transitions onto this digital ecosystem or Metaverse, the incentives for anonymous and well-equipped attackers increase exponentially. The stakes for this cat-and-mouse game have never been higher. From private key storage, to sensitive financial information, to corporate secrets, to protocol funds access, to customer databases, the quantity and variety of exploitable, valuable digital data and assets is significant – so significant that it has recently come to light that criminal enterprises and nation states are saving hashed or encrypted data with a view to decrypting it as quantum computers become available in the future.[18]

To add to this, the status quo based on the centralized storage of data and access to digital assets has often fallen short of providing adequate protection. Digital hacks and exploits are endemic and advances in malicious technology, hacking techniques, and computing power have significantly increased the abilities and incentives for bad actors to access information protected by even the most advanced security and encryption systems [8]. Perhaps most concerningly, despite the emergence of a decentralized Metaverse, almost all password and private key storage systems used to access this Metaverse are centralized and suffer from the issue of having single points of failure.

Clearly, as the Metaverse becomes an increasing part of human life, there is an immediate opportunity present for a technology such as Nillion to become the default, decentralized, and post-quantum source of digitally secure processing and storage for all kinds of data.

It is worth noting that pioneers in decentralized file storage, such as Storj, Filecoin, and Arweave, address data storage use cases that optimize for the efficiency and cost of data storage alone, rather than optimizing for *secure* data storage while retaining the ability to process/compute the data without it being decrypted, leaked, or revealed. There are also many other non-storage related implications and use cases of Nillion (see Section 5). These existing file storage protocols thus constitute and address partially overlapping but significantly different markets and use cases.

[17] Snider Mike and Molina Brett, 'Everyone wants to own the metaverse including Facebook and Microsoft. But what exactly is it?' (USA Today, 20 January 2022) <https://eu.usatoday.com/story/tech/2021/11/10/metaverse-what-is-it-explained-facebook-microsoft-meta-vr/6337635001/> accessed 1 February 2022

[18] Goldman P. David, 'Quantum hackers can bring down Bitcoin: expert' (Asia Times, 2 December 2021) <https://asiatimes.com/2021/12/quantum-hackers-can-bring-down-bitcoin-expert> accessed 1 February 2022

2.3 Privacy, Private Blockchains, and Decentralized Private Data

Zero-Knowledge Proof and Nil Message Compute

Developments in blockchain over recent years have seen ZKP emerge as the preferred technology to achieve privacy in a decentralized context. A ZKP allows a "prover" to prove to a "verifier" that a given statement based on the prover's data is true without conveying any information apart from the fact that the statement is indeed true. This is extremely useful in the context of blockchain technology (as evidenced by protocols such as Zcash and Mina). However, NMC is not blockchain technology. Consequently, while ZKP relates to proofs of the state of data that one party has in their possession, NMC delivers rich, privacy-preserving computational capabilities for data that any number of parties can hold. While this provides a basic juxtaposition of ZKP and NMC, a more comprehensive description and technical comparison is given in Section 3.3.

Private Blockchains

Multicoin Capital contends that decentralized solutions that offer *privacy-as-a-service*, rather than blockchains that are *private-by-default* (e.g. Zcash, Monero), are the likely future of decentralized privacy.[19] This is because: (1) privacy-by-default blockchains such as Zcash suffer from auditability concerns, as it is impossible for the public to verify that inflation or double spend has not occurred in private or shielded transactions;[20] (2) private Layer 1 solutions are often expensive to use, have scalability issues, and can be limited regarding smart contract capabilities; and most importantly, (3) private Layer 1 solutions require migration from blockchains that have been more successful in achieving network effects (e.g. Ethereum). So far, such trade-offs have prevented privacy-by-default Layer 1 solutions from achieving the same traction as their public blockchain counterparts.

While Nillion incorporates cryptoanalytically unbreakable privacy (see Section 3.2), unlike Layer 1 privacy protocols, Nillion's objective is not to provide another privacy-by-default Layer 1 solution. Instead, Nillion seeks to create new and improved decentralized processing functionality for existing blockchains and native applications on Nillion as its primary objective, with privacy being an ancillary (but valuable) benefit. To this end, Nillion's processing-related benefits and the associated use cases are explored in Section 5.

Decentralized Private Data

If it has been accepted that decentralized public networks provide optimal security and trust for the sending, storage, and computation of *value* (e.g. Bitcoin, Ethereum), then why should a decentralized public network not also be the most secure and trusted solution

[19] Gentry Ryan, 'Multicoin Capital: Privacy Is a Feature, Not a Product' (Multicoin Capital, 24 September 2019) <https://multicoin.capital/2019/09/24/privacy-is-a-feature/> accessed 1 February 2022

[20] Nillion will support private storage and processing auditing functionality, which will be optionally offered to the developer community. This could be relevant in scenarios such as services and products operating in the regulated sector, token transaction auditing, and more.

for the sending, storage, and computation of private *data*? Beyond financial transactions, the lack of privacy on blockchains limits the potential expansion of current blockchain systems in use cases where private data is at stake. Certain industries and use cases require privacy if they are to adopt decentralized solutions. For example, when blockchain is used in scenarios such as Mobile CrowdSourcing (MCS) or Internet of Things (IoT), direct privacy leakage due to transaction exposure becomes a crucial issue. By analyzing transaction graphs, adversaries could obtain a correlation between transaction addresses and infer a user's real identity from public blockchain data [9]. The NFT industry could also benefit from private or gated decentralized data, as public NFT metadata has previously led to the undermining of mint and reveal processes. Additionally, the lack of provably private, randomized metadata forces users to trust teams not to leak or maliciously trade on hidden information for their own advantage. There has been speculation that many large NFT drops have fallen prey to metadata manipulation or insider trading, such as the allegations surrounding the recent high-profile Mekaverse launch.[21]

Furthermore, the standard hashing and encryption used for personal private data are also largely inadequate from a technological, legislative, and compliance perspective. For example, in Europe, the General Data Protection Regulation (GDPR) applies to pseudonymized data that can be traced back to an individual person. The hashing of personal data such as an ID card or medical record accomplishes only pseudonymization, and not anonymization, and the European Union has thus determined that biometrics and other identity data stored using hash functions still qualify as "identifiable" (meaning that they are subject to the legal obligations of a Data Controller as defined in the Regulation).[22] Furthermore, even with a hash function, given enough time and/or computing power, reversibility is not only possible, but likely. Consequently, the immutable storage of sensitive personal data that is confirmed to be correct via a hash function is not ideal for the security of personal private data over time.

In contrast, all data on the Nillion Network is transformed via a masking and secret sharing function, which is mathematically proven to be ITS, *before* it is distributed over the network. It is therefore impossible to leak information, as what is stored in the network contains no trace of the original data. This addresses the issues in the MCS, IoT, and NFT examples presented above and provides a novel solution for the secure management of private data on a public network. Moreover, unlike a hash function, transformed data on Nillion has no pseudonymous identifier that leaks information, and it is mathematically provable that no amount of computing power could discover the underlying data (unlike inverting a hash). These properties mean that Nillion natively handles private data in a GDPR compliant way, enabling a range of decentralized regulatory and compliance tools.

[21] Hayward Andrew, 'MekaVerse Ethereum NFT Rollout Dogged by Fraud Allegations' (Decrypt, 15 October 2021) <https://decrypt.co/83600/mekaverse-ethereum-nft-rollout-dogged-fraud-allegations> accessed 1 February 2022

[22] Zetoony A. David, 'What is 'hashing', and does it help avoid the obligations imposed by the new privacy regulations?' (GT Law, 31 March 2021) <https://www.gtlaw-dataprivacydish.com/2021/03/what-is-hashing.> accessed 1 February 2022

2.4 Processing and the Evolution of Decentralized Ecosystems

The mining of Bitcoin's genesis block in 2009 heralded the beginning of the decentralized ecosystem. Ethereum's release in 2015 was another watershed moment, enabling the otherwise simple transactions that Bitcoin relied on to act as "smart contracts" that allowed developers to craft complex protocols constrained only by the processing available on-chain. Where Bitcoin sought to be a public financial ledger, Ethereum took steps towards becoming the "world computer." The growing popularity of Ethereum has led to network congestion and soaring gas prices that have pushed premiums for block space and compute sky-high. Protocol developers must wrestle with the technical constraints of block rate limits and pricing that limits their access to processing. As use cases on the network become increasingly complex, the limitations placed on protocols by the price and capacity of Ethereum's processing have become clear. Another boom of new players, some touting themselves as "ETH killers," are challenging the giant with a general focus on being faster, cheaper, or more efficient blockchains. While these may offer some advantages against Ethereum, the state of on-chain processing today is still limited by the architecture of the blockchain.

3 The Core Technology: Nil Message Compute

The NMC protocol powers Nillion's decentralized and scalable public utility for the secure and fast information processing. In this section, we go through background context and then progressively explain the different elements comprising the NMC protocol.

Section Structure

Figure 1 illustrates how Section 3 is structured. At a high level, we divide the description into three blocks of sections. The first block begins by covering standard encryption, transitions through SMPC and ends by introducing NMC, Nillion's core technology. The second block explores NMC's performance, decentralization, and security characteristics – the three elements in the *blockchain trilemma* [10]. The third block focuses on processing, which is NMC's core capability.

We begin in Section 3.1 by analyzing encryption as a building block underpinning security in most modern systems. In Sections 3.2 and 3.3 we then introduce the concept of ITS, which enables the design of cryptographic protocols exhibiting security properties superior to those based on widely-deployed encryption and ZKP technologies. As prominent examples of such ITS protocols, a number of SMPC protocols, such as SPDZ or BGW, provide a general-purpose processing capability. However, the problem with SMPC protocols (whether ITS or not) is their scalability. In particular, processing non-trivial algorithms requires inter-node messaging, which has become a performance bottleneck preventing SMPC from reaching mass adoption. In Sections 3.4, 3.5, and 3.6 we introduce NMC, a new technology that solves this key problem. NMC's core innovation uses two cryptographic primitives in tandem in order to provide an ITS general-purpose processing capability that scales and performs very close to plaintext centralized processing.

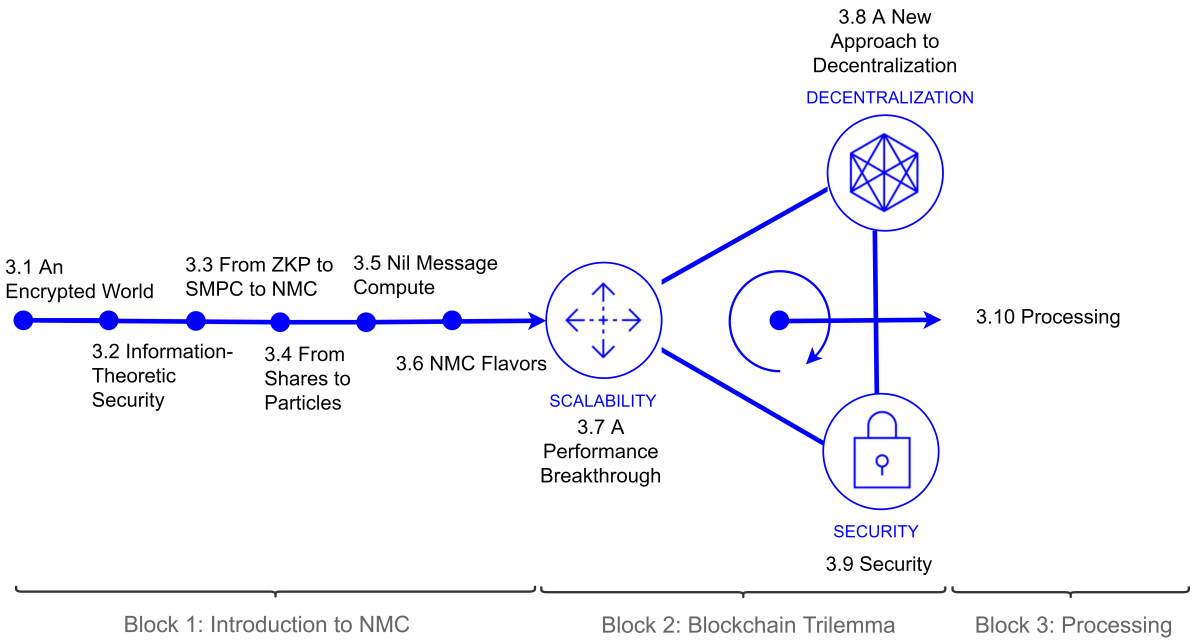

Figure 1: Structure of Section 3. Sections 3.1–3.6 (Block 1) introduce the NMC protocol. Sections 3.7–3.9 (Block 2) explore the blockchain trilemma applied to NMC, and Section 3.10 (Block 3) discusses NMC's main capability: processing.

