

Virtual & Invisible Private Network: a Zero-Trust Architecture for Anonymous Communication on the Internet

Frédéric Laurent^{1,†}, Baptiste Polvé^{1,*,†}, Guillaume Nibert^{1,2,*,†} and Alexis Olivereau^{3,†}

¹ Snowpack, F-91400, Orsay, France

² Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

³ Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

Abstract

We present the first version of *Virtual & Invisible Private Network (VIPN)*, a novel low-latency, secure, and anonymous communication technology designed to overcome the limitations of existing network anonymization architectures. Traditional decentralized systems require trusting intermediary proxies in charge of relaying users' traffic. This reliance introduces significant vulnerabilities, including potential traffic attribution, man-in-the-middle attacks and sensitive information exposure if a relay is compromised. Additionally, these architectures often lack compatibility with all IP protocols and QoS; thus they fail to support modern applications effectively. In this paper, we review the shortcomings of current approaches and propose a new architecture that mitigates these risks. We evaluate the guarantees of our solution against various adversarial models and provide first insights from a real-world deployment. Finally, we highlight the current architecture limitations and future ongoing challenges.

Keywords

Anonymous communications, Security and privacy, Anonymity and untraceability, Anonymization, Network overlay, Routing protocol

1. Introduction

While it can be used for criminal purposes, anonymity is essential for multiple reasons such as privacy protection, freedom of speech, investigations (law enforcement and journalism), and fight against crime or foreign influence. Indeed, detecting and combating criminal networks and their operations, organized as an industry often on social networks or hidden forums, require extensive anonymous data collection capabilities. This involves using scrapers and avatars for targeted infiltration of social media platforms or forums. However, present anonymization solutions rely today on trusted third parties (TTP), particularly the relay, requiring a trade-off between anonymization quality and operational security. Ad hoc anonymization chains offer full control but lack mass effect and raise attribution risks. On the contrary, common anonymization solutions (mass-market VPNs, residential proxies, Tor, I2P, etc.) blend users into

C&ESAR'25: Computer & Electronics Security Application Rendezvous, Nov. 19-20, 2025, Rennes, France

*Corresponding author.

†These authors contributed at different levels from idea to implementation and evaluation.

✉ frederic.laurent@snowpack.eu (F. Laurent); baptiste.polve@snowpack.eu (B. Polvé);

guillaume.nibert@snowpack.eu (G. Nibert); alexis.olivereau@cea.fr (A. Olivereau)

🆔 0000-0002-3277-8533 (G. Nibert); 0000-0002-6315-5363 (A. Olivereau)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

a larger users' pool but present risks [1] that can be unacceptable in the context of national security and defense. In this article, authors are presenting *Virtual & Invisible Private Network (VIPN)*, a new anonymization architecture that aims not to rely on TTP. Thus, Section 2 reviews current benefits and limitations of state-of-the-art anonymization solutions. Then, Section 3 highlights the objectives of the proposed zero-trust architecture, in particular its threat model. Section 4 describes the proposed designed architecture and its protocol. Section 5 evaluates the solution against different threat actors as well as experience feedback from a real deployment. Finally, Section 6 presents additional challenges that the architecture still needs to solve.

2. Related Work

David Chaum laid the foundations of anonymous communications field in 1981 in *Untraceable electronic mail, return addresses, and digital pseudonyms* [2]. The approach involves inserting intermediary nodes, called mixes, between senders and receivers, forming a mix cascade. To complicate traffic analysis and prevent identifying senders and receivers from network data, each mix must receive a certain number of messages until a *threshold* is reached, forming a *batch* of messages. Each mix then applies cryptographic operations to messages, lexicographically reorders them, and flushes them out from the newly ordered batch. This batching process introduces delays, rendering traffic analysis attacks more difficult, but also degrades quality of service due to bursty communication. Moreover, it requires large storage space. As a result, Chaum's mix cascades are unsuitable for modern low latency and data intensive applications.

From this seminal paper, subsequent research on anonymity networks has advanced significantly, enabling the identification of four classes of anonymity networks [1]:

- mixnets [2, 1] based, which apply cryptographic operations and mix messages through several intermediate servers to mask origin and destination of messages;
- Tor and Onion Routing [3, 1, 4] based, route upstream traffic by successively encrypting the message (like an onion) through a series of relay nodes to mask origin and destination. For downstream, nodes decrypt onion layers until the destination is reached;
- random walks and DHT [1] based, which work by sending messages through a nodes decentralized network. In these networks, messages are routed either by using random walk algorithms or through a structured DHT-based mechanism. Nobody has a complete route view from the sender to the receiver, including themselves;
- DCnets (Dining-Cryptographer) [1, 5] based, allow group members to communicate public messages anonymously with no one knowing who sent or received a specific message.

Unclassified anonymity networks (I2P [6], P5 [7], CAR [8], etc.) exist and can fall into several classes. In addition to those classes, Pfizmann and Köhntopp introduced a precise terminology for the field, defining key concepts such as anonymity, unlinkability, unobservability, pseudonymity, as well as the roles of sender, receiver, etc. in 2001 [9].

Mixnets. Vuvuzela [10] operates in synchronous rounds and requires only one honest node to ensure anonymity properties. But, its application is limited to private chat. Stadium [11], an improvement over Vuvuzela, operates on a tree structure rather than the Vuvuzela single

chain. It uses differential privacy techniques along with cover traffic, offering stronger privacy guarantees and significantly enhancing Vuvuzela scalability. AnonPoP [12], similar to Vuvuzela, is vulnerable to active traffic analysis attacks and is limited to email or messaging use.

Atom [13] overcomes secure low-latency communications challenge. Instead of letting users define message route, it uses a modified ElGamal encryption scheme to let servers collaboratively process messages without revealing their keys to users. Servers are grouped into 'anytrust' groups; thus if there is at least one honest server per group, the system is protected from malicious servers. The architecture is ideal for anonymous microblogging and secure initial contact, as it effectively balances scalability with robust protection against traffic analysis.

