The Nym Network

The Next Generation of Privacy Infrastructure

Claudia Diaz [†], Harry Halpin [‡], and Aggelos Kiayias [§]

Nym Technologies SA

Version 1.0
February 26th 2021

Abstract. The Nym network ("Nym") is a decentralized and incentivized infrastructure to provision privacy to a broad range of message-based applications and services. The core component of Nym is a *mixnet* that protects network traffic metadata for applications, providing communication privacy superior to both VPNs and Tor against global adversaries that can watch the entire internet. Nodes in the mixnet are rewarded via a novel *proof of mixing* scheme that proves that mix nodes are providing a high quality of service. Rewards given by NYM tokens allow anyone to join the Nym network and enable a sustainable economic model for privacy. NYM tokens can be transformed into *anonymous credentials* that allow users to privately prove their "right to use" services in a decentralized and verifiable manner. The Nym network can serve as the foundation for a vast range of privacy-enhanced applications that defend the fundamental freedoms of people across the globe against traffic analysis by powerful adversaries.

1 Introduction

The current lack of privacy on the internet exposes billions of people to mass surveillance and data breaches, undermining trust in digital services and stifling innovation. Thanks to the failure of governments to ban strong encryption during the original "crypto wars" of the 1990s [72], open source libraries like OpenSSL created an opening for new services like Paypal, Amazon, and eBay to emerge – which would have been impossible without the end-to-end encryption between browsers and websites needed to secure financial transactions. Today is eerily similar: privacy is undermined by pervasive data collection and centralized monopolies, preventing innovative services and platforms from arising. Yet the tide is turning due to popular discontent against mass surveillance and the new possibilities opened by Bitcoin for creating incentive mechanisms that enable a network to cooperate and self-sustain. Inspired by this, Nym puts forward new foundational standards and open-source libraries for privacy that can enable previously inconceivable applications and markets. Nym is a permissionless and incentivized network designed to defend user privacy, even against corporations and government actors with the capacity to capture all global internet traffic. The Nym network provides a scalable privacy infrastructure to support third-party applications and services in offering private access features to their users.

Privacy tools sufficient for tackling the scope of private information exposure are lacking today. Privacy must be understood as a holistic property because leakage of information in one layer of a system can undermine the privacy provided by even the most advanced cryptographic protocols in another layer. Taking blockchains as an example, the "Layer 0" peer-to-peer broadcast of any "Layer 1" blockchain is usually the weakest and most neglected link of blockchain systems. Peer-to-peer broadcasts expose IP addresses, timing, and other network metadata that can deanonymize transactions even when a blockchain uses zero-knowledge proofs on-chain [6]. This fundamental issue goes far beyond cryptocurrencies, affecting all internet users. The primary protocol all Web browsers use to encrypt their connections to websites, Transport Layer Security (TLS), can be attacked via traffic analysis to reveal the contents of encrypted communications [21]. The analysis of network traffic metadata also reveals the content of encrypted Domain Name System (DNS) queries in protocols such as DNS-over-TLS (DoT) and DNS-over-HTTPS (DoH) [98]. The

[†] Associate Professor at KU Leuven and Chief Scientist of Nym Technologies SA.
[‡] CEO of Nym Technologies SA.
[§] Chair in Cyber Security and Privacy at the University of Edinburgh and Chief Scientist at IOHK. Advisor to Nym Technologies SA.

internet was simply not designed with privacy as a fundamental property at its inception, and today we are living with the consequences of those design choices.

Nym is an overlay network that supports private access features for applications and their users. The Nym network is composed of a decentralized **mixnet** (a network of mix nodes) [89] and an **anonymous credential** cryptosystem [101]. The mixnet routes network traffic in a way that makes it untraceable, even for powerful global adversaries that can observe the whole internet. This is in contrast to VPNs or Tor, which are designed to protect against weaker adversaries that are limited to just a few network locations. Anonymous credentials allow users to prove their "right of use" when privately accessing services over the mixnet, while preserving unlinkability. A blockchain maintained by validators decentralizes the operations of the entire Nym network, including the membership and configuration of the mixnet, the issuing of credentials and the distribution of rewards. The NYM token[1] rewards nodes that provision a high quality of service via a novel proof-of-work scheme, **proof of mixing**, which proves that a mix node has been adequately routing traffic. NYM tokens are used to pay transaction fees in the Nym network and so allow access to the Nym network.

Nym's combination of a mixnet and anonymous credentials provides "full-stack" private access features that can be flexibly integrated with myriad services. For example, Nym can anonymously broadcast transactions to any blockchain and do so with better network-level privacy than either Tor or Dandelion [42,46]. However, network-level privacy for blockchains is only one of the many kinds of service Nym can support – *any* networked application, including those that do not use blockchain technology, from secure messaging to file-sharing, can benefit from Nym's privacy features. By providing a communication infrastructure that can support a broad range of applications, Nym is able to blend large and diverse user bases into one massive anonymity set. This is crucial in a privacy system as *anonymity loves company*: In order to be anonymous when using a system, one must be indistinguishable among a large group of users [41]. Thus, in offering a generic infrastructure shared by many applications, Nym provides stronger privacy guarantees than is possible for a standalone network dedicated to a single blockchain or privacy-enhanced application. Furthermore, mixnets scale horizontally, as a growing user base and increased demand can be serviced by simply adding more nodes to the mixnet. In turn, increased traffic allows for lower mixing latency and lower levels of cover traffic while maintaining high anonymity. Nym offers a positive relation between privacy and scalability: the more users join the network, the better the privacy and performance trade-offs for all users.

Table of Contents. This whitepaper lays out the different aspects of the Nym network and is organized as follows:

- Section 2 discusses the problem of privacy with a focus on network-level surveillance.
- Section 3 introduces the overall architecture of the Nym network, the considered threat model, requirements, and privacy properties.
- Section 4 gives a thorough description of the Nym mixnet.
- Section 5 provides an in-depth explanation of Nym credentials.
- Section 6 discusses the cryptoeconomics that allow the Nym network to evolve to meet demand.
- Section 7 outlines possible use-cases and the next steps to defend privacy and freedom on the internet.

This paper presents the concept, overall architecture and main components of the Nym network, and the functioning principles that underpin its design. The specific algorithms and implementation details of each part of the system will be fleshed out in separate documents. We expect the components of the Nym network to evolve over time.

2 Network-level surveillance

The internet as we know it was not designed for privacy. Cryptography was bolted on as an afterthought to protocols such as those used for instant messaging and web browsing, which otherwise expose data in plaintext to anyone who can eavesdrop on communications. Today, widely deployed secure web browsing and messaging protocols like TLS and Signal implement end-to-end encryption to protect the content of communications. However, even if data payloads are encrypted, these protocols still expose *metadata* to network eavesdroppers, including information such as who communicates with whom, when, how much, how often, for how long.

[1] To distinguish the Nym network from the NYM token, we use the convention to capitalize the token as "NYM" and to call the network itself "Nym."

Besides acting as a side channel that may reveal encrypted content [21, 63, 84, 98], the network-level metadata associated to online activities can be exploited to uniquely identify users, enable pervasive tracking, and reveal private information about personal or business activities [39, 51, 79]. By applying "artificial intelligence" algorithms to metadata, it is possible to predict people's political preferences, associations, social relations, purchasing behavior and other intimate details of their private lives and minds [77]. In addition to the current widespread corporate exploitation of metadata for advertising and sales purposes, the documents revealed by Edward Snowden provided evidence of metadata collection and analysis being carried out *en masse* by the US National Security Agency (NSA) and similar agencies around the world [77].

In the case of cryptocurrencies, the lack of network-level privacy allows even private companies like Chainalysis to monitor not just the data available on blockchains but also the metadata of the peer-to-peer traffic of blockchain transactions [22]. An entity that observes a significant percentage of the peer-to-peer traffic of a blockchain can discover the network addresses that broadcast particular transactions, identify users, and make inferences about private transactions even in systems with on-chain protection such as Zcash or Monero [6, 92]. For any blockchain, including Bitcoin and "Layer 2" solutions like the Lightning Network, the lack of privacy at the network level creates opportunities for malicious parties to monitor peers' traffic and use the traffic data they collect to undermine privacy and carry out large-scale attacks [34, 53].

As shown next in Section 2.1, existing solutions like VPNs, Tor, and peer-to-peer variants do not adequately defend from powerful real-world adversaries that can watch the entire network and correlate traffic. Section 2.2 argues that there's no "free lunch" in terms of privacy, and that economic incentives built into the heart of a mixnet-based system are the key to build a privacy infrastructure that scales to billions of users.

2.1 Existing solutions for network privacy

VPNs. The most widely used means of improving network-level privacy are VPNs (Virtual Private Networks). VPNs build an encrypted tunnel between a user's client device and a centralized server run by a VPN provider, which acts as proxy that forwards the client's communications. VPNs are often misconfigured in ways that compromise security [68, 86] and even when configured correctly, VPNs are trivial to monitor and censor by adversaries with just local network access at the VPN server. Such network adversaries can correlate traffic in and out of the VPN server to reveal the identity and destination of the VPN user, as well as block the VPN from being accessible [65].

In terms of their trust model, VPN providers can *fully* observe all network traffic between their users and the public internet, meaning that VPN providers know exactly what services their users are accessing at all times. VPN providers are thus fully trusted parties even though they may not be trustworthy in practice. For example, users trust VPN providers to not keep any logs. However, this is rarely the case, as most VPN providers keep logs, even if they tell their users they do not [76]. Worse, as VPN providers usually charge for their services, the detailed history of each of their user's online activities can be linked by VPN providers to personal data (legal name, address, credit card number, and so on) through the information disclosed in the payment.

Recently, a number of decentralized VPNs (dVPNs) have been launched, such as Orchid [15] and Sentinel [93]. These designs successfully decentralize the single point of trust in the VPN provider to multiple VPN providers, but can still easily be deanonymized by network adversaries that watch the dVPN entry and exit points to correlate traffic; consequently, they do not offer the level of privacy needed for high-value applications. Nym can complement rather than compete with decentralized VPN providers, as a dVPN can provide entry points to the Nym mixnet for users who want to conceal towards their local network the fact that they are using Nym. At the same time, Nym adds a layer of privacy to dVPNs that the VPN technology does not offer by itself.

Tor. With more than six thousand nodes and an estimated eight million daily users [75], Tor is currently the largest and most widely used anonymous overlay network [42]. Tor is based on *onion routing* [49], a protocol where the client selects a set of nodes (typically three) and builds a multi-hop connection that uses those nodes as a sequence of intermediate proxies. Clients encrypt data multiple times, each layer of encryption corresponding to a node in the route. Each node then removes a layer of encryption and forwards the result to its successor in the route. In contrast to centralized VPN solutions, none of the Tor nodes by

itself has visibility on both the network address of the client originating the traffic and the final destination: only the first node (*guard*) can see the client and only the last node (*exit*) can see the destination.

Even though Tor protects metadata much better than VPNs, it is designed to do so only against network adversaries who have limited ability to monitor the entire network. Network adversaries that observe *both* the traffic incoming and outgoing from Tor can perform end-to-end correlation attacks and deanonymize the connection [62,71,74,83,96,97]. Even when adversaries only have access to a client's local connection, it is possible to identify web pages visited by the user via Tor through website fingerprinting attacks, which exploit persistent and distinctive traffic patterns in web traffic that the Tor network leaves unaltered [63,84]. These attacks are possible because even though onion routing cryptographically transforms data packets such that inputs and outputs are content-wise unlinkable, it maintains linkability for all the packets in a connection and does not add timing obfuscation or cover traffic. As result, distinctive metadata patterns act as a traffic fingerprint that can be exploited to link connections end-to-end and deanonymize users.

Regarding the trust model, even though the Tor relays themselves are run by volunteers in a decentralized fashion, the Tor network includes semi-centralized trusted components, such as the directory and measurement authorities. The (currently ten) directory authorities are chosen and hard-coded by the Tor software developers, and they determine the list of Tor nodes that is distributed to clients to create their connections. Tor is thus a permissioned system in which all users must fully rely on these directory authorities to determine which nodes are part of the Tor network at any given time.

Peer-to-peer networks. The "Invisible Internet Project" (I2P) is a peer-to-peer alternative to Tor, meaning that in I2P, all participants simultaneously serve both as clients that originate communications and as proxy routers that relay communications for others [58]. In a similar manner to Tor clients, I2P clients build multi-hop connections where data is layer-encrypted once per hop in the route. Differently from Tor, I2P uses short-lived unidirectional tunnels instead of long-lived bidirectional circuits. I2P's variant of routing is called *garlic routing*. I2P's trust model avoids relying on centralized components and replaces directory authorities with a Distributed Hash Table (DHT). Yet the issue of how to design a secure and private distributed hash table remains an open research question [102]. In practice, I2P is open to a number of attacks that censor, isolate (eclipse), misdirect, and deanonymize users [45,54,55,61]. Like Tor, I2P aims to defend only against local network adversaries who do not have the ability to monitor most of the network — while global network adversaries can successfully trace and correlate traffic flows end-to-end, linking source to destination and defeating the anonymity of I2P.

The same problems affect Dandelion [8]. Dandelion probabilistically forwards transactions similarly to Crowds [90]. Each intermediary receiving a transaction probabilistically decides either to forward it onward or to broadcast it to the full network. Again, this does not provide protection against network adversaries that observe the communications of all peers and perform timing correlation attacks to trace data flows through the network. Running Dandelion over Tor leads it to inherit all the aforementioned problems of Tor [46].

HOPR is a peer-to-peer network for anonymous messaging with payments per hop [56]. Similarly to I2P, HOPR suffers from various vulnerabilities and shortcomings that are common to peer-to-peer anonymity network designs: partial network knowledge makes it susceptible to eclipse attacks [99,100] as well as predecessor attacks [104]; the dual function of participants as clients and relays exposes users to enumeration of their network addresses; and the thin spread of traffic allows global network adversaries to trace packets through timing and traffic analysis attacks. In addition to challenges that are common to peer-to-peer anonymous routing, HOPR's payment protocol leaks information that undermines the protection provided by onion routing and further enables intermediate hops to deanonymize the messages they route [91].

Mixnets. A mix network, or *mixnet*, is an overlay network of mix nodes that routes messages anonymously. Similarly to Tor, each packet relayed via a mixnet is encrypted multiple times [29] and sent through a sequence of nodes, each of which removes a layer of encryption, until the last mix node sends the message to its final destination. Differently from onion routing, however, mixnets route each individual message *independently*. Critically, what differentiates mixnets from Tor and peer-to-peer networks is that mixnets are designed to provide metadata protection from *global network adversaries*. Mix nodes achieve this by reordering the messages they route in addition to transforming them cryptographically. This makes messages untraceable both in terms of appearance as well as timing. Mixnets thus provide stronger metadata protection than Tor or any of its peer-to-peer alternatives. Mixnets did not take off the same way Tor did

due to concerns over the cost and latency of bandwidth and public key operations that made the design seem impractical at the turn of the millennium when Tor was launched. Yet today, bandwidth and computation are cheaper and faster than ever before and, given the scope of modern traffic analysis, mixnets look increasingly the best available option for achieving communication privacy.

The mixnet concept was proposed by Chaum in the early 1980s [16] and early deployments include the Cypherpunk remailers (first developed by Eric Hughes and Hal Finney), Mixmaster [23] and Mixminion [28]. Loopix [89] is an advanced mixnet for anonymous email and instant messaging. Nym extends Loopix into a general-purpose incentivized mixnet architecture. The Nym mixnet is introduced in Section 3 and described in detail in Section 4.

2.2 Economic incentives and scalability

Despite increased public awareness and interest in privacy over the last years, no system has yet successfully taken advantage of cryptocurrency-based incentives to create an economically sustainable network that provides privacy against powerful network adversaries at scale. At the same time, blockchains that feature incentives still depend on peer-to-peer broadcast networks that often provide little to no privacy, and in the best case protect only against weak adversaries who lack the ability to observe a large portion of the peer-to-peer network.

Tor and I2P, currently the two largest deployed anonymity networks, suffer from stagnant growth due to the lack of economic incentives for node operators. Tor is dependent on crowd-funding, non-profit, and government grants for software development and maintenance. These grants do not cover the costs of running and maintaining the relays in the Tor network itself, a cost borne entirely by volunteers. The number of Tor relays has been relatively constant at six to seven thousand over the last few years. As there are no economic incentives to run a relay, Tor cannot dynamically add new servers to scale up to meet a large increase in demand.

Tor has benefited from significant attention by researchers; it has been thoroughly studied and improved over the last fifteen years. As a result, Tor's security properties and weaknesses are well documented and understood. I2P, on the other hand, has received less attention from academic researchers, chiefly because of its lack of clear design documentation and rigorous security models and arguments. These problems are exacerbated by the fact that I2P is purely volunteer-driven. Although admirable in many regards, reliance on volunteers for mission-critical network operations may also complicate systems-level integration with other software such as cryptocurrency, as demonstrated by the failed attempt by the Monero community to develop Kovri based on I2P [69].

Incentives work. Bitcoin has demonstrated this. One fundamental premise of Nym is that well-designed incentives can sustain a large-scale privacy infrastructure as a collective public good. This challenges the status quo, making possible a new generation of services and applications that offer privacy as a feature rather than indiscriminately amassing user data. In Nym, incentives ensure that those who provision the core infrastructure are rewarded for their work. A market model for node operators, in contrast to volunteer-driven models, makes it possible for the Nym network to arbitrarily scale up to meet increased demand as the user base grows. Furthermore, in mixnets, there is a virtuous circle between privacy and scalability: both the privacy and efficiency of Nym increase as more users join. Scaling to route traffic for more applications increases the privacy provided to *all* users, as more participants are included in a larger and more diverse anonymity set. Moreover, a larger user base allows for lower latency and cover traffic rates while still guaranteeing strong anonymity to users. By following the lead of Bitcoin, Nym weaves incentive-based scalability into an economically sustainable infrastructure for provisioning privacy.