Specifically, the NMC Nodes can process non-trivial algorithms without inter-node messaging, which allows the decentralized network to process computations at CPU speed regardless of its size. In Section 3.7, we explore NMC's theoretical performance, which can be many orders of magnitude faster than SMPC. In Section 3.8, we explore how NMC can be used to build decentralized trustless networks, and in Section 3.9, we present the main security features of the protocol. All of this leads up to Section 3.10, in which we present NMC's processing capability as it will be experienced and consumed by the development community.

3.1 An Encrypted World

All of the most prevalent solutions used to secure sensitive information are based on encryption. However, encryption is not a silver bullet; numerous attacks on it exist, including cryptographic attacks, attacks on encryption keys, insider attacks, stolen ciphertext attacks, data destruction, corruption or integrity attacks, and ransomware attacks. Moreover, many widely-used and well-established cryptographic algorithms, such as RSA-based encryption/signatures and elliptic curve cryptography, are not quantum safe.

Encryption presents several vulnerabilities that make these attacks possible:

- First, a set of encryption keys (typically containing just one key) is used to decrypt an encrypted message. Therefore, encryption addresses the problem of preserving

data confidentiality by creating the new problem of securing the set of encryption keys.

- Second, the security of encryption is considered largely in the computational setting, meaning it is based on cryptographic hardness assumptions. If an attacker does not have access to the set of encryption keys, the underlying security of the encryption primitive depends on the hardness of a mathematical problem, such as the factoring of a product of two large prime numbers. As hard as the problem may be, it is always possible to solve it. For example, though the difficulty of factoring the product of two prime numbers increases as a function of the number of decimal digits of that product, it never becomes unsolvable [11].

- Third, traditional encryption can only encrypt data when it is stored or sent between Information-nodes but not when it is processed by a node. In such a case, the data needs to be decrypted first, processed in plaintext, and then encrypted again, opening an important attack vector.

Homomorphic Encryption (HE) [12][13] is an emerging technique that allows data to be processed in encrypted form, addressing the third vulnerability described above. However, HE generally incurs high computational costs when applied to general Turing-complete computations, and it still exhibits the other vulnerabilities present in standard encryption.

3.2 Information-Theoretic Security

The security of most cryptographic building blocks depends on the hardness of one or several mathematical problems. This is the approach taken by encryption and of ZKP primitives with practical applications, such as ZK-SNARKS or Bulletproofs, and means that their security depends on the computational power of an adversary and can be broken over a sufficiently long enough timeframe.

Sensitive information is transferred, stored, and processed in Nillion using the strongest form of security known in cryptography, known as Information-Theoretic Security. With ITS, sometimes called unconditional security, the cryptography cannot be broken even if an adversary has unlimited computing resources. In particular, unlike encryption and practical ZKP implementations, Nillion does not make use of computational hardness assumptions, such as factoring products of large prime numbers, the Discrete Logarithm,[23] or the Knowledge of Exponent[24] assumptions. As an ITS system, Nillion's cryptographic building blocks are considered cryptoanalytically unbreakable, as adversaries do not have,

[23] Analogously to logarithms for real numbers, given two group elements a and b, the discrete logarithm of a is an integer k such that $b^k = a$. The assumption is that for certain groups it is hard to find k given a.

[24] If a party is given $(a, a' = a^c)$, where c is unknown, then the assumption is that the only computationally feasible way for the party to return (b, b') such that $b' = b^c$ is that the party has exponentiated a and a' to the same power.

and will never have, sufficient information to break them. This means that Nillion is safe even against future quantum attacks regardless of their computing power. This is a key differentiator when comparing Nillion to current state-of-the-art privacy or confidential blockchain technology (used in protocols such as Mina, Zcash or Monero). Notice that we are referring here to attacks directly targeting Nillion's cryptographic building blocks rather than to the security of the system as a whole. Hacking a system like Nillion would require an expensive non-trivial coordination effort which only gets harder as more nodes are added to the network (see Sybil attacks below and Section 4.2 Permissionless Nodes).

At its core, Nillion uses and combines two ITS cryptographic building blocks. The first is a *One-Time Mask* (OTM), which transforms a secret into a *particle* by making use of random *blinding factors*. The second is *Linear Secret Sharing* (LSS) based on a threshold of $T + 1$, which transforms the blinding factors into *shares* distributed among Nillion nodes.[25] On its own, a share does not reveal any information about the original blinding factor – it will only do so if combined with sufficient shares from other Nillion nodes. Specifically, any subset of T or less Nillion nodes will be unable to reconstruct the blinding factor under ITS; at least $T + 1$ shares are required to do so. Note that Nillion's LSS is based on Shamir's Secret Sharing and will be covered in greater depth later on in the paper.

The following is a summary of the advantages of ITS Nillion technology when compared to encryption:

- Encrypted data is *one hack away*, whereas Nillion data is $T+1$ *hacks away.* For the former, an attacker is required only to compromise the security of a single system to access the encrypted data. In contrast, since the blinding factors masking the data under ITS in Nillion are not in any one particular place, an attacker is required to hack a large proportion of all the nodes (which are themselves secured by ITS)[26] in the Nillion Network to be able to access its data (see Section 3.4). Therefore, attacks in SMPC networks (and also in Nillion) such as Sybil attacks, collusion between nodes, and bribery, typically involve controlling a large ($T + 1$ or more) group of nodes. These attacks also apply to other decentralized networks, such as blockchains.

- Encryption relies on cryptographic assumptions and can be broken. Nillion relies on ITS and is cryptoanalytically unbreakable.

- Typically, encrypted data cannot be processed. It needs to be decrypted first, which opens an important attack vector (HE is an exception to this, but as addressed above, it tends to be computationally expensive and relies on cryptographic

[25] $T + 1$ is used to mathematically represent crossing the threshold of number of shares required to reconstruct confidential data held in Nillion.

[26] The Nillion Network will itself be used to secure and establish access to network nodes, thus creating a fully self-secured ITS network.

assumptions). In Nillion, data can be transferred, stored, and processed while being kept perfectly confidential and private.

- In traditional encryption, key management is typically a complex task that involves securing the generation, distribution, and storage of secret keys. Nillion solves this by allowing nodes to generate, distribute, and store shares of the blinding factors (which are equivalent to a key), and these are protected by the ITS properties of LSS.

We thus believe that Nillion will render traditional encryption-based networks obsolete. Once a secret is stored in Nillion, users will be able to authenticate and retrieve data and authorize processing as needed. Multi-Factor Authentication (MFA) will be used, combining biometrics, multi-signature schemes, device information, geographic location, and potentially many other verification factors adopted by the network and selected by the individual user. Due to the nature of the previously described blind computation of transformed information, nodes in the Nillion Network will perform this key security and gating function without any knowledge of what they are computing.

3.3 From Zero-Knowledge Proof to Secure Multi-Party Computation to Nil Message Compute

In this section, we provide a brief comparison between ZKP, SMPC, and NMC.

ZKP is currently the main cryptographic primitive providing privacy to blockchain technology, and it underpins many significant privacy tokens and private smart contract enabled protocols. ZKP allows a node (the "prover") to prove to another node (the "verifier") that a given statement based on the data they hold is true without sharing this data. Each cryptographic primitive is built for a different purpose and serves different goals.

SMPC is a decentralized protocol that allows multiple parties to collaboratively store and process data while guaranteeing data confidentiality. In addition, some SMPC protocols are ITS (such as BGW [14], GMW [4], and SPDZ [15]). However, the main problem with SMPC is its scalability. Processing confidential data in an SMPC network requires the nodes to exchange a large number of messages when the computation is not trivial. This can easily lead to execution times that take hours. This lack of scalability has been the main factor preventing SMPC from use in many real-world applications.

NMC is the cryptographic foundation underlying Nillion, as it is better suited than ZKP or SMPC for the decentralized storage and processing of confidential data. NMC allows for the implementation of a decentralized, private, performant, fast, Turing-complete, ITS, and versatile public network, enabling a range of interesting new use cases.

The differences between ZKP, SMPC protocols using LSS, and NMC are illustrated in Table 1.

	ZKP	SMPC	NMC
Can transfer, store, and process data while preserving its confidentiality	No. ZKP can only generate proofs from data and the processing of data, which the prover needs to have access to in plaintext. ZKP is not designed to transfer, store, or process the actual data. Proofs are simple true or false statements.	Yes.	Yes.
Preserves data confidentiality for any number of data sources	No. With ZKP, the prover that generates the proof needs to have access to the data in plaintext; thus, confidentiality with more than one data source is not possible.	Yes. SMPC can combine data from several sources and guarantee the confidentiality of each source's data.	Yes. NMC can combine data from any number of sources and guarantee the confidentiality of each source's data.
Is Information-Theoretic Secure	Instantiations of ZKP with practical applications are typically not ITS.	Some SMPC protocols with practical applications are ITS such as BGW [14], GMW [4], and SPDZ [15].	Yes. NMC is ITS.
Can process data without inter-node messaging	Does not apply as ZKP cannot process confidential data but rather generates proofs from data that has been processed in plaintext.	No. Certain operations (e.g. multiplications, AND gates) require each node in SMPC to exchange information with every other node in the network.	Yes. NMC nodes do not need to exchange messages in order to process confidential data. This means processing takes place at essentially plaintext centralized speed.

Table 1: Main differences between ZKP, SMPC protocols using LSS, and NMC.

3.4 From Shares to Particles

LSS is a cryptographic primitive that transforms a confidential statement into N numbers called *shares*. The N shares are distributed across N SMPC nodes such that:

- Every node gets a unique share,
- Any group of $T + 1$ nodes (where $T + 1 \leq N$) can reconstruct the confidential statement, and
- Any colluding group containing strictly less than $T+1$ nodes cannot deduce anything from the shares they hold.

In what follows, we refer to an *adversary* as a bad actor controlling one or several Nillion nodes, either through direct ownership of the nodes (e.g. through a Sybil attack, see Section 4.2) or because they have been able to gain unauthorized access to them. We also refer to Nillion's technology as needing to satisfy the following requirements:

- *Correctness*: The result reconstructed at the end of the computation is the same one that would have been obtained if the processing of the input data had taken place in plaintext.
- *Passive security*: No group of less than $T+1$ NMC Nodes is able to reconstruct the confidential statement under ITS. Adversaries who try to reconstruct the confidential statement from the information contained in the nodes they control but follow the NMC protocol are called *passive adversaries*.
- *Active security*: The NMC network can tolerate a certain number of nodes deviating from the protocol. This includes nodes sending the wrong shares for the result reconstruction, or not sending shares at all, or delaying their transmission with unbounded message delays. These types of adversaries are called *active adversaries*.

Some of the most powerful SMPC protocols are based on an LSS primitive. An SMPC computation with LSS typically proceeds following three steps:

- Step 1: The inputs to the computation are transformed into shares, which are distributed across the SMPC nodes.
- Step 2: The actual computation takes place, which typically involves the exchange of messages between the SMPC nodes. At the end of this step, every node will have a share from each one of the computation output values.
- Step 3: The result shares are sent to one or several Result Nodes, which run LSS to reconstruct the outputs.

The scalability problems associated with SMPC originate from the inter-node communication in Step 2, which is very time-consuming. To address this, NMC uses and combines two cryptographic primitives:

- *One-Time Mask*, which supports passive ITS and achieves correctness, and
- *Linear Secret Sharing*, which supports passive ITS and active security.

By combining these two cryptographic primitives into a divide-and-conquer approach, OTM can be optimized for both correctness and efficiency, removing the need for NMC Nodes to exchange any messages to perform a computation, thus avoiding SMPC's scalability problem. Specifically, OTM does not need to be designed to operate in the presence of active adversaries as this feature is provided by the other cryptographic primitive, LSS. Thus, there are more degrees of freedom available for the design of OTM, which are used to eliminate the need for inter-node communication during the computation step. This has very important practical consequences, increasing performance to centralized compute speed in plaintext (see Section 3.7) and allowing for consensus to be reached between the NMC Nodes without inter-node communication (see Section 3.8), while exhibiting best-in-class security (see Section 3.9). Some additional detail on these cryptographic primitives is provided below:

- OTM is an ITS cryptographic primitive that makes use of a series of random numbers called *blinding factors* to mask a secret, similar to one-time-padding or ITS Message Authentication Codes [14]. OTM is designed to deliver correctness with efficiency, meaning NMC Nodes are not required to exchange any messages to perform a computation. This means that NMC does not suffer from SMPC's scalability problem, which has very important practical consequences, as discussed in Section 3.7.
- The LSS scheme is based on Shamir's Secret Sharing, a well-known primitive that hides the blinding factor in a polynomial $p(x)$ of degree T, such that $p(0)$ is equal to the blinding factor and the shares that correspond to N evaluations of the polynomial at abscissae different than $x = 0$. In SMPC protocols such as BGW, LSS is directly used to hide confidential data. In contrast, in NMC, LSS is used to hide the blinding factors for the OTM primitive.

