DOI: 10.18372/2310-5461.68.20240
UDC 004.77:004.056

Maksym Kotov, PhD Student
Taras Shevchenko National University of Kyiv,
Kyiv, Ukraine
https://orcid.org/0000-0003-1153-3198
e-mail: maksym_kotov@ukr.net;

Serhii Toliupa, DSc (Engin.), Prof.
Taras Shevchenko National University of Kyiv,
Kyiv, Ukraine
https://orcid.org/0000-0002-1919-9174
e-mail: tolupa@i.ua;

GROUP STATE ROUTING PROTOCOL (MYCELIA): A NEW MODEL FOR DISTRIBUTED NETWORK COORDINATION

Introduction

In a post-quantum technological development, it becomes increasingly important to provide significant guarantees of communication secrecy and integrity through the public infrastructure [1]. At the same time, a continuously growing demand for such services leads to a correlating need for scalability and management of large traffic volumes. Hence, it is becoming increasingly complex to manage and maintain mechanisms for coordinating the secure communication channels, which minimize probabilities of successful cryptographic breach and at the same time incur minimal operational overhead of security measures.

Generally, most modern solutions for anonymized communication over public networks leverage cryptographic capabilities to secure the communication channel and provide guarantees of confidentiality and integrity. While, at the same time, availability is generally achieved through the distributed nature of networking systems. When a single node fails, other relay members could substitute its computational capabilities when requested. The problem arises due to potential cryptographic breaches, either through cryptoanalysis efforts or security misconfigurations.

There are ongoing efforts to both improve the cryptographic standards in the face of new unauthorized decryption methods and improve the topological construction of communication in such a way that minimizes the probability of continuous interception of the traffic and its later analysis. The reduction of cryptographic windows becomes a more relevant research topic for such systems. Thus, different models for traffic relay and retranslation have been proposed that disperse packages between nodes in a large network.

The recent tendency for decentralized solutions has influenced the research focus for the organization of public services. Centralized providers require trust to operate and could become a critical component in security architecture. It is not unprecedented for such entities to introduce policies influenced by local governments that compromise the anonymity of users or data integrity. Additionally, the central responsible entity could become the target of a coordinated attack, which could leverage authoritative permissions in the system [2, 3].

Hence, it becomes more popular to transition services into a decentralized environment such as blockchain to reduce responsibility and improve the availability characteristics of services. Here, a blockchain network is a group of nodes maintaining a common ledger of transactions or other records in a consistent manner through the process of general consensus between network participants. Such capabilities are often used to organize a directory service for available retranslation nodes and their associated metadata, such as latencies, availability, geographic location, and fees for used services. This greatly improves durability due to redundancy with mirrored records.

Analysis of recent research and publications

Traditional systems of building Virtual Private Networks (VPNs) tend to work as centralized services managed by a third party. In such a case, a proxy node on the provider's side performs retranslation of the incoming traffic to the target, making the appearance of a mediator between the initiator and the target node. This allows, in a simplistic way, to hide the destination of the outgoing traffic as well as its origin from the recipient. Though, such an approach is vulnerable to analysis since the compromise of the channel is contingent upon a single node [4, 5].

A more advanced approach is to chain multiple relay servers in a single channel, which complicates the analysis effort for the potential sniffer. The most widely known and popular system following such an approach is Tor with its time-bound circuits mechanisms. Now since the entry and outgoing points in the system are separated, the correlation analysis is required and is generally a more complex task, especially since the channel rotates frequently. Though, a single input and output node with a fixed path of hops is the main attack vector. Compromise of edge nodes poses security risks [6–8].

To encrypt information within the relay system, either the Onion or the Garlic scheme is used. With Onion, each transit node removes the encryption level before transmitting packets to the next node. Let m be the payload transmitted through the tunnel, to which are added headers $h_1, \dots, h_n$ and which are associated with symmetric keys $k_1, \dots, k_n$. Then the Onion encryption can be described as [9]:

$$L_n = \mathrm{Enc}_{k_n}(h_n \parallel m),$$
$$L_i = \mathrm{Enc}_{k_i}(h_i \parallel L_{i+1}), (i = n-1, \dots, 1),$$

where the ciphertext is O = L_1, and decryption is performed in the reverse order [9]:

$$(h_i \parallel L_{i+1}) = \mathrm{Dec}_{k_i} L_i, (i = 1, \dots, n-1),$$
$$(h_n \parallel m) = \mathrm{Dec}_{k_n}(L_n).$$

This approach maintains confidentiality while relying on intermediate relay nodes in a secure communication.

As for Garlic routing, a group of messages, called “cloves”, are combined into a single ciphertext “garlic”. Let m be the payload transmitted through the tunnel, and the external AES key be K_0. Each clove $c_j = (h_j, m_j)$ is associated with its own key K_j. Then Garlic encryption can be described as [10,11]:

$$C_j = \mathrm{Enc}_{K_j}(h_j \parallel m_j),$$
$$G = \mathrm{Enc}_{K_0}(H \parallel [C_1, C_2, \dots, C_t]),$$

where H is the header with settings, ciphertext is G, and decryption is performed in the reverse order as follows [10,11]:

$$(H \parallel [C_1, C_2, \dots, C_t]) = \mathrm{Dec}_{K_0}(G),$$
$$(h_j \parallel m_j) = \mathrm{Dec}_{K_j}(C_j), (j = 1, \dots, t).$$

Unlike Onion, this approach allows for distributed routing of individual packets which reduces the chain compromise risks but complicates nesting of layers.

Now, the mesh-like topology leveraged in some systems avoids dependency on intermediary chain nodes. Every node in such a network could be simultaneously connected to multiple others and retranslate the traffic in multiple directions to avoid any critical dependencies on a single service provider. Such mechanisms tend to be more complex in coordination efforts but generally scale better and reduce the impact of intermediary nodes' compromise. The entry and out-nodes are generally fixed and remain the critical points. To reduce the likelihood of successful traffic correlation analysis, different techniques such as delay randomization are utilized on edge nodes. Such techniques increase the latencies of communication and do not guarantee the protection against analysis [12, 13].