Riffle [14] focuses on efficiency while providing strong privacy. It combines verifiable shuffling and onion encryption with a unique approach called symmetric-key private information retrieval (PIR). To make tracking origin difficult, messages are shuffled and encrypted multiple times by a series of servers. Moreover, it significantly reduces the computational load and latency making it well-suited for scenarios where both anonymity and performance are crucial.

cMix [15] reduces real-time cryptographic latency and computational costs, particularly for lightweight devices. This achieved with a precomputation phase to eliminate time-consuming real-time public-key operations need during the actual communication process. This allows core real-time phase to focus solely on fast modular multiplications, making it highly efficient. By sending messages batches through a mixnodes cascade with an encoded route, cMix ensures strong anonymity, unlinkability between senders and receivers, and robust resistance to traffic analysis attacks. Thus cMix is particularly suitable for low-latency applications such as secure chat, offering strong privacy protections even against global adversaries.

Loopix [16] is the anonymity network, behind Nym [17], that builds a layered mixnet, enhancing privacy through Poisson-distributed message timing and cover traffic. Each client sends both real and cover messages, including "loop" messages, at Poisson determined intervals. Loop and cover messages obscure communication patterns and improve resistance to traffic analysis. The layered architecture helps to scale effectively as the network grows. Although Loopix offers strong anonymity and lower latency than usual, the latency remains too high for real-time applications like VoIP or streaming.

Tor and Onion Routing. Tor [3] is the most well-known and with 3M daily users is widely adopted. It routes internet traffic through a network of volunteer-operated relays, which are nodes that pass on encrypted data. The user's data is encrypted in layers, like an onion. Each relay encrypts or decrypts a layer before passing to the next one. While Tor offers good balance between anonymity and latency, enabling relatively fast web browsing while masking users' identities, it is more vulnerable to traffic correlation attacks compared to Vuvuzela or Stadium and requires trust in the relays, especially the exit one.

Random walks and DHT. Freenet [18] is a peer-to-peer, censorship-resistant file-sharing and storage system. It operates on a decentralized network where files are redundantly stored across multiple nodes. Freenet uses adaptive routing based on Distributed Hash Tables (DHT) to locate files via their binary keys over a depth-first search combined with heuristic methods. The system ensures plausible deniability by making it unclear whether a node is the request originator or simply relaying it. The Darknet mode enhances privacy by relying on trusted

connections for routing within the user's social network.

GNUnet [19] is a censorship-resistant content-sharing system that has expanded to support various applications, including anonymous file-sharing via the GNUnet Anonymous Protocol (GAP). GAP provides anonymity for both requesters and responders using a credit-based routing system, where nodes earn credits by relaying requests and responses. GAP also includes mechanisms to prevent routing loops and allows nodes to opt out of return paths based on network conditions and their load, enhancing both network efficiency and privacy.

Octopus [20], aims to prevent malicious nodes from biasing the routing process while ensuring anonymity in communication paths. It uses iterative lookups, where queries are sent to the closest node in the local routing table until the target is found. Node selection occurs in two phases: initially by the path initiator and then by the last node chosen. Octopus conceals real queries by splitting them across multiple paths and adding dummy traffic. It also includes security measures to detect and exclude malicious nodes.

DCnets. Dissent [21], introduced by Corrigan-Gibbs and Ford in 2010, is a latency-tolerant protocol designed for anonymous communication within small groups. It is built on DC-nets principles to provide strong sender anonymity through multiparty computation and layered encryption. Messages are encrypted and shuffled among participants to obscure their origin, ensuring that the sender remains anonymous. While it guarantees the message content confidentiality through encryption, the protocol's design allows the possibility of revealing the message's presence without disclosing the sender. It also includes mechanisms to maintain accountability and network integrity by addressing denial-of-service (DoS) attacks and malicious behavior through techniques such as go/no-go messages and blame phases.

Unclassified. Among unclassified networks, I2P is a popular alternative to Tor. It operates as a distributed overlay network, composed in 2023 of between 40,000 and 50,000 routers, with the primary goal of facilitating secure anonymous end-to-end communication between nodes within I2P. It uses a distributed hash table (DHT) based on Kademlia [22] protocol to manage network metadata and facilitate node lookups. When users join I2P, they retrieve a list of peers from external sources and contact floodfill nodes which store and manage routing information to gather router details. There is eight synchronized floodfill nodes to prevent malicious manipulation. I2P uses a source-routed approach called garlic routing, where communication is routed through a series of randomly selected nodes. Unlike Tor's bidirectional tunnels, I2P uses distinct unidirectional tunnels to send and receive data. Each user maintains multiple tunnels, and information about these tunnels is obtained from floodfill nodes. The routing process involves selecting inbound and outbound tunnels to form the communication path.

P5 is an anonymous communication system that uses a tree-like structure to broadcast messages. Users are organized into subgroups linked to hierarchical nodes, which allows for efficient message dissemination. However, the hierarchical mode implies that top position gets better performance in exchange of being more identifiable. Thus, P5 highlights the trade-off between communication efficiency and anonymity level.

CAR enhances privacy in wireless ad hoc networks, which are vulnerable to passive attacks and eavesdropping. Along a path, nodes are linked in a virtual chain where each node only knowing its immediate neighbors. In addition, CAR uses unicast-based broadcasting for route

discovery, and data is exclusively transmitted along the established routes. This approach conceals node identities and protects against tracking and linking, providing robust privacy against various passive and some active threats.