3 The Nym Network Design

Nym is designed to support privacy-enhanced access to applications and services, both in terms of protecting communications metadata and privately proving the "right to use" services. Nym leverages token-based incentives to fund operational costs, dynamically scale to meet user demand for privacy, and reward nodes for contributing to the protocols with a high quality of service.

This section introduces the concept and design goals of the Nym network. The participants in Nym are introduced in Section 3.1. Section 3.2 describes the information flows between participants that enable private communication and access to services. Section 3.3 briefly describes the flow of value in the Nym

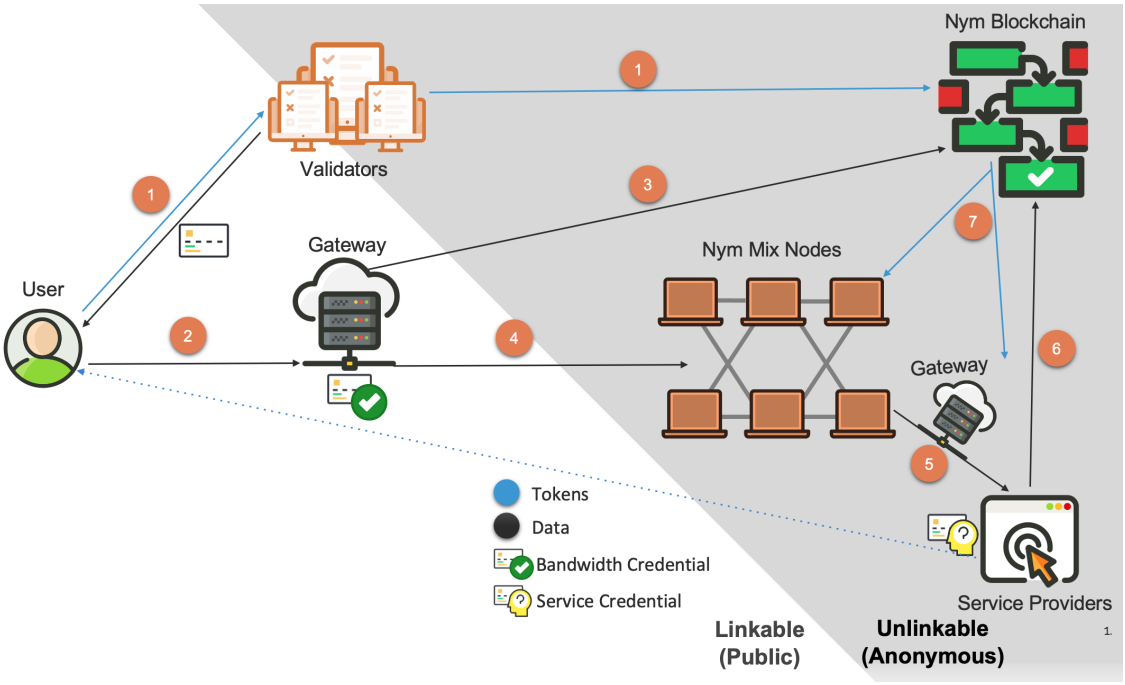

Fig. 1: Nym architecture and flows

network. The powerful threat model that Nym is meant to counter is presented in Section 3.4. Section 3.5 explains the main privacy features of Nym, while other requirements are listed in Section 3.6.

3.1 Nym Network Participants

The participants of the Nym network are shown in Fig. 1. There are three types of *nodes* that make up the Nym infrastructure: **validators**, **gateways**, and **mix nodes**, whose functions are described next in this section. Third-party **service providers** can make themselves accessible via Nym to offer enhanced privacy to **end users**. Providers can also act as an interface between the Nym network and external components that do not require modification or even awareness of Nym. For example, Nym can be used to anonymously broadcast Bitcoin transactions by means of a service provider that relays transactions received via the Nym mixnet to the Bitcoin peer-to-peer network.

End users. The user base of Nym includes the users of all the services available via Nym who choose to communicate privately, whether to anonymously broadcast a transaction, access information, or maintain a messaging conversation with a friend. The larger and more diverse the user base, the better the anonymity provided to all users and the more cost-effective the network becomes.

Mix nodes. Mix nodes provision communication privacy to end users by relaying data packets anonymously. Mix nodes are organized in a network, called *mixnet*, and packets traverse multiple nodes before being delivered to their final recipient. Mix nodes receive data packets that they both (1) transform cryptographically and (2) reorder, such that it is not possible to tell which input packet corresponds to which output, neither based on data content nor on timing. The Nym mixnet is based on the Loopix design [89]: it uses the Sphinx packet format [29], a stratified network topology [36], continuous-time mixes [66], and loops of cover traffic [30]. The Nym mixnet, described in detail in Section 4, includes further features for reliable transport, scalability, sybil protection, fair routing, quality-of-service measurements, incentives, and other modifications and extensions needed for real-world deployment.

Gateways. Gateways make the Nym mixnet accessible while protecting it from free riding. Participants may choose to always use the same gateway for all their traffic, split their traffic over multiple gateways,

or pick a different gateway every day. Gateways also cache received messages for participants who are offline or unreachable, and are instrumental in the implementation of reliable transport features. Gateways are described in detail in Section 4.2.

Validators. Validators collaboratively perform several core functions in the Nym network. They maintain the **Nym blockchain**, which acts as a secure broadcast channel for distributing network-wide information, such as: the list of active nodes and their public keys, network configuration parameters, periodic random beacons, participant's stake, deposits into the **nympool** (the shared pool of funds used to support the Nym network, described below), rewards distributed from the nympool, and any other data that needs to be available to all participants to ensure the secure operation of the network. In addition, validators issue credentials to participants in a distributed manner. **Bandwidth credentials** encode a proof of deposit in the nympool made in exchange for an allowance of data to send through the Nym mixnet. These credentials are shown to gateways to prove the right to send traffic through the mixnet. **Service credentials** can encode arbitrary attributes required for proving "right to access" to a service, including proofs of having made a deposit in the nympool to pay for the service.

Service Providers. The Nym network is not a standalone service; it is an infrastructure that supports privacy for a broad range of third-party applications and services accessible through it. Service providers can send and receive messages through the Nym network to privately communicate with their users, as well as optionally use Nym service credentials for granting paid access to their services without requiring privacy-invasive user identification.

3.2 Information flow

Figure 1 depicts the steps followed by Nym users to privately access and prove their 'right to use' a service. The flow of data is given in black, the flow of NYM tokens in blue.

(1) Deposit funds in the nympool to obtain credentials. Consider a user that has obtained NYM token, either through a purchase, bundled as part of a service package from a service provider, or earned as rewards by operating a node or services integrated with Nym. The user first makes a token deposit in the *nympool*, the shared pool of funds managed by the Nym validators used to support the Nym network. In this example, the user converts part of their deposit into a bandwidth credential and the rest into a service credential. The user generates the two unsigned credentials together with proofs of correctness and proofs of deposit in the nympool and submits them to the Nym validators. If everything is correct, the validators issue partial signatures on the credentials in a decentralized manner. The user collects a threshold number of partial signatures to combine into a valid credential.

Service credentials that encode a proof of deposit simply embed the amount of token they represent and for which they are redeemable, regardless of token value fluctuations. In the case of bandwidth credentials on the other hand, a token deposit is converted into an allowance for sending packets through the mixnet. The exchange rate between NYM token and amount of bandwidth acquired is determined on-chain in the form of an auction. In order to join the network, individual mix node operators submit a "price per packet" function as part of their node information. The exchange rate is computed considering the median of the price per packet functions of the set of nodes that are part of the network in the current time window (cf. Section 6.3). The Nym blockchain is used to serialize bandwidth credential acquisition requests, enabling validators to process and accurately price each request using the median price per packet function. Note that bandwidth credentials can contain large packet allowances; thus, credential operations are several orders of magnitude fewer than the number of packets sent thought the network.

(2) Use the bandwidth credential to get access to the gateway. The user selects a gateway according to their own personal criteria and uses a bandwidth credential to prove their "right to use" the Nym mixnet based on their deposit. The gateway checks the validity of the bandwidth credential and verifies all associated proofs.

(3) Establish temporary nym with the gateway. The gateway checks that the bandwidth credential presented by the user has not already been marked as spent in the Nym blockchain. It then submits to the validators a commitment to the serial number encoded in the credential that they publish in the blockchain. This commitment marks the bandwidth credential as used and prevents its double-spending in the future. It also allows gateways to prove the share of traffic they have routed so they can receive proportional rewards. The user and the gateway establish a shared secret and a nym (i.e., a temporary pseudonym) associated to the data allowance and validity period represented by the credential. The data exchanged between the user and the gateway is encrypted with the shared secret, and the nym is used by the gateway to keep track of the data allowance consumed by the user. In addition, the user can use the nym as a temporary mailbox address where it can receive downstream data that is cached by the gateway.

(4) Route traffic via the mixnet. Until the data allowance of the bandwidth credential is consumed or expired, the user can send traffic to the Nym mixnet via the gateway. The mixnet routes user messages anonymously, at each step making inputs unlinkable to outputs. This conceals which user is accessing which service or conducting which transaction, as represented in Fig. 1 by the white and shaded background areas. The last mix node sends the message to the recipient's gateway, and the end recipient pulls their messages from it. Recipients may be service providers, validators, or other users.

(5) Use service credential to prove "right to use" a service. If the provider being accessed is a paid service that accepts Nym service credentials as proofs of payment, the user proceeds to use their service credential to gain access via the mixnet. For this, they execute the credential show protocol, where the user includes in the message to the provider proofs that the credential is valid, that it embeds the correct amount required to access the service, and that it has not yet been spent. The privacy features of Nym credentials include unlinkability between issued and used credentials, again corresponding in Fig. 1 to the white and shaded areas, making it impossible to identify the user from the show of the credential.

(6) Mark a received credential as used. The provider that receives the service credential submits to the blockchain a commitment to the serial number encoded in it. This commitment is made publicly available to prevent double-spending and to enable the provider to redeem rewards for the amount encoded in the credential.

(7) Distribute rewards from the nympool. At periodic intervals, validators algorithmically distribute rewards from the nympool. This includes rewards destined to nodes (mix nodes, gateways, and validators) for their work maintaining the core Nym network, as well as rewards to service providers that redeem service credentials received from users in exchange for services.

3.3 Token flow

In Nym, the mix nodes, gateways, and validators supply the features provided by the network, and they are rewarded for their honest work. To become a node operator, participants need to stake NYM token, part of which can be delegated by other stakeholders. In Nym, stake serves as a measure of node reputation and prevention of sybil attacks where adversaries cheaply introduce an unlimited number of nodes. Delegation allows stakeholders to vouch for a candidate node operator by supporting it with their stake. Nym nodes are incentivized via rewards to provide a high *quality of service* (QoS), which is measured in a fair and decentralized manner via a novel "proof of mixing" scheme. Details on staking, node selection, and QoS measurements are explained in Section 6.2.

While Nym network nodes contribute their work and resources to the functioning of the network, the beneficiaries are third-party service providers integrated with Nym and their end users. These participants need bandwidth credentials to send data through Nym, and the proceeds obtained from issuing bandwidth credentials are distributed to nodes as rewards in NYM token. In addition, Nym service credentials can encode redeemable proofs of payment that can be used to access paid services. In this case, a fee is taken on the deposit made by the user, and the remainder amount is embedded in the credential. Similarly to deposits made to obtain bandwidth credentials, the fee is used to reward the set of nodes in the Nym network. The redeemable amount of token embedded in the service credential is awarded to the service provider with

whom the user spent the credential. Thus, any user who enables the proper functioning of the Nym network by providing services as a mix node or validator is rewarded with NYM tokens for its work, namely the intensive but necessary calculation of mix nodes to route packets on behalf of other users confidentially or to sign the transactions that build blocks as a validator.

3.4 Threat model

Nym is designed to provide privacy from adversaries more powerful than those considered by VPNs, Tor, and existing peer-to-peer systems. The threat model considered by Nym is an adversary with *any combination* of the following capabilities:

- **Global network monitoring:** The adversary is able to monitor communications between *all* the participants and components of the network, and even all communications in the internet, globally. In addition, active network adversaries are able to inject, modify, and drop packets while they are in transit through the underlying internet infrastructure. This is in line with the capabilities of large nation-states and telecommunications providers that operate the internet infrastructure.
- **Corrupt participants:** The adversary can compromise a subset of Nym participants and coordinate adversarial resources to conduct attacks. Participants that may be malicious include validators, mix nodes, gateways, service providers, and end users. We assume the adversary has a limited budget (stake) to spend on the attack. This budget bounds the number of adversarial entities that the adversary is able to introduce in the network.

The adversarial objectives may involve attacking:

- **Privacy**: Learn private information protected by Nym, such as: sender of an anonymous broadcast message; identity of an anonymous user accessing a service via Nym; social graph of relationships between users who chat in a messaging app; linkage (clustering) of credentials or transactions as belonging to the same user.
- **Integrity and authenticity**: Violate service integrity guarantees by: impersonating other participants; forging or replaying credentials (free riding, double spending); biasing the placement of nodes in the mixnet topology; biasing network measurements; biasing the distribution of rewards.
- **Availability:** Disrupt the availability of the Nym network so that users can no longer rely on its protection to access services and instead are forced to communicate in ways that compromise their privacy.

The Nym network is designed using state-of-the-art security and privacy technologies to maximize resistance against attacks, considering that adversaries must be countered throughout the whole system, including the mixnet, measurement components, payment, and credential protocols. We expect the design to continuously evolve to include new features and counter ever more sophisticated attacks.

3.5 Privacy features

The Nym network is an infrastructure that provides privacy-enhanced access to applications. Here we introduce the privacy concepts that Nym is designed to implement. Note that Nym provides better privacy as the network scales up: a larger number of users and services being accessible through Nym makes its privacy protections stronger for *all* users of *all* applications compared to what each individual application could offer by itself to its user base.

Unlinkability and anonymity. In an abstract sense, **unlinkability** refers to the *inability to determine which pieces of data available at different parts of a system may or may not be related to each other* [87]. More concretely, consider user *identities* at the client side (e.g., IP addresses, public keys, device identifiers) and *messages*, *accesses*, or *transactions* at the service side. This is illustrated in Fig. 1 with the white and shaded backgrounds. Unlinkability between user identities and service accesses is equivalent to those accesses being **anonymous**. From the perspective of a service provider, it is not possible to identify *who* is the user conducting a specific transaction; while observing the client side does not leak *which services* users are accessing or *with whom* they are communicating. In some cases, sender and recipients want to identify each other while achieving **third-party anonymity**, meaning that they want to be sure that they are interacting with the intended party – while not wanting any other external party to be able to determine that they are communicating with each other.

Anonymity set. In *any* anonymity system, the anonymity provided to users and their transactions is always necessarily relative to an "anonymity set" [87]. The **anonymity set** is the set of subjects who, from the perspective of an adversary, may plausibly be associated with a message or transaction of interest. Transactions are more anonymous when this set is larger and more diverse. At the same time, transactions are less anonymous if a small number of individuals in the set are much more likely than the others to be the actual subject related to the transaction [38,94]. The importance of assembling large and diverse anonymity sets is what makes the Nym network uniquely equipped for providing high levels of privacy against powerful adversaries. Nym is an anonymous overlay network that can route communications for many applications and thus *blend* the user bases of all those applications *into one large anonymity set*. This level of privacy is not achievable by single-purpose, standalone applications and can only be provided by a generic, multi-purpose infrastructure such as Nym that aggregates user traffic for many applications.

Unobservability is an even stronger privacy feature than unlinkability and anonymity. With *anonymity*, when a user sends a message the adversary cannot identify *which* output message corresponds to the user – while *unobservability* means that the adversary cannot even determine *whether* the user is sending any message *at all*, or just being idle. Unobservability conceals the activity patterns of users and adds idle users to the anonymity set. Unobservability is achieved through the use of "cover" (or "dummy") traffic as explained in Section 4.6.

3.6 Requirements

In addition to providing privacy features in the form of unlinkability and anonymity, the Nym network is designed to meet the following requirements:

Generic. The Nym network is a general-purpose communication infrastructure that can be easily integrated with any networked applications that want to benefit from its privacy features.

Scalable. The Nym network scales to support a broad range of services with large user bases. Nym uses an architecture that can dynamically add more mix nodes to match increasing demand for routing privacy-enhanced traffic. Furthermore, the privacy provided by Nym increases as the network grows: integrating more applications leads to blending more users in the anonymity set.

Incentivized. To guarantee that high-value applications can rely on the Nym network for their communications, Nym incentivizes validators, gateways, and mix nodes via tokenized rewards to ensure professional maintenance, adequate performance, and high quality of service. Incentives are designed to ensure that honest cooperative behavior over long periods of time is in the best interest of all the network participants.

Decentralized. The Nym network has no trusted parties, centralized components, or single points of failure for *any* function, including: routing communications, public key discovery, issuing and verifying credentials, performing quality of service measurements, and distributing rewards.

Permissionless. Anyone can become a validator, gateway, or mix node in the Nym network, provided that they are able to contribute capacity to the network with an adequate quality of service. Participants are required to stake NYM token to become part of the Nym network, but are rewarded based on performing useful work.

Secure. The Nym network is designed using state-of-the-art security and privacy technologies to maximize its resistance to attacks, such that it can support users and services even in the face of powerful adversaries that monitor traffic in the entire internet and compromise a subset of Nym network participants.

4 Mixnet for Network-Level Privacy

A mixnet is a network of *mix nodes*, which are *overlay routers that transform and reorder messages*, such that their inputs cannot be correlated with their outputs, whether according to message appearance, timing, or other metadata [25]. The Nym mixnet is based on Loopix [89], with various modifications and extensions to meet the requirements outlined in Section 3.6. This section presents an overview of the Nym mixnet architecture, core mechanisms, design principles, and properties.

In a nutshell. The Nym mixnet is structured as layers of mix nodes, a design choice justified in Section 4.1. Nym uses a *source-routed* decryption mixnet, with users obtaining the node's public keys and contact information from the Nym blockchain. In source routing, the sender of a message chooses the route that the message will traverse before reaching its final destination. This choice is made according to the mixnet's public parameters and routing policy, which assigns mix nodes to mixnet layers and distributes traffic among the nodes of a layer weighted according to node capacity.