With the growing demand for decentralization, many traffic retranslation systems began to integrate with the public blockchain networks as the basis for economic interactions and consensus [2, 3]. The blockchain networks hold directories of available nodes, their processing capabilities, and fees associated with services. By using techniques such as zero-knowledge proofs, it is possible to achieve anonymous yet publicly verifiable interactions between multiple parties without any prerequisites for trust [14, 15]. Examples of such technologies are Nym and Orchid protocols aimed at decentralized coordination and management of secure communication channels [16–18].

The purpose of the article

With the limitations in current protocols discussed previously, a new approach is needed to reduce cryptographic windows and attack vectors for malicious actors. The purpose of this article is to develop a new model of building secure communication channels that aims at improving confidentiality characteristics of the held communication. The model and methods used to maintain such a connection have to meet a few conditions related to the handling of the relay mechanisms.

Firstly, the model needs to lessen reliance on the virtual network's outer nodes and distribute the outgoing traffic. Such a mechanism will significantly complicate the correlation analysis effort since more physical channels will have to be monitored or controlled. Secondly, with distribution and decentralization in mind, the newly proposed protocol should avoid authoritative concentrations of coordination to provide the scalability required for contemporary traffic volumes as well as mitigate security risks.

Finally, the protocol must maintain complete functionality, consistency, and coherence with the existing public network infrastructure. The dispersion of the traffic should appear seamless for the initiator and recipient to support the myriads of existing modern network protocols. Consideration for such support has to be made at the transport layer of the OSI model to guarantee operational stability for the application layer.

Path routing architecture

The protocol operates in discrete epochs $t \in \mathbb{N}$ specific to the entity in context, be it user, cluster or relay node. The epochs are configurable and allow entities to define secrets or channel rotation periods. Here, relay nodes represent a finite set V of functional workers capable of traffic retranslation within a secure channel.

These nodes are grouped into local clusters, where $\mathcal{C} = \{C_1, \dots, C_m\}$ with $C_i \subset V$ and $\bigcup_{i=1}^{m} C_i = V$. These clusters could overlap, so some nodes could be simultaneously joined to multiple secluded organizations if such subnetworks do not forbid multi-membership. Formally, $C_i \cap C_j \neq \emptyset$ may hold for some $i \neq j$. For $v \in V$, its membership set is $M(v) = C \in \mathcal{C}: v \in C$, where it is possible that $|M(v)| > 1$.

Within each epoch t , the following directed multigraph applies to describe the connectivity:

$$G_t = (V, E_t), \qquad E_t \subseteq V \times V.$$

For any $C \in \mathcal{C}$, the induced subgraph $G_t[C]$ is the intra-cluster fabric for C, representing prearranged communication links.

A node that is simultaneously connected to multiple clusters $C \in \mathcal{C}$ operates independently within an epoch t for each cluster. That is, such connections are cluster-agnostic, within each subnetwork the node would maintain its own communication secrets, tasks, and channels.

The edges between the clusters may not be set up initially, there is no requirement for clusters to maintain stable connections, at least in this version of the protocol.