Figure 2 illustrates how OTM and LSS operate in tandem:

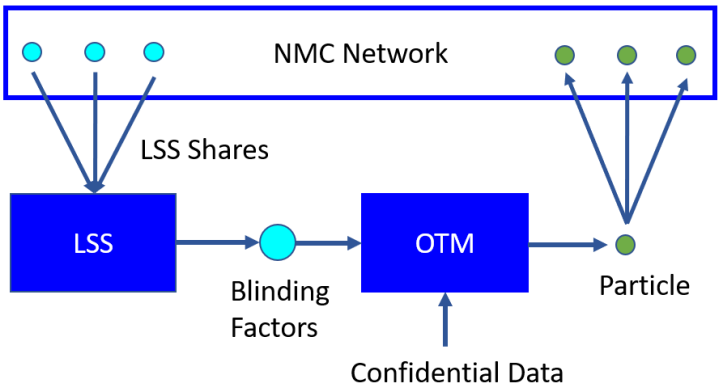

Figure 2: NMC makes use of two ITS cryptographic primitives to handle confidential data.

To provide an illustration through the use of a practical example, let us assume that an end user (e.g. an individual on a mobile device or a corporation operating from a company server) wishes to store confidential data in the NMC network. The end user first recalls the shares from a fresh set of blinding factors from the NMC network and locally reconstructs them using LSS. The end user then uses the reconstructed blinding factors to compute a masked version of the confidential data using OTM. This masked version is what we call a *particle*.

Since OTM is an ITS cryptographic primitive, the particles hide the confidential data perfectly. The particles are then sent back to the NMC network and broadcasted to the NMC Nodes, which can use them to perform computations on the confidential data. Every data element is transformed into a particle and every NMC Node receives a copy of this particle. NMC therefore meets the three core requirements as described below:

- *Correctness*: OTM guarantees this property. Since OTM focuses only on correctness, it is optimized such that the operations that have to be performed on the plaintext inputs for a computation can be performed locally and in parallel by each NMC node on the particles derived from those inputs. That is, the NMC Nodes do not require the exchange of any messages during Step 2. This innovation allows NMC to run at close to centralized server speed.

- *Passive security*: As long as the blinding factors in OTM are secret, masked confidential data cannot be reconstructed under ITS. Hence, the passive security of NMC is directly provided by that of the blinding factors – this is guaranteed by LSS. No group of less than $T + 1$ NMC Nodes is able to reconstruct the blinding factors under ITS.

- *Active security*: After performing operations on the NMC particles, each node obtains a masked version of the computation output, with the blinding factors for the

OTM being a combination of those used to mask the inputs of the computation. Therefore, if the input blinding factors are reconstructed, then the output of the NMC computation can be obtained from *any* of the output particles. Otherwise this operation is not possible due to the ITS of the OTM primitive. Hence, it is possible to use a Verifiable Secret Sharing version of the LSS in order to reconstruct the input blinding factors even when a certain number of NMC Nodes are deviating from the protocol. Once they are reconstructed, they can be applied to any particle to unmask the output from the computation.

3.5 Nil Message Compute – Confidential Computations at Plaintext Speed

In this section, we introduce the various elements in an NMC network and the method by which NMC makes use of particles to perform decentralized computations on confidential data at plaintext, centralized server speed.

The main roles of the nodes participating in an NMC computation are as follows:

- **Dealer Nodes**: These are nodes that transform their confidential and private inputs to the NMC computation into particles and distribute them across the NMC Nodes. For instance, a Dealer Node can be an individual using their mobile device, or a corporation using a company server.
- **NMC Nodes**: These nodes store the particalized inputs provided by Dealer Nodes and perform the actual NMC computations on them.
- **Result Nodes**: These nodes reconstruct the results from a finished NMC computation.

The NMC Nodes form a network of processors that provide secure communication links between processors that are directly connected to each other.

Figure 3 shows the data flow in a generic NMC computation. The problem with multi-party computations can be reduced to the problem with multi-party function evaluations [4]. Functions can be described in the *Boolean setting* as a circuit of logic gates (e.g. ANDs, ORs, XORs) or in the *arithmetic setting* as a circuit of addition and multiplication gates. NMC operates in the arithmetic setting running what we call *Multi-Party Programs* (MPPs). Thus, the purpose of the network is to jointly compute the result from evaluating an arithmetic function f over a set S of secret input values without revealing them. These values $\{s_0, s_1, \ldots, s_{s-1}\}$ with $s_i \in \mathbb{Z}/p\mathbb{Z}$ for a large prime p, are secret inputs from a set D of Dealer Nodes. Specifically, each Dealer Node may contribute one or more values to the joint evaluation of the function f. Dealer Nodes aim to keep their values secret at all times. Thus, they use different blinding factors with an ITS OTM cryptographic primitive to hide every secret, creating particles. The particles are then broadcasted to the network. The NMC Nodes carry out the actual function evaluation using only the particles from the secret inputs, which prevents them from seeing the actual secret inputs. Following

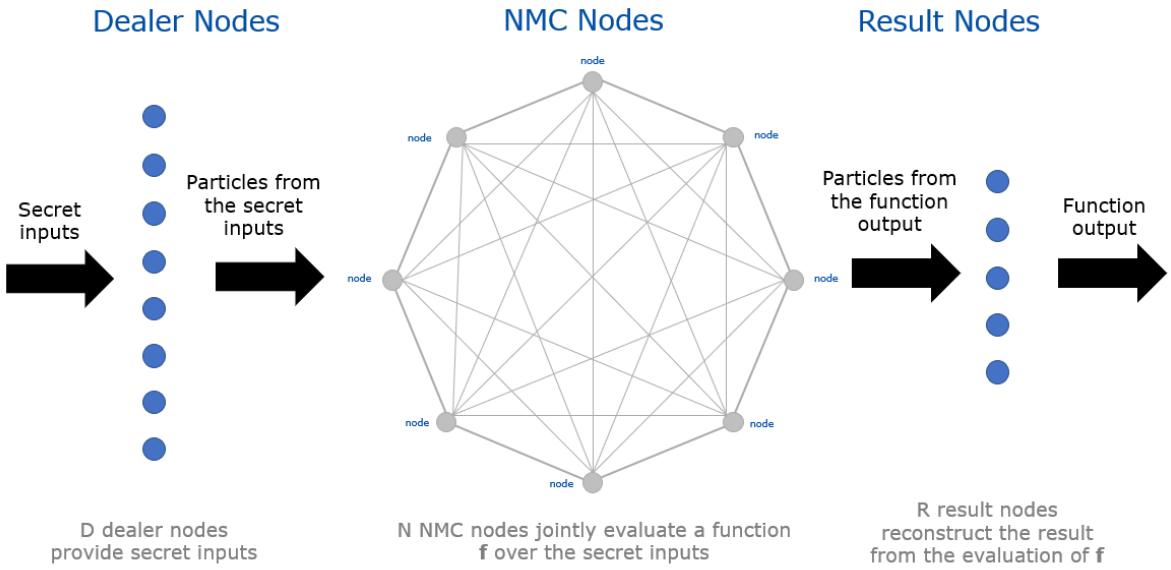

Figure 3: NMC computation flow.

the NMC protocol, the NMC Nodes are able to compute particles corresponding to the result obtained after evaluating the function with the secret inputs. More specifically, multiplications and additions of secret inputs translate into local multiplications and additions of particles, respectively, for which no inter-node communication is required. The NMC Nodes then send their result particles to one or several Result Nodes, which use NMC's Reconstruction Mechanism (described later in the paper) to reconstruct the actual result.

For the sake of simplicity and without loss of generality, we assume in this section that every node in the network stores particles and/or shares from a secret. In the actual Nillion implementation this may not be the case since only a subset of the total number of available nodes in the network (e.g. 100 nodes) will typically be used to store a particular secret.

NMC comprises four steps:

Step 0 – Pre-Processing: Similar to BGW with Beaver's triples [15] and SPDZ [16], NMC requires the NMC Nodes to run a pre-processing step with preliminary computations. These computations are not related to the private inputs of any particular NMC execution and can therefore be pre-computed hours, days, or even months before they are actually needed.

The purpose of the pre-processing step is for the NMC Nodes to generate shares of the blinding factors used by the OTM cryptographic primitive. In some instances (e.g. when all the secret inputs come from the same source), the Dealer Node(s) can generate the blinding factors, compute their shares, and distribute them across the NMC Nodes. In a more general setup, the pre-processing step is carried out by the NMC Nodes, which

generate individual/local shares of the blinding factors without having access to the full, reconstructed value of the blinding factors. We now explain the more general setup in more detail.

Using Shamir's Secret Sharing, each blinding factor is defined by a random degree-T polynomial. The N NMC Nodes are divided into two groups, with Group 1 containing $T+1$ nodes and Group 2 containing $N-(T+1)$ nodes. N shares from a blinding factor polynomial are generated as follows. First, the $T+1$ nodes in Group 1 locally generate a random share, which together define the degree-T polynomial. They treat their locally-generated random share as a secret and make use of an LSS scheme to provide the rest of the nodes in the two groups with a share from this secret. Second, using the linear property from the LSS, each node locally computes a share from each one of the remaining $N-(T+1)$ evaluations from the polynomial defining the blinding factor. Third, each node in Group 2 receives from every other node the computed share from its polynomial evaluation and reconstructs it using the Berlekamp-Welch [17] decoder algorithm for Reed-Solomon codes. For $N=3T+1$, this algorithm can reconstruct the polynomial in the presence of up to T wrong or missing shares. At this point, every node in Groups 1 and 2 has a share from a degree-T polynomial that hides a random blinding factor, but no node knows its value.

Notice that the pre-processing step does require inter-node communication. In order to improve efficiency, the pre-processing step is performed in parallel for many blinding factors, and it can also make use of batch secret sharing [18].

Step 1 – Particle Distribution: In this step, each Dealer Node applies an ITS OTM to hide each private input using different blinding factors. We call the resulting masked private input a particle. The blinding factors used to mask the private input are generated by the NMC Nodes during the pre-processing step and reconstructed by the Dealer Nodes from the LSS shares they receive. Notice that since Step 0 makes use of an LSS with a polynomial of degree $T+1$, no individual NMC Node or group of T or less colluding NMC Nodes can reconstruct the *blinding factors* from the shares, which are also ITS. At the end of Step 1, the Dealer Nodes broadcast their particles to the network. More specifically, every Dealer Node sends a copy of the particle obtained from each one of its private inputs to every selected node in the NMC network. The NMC Nodes can store particles either privately or publicly, since they are ITS and hence do not leak any information.

Step 2 – Computation: After Step 1, each NMC Node contains one particle from each private input to a computation. The computation consists of evaluating the output of a function over the private inputs. To achieve this, the NMC Nodes perform local multiplication and addition operations on their particles as dictated by the arithmetic function that needs to be evaluated. That is, every plaintext multiplication or addition is matched with a local multiplication or addition of particles. Thus, this step requires no inter-node communication. This is the key property and innovation that enables NMC to operate at plaintext, centralized server speed, since the bulk of the computations are performed in parallel by the nodes' CPUs operating at giga floating point operations per

second (GFLOPS).[27]

Step 3 – Result Reconstruction: After Step 2, each NMC Node will have obtained a particle from the result of the computation (i.e. the function to be evaluated). Each node sends its particle and the blinding factor shares to one or several Result Nodes (depending on where the result is required and/or how it is to be used, as determined by the end user). Then, each Result Node runs NMC's Reconstruction Mechanism to obtain the output of the function that was jointly evaluated. The Reconstruction Mechanism makes use of a Verifiable Secret Sharing mechanism to reconstruct the blinding factors in the presence of active adversaries, and then unmasks the result of the computation, inverting the OTM operation. The result of the computation is thus revealed but not the computational inputs.

3.6 2-NMC and D-NMC

Nillion uses two different flavors of the NMC protocol, both with the key property of allowing for fast computations without inter-node communication:

- 2-NMC allows a network of NMC Nodes to evaluate **any** function with private inputs coming from **1** or **2 Dealer Nodes**. This function can be expressed as an inner product of two A-dimensional vectors, where A is the number of additions in the computation.
- D-NMC allows a network of NMC Nodes to evaluate **any** function with private inputs coming from **any** number D of Dealer Nodes. This function can be expressed as a sum of the products of confidential input values.

Both NMC flavors allow the NMC Nodes to perform a computation without exchanging any messages. However, 2-NMC is more efficient than D-NMC in terms of bandwidth. For N NMC Nodes:

- Each secret in D-NMC is turned into a particle that needs to be broadcast to the network. Therefore, with D-NMC, there is a 1-to-N ratio between a secret and the number of messages that its dealer needs to send to the network.
- N secrets in 2-NMC with passive adversaries are turned into N particles, each one to be broadcasted to a different node in the network. Therefore, in this case there is a 1-to-1 ratio between a secret and the number of messages that its dealer needs to send to the network.