3. Motivation and Goals

Among the solutions presented in Section 2, our focus is put on the systems with an active community of users while serving popular anonymization purpose such as VPNs (equivalent to a deterministic single-hop mixnet), Tor network, I2P and Nym. All require making trade-offs between quality of service and security which imply limitations. Here are few of these limitations: consumer VPNs are TTPs, so compromising the VPN service operator gives full content access; moreover VPN connectivity provider can easily perform network analysis for deanonymization purposes. Tor, assimilated to a VPN cascade, remains exposed to traffic analysis attacks [23] and the exit node also acts as TTP [24]. I2P is vulnerable to Sybil attacks¹ where an attacker creates multiple nodes to compromise the network. The recent Nym network is limited to very few services even if work is underway to be usable for other applications (e.g.: use the infra as a peer-to-peer VPN), and the way the onion is constructed is based on the Sphinx packet format [16, 25], which is known to be vulnerable [26]. Unfortunately, Nym's Sphinx modified version is recent and has no specification². In addition, most solutions do not use perfectly secure encryption, most cannot withstand a polynomially unbounded adversary (i.e.: attackers able to break computationally bounded protocols). Finally, as the architectures authorize access outside the overlay, exit nodes are TTP, relay sees both user's upstream and downstream traffic and can therefore perform man-in-the-middle (MITM) attacks.

With these limitations in mind, our goal is to design a technology that presents novel anonymity properties without impairing as much on users quality of experience. In this paper we address the specific issue of TTP in core anonymization networks' architecture building block as it is an aspect that has so far been insufficiently covered by present solutions. We therefore designed a novel architecture that does not rely on any trusted third party at this level. Our design is based on a self reinforcement of security and anonymity properties to eliminate TTP at message routing level, thereby enhancing resilience against powerful adversaries. Because it offers properties beyond sole anonymity and presents a spectrum of applications similar to VPN, we call this technology Virtual and Invisible Private Network (VIPN). This system will have to be improved to address trust in the management component of the system.

3.1. Threat Actors

We consider the following threat actors, ordered from least to most powerful, who may attempt to deanonymize users, intercept or modify content, or alter VIPN behavior: i. **Individual hacker**: it may try different scripts to alter the normal VIPN behavior; ii. **ISP**: user's ISP may identify traffic going through VIPN; iii. **Core network operator**: it can have access to several

¹Invisible Internet Project (I2P), I2P's Threat Model, 2010. Available at <https://geti2p.net/id/docs/how/threat-model#sybil>, accessed on 22 May 2024.

²D. Hryczyszyn, Document the packet format, 2020. Available at: <https://github.com/nymtech/sphinx/issues/35>, accessed on 27 August 2024.

ISPs traffic; iv. **Cloud provider**: it can have access to several servers; v. **Data center operator**: it can have access to servers from several operators; vi. **State**: it can request access to several data center, as well as ISPs; vii. **Global cooperation**: it potentially has access to data from several colluded states.

We propose to study rigorously the network anonymity and security properties as detailed in subsection 3.2, we pool threat actors into adversaries classes commonly used in the literature. As for any anonymity network, we study the *polynomially bounded global passive adversary (GPA)*. This adversary is able to observe the entire network, the traffic between nodes and users. They know the network topology and can carry out traffic analysis or passive network attacks such as BGP-rerouting [27]. They cannot modify, inject, delete or add IP network packets in traffic, nor break computationally bounded security protocols. We also study a *local adversary* which can only see a few network resources, may modify, inject, delete or add network packets, can compromise certain network nodes and their secrets and attempt to diverge from the protocol. This adversary is studied under two characteristics: *polynomially bounded*, by not being able to break computationally bounded cryptographic protocols (e.g. any public-key encryption schemes, AES, TLS, Onion encryption, etc.), and *polynomially unbounded*, by being able to break computationally bounded cryptographic protocols but not information-theoretic secure protocols (e.g. One-time Pad, Shamir's secret sharing, etc.). Finally, we end with two other very much stronger adversary capabilities: the *global active polynomially bounded adversary* and its *unbounded version*. In both cases, the adversary does not control any node, but controls all the network overlay links.

The primary adversary goal is to break honest anonymous users anonymity and security, i.e. to determine users' identities, their contacts, and the messages they exchange. The second adversary goal is to perform a MITM attack inside the network overlay, no matter if anonymity is broken or not. Then, its third goal is to degrade network quality of service, by performing DoS, crashing nodes, trying to make the protocol deviate so that it stops working, etc.

3.2. Anonymity and Security Goals

As mentioned above, we describe and consider properties related to the core VIPN principles (i.e. network overlay architecture and protocol). We are not addressing other system component aspects such as interactions with the nodes' directory and access control to this directory. In particular the two latter being TTP, they raise specific challenges that will be addressed in future papers. As a result, we focus only on the network overlay architecture including the route creation, between a VIPN user and one Online Service they want to access anonymously. Finally, similar to Tor's onion services, VIPN can support a "tunnel" mode for direct communication between two users. This mode builds upon the "privacy mode" described in this paper and will be detailed in future works.

VIPN intends to guarantee the following anonymity properties, based on those defined in the Pfizmann and Köhntopp paper [9], formally defined in Backes et al. paper [28]:

- *Sender anonymity*: a particular message cannot be linked to any sender, and no message can be linked to a specific sender. This concept implies that an adversary cannot determine which of two possible senders is communicating with a target receiver.

- *Sender unobservability* ensures that an adversary cannot tell whether a sender is communicating or not. It directly implies sender anonymity notion. If an adversary cannot observe whether a particular user has sent a useful message or a dummy message (noise, loop message, etc.), this ensures that the adversary cannot identify which user is communicating with the target receiver.
- *Sender-Receiver unlinkability* ensures that unauthorized third-parties cannot link senders and receivers, meaning that an adversary cannot determine whether two specific users are communicating with each other, this property is key for guaranteeing against attribution.

VIPN also aims to guarantee user's message *secrecy* inside the overlay. In addition, we introduce a new property called *transparent man-in-the-middle resistance*, guaranteeing a MITM attack detect-ability between the anonymity network user and the online service contacted. Thus, the packet alteration inside the network should not be meaningful for the service.