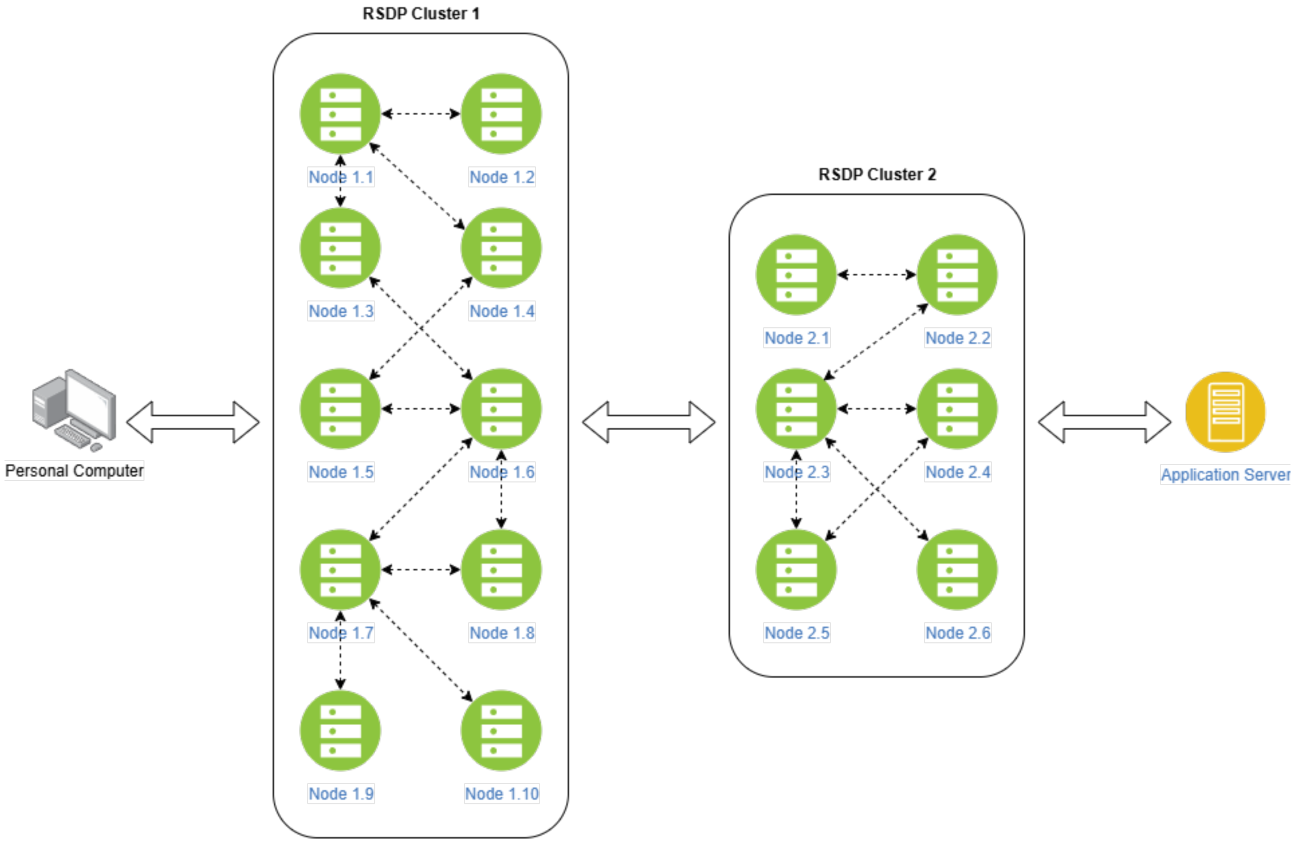

Fig. 1. The GSRP inter-cluster routing schema

The connections between clusters could be cached if they were established previously. Withing a cryptographic space, a few core building blocks are leveraged to organize the protocol. Firstly, the KEM (Key Encapsulation Mechanism) [9,19]:

$$(k, c) \leftarrow \mathrm{Encaps}(\mathrm{PK}), \qquad k \leftarrow \mathrm{Decaps}(\mathrm{SK}, c).$$

The purpose of such mechanism is to produce a symmetric key k to be used to encrypt cyphertext c with IND-CCA guarantees. Such guarantees mean that the potential attack would not be able to tell which text was encrypted even if he can query the space of plaintexts [19].

Next component is AEAD, which stands for Authenticated Encryption with Associated Data. Its responsibility is to perform an encryption of message m alongside with the metadata and headers required for operational readiness [19].

It could be formally expressed as:

$$\mathrm{Enc}_k(\mathrm{AD}, m), \qquad \mathrm{Dec}_k(\mathrm{AD}, c),$$

and encrypts m with k as well as the required metadata, where key schedule:

$$k' = \mathrm{KS}(k, \mathrm{ctx}),$$

which takes an exiting key k, some context and deterministically derives subkeys.

We treat such components as black boxes; the article does not aim at creating a new cryptographic or key derivation algorithm. Its purpose is to construct a protocol which uses existing cryptographic methods to build secure communication channels which minimize impact of compromised nodes.

Each node v produces a static identity key used for message control, membership authentication, and coordination participation within a cluster.

With that, each node also maintains temporary keys used for establishing ephemeral session within epoch t:

$$\left(\mathrm{PK}_v^{\mathrm{kem}}(t), \quad \mathrm{SK}_v^{\mathrm{kem}}(t)\right),$$

which captures cryptographic state of every node, with long-living identity key and temporal session keys used for establishing channels.

Each cluster $C \in \mathcal{C}$ and epoch t operates in two possible modes, which could be represented by a parameter $\mu_C \in \{private, public\}$. Both share the same probabilistic internal routing but differ in how they handle internal and external communication.

Regardless of its mode, they operate internally based on a transition kernel:

$$P_C(t) \in \Delta(C \times C), \qquad \sum_{w \in C} P_C\,(t)[u, w] = 1,$$

where $P_C(t)$ must be aperiodic and irreducible on its support, which means it should provide an ergodic random walk within C with unique stationary distribution $\pi_C(t)$. Hence, after enough random walks, the location of messages, that is, the time messages spend on a specific node converges to $\pi_C(t)$ regardless of an incoming node.

Now, the private cluster maintains a shared security architecture which allows to greatly improve the trust both internally and externally, as well as optimize the construction of internal communication channels.

Each private cluster C, through the process of BFT-compliant consensus process based on RSDP [20–22] (which will be discussed later), maintains a shared secret key:

$$s_C(t) \in \{0,1\}^\lambda,$$

where λ is also chosen through the consensus process.

Using $s_C(t)$, members can deterministically derive keys to be used for internal communication:

$$k_v^{\mathrm{hop}}(t) = \mathrm{KS}(s_C(t), v \parallel \text{“hop”}),$$
$$k_{(u,v)}^{\mathrm{link}}(t) = \mathrm{KS}(s_C(t), u \parallel v \parallel \text{“link”}),$$

where keys are synchronized, forward secure, and do not require pairwise negotiation, greatly improving the latencies during consensus rounds or epoch transitions.

With public cluster it is infeasible to establish a shared secret key since it could be easily leaked by untrusted nodes. If the secret key is not shared, its advantage diminishes in terms of trust and performance from both intra-cluster perspective and potential clients.

Hence, each node $v \in C$ maintains its own per-epoch KEM keypair:

$$\left(\mathrm{PK}_{v \in C}^{\mathrm{kem}}(t), \quad \mathrm{SK}_{v \in C}^{\mathrm{kem}}(t)\right),$$

where links between nodes are negotiated individually. In such case RSDP would provide the general view of the transition kernel $P_C(t)$, and a map of public keys without a shared secret or key derivation mechanism.

Public clusters guarantee topological consistency but not the shared security mechanism. It is easier to grow the number of nodes and incentivize sharing of computational power, but they are generally less secure and incur larger latencies during internal negotiations.

Moving on to the description of general model of traffic forwarding, let as first define a set of user nodes $\mathcal{U}$ disjoint from V. Let $A \in \mathcal{U}$ ne a client node sending message m to the application server. The task of A is to generate (H, Q), where H – set of headers, and Q – encrypted payload.

The client should first make topological decisions regarding ingress $I \subseteq V$ and egress $O \subseteq V$ sets of nodes. Here, ingress nodes are the ones receiving messages from the client and egress nodes – the ones facing the application server (the exit nodes). Each of these sets must be large enough, having at least k nodes each to prevent trivial deanonymization. Formally: $|I|, |O| \geq k$.