Section 4.2 introduces *gateways*, which are nodes that connect users and services to the mixnet. Gateways allow sending and receiving messages with improved reliability and accessibility, even if users are intermittently offline.

As described further in Section 4.3, Nym uses the Sphinx [29] packet format to encapsulate and anonymously route data payloads. Senders prepare Sphinx messages by encrypting them multiple times in reverse routing order. First, a message is encrypted for the recipient; then, for the last mix node in the path; then, for its predecessors in the path, ending with the outermost encryption, which corresponds to the mix node in the first layer. The complete Sphinx packet is finally encrypted with a key shared with the gateway, which forwards decrypted Sphinx packets to the mixnet.

When a mix node receives a Sphinx message, it strips a layer of encryption using its own private key material, retrieves information for routing the message to the next hop, and cryptographically transforms the message, rendering it unrecognizable. The mix node also retains the message a randomized amount time, as explained in Section 4.4, before forwarding it to the next node in the message's route. Responses to messages can be sent using "single use reply blocks" (SURBs), which are detailed in Section 4.5.

The Nym mixnet generates cover traffic "loops" as described in Section 4.6. Loops generated by mix nodes ensure a minimum level of anonymity at all times, while end users can generate loops to obfuscate the timing and volume of their active communication through Nym and thus achieve unobservability.

The mix node membership and the size and parameters of the mixnet are periodically updated on a per-epoch basis, as explained in Section 4.7. This allows the network to evolve and scale up to adequately serve all user demand as more services become accessible through Nym.

Finally, in Section 4.8, we discuss the trade-offs between latency, bandwidth, and privacy that must be considered when setting the parameters of the mixnet. We also explain the positive correlation between privacy and scalability: as usage increases, new mix nodes can be added to (linearly) increase capacity, and privacy can be maintained at high levels while allowing for lower latency and cover traffic overhead.

Comparison to onion routing. "Overlay proxy routers," "source routing," and "layered encryption" are also features of onion routing. Despite the similarities between mixnets and onion routing networks like Tor, there are also important differences between the two types of systems:

1. While onion routing creates circuits that carry all data packets associated to the circuit during a session, *mixnets route each message independently*, through a different route and without guaranteeing in-order delivery. This makes it significantly *harder to trace end-to-end data flows*, even for adversaries that monitor all connections in the network.
2. Onion routers forward packets along the circuit in a first-in-first-out order, without altering the timing characteristics of traffic. In contrast, mix nodes not only cryptographically transform but also *delay and reorder messages*, preventing attacks that link input to output messages based on timing information.
3. While onion routing systems do not typically employ any cover traffic, the Nym mixnet generates loops of *cover traffic* to guarantee sufficient levels of anonymity at all times. In addition, cover traffic provides unobservability properties to clients, such that it is *not even possible to determine whether users are actively communicating at all*.

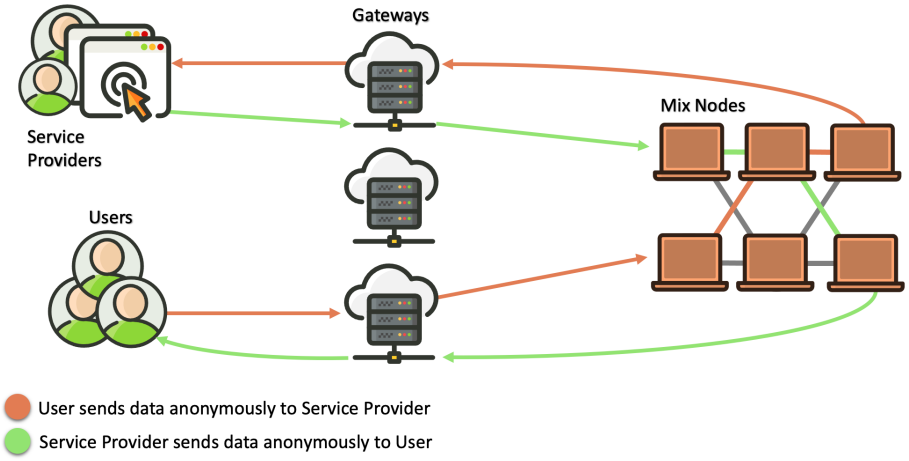

Fig. 2: Users and services communicating privately through gateways and mixnet with 3 layers of nodes

4.1 Layered mixnet topology

The Nym mixnet has a *layered* network topology, where mix nodes are arranged into layers as shown in Fig. 2. This topology has been shown optimal for privacy, scalability and ease of analysis [7, 36]. In particular, layered mixnets scale up to handle more traffic by simply adding mix nodes to existing layers, while privacy can be increased by adding layers.

Considering a sufficiently large number of candidate mix nodes, a subset is periodically selected to actively operate in the network. The selection probability is proportional to nodes' stake, as explained in Section 6.3, which acts as an indicator of node reputation. Non-selected nodes remain in reserve until the next period. Once membership for the next time period has been determined, the assignment of nodes to layers is *randomized* to prevent a malicious coalition of nodes from strategically optimizing their placement in the network to facilitate attacks. In line with avoiding reliance on centralized trusted parties, Nym's algorithm for assigning mix nodes to layers is *decentralized* and publicly *verifiable*. The algorithm utilizes a verifiable random function (VRF) [80] that is run before each epoch. Each mix node i has a public key pair (pk_i, sk_i) that is publicly available in their node descriptor. Before the start of an epoch, validators jointly publish a randomness beacon x in the Nym blockchain. Mix nodes compute the VRF output y_i and associated proof π_i that correspond to input x in combination with their own private key sk_i; i.e., $y_i, \pi_i \leftarrow \mathsf{VRF}(\mathsf{x}, \mathsf{sk_i})$. Considering three mixnet layers, the layer of a node for the epoch is computed as: $y_i \mod 3$, which is generalized to any number L of layers as: $y_i \mod L$. Mix nodes broadcast (y_i, π_i) to validators, which publish the values in the Nym blockchain. All participants can use π_i to verify that y_i is correct and reconstruct which nodes belong to which layer of the mixnet. Note that nodes' public keys are published in advance, when the participant signs up as a candidate operator in the network. It is not possible for nodes to predict at that time which public key will lead to a specific network position over the following epochs.

Layered routing implies that first-layer mix nodes only receive messages from gateways while nodes in subsequent layers only receive messages from nodes in the preceding layer. Similarly, gateways and mix nodes only relay messages to nodes in the succeeding layer. The final recipients may be validators, service providers, or any participants who run a Nym client in their device. In order to balance the traffic load in the network, the routing algorithm selects a node within each layer with probability weighted according to the node's capacity. Minimum and maximum capacity limits ensure that all mix nodes contribute meaningfully to the network operation while no mix node routes a disproportionately large fraction of traffic.

Layered routing and randomized allocation of mix nodes to layers make it difficult for adversaries who control a subset of nodes to compromise a significant fraction of messages. Considering a network with L layers, an adversary needs to control a node in *every* layer in order to fully deanonymize messages – if the adversary fails to populate even one layer, then no message will be traceable end-to-end. The best case for the adversary is when adversarial nodes are evenly distributed over all layers, as that maximizes the fraction of end-to-end routes that can be compromised. Even in that best adversarial case, a high fraction of malicious nodes will only compromise a small fraction of messages: considering L layers and a fraction α

of malicious node capacity per layer, the fraction of end-to-end traceable routes is given by α^L. Thus, in a network with $L = 3$ layers, an adversary that controls 50% of the mix nodes ($\alpha = 0.5$) would be able to observe 12.5% of messages *in the most favorable topology configuration*, while an adversary that controls 10% of nodes in each layer ($\alpha = 0.1$) would only fully observe 0.1% of messages. If the number of layers is increased to $L = 4$, then an adversary that controls 50% of nodes would in the best case observe 6% of messages and an adversary that controls 10% would just see 0.01%, i.e., one message out of every $10,000$.

4.2 Gateways

Gateways mediate access to the Nym network. They are proxies situated between the mixnet and the participants sending and receiving messages, as shown in Fig. 2. Participants must show a valid unspent *bandwidth credential* to a gateway of their choice, as explained in Section 5, to send messages through the Nym mixnet. Gateways protect the mixnet from free riders by collecting bandwidth credentials and limiting traffic to credential allowances. These credentials are also used by gateways to prove the amount of traffic they have serviced and to claim proportional rewards. Upon showing a valid bandwidth credential, the sender and the gateway establish a shared secret that is valid until the data allowance of the bandwidth credential is consumed or expired. Keys derived from the shared secret are used to symmetrically encrypt traffic between the sender and the gateway. Gateways do not perform Sphinx processing or mixing; consequently, they can forward messages to the mixnet quite efficiently.

Participants are free to use a single gateway for all their messages or split their traffic among multiple gateways. In the latter case, they use separate bandwidth credentials per gateway. Note that participants are not anonymous towards gateways from a network perspective, as gateways can see the IP addresses of the participants who connect to them. Yet gateways have no information about the destination of their messages or even whether the messages carry any real data at all. Gateways also receive and store downstream messages for their users, who can specify their temporary pseudonym as a return address for receiving messages. In this way, users are able to reliably receive messages even if they are not always online.

Gateways do not delay or reorder messages and therefore do not contribute to providing the unlinkability properties of the mixnet. Instead, gateways are in a position to facilitate access to the Nym network to participants all over the world by providing censorship-resistance capabilities. Users in certain locations may even need to disguise their use of the Nym network so that a local adversary can't even detect if the user is sending internet traffic via Nym at all. Gateways can use protocols that disguise the first "hop" into the Nym network as some other kind of traffic to make this possible.

Gateways must register a descriptor including their public keys, network addresses (IP, port), capacity, and proofs of having committed stake, like all Nym nodes. Note that the network address of a gateway is important towards first-layer mix nodes, who only accept messages from gateways that are part of the Nym network, and last-layer mix nodes, who only send messages to valid gateways. With respect to end users, gateways can offer many different user-facing network addresses and connection protocols advertised via different channels – as otherwise, it would be trivial to censor access to the Nym network by simply blacklisting the publicly available list of gateway IP addresses. Nym allows a diversity of gateway implementations, configurations, and censorship circumvention strategies in order to offer to end users many ways to connect to the Nym network without being detected or censored. The interface between users and gateways will build on concepts similar to those proposed for Tor's *pluggable transports* to enable access for users in locations subjected to high levels of censorship [11,81]. Gateways may also use VPN protocols, ranging from standards like OpenVPN to more recent protocols like Wireguard, to communicate to their users.

4.3 Sphinx packet format

Nym encodes application payload data using the Sphinx packet format [29]. Sphinx is a compact and efficient cryptographic packet format originally designed for providing bitwise unlinkability for multi-hop routing in mixnets. For payload sizes of tens of kilobytes the overhead introduced by Sphinx headers is around 1% – a few hundred extra bytes.

Sphinx allows for encrypting messages in layers and routing them through multiple hops, such that the message headers and payload are cryptographically unlinkable per hop. To create a Sphinx message, the client first derives a shared key with each mix node in the message's route using a randomly chosen group element and the public keys of the node. The client then wraps the message in multiple layers of

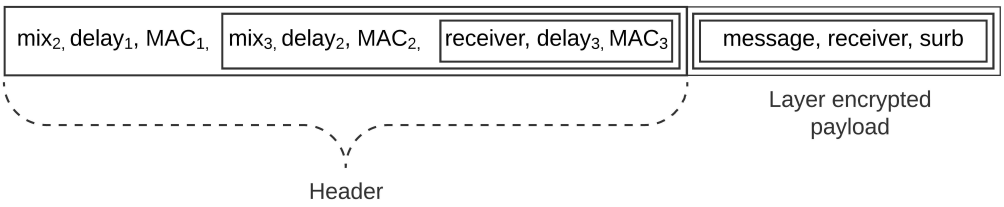

Fig. 3: Sphinx packet format

encryption using the derived shared keys, as shown in Fig. 3, adding message authentication codes and routing information for each node in the message's path. This enables header integrity verification per hop while preserving message unlinkability properties, even towards colluding adversarial mix nodes in the message route. Sphinx packets are padded to a constant length to avoid size-based attacks. While multiple packet sizes can be supported by the same mixnet, packets are only mixed with those of the same size.

Upon receiving a Sphinx message, a mix node combines the cyclic group element in the message header with its own private key. This enables the node to compute the shared keys necessary to decrypt the message and verify its integrity. The decrypted routing information per hop includes the amount of time the message should be delayed before forwarding as well as the address of the next hop to which it should be sent. The node blinds the group element in the message as part of the message processing, so that the output message is cryptographically unlinkable to the input. The transformed (blinded) group element will be used by the next node in the message's route to compute its own correct shared secret key and perform its own Sphinx message processing. Sphinx has built-in protections against active tagging attacks [88], enabling mix nodes to detect and discard messages that have been tampered with. Mix nodes also keep a list of the group elements of seen messages and discard repeated messages to prevent replay attacks.

4.4 Continuous-time mixes and exponential delays

Mix nodes get their name from the fact that they "mix" messages before sending them on. However, there is a variety of mixing strategies that can be implemented to reorder messages. Early mix node proposals implemented message reordering through batching and shuffling [16]. This simple design, which collects a number of input messages and outputs a random permutation of those messages, is known to suffer from some disadvantages: the end-to-end latency of messages in such mixnets is neither bounded nor predictable, and the bursty communication caused by periodically flushing batches of messages makes these mix designs inefficient at utilizing bandwidth. In terms of anonymity, simple batching strategies are known to offer low anonymity [35, 37] as well as being particularly vulnerable to attacks [26, 31, 95].

Similarly to Loopix [89], Nym uses continuous-time mix nodes [66]. Continuous-time mixing strategies, illustrated in Figure 4, delay each message independently, forwarding it to its next destination once a specified delay has timed out. The aggregate effect of independently delaying each message is an output sequence of messages that is randomly reordered with respect to the input sequence. Continuous-time mixes with exponentially chosen delays are known to offer optimal anonymity properties for a given mean end-to-end latency [24]. This is because of the memoryless properties of the exponential distribution. This property implies that if we observe two messages going into the mix node at times $t_0 < t_1$, and a message coming out at a later time t_2, the probability that the output is any of the two inputs is equal, regardless of differences in their arrival times t_0 and t_1.

Compared to batch mixes such as those used in Elixxir [19] XRD [70], and Vuvuzela [103], which produce finite anonymity sets, continuous-time mixes have larger – theoretically unbounded – anonymity sets. This is because the exponential distribution is long-tailed and even if most delays will be short, there is a non-zero probability that a message will incur a large delay; therefore, the adversary cannot discard future mix output messages as candidates for an input it wants to trace (in practice though, the infinitely long tail of the exponential distribution is truncated once the probability falls below a negligible threshold). Also, because these mix nodes send and receive messages continuously, they use bandwidth more effectively than batch mixes.

The amount of time a message dwells in a node is chosen by the original sender and encoded in the Sphinx header addressed to each mix node. This brings an important practical advantage of continuous-

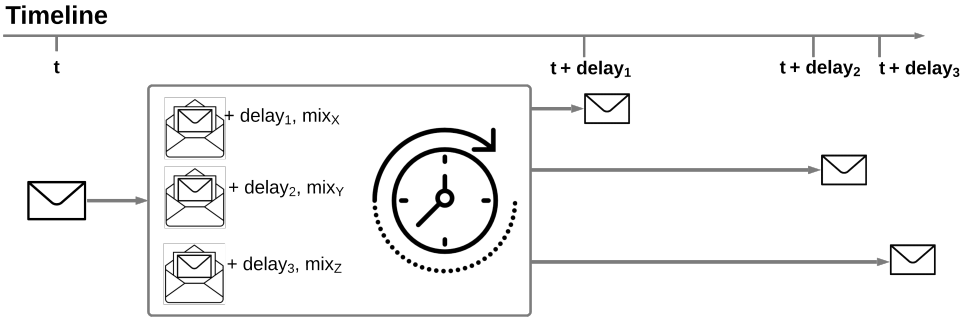

Fig. 4: Continuous time mix

time mixes compared to batch mixnets: given that the senders themselves generate and encode the delays for the message at each hop in the route, they can predict when their messages will be delivered. The mean parameter of the exponential distribution used by the sender determines the trade-off between expected end-to-end latency and the anonymity afforded to messages. If one holds the volume of traffic in the mixnet constant, configurations that incur in more latency automatically provide stronger anonymity.

We expect applications in the Nym ecosystem to have end-to-end latency tolerances ranging from one second (e.g., for instant messaging and Lightning Network payment channels) to minutes (e.g., for non-interactive transactions and file sharing apps). To offer a generic infrastructure that can accommodate a broad range of use cases, when routing traffic for an application, the Nym client takes into account its latency tolerance to set the mean of the distribution from where it draws the per-hop delays. Note that application-dependent variance of delay distributions trades flexibility for increased distinguishability of application traffic towards intermediary mix nodes, as certain (very big or very small) delays may be more likely associated to some applications rather than others. On the other hand, applications with large latency tolerance add protection to low-latency applications when considering network adversaries that monitor the inputs and outputs of all mix nodes but cannot see the per-hop delays encoded in messages [43]. In addition, Nym supports use cases with varying traffic loads, in contrast to systems such as XRD [70] and Vuvuzela [103], which assume that each user communicates exactly one message per round.

4.5 Anonymous replies and reliable transport with Single Use Reply Blocks (SURBs)

The lack of reliable transport has historically been an issue for mixnets – a mixnet's functionality is severely diminished if users cannot determine whether or not their messages are delivered. To address this long-standing issue and serve as a generic internet infrastructure for privacy, Nym takes advantage of the predictable end-to-end latency afforded by continuous mixes, in combination with a very useful feature of Sphinx: Single-Use-Reply-Blocks (SURBs). SURBs are pre-computed Sphinx packet headers encoding a mixnet route that ends in the participant that created the SURB. A sender can generate one or more SURBs and include them in their Sphinx message to a recipient. The recipient can use the SURBs as Sphinx headers to send back replies – or acknowledgements – that anonymously reach back the original sender after going through the mixnet.