2-NMC is the perfect protocol for scenarios with a single dealer (e.g. biometric authentication, such as facial recognition or fingerprint matching, or signing transactions or

[27] Indiana University, 'Understand measures of supercomputer performance and storage system capacity' <https://kb.iu.edu/d/apeq> accessed 1 February 2022

documents with a private key) or only two dealers (e.g. searching against another dealer's database or finding the intersection between two datasets). This whitepaper, particularly Section 3.4, focuses on D-NMC, although many of the results presented also apply to 2-NMC. An exception to this is the process through which data is transformed into particles. In D-NMC, a data item is transformed into a particle and then it is broadcasted so that every NMC Node receives an identical copy of it. However, in 2-NMC, all the input data items from a dealer are shredded/"horcruxed," producing several non-identical particles, which are then distributed (not broadcasted) across the network. The details of 2-NMC will be explained in depth in future publications in updates to this whitepaper.

3.7 A Performance Breakthrough

NMC allows a decentralized network of nodes to store and process confidential information under ITS, with the processing taking place at essentially centralized server speed operating on plaintext data. This unique feature represents a significant performance breakthrough compared to SMPC and originates from the fact that NMC requires no inter-node communication. We will explore this result in more detail in the present section.

Figure 4 (below) provides a comparison between centralized architectures, best-in-class SMPC, and NMC in terms of the number of messages sent during each one of the steps. Since the pre-processing steps in SMPC and NMC can be efficiently performed in batch and do not depend on the actual inputs to the computations, they are not included in Figure 4. To facilitate the comparison, we assume that: (1) we have a network with N nodes, (2) the number D of dealers equals N, (3) each dealer only contributes one confidential input to the computation, and (4) the computation involves the evaluation of M multiplications in series. The results in each step in Figure 4 are explained below:

- **Step 1**. In all cases, either 1 or N messages are sent. Since N messages can be sent in parallel, the required time taken to send multiple parallel messages is almost the same as for a single message. For large values of N, message aggregation can be used to reduce the total number of messages sent from the dealers.

- **Step 2**. NMC and the centralized architecture require no message exchange, whereas BGW does. This step is where most of the delays occur in SMPC protocols such as BGW. Specifically, the term $O(MN^2)$ prevents BGW from scaling to large networks running computations involving many multiplications. For example, in BGW, with each multiplication $2N$ broadcast messages are required to use Beaver's triples, which results in $2N^2$ point-to-point messages. For M multiplications, $2MN^2$ messages need to be sent. This is clearly prohibitive and is one of the primary issues that NMC solves.

- **Step 3**. NMC and BGW require N messages to be sent, whereas the centralized architecture requires only 1. However, these messages can be sent in parallel, thus making the user experience essentially the same.

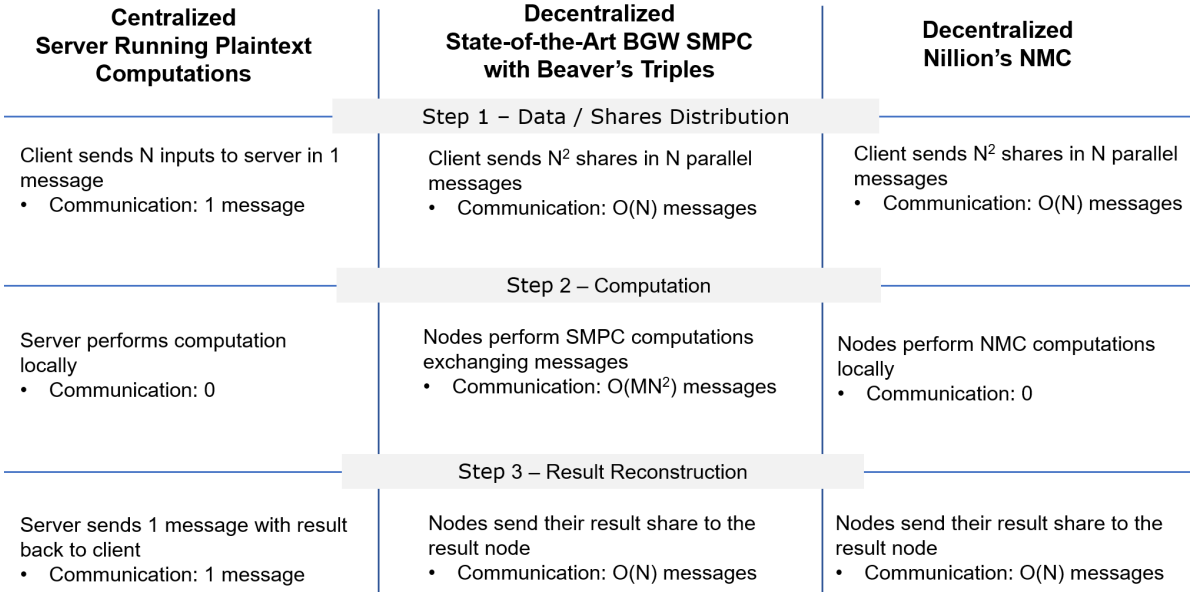

Figure 4: Performance comparison in terms of communication complexity between a centralized computation in plaintext, a state-of-the-art SMPC protocol, and Nillion's NMC (D-NMC). For comparison purposes, in this setup, the number D of dealers equals the number N of nodes in the network, and the computation involves the evaluation of M multiplications in series.

This means that use cases can run in NMC at essentially the same speed and for the same amount of time as they do in current centralized architectures because: (1) network nodes can perform the computations locally and in parallel without sending or receiving information, and (2) the nodes in Steps 1 and 3 can receive and send the N messages in parallel, respectively.

Bandwidth Consumption

During the computation step of NMC, zero, or "nil" messages are exchanged between the network nodes. This contrasts with the high number of messages that state-of-the-art SMPC protocols, such as BGW with Beaver's triples, need to exchange. Figure 5 shows the number of messages exchanged, as well as the amount of network bandwidth required by a best-in-class SMPC as a function of the number N of nodes in the network and the number M of products in series to be executed. As can be appreciated, standard SMPC cannot scale to the size of even small blockchains with hundreds of nodes, let alone to the size of large ones, such as Ethereum (which has over 10,000 nodes). As can be seen below, even for several hundred nodes, billions of messages need to be exchanged, thus flooding the network with terabytes of data. In contrast, since NMC requires no message exchange, Nillion can theoretically scale up to any number of nodes.

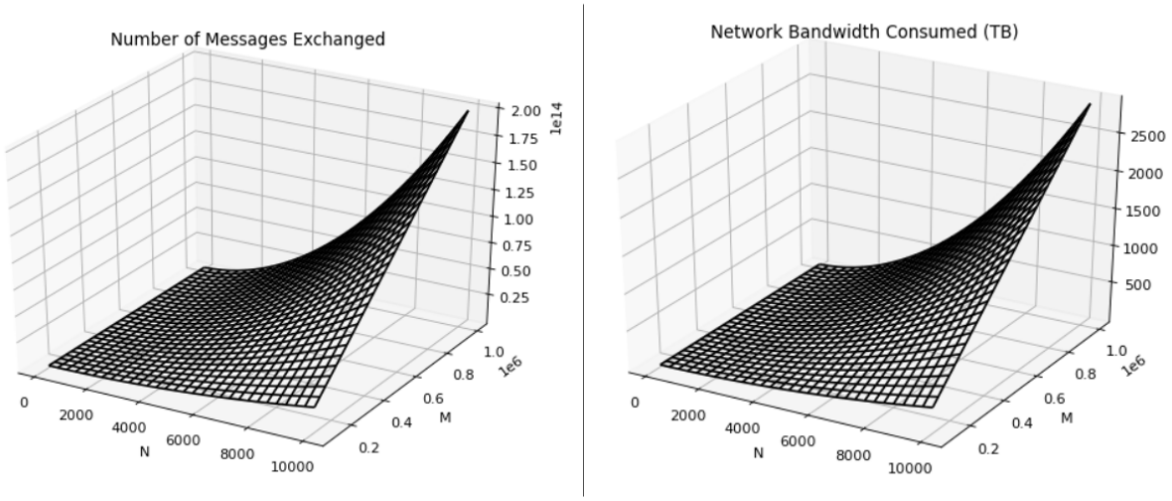

Figure 5: (left) Number of messages exchanged and (right) terabytes of bandwidth consumed by the nodes in state-of-the-art SMPC BGW with Beaver's triples during the computation step as a function of the number N of nodes in the network and the number M of multiplications in series to be processed. A 128-bit security level for the LSS is assumed. With NMC, no messages are exchanged and no bandwidth is consumed during the computation step.

Execution Time

In state-of-the-art SMPC, every multiplication requires the exchange of $2N^2$ messages in a network with N nodes. In NMC, it involves a local multiplication between particles that requires no message exchange. In fact, computations in NMC run at plaintext CPU speed, whereas in SMPC, they are impacted by network propagation and transmission times as well as the number of nodes in the network. CPU computations run at tens of billions of floating-point operations per second (FLOPS), whereas network message propagation takes hundreds of milliseconds. NMC is thus theoretically billions of times faster than standard SMPC.

To illustrate this, let us consider an idealized situation with the following two assumptions:

- All $2N^2$ messages can be sent in parallel by the network nodes in only 100ms.
- Message transmission times are zero.

It is clear that both assumptions are highly optimistic. In a large network with hundreds or thousands of nodes, $2N^2$ messages cannot be sent in 100ms because nodes do not operate in perfect synchronization, and each node cannot send $2(N-1)$ messages while receiving $2(N-1)$ messages as required by Beaver's triples, all in perfect parallelism. Additionally, for a large number of messages (see Section Bandwidth Consumption), the time it takes a node to transmit large amounts of data will be significant.

A simple example can be used to illustrate that even when comparing SMPC and NMC under such ideal conditions, and assuming low network propagation times of 100ms, the fact that messages need to be exchanged in SMPC will lead to very long execution times relative to Nillion's NMC. In these circumstances the execution time in standard SMPC primarily depends on the number M of multiplications in series that need to be processed by the nodes. Figure 6 illustrates the difference in execution times between a state-of-the-art SMPC protocol and Nillion's NMC running 1 million multiplications in series. As it can be seen, most of the time in an SMPC transaction is spent during the computation step (Step 2 of SMPC). In this step, every node needs to send 2 messages to each other node and receive two messages from each other node in the network for every multiplication. This step consumes 99.998% of the total execution time and requires almost 28 hours, compared to the 0.1ms it takes to compute 1 million multiplications of particles in an NMC Node supporting 10 GFLOPS.

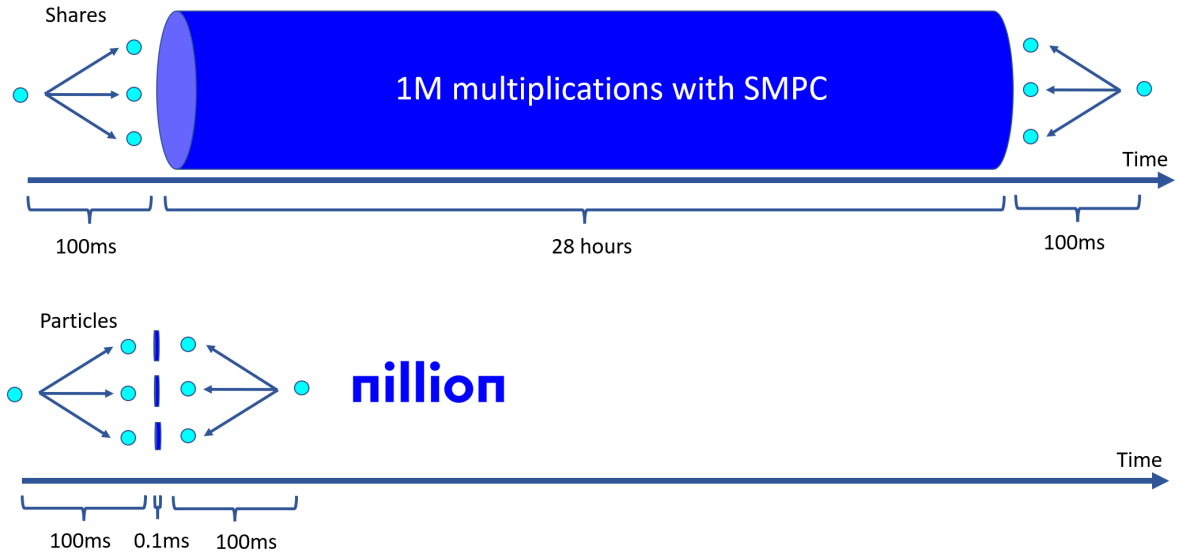

Figure 6: Performance comparison for the execution of 1 million multiplications in series in state-of-the-art SMPC BGW with Beaver's triples and Nillion's NMC. Inter-node propagation times are assumed to be 100ms and the network is assumed to be able to send any number of messages in parallel in 100ms (this assumption primarily benefits SMPC). The nodes are assumed to support 10 GFLOPS, and multiplications are assumed to be the CPU bottleneck.