Then the client builds a nested onion structure using respective public keys of chosen nodes for the I and O. There is an important distinction between the nodes belonging to public and private clusters. With private structures a client can use the cluster public key instead of individual keys for every node, which greatly simplifies decisions and computational complexity related to negotiations.

The application of encryption is done in a reversed order. First, the client uses public keys of egress nodes and then the public keys of ingress nodes. This guarantees that no intermediary cluster or relay node can see the payload before it reaches the end of the secure channel.

After the encryption is done, the message follows a sequence of transition through nodes and clusters, following the stochastic kernels defined previously.

The network states:

$$\mathcal{S} = \bigcup_{C \in \mathcal{C}} \mathrm{C} \cup \{\mathrm{A}, \mathrm{Srv}\},$$

where Srv is the destination (the application server), and a message at discrete time step $\tau \in \mathbb{N}$ is located at a state $S_\tau \in \mathcal{S}$. With this we can define the transition as a random process over network states.

At initiation time:

$$S_0 = A, \qquad M_0 = (H, Q),$$

where M_τ denotes the current ciphertext payload carried by the message.

The client then selects the ingress node to send the package according to dispatch distribution:

$$\iota_t \in \Delta(I).$$

Then the initial transmission event:

$$S_1 \sim \iota_t, \qquad M_1 = M_0.$$

If $S_\tau = x \in I$ for some ingress node, then:

1. The node locates its encapsulation $c_x^{\mathrm{in}} \in H$;
2. Derives $k_x^{\mathrm{in}} = \mathrm{Decaps}\left(\mathrm{SK}_x^{\mathrm{kem}}(t), c_x^{\mathrm{in}}\right)$;
3. Computes the partially decrypted ciphertext $M_{\tau+1} = \mathrm{Dec}_{k_x^{\mathrm{in}}}(\mathrm{AD}, M_\tau)$, if the decryption succeeds or otherwise it drops the message.

After peeling x, the ingress node chooses the next intra-cluster recipient by using the transition kernel defined earlier:

$$S_{\tau+1} \sim P_{C(x)}(t)[x, \cdot],$$

where $C(x)$ denotes the cluster containing x, and if x belongs to multiple C, $C(x)$ is either chosen by user preference or sampled uniformly from cluster membership set $M(x)$.

The intra-cluster propagation operator:

$$\Phi_{u \to v}^{\mathrm{intra}}(t)(M) =$$
$$\mathrm{Enc}_{k_{(u,v)}^{\mathrm{link}}(t)}\left(\mathrm{AD}, \mathrm{Enc}_{k_u^{\mathrm{hop}}(t)}(\mathrm{AD} \parallel u \parallel v, M)\right),$$

Then the random evolution of message content inside cluster C is:

$$M_{\tau+1} = \Phi_{S_\tau \to S_{\tau+1}}^{\mathrm{intra}}(t)(M_\tau), \;\; S_{\tau+1} \sim P_C(t)[S_\tau, \cdot],$$

which defines a time-homogeneous Markov chain on C with transition kernel $P_C(t)$ [23].

Moving on to the inter-cluster communication, if $S_\tau = x \in C_i$ and the next node is outside C_i, x chooses a target cluster $C_j \neq C_i$ and a bridge ingress set $I' \subseteq C_j$ according to a cluster-level kernel:

$$Q_{i \to j}(t) \in \Delta\left(C_i \times C_j\right).$$

For each $y \in I'$, if a bridge key exists in the cache $\left((x, y), b\right) \in \mathcal{B}_{i \to j}(t)$, the transmitted ciphertext is:

$$M_{\tau+1} = \mathrm{Enc}_{k_{i \to j}^{\mathrm{bridge}}(t)}(\mathrm{AD}, M_\tau), \;\; S_{\tau+1} = y.$$

If the bridge was not established previously, a new key pair must be initiated between the cluster nodes via KEM and stored in $\mathcal{B}_{i \to j}(t)$. Maintaining long-term connections is possible and optimizes choices when multiple inner clusters are used.

When $S_\tau = y \in O$ for some egress node, and the next node is Srv, it performs:

$$k_x^{\mathrm{eg}} = \mathrm{Decaps}\left(\mathrm{SK}_y^{\mathrm{kem}}(t), c_y^{\mathrm{eg}}\right),$$
$$M_{\tau+1} = \mathrm{Dec}_{k_y^{\mathrm{eg}}}(\mathrm{AD}, M_\tau).$$

Then:

$$M_{\tau+1} = m, \qquad S_{\tau+1} = \mathrm{Srv},$$

and the message reaches its destination.

The transition is a discrete-time random process:

$$(\, S_\tau, M_\tau)_{\tau \geq 0},$$

on the state space $\mathcal{S} \times \mathcal{M}$, where $\mathcal{M}$ is the cyphertext space, governed by transition operator:

$$\mathbb{T}_t(u, M; v, M') =$$
$$\begin{cases} f_{\mathrm{intra}}^{\mathrm{tr}}, & v \in C(u), \\ f_{\mathrm{inter}}^{\mathrm{tr}}, & v \in C_j, \\ \iota_t[v] \cdot 1\{\mathrm{M}' = \mathrm{M}\}, & u = A, \\ 1\{\mathrm{v} = \mathrm{Srv}, \mathrm{M}' = \mathrm{m}\}, & u \in O. \end{cases}$$

where:

$$f_{\mathrm{intra}}^{\mathrm{tr}} = P_{C(u)}(t)[u, v] \cdot 1\{M' = \Phi_{u \to v}^{\mathrm{intra}}(t)(M)\},$$

and:

$$f_{\mathrm{inter}}^{\mathrm{tr}} = Q_{i \to j}(t)[u, v] \cdot$$
$$1\{M' = \mathrm{Enc}_{k_{i \to j}^{\mathrm{bridge}}(t)}(\mathrm{AD}, M_\tau).$$

This operator defines a Markov decision process with cryptographic state updates [23].