SURBs are the Sphinx equivalent of "onion addresses" in Tor, with the caveat that a SURB can only be used *once* (to prevent replay attacks) and within its epoch of validity (the mix node public keys used to prepare the SURB are only valid for a limited period). SURB headers are encrypted by the sender, so the recipient sending it back cannot infer from it any information about the message route, the per-hop latency, or the sender's address, which is encoded in the innermost (last) routing layer of the SURB. Note that the identity of the first mix node in the path of the reply must be visible to the recipient, as they must know to which node to address the reply (tunneled via the recipient's own gateway). A SURB effectively contains: (1) the encrypted headers of a Sphinx message that, if sent to the mixnet, will be routed back to the original sender; (2) the address of the first-layer mix node where the message should be sent; and (3) a cryptographic key to encrypt the reply payload.

Upon receiving a message containing a SURB, the recipient can take a data payload, process it with the cryptographic keys provided along with the SURB, attach the payload to the headers received in the SURB, and send the message to the appropriate first layer mix node (through the receiver's gateway of choice). The

design of Sphinx is such that new (forward) messages are indistinguishable from SURB (reply) messages, even to the mix nodes that route the messages. In this way, replies and forward messages blend into one large anonymity set, reinforcing the overall anonymity properties of the mixnet.

Besides enabling anonymous replies, the Nym network uses SURBs to provide *reliable transport* even if there's packet loss in the network. This is achieved by including a SURB in Sphinx messages, in the layer addressed to the recipient's gateway. The gateway immediately sends the SURB back to the mixnet to let the anonymous sender know that the message has been received, optionally including flags in the payload, e.g., to signal errors such as "recipient unknown" or "recipient mailbox full." The SURB is addressed to the recipient's gateway rather than the end recipient because the recipient may not always be online, while the gateway is an always-online service that can immediately send back the acknowledgement when a message is received. The sender re-transmits a message if no acknowledgement is received within the expected time frame (the sum of per-hop latencies in the forward message and the SURB and an estimate of the propagation time between mix nodes for both directions).

Participants may also want to broadcast SURBs as a way of being publicly reachable without exposing their location, e.g., Lightning Network nodes that want to establish payment channels without disclosing their IP addresses, in order to protect wallets hosted in those servers. In summary, SURBs are a feature that makes the Nym mixnet a private communication infrastructure versatile enough to support a broad range of applications, far beyond electronic mail and voting – the two canonical use cases in the mixnet literature.

4.6 Unobservability via loops of cover traffic

Hiding "who messages whom" is a necessary mixnet property in terms of metadata protection – but it is not always sufficient to prevent surveillance. Adversaries that observe the volume and timing of sent and received messages *over time* may still be able to infer private information, even if individual messages are strongly anonymized [32]. Detailed data on the volume and timing of user interactions leaks information over time about which services the user accesses and the patterns of usage of such services. If the user engages in persistent behavior (e.g., always messaging the same friend or always accessing the same service), communication profiles may leak over time and be recoverable through long-term statistical disclosure attacks [26, 31]. Mixnets – including classical Chaumian batch mixnets – provide unlinkability, but do not provide unobservability [87]. In order for access to Nym to be unobservable, the adversary should not know when or how much actual traffic is being sent or received by a participant.

Cover traffic disguises real traffic patterns by adding "dummy" messages that carry no payload data and are simply discarded at their final destination. While routing a message, mix nodes cannot distinguish whether it is a dummy message or a normal message carrying user data. Routing dummy traffic to circle back to the sender rather than ending at a randomly chosen destination was originally proposed to proactively detect active attacks on mixnets [30]. Loopix, the name of which refers to its use of "loops" of dummy traffic, extends this approach to guarantee both a lower bound on anonymity and unobservability properties for end users [89]. Nym follows a similar approach, with participants generating dummy messages that travel in a loop and have themselves as final destination.

Loop messages are generated at random intervals that follow a Poisson process. They are generated by mix nodes to ensure that there is sufficient traffic in the network over time, and thus guarantee large anonymity sets and high levels of privacy. End users and services can optionally generate loop messages to obfuscate their activity patterns: when an adversary sees participants sending or receiving messages, it cannot distinguish whether they are actively communicating or just sending and receiving empty loops. Sending cover traffic in routes that loop back to the sender enables both users and mix nodes to maintain local updated knowledge on the health of the network, including the ability to detect when nodes go down, are congested, or under attack [30].

4.7 Mixnet epochs

Epochs are regular intervals of time at which the Nym mixnet is reconfigured. This periodic reconfiguration is necessary for multiple reasons. First, mix node decryption keys need to be updated to provide *forward security*, as mix nodes can otherwise be coerced to use their private keys to decrypt old replayed messages. An added advantage of updating keys every epoch is that nodes can forget the list of seen messages with each epoch change, as replayed messages from previous epochs can no longer be decrypted or routed because the necessary decryption keys are no longer valid or available. Note that per-epoch updates only

affect mix node *decryption keys*, and not long-lived signing keys used for authentication purposes. Second, periodically reconstituting the mixnet topology enables the system to recover from nodes that are down or have lost reputation as well as to re-balance layers and add new mix nodes to the network if more capacity is required. And finally, obtaining accurate measurements requires taking samples of a mix node's performance when placed at different mixnet layers, with different sets of predecessors and successors, so that even if some nodes are malicious they cannot consistently target a node to bias its measurements.

In terms of epoch duration, shorter epochs are advantageous for forward security, better sampling of node performance, and faster reactions to mix node churn and fluctuations of user demand. On the other hand, shorter epochs also incur more overhead, as node descriptors need to be collected and distributed to all participants more frequently and the protocols for assigning nodes to layers have to be run more often. Short epochs also have an adverse effect on anonymity. This is because the anonymity sets are disjoint per epoch: the fact that messages are encrypted under different keys and following a different network topology means that they cannot possibly be "confused" (i.e., *mixed*) across epochs. Although there will be dynamic optimization of the epoch duration, it is expected epochs will be in the order of hours or days. This is orders of magnitude larger than the end-to-end latency for messages and thus has a limited impact on the anonymity set: a message is not expected to remain in the mixnet for hours, and thus those future messages do not anyway contribute meaningfully to the anonymity set of messages that were input much earlier. Note that the network must make available keys and configuration information for at least one mixnet epoch in advance, in order for users to have the ability to prepare data packets and response headers that are valid for some time in the future. Careful engineering as well as consideration of corner cases is required to ensure that adversaries cannot exploit epoch handover conditions to diminish the anonymity of some messages.

4.8 Latency, bandwidth, scalability, and privacy

Anonymous communication systems must strike trade-offs between latency, bandwidth consumption, and privacy properties [33]. The continuous-time mix nodes used by Nym enable supporting applications with different latency constraints, as the trade-off between privacy and end-to-end latency can be tuned by changing the mean of the exponential distribution used to select the per-hop delays encoded in messages. Another trade-off is between the bandwidth consumed for cover traffic loops and the protection offered by this cover traffic, both in terms of anonymity and unobservability properties. Nym clients can be configured to generate more or less cover traffic by changing the Poisson parameter values, thus tuning the trade-offs between bandwidth consumption and privacy.

Scalability properties are key to enabling the large-scale adoption of any system; but this is even more crucial in the case of anonymity systems, as it is well known that *anonymity loves company* [41], meaning that the strength of anonymity properties is dependent on the number of users in the system. Therefore, systems for network-level anonymity should scale to millions of users in order to be truly effective. The layered mixnet topology used by Nym can scale by adding more nodes to the mixnet layers. This is in contrast to cascade mixnets, as used in Elixxir [16, 19], XRD [70], or Vuvuzela [103], in which new disjoint cascades need to be used once a cascade reaches its full capacity, with the consequent partitioning and size caps for anonymity sets. Instead, in layered networks, adding more mix nodes to increase capacity and handle more traffic load contributes to increasing the *overall* anonymity offered to all the users of the mixnet, creating a synergy between scalability and anonymity properties.

Furthermore, an increased number of users and messages allows lowering the end-to-end latency and the amount of cover traffic while still offering high levels of anonymity, which in turn enables the Nym mixnet to support applications with more stringent latency constraints. Serving a diverse set of applications also impacts the *quality* of the anonymity set by making the mere fact of using the network very uninformative for adversaries, who cannot infer much about which services the user is accessing, if any. By focusing on delivering a flexible, configurable private communication infrastructure that can scale up and support a broad range of applications, the Nym network is able to offer large and diverse anonymity sets to all of its users – something that no application integrated with Nym could achieve by itself.

Nym features *horizontal scalability*, as meeting increased demand simply requires adding more mix nodes. The more traffic that is sent from real users, the more privacy all users receive and the amount of cover traffic can even decrease while maintaining privacy.

5 Credentials for Privately Proving the 'Right to Use'

The protection of network metadata is necessary for implementing privacy-enhanced services – but it is not sufficient. In particular, mechanisms to determine whether users are entitled to services often rely on privacy invasive practices involving unnecessary user identification and tracking. As Edward Snowden put it, what is needed is privacy-preserving proofs of "right to use."

> *"We are gating access to the infrastructure necessary for life through this process of proving who you are rather than proving a "right to use" — that you paid for this, that you should be able to access this, that you have a blinded token of some type... It [shouldn't] matter who I am, I'm allowed to be here, I'm supporting the infrastructure, I've done my part, and that's all it should be. Otherwise we're being forced to give up ownership of our identities and our histories. We're allowing others to control the story of our private lives."*
>
> *Edward Snowden, Web3 Summit, 20 August 2019*

In many cases, access rights are granted to users upon payment for a service. Cryptocurrencies such as Zcash [92] or Monero [82] provide private payment functionalities, and if the determination of a user's *right to use* is done solely on the basis of a monetary payment, then these cryptocurrencies can be used to make that payment. However, managing users' *right to use* in services – what can also be referred to as *identity management* as rights may be dependent on users' identity attributes – is often more complex. For example, some services may require users to present proofs of age, country of residency, or KYC/AML (Know-Your-Customer / Anti-Money Laundering) status before granting them access. In other services, users may have the right to discounts because they are students, senior citizens, or premium members. Existing privacy-preserving cryptocurrencies such as Zcash and Monero lack the flexibility to support the management of identities and access rights in a broader sense beyond simple currency payments.

Nym credentials can encode any attributes that are relevant to proving *right to use*, including third-party certified identity attributes as well as proofs of payment for services. Nym instantiates **two concrete credential types** that attest to deposits to the nympool and are certified (signed) by Nym validators in a decentralized manner. Services can define additional credential types certified by other entities and use them jointly with validator-issued credentials.

Bandwidth credentials encode a proof of payment made to send data through the mixnet. The entire nympool deposit encoded in bandwidth credentials is later distributed as rewards to Nym network nodes (mixnet nodes, gateways, and validators) for their work providing Nym's privacy features. This is key to scalability, as increases in bandwidth demand fund the additional mixnet capacity needed. Note that service providers may want to obtain bandwidth credentials on behalf of their users and embed them in client software to facilitate usability and lower the barrier to adoption.

Service credentials encode a proof of deposit to the nympool made to use paid third-party services accessible via Nym. A small fee is taken from the deposit (to later be distributed as rewards to Nym network nodes). The user can show this credential to any service that is enabled to accept Nym service credentials in exchange for accessing paid services. A service provider receiving a credential can redeem it with the Nym validators, which will later award to that provider the equivalent amount of rewards. Instead of (or in addition to) payments, some services may require users to present proofs of attributes, such as their age certified by a third party, or self-certified proofs of knowledge of a secret (e.g., a password to their account). Services that require such proofs can extend Nym credentials with the desired attributes, defining new credential types. The flexibility for instantiating credentials with additional service-specific attributes and proofs of knowledge allows Nym to support services in implementing sophisticated yet privacy-preserving access policies.

Note that bandwidth credentials are mandatory for users who want to send data through the mixnet, while service credentials are entirely optional and up to the service provider, who may or may not charge for their services. Even in the case of services that charge users, they can do so using whatever means of payment they prefer. Nym's service credentials are just one of their options if they want to ensure that received payments preserve unlinkability, and thus that the privacy obtained through the mixnet is not undermined.

We introduce the general concept of anonymous credentials in Section 5.1 and describe the main characteristics of Nym credentials in Section 5.2. The Nym credential format is described in Section 5.3. Section 5.4 and Section 5.5 describe, respectively, two key features of Nym credentials: the ability to bind

multiple arbitrary attributes in an application-custom manner and the ability to perform split and merge operations with credentials that encode payment amounts. Section 5.6 discusses credential freshness, updates to the validator set, and credential redemption.

5.1 Anonymous credentials

Anonymous credentials allow users to share information privately without linking their transactions to their identity or leaking personally-identifiable information on a blockchain. For example, when a person shows the results of a COVID-19 test result in order to travel, they may also leak personal or medical information that are irrelevant to the task at hand and could even be used against them. In contrast, an anonymous credential scheme would allow a person to display the fact they had a negative COVID-19 test result within a certain amount of time, without revealing their name, date of birth, or even serial number or exact timing of the test – making it much harder to link medical and travel files. This same technology allows users to share information with the Nym network without violating the privacy the user has gained using the mixnet.

Nym credentials are realized by a cryptographic scheme called "anonymous credentials" [14], also known as "private credentials" [13], "attribute-based credentials" [12], and "minimal disclosure tokens" [9], a cryptographic solution for identity management that reconciles user privacy with service integrity. Like mixnets, the earliest proposals date back to Chaum's work in the mid-80s [18] and are closely related to untraceable payment schemes based on blind signatures [17]. The central insight still holds: payments made with bearer instruments are effectively a concrete case of proving ownership of a credential (certified by a third party as representing an amount of credit). Beyond encoding amounts of currency, credential systems can encode any attributes that may be relevant to the access and authorization decisions of arbitrary applications. In this manner, anonymous credentials provide a replacement for centralized and privacy-invasive solutions such as the OAuth standard used by Facebook Connect [52].

The basic model in anonymous credential cryptosystems considers three parties: a *credential issuer*, a *credential holder*, and a *credential verifier*. The credential holder obtains credentials from the issuer, who certifies attributes of the holder such as date of birth, address, money deposit, or KYC/AML status. The holder can later show the credentials to a verifier, who is able to check that the issuer has indeed certified the claims of the holder. The main privacy features of anonymous credential systems are *selective disclosure* and *unlinkability*.

Anonymous credentials allow the **selective disclosure** of attributes, meaning that credential holders can select what to reveal to a verifier out of all the attributes encoded in the credential. In the simplest form, this means that holders can reveal a subset (rather than all) of the attributes in the credential. However, anonymous credentials allow for far more sophisticated forms of selective disclosure, enabling holders to prove in zero-knowledge arithmetic and logical statements about the credential attributes.

In anonymous credentials, **unlinkability** means that it is cryptographically impossible to link the operations made with the same credential (i.e., the issuing of the credential and one or multiple showings of it). Therefore, based on the protocol transcripts of their interactions with the holder, even if the credential issuer and verifier collude, they are not able to determine which of the holders who obtained a credential from the issuer was the one who showed it to the verifier, or whether two credential shows were made by the same holder. Note that an anonymity set of credential holders is still needed – if only one person in the world obtained a credential from the issuer, it will be trivial to identify that it must be same person when the credential is "anonymously" shown to the verifier. An underlying anonymous communication channel (such as a mixnet) is also *necessary* to maintain the anonymity provided by cryptographic credential protocols – otherwise, users' IP addresses can be exploited to link credential issue and show operations and deanonymize credential holders.

A major limitation of most anonymous credential systems is that they are designed considering a *centralized* credential issuer. While this may be adequate for use cases where the issuer is naturally a centralized organization – for example, the issuance of passports by states and of membership cards by organizations – reliance on a centralized issuer is incompatible with open decentralized networks such as the Nym network. The next section introduces Nym credentials, which have decentralized credential issuance features that address this issue.

5.2 Nym credentials

Nym credentials extend the Coconut [101] anonymous credential cryptosystem, which supports decentralized credential issuance and is thus resistant to adversaries that compromise or take down a subset of issuers.

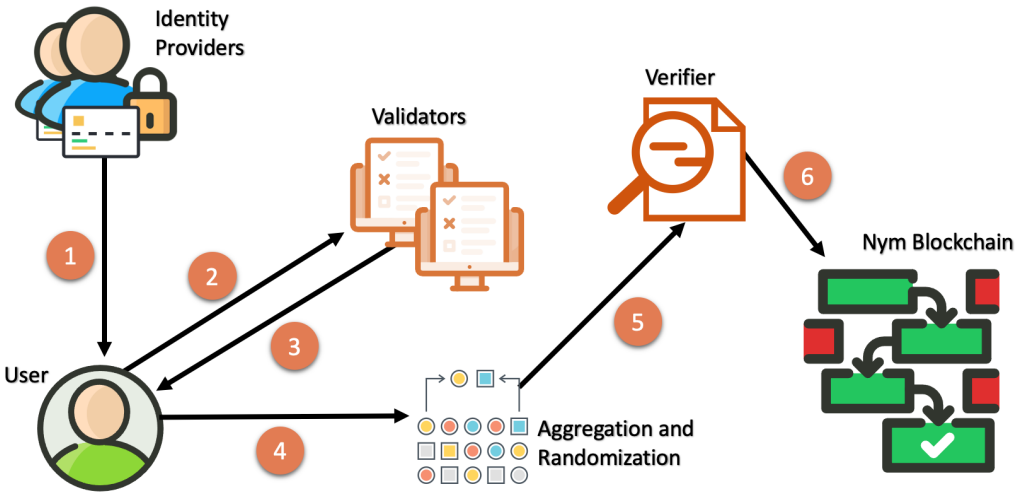

Fig. 5: Steps for obtaining and showing Nym credentials

With respect to privacy, malicious issuers cannot learn any information about private attributes encoded in the credential beyond what is explicitly proven by the user in zero-knowledge. Even in collusion with malicious verifiers, issuers cannot deanonymize users by cryptographically linking credential issuance and verification transcripts. Furthermore, the cryptosystem guarantees unforgeability of credentials, correctness of attributes and statements proven in zero-knowledge, and protections against double spending attacks.