3.8 A New Approach to Decentralization

Distributed Ledger Technologies (DLTs), such as blockchains, decentralize the transmission, storage and processing of information across a network of nodes. A common approach is for each node to store its own version of the ledger, represented as a chain of blocks (e.g. Bitcoin, Ethereum), or to use a more complex non-blockchain structure (e.g. IOTA, Hedera Hashgraph) to keep and maintain state. Not every node is assumed to be

a good actor or to properly function at all times. Thus, the key element to DLT decentralization is a consensus mechanism that allows the nodes to reach an agreement on the correct *value* and *order* of the transactions on the ledger, even in the presence of faults introduced by dishonest nodes.

While NMC supports ITS, thus bringing best-in-class confidentiality while achieving the speed of plaintext centralized computing, the unique properties of the technology allow it to function as much more. As a technology, NMC was also designed to operate in the presence of bad actors. By focusing on the *value* of data and computations but not on their *order*, NMC Nodes are able to deliver consensus to the Result Nodes. However, unlike DLTs, they do not require the exchange of messages, nor do they need to perform any computations to reach consensus, thereby achieving what we refer hereinafter to as "instant consensus." Instead, the Result Nodes are able to reconstruct the correct value of data by running a local computation from the data received by the NMC Nodes. Consensus on the *value* of data and computations is sufficient to support the vast majority of Nillion use cases. In the instances in which consensus on both the *value* and the *order* of data is required, the *order* information (e.g. timestamps) can be carried on the back of messages the NMC Nodes send to each other to broadcast the particles received from the dealers. However, while consensus still does not require the exchange of extra messages by the nodes in such cases, it will no longer be instant.

The security models of NMC and DLT are very similar. Bad actors can send wrong messages, not send messages at all, or delay their transmission with unbounded message delays. However, NMC enables a method of reaching agreement that is not based on the exchange of messages between the network nodes, representing an alternative approach to node decentralization. This approach is based on NMC's Reconstruction Mechanism, where the OTM and a Verifiable Secret Sharing version of LSS are used in order to reconstruct the input blinding factors from which the results of a computation can be unmasked (see Section 3.4). Using this approach, NMC agreement presents similar results to those obtained in the DLT and blockchain space [6]:

- Byzantine agreement[28] is guaranteed in a network with less than ⅓ of bad actors that provide secure communication links between nodes [19].
- Agreement is guaranteed in a network with less than ½ of bad actors that provide secure communication links between nodes and a broadcast channel.

However, unlike in a DLT, there are two key innovations presented in NMC:

- The secrecy achieved in both cases is ITS.
- The agreement does not require the exchange of messages by the network nodes.

[28] An agreement in the presence of Byzantine faults. In a Byzantine fault, a node can inconsistently appear as both failed and functioning to failure-detection systems. This could be caused by a malicious node that sometimes behaves as expected and sometimes does not.

Thus, by focusing on the *value* but not on the *order* of data, not only is NMC agreement ITS, but it is also extremely fast. More specifically:

- Agreement in NMC takes place at the point at which information is reconstructed. It is obtained on-demand and only pertains to the piece of information being reconstructed. Shares and particles from bad actors are automatically discarded during reconstruction and the correct result is read from the NMC network. Since information is not organized in distributed blocks that need to wait for consensus, NMC agreement does not require any message exchange between the NMC Nodes. Instead, it is guaranteed by the result reconstruction; a local computation that runs at CPU speed.

- Agreement in a DLT is something that happens globally and regularly in the network during consensus. It requires information to be exchanged between the nodes and is independent of whether DLT data is required for consumption or not. Since information is organized into blocks, even in blockchains with fast finality, a node is required to wait for its data to be included in a block before it is accepted into the blockchain.

Consequently, while building an NMC-powered decentralized network or Layer 1 solution is not the present focus or goal, the technology could nonetheless power a new generation of decentralized networks in the future, with thousands of nodes running arbitrarily complex algorithms under ITS and Byzantine fault tolerance in real-time.

3.9 Security

Nillion's security model is based on the globally recognized three elements of security:[29]

- *Confidentiality*: Data and computations are confidential.

- *Integrity*: Data is protected from unauthorized changes. Data and computations are reliable and correct.

- *Availability*: Resources are always accessible to authorized users even in the event of node failures

In NMC, availability is a tunable security parameter that does not come at the expense of confidentiality or integrity. Confidentiality is achieved under ITS, and unlike blockchain consensus mechanisms, integrity is obtained upon completion of the computation without any message exchange between the network nodes, providing "instant finality."

Let N be the number of nodes in the network, and $T + 1$ be the minimum number of nodes (shares) required to reconstruct the blinding factors in the OTM cryptographic

[29]Center for Internet Security, 'Election Security Spotlight – CIA Triad' (Center for Internet Security, 15 June 2021) <https://www.cisecurity.org/spotlight/ei-isac-cybersecurity-spotlight-cia-triad/> accessed 1 February 2022

primitive. We also refer to $T+1$ as a *qualified majority* of nodes. We distinguish between *front door* attacks, which require user interaction (e.g. user impersonation) and *backdoor* attacks, which exploit a network or software vulnerability (e.g. a Nillion node becoming a bad actor).

Confidentiality

Confidentiality in Nillion is guaranteed under ITS as long as fewer than $T+1$ nodes are bad actors. Nillion supports two types of modes:

- *Compute and reconstruct*: The particles are processed in such a way that the secret inputs to a computation can be reconstructed by their corresponding dealer(s).
- *Compute only*: The particles are processed in such a way that the secret inputs to a computation can never be reconstructed. This includes both their dealer(s) and any attacker. However, the particles can still be used to perform a predefined computation.

For example, a dealer can decide to store their fingerprint template in compute only mode. This would mean that the dealer can use Nillion to match their fingerprint against this template without it being possible for anyone to reconstruct their biometric information.

To prevent front door attacks on the Dealer or Result Nodes (e.g. impersonation attacks), Nillion has an advanced, user-defined, Multi-Factor Authentication system (see Section The Nil Service Layer). MFA is an NMC computation that a Dealer or Result Node performs against the NMC network. The NMC Nodes themselves receive the result from this computation and can individually determine if a Dealer or Result Node is successfully authenticated. When an authenticated Dealer Node instructs the NMC network to perform a computation, they also specify the list of Nillion identifiers (i.e. addresses) that correspond to the Result Nodes that are authorized to reconstruct the result. Authenticated Result Nodes with Nillion identifiers included on the Dealer Node's list then receive the particles and shares required for the result reconstruction. Since NMC consensus does not require inter-node messaging, MFA authentication will be processed at CPU speed in spite of running on a decentralized network of nodes.

Regarding backdoor attacks, it should be noted that with traditional encryption, the confidential information is *one hack away*. In other words, if the security of one system (node) holding encrypted data is compromised, the data will be revealed. In contrast, in NMC, the confidential information is $T+1$ *hacks away*. Accordingly, a hacker must breach $T+1$ different systems (nodes) in order to reconstruct a secret. With large T values, this becomes virtually impossible.

Nillion's ITS security model prevents the NMC Nodes from learning anything about the inputs to a computation, but the outputs revealed to the Result Nodes could potentially contain information about the secret inputs to a computation. For example, one could devise a malicious computation that directly outputs the values of some private

inputs. However, the user controls: (1) what algorithms (computations) run on their secret inputs, which are publicly audited and vetted, and (2) which node(s) learn about the result from the computations (e.g. themselves or a bank), which will depend on the use case. In other words, they control who learns what and when.

Integrity

Integrity means that data is protected from unauthorized alteration, which provides assurance regarding the accuracy and completeness of the data. Data integrity is preserved against: (1) front door attacks using an advanced MFA system, and (2) backdoor attacks using NMC's Reconstruction Mechanism (see Section 3.8 A New Approach to Decentralization). That is, wrong particles or shares sent to Result Nodes by NMC Nodes in a backdoor attack are discarded, and missing particles or shares are ignored during the result reconstruction as long as there are at least $T+1$ correct particles and shares. Shares and particles are discarded as follows:

- *Shares selection*: Wrong shares are detected since all shares are authenticated with an ITS Message Authentication Code (see Nillion's Verifiable Secret Sharing scheme presented in Section 3.5). The use of error correction codes is also possible [17].
- *Particle selection*: If all nodes were good actors, all particles would have the same value. Therefore, the particle selected for reconstruction is the most prevalent amongst the output particles received by the nodes.[30]

We now describe two relevant setups for Nillion that guarantee Integrity under different operating conditions:

- *Byzantine setup*: Under the worst operating conditions (Byzantine faults), we set $N = 3T + 1$, in which case NMC achieves Byzantine agreement tolerating up to T bad actors (less than ⅓) who might try to reconstruct a secret or prevent a secret from being reconstructed. This includes the unlikely case of a firewall attack (this type of attack has never been successfully executed on Bitcoin, or any other well-known cryptocurrency, so it is largely academic at this point, though still important to consider given what is at stake), in which a bad actor can control T nodes and prevent the particles and shares from other T nodes from arriving at the Result Node(s) by using a firewall.
- *Broadcast Channel setup*: A firewall attack is somewhat improbable and would likely come from a government. Despite this, it could be further mitigated by using a system that chooses node IPs for secrets that (a) are a well-balanced mix of multiple jurisdictions or that (b) match the jurisdiction from which the secret originates. In both cases, as a firewall attack is extremely unlikely, what we are

[30] We are assuming here that the output of the computation only comprises one number and that there is consequently only one output particle. The extension to multiple outputs is trivial.

assuming is essentially the existence of a functional broadcast channel. In this case, by setting $N = 2T + 1$, NMC achieves agreement tolerating up to T bad actors (less than ½) who might try to reconstruct a secret or prevent one from being reconstructed.

Availability

One important aspect of availability is guaranteeing timely and uninterrupted access to Nillion. Some threats to availability are malicious, such as various forms of sabotage and denial-of-service attacks. Others are non-malicious, such as unscheduled software down-time, network bandwidth issues, and hardware failures. Another aspect is guaranteeing that information is only accessible to the right end user. In this section we focus on the former, while the latter is addressed in Section 4.1 (see *authentication* service).

A confidential data asset is available in Nillion if and only if the following conditions are met:

- (1) There is at least 1 node available holding a particle from that asset.
- (2) There are at least $T+1$ nodes available to reconstruct the blinding factors using LSS in order to extract the data from the particle.

Particles are ITS, meaning the particle from a data asset can be sent to as many nodes as needed. Therefore, Nillion's availability is impacted by point (2) above.

The integrity mechanisms described above provide a certain degree of availability, since one of the possible node faults considered is a node becoming unavailable. In the Byzantine case and Broadcast Channel case, at least ⅓ and ½, respectively, of the nodes must be down for Nillion's availability to be affected. In this section, we discuss an additional mechanism that improves Nillion's availability even further.

One overly simplistic approach would be to send more than one copy of a share to different nodes in the network. In this case, if a node holding the share went down, there would be other nodes holding the same share. However, this approach is not viable in the presence of a proportion of malicious nodes. By sending the same share to many nodes, the probability that a set of colluding malicious nodes receives the $T+1$ shares necessary to reconstruct the blinding factors would increase, threatening confidentiality.

Nillion is able to fine-tune availability without impacting confidentiality by sending multiple *shuffles* of shares from the same secret to different nodes in the network. Each shuffle corresponds to a different polynomial that hides the same blinding factor. If less than $T + 1$ nodes are available in a shuffle, they are unable to reconstruct a blinding factor. In this case, Nillion operates with sets of nodes corresponding to other shuffles until a qualified majority of available nodes is obtained.

The number of shuffles is called the *redundancy factor*. In order to quantify the effect of the redundancy factor, we look at the *reliability* R, defined as the probability that the system functions as intended.

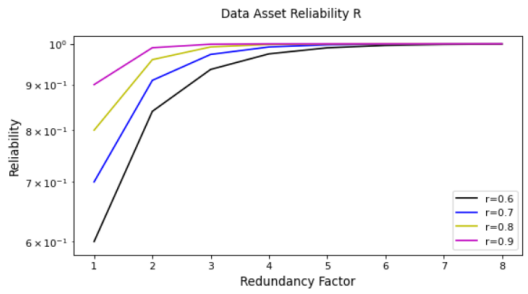

Figure 7: Nillion's reliability as a function of the redundancy factor.

Assuming that the reliability of the nodes holding a shuffle of shares is r, the reliability of a group of G shuffles for a given particle corresponds to that from a parallel system: $R = 1 - (1 - r)^G$, where G is the redundancy factor. Figure 7 illustrates how the reliability of the system quickly approaches 1 as the redundancy factor increases. In this way, Nillion can achieve high availability through redundancy.