Dispersion of the outer nodes

After discussing the routing principles for the outgoing traffic, we should note how the outgoing traffic shuffles and the responses from the application server get retranslated back to the original client.

For $v \in V$, let $A_v \subseteq \mathbb{A}$ be the set of external address-port pairs usable by v. Each overlay flow is associated with an identifier $\theta \in \Theta$ used to keep the backwards retranslation mapping $R_v(t) \colon \Theta \to \mathcal{R}$.

Now, the handling is divided on the transport layer of the OSI model between the sessionless (UDP) protocols and stateful protocols (TCP). The retrainslation must work seamlessly for the application server, and the protocol should not expect any additional integration from the outside perspective.

For UDP, each node v maintains its local retranslation table:

$$U_v(t) \subseteq \Theta \times A_v \times \mathbb{A} \times \mathcal{R} \times \mathbb{N}, \qquad \left(\theta, a, d, \rho, \tau_{\mathrm{exp}}\right),$$

where θ is the overlay flow identifier, a is external source address/port from A_v, $d \in \mathbb{A}$ is the destination server address, $\rho \in \mathcal{R}$ is the return path and τ_{exp} is the expiration time for the entry to drop stale unused records. If v must send a UDP packet for flow θ to destination d and no mapping exists for (θ, d), then it should choose $a \in A_v$, τ_{exp} and insert them into the memory as $\left(\theta, a, d, \rho, \tau_{\mathrm{exp}}\right)$.

Upon receiving response from the application server, the same tuple is used to find out the backwards retranslation path ρ and propagate the message. The response follows the encryption schema similar to the one we've defined for the ingress data propagation.

The expiration of records is important to avoid memory bloat over time and is defined as follows:

$$U_v(t) \leftarrow \{ (\theta, a, d, \rho, \tau_{\text{exp}}) \in U_v(t) : t \le \tau_{\text{exp}} \}.$$

So, the stale records should be automatically recycled and the memory reclaimed.

Now, TCP is connection-oriented, meaning that during a single session it should appear as coming from the same source, otherwise it will get reset, which would either break the connection or greatly increase latencies due to constant re-establishment of sessions.

Therefore, a cluster must maintain a single façade node for the duration of a TCP session. For that, each cluster C maintains a table:

$$T_{C(t)} \subseteq \Theta \times C,$$

whose tuples are:

$$(\theta, z^\star),$$

where $z^\star \in C$ is the facade node for flow θ.

The façade node $z^\star$ maintains local tables:

$$S_{z^\star}(t)[\theta] = a^\star \in A_{z^\star},$$
$$R_{z^\star}(t)[\theta] = \rho_\theta \in \mathcal{R},$$

which respectively store the external address used for the session and the return path for that session.

When any node $v \in C$ needs to open a TCP connection for flow θ to an application server d:

1. If $(\theta, z^\star) \notin T_{C(t)}$, select a facade node for retranslation $z^\star = F_C(\theta) \in C$;
2. Insert $(\theta, z^\star)$ into $T_{C(t)}$;
3. At $z^\star$, allocate $a^\star \leftarrow \text{Choose}(A_{z^\star})$, $S_{z^\star}(t)[\theta] = a^\star$, $R_{z^\star}(t)[\theta] = \rho_\theta$.

With those tables, the egress node can effectively handle retranslations.

When node $v \in C$ sends a segment for flow θ to destination d, it looks up $(\theta, z^\star) \in T_{C(t)}$ from the cluster-wide shared state to find out the mapping of the responsible façade node. Then, before sending out the peeled message into the public network to the destination, it replaces its own address from the IP package as well as the port from the datagram to match the facade node's configuration.

Upon receipt of a FIN/RST segment or expiration at $t > \tau_{\text{exp}}$:

$$T_C(t) \leftarrow T_C(t) \setminus \{(\theta, z^\star)\},$$
$$S_{z^\star}(t) \leftarrow S_{z^\star}(t) \setminus \{\theta\},$$
$$R_{z^\star}(t) \leftarrow R_{z^\star}(t) \setminus \{\theta\},$$

reclaiming the resources allocated for the session.

Comparison with existing protocols

To highlight the improvements this protocol introduces, we will compare the influence attackers have over communication channels between multiple existing methods, including Tor, I2P, and Nym. The purpose is to show the probability of a successful of traffic analysis or service degradation within each analyzed methods [24–26].

For each protocol we assume a set of active relays V and $B \subseteq V$ representing a set of adversarial nodes. Then $f = |B|/|V|$ is a compromise rate which also represents a probability of a chosen node being adversarial.

Now, for Tor protocol, a circuit (G, M, E) is fixed for the duration of a session, where G is the guard node, M is the middle node, and E is the exit node. All traffic goes through the same nodes until the circuit switches either due to client's request or an automatic expiration after a preconfigured amount of time. Tor utilizes Onion routing for its operations [24, 25].

For a successful correlation, the attacker must control the entry and exit nodes:

$$P_{\text{corr}}^{\text{Tor}} = \Pr(G, E \in B) = f^2,$$

and the fraction of a flow that the attacker can observe and analyze is:

$$\Psi_{\text{Tor}} = 1\{E \in B\},$$
$$\mathbb{E}[\Psi_{\text{Tor}}] = f,$$
$$\text{Var}(\Psi_{\text{Tor}}) = f(1 - f).$$

That means that if $E \in B$, the attacker can analyze the entire flow during the lifetime of the circuit.

With I2P, communication uses two simplex channels for inbound and outbound traffic. Additionally, this protocol leverages the Garlic routing schema, allowing for sending messages with multiple instructions and destinations as opposed to Tor's model [25].