In Nym, the set of validators functions as decentralized issuers for credentials that encode proofs of payment. Given that validators jointly control the Nym blockchain and the *nympool* where token deposits are made, they are in an ideal position to attest that the bearer of the credential made a token deposit. On the other hand, credentials containing certified user attributes are issued by third-party *identity providers* that can vouch for the authenticity of the attribute values. Credentials containing self-certified attributes can be issued by users themselves. Any participants who wish to use the Nym network, and notably *end users*, are in a position to obtain and show credentials. Similarly, multiple entities may act as credential verifiers, including *gateways*, *validators*, and *service providers* reachable through Nym.

As an initial step to self-organize as distributed credential issuers, the Nym validators run a key distribution and setup phase to agree on public parameters, such as the threshold t of validators that are sufficient to successfully issue a credential. Each validator i has a public key vk_i and the overall public key vk for verifying validator-issued credentials can be reconstructed given any subset of t public validator keys vk_i. After the setup phase, validators do not need to synchronize or further coordinate to issue credentials.

Once the setup is completed, the steps for obtaining and using Nym credentials, illustrated in Fig. 5, are as follows:

1. The user encodes each desired attribute in a credential. As described in Section 5.4, Nym provides features for binding credentials together when users are required to provide complex proofs involving multiple attributes. Depending on the use case, attributes may need third-party certification from Identity Providers, e.g., KYC/AML status certified by a bank, date of birth certified by a public authority, membership in an organization certified by the organization representatives, or discount coupon emitted by a seller. In other cases, attributes may be self-certified by the user, e.g., if they encode user preferences or knowledge of a password. Note that this first step is optional, as some verifiers do not require any attributes beyond proofs of deposit.
2. To obtain a valid bandwidth or service credential that proves that a user made a deposit in the nympool, the user first generates an unsigned credential representing the deposited amount and a zero knowledge proof of correctness satisfied by the credential. The user then sends a request to validators containing the unsigned credential and the zero knowledge proof.
3. If all is correct when validators check the deposit and verify the proof, they respond with a partial signature σ_i on the credential. Partial signatures are obtained for a threshold t number of validators.
4. The aggregation of t of those shares by the user yields a valid credential σ. Importantly, the user re-randomizes the credential to make its usage *unlinkable* to all prior interactions involving the credential.

5. The user *shows* the credential to a verifier to prove their "right to use" a service. The credential show protocol involves zero-knowledge proofs that prove that the user holds credentials that satisfy the access requirements of the service, both in terms of payment as well as any required authentication attributes.
6. To prevent double-spending attacks, before accepting a proof of payment, the verifier checks in the blockchain that the credential has not yet been marked as spent. Upon acceptance, the verifier publishes in the blockchain a commitment to the serial number of the credential to prevent it from being double spent with another verifier.

5.3 Credential format

Nym modifies the original Coconut credentials [101] into a modular format that allows applications to define and issue their own service-specific attributes, and have them interoperate with Nym-issued proofs of deposit. In Coconut, the size of the verification key is dependent on the total number of possible attributes that can be included in a credential, which removes flexibility to add new attributes on demand by applications. By contrast, in Nym, each credential contains exactly one attribute and it is possible to link together multiple credentials, each embedding different attributes, via the same key material. In this way, Nym credentials offer more versatility and efficiency for proving arbitrary combinations of attributes without forcing validators to set a longer key for every new type of attribute. A Nym credential contains information for a single attribute with the following five fields, of which the first two are encoded in clear and the last three are encrypted and only available via proofs of knowledge:

- (***public***) **Epoch of issuance**: $[epc]$ Time interval when the credential was issued. It determines the public key of the credential issuers, which is required to verify the credential.
- (***public***) **Attribute type** $[type]$: Identifier of the type of credential; e.g., bandwidth credential, service credential with a proof of payment, date of birth certificate, etc.
- (*private*) **Attribute value** $[val]$: Value of the attribute embedded in the credential; e.g., "100 MB" of data in a bandwidth credential, "50 NYM" of deposited token for a service credential, or "1975-11-20" for a date of birth.
- (*private*) **Secret key** $[k]$: Secret key of the credential, randomly generated by the user and only known to the user. This key is needed to prove credential ownership and to bundle credentials together as belonging to the same user.
- (*private*) **Serial number** $[s]$: Random number generated by the user and only known to the user. This serial number is used to prevent double-spending of credentials that involve proofs of payment. The field is not required in credentials meant to be shown unlimited times in an unlinkable manner, e.g., a proof of age.

5.4 Binding multiple credentials together

Nym is an open network where service providers can define their own attribute types and request users to include them in Nym credentials for authorization purposes that are independent of the Nym network. Arbitrary attributes required by service providers are not issued or certified by Nym validators. Depending on the use case, these attributes may be signed by third party identity providers or by the service provider itself, or they can be self-certified by users. The binding of credentials containing attributes certified by different issuers is done via the secret credential field k. Users must set the same key k in the credentials they want to bind together. They can then prove (in zero-knowledge) that they own a set of credentials that are all bound by the same k.

To illustrate, consider the case of a service provider that gives a discount to users that participated in a promotional event. The service requires proof of payment and proof of participation in the event, but does not need to know the name of the user or any other personal information that may have been provided by the user at the event. To do this, the user's client locally generates a custom Nym credential with the custom attribute type "promotionXYZ", that has a randomly generated serial number s. The credential secret key k is set to match a secret long-term pseudonym that binds a user's non-transferable credentials. The user gets their credential signed upon participation in the event, according to the criteria of the promotion organizers.

At a later point in time, the user makes a token deposit to the nympool and obtains a service credential from the validators for the deposit value minus any applicable fees. If needed, the user performs a split operation (§5.5) to obtain a service credential that embeds the (exact) discounted promotional rate. Importantly,

the secret key k of the service credential is chosen to match the key in the custom promotional credential. At the time of paying with a discount, the user proves: (1) that they own a valid unspent "promotionXYZ" credential, (2) that they own a valid unspent service credential for the exact discounted rate, and (3) that both credentials contain the same secret key k and thus belong to the same user, while keeping the identity of the user unknown.

In addition to enabling credential holders to prove that they fulfill authorization conditions that involve multiple attributes, the credential's secret key k can function as a hidden long-term pseudonym that prevents credential-pooling and disincentivizes credential-sharing – while preserving the anonymity of the credential owner.

5.5 Split and merge operations

As argued above, privacy is a holistic property of a system; information leakages in one component may be enough to break unlinkability and de-anonymize users. Nym credentials rely on cryptographic techniques that guarantee unlinkability at the protocol level. In the case of service credentials that encode proofs of payment, however, payment amounts may act as a side channel that undermines privacy. For example, if each service charges a different amount, it would be possible to infer which service a user intends to use from the amount of their deposit in the nympool.

Nym credentials extend Coconut with features that can address this problem: users can perform shielded *split* and *merge* operations with the amounts encoded in their unused credentials. These operations take place in the encrypted domain, so the amounts involved are protected. This allows users to, for example, make various deposits of a default amount over a period of time, then aggregate them in zero-knowledge with a *merge* operation into one large credential used to pay for an expensive service. Users can also deposit a default lump sum of token in the nympool. They then use *split* to get two credentials: one to pay the exact amount charged by the service and another containing the change that they keep for later use. In this way, they can pay for subscriptions to multiple services, or cover recurring payments that a service charges them over time, until they consume the original lump sum. Finally, users can also redeem an unused credential themselves if they no longer wish to use it to access services.

Note that it is not possible to distinguish the case in which a participant is recovering (part of) an unused credential that they themselves acquired for a deposit that they didn't spend with others, from the case in which the participant received the credential from someone else in exchange for a service they provided to others. This ensures that all users who acquire credentials are blended into the same anonymity set, within the bounds of their total deposits. While the diversity of applications necessarily entails a diversity of pricing amounts, deposits should have values taken from a set of fixed amounts (e.g., powers of 10) and then split and merge operations are performed as needed to use services or refund the change after a payment. We stress that split and merge operations are conducted in the encrypted domain, meaning that it is not possible to see the amounts being split or merged. Given a set of deposits and redeemed credentials, matching them based on amounts becomes infeasible once users engage in frequent split and merge operations in between deposits and payments.

The split and merge functionalities are provided by Nym validators. A user who wishes to *merge* two (or more) credentials generates a new credential σ_3 with a value that is the sum of the two (or more) credentials σ_1 and σ_2 to be merged. The user sends σ_3 to the validators together with proofs attesting that:

- The value val_3 in σ_3 is such that: $val_3 = val_1 + val_2$.
- The serial numbers s_1 and s_2 (of σ_1 and σ_2, respectively) are unspent.

If everything is correct, the validators mark s_1 and s_2 as spent and issue partial signatures $\sigma_{3,i}$ on the new merged credential σ_3. As in the previous cases, the user aggregates t partial signatures to obtain σ_3, and randomizes the voucher to make subsequent transactions unlinkable.

Similarly, a user may want to *split* a credential σ_1 into two (or more) smaller credentials σ_2 and σ_3. A common use case for this situation is a user who has a credential for an amount larger than what is needed for a service, who wants to split it in two: one for the immediate payment and another one to keep as change. To split the large credential σ_1, the user generates the new credentials σ_2 and σ_3 and sends them to the validators along with proofs that:

- The values val_2 and val_3 are such that $val_2 \geq 0$, $val_3 \geq 0$, and $val_1 = val_2 + val_3$.
- The serial number s_1 is unspent.

Upon successful verification, validators mark s_1 as spent and they issue partial signatures $\sigma_{2,i}$ and $\sigma_{3,i}$. Note that it is possible for users to arbitrarily combine split and merge operations and submit a set of unspent input credentials $\{\sigma_{in}\}$ to be merged and split into a set of output credentials $\{\sigma_{out}\}$, such that (1) all values in $\{\sigma_{out}\}$ are non-negative and (2) the sum of values of the credentials in $\{\sigma_{in}\}$ equals the sum of values in $\{\sigma_{out}\}$.

5.6 Credential Intervals

When the set of Nym validators is updated, the threshold key used to sign credentials is also updated, and so is the public verification key that verifiers use to check validator-issued credentials. Validator nodes are expected to be very stable, given that uptime and quality of service increase a node's reputation and likelihood of selection for the validator set. In addition, the threshold nature of all validator protocols means that the the validator set can continue to function undisturbed even if multiple validators fail or are attacked simultaneously. Given this, the intervals of validator set updates can be longer than the mixnet epochs. We expect them to be between a week and a month.

When updating the validator set, the version of a credential's public verification key again constitutes a potential side channel. If credentials are distinguishable based on their verification key, which is tied to their time of issuance, then credential anonymity sets are partitioned every time the validator set is updated. This partitioning in combination with online activity timings could deanonymize some users. For example, consider a user who obtains a credential and then immediately goes offline and stays so for a long time. By the time the user reconnects, all the other credentials issued at the time the user got her credential have already been used. When she shows her credential, it may be possible to infer the identity of the "anonymous" user by seeing that she is presenting a rare old credential.

To prevent this, Nym requires credentials to be up-to-date in order to be acceptable by verifiers, who only accept credentials whose verification key corresponds to the current set of validators. Users who own unused outdated credentials need to *refresh* them with validators before they are able to use them with a verifier. In this way, all issued credentials are aggregated in a large anonymity set that blends all users over time, rather than partitioning the users in small disjoint anonymity sets that in corner cases could lead to user identification.

To refresh an outdated credential σ_0, the user (or any participant who owns a credential) effectively "burns" σ_0 with the current set of validators in exchange for an equivalent updated credential σ_1. To do this, the user generates a new credential with the same $type$, value val, and key k as the old credential. The new credential updates the period of issuance epc to the current one and generates a fresh random serial number s_1. Users send the newly generated credential to validators together with proofs that value $val_0 = val_1$ and $k_0 = k_1$. Validators check that the serial number s_0 has not yet been spent, as well as verifying the correctness of the provided credential and proofs. If σ_0 is unspent and σ_1 is correct, validators mark s_0 as spent and issue signed credential shares $\sigma_{1,i}$. Users again aggregate t of those shares to reconstruct their updated credential σ_1.

Service credentials issued by validators encode a redeemable token deposit. When a user shows a service credential to a verifier, the verifier marks the serial number of the credential as spent. The verifier can then redeem the credential with the validators by proving that they signed a commitment to the serial number of the credential they are redeeming. Verifiers can also batch together a set of credentials to conceal the individual amounts received per credential, instead proving the correctness of the claimed total. The validators check the validity of all proofs and transfer rewards from the nympool to the appropriate recipients. This may be executed at periodic *reward intervals* to increase timing unlinkability between nympool deposits and payouts of rewards to service providers. For simplicity, reward intervals may be the same as credential intervals, though different interval lengths can be chosen if there are compelling reasons to do so.

Bandwidth credentials encode an allowance to send packets through the mixnet during the credential interval. Shown bandwidth credentials are collected by gateways and submitted periodically to validators to claim rewards at the end of the interval. Gateways are rewarded similarly to mix nodes, and the reward share of a gateway is proportional to the amount of bandwidth allowance it collected from its users via bandwidth credentials – as a fraction of the total amount collected by all gateways for the full interval. Note that nympool deposits made in exchange for bandwidth credentials are shared among *all* Nym node operators (mix nodes, gateways, and validators) in the form of rewards; that is to say, validators do not directly "refund" to gateways the value encoded in bandwidth credentials in the way they redeem service credentials claimed by a service provider.

6 Cryptoeconomics

Scalability and sustainability are necessary features for the Nym network to grow to be successful. **Scalability** refers to the ability to support a broad range of applications with large user bases. The Nym network must provide reliable, high-performance, high-quality privacy for services and remain secure and operative even in the face of the byzantine failure of some of its components. Reliably provisioning privacy services at scale requires resources, both in terms of bandwidth and computation. **Sustainability** ensures that Nym node operators can pay for their investment in running nodes and profit from their labour via rewards. In order to grow, the network needs to add more capacity to serve more users and route more data privately. Nym leverages cryptoeconomics – *cryptonymics* in our case! – so that demand by service providers and their users funds the infrastructure needed for scaling the network in a sustainable manner [2].

The problem of giving mix nodes an accurate **reputation** score that serves to make decisions regarding which nodes to include in the network predates the development of both Tor and Bitcoin [40]. In the context of mixnets, the core challenge is: how can we prove that mix nodes are contributing to routing by correctly following the protocols, and give them a reputation based on how well they provision privacy over time? Reputation systems were theorized for mixnets but did not have an efficient and scalable way to prove if mix nodes were reliable or not, and so failed to be implemented by programmers intent on shipping out usable software [44]. Although there has been work on verifiable shuffle proofs for re-encryption mixnets used in e-voting, these approaches apply only to batch mixnets that do not scale dynamically and the zero knowledge proofs are too computationally expensive to generate for large amounts of packets [4]. The concept of paying for mixnet access has long been theorized [47], and there were even attempts at the turn of the millennium by startups such as Zero Knowledge Systems to build commercial mixing services [48], but these earlier attempts did not succeed in part due to a lack of privacy-enhanced digital payment system (as payment to a central company for anonymity was problematic). Previous attempts to incentivize Tor nodes have often sought to have users reward the nodes in some sort of "hop by hop" basis [60], but these kinds of payment systems generally harm privacy and lead to attacks [64, 91], so the Tor network was built without any kind of payment infrastructure to sustain the node operators.

Building on insights from Bitcoin, the Nym network provides privacy as a common good while incentivizing returns and sustainability for the node operators that provision this critical infrastructure. Inspired by Bitcoin transaction fees, deposits in NYM are required to obtain bandwidth credentials and communicate privately through the Nym mixnet. To that end, the NYM token's value reflects the demand for privacy given the available mix nodes that can supply this privacy by routing packets. On the demand side, as the demand for privacy-enhanced network access increases, more NYM then can be distributed to Nym nodes. More nodes can then join the Nym network to obtain the increasing rewards. The goal of this mechanism design allows the Nym network to dynamically scale the supply of quality nodes to meet the demand for privacy.

On the supply side, the Nym network tokenizes the reputation of nodes over time. The reputation of a node in NYM will ultimately be tied to a prediction of their reliable provisioning of network capacity in the future; it is represented by the aggregate of the node's own pledge and the token delegated by other stakeholders. Anyone who wants to run a node themselves must first **pledge** some amount of NYM. This pledge acts as seed reputation (it is the "skin in the game" that the node operator invests to join and operate, that could even be slashed in extreme cases of misbehavior), and it is a sybil prevention measure, as creating many node identities becomes expensive. In addition to node pledges, all NYM token holders can **delegate** to candidate nodes run by others as a way to increase the reputation of the candidate node, which in turn determines the node's chances of being selected as part of the network in the next interval. When the candidate node becomes active in the network and meets operational requirements, it receives rewards that are shared with the token delegators that supported the node. Nodes regularly have their quality of service measured using the *proof of mixing* protocol described in Section 6.4, and measured node performance has a direct impact on the amount of rewards earned by the node. Nodes that suffer from frequent faults fail to gain rewards. This disincentivizes token holders from delegating to those nodes, decreasing their reputation and making them be selected less frequently. Eventually, underperforming nodes are excluded, ensuring the continued reliability of the Nym network.

Although Nym is not a "proof of stake" network, we use the term "stake" as a proxy for "reputation," as Nym nodes are rewarded proportionally to their *work* and *not* their *stake*. Rather than undertaking "proof of work" through hash puzzles like Bitcoin, Nym operators perform resource-intensive work required by the Nym network to collaboratively provide a privacy infrastructure for services. Mix nodes are rewarded

for mixing rather than mining. Validators do work by signing data blocks and credential shares. Gateways validate bandwidth credentials and route traffic to and from the mixnet. Stake is used to signal node reputation and ensure that node operators are incentivized to provide a high quality of service. Nodes with a good reputation are chosen more frequently to perform work and obtain rewards.