3.10 Processing

Thus far, this paper has covered how NMC works and has explored its performance, decentralization, and security characteristics. The present section will explore NMC's core capability: processing.

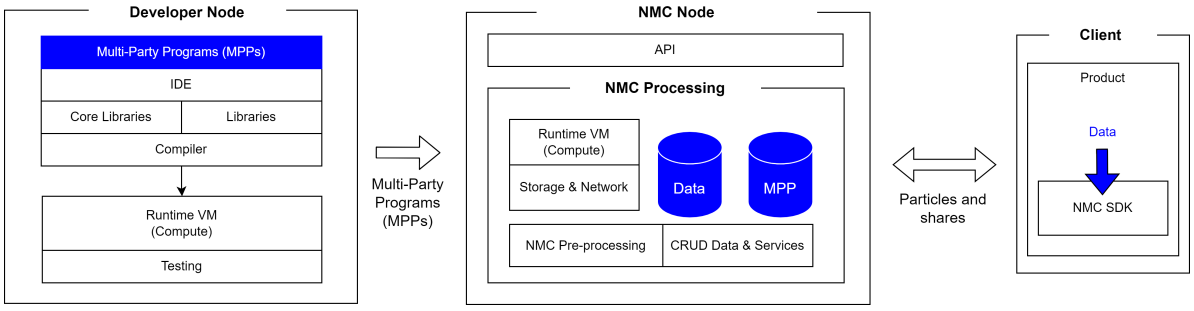

Figure 8: Main components involved in processing.

Figure 8 highlights the main components involved in processing that Nillion plans to build, including the flow of data (particles and shares) and code (services) between them. Focusing on the use of NMC's processing capability, we distinguish between *compile time* and *runtime* activities.

Compile Time

At compile time, Nillion and the community will develop services that process data on the network using a Developer Node (see Figure 8). NMC will support a high-level (potentially several) programming language(s) that will be compiled into an arithmetic circuit representation of the computation in bytecode, which we call a *Multi-Party Program*. An Integrated Development Environment (IDE) will assist the developer in the creation of services, for example, by providing syntax checks and autocompletion. Developers will also be able to:

- Make use of standard libraries or create their own to abstract and reuse code.

- Test a service in a runtime virtual machine (VM) that runs on a local virtual test network.
- Send MPPs to the network of NMC Nodes.

Runtime

At runtime, the network of NMC Nodes will run services on particles created from real data. An NMC Node comprises an API to receive MPPs, particles, and shares. At the core of an NMC Node are its main processing components:

- A data storage module for particles and shares.
- A service storage module for MPPs.
- A runtime VM that runs on the storage and network layers (see Section 4.1).

Particles and shares have their own Create, Read, Update, and Delete (CRUD) module that is used when the NMC Node acts as a Dealer Node or as a Result Node (see Section 3.5), enabling it to create input particles and to reconstruct a result from a set of output particles. The NMC Node also has a pre-processing module that runs in batch the Pre-Processing Step (Step 0) outlined in Section 3.5. This module represents one instance where an NMC Node has to act as a Dealer and Result Node for shares and thus makes use of the aforementioned CRUD module.

Client applications interact with the NMC Nodes by using an *NMC SDK*[31] (see Figure 8), which computes input particles from sensitive data and reconstructs processing results from output particles. The use of an SDK to intermediate end-user interactions with the NMC network as opposed to an API is central to Nillion's security model introduced in Section 3.9:

- *Availability*: The SDK enforces access control through Multi-Factor Authentication to prevent attacks, such as user impersonation and account theft (see *authentication* service in Section 4.1).
- *Confidentiality*: The SDK ensures that sensitive information never leaves the client application (only ITS particles), and processing results from user-related computations are sent back to the SDK in ITS form and are only reconstructed locally inside of the client application.
- *Integrity*: The SDK runs the reconstruction algorithm and makes use of a Verifiable Secret Sharing mechanism to ensure that the right processing result is reconstructed, despite the fact that a certain percentage of NMC Nodes are bad actors (e.g. they send wrong messages, do not send any message, or delay the transmission of their messages). That is, the SDK delivers consensus directly to the end user, who is

[31] By SDK we mean a library or module that runs in the client application.

consuming information by running an algorithm locally in their system that does not require the NMC Nodes to exchange any messages (i.e. instant consensus).

The client that runs the NMC SDK can be a Dealer Node or a Result Node, respectively, depending on whether it injects or extracts data into or from a computation. Nillion envisages different types of clients, including mobile applications, web applications, browser extensions, backend servers, and smart contracts, and will provide a wrapper of the NMC SDK in the appropriate programming language(s). The NMC SDK will connect to the network of NMC Nodes through their API, offering three core capabilities:

- *Transfer*: Allows the transfer of data under ITS. For example, this capability can be used to transfer the private key that controls a crypto wallet if certain conditions are met, such as the death of the owner of the key.
- *Store*: Allows the storing and retrieving of data under ITS. For example, this capability can be used to securely store the private key.
- *Compute*: Allows the processing of data utilizing particles under ITS. For example, the owner of the private key can now send particles from a transaction to Nillion and have the nodes sign the transaction on their behalf without ever having to reconstruct the private key. Thus, the owner is able to reconstruct the signed transaction and send it to a blockchain (e.g. Ethereum) at roughly the same speed as a centralized client-server transaction.

Therefore, developing on Nillion is as simple as:

- Writing code in a high-level programming language and sending it over to the NMC network.
- Developing an application that makes use of the NMC SDK to facilitate the processing of the end user's sensitive information, including its transfer to and from the network, its storage, and the computations that run on it. The application can run on multiple platforms, including smart contracts, mobile devices, web browsers, servers, and desktops.

4 The Nillion Network

In Section 3, we introduced the NMC protocol and analyzed its performance, decentralized nature, and security features. We also outlined how developers will experience and consume its core processing capability involving the transfer and storage of sensitive data, as well as how they will design and run secure decentralized computations. In this section, we take an engineering and operational perspective to describe a number of key aspects of the decentralized network surrounding the NMC protocol. In Section 4.1, we

present the three main layers comprising the network's foundational architecture. In Section 4.2, we discuss the main attack vectors on NMC, such as Sybil attacks, bribery, and collusion between nodes, and we describe how node composition prevents these attacks while remaining permissionless. In Section 4.3, we discuss the important role of the NIL token in the Nillion Network, including node incentives, staking, network security, and decentralized governance.

4.1 Architecture

Nillion's network architecture comprises three layers: the Infrastructure Layer, NMC Layer, and Service layer. The Infrastructure Layer provides basic decentralized network and storage services. The NMC Layer is where Nillion NMC's protocol is implemented with all its various elements and features. The Service Layer builds on NMC to provide several core services to the developer community. This layer can also be extended by the community, who can build products and applications on top of it.

The Infrastructure Layer

The network and storage infrastructure in the Nillion Network builds on a Distributed Hash Table (DHT) and underlying peer-to-peer network with the following properties:

- **Decentralized**. Unlike blockchain storage, the Nillion storage layer can distribute data among the nodes so that no node has a copy of all the data. This layer is decentralized because the nodes operate without any centralized authority.
- **Resilient**. The storage layer is both fault tolerant and resilient. The nodes coordinate to balance and store data in the network without the need for a central coordinating authority.
- **Supports LSS**. We will adapt the DHT storage mechanism to distribute the storage of the *shares* of a data item (i.e. blinding factors) across several nodes.
- **Supports other storage forms**. Our storage layer can also store particles and metadata.

The NMC Layer

The NMC Layer implements our two proprietary protocols: 2-NMC and D-NMC. These protocols are able to run Multi-Party Programs without any message exchange during the computation step with inputs from 2 and D dealers, respectively. The NMC Layer implementation is secure, exhibiting high availability, confidentiality, and integrity. The data required for the computation is provided by the Infrastructure Layer. NMC will offer three core capabilities to developers through an SDK: transfer, store, and compute (see Section 3.10).

The Nil Service Layer

While the transfer, storage, and compute core capabilities are covered by the NMC Layer above, the Nil Service Layer builds on top of this via Nillion's MPPs to provide additional services. These services will be used by Nillion to implement the initial use cases showcasing the network's capabilities (see Section 5), and can be further extended by the developer community. To do so, Nillion will offer its own high-level programming language called *Nada*, and may support additional existing languages, such as Solidity. The list of initial services follows:

- *Authentication*: Offers various out-of-the-box methods for authenticating and authorizing user operations using MFA. All authentication factors are transformed into ITS particles and distributed throughout the network with all authentication processes being executed directly in Nillion. Consequently, authentication factors are never stored in a centralized server or on a user's device. Nillion's MFA will support the following types of factors:
  - *Something you are*: Biometric information such as face and fingerprint data. Clients will use their mobile or tablet device to capture this biometric information. The information will be stored in ITS particles in the network and matching will take place without ever having to reconstruct any of the factors.
  - *Something you have*: Information that uniquely identifies the client's device, email address, and phone number.
  - *Something you know*: Passwords and pin codes.
  - *Something you do:* Behavior in response to a challenge.

  Additional factors may include geolocation information and time. If the client is unable to provide all the factors (e.g. they forget their password or their device is stolen or lost), they can recover the control over their personal data through the use of other factors. The number of factors and the minimum number of factors required to associate a new factor, or to change an existing factor in the account, will depend on its security level or its level of assurance.

- *Key management*: Offers different functionalities around the storage, transmission, and processing of digital certificates and keys.

- *Enrollment*: Offers decentralized Know-Your-Customer (KYC) and Know-Your-Business (KYB). It allows the onboarding of an individual or organization to a desired level of assurance by capturing and running identity and suitability checks.

- *Voting:* Offers decentralized anonymous voting with configurable quorum and majority definitions, as well as definable rules that can be used, for instance, to automate governance functions in DAOs.

Future services that may follow include machine learning, data analytics, identity management, and more. Also refer to Section 5 for an expanded description of more advanced use cases.

4.2 Permissionless Nodes

Nillion is a permissionless network supporting two types of NMC Nodes:

- *Light Nodes*: These transfer, store, and process particles from the data. Particles are public.
- *Full Nodes*: These transfer and store shares from the blinding factors, and also transfer, store, and process particles from data. Shares are private.

Light Nodes participate in Nillion's core services. In particular, they store confidential data and carry out computations on the particalized data they hold. Shares are computed from the blinding factors and are completely independent of confidential data. These shares are stored in Full Nodes, as shares are key to maintaining the confidentiality of the network. Full Nodes are also able to participate in the same core services as Light Nodes, such as storing particles and carrying out computations.

In a Sybil attack, a single entity pretends to be multiple independent nodes, hoping to be assigned all the shares for a data asset so as to be able to reconstruct it. To prevent Sybil attacks against Full Nodes, node operators will be required to stake a certain number of NIL tokens in order to participate in the storage of shares and calculations. This will make it economically prohibitive for an attacker to launch a large number of nodes. Further, either to bootstrap the network or at network inception, some of the initial Full Nodes may be required to be enrolled so that the same individual or organization cannot run an overwhelming number of nodes while the network is in its infancy (see Section Node Self-Governance System). As the network (and thus the number of staking nodes) grows, the chance that a group of malicious nodes will receive a sufficient number of shares to reconstruct blinding factors will approach zero. For example, in a network with 100 Full Nodes, a bad actor who controls 51 Full Nodes poses a threat. However, the same bad actor is impotent in a network of 10,000 Full Nodes. Further, as both the number of Full Nodes and the value of NIL tokens increases, nodes will be required to stake an increasingly large dollar value of NIL tokens to qualify as Full Nodes, thus making it economically prohibitive for any single entity to control a sufficient number of nodes to reconstruct a secret. Consequently, Sybil attacks will become exponentially more difficult to execute over time, and enrollment for Full Nodes will eventually cease to be important. Finally, note that since the particles from a secret are all identical, Sybil attacks do not apply to Light Nodes. Due to the fact that the particles held by Light Nodes are ITS, there is no amount of computational power a node can apply to obtain information from them. This is why particles can be made public.

4.3 The Nillion Token

The NIL token will be Nillion's native utility token. NIL will initially be implemented using Ethereum's ERC-20 standard.[32] The full details of the settlement mechanics between the network, users, and node providers will be released alongside initial node testnet deployments. NIL tokens will be used to:

- Access the network and use various Nillion services.
- Align economic incentives and node cooperation through staking.
- Incentivize nodes to maintain high performance and uptime.

NIL tokens will also play a key role in the governance of the network through the Nillion Decentralized Autonomous Organization (nilDAO). nilDAO will be formed upon the public launch of the NIL token and be delegated authority over certain aspects of network governance. As Nillion becomes more decentralized and hardened, nilDAO will play an increasing role in network governance until it becomes the only remaining governing body of the network.