3.3. Service Goals

VIPN also should offer several service goals:

- *Low latency and high-capacity*: the network overlay should be suitable for low-latency and data intensive applications such as VoIP, instant messaging, file transfer, etc.
- *Scalability*: as other anonymity networks, it should scale to serve large number of users by efficiently deploying new proxies if necessary.
- *Compatibility*: as other anonymity networks, it should be IP compatible but contrary to most anonymity networks, it should be compatible with all transport protocols.

4. Virtual & Invisible Private Network Design and Protocol

4.1. Overview

4.1.1. Principles

VIPN is made to be deployed over an IP network, to anonymize IP traffic using common transport protocols (TCP, UDP, ICMP) and its architecture relies on several complementary principles: i. **A distributed network overlay** composed of several proxies (also called nodes) operated by various entities (active users or partners similar to miners). Nodes relay protocol messages in a specific way, depending on their role in the transmission; ii. **Multiple and user-defined complementary paths**: Nodes are used at least on two complementary paths defined by the users' device through its application (*connector*). iii. **Packet anonymization and fragmentation**: because VIPN relies on circuit routing, source public IP addresses are unnecessary and are therefore discarded to protect users anonymity. Anonymized IP packets are therefore fragmented into complementary "snowflakes" through a secret-sharing mechanism (currently the XOR logic operator with a randomly generated one-time pad); iv. **Paths anonymous auto-discovery**: paths are anonymously connected inside the network thanks to specific autodiscovery messages based on the aforementioned snowflake-based secret-sharing mechanism; v. **A distributed exit node** to asymmetrize the traffic, external communication is done through collaborative spoofing between two nodes chosen by the user, which helps prevent MITM attacks.

4.1.2. Components

VIPN is made up of various elements necessary for its operation:

- **Nodes (proxies):** servers forwarding the traffic. Those nodes can be categorized in different categories for a specific user route:
 - *Holonode*: node seen by the contacted online service.
 - *S-Node*: standard relay in the network. From the S-Node pool, a user selects a subset of nodes to perform specific functions for their session:
 - * *Pu nodes*: nodes close to the user, they act as entry point and therefore know the user's IP address;
 - * *Ps master node*: node responsible for managing upstream traffic and relaying snowflakes for downstream traffic as well as to discover its slave;
 - * *Ps slave node*: relay to Ps master node snowflakes for upstream traffic and receives snowflakes from holonode for downstream traffic.
- **Nodes' directory** (*currently an accepted trusted third party*): directory containing the different nodes information and status. It is available only to users in users' directories.
- **VIPN agent**: software installed on a user's device to anonymize it through the overlay.
- **VIPN Users directory** (*currently an accepted trusted third party*): central databases containing user's information and authorization for access control management. These directories users have access to nodes' directory. VIPN includes this access control to comply with regulatory laws, but it is not required for achieving the system's properties.

To anonymize a device, VIPN agent needs to establish a *route*. This route consists of at least 5 nodes: 4 S-Nodes (including 2 Pu nodes, 1 Ps master, and 1 Ps slave) and 1 Holonode. Each route has two separate paths, each made up of several serialized circuits between the nodes.

4.2. Route construction

There are several steps involved in creating this route, all based on the user choices (cf. figure 1):

- **Step 1:** User selects at least 5 distinct nodes: 1 Holonode (H) and at least 4 S-nodes.
- **Step 2:** User builds two circuits, one between himself and S-node 1 (S_1) and one between himself and S-node 2 (S_2).
- **Step 3:** To complete the two disjoint paths (i.e.: paths where nodes cannot be part of the two paths), the user requests through his channels established on step 2 that two other circuits get created, one between S_1 and S_3 , one between S_2 and S_4 .
- **Step 4:** S_3 (master node, role defined by the user) searches for its slave (S_4), this is the auto-discovery step. Indeed, to prevent an observer on user- S_3 path to be in a position to discover the second path, the user will not communicate itself S_4 IP address to S_3 . It uses a key anonymity preserving autodiscovery scheme. This scheme includes following steps:
 - i. User generates a token (T) and builds an autodiscovery secret ($SECRET$) as follow: $SECRET = T \parallel hash(T \parallel IP_{S_3} \parallel IP_{S_4}) \parallel IP_H$;
 - ii. User generate a one-time pad (ad_a) and computes ad_b as follow: $ad_b = ad_a \oplus SECRET$;
 - iii. User passes ad_a and