The Nym network earns income from fees for the usage of the network. This includes the purchase of bandwidth credentials for sending data via the mixnet and a fee on deposits converted into service credentials (note that the service credential value is redeemable by the receiver of the credential). These nympool deposits are used as rewards for redistribution to nodes active during the reward interval, with a small percentage being kept in reserve to ensure long-term network stability. Similar to Bitcoin mining rewards, node operators also receive rewards from **mixmining rewards**. Mixmining rewards come from a pool that is initialized at the genesis of the network with a large number of NYM token, which are released over time to reward node operators in a deflationary manner. The expectation is that mixmining funds will initially be the main source of rewards and that over time, as the mixmining rewards decline and more applications and users join the Nym network, fees will overtake mixmining as the primary source of income for node operators.

In this section, we outline the cryptoeconomics used to self-organize the Nym network. First, we describe the role of the NYM token in Section 6.1. Next, in Section 6.2 we explain how mixmining rewards incentivize the bootstrapping of the Nym network while the reserve ensures long-term sustainability. Then we look at the life-cycle of the Nym mix network in terms of an epoch. We first show how nodes may join the Nym network in Section 6.3, and then how the NYM token facilitates an auction for privacy-enhanced bandwidth that leads to mix nodes being selected and then assigned to the network in Section 6.3. The nodes in the network have their quality of service measured via the proof of mixing protocol given in Section 6.4, and gain rewards as per Section 6.5. At the end of the epoch, the Nym network dynamically re-organizes to match demand.

6.1 The NYM Token

The NYM token is required by all actors to participate in the network, which will be fully operational at launch. In a nutshell, NYM rewards participation, helps prevent network abuse, and incentivizes the reliability and availability of the network as it scales up to meet growing demand.

Stakeholders are all participants holding NYM token, which translates to reputation used to select which nodes actively contribute to the network. Following their preference and aptitude, some NYM stakeholders volunteer to operate mix nodes by **pledging** NYM token as their initial deposit while others review node statistics and characteristics to decide whether to support a node by **delegating** their stake to it, increasing its reputation and chance of being selected. Thus "staking" NYM refers to these actions: (1) setting up a node and pledging NYM, or (2) reviewing node characteristics and delegating NYM to an existing node. Nodes are then selected to operate in the network with likelihood proportional to their total stake (pledged and delegated).

The NYM token is needed by users to privately communicate and access services via the Nym network. Yet exactly how payment is made to services and the Nym network itself by users depends on the service provider. For example, a service provider may sponsor bandwidth in bulk for all its users so they can easily access the service via the Nym network. Alternatively, users may pay an additional "privacy fee" similar to a Bitcoin transaction fee on an individual basis. While we expect some users to pay for access to the Nym network in cryptocurrencies like Bitcoin, users of Nym may continue to use traditional fiat relationships with their provider. Regardless of how payments are made, fees taken from deposits are converted to NYM in order to measure the demand for privacy and sustain the Nym network via rewards. Fees are collected in the nympool over the **reward interval** and then distributed in the form of rewards at the end of the interval, as explained in Section 6.5.

The NYM token is a cryptoasset necessary for the functioning of the Nym network. The token represents the economic value the Nym network provides and provides a reward for nodes that provision the privacy generated by the Nym network. We do not envision NYM being used as a cryptocurrency outside the Nym network; its primary usage in aggregate is to balance the supply and demand for privacy given by the Nym network. Nym users may continue to use fiat to pay for services; some advanced users will prefer to pay for services using cryptocurrencies that are already private. We do not wish to constrain how services may receive payment and any conditions they may have on payment, as it benefits the privacy of all to maximize the number of services and users on the Nym network. Again, Nym service credentials can be used as a

"proof of payment" in Bitcoin, fiat, or any other cryptocurrency accepted by services; they can contain any other additional information needed by service providers. Nym nodes may also wish to use their NYM tokens to obtain the resources necessary to maintain and increase their role in the network, and so Nym operators may chose to receive payment in Bitcoin or even another (possibly private) cryptocurrency rather than hold their reward in NYM token. The main function of the NYM token is not to be used, now or in the future, as a means of payment for acquiring goods or services, as well as a means of money or value transfer. The NYM token can be a means of payment to pay transactions fees, but only within the Nym network. Via using Bitcoin as a cryptocurrency for means of payment, Nym can support Bitcoin while using advanced functionality such as Liquid [3]. This design also allows the Nym mixnet to be agnostic to any underlying blockchain and support the maximum number of services in order to maximize the privacy of all users.

6.2 Mixmining Rewards and Reserve

The rewards periodically available for distribution to node operators are the sum of mixmining rewards and fees in the nympool paid to the network. The mixmining emission schedule is designed so that node operators surpass their break-even point at network launch and remain profitable throughout the lifetime of the network. We expect that the mixmining pool will be the main source of rewards the first few years of the network's lifetime, with income from fees taking over as the network attracts more usage in a similar fashion to Bitcoin. The precise cryptoeconomics of this reward schedule will be detailed in a separate document.

If the mixmining pool fully depletes, the Nym network would have no resources to recover from a temporary drop in demand (and corresponding disruption of income from fees). To prevent this, a small fraction of obtained fees are returned to a long-term *reserve*. This reserve can then be used to stabilize the node returns in case of a sudden drop in demand. The reserve is envisaged to be governed by the stakeholders. Note that the details and precise parameters of the mixmining pool, reserve, and token release schedule are in a separate cryptoeconomic document.

6.3 Joining the Nym mixnet: pricing bandwidth and node selection

The selection of nodes to operate in the Nym network during each time period relies on a market mechanism. As a preliminary matter, an estimate on the expected required capacity is needed to decide on the number of mix nodes that should be selected to compose the mixnet. In Nym, this is predicted based on the recent history of used bandwidth credentials as claimed by gateways, which provide an upper bound on the number of packets sent through the network per epoch.

Operators who wish to run a mix node submit a **declaration** that states their price per packet (PP) function as well as a profit margin (PM) to be applied on the node revenue. Validators on the other hand simply state a flat cost for maintaining their validator server and signing credentials and data blocks. Gateways are rewarded proportionally to the bandwidth credentials they collect from their users, as explained in Section 5.6. For mix nodes the PP function maps the number of routed packets to the amount of NYM that the node operator spends per packet. It also implicitly defines the capacity c of the node. $PP(0)$ expresses the bare operational costs without routing any packet, while $PP(x) = \infty$ when x exceeds the capacity c of the node.

The Nym network may impose a minimum capacity that each node must offer to be eligible and a maximum capacity accepted per node. If a node declares PP above the minimum capacity, it can be considered *registered* to the Nym network and may be chosen to operate in the network. If (as expected) more nodes register than are needed, the network chooses nodes with probability proportional to their staked (pledged plus delegated) NYM token, as that is a "signal" from the market of the nodes' reputation.

Given the mix node declarations, we can infer a median node function $PP_{\mathsf{med}}(w) = \mathsf{median}\{PP(w)\}$, where the median runs over all submitted declarations for a given per-epoch traffic load w. Now suppose that we want to price in advance units of work in terms of bandwidth, as required by the pricing of bandwidth credentials. If there was a single node with pricing declaration $PP()$ that would handle the total work, a specific scheme that can be used is the following. Suppose that the actual demand follows a probability distribution $\mathcal{D}$. We denote by W the random variable of work that follows the distribution $\mathcal{D}$. The price of the x-th unit of work is defined as follows.

$$PP(x) - PP(x-1) + PP(0) \cdot \frac{1}{\mathbb{E}[\mathsf{W}]}$$

Under the above pricing scheme, the expected revenue of the node is $\mathbb{E}[PP(\mathsf{W})]$, where W is the random variable of the work which is distributed according to $\mathcal{D}$.

A simple proof of the above fact can help clarify why the bandwidth credentials pricing is set adequately. Summing across all units of work sold, it holds that the revenue will be equal to:

$$\mathsf{W} \cdot PP(0)/\mathbb{E}[\mathcal{D}] + \sum_{x=1}^{\mathsf{W}} (PP(x) - PP(x-1)) = \mathsf{W} \cdot PP(0)/\mathbb{E}[\mathsf{W}] + PP(\mathsf{W}) - PP(0)$$

By linearity of expectation, we conclude the correctness of the statement, and, as a corollary, the fact that as long as the variance of $\mathcal{D}$ is small, the per unit pricing scheme breaks even. We remark that this is a simple linear scheme; adapting more sophisticated models [5, 67] is also possible.

As the above equation describes the pricing scheme for a single hypothetical mix node, for the entire Nym mixnet we use the modified formula below that takes into account the fact that the mixnet has L layers (hence the work is multiplied by L) while the load is balanced by node capacity within each layer. The median $PP_{\mathsf{med}}()$ packet price formula is used.

$$L \cdot (PP_{\mathsf{med}}(\lceil x/W \rceil) - PP_{\mathsf{med}}(\lceil x/W \rceil - 1) + PP_{\mathsf{med}}(0) \cdot \frac{1}{\mathbb{E}[\mathsf{W}]})$$

Based on all the above, we observe that a small variation between mix nodes, a reasonably concentrated demand function and a smooth $PP(\cdot)$ pricing function results in a situation where the total revenue generated from bandwidth credentials offsets the operational costs of the mixnet as a whole. In the case where there is large variation in demand, $\mathcal{D}$ has high variance, which would result in high variance in the node revenue, the mixmining rewards can help maintain the stability of pricing [59].

At each epoch, k nodes are registered and available to serve as mix nodes, where k is a parameter that is dynamically adjusted. Below, we describe the mechanism for populating the mixnet; the process for selecting and rewarding validators is similar. Some nodes are elected to be operational in the mixnet, while other nodes are held idling for the epoch forming the "reserve." If r denotes the number of reserve nodes, then the mixnet is populated with an average of $(k - r)/L$ active nodes per layer. The selection of nodes is made probabilistically, using randomized sampling without replacement, and with node selection weights proportional to their staked NYM token. Once the set of nodes for the next epoch is selected, nodes are randomly allocated to mixnet layers in order to prevent adversaries from strategically positioning compromised nodes.

The NYM tokens that are "staked" on a node are a measure of the node's reputation. Therefore, weighing by stake the selection of which mix nodes to include in the network topology results in a mixnet populated by nodes that have the reputation of offering a high quality of service. If a mix node has a large amount of delegated stake in NYM tokens, this suggests that the community stands behind the reputation of the node, as the node regularly provides rewards to its delegates due to being reliably online all the time and keeping a good performance factor. Moreover, this node selection process also optimizes the cost for users, favoring the selection mix nodes that are not only reliable but also economically efficient. A mix node that declares a PP function that is too expensive will distribute lower rewards to delegates, since the lower PP is, the higher rewards available for the node's delegates.

To attract delegates, new mix nodes must offer attractive parameters: either offering more competitive $PP(\cdot)$ and margin PM, which leaves more share of rewards for delegates, or introducing other attractive externalities, such as giving some proceeds to a good cause such as "free access" to Nym for human rights activists, which will resonate with some delegates. Recall that underperforming nodes receive diminished or no rewards, causing delegates to move away from backing them. This may occur voluntarily, because rewards fall, or forcibly if the mix node registration is revoked due to subpar performance. This creates opportunities for new mix nodes to attract delegates from existing nodes that have failed in some way. The goal is that, over time, the Nym network will converge on an optimal price for provisioning privacy for users.

6.4 Proof of mixing

The viability of Nym critically depends on nodes providing a consistently high quality of service (QoS) in terms of mix nodes correctly and reliably processing, routing, and delivering messages. The *performance*

factor PF denotes the fraction of messages correctly routed by the mix node. Nym distributes rewards to nodes that meet QoS criteria (a PF close to 1), raising the need for reliable measurements per node to estimate PF.

Obtaining accurate and reliable PF measurements for mix nodes in a decentralized manner is a challenge, particularly when we consider that a subset of adversarial participants may collude to actively and strategically attempt to bias measurements. Part of the challenge has to do with communications within the mixnet being visible only to the two entities involved in the traffic exchange (and to network adversaries), but not to others. Therefore, there is no collective visibility regarding whether mix nodes are reliably routing the traffic they receive, meaning that they do not drop or inadequately delay packets. Furthermore, in Nym, the measurement of node performance cannot rely on trusted measurement servers – the kind of solution adopted by Tor[2] – as that would introduce a centralized component (and single point of failure) with control over a highly consequential network function that determines who gets rewarded and how much.

Mix node operators self-declare their capacity when joining the Nym network through the PP function. On the other hand, their PF is measured on a continuous basis in a decentralized manner. Nodes that overstate their capacity in an attempt to claim a larger share of rewards are not able to sustain a high PF when used at their declared capacity. Far from increasing income, overstating capacity leads nodes to miss out on rewards and lose reputation.

In the e-voting literature, there exist mix node designs that produce *verifiable proofs of mixing* [1,4,50]. However, this verifiability comes at a significant cost. These mixes are slow, batch-based, require a costly final threshold decryption, and necessitate that mix nodes communicate not directly, but via a publicly verifiable broadcast channel (such as a blockchain) where *all* input and output messages are posted, along with zero-knowledge proofs of correct permutation between input and output message batches, for *each* mix node in the network, for *every* batch of messages. While these constraints and costs may be tolerable for voting use cases, they are impractical for our purposes, deployment context, and application requirements. Therefore, to accurately estimate the rate at which mix nodes are dropping packets (regardless of whether this is done maliciously or simply due to lack of resources), we resort to sampling their performance in a universally verifiable yet decentralized manner. This is done with the use of special Sphinx-encoded measurement messages that are indistinguishable from all other messages while they are in transit through the mixnet and are only revealed as measurement messages after the end of the epoch.

1. Generation of measurement messages. Measurement messages are generated by both mix nodes and end users, following circular routes identical to those of dummy loop messages. The reason why end users need to generate some of the measurement messages is to measure QoS end-to-end with traffic that is completely indistinguishable from other types of traffic, such as application messages, acknowledgments, and loops of cover traffic.

To ensure that participants cannot (perhaps maliciously) bias the timing and routing of measurement messages, these messages are generated pseudo-randomly using a Verifiable Random Function (VRF) [80] and a sampler. We consider that all participants in the measurement protocol have a public key pair (pk, sk) that is verifiably associated to their role in the Nym network. Nym nodes and publicly accessible service providers have their public keys distributed through the blockchain as part of the network's public key discovery. End users are required to own a valid bandwidth credential to generate measurement messages, and the VRF key used for generating measurement messages is bound to secret key material included in the credential.

In advance of each *epoch*, validators jointly publish a random beacon x. Each participant i computes the VRF output y_i and associated proof π_i, that correspond to input x in combination with their own private key sk_i; i.e., $y_i, \pi_i \leftarrow VRF(x, sk_i)$. Note that the implementation may use an input deterministically derived from x rather than x itself, e.g., x concatenated by a label, in order to obtain VRF outputs that are different from those used for other purposes, e.g., to randomize the mixnet topology. Participants then seed a sampler with their y_i and public information about the network (mix node descriptors and network parameters). This sampler outputs pseudo-random values for creating valid sphinx packets: random cyclic group element, path selection through mixnet layers, exponential delays per hop, and time window when the message should be sent.

The fact that all the parameters in measurement messages are pseudo-randomly derived from the sampler makes these messages *reproducible* given y_i in terms of time of sending, full path through the network,

[2] As given by https://metrics.torproject.org.

delays incurred at each hop, and randomness used to derive per-hop shared keys; all this goes for both forward packets and for any SURBs embedded in them. This enables public verifiability, ensuring that participants cannot maliciously craft valid headers for measurement messages. Note that the choice of gateway is *not* part of the sampler output. This is because participants must be free to send the measurement messages through the gateway that they use for their other traffic, which is not public information and thus not known by the sampler. Communications between senders and gateways are encrypted with a symmetric session key, but otherwise gateways do not actually perform any Sphinx processing operations; instead, they simply forward received Sphinx packets to the mixnet.

2. Routing measurement messages. All participants locally generate their measurement messages in advance, before the start of the epoch in which the messages are valid. Once the epoch starts, participants send their measurement messages within the windows of time that were determined by the sampler. The messages are Sphinx-encrypted, indistinguishable from any other messages routed in the mixnet, and they are thus routed normally through the network.

As part of the Sphinx replay protection mechanism, mix nodes keep track of the messages they have seen, identified by a per-hop reply tag. In this way, if a message is replayed, there's a record that the message was already seen in that epoch, which enables the identification and discarding of the duplicate. The Nym PF sampling method piggy-backs on this replay protection mechanism. In addition to keeping a list of seen tags, mix nodes add these tags (together with timestamps identifying when the message was received and sent) to a local Merkle tree. At the end of the epoch, they commit to the root of this Merkle tree by broadcasting it to validators, who publish it in the blockchain. Note that failing to broadcast the Merkle root would result in a mix node being considered as having $PF = 0$ for that epoch, and thus receiving no rewards.

3. Opening measurement messages. Once the Merkle roots of all mix nodes have been published, all measurement participants submit to the validators their VRF pseudo-random seed y_i and corresponding proof of correctness π_i. The validators publish this information in the blockchain, enabling everyone to verify the correctness of the seeds and to re-create the measurement messages that were generated in the epoch, by feeding each participant's y_i to the sampler together with the public network information. Mix nodes then check which measurement messages include their own address in the route. For those that do, they check whether they have recorded that message tag. For the positive cases, they open the corresponding leaf in the Merkle tree to prove that they had committed to that message tag along with the timestamps, proving to everyone having reported processing the message before it was revealed that it was a measurement message. Note that mix nodes have a strong incentive to open measurement messages in their Merkle tree, as failing to do so decreases their performance factor and thus their rewards.