Accessing the Network

NIL tokens will be required to pay network nodes to carry out core transfer, store, and compute functions. Users will need to provide the necessary payment when they submit a request to the Nillion Network. Upon completion of the processing, the network participants will be rewarded for their work. The exact cost for various Nillion services will depend on several factors, including the computational complexity, storage requirements, and number of nodes involved. Given the use of NMC technology, node processing will be lower cost relative to decentralized blockchain-based protocols.

Node Staking Mechanics

The decentralization of data across multiple nodes on the Nillion Network ensures data security even when malicious actors are present. However, it is equally important to ensure that nodes on the network have an incentive to maintain network excellence and disincentivize attacks by bad actors. This is achieved by incorporating a staking requirement for nodes on the network. Similar to Proof of Stake blockchains, nodes on the Nillion Network will be required to stake NIL tokens in order to participate in the network, store particles, and run calculations. Staking (i.e. requiring nodes to provide a deposit of NIL tokens) ensures that nodes have an explicit financial incentive to behave correctly, while concurrently increasing the cost of attack by malicious actors. nilDAO will determine the

[32] While a Turing-complete Nillion Network could also theoretically create its own native token standard, at the date of publication of this whitepaper, current scalable blockchain solutions purpose-built for tokenization (e.g. ETH, SOL, AVAX, HBAR) seem to adequately address market needs. Consequently, we have opted to initially utilize the Ethereum ERC-20 standard. Details of the mechanics of the payment system will be published in an upcoming report.

minimum staking required for a node to participate in network functions. Light Nodes will be subject to a lower minimum staking requirement to receive particles and participate in calculations, while Full Nodes will be subject to a higher minimum staking requirement, given their enhanced role in the network.

The operational performance of nodes on the network is imperative to overall user experience and functionality. As such, all nodes will be subject to slashing (i.e. the confiscation of staking deposits) in the case of misbehavior or failure to meet operational requirements. Slashing will be performed automatically by the network when a node fails to meet the performance criteria established by the nilDAO. Nodes that fail to meet the minimum staking threshold, due to slashing or withdrawal of the staking deposit, will not be able to participate in network functions or store network data until the minimum staking deposit is restored.

Node Self-Governance System

The staking mechanics are an integral part of the larger Node Self-Governance System fortifying the network. This system will be monitored by nilDAO and incorporate both on-network and off-network information to create a holistic representation of each node's contribution to network integrity and performance. The Node Self-Governance System creates an important internal incentive mechanism for nodes and is a distinguishing feature of the Nillion ecosystem.

As part of the Node Self-Governance System, nodes will be assigned a Node Health Score (NHS) and a DAO Support Score (DSS), if applicable. All nodes will receive an NHS, while only Full Nodes will receive a DSS. The NHS and DSS of a node will be publicly available on the Node Leaderboard. Users will be given an optimized and user-friendly experience by using the nodes that win selection via NHS criteria for the network. However, if users want a more customized experience, they can also choose their desired levels of NHS and DSS levels by referring to the leaderboard. The Node Self-Governance System creates a financial incentive for node operators to achieve a high NHS, as this will result in more particles being stored on the node and more participation in network functions. Over time, this will increase the robustness of the Nillion Network, as nodes that do not maintain high performance will be disincentivized from continuing to participate in the network. The NHS will be based upon on-network Node Health Factors, and the DSS will be based upon DAO Support Factors, as discussed below. All factors and their impacts on NHSs and DSSs will be clearly outlined for node operators by nilDAO.

Node Health Factors: The historical and verifiable on-network performance of a node is generally an accurate predictor of future node performance. As such, nodes will receive an NHS based on several on-network factors that relate directly to the likelihood that a node will perform at or above network standards. These Node Health Factors are compiled from the immutable record of the operational performance of a node. Factors affecting the NHS include node uptime, length of time on the network, number of successful calculations

performed, error rate, slashings, and response times. In the future, other on-network factors may be added into the NHS calculation by the nilDAO.

Staking by node operators will also factor into the NHS of a node. Nodes that stake NIL tokens beyond the minimum requirement will increase their NHS. Conversely, actions that result in slashing simultaneously lower a node's NHS and its staking deposit, creating a double penalty for malicious nodes. Over time, the Node Health Factors will lead to operational natural selection for the network, resulting in accumulating value for high-performance nodes and dissipating value for low-performance nodes.

DAO Support Factors: Full Nodes, which hold blinding factors, play a large role in the overall security of the network. As such, the operator of a Full Node will be required to meet additional factors beyond on-network performance. These additional factors, known as DAO Support Factors, will be a combination of objective and subjective factors based on the technical diversity and external reputation of a node operator. DAO Support Factors increase the technical diversity among nodes by preventing the creation of centralized points of failure on the network. For example, a concentration of nodes controlled by the same operator, running on the same cloud network, or located within one geographical region increases the risks of a "black swan event" impairing the network. The objective criteria will be combined with external reputation to generate an overall DSS for a Full Node. By external reputation, we mean the perception of trustworthiness attached to real-world identities, as opposed to pseudonyms and on-network factors. The benefit of using external reputation instead of purely on-network factors is that it links trustworthiness to a broader swath of verifiable data points beyond pseudonymous performance records. In other words, the impact of real-world reputation attached to a person or established company can be an important factor in predicting future performance and consistency of action. In addition, external reputation evaluation can ensure that the network is sufficiently decentralized by preventing multiple Full Nodes being operated by a single party. nilDAO will assess all of the DAO Support Factors and generate a DSS for any node operator who wants to become a Full Node. Node operators that do not meet the minimum DSS requirement, as determined by nilDAO, will not qualify to become a Full Node.

The Nillion Node Self-Governance System strongly disincentivizes malicious actors and protects the network against Sybil attacks. A malicious actor creating hundreds of nodes will be unable to stake sufficient NIL tokens or insert a critical mass of Full Nodes to successfully attack the network. Moreover, the NHSs and staking create a strong incentive for node operators to maintain and be part of a reliable, performant, and robust permissionless network.

Note: While this section outlines important elements of our vision for the evolution of the node ecosystem, the nilDAO, acting in the best interests of the network, will ultimately determine the standards that create the most robust, hardened, and secure network. We plan to release focused papers on the tokenomics, staking mechanics, and Node Self-

Governance System features in the future.

5 Applications

The Nillion Network will be developed in multiple phases, each contributing additional depth to the network capabilities and ultimately culminating in the full-featured secure processing layer. Phase 1 of development will focus on the construction of a secure, public network for the storage and use of private information, and will establish the accompanying node and token infrastructure. Phase 2 will focus on the development of a Meta Layer that enables the Nillion Network to enhance and connect different blockchains, thus providing interoperability and cross-chain computation solutions. Phase 3 will be the implementation of the final vision of Nillion – the secure processing layer, which could give rise to a whole host of new decentralized use cases.

5.1 Phase 1: The Fort Knox of the Metaverse

Phase 1 of the Nillion Network lays the foundation for a "Fort Knox of the Metaverse" and enables the creation of applications that safeguard the world's most valuable digital secrets.

Within crypto, applications for private key storage and retrieval will allow users to upload their private keys to the network to be stored in an unconditionally secure (ITS) and particalized form. Only the user or those granted authority will be able to access the private key through authentication access points. Authentication will be performed and computed by decentralized nodes unaware of what function or underlying information they are actually relaying or computing. One iteration of such an application would be the integration of this private key storage solution within a crypto wallet. The wallet would then be able to sign transactions on behalf of the user without needing the private key to be held on a device or centralized server (discussed further in Section 5.2). Such applications would provide an extremely secure storage solution as well as offer a more decentralized method for wallets to interact with private keys.

In the real world, applications can be built to store official documents on Nillion, such as identity documents, deeds, wills, titles, contracts, and agreements. Individuals will be able to link these documents to a set of decentralized authentication factors, whereby only the individual will be able to access or edit the contents. Users could even pre-program alternative circumstances upon which data can be retrieved (e.g. based on multi-party biometric authentication), should a primary mechanism fail, or a user become incapacitated (e.g. in the case of death). Furthermore, official purchasing documents such as deeds or other entitlements could be transferred between parties without the need for an intermediary, which would lead to a reduction in impersonation and document fraud. Nillion can also enable trusted payment ecosystems in which the identity of all parties can be verified as a core element of the transaction, reducing transaction costs and losses due to fraud, which can be substantial.

Similarly, businesses will be able to store information on Nillion, such as customer

data, secret formulas, trade secrets, and any other intellectual property, in a manner that would be significantly more secure than existing encryption and centralized server solutions. Nillion's Multi-Factor Authentication, which can include biometric information, geolocation data, device identifiers, passwords, and pins, will make it significantly harder for malicious insiders to access and steal private company data because all the authentication factors are transformed into ITS particles and distributed securely throughout the network. Such business applications will help companies protect themselves from the many data breaches, hacks, and customer information leaks that regularly occur in current status quo solutions.[33]

Finally, these same "Fort Knox" applications could have valuable use cases for governments to store secure and/or confidential data, documents, or records. In the event that government regulations prevent the use of nodes on public networks, governments could run a set of their own, exclusive, federated nodes on Nillion. In this way, they could use their own private nodes to store data while at the same time leveraging the hardened and highly secure Nillion infrastructure.

5.2 Phase 2: The Meta Layer for the Blockchain

Looking forward, the unique computational ability, flexibility, and speed of NMC nodes will enable Nillion to act as a Meta Layer between and for blockchains. The Meta Layer will enable enhanced functionality, interoperability, decentralized off-chain processing (including smart contracts), and privacy-as-a-service to be offered to and between existing blockchains and their native communities. Instead of competing, Nillion will integrate with other chains and leverage the security, ecosystems, and communities of every chain, thus enabling a new design space to interact with. While there are a plethora of interesting potential use cases for such a Meta Layer, and while the purpose of this paper is not to explore or even canvas each potential use case, examples of possibilities illustrating decentralized applications and use cases envisioned by the Nillion Network that have significance to the current decentralized landscape are presented below (in no particular order).

Private smart contracts on public blockchains: One key use case is the enabling of decentralized private smart contract compute and a secure execution environment for public blockchains (e.g. Ethereum). As previously established, Nillion is able to execute algorithms confidentially in a decentralized network with agreement in the presence of bad actors. Let us consider the example of running private smart contracts for Ethereum on Nillion. In this case:

- The user writes the smart contract in Solidity.
- The user sends it to Nillion for execution.

[33]Morris Chris, 'Data Breaches in 2021 Already Top All of Last Year' (Nasdaq, 21 October 2021) <https://www.nasdaq.com/articles/data-breaches-in-2021-already-top-all-of-last-year-2021-10-21> accessed 1 February 2022

- The user also sends a hashed version of the inputs to Nillion's smart contract in Ethereum.
- The smart contract is executed in Nillion with full confidentiality.

If the result of the private computation is:

- Private – Nillion nodes calculate an authenticated hash from the computed particles that represent the result from the smart contract execution. A node selected according to a randomized algorithm writes the authenticated hash into Nillion's smart contract in Ethereum. If this node fails to do so or writes the wrong hash, the next node down the algorithm's list performs this operation, and so forth. By design, the result cannot be reconstructed in Ethereum, but the hashes constitute cryptographic evidence that can be used as proof of execution and for conflict resolution.
- Public – Nillion nodes exchange authenticated shares corresponding to the blinding factors used in the smart contract execution. Bad actors can send wrong shares; thus each node keeps only the shares that are successfully authenticated and discards the rest. Each node then reconstructs the blinding factors using Lagrange polynomial interpolation and obtains the smart contract execution result from the computed particle(s). Using the same mechanism as described above, a selected node writes the particles, the authenticated shares, and the reconstructed result into Nillion's smart contract in Ethereum. Every Ethereum node is then able to verify that the shares were properly authenticated, and that the result from the smart contract execution was reconstructed by them using Lagrange polynomial interpolation. As Lagrange polynomial interpolation simply involves addition and multiplication by a constant, the verification process is therefore very lightweight; however, it is still able to guarantee agreement, even in the presence of bad actors.

Notice that this mechanism could also become an alternative to optimistic rollups and ZK-rollups. In particular, Nillion could potentially scale better than existing Layer 2 solutions that rely on ZKP, which is notoriously slow and requires the centralized holding of information in order to compute. In comparison, Nillion will be able to compute the result particles at centralized server speed without having to generate a ZKP, but while still supporting verifiable result reconstruction. Therefore, it could outperform ZKP as a scaling alternative, both in terms of confidentiality, as functions with private inputs from more than one party can be computed, and speed, as it has no need to compute a ZKP. This will undoubtedly constitute a use case of extreme interest once the Nillion Network has been built.