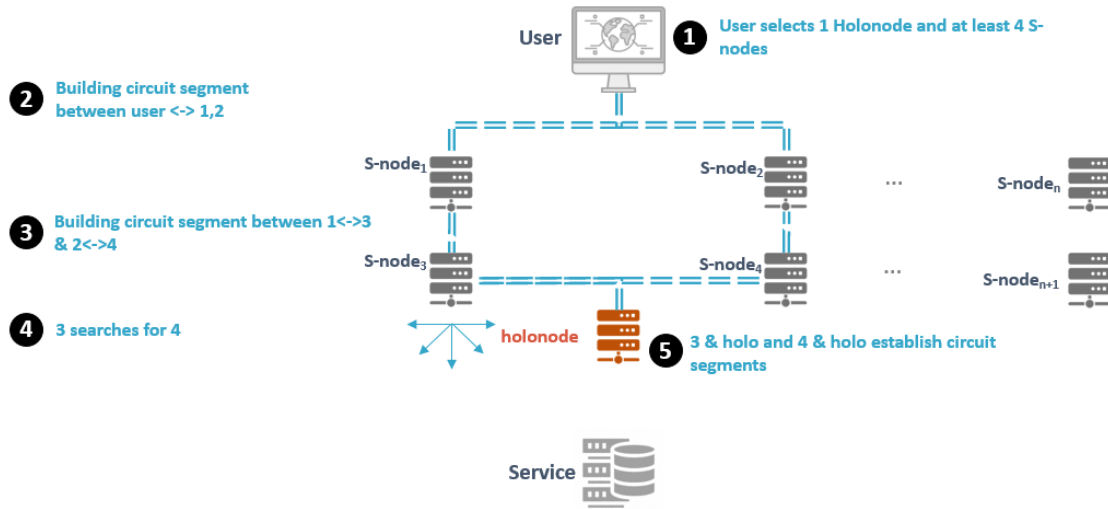


Figure 1: Route creation

ad_b respectively up to S_3 and S_4 through their respective paths; iv. S_3 broadcasts ad_a to the whole network overlay. v. Because S_4 holds ad_b it can recover the *SECRET* and knowing S_3 is the sender confirms with guarantees that it needs to collaborate. In addition it discovers H . vi. S_4 can now send ad_b to S_3 . S_3 can then perform the similar operation and verify S_4 legitimacy as well as discovering H . The bruteforce is prevented thanks to T and the exchange through TLS of the different autodiscovery messages.

- **Step 5:** S_3 , S_4 and H can create circuits between each other to complete the route.

4.3. Nominal mode

After the user creates their VIPN route (as explained earlier), they can use it to connect to online services. Here, we explain how they communicate with a service (cf. figure 2).

- **Steps 1 & 2.** OTP encryption is used. User removes the source IP address from the metadata of the plain request, then produces a ciphertext by randomly generating a mask (same size of the request) which they XOR with their plaintext request ($mask \oplus plaintext\ request = ciphertext$). The mask is sent on one path of the route (S_1 - S_3), the ciphertext on the other (S_2 - S_4 - S_3). Mask and ciphertext are called *snowflakes*.
- **Step 3 & 4.** Once both snowflakes are received by S_3 , they are recombined ($mask \oplus ciphertext = plaintext\ request$). S_3 then fakes H 's IP address as the source in the IP packet header when sending the request to the service, alerts H that it will receive IP packets from Service, and then sends the IP packet over Internet. When it receives the request, Service does not know about S_3 and sends its response to H . With this collaborative spoofing, VIPN distributes the exit node, so upstream and downstream traffic do not pass through a single node, preventing MITM attacks in the network overlay.

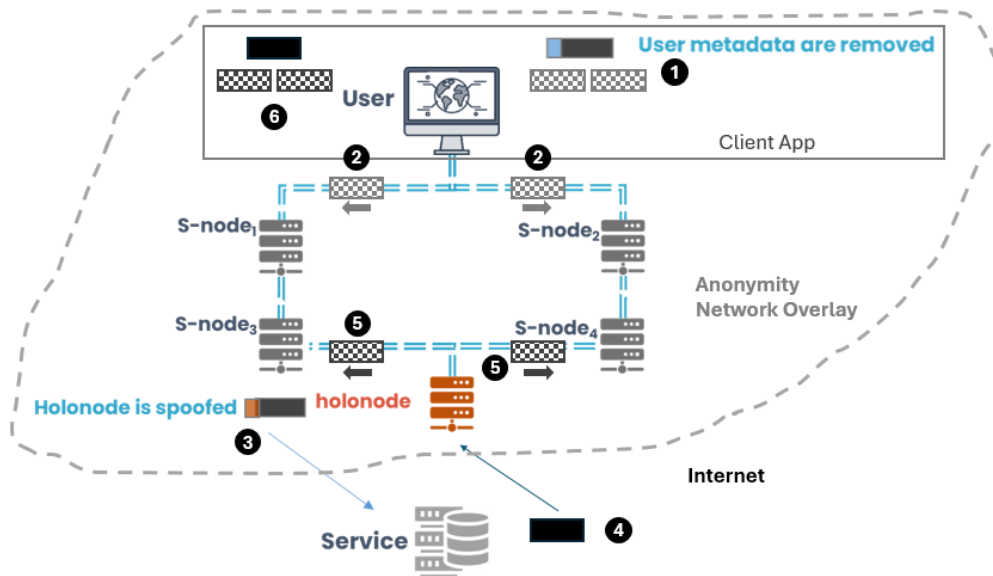


Figure 2: Exchanging messages between a user and a service

- **Steps 5 & 6.** Because H has been warned by S₃, it intercepts IP packets from Service and knows the associated session so it can put them in the proper circuits. Therefore, when the service replies, H intercepts the response and removes its IP address from the destination field in the IP packet, fragments the same way as User into two new complementary snowflakes, and sends them on the two paths. Once User has received them, User XORs them back, adds its own IP address in the destination field and reinjects the traffic in its local network stack. This process allows multiple users to contact the same service, as the session identification relies on the source port used by the user. In case of a port collision, collaborative PAT (synchronized between S₃ and H) handles the transformation.

Finally to enhance security and make it harder to track traffic, all circuits between nodes are bidirectional and multiplexed through the same TLS-protected socket. However, we consider that the system must keep its properties if TLS gets broken.

4.4. Observed properties

With this architecture, no node has access to both the full content, User identity, and Service identity at the same time.

Specifically: i. S₁ and S₂ knows User, but not Service; ii. S₃ knows Service but not User; iii. S₄ knows neither User nor Service; iv. H knows only Service. Additionally, no node can know the entire route, regardless of its computing power. User remains the only one who controls and knows the full route. As for the traffic content, no single node can access all of it since it is mostly protected by fragmentation or asymmetrization: i. S₁ and S₂ only see snowflakes; ii. S₃ only sees upload traffic; iii. S₄ route only snowflakes; iv. H only has access to download traffic.