4. Evaluation of quality of service. Based on the seeds y_i and the Merkle tree openings, all participants can verify which messages were reportedly sent and received by mix nodes, which messages were reported lost in which links, and which were inadequately delayed at some hop. Note that the measurement method yields the link between two network entities where a message was lost, rather than exposing the entity that lost a message. This is because it is impossible to distinguish a case in which a node adds the tag to its local Merkle tree and subsequently drops the packet (being thus responsible for the loss) from a case in which a node adds the tag to its local Merkle tree and successfully sends the packet, which is then lost by the successor before recording the tag (in which case the successor is responsible for the loss). In order to statistically identify which nodes are responsible for message losses, the sampler ensures that measurement messages test all combinations of routes within an epoch, in addition to reorganizing the mixnet across epochs to take measurements with different topology configurations.

6.5 Epoch Rewards Allocation Mechanism

At the end of each reward interval, an amount R of rewards is available for distribution to nodes, who subsequently share a part with the stakeholders that supported them by delegating stake and thus increasing their reputation. The rewards R include fees from deposits in the nympool (effective demand) combined with mixmining rewards (inflation). The objective of the reward allocation mechanism is to compensate the nodes that do the work of mixing user traffic and the delegates for backing up good node operators. For the

sake of brevity, in this section we will describe the rewards calculation mechanism only for mix nodes; a similar mechanism is applied to validators.

The goal of the mechanism is to incentivize convergence to an equilibrium with the following properties:

- There are k nodes running continuously (where k is a parameter that depends on the user demand for traffic, the topology of the mixnet and the nodes that are needed to keep on reserve for fast growth in demand), each one with delegated stake that amounts to $1/k$ of total stake.
- All delegates receive the same rewards per unit of stake.
- The stake delegated to each node corresponds to the trust the community places in its operator and is not sensitive to the node's specific capacity or cost effectiveness (provided the node is sufficiently competitive to be selected for the mixnet).
- Nodes with down-times or otherwise diverging from proper mixing operation, lose their delegated stake.
- More competitive nodes (e.g., those with lower costs) are able to translate their competitiveness to higher profit by claiming more profit margin.
- Operators who register multiple nodes, either publicly or covertly (in what amounts to a sybil attack), are penalized by having their node rewards reduced relative to what they would have earned by registering only one node. This is because partitioning stake into multiple pledges leads to lower overall returns.

To achieve the above, the mechanism constraints the rewards for every node in a certain manner and associates them to stake pledged by the node operator and the stake delegated to the node. The constraint creates a "soft-cap" on how much stake it is rational to delegate to a node, nudging stakeholders to an equitable organization behind a target number of mixing nodes and preventing centralization of stake on just a few nodes or fragmentation where too many weak nodes are proposed. In more detail, the mechanism is as follows.

1. The mechanism uses two parameters $\alpha > 0, k \in \mathbb{N}$. There are k nodes that are to be compensated. It is a generalization of the mechanism of [10] to a setting where different amount work is performed by the nodes in each epoch.
   The parameter k sets the number of nodes that the stakeholders are incentivized to create, as rewards are maximized when the mix network converges to k nodes. The exact choice of the parameter k and of the number of nodes that are active and in reserve depend on the expected capacity demand and on the number of mixnet layers. For example, if we want a throughput of 10 Mbs, with each node at a minimum offering 1 Mbs in a mixnet with three layers, we can choose $k = 100$, which accommodates 30 operational nodes and $r = 70$ idling nodes on reserve. The parameter α relates to the sybil resilience of the system.
2. The number of packets transmitted by the i-th node is denoted by w_i. Note that if $w_i = 0$, this means that the node was selected to be on reserve for the epoch. Based on w_i, a fraction ω_i is determined which specifies the fraction of the total effort that was undertaken by the i-th node (e.g., in case no one is idling and every one sends the same amount of packets, we have that $\omega_i = 1/k$). Note that $\sum_{i=1}^{k} \omega_i = 1$.
3. A performance factor (PF) is calculated for each node using the "proof of mixing" protocol given in Section 6.2 that proportionally affects their rewards. For new mix nodes, a grace period may be given where $PF = 1$. After each epoch, the value of PF is adjusted according to the node's performance as determined by proof of mixing. If a node underperforms in a number of epochs (which is a system parameter), additional actions are taken by the Nym network by removing the byzantine node from the set of eligible nodes and temporarily revoking its registration, requesting the node to fix any technical issues. If a node continues to fail to perform, the node operator can be banned from re-registering by freezing or slashing their pledge.
4. The amount of rewards apportioned for the i-th node and its delegates is equal to

$$R_i = PF_i \cdot R \cdot (\sigma'_i \cdot (\omega_i \cdot k) + \alpha \cdot \lambda'_i)/(1 + \alpha),$$

   where σ_i is the total stake delegated to the i-th node, λ_i is the stake that the operator has pledged to their node, $\sigma'_i = \min\{\sigma_i, 1/k\}$ and $\lambda'_i = \min\{\lambda_i, 1/k\}$. We observe the following budget balance property

$$\sum_{i=1}^{k} R_i \leq R.$$

This follows immediately as $\sum_{i=1}^{k} \omega_i = 1$, $\sum_{i=1}^{k} \lambda_i \leq 1$ and $(\sigma'_i \cdot k) \leq 1$ for $i = 1, \ldots, k$.
Note that the above ensure that there are no incentives to delegate more than $1/k$ of the total stake to one node and that stakeholders are incentivized to create k nodes.

5. Given the above, the i-th operator is credited with the following amount:

$$[\min\{PP(w_i), R_i\} + (PM_i + (1 - PM_i) \cdot (\lambda_i/\sigma_i)) \cdot (R_i - PP(w_i))]^+,$$

where $[\cdot]^+ = \max\{0, \cdot\}$, PM_i is the declared profit margin of the i-th operator, while a delegate with delegated stake s, receives

$$[(1 - PM_i) \cdot (s/\sigma_i) \cdot (R_i - PP(w_i))]^+,$$

6. The above allocation process may result in leftover funds. This can happen whenever some rewards remain unclaimed due to nodes being less than "saturated" (i.e., they have an amount of stake that is less than $1/k$) or their performance is below par. These leftover funds are returned to the reserve, where they can be claimed in the future to stabilize rewards and retain node operators in the event of temporary drops of demand.

The parameter α incentivizes pledging and can be used to mitigate sybil attacks. Other sybil-mitigation measures are implicit in the incentives mechanism, such as the screening that stakeholders perform to make stake delegation decisions. The mechanism offered by the α parameter is not a substitute to screening by stakeholders, but rather a complementary mechanism. Note that the higher the value of α, the more the mechanism tilts towards high pledged operators — while the lower the value of α, the more mechanism will tilt towards the most efficient operators. Observe that in typical conditions where $\sum_i \lambda_i < 1$, a certain portion of rewards will remain unclaimed and be returned to the reserve.

The value of k as well as the percentage of idling nodes will be adjusted at regular intervals (such as on a monthly basis), with the adjustment based on the predicted demand of mixnet capacity. Note that adjusting k requires delegates to take actions to converge to the new equilibrium, so stakeholders should be actively participating in judging the reputation of nodes to make delegation decisions. While stakeholder inaction is not harmful, it leads to a smaller amount of rewards.

7 Conclusion

The internet was once heralded as a great step forward in the co-evolution of humanity and technology [73]. It was to provide universal access to information, establishing a planetary dialogue and the possibility of collective self-determination on a scale previously unimaginable. Instead, it is becoming the infrastructure for a society of control. Metadata analysis is the linchpin in this process, as it can be even more predictive than content analysis at anticipating – and thereby controlling – human behavior [85].

But the internet is not doomed to be a surveillance machine. There is still a way out. The key lies in work that has almost been forgotten. Near the dawn of the Internet, a small band of pioneers – the cypherpunks – realized that the Internet could be transformed into a net of surveillance on a scale never previously imagined and set out to come up with a solution.

The word *nym* comes from the Greek for "name." It was first used as a term of art in early cypherpunk re-mailers like Mixmaster in the 1990s [23]. As outlined in *True Nyms and Cryptoanarchy*, "digital pseudonyms, the creation of persistent network personas that cannot be forged by others yet are unlinkable to the 'true names' of their owners," are essential to "ensuring free speech, allowing controversial opinions to be aired, and providing for economic transactions that cannot be blocked" [78]. For decades, the crucial components of a privacy-enhanced Internet – digital cash, anonymous credentials, and mix networking – were never fully assembled into a usable platform. Although startups like Zero Knowledge Systems deployed commercial mixnets during the original "dot com" boom in the late 1990s, the lack of a digital cash like Bitcoin led to a failure to reach widespread adoption in the early 2000s. Meanwhile, non-profit funding models such as the one demonstrated by Tor have produced crowd-sourced or government subsidized privacy-enhancing technologies that do not need to tackle problems of reputation, incentives, and economic sustainability.

In tackling these challenges, Nym can provide the foundation for a secure and private internet, enabling users to access services without being monitored and tracked. While Tor and p2p systems only aim to protect against a weak threat model, mixnets can provide resistance against adversaries that have a godlike

view of all the activity on the entire network. Although such adversaries once seemed perhaps too paranoid, the Snowden revelations have showed that the NSA has these capabilities. With the rise of big data and artificial intelligence, an ever-increasing number of actors can deploy these capabilities – ranging from small nation-states to commercial ventures and super-empowered individuals.

Mixnets are currently the only known working solution to large-scale traffic analysis. Prior to Nym, one of the shortcomings of mixnets has been the need for expensive computation to make network traffic private. Tokenizing reputation via the NYM token is a core breakthrough, in that it makes it possible for the computation necessary to provision privacy to become profitable and arbitrarily scale to meet demand. Tokenized rewards ensure that each component in the Nym network has an incentive to provision privacy. The cycle of earning and using NYM enables a mutually beneficial relationship between users, service providers, and the Nym network itself. The second core breakthrough is using statistical sampling in the proof of mixing protocol, which enables Nym to achieve a high degree of reliability and quality of service in mixing without requiring slow and expensive cryptographic techniques. By routing with verification by lightweight statistical sampling rather than the computationally expensive task of brute-forcing hash puzzles, Nym avoids excessive environmental consequences. What energy costs are incurred by Nym are due to providing useful work in the form of mixing packets.

There are many important applications developers can build on top of Nym. Some of these applications could save money, others could save lives. For example:

- *Cryptocurrency wallets*: While there are some Tor-enabled cryptocurrency wallets, there is demand for privacy-enhanced wallets that enable both private transactions and privacy at the network level to secure the identities of users from third parties.
- *DeFi*: Decentralized finance needs transactions to be indistinguishable, including sender anonymity on the network-level, in order to prevent front-running and enable darkpools, as per mature institutional markets.
- *Payment channels*: Due to their network structure, off-chain payment channels are vulnerable to many attacks that a mixnet would ameliorate [64].
- *Secure Messaging*: Current secure messaging applications use sophisticated end-to-end encryption, but expose metadata and do not resist traffic analysis of their servers.
- *File-sharing*: In some cases, it is important to share sensitive large files in a way that preserves the anonymity of both the sender and the receiver – for example, in communication between journalists and whistle-blowers.
- *Identity management*: Personal data, such as medical records, would benefit from the use of anonymous credentials and network-level privacy to prevent identity theft and algorithmic discrimination.
- *Multimedia Streaming*: Audio and video conferencing is increasingly used for highly sensitive communication; because they can tolerate packets being lost and arriving out of order, these communication protocols are suitable for usage with a mixnet [27].

Our goal is for the Nym mixnet to provision the future of privacy for a large fraction of the internet, where message-based routing is becoming increasingly relevant in domains from instant messaging to cryptocurrency transactions. This is plausible because, unlike most blockchain designs, Nym *scales horizontally*: just as websites can increase capacity by adding additional web servers, Nym can handle more traffic simply by adding new mix nodes. As more users join the network, Nym can speed up and become more efficient, as the involvement of more users on the mixnet as a whole makes it possible to achieve the same amount of privacy by means of smaller mixing delays and lower cover traffic. Other technologies, such as Tor with its circuit-based routing, may ultimately be better suited for use-cases like Web browsing, despite their weaker threat models.

Nym aims to upgrade the internet in a way that honors its fundamental engineering values, enabling full decentralization of components while inspiring developers to build the next generation of privacy-enhanced applications taking advantage of Nym's technology. This may sound ambitious, but it has been done before – as an overlay network running on top of the existing internet, Nym can add users and applications the same way that TLS spread encryption across the internet, starting with one application and growing as more and more applications require improved privacy. We envision that the widespread adoption of Nym will begin with applications that involve a high demand for privacy – such as cryptocurrency – then grow into ever more generic applications, like secure messaging and decentralized VPNs.

Although it is more specified than any competitor, the design put forward here is not complete. There is work to do. As we continue to build the Nym network, we will conduct a full analysis of each component

with security arguments and proofs, including empirical analysis of performance and anonymity trade-offs via simulation. Cypherpunks write code, as code impacts the real world more than any theoretical design [57]. In shipping code early and often, we will no doubt encounter unexpected issues; we may have to alter, upgrade, or even remove components of the design outlined in this whitepaper. In this spirit, the cryptoeconomics and security parameters, as well as a fully-fledged security analysis, should be presented separately.

There are a number of crucial avenues for future research. The architecture of Nym should be tested against various kinds of attacks; researchers must engage in working with obfuscation tools at the gateway into the network in order to make access to Nym undetectable. The roles of gateways and validators need to be more thoroughly specified and verified. Verifiable and location-aware routing is essential to improve speed and prevent attacks. The more social aspects of Nym deserve further research – for example, to develop best practices for service providers to deal with potentially malicious user behavior on a per-service manner. The use of secure hardware for key generation and the cryptographic operations of nodes should be explored as soon as secure hardware itself becomes more open and mature. Various considerations around post-quantum cryptography and algorithm agility will be mapped for every component. The Sphinx packet format itself needs to be streamlined and optimized for performance. One of the most exciting avenues of research is for crucial operations in the processing of Sphinx packets to happen in router hardware; this could enable Nym to scale to use-cases previously unsuitable for mix networking, becoming the default for the entire internet [20].

As intertwined economic and ecological catastrophes beset us, the internet is at a crossroads – and with it, the fate of all humanity. Today, the internet permits pervasive surveillance that could become the basis of unprecedented forms of tyranny. Yet at the same time, the kind of communication and coordination that the internet makes possible represents the last best hope for humanity on a planetary scale. Whether the core protocols of the internet end up supporting privacy as a core feature will determine whether our future holds servitude or liberation. Already, ordinary people from Syria to Europe are facing life or death consequences for their software choices. While privacy might seem like a luxury, privacy options can make or break freedom of the press and popular resistance to totalitarian power. With the power of the internet enhanced by decentralized networks like Nym, freedom will have a fighting chance. Message us via one of our channels. Install the Nym software. If you are a programmer, create privacy-preserving applications. Use privacy-enhancing technologies in your daily life. Educate others. Help us to make a privacy-enhanced internet a reality.

Acknowledgments

The concept of Nym was originated in 2017 at the *Financial Cryptography and Data Security* conference when Adam Back expressed to Harry Halpin that he thought decentralized computation could be used for privacy, and Harry Halpin formulated the core idea of combining anonymous credentials with an incentivized mixnet. This concept was initially reviewed by George Danezis (University College London) and Aggelos Kiayias, with critique from Moxie Marlinspike and Trevor Perrin, and then presented to the European Commission-funded H2020 PANORAMIX project. The first draft version of this document was written by Harry Halpin with George Danezis and Ania Piotrowska. Ania Piotrowska (Nym Technologies) contributed valuable commentary and also some of the graphics used in this document. The Nym team then continued to contribute commentary, including Dave Hrycyszyn, Jess Hrycyszyn, Alexis Roussel, and Jedrzej Stuczynski. The Nym Scientific Advisory Board provided detailed reviews on this final document, including Mustafa al-Bassam (LazyLedger), Alfredo Duran (University of Luxembourg), Tariq Elahi (University of Edinburgh), Ben Laurie (Google), Bart Preneel (KU Leuven), and Carmela Troncoso (EPFL). We'd like to thank Alexis Aiono, Jeff Burdges (Web 3.0 Foundation), Teck Chia (Binance), Lasse Clausen (1kx), Cory Doctorow, Sam Hart (Interchain), Christopher Heymann (1kx), Nadim Kobeissi, Ben Livshits (Brave), Richard Ma (Quantstamp), Chelsea Manning, Lior Messika (Eden Block), Rachel Rose O'Leary, Dermot O'Riordan (Eden Block), Stephen Palley, Nicola Santoni (Lemniscap), Amir Taaki, B. Traven, Guy Weress, Will Wolf (Polychain), and Guy Wuollet (a16z) among other anonymous reviewers for their comments.

References

1. Masayuki Abe. Mix-networks on permutation networks. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 258–273. Springer, 1999.