Let's denote the corresponding edge nodes as G_{out} and G_{in}. Then the probability of a successful correlation attack is as follows:

$$P_{\text{corr}}^{\text{I2P}} = \Pr(G_{\text{out}}, G_{\text{in}} \in B) = f^2,$$

where fraction of traffic observed per one channel direction:

$$\Psi_{\text{I2P,out}} = 1\{G_{\text{out}} \in B\},$$
$$\Psi_{\text{I2P,in}} = 1\{G_{\text{in}} \in B\},$$

meaning that with I2P, each tunnel's traffic is either completely traceable or fully hidden:

$$\mathbb{E}[\Psi_{\text{I2P,in}}] = \mathbb{E}[\Psi_{\text{I2P,out}}] = f.$$

Thus, each direction leaks with probability f, and the combined correlation is f^2.

In that regard, Tor and I2P provide comparable resistance against tracing but I2P provides stronger anonymity guarantees.

One of the primary distinctions between Tor and I2P is that the latter was designed to establish secure communication between peers that both support and are aware of the protocol, while Tor was designed as a public proxy service that anonymizes the traffic source.

Nym is a stratified Mixnet, that does not operate on the basis of prearranged circuits. Each package within the network could travel different routes, and transition between L layers is done with random timing skews, reducing the possibility of a successful correlation analysis [26].

To directly link the input and output throughout all layers, each one must be compromised:

$$P_{\text{corr}}^{\text{Nym}} \leq f^L + (1 - f^L) \cdot \kappa \cdot \varepsilon_{\text{mix}},$$

where:

1. f^L is the probability that all layers are compromised;
2. ε_{mix} is the residual matching probability per batch;
3. $\kappa \in (0,1)$ represents the cover-traffic attenuation factor.

Thus, the probability of a successful correlation analysis falls exponentially with the size of L.

The problem is that it still uses single ingress G_{in} and egress G_{out} nodes, meaning the fraction of traffic that could be observed in case of compromise:

$$\Psi_{\text{Nym,out}} = 1\{G_{\text{out}} \in B\},$$

meaning that in case if edge node gets compromised, the visibility of traffic:

$$\mathbb{E}[\Psi_{\text{Nym,out}}] = f,$$

which is similar to Tor and I2P.

Finally, within GSRP, each flow uses multiple ingress n_I and egress n_O nodes chosen dynamically. For UDP packages are distributed randomly among the egress nodes which also handle the responses, while for TCP, responses are handled by one egress nodes within a single TCP session.

Let $X_i \sim \text{Bern}(\beta_i)$ mark whether egress i is compromised, with traffic share w_i, where $\sum w_i = 1$. Then the effective adversarial mass is:

$$\beta = \sum \beta_i w_i .$$

The adversary's total observed traffic share:

$$\Phi_N = \sum_{i=1}^{n_O} X_i F_i ,$$

where F_i is the empirical fraction of packets sent via egress i.

Then:

$$\mathbb{E}[\Phi_N] = \sum_{i=1}^{n_O} \beta_i w_i = \beta.$$

That means that the expected portion of visible traffic does not depend on the amount of chosen egress nodes and rather depends on the portion of compromised nodes. When increasing the amount of egress nodes, the portion of traffic visible by each compromised node decreases.

If all egresses were compromised, in uniform case:

$$\beta_i = \beta,$$

$$w_i = \frac{1}{n_O},$$

$$\Pr(\Phi_N = 1) = \beta^{n_O}.$$

Which means that the probability decays exponentially with the number of egresses in contrast to previously analyzed protocols like Tor, I2P, and Nym where $\Pr(\Phi_N = 1) = f$.

To analyze tail bounds, let $K = \sum_i X_i$ denote number of compromised egresses. Then for any threshold $\theta > \beta$ [27]:

$$\Pr\,(\Phi_N \geq \theta) \leq \exp\,(-n_O, D_{\text{KL}}(\theta \parallel \beta)),$$

where D_{KL} is the binary Kullback–Leibler divergence.

As the number of egress nodes grows, the chances that the adversary nodes control large portions of traffic drop exponentially.

Conclusions

The advancements in computational, analytical, and post-quantum capabilities introduce new challenges for building secure and stable connections over public infrastructure. With existing protocols, it is quite common to have traffic concentration points, which increases the likelihood of a successful interception and correlation analysis. Randomized delays help in stifling such efforts but do not completely obstruct them. Such tendencies have led to an increasing demand for methods aimed at improving confidentiality characteristics of communication systems. Hence, a novel approach to managing secure connections has been proposed and evaluated within the scope of this article.

Firstly, a new path routing architecture has been proposed based on relay groups rather than individual nodes, which further improves the meshing capabilities of the network. Multiple organization modes for such groups, including public and private, provide their respective tradeoffs between decentralization and traffic dispersion capabilities. Inherently, additional encryption sessions and relay hops through the network imply additional latencies. Hence, for most purposes, relaying through two levels of groups would suffice for meshing while minimizing the growth of latencies. Evaluating the increase in latency is an area for further empirical research.

Within the developed protocol, a new approach towards outgoing traffic from a coordinated network has been proposed to reduce the likelihood of correlation attacks. These methods scope attention to the transport layer of the OSI model. Sessions are considered only for TCP-based transmissions while providing a complete fragmentation of the flow for sessionless protocols like UDP. Techniques such as targeted sender replacement have been leveraged to disperse the outgoing traffic from the network while preserving the functional coherence and completeness of classical network communication.

Overall, the purpose of this article is to spark further research into the building and managing secure communication channels over public infrastructure. The provided mathematical models and descriptions of the protocol aim to simplify the integration process in contemporary infrastructure and workflows.

REFERENCES

[1] Barrett-danes F., Ahmad F. Quantum computing and cybersecurity: a rigorous systematic review of emerging threats, post-quantum solutions, and research directions (2019–2024). *Discover Applied Sciences*. 2025. Vol. 7, No. 10. P. 1083. DOI: 10.1007/s42452-025-07322-5.