5. Evaluation

5.1. Resistance against threat actors

5.1.1. Passive adversaries

Polynomially bounded global passive adversary (GPA). In the bounded GPA case, VIPN provides sender anonymity. The attacker might know a user is connected to a node since it knows all the nodes, but it cannot track how the user is anonymized because it cannot follow the traffic. Indeed, it cannot break the TLS encryption between nodes or access the routing table to trace the traffic. To reduce sender-recipient unlinkability, the attacker could theoretically monitor inbound and outbound traffic. However, the traffic is asymmetrized, so the attacker cannot distinguish users. Regarding secrecy, the TLS encryption is unbreakable by this attacker, so the messages remain safe within the overlay. As for resistance to transparent MITM attacks, the user is passive and cannot modify the packets. Finally, the protocol does not offer sender unobservability by default. However, if the user chooses to send dummy packets, the attacker cannot tell them apart from real packets, as the connection between users and nodes is TLS-protected.

5.1.2. Active adversaries

Local polynomially bounded and unbounded attacker. As a reminder, VIPN offers unique features compared to other anonymization networks. Besides ensuring that no node knows both the User and the Service of a specific session at the same time, it also prevents a local attacker from knowing if a specific user is contacting a specific service. This guarantees sender-recipient unlinkability. VIPN secures both the content and the operation: i. regardless of computing power, nodes S_1 , S_2 , S_4 , and any observer on these circuits cannot access the content; ii. even if S_3 and H see User traffic, they can only access upstream or downstream flows, preventing a MITM attack. Thus, against a local corrupt node, VIPN is transparent MITM resistant for each node's role. Secrecy is maintained in all cases, as no node has access to both upstream and downstream flows. Even with unbounded nodes, snowflakes preserve secrecy due to the one-time pad encryption. Regarding sender unobservability, it is ensured in all cases because nodes only see snowflakes when they know the user, and due to the OTP encryption, they cannot decrypt the packets. Finally, sender anonymity is preserved in all cases, as no node can know how the user is anonymized. P_{ss} know the users but not H , while P_{ss} and H know the H but not the associated users.

Polynomially bounded global active adversary (GAA). If the attacker has deployed probes on all network elements but cannot decrypt TLS, they will not be able to recombine the snowflakes, keeping the secrecy intact. Additionally, the attacker cannot actively modify the traffic, even if they intercept and hold back all the data, maintaining VIPN's transparent MITM resistance. However, the attacker can still analyze traffic by adding delays to the snowflakes, which could break both sender anonymity and sender-receiver unlinkability. Adding delays is more complex than in traditional anonymity networks because the complexity increases with the traffic of other snowflakes. Active attacks on one path will not affect the exit traffic on another path, as the paths have different lengths. Also, the attacker

cannot watermark the snowflakes since they cannot access the content to generate a fake snowflake (snowflakes are protected by TLS 1.3). As a result, we consider our proposal partially resistant to this type of attack on sender anonymity and sender-receiver unlinkability. Finally, regarding sender unobservability, we are in the GPA case where the protocol does not offer it by default. However, if the user decides to send dummy packets, the adversary cannot tell them apart from real ones, since the connection between users and nodes is protected by TLS.

Polynomially unbounded global active adversary. If the attacker has deployed probes on all network elements and can decrypt TLS, they can access the snowflakes and, being unbounded, gain access to the full message. From this message, they can also determine the number of circuits, allowing them to break all anonymity properties, including sender anonymity, sender-recipient unlinkability, and sender unobservability. Regarding secrecy, the attacker can access and decrypt the message if it is encrypted. To manipulate the traffic and break transparent MITM resistance, the attacker could theoretically synchronize their clock to make the modified traffic intelligible.

5.1.3. Summary

In Table 1, we summarized the guarantees offered by VIPN against the studied threat actors. VIPN proves to be robust and effectively maintains the desired properties against both passive and local adversaries. However, it is not fully resistant to global active adversaries, although the bounded one is blocked in many cases due to the architecture. This remains acceptable as these adversaries would need to: i. deploy probes across the overlay, requiring collaboration or infiltration in foreign countries; ii. store all overlay traffic, which requires significant storage capacity; iii. decrypt the traffic, even in real time, if the attack is time-sensitive; iv. recombine the right snowflakes, which requires substantial computing power (the recombination is a combinatorial problem that follows a power law based on the number of snowflakes in the overlay), and there is no guarantee that the elements will be coherent due to OTP encryption. Finally, the theoretical approach does not account for the time needed to break these guarantees.

Table 1
Summary of VIPN's guarantees against different adversaries.

Adversaries				Properties				
				Sender anonymity	Sender-recipient unlinkability	Sender unobservability	Secrecy	Transparent MITM resistant
Passive adversary	GPA			X	X	X*	X	X
Active adversary	Corruption power							
	Polynomially bounded	Local	Pus	X	X	X	X	X
			Master	X	X	X	X	X
			Slave	X	X	X	X	X
			Holonode	X	X	X	X	X
		Global		~	~	X*	X	X
	Polynomially Unbounded	Local	Pus	X	X	X	X	X
			Master	X	X	X	X	X
			Slave	X	X	X	X	X
			Holonode	X	X	X	X	X
		Global		-	-	-	-	-

X means that the property is verified.

X* means that the property is not offered by default but is verified if the user sends useless packets.

~ means that the property is partially verified: it is verified on the basis of an uncalculated minimum quantity of network traffic processed by the adversary.

Since global active adversaries are not fully realistic at present, especially with polynomially unbounded corruption power, this initial evaluation will be followed by a future paper. The upcoming paper will explore an additional adversary capable of corrupting only a partial number of nodes. This will help demonstrate that if the user trusts a certain percentage of nodes in their route, the guarantees remain intact, rather than succumbing to the effects of a GAA. The future work will also investigate the time needed for a bounded GAA to execute an attack.

5.2. Early experiences: VIPN in the Wild

VIPN is live on public Internet since early 2022. This public deployment includes global validation of the possibility to deploy collaborative spoofing over several cloud operators (i.e.: spoof and receive answers); this was quite challenging because by default operators blocks the spoofing in order to prevent malicious usage of their IPs. For VIPN the spoofed IPs are part of the overlay.