2. Alessandro Acquisti, Roger Dingledine, and Paul Syverson. On the economics of anonymity. In *International Conference on Financial Cryptography*, pages 84–102. Springer, 2003.
3. Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. Enabling blockchain innovations with pegged sidechains, 2014. http://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains.
4. Stephanie Bayer and Jens Groth. Efficient zero-knowledge argument for correctness of a shuffle. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 263–280. Springer, 2012.
5. Omar Besbes and Assaf J. Zeevi. On the minimax complexity of pricing in a changing environment. *Oper. Res.*, 59(1):66–79, 2011.
6. Alex Biryukov and Sergei Tikhomirov. Deanonymization and linkability of cryptocurrency transactions based on network analysis. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 172–184. IEEE, 2019.
7. Rainer Bohme, George Danezis, Claudia Diaz, Stefan Kopsell, and Andreas Pfitzmann. Mix Cascades vs. Peer-to-Peer: Is One Concept Superior? In *Proceedings of Privacy Enhancing Technologies, PET 2004*, volume 3424 of *LNCS*, pages 243–255. Springer-Verlag, 2004.
8. Shaileshh Bojja Venkatakrishnan, Giulia Fanti, and Pramod Viswanath. Dandelion: Redesigning the bitcoin network for anonymity. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(1), June 2017.
9. Stefan A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA, 2000.
10. Lars Brünjes, Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. Reward sharing schemes for stake pools. *arXiv preprint arXiv:1807.11218*, 2018.
11. Sam Burnett, Nick Feamster, and Santosh Vempala. Chipping away at censorship firewalls with user-generated content. In *Proceedings of the 19th USENIX Conference on Security*, USENIX Security'10, pages 29–29, Berkeley, CA, USA, 2010. USENIX Association.
12. Jan Camenisch, Maria Dubovitskaya, Anja Lehmann, Gregory Neven, Christian Paquin, and Franz-Stefan Preiss. Concepts and languages for privacy-preserving attribute-based authentication. In Simone Fischer-Hübner, Elisabeth de Leeuw, and Chris Mitchell, editors, *Policies and Research in Identity Management*, pages 34–52, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
13. Jan Camenisch, Anja Lehmann, and Gregory Neven. Electronic identities need private credentials. *IEEE Security and Privacy*, 10(1):80–83, January 2012.
14. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, pages 93–118, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
15. Jake S Cannell, Justin Sheek, Jay Freeman, Greg Hazel, Jennifer Rodriguez-Mueller, Eric Hou, Brian J Fox, and Steven Waterhouse. Orchid: A decentralized network routing market. Technical report, Orchid Labs, Tech. Rep., 2019.[Online]. Available: https://www. orchid. com . . . , 2019.
16. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.
17. David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology*, pages 199–203, Boston, MA, 1983. Springer US.
18. David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, October 1985.
19. David Chaum, Debajyoti Das, Farid Javani, Aniket Kate, Anna Krasnova, Joeri de Ruiter, and Alan T. Sherman. cmix: Mixing with minimal real-time asymmetric cryptographic operations. In *Applied Cryptography and Network Security - 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings*, pages 557–578, 2017.
20. Chen Chen, Daniele E Asoni, Adrian Perrig, David Barrera, George Danezis, and Carmela Troncoso. Taranet: Traffic-analysis resistant anonymity at the network layer. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 137–152. IEEE, 2018.
21. Shuo Chen, Rui Wang, XiaoFeng Wang, and Kehuan Zhang. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, SP '10, pages 191–206, Washington, DC, USA, 2010. IEEE Computer Society.
22. Coinbuzz. Is this startup threatening the entire bitcoin network? http://www.coinbuzz.com/2015/03/15/is-this-startup-threatening-the-entire-bitcoin-network/.
23. Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster protocol version 2< draft-moeller-v2-01.txt. *Online specification*, 2003.
24. George Danezis. The traffic analysis of continuous-time mixes. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies*, pages 35–50, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
25. George Danezis, Claudia Diaz, and Paul F. Syverson. Systems for Anonymous Communication. In B. Rosenberg and D. Stinson, editors, *CRC Handbook of Financial Cryptography and Security*, pages 341–390. Chapman & Hall, 2010.

26. George Danezis, Claudia Diaz, and Carmela Troncoso. Two-sided statistical disclosure attack. In *Proceedings of the 7th International Conference on Privacy Enhancing Technologies*, PET'07, pages 30–44, Berlin, Heidelberg, 2007. Springer-Verlag.
27. George Danezis, Claudia Diaz, Carmela Troncoso, and Ben Laurie. Drac: An architecture for anonymous low-volume communications. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 202–219. Springer, 2010.
28. George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a Type III anonymous remailer protocol. In *2003 Symposium on Security and Privacy, 2003.*, pages 2–15. IEEE, 2003.
29. George Danezis and Ian Goldberg. Sphinx: A compact and provably secure mix format. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, SP '09, pages 269–282, Washington, DC, USA, 2009. IEEE Computer Society.
30. George Danezis and Len Sassaman. Heartbeat traffic to counter (n-1) attacks: Red-Green-Black Mixes. In *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society*, WPES '03, page 89–93, New York, NY, USA, 2003. Association for Computing Machinery.
31. George Danezis and Andrei Serjantov. Statistical disclosure or intersection attacks on anonymity systems. In *Proceedings of the 6th International Conference on Information Hiding*, IH'04, pages 293–308, Berlin, Heidelberg, 2004. Springer-Verlag.
32. George Danezis and Bettina Wittneben. The economics of mass surveillance and the questionable value of anonymous communications. In *5th Annual Workshop on the Economics of Information Security, WEIS 2006, Robinson College, University of Cambridge, England, UK, June 26-28, 2006*, 2006.
33. Debajyoti Das, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate. Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency - choose two. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 108–126, 2018.
34. Sergi Delgado-Segura, Surya Bakshi, Cristina Pérez-Solà, James Litton, Andrew Pachulski, Andrew Miller, and Bobby Bhattacharjee. Txprobe: Discovering bitcoin's network topology using orphan transactions. In Ian Goldberg and Tyler Moore, editors, *Financial Cryptography and Data Security, FC 2019*, volume 11598 of *Lecture Notes in Computer Science*, pages 550–566. Springer, 2019.
35. Claudia Diaz. *Anonymity and Privacy in Electronic Services*. PhD thesis, Katholieke Universiteit Leuven, 2005. Bart Preneel and Joos Vandewalle (promotors).
36. Claudia Diaz, Steven J. Murdoch, and Carmela Troncoso. Impact of network topology on anonymity and overhead in low-latency anonymity networks. In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, pages 184–201, Berlin, Heidelberg, 2010. Springer-Verlag.
37. Claudia Diaz and Bart Preneel. Taxonomy of Mixes and Dummy Traffic. In *Working Conference on Privacy and Anonymity in Networked and Distributed Systems*, pages 215–230, Toulouse,FR, 2004. Kluwer Academic Publishers.
38. Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies*, PET'02, pages 54–68, Berlin, Heidelberg, 2002. Springer-Verlag.
39. Whitfield Diffie and Susan Landau. *Privacy on the Line: The Politics of Wiretapping and Encryption, Updated and Expanded Edition*. The MIT Press, 2007.
40. Roger Dingledine, Michael J Freedman, David Hopwood, and David Molnar. A reputation system to increase mix-net reliability. In *International Workshop on Information Hiding*, pages 126–141. Springer, 2001.
41. Roger Dingledine and Nick Mathewson. Anonymity loves company: Usability and the network effect. In *WEIS*, 2006.
42. Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC, 2004.
43. Roger Dingledine, Andrei Serjantov, and Paul Syverson. Blending different latency traffic with alpha-mixing. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies*, pages 245–257, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
44. Roger Dingledine and Paul Syverson. Reliable mix cascade networks through reputation. In *International Conference on Financial Cryptography*, pages 253–268. Springer, 2002.
45. Christoph Egger, Johannes Schlumberger, Christopher Kruegel, and Giovanni Vigna. Practical attacks against the I2P network. In *Proceedings of the 16th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2013)*, October 2013.
46. Giulia Fanti, Shaileshh Bojja Venkatakrishnan, Surya Bakshi, Bradley Denby, Shruti Bhargava, Andrew Miller, and Pramod Viswanath. Dandelion++: Lightweight cryptocurrency networking with formal anonymity guarantees. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(2):29, 2018.
47. Elke Franz, Anja Jerichow, and Guntram Wicke. A payment scheme for mixes providing anonymity. In *Trends in Distributed Systems for Electronic Commerce*, pages 94–108. Springer, 1998.
48. Ian Goldberg and Adam Shostack. Freedom network 1.0 architecture and protocols. *Zero-Knowledge Systems White Paper*, 1999.

49. David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In Ross Anderson, editor, *Information Hiding*, pages 137–150, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.
50. Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson. Universal re-encryption for mixnets. In *Cryptographers' Track at the RSA Conference*, pages 163–178. Springer, 2004.
51. Benjamin Greschbach, Gunnar Kreitz, and Sonja Buchegger. The devil is in the metadata — new privacy challenges in decentralised online social networks. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 333–339, 2012.
52. Harry Halpin. Nym credentials: Privacy-preserving decentralized identity with blockchains. In *IEEE International Cryptovalley Conference on Blockchain Technologies (CVCBT)*, pages 56–67, 2020.
53. Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 129–144, 2015.
54. Michael Herrmann and Christian Grothoff. Privacy implications of performance-based peer selection by onion routers: A real-world case study using I2P. In *Proceedings of the 11th Privacy Enhancing Technologies Symposium (PETS 2011)*, July 2011.
55. Nguyen Phong Hoang, Panagiotis Kintis, M. Antonakakis, and M. Polychronakis. An empirical study of the i2p anonymity network and its censorship resistance. In *Proceedings of the Internet Measurement Conference 2018*, 2018.
56. HOPR. https://hoprnet.org.
57. Eric Hughes. A cypherpunk's manifesto. https://news.bitcoin.com/eric-hughes-a-cypherpunks-manifesto/, May 2020.
58. Invisible Internet Project (I2P). https://geti2p.net/en/.
59. Mitsuru Iwamura, Yukinobu Kitamura, Tsutomu Matsumoto, and Kenji Saito. Can we stabilize the price of a cryptocurrency?: Understanding the design of bitcoin and its potential to compete with central bank money. *Hitotsubashi Journal of Economics*, pages 41–60, 2019.
60. Rob Jansen, Aaron Johnson, and Paul Syverson. Lira: Lightweight incentivized routing for anonymity. Technical report, Naval Research Lab Washington DC, 2013.
61. Seong Hoon Jeong, Ah Reum Kang, Joongheon Kim, Huy Kang Kim, and Aziz Mohaisen. A longitudinal analysis of .I2P leakage in the public dns infrastructure. In *Proceedings of the 2016 ACM SIGCOMM Conference*, SIGCOMM '16, pages 557–558, New York, NY, USA, 2016. ACM.
62. Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. Users get routed: Traffic correlation on tor by realistic adversaries. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 337–348, New York, NY, USA, 2013. ACM.
63. Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. A critical evaluation of website fingerprinting attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 263–274, New York, NY, USA, 2014. ACM.
64. George Kappos, Haaroon Yousaf, Ania Piotrowska, Sanket Kanjalkar, Sergi Delgado-Segura, Andrew Miller, and Sarah Meiklejohn. An empirical analysis of privacy in the lightning network. *arXiv preprint arXiv:2003.12470*, 2020.
65. Jacob Kastrenakes. NordVPN reveals server breach that could have let attacker monitor traffic. https://www.theverge.com/2019/10/21/20925065/nordvpn-server-breach-vpn-traffic-exposed-encryption, October 2019.
66. Dogan Kesdogan, Jan Egner, and Roland Büschkes. Stop- and- Go-MIXes providing probabilistic anonymity in an open system. In David Aucsmith, editor, *Information Hiding*, pages 83–98, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
67. N. Bora Keskin and Assaf J. Zeevi. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Oper. Res.*, 62(5):1142–1167, 2014.
68. Mohammad Taha Khan, Joe DeBlasio, Geoffrey M Voelker, Alex C Snoeren, Chris Kanich, and Narseo Vallina-Rodriguez. An empirical analysis of the commercial vpn ecosystem. In *Proceedings of the Internet Measurement Conference 2018*, pages 443–456. ACM, 2018.
69. Kovri. https://web.getmonero.org/resources/moneropedia/kovri.html.
70. Albert Kwon, David Lu, and Srinivas Devadas. XRD: Scalable messaging system with cryptographic privacy. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 759–776, Santa Clara, CA, 2020. USENIX Association.
71. Brian N. Levine, Michael K. Reiter, Chenxi Wang, and Matthew Wright. Timing attacks in low-latency mix systems. In Ari Juels, editor, *Financial Cryptography*, pages 251–265, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
72. Steven Levy. *Crypto: How the code rebels beat the government–saving privacy in the digital age*. Penguin, 2001.
73. Joseph CR Licklider. Man-computer symbiosis. *IRE transactions on human factors in electronics*, HFE(1):4–11, 1960.
74. Z. Ling, J. Luo, W. Yu, X. Fu, D. Xuan, and W. Jia. A new cell-counting-based attack against tor. *IEEE/ACM Transactions on Networking*, 20(4):1245–1261, Aug 2012.

75. Akshaya Mani, T. Wilson-Brown, Rob Jansen, Aaron Johnson, and Micah Sherr. Understanding tor usage with privacy-preserving measurement. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, page 175–187, New York, NY, USA, 2018. Association for Computing Machinery.
76. Adam Martin. LulzSec hacker exposed by the service he thought would hide him. https://www.theatlantic.com/technology/archive/2011/09/lulzsec-hacker-exposed-service-he-thought-would-hide-him/337545/, September 2011.
77. S. C. Matz, M. Kosinski, G. Nave, and D. J. Stillwell. Psychological targeting as an effective approach to digital mass persuasion. *Proceedings of the National Academy of Sciences*, 114(48):12714–12719, 2017.
78. Timothy C May. True nyms and crypto anarchy. *True names and the opening of the cyberspace frontier*, pages 33–86, 2001.
79. Jonathan Mayer, Patrick Mutchler, and John C. Mitchell. Evaluating the privacy properties of telephone metadata. *Proceedings of the National Academy of Sciences*, 113(20):5536–5541, 2016.
80. Silvio Micali, Salil Vadhan, and Michael Rabin. Verifiable random functions. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99, page 120, USA, 1999. IEEE Computer Society.
81. Hooman Mohajeri Moghaddam, Baiyu Li, Mohammad Derakhshani, and Ian Goldberg. Skypemorph: Protocol obfuscation for tor bridges. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 97–108, New York, NY, USA, 2012. ACM.
82. Monero: Private digital currency. https://web.getmonero.org/.
83. Steven J. Murdoch and Piotr Zieliński. Sampled traffic analysis by internet-exchange-level adversaries. In Nikita Borisov and Philippe Golle, editors, *Privacy Enhancing Technologies*, pages 167–183, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
84. Rebekah Overdorf, Mark Juarez, Gunes Acar, Rachel Greenstadt, and Claudia Diaz. How unique is your .onion?: An analysis of the fingerprintability of tor onion services. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 2021–2036, New York, NY, USA, 2017. ACM.
85. Alex Pentland. On the collective nature of human intelligence. *Adaptive behavior*, 15(2):189–198, 2007.
86. Vasile C Perta, Marco V Barbera, Gareth Tyson, Hamed Haddadi, and Alessandro Mei. A glance through the VPN looking glass: IPv6 leakage and DNS hijacking in commercial VPN clients. *Proceedings on Privacy Enhancing Technologies*, 2015(1):77–91, 2015.
87. Andreas Pfitzmann and Marit Hansen. Anonymity, unobservability, and pseudeonymity — a proposal for terminology. In *International Workshop on Designing Privacy Enhancing Technologies (PETS): Design Issues in Anonymity and Unobservability*, pages 1–9, Berlin, Heidelberg, 2001. Springer-Verlag.
88. Birgit Pfitzmann and Andreas Pfitzmann. How to break the direct rsa-implementation of mixes. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology — EUROCRYPT '89*, pages 373–381, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.
89. Ania M Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. The Loopix anonymity system. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1199–1216, 2017.
90. Michael K. Reiter, Aviel D. Rubin, and Aviel D. Rubin. Crowds: Anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92, November 1998.
91. Nym Technologies SA. HOPR network design and flaws — a system analysis. https://medium.com/nymtech/hopr-network-design-and-flaws-a-system-analysis-43c1d244a779, 2020.
92. Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 459–474. IEEE, 2014.
93. Sentinel dVPN. https://sentinel.co/dvpn.
94. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies*, PET'02, pages 41–53, Berlin, Heidelberg, 2002. Springer-Verlag.
95. Andrei Serjantov, Roger Dingledine, and Paul Syverson. From a trickle to a flood: Active attacks on several mix types. In *International Workshop on Information Hiding*, pages 36–52. Springer, 2002.
96. Andrei Serjantov and Peter Sewell. Passive-attack analysis for connection-based anonymity systems. *International Journal of Information Security*, 4(3):172–180, Jun 2005.
97. Vitaly Shmatikov and Ming-Hsiu Wang. Timing analysis in low-latency mix networks: Attacks and defenses. In *Proceedings of the 11th European Conference on Research in Computer Security*, ESORICS'06, pages 18–33, Berlin, Heidelberg, 2006. Springer-Verlag.
98. Sandra Siby, Marc Juárez, Claudia Díaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. Encrypted DNS -> privacy? A traffic analysis perspective. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020.
99. Atul Singh, Tsuen-Wan Ngan, Peter Druschel, and Dan S. Wallach. Eclipse attacks on overlay networks: Threats and defenses. In *25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006.
100. Emil Sit and Robert Tappan Morris. Security considerations for peer-to-peer distributed hash tables. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *First International Workshop on Peer-to-Peer Systems (IPTPS)*, volume 2429 of *Lecture Notes in Computer Science*, pages 261–269. Springer, 2002.

101. Alberto Sonnino, Mustafa Al-Bassam, Shehar Bano, and George Danezis. Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers. *CoRR*, abs/1802.07344, 2018.
102. Carmela Troncoso, Marios Isaakidis, George Danezis, and Harry Halpin. Systematizing decentralization and privacy: Lessons from 15 years of research and deployments. *Proceedings on Privacy Enhancing Technologies*, 2017(4):404–426, 2017.
103. Jelle van den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles*, SOSP '15, page 137–152. Association for Computing Machinery, 2015.
104. Matthew K. Wright, Micah Adler, Brian Neil Levine, and Clay Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Trans. Inf. Syst. Secur.*, 7(4):489–522, November 2004.

Disclaimer

ANY TOKEN PURCHASE AGREEMENTS HAVE NOT BEEN AND WILL NOT BE REGISTERED UNDER THE SECURITIES ACT OF 1933 (THE "SECURITIES ACT") OR THE LAWS OF ANY U.S. STATE, AND MAY NOT BE OFFERED, SOLD, OR DELIVERED WITHIN THE UNITED STATES OR TO OR FOR THE ACCOUNT OR BENEFIT OF U.S. PERSONS EXCEPT PURSUANT TO AN EXEMPTION FROM, OR IN A TRANSACTION NOT SUBJECT TO THE REQUIREMENTS OF THE SECURITIES ACT. THE NYM TOKEN DOES NOT PROVIDE ANY CLAIMS OR MEMBERSHIP RIGHTS IN NYM TECHNOLOGIES SA.