Interoperability: Nillion will enable blockchain interoperability solutions to allow for the transmission of assets, events, and data. For example, in a *relay* interoperability solution [20], Nillion could become the *sidechain* and address the existing trade-off between

sidechain throughput and security. This could be done using Nillion's Byzantine agreement mechanism, which is instant and does not require the nodes to exchange any messages. Thus, a Nillion sidechain could guarantee Byzantine agreement with many nodes (high security) with the speed of a centralized solution (high throughput). As another example, Nillion could be used to implement a decentralized *notary* interoperability scheme [20], whereby the notaries are the Nillion nodes operating under Byzantine agreement. Using a threshold multi-signature scheme, the Nillion nodes would have the ability to sign a transaction while guaranteeing that no group of nodes below the threshold can sign or reconstruct the private key. The multi-signature scheme itself can be implemented in Nillion so that the private key is stored in particalized form and controlled through a threshold-based NMC voting algorithm.

Advanced cross-chain solutions become possible when built on NMC-enabled interoperability schemes. Multi-Party Programs running in the Meta Layer could be used to link complex systems that span chains and protocols, providing scaffolding for bridges, liquidity aggregators, and cross-chain decentralized exchanges. Value could potentially be bridged into and held on the network, creating further design space for chain-agnostic applications and providing users on-demand access to liquidity on any chain on-demand, with Nillion's contracts on each chain interacting as a proxy. As a second-order effect, increased use of the Nillion Network for this style of cross-chain interaction will also increase the privacy of the multi-chain by organically attracting large pools of liquidity across all chains. Every wallet that is swapping from and bridging into the network to take advantage of cross-chain Nillion infrastructure will increase the privacy of the network's users, as transactions are increasingly "lost in the crowd" between chains. Privacy-conscious users would therefore no longer be limited to "tumbling" and withdrawing a single asset from a single chain, and could use their anonymized funds to withdraw, swap, and interact with smart contracts across *any* chain.[34] As the network scales, the volume of varied transactions performed across chains will endow an ever-growing amount of privacy to users that would be unfeasibly complex to expose.

A unified multi-chain wallet with ITS authentication: Nillion enables a true multi-chain wallet by allowing a user to securely store all the private keys from different blockchain wallets in a single, completely decentralized, Nillion account or wallet (that is itself secured via authentication factors that are decentralized and stored in Nillion). The keys are kept in particalized form so that an attacker would have to compromise an unfeasibly large proportion of all Nillion nodes in order to piece them back together. NMC also has the ability to perform computations across these particles, allowing a multi-chain wallet to sign transactions and interact with decentralized applications without ever having to reconstruct the keys on a centralized server, and without the need for the nodes to exchange messages. Nodes turn the transaction into particles, so that each Nillion node ends up with one particle from the transaction and one from the key. Then, each Nillion

[34]Privacy-as-a-service "tumbling" services that enable private transactions are currently provided by applications such as Tornado Cash, Heiswap, and Aztec.

node follows a signing algorithm implemented with NMC in order to process its particles and generate a result particle. This does not require any inter-node communication and can therefore occur almost instantly. The result particles are then sent back to either the user or the designated Result Node, which locally reconstructs the signed transaction to be sent to the corresponding blockchain. A key element of this solution is user authentication to facilitate access to all the private keys under a single high-assurance MFA account. To this end, the *authentication* service will be used (see Section The Nil Service Layer).

Gated NFT metadata and content: The privacy and granular control of access to secure information enabled by Nillion's design unlocks the next generation of use cases for NFTs on any chain. When tokens point to data hosted in the network, Nillion can offer tools to gate access to the metadata or underlying content, providing minters and owners the ability to decide who can see or interact with their property. Creators can use this to test, seek external feedback, and launch without the fear of bots finding their hidden uploads early and unfairly trading on unrevealed information. Collectors, if the creator enables these features, could grant exclusive access by picking and choosing who gets to see the works stored across Nillion's secure nodes. These same tools could also enable a whole suite of new web3 platforms that could disrupt traditional membership models (e.g. Patreon, Substack) that rely on limiting access to content. Coupled with the composability built-in to the format, these new membership models could create novel ways for community interaction and discoverability. The airtight security of the network means sensitive information can be handled on public blockchains without compromising the underlying data to prying eyes. Data could be uploaded, signed, or timestamped on any blockchain and pieces of the information could be made public as desired.

DAO tools: DAO tools represent another use case that takes advantage of the Nillion Network's privacy features. NMC can support decentralized anonymous voting through the creation of a voting core service. After successful authentication, each party casts their vote, which is transformed into particles that are distributed across the Nillion Network. The Nillion nodes locally process the particles in order to count the votes and obtain a particle from the result. The particles and shares are then exchanged so that all the DAO parties learn about the vote count.

The network can also help DAOs manage private information. Upon creation, a DAO's core member(s) may elect to store their identities or the DAO's legal documentation (e.g. private legal entity or incorporation documents) in the Nillion Network. This provides a decentralized secure private enclave in which sensitive DAO documents can be stored. The same systems can be used to allow DAO representatives to comply with vote- or majority-based confidentiality, or Non-Disclosure Agreements (NDAs) and secrecy provisions, or to handle other sensitive information (e.g. online bank account access information or private non-open source code) without preventing the DAO's ability to later access the data if needed. For example, for a situation in which a qualified majority of votes is collected, the information is automatically revealed to the DAO parties so that legal measures can

be taken. This provides new members or investors in a DAO with peace of mind, while allowing good actors to stay anonymous or manage their DAOs effectively.

As previously mentioned, the above use cases represent just a small handful of the possibilities associated with the Nillion Meta Layer. Similar to DLTs such as Ethereum, we expect a wide range of interesting applications to be created once the network is fully deployed and its capabilities are able to be comprehensively explored by developers and the community.

5.3 Phase 3: The Universal Decentralized Secure Processing Layer

The larger vision of the Nillion Network is to become the default layer for secure processing in web3 and beyond. While this means that Nillion will have a significant impact on the current crypto/blockchain ecosystem (see Section 5.2), the effect that enhanced, secure, and economical decentralized processing has going forward will likely mirror the impact that improved processing had on software applications over time. Put another way, better and more secure decentralized processing opens up the decentralized landscape to new and potentially unexpected interactions and industries.

Just as DeFi and web3 pioneered exciting applications and primitives based around the on-chain transaction processing power of blockchain, Nillion will scaffold innovations by offering fast and secure general-purpose processing for everything else. The off-chain (i.e. NMC-based) yet fully decentralized execution of smart contracts, private enclaves for data storage, and Meta Layer connectivity allow decentralized participants and protocols to interact in novel ways, creating various possible use cases, such as private payment processing, chain-agnostic applications, and deeper integrations between traditional and decentralized finances. Guarantees provided by the network's retic Security, particalization, and blind data processing unlock design space for builders working with identity, health records, compliance, biometrics, facial recognition, and all sorts of sensitive user information that require user-owned and GDPR-compliant tooling. Forward-looking organizations, or even governments, who rely on mission-critical security will gain access to quantum-proof decentralized data and compute infrastructure by leveraging the Nillion Network. These are just a few of the many possible outcomes that we see Nillion enabling in the longer term.

Nillion seeks to continually push the cutting edge of decentralized processing. NMC is the first of many technological innovations that we expect to develop to further that goal. At the time of this whitepaper, several additional inventions are in development that will further add to the already state-of-the-art decentralized processing advancements that constitute Nillion technology. In the medium to long term, we contend that these continual improvements to, and innovations in, decentralized processing will serve as the catalyst for the disruption of new industries and use cases, and will become a key strategic differentiator of the Nillion Network.

6 A Final Note: The Nillion Ecosystem

The primary purpose of this paper is to describe the technical elements and explore some of the implications of Nillion and NMC technology. However, the reality is that the success of Nillion, like that of all other technology, depends on adoption. Beyond including innovative technology, crypto projects in this day and age must create applications that have real-world functionality and adoption. Projects such as Terra have proven this true by adopting a strategy according to which the in-house development of initial use cases (e.g. Anchor, Mirror) led to significant ecosystem growth. Similarly, as part of its founding team, Nillion will have an internal team specifically tasked, and compensated in line, with the creation and popularization of several of the key use cases in both Phases 1 and 2. This internal team of Founding Entrepreneurs (FEs) will form part of the early-stage protocol development team so that they gain a comprehensive understanding of the network and its capabilities, while creating efficient feedback loops that guide network development. We expect the FEs to contribute in a meaningful way to the initial and ongoing traction of the Nillion ecosystem.[35]

Building a thriving ecosystem is a key part of the success of Nillion. The FE strategy, an SDK for third-party developers to create additional applications, several proprietary referral-based systems that will propel adoption, and other significant community growth initiatives will collectively support the propagation of the Nillion Network and the adoption of the ecosystem. While specific details of these growth initiatives are beyond the scope of this paper, this section is written in recognition of the market reality that it is not only great technology, but product, community, and go-to-market, which will determine the overall success of the Nillion Network.

7 Conclusion

We have proposed an NMC-based decentralized architecture which is distinct from blockchain, SMPC, and computational encryption technology. Unlike blockchain, network nodes do not run immutable ledgers designed to store transaction data. Unlike SMPC, nodes do not have to communicate with one another. Unlike computational encryption, data is held in a cryptoanalytically unbreakable manner and does not rely on the hardness of a mathematical problem to achieve security.

The concept of an SMPC-based decentralized network that is open-ended by design is well-suited to serve as the foundational secure processing layer that will provide fast compute and post-quantum storage to both blockchains and users alike. Such a platform has the unique potential to build on Bitcoin's and Ethereum's contributions, enabling a substantial array of new blockchain and non-blockchain applications in a decentralized, fault tolerant, and permissionless manner.

[35] Entrepreneurs potentially interested in joining the FE team can email founding@nillion.com.

References

[1] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 1st ed., 2015.

[2] Y. Liang, H. V. Poor, and S. Shamai (Shitz), "Information theoretic security," *Foundations and Trends in Communications and Information Theory*, vol. 5, no. 4–5, pp. 355–580, 2009.

[3] A. C. Yao, "Protocols for secure computations," in *23rd annual symposium on foundations of computer science*, pp. 160–164, 1982.

[4] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing (STOC'87)*, pp. 218–229, 1987.

[5] D. Beaver and S. Goldwasser, "Multiparty computation with faulty majority," in *CRYPTO '89: Proceedings on Advances in Cryptology*, pp. 589–590, 1989.

[6] A. Ben-Efraim, Y. Lindell, and E. Omri, "Optimizing semi-honest secure multiparty computation for the internet," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 578–590, 2016.

[7] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[8] C. M. Williams, R. Chaturvedi, and K. Chakravarthy, "Cybersecurity risks in a pandemic," *Journal of Medical Internet Research*, vol. 22, no. 9, 2020.

[9] L. Peng, W. Feng, Z. Yan, Y. Li, X. Zhou, and S. Shimizu, "Privacy preservation in permissionless blockchain: A survey," *Digital Communications and Networks*, vol. 7, no. 3, pp. 295–307, 2021.

[10] J. Abadi and M. Brunnermeier, "Blockchain economics," tech. rep., National Bureau of Economic Research (NBER) Working paper w25407, 2018.

[11] A. K. Lenstra, "Integer factoring," *Designs, Codes and Cryptography*, vol. 19, pp. 101–128, 2000.

[12] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st annual ACM symposium on Theory of Computing (STOC)*, pp. 169–178, 2009.

[13] M. v. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Advances in Cryptology – EUROCRYPT 2010*, pp. 24–43, 2010.

[14] U. Maurer, "Information-theoretic cryptography," in *Advances in Cryptology – CRYPTO '99*, pp. 47–65, 1999.

[15] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Advances in Cryptology – CRYPTO '91*, pp. 420–432, 1991.

[16] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *CRYPTO 2012: Advances in Cryptology. Lecture Notes in Computer Science*, pp. 643–662, 2012.

[17] N. Smart, *Cryptography: An Introduction*. McGraw-Hill New York, 3 ed., 2003.

[18] S. Krenn, T. Lorünser, and C. Striecks, "Batch-verifiable secret sharing with unconditional privacy.," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, pp. 303–311, 2017.

[19] L. Lamport, "The weak byzantine generals problem," *Journal of the ACM*, vol. 30, no. 3, pp. 668–676, 1983.

[20] V. Buterin, "Chain interoperability," *R3 Research Paper*, vol. 9, 2016.

[21] D. Beaver, S. Micali, and P. Rogaway, "The round complexity of secure protocols," in *Proceedings of the 22nd annual ACM symposium on Theory of Computing (STOC)*, pp. 503–513, 1990.

Special Thanks

We would like to recognize and give special thanks to all of the individuals who contributed their time in reviewing and providing feedback on this whitepaper (listed alphabetically):

Bryan Gross, Claire Kelly, Conor O'Higgins, Conrad Whelan, Dan Paikowsky, Elizabeth Quaglia, Gavin Galloway, Jack Bicer, John Kenevey, Lucy Wang, Lukas Bruell, Mark McDermott, Matt Sweeney, Michael Housman, Patrick Curry, Roel Nuyts, Slava Rubin, Taran Sabharwal, and William Reinisch.