As of late August 2025, VIPN is deployed over 45 nodes in Europe across 11 countries and more are added each month depending on users' demand and code maturity (for comparison Tor, created 20 years ago, has currently around 8000 relays and bridges but started with 32 relays). Each node is currently connected with at least 300Mbps/300Mbps links, and one half of the nodes on 10 Gbps/10 Gbps. The number of users or sessions on nodes varies and is difficult to monitor as we do not want to create hints for any observer but VIPN currently has over 1500 subscribers, i.e. professional or individual user with an active account.

Users reported to use VIPN for various purposes such as web browsing, obtaining dynamic fixed IP list, access to TOR, connecting to their WebRTC applications, access censored content from foreign countries, etc. The average speed reported with fiber is around 100 Mbps, with peaks at 250 Mbps. For example to reach a Netherlands IP over Ethernet (via France and the UK), latency is about 40 ms. In lab tests, launching VIPN locally without any physical routing latency, resulted in a 2 ms increase compared to the direct route. In comparison, a dedicated OpenVPN session showed an added 31 ms of latency. This latency contributes to the bandwidth limitations, which are also affected by the current code version, node capacities, and the use of TCP over TLS 1.3, which is limited to 10-20 Gbps on standard computers. Since November 2023 and a 1000% increase in users, we have faced no abuse issues and users have reported only few captchas and blocked websites, alongside those with predefined geographic limitations.

5.3. Other limitations

Several additional risks have been identified that could weaken the system's anonymity.

- *Traffic analysis*: the system theoretically remains vulnerable to traffic analysis, even in the presence of collaborative spoofing, particularly if adversaries control a significant portion of the nodes or communication flows. To mitigate this threat, we consider integrating: mechanisms inspired by Loopix, such as cover traffic; integrate variable padding in snowflakes; leveraging the possibility to split parallelly traffic across multiples routes; as well as developing agent with node capabilities to hide if traffic is related to this user or only relaying others' traffic.
- *Denial-of-service*: the risk of denial-of-service attacks, especially through the exposure of IP addresses associated with active nodes is existing due to the centralize nodes database.

To address this, we intend to leverage architectures based on random walks and distributed hash tables (DHTs), explore entry proxy-based solutions such as Tor Snowflake and bridge, and introduce nodes with restricted dissemination capabilities.

- *Fingerprints*: even if the traffic is caught before going to the overlay (intermediary nodes are seeing snowflakes), some networking indicators such as the TTL or TCP options may fingerprint the traffic to an observer as they are affected by the spoofing mechanism. Even if this is probably limited, a dedicated study on this subject will help to give insights on how the traffic must be modified (either at user or at Ps_master to be coherent before going through the overlay or before leaving it (e.g.: based on information related to the distance between Ps_master and H to the Service)).

6. Future Work

Firstly, VIPN proves robustness against passive and local threats but only partially resists GAA, whose attacks are limited by practical constraints. Future work will focus on adversaries corrupting only some nodes, showing that guarantees hold if enough nodes remain trusted, and will evaluate attack feasibility over time. While VIPN improves resistance to traffic alteration, it still relies on TTP at systemic level, which is a study accepted limitation: i. as the directories managing node or user access remain centralized, they could be manipulated to mislead users or launch attacks. A solution is to implement an anonymity-preserving access control system but also the decentralization; ii. if node operators are not running the official VIPN protocol and code, vulnerabilities could be introduced to undermine invisibility. Additionally, providing users with a clear overview of the overlay status to help them make informed decisions when defining their route without compromising their anonymity or the anonymity of other users is a complex challenge. This requires developing metrics based on parameters like network heterogeneity and route duration, which must be disclosed while maintaining privacy. Furthermore, the current work does not offer a formal analysis of the protocol; the properties are only proven through analysis of the architecture models. Therefore, if the implementation or protocol messages are flawed, attacks remain possible. Finally, building an anonymity network involves more than only protocol and architecture. Today, there are various ways to detect and block traffic from proxies, based on factors like communication reactivity, IP reputation, IP provider, etc. As a result, large-scale deployment of VIPN needs also to address these IP marking challenges.

Acknowledgments

This work was supported by the France 2030 cybersecurity acceleration strategy.