[2] Bhutta M.N.M., Khwaja A.A., Nadeem A., Ahmad H.F., Khan M.K., Hanif M.A., Song H., Alshamari M., Cao Y. A Survey on Blockchain Technology: Evolution, Architecture and Security. *IEEE Access*. 2021. Vol. 9. P. 61048–61073. DOI: 10.1109/ACCESS.2021.3072849.

[3] Alghamdi T.A., Khalid R., Javaid N. A Survey of Blockchain Based Systems: Scalability Issues and Solutions, Applications and Future Challenges. *IEEE Access*. 2024. Vol. 12. P. 79626–79651. DOI: 10.1109/ACCESS.2024.3408868.

[4] Qollakaj K., Larsson L.E., Memeti S. Cybersecurity of remote work migration: A study on the VPN security landscape post Covid-19 outbreak. *Array*. 2025. Vol. 27. P. 100437. DOI: 10.1016/j.array.2025.100437.

[5] Sudipti Banerjee., Dr. Rajni Ranjan Singh Makwana. Virtual Private Network: Survey and Research Challenges. *International Journal of Latest Technology in Engineering Management & Applied Science*. 2025. Vol. 14, No. 6. P. 211–226. DOI: 10.51583/IJLTEMAS.2025.140600028.

[6] Alsabah M., Goldberg I. Performance and Security Improvements for Tor: A Survey. *ACM Computing Surveys*. 2017. Vol. 49, No. 2. P. 1–36. DOI: 10.1145/2946802.

[7] Karunanayake I., Ahmed N., Malaney R., Islam R., Jha S.K. De-Anonymisation Attacks on Tor: A Survey. *IEEE Communications Surveys & Tutorials*. 2021. Vol. 23, No. 4. P. 2324–2350. DOI: 10.1109/COMST.2021.3093615.

[8] Cui J., Huang C., Meng H., Wei R. Tor network anonymity evaluation based on node anonymity. *Cybersecurity*. 2023. Vol. 6, No. 1. P. 55. DOI: 10.1186/s42400-023-00191-8.

[9] Reed M.G., Syverson P.F., Goldschlag D.M. Anonymous Connections and Onion Routing. *IEEE Journal on Selected Areas in Communications*. 1998. Vol. 16, No. 4. P. 482–494. DOI: 10.1109/49.668972.

[10] S. M., Thangavel G., Basheer S. A Review on Garlic Routing and Artificial Intelligence Applications in Public Network. *2023 International Conference on Computer Science and Emerging Technologies (CSET)*: 2023 International Conference on Computer Science and Emerging Technologies (CSET). Bangalore, India: IEEE, 2023. P. 1–6. DOI: 10.1109/CSET58993.2023.10346642.

[11] Jadav N.K., Gupta R., Tanwar S. Garlic Routing-Based Privacy Preserving Framework for Secure Data Exchange Between IoMVs with 5G. *2023 International Conference on Network, Multimedia and Information Technology (NMITCON)*: 2023 International Conference on Network, Multimedia and Information Technology (NMITCON). Bengaluru, India: IEEE, 2023. P. 1–6. DOI: 10.1109/NMITCON58196.2023.10276185.

[12] Kjorveziroski V., Bernad C., Gilly K., Filiposka S. Full-mesh VPN performance evaluation for a secure edge-cloud continuum. *Software: Practice and Experience*. 2024. Vol. 54, No. 8. P. 1543–1564. DOI: 10.1002/spe.3329.

[13] Subratie K., Aditya S., Figueiredo R.J. EdgeVPN: Self-organizing layer-2 virtual edge networks. *Future* Generation *Computer Systems*. 2023. Vol. 140. P. 104–116. DOI: 10.1016/j.future.2022.10.007.

[14] Sun X., Yu F.R., Zhang P., Sun Z., Xie W., Peng X. A Survey on Zero-Knowledge Proof in Blockchain. *IEEE* Network. 2021. Vol. 35, No. 4. P. 198–205. DOI: 10.1109/MNET.011.2000473.

[15] Capko D., Vukmirovic S., Nedic N. State of the Art of Zero-Knowledge Proofs in Blockchain. *2022 30th* Telecommunications *Forum (TELFOR)*: 2022 30th Telecommunications Forum (TELFOR). Belgrade, Serbia: IEEE, 2022. P. 1–4. DOI: 10.1109/TELFOR56187.2022.9983760.

[16] Halpin H. Nym Credentials: Privacy-Preserving Decentralized Identity with Blockchains. *2020 Crypto Valley Conference on Blockchain Technology (CVCBT)*: 2020 Crypto Valley Conference on Blockchain Technology (CVCBT). Rotkreuz, Switzerland: IEEE, 2020. P. 56–67. DOI: 10.1109/CVCBT50464.2020.00010.

[17] Danezis G., Goldberg I. Sphinx: A Compact and Provably Secure Mix Format. *2009 30th IEEE* Symposium *on Security and Privacy (SP)*: 2009 30th IEEE Symposium on Security and Privacy.

Oakland, CA, USA: IEEE, 2009. P. 269–282. DOI: 10.1109/SP.2009.15.

[18] Halpin H. Simulation of Mixmining Reward Parameters for the Nym Mixnet. *Ubiquitous Security*. / ed. by Wang G., Wang H., Min G., Georgalas N., Meng W. Singapore: Springer Nature Singapore, 2024. Vol. 2034. P. 363–379. (Communications in Computer and Information Science). DOI: 10.1007/978-981-97-1274-8_24.

[19] Alwen J., Blanchet B., Hauck E., Kiltz E., Lipp B., Riepel D. Analysing the HPKE Standard. *Advances in Cryptology – EUROCRYPT 2021*. / ed. by Canteaut A., Standaert F.-X. Cham: Springer International Publishing, 2021. Vol. 12696. P. 87–116. (Lecture Notes in Computer Science). DOI: 10.1007/978-3-030-77870-5_4.