Declaration on Generative AI

During the preparation of this work, the authors used Le Chat and ChatGPT-4 in order to: Grammar and spelling check, Text Translation, Improve writing style, Paraphrase and reword. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] F. Shirazi, M. Simeonovski, M. R. Asghar, M. Backes, C. Diaz, A Survey on Routing in Anonymous Communication Protocols, *ACM Computing Surveys* 51 (2018) 51:1–51:39. doi:10.1145/3182658.
- [2] D. L. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, *Communications of the ACM* 24 (1981) 84–90. doi:10.1145/358549.358563.
- [3] R. Dingledine, N. Mathewson, P. Syverson, Tor: The Second-Generation Onion Router, Technical Report, Defense Technical Information Center, Fort Belvoir, VA, 2004. doi:10.21236/ADA465464.
- [4] D. Goldschlag, M. Reed, P. Syverson, Onion routing, *Communications of the ACM* 42 (1999) 39–41. doi:10.1145/293411.293443.
- [5] D. Chaum, The dining cryptographers problem: Unconditional sender and recipient untraceability, *Journal of Cryptology* 1 (1988) 65–75. doi:10.1007/BF00206326.
- [6] B. Zantout, R. Haraty, I2P data communication system, in: *Proceedings of ICN*, Citeseer, 2011, pp. 401–409.
- [7] R. Sherwood, B. Bhattacharjee, A. Srinivasan, P5: A protocol for scalable anonymous communication, *Journal of Computer Security* 13 (2005) 839–876. doi:10.3233/JCS-2005-13602.
- [8] R. Shokri, N. Yazdani, A. Khonsari, Chain-Based Anonymous Routing for Wireless Ad Hoc Networks, in: *2007 4th IEEE Consumer Communications and Networking Conference*, IEEE, Las Vegas, NV, USA, 2007, pp. 297–302. doi:10.1109/CCNC.2007.65.
- [9] A. Pfitzmann, M. Köhntopp, Anonymity, Unobservability, and Pseudonymity – A Proposal for Terminology, in: H. Federrath (Ed.), *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability* Berkeley, CA, USA, July 25–26, 2000 *Proceedings*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2001, pp. 1–9. doi:10.1007/3-540-44702-4_1.
- [10] J. van den Hooff, D. Lazar, M. Zaharia, N. Zeldovich, Vuvuzela: Scalable private messaging resistant to traffic analysis, in: *Proceedings of the 25th Symposium on Operating Systems Principles, SOSP '15*, Association for Computing Machinery, New York, NY, USA, 2015, pp. 137–152. doi:10.1145/2815400.2815417.
- [11] N. Tyagi, Y. Gilad, D. Leung, M. Zaharia, N. Zeldovich, Stadium: A Distributed Metadata-Private Messaging System, in: *Proceedings of the 26th Symposium on Operating Systems Principles, SOSP '17*, Association for Computing Machinery, New York, NY, USA, 2017, pp. 423–440. doi:10.1145/3132747.3132783.
- [12] N. Gelernter, A. Herzberg, H. Leibowitz, Two Cents for Strong Anonymity: The Anonymous Post-office Protocol, in: S. Capkun, S. S. M. Chow (Eds.), *Cryptology and Network Security*, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2018, pp. 390–412. doi:10.1007/978-3-030-02641-7_18.
- [13] A. Kwon, H. Corrigan-Gibbs, S. Devadas, B. Ford, Atom: Horizontally Scaling Strong Anonymity, 2017. doi:10.48550/arXiv.1612.07841. arXiv:1612.07841.
- [14] A. Kwon, D. Lazar, S. Devadas, B. Ford, Riffle: An efficient communication system with strong anonymity, *Proc. Priv. Enhancing Technol.* (2016) 115–134. doi:10.1515/POPETS-2016-0008.

- [15] D. Chaum, D. Das, F. Javani, A. Kate, A. Krasnova, J. de Ruiter, A. T. Sherman, cmix: Mixing with minimal real-time asymmetric cryptographic operations, in: D. Gollmann, A. Miyaji, H. Kikuchi (Eds.), *Applied Cryptography and Network Security - 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings*, volume 10355 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 557–578. doi:10.1007/978-3-319-61204-1_28.
- [16] A. M. Piotrowska, J. Hayes, T. Elahi, S. Meiser, G. Danezis, The loopix anonymity system, in: 26th USENIX Security Symposium (USENIX Security 17), USENIX Association, Vancouver, BC, 2017, pp. 1199–1216.
- [17] C. Diaz, H. Halpin, A. Kiayias, The Nym Network, White Paper, Nym Technologies, 2021.
- [18] I. Clarke, S. Miller, T. Hong, O. Sandberg, B. Wiley, Protecting free expression online with Freenet, *IEEE Internet Computing* 6 (2002) 40–49. doi:10.1109/4236.978368.
- [19] K. Bennett, C. Grothoff, GAP—practical anonymous networking, in: *International Workshop on Privacy Enhancing Technologies*, Springer, 2003, pp. 141–160.
- [20] Q. Wang, N. Borisov, Octopus: A Secure and Anonymous DHT Lookup, in: 2012 IEEE 32nd International Conference on Distributed Computing Systems, IEEE, Macau, China, 2012, pp. 325–334. doi:10.1109/ICDCS.2012.78.
- [21] H. Corrigan-Gibbs, B. Ford, Dissent: Accountable anonymous group messaging, in: *Proceedings of the 17th ACM Conference on Computer and Communications Security - CCS '10*, ACM Press, Chicago, Illinois, USA, 2010, p. 340. doi:10.1145/1866307.1866346.
- [22] P. Maymounkov, D. Mazières, Kademia: A Peer-to-Peer Information System Based on the XOR Metric, in: P. Druschel, F. Kaashoek, A. Rowstron (Eds.), *Peer-to-Peer Systems, Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2002, pp. 53–65. doi:10.1007/3-540-45748-8_5.
- [23] L. Basyoni, N. Fetais, A. Erbad, A. Mohamed, M. Guizani, Traffic Analysis Attacks on Tor: A Survey, in: 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), 2020, pp. 183–188. doi:10.1109/ICIOT48696.2020.9089497.
- [24] P. Winter, R. Köwer, M. Mulazzani, M. Huber, S. Schrittwieser, S. Lindskog, E. Weippl, Spoiled Onions: Exposing Malicious Tor Exit Relays, in: E. De Cristofaro, S. J. Murdoch (Eds.), *Privacy Enhancing Technologies*, Springer International Publishing, Cham, 2014, pp. 304–331. doi:10.1007/978-3-319-08506-7_16.
- [25] G. Danezis, I. Goldberg, Sphinx: A Compact and Provably Secure Mix Format, in: 2009 30th IEEE Symposium on Security and Privacy, 2009, pp. 269–282. doi:10.1109/SP.2009.15.
- [26] C. Kuhn, M. Beck, T. Strufe, Breaking and (Partially) Fixing Provably Secure Onion Routing, in: 2020 IEEE Symposium on Security and Privacy (SP), 2020, pp. 168–185. doi:10.1109/SP40000.2020.00039.
- [27] H. Ballani, P. Francis, X. Zhang, A study of prefix hijacking and interception in the internet, *ACM SIGCOMM Computer Communication Review* 37 (2007) 265–276. doi:10.1145/1282427.1282411.
- [28] M. Backes, A. Kate, P. Manoharan, S. Meiser, E. Mohammadi, AnoA: A Framework for Analyzing Anonymous Communication Protocols, in: 2013 IEEE 26th Computer Security Foundations Symposium, IEEE, New Orleans, LA, 2013, pp. 163–178. doi:10.1109/CSF.2013.18.