[20] Kotov M., Toliupa S. Layered Architecture for RSDP V3.0: Modular Distributed Consensus and Coordination. Networks *and Sustainability*. / ed. by Luntovskyy A., Klymash M., Melnyk I., Beshley M., Gütter D. Cham: Springer Nature Switzerland, 2025. Vol. 1473. P. 140–160. (Lecture Notes in Electrical Engineering). DOI: 10.1007/978-3-032-02272-1_6.

[21] Toliupa S., Kotov M., Buchyk S., Boiko J., Shtanenko S. Stateful Cluster Leader Failover Models and Methods Based on Replica State Discovery Protocol. *Information Technology and Implementation Workshop (IT&I-WS 2024: Intelligent Systems and Security)*: Proceedings of the Information Technology and Implementation (IT&I) Workshop: Intelligent Systems and Security (IT&I-WS 2024: ISS), Kyiv, Ukraine, November 20 – 21, 2024. CEUR-WS.org, 2024. Vol. 3933. P. 120–140. URL: https://ceur-ws.org/Vol-3933/Paper_10.pdf.

[22] Kotov M., Toliupa S., Nakonechnyi V., Buchyk S., Buchyk O. Byzantine Fault Tolerance in Distributed Systems: Advancing the Replica State Discovery Protocol v2. 0. CEUR-WS.org, 2025. Vol. 3991. P. 121–138. URL: https://ceur-ws.org/Vol-3991/paper10.pdf.

[23] Kaplan D. An overview of Markov chain methods for the study of stage-sequential developmental processes. *Developmental Psychology*. 2008. Vol. 44, No. 2. P. 457–467. DOI: 10.1037/0012-1649.44.2.457.

[24] Imani M., Amirabadi M., Wright M. Modified relay selection and circuit selection for faster Tor. *IET Communications*. 2019. Vol. 13, No. 17. P. 2723–2734. DOI: 10.1049/iet-com.2018.5591.

[25] Ali A., Khan M., Saddique M., Pirzada U., Zohaib M., Ahmad I., Debnath N. TOR vs I2P: A Comparative Study. *2016 IEEE International Conference on Industrial Technology (ICIT)*: 2016 IEEE International Conference on Industrial Technology (ICIT). Taipei, Taiwan: IEEE, 2016. P. 1748–1751. DOI: 10.1109/ICIT.2016.7475027.

[26] Halpin H. Nym Credentials: Privacy-Preserving Decentralized Identity with Blockchains. *2020 Crypto* Valley *Conference on Blockchain Technology (CVCBT)*: 2020 Crypto Valley Conference on Blockchain Technology (CVCBT). Rotkreuz, Switzerland: IEEE, 2020. P. 56–67. DOI: 10.1109/CVCBT50464.2020.00010.

[27] Janson S. Large Deviation Inequalities for Sums of Indicator Variables. arXiv, 2016. DOI: 10.48550/ARXIV.1609.00533.

Котов М. С., Толюпа С. В.

GROUP STATE ROUTING PROTOCOL (MYCELIA): НОВА МОДЕЛЬ КООРДИНАЦІЇ РОЗПОДІЛЕНОЇ МЕРЕЖІ

У даній статті описується нова модель управління безпечними каналами зв'язку в розподіленому середовищі з метою мінімізації криптографічних вікон та векторів атак на основі аналізу кореляції трафіку. У сучасному світі цифрового взаємозв'язку, здатність зберігати конфіденційність обміну даними між кількома вузлами набуває відчутного значення. Безпечний канал зв'язку повинен забезпечувати значущі гарантії конфіденційності, цілісності та доступності даних, що передаються, а також мінімізувати затримки через накладні витрати механізмів безпеки.

Існуючі рішення, зосереджені навколо віртуальних приватних мереж, сітчастих топологій та ретрансляцій, мають спільні обмеження у своїй здатності уникати точок концентрації трафіку, таких як вхідні та публічні вузли. Таким чином, криптоаналітики можуть корелювати трафік у деяких випадках та потенційно порушувати секретність зв'язку. Отже, пропонується нова модель, метою якої є уникнення появи таких точок концентрації під час передачі трафіку шляхом використання можливостей групової маршрутизації та децентралізованої координації стану кластера на основі протоколу виявлення стану репліки (Replica State Discovery Protocol).

У рамках цієї статті розроблено та представлено нову модель побудови безпечних каналів зв'язку. Було проведено порівняльний аналіз для ілюстрації різниці в топологіях зв'язку між існуючими рішеннями та запропонованим протоколом. В основі цього дослідження лежить теоретичний аналіз топології запропонованої моделі та методів, що використовуються для оптимізації створення безпечних каналів та механізмів ретрансляції трафіку. Математичне моделювання було використано для формального опису властивостей запропонованого протоколу, що сприятиме його інтеграції в сучасні системи зв'язку.

В результаті було розроблено та проаналізовано модель розподіленого управління безпечними каналами зв'язку, яка зменшує ймовірність атак на основі кореляції трафіку та таким чином покращує характеристики конфіденційності з'єднання між учасниками. Було запропоновано новий метод маршрутизації шляхів на основі груп ретрансляції, а також принципи розсіювання вихідного трафіку mesh-мережі. По-друге, було розроблено метод розподіленої координації трафіку на основі RSDP, який дозволяє уникнути концентрації повноважень і таким чином закладає основу для децентралізованої версії побудови безпечних каналів зв'язку.

Ключові слова: захищені канали зв'язку, мережеві протоколи, шифрування, інциденти безпеки, розподілені системи, децентралізовані системи, координація розподілених мереж, ранжування трафіку на основі RSDP.

Kotov M., Toliupa S.

GROUP STATE ROUTING PROTOCOL (MYCELIA): A NEW MODEL FOR DISTRIBUTED NETWORK COORDINATION

The following article is describing a new model of managing secure communication channels in a distributed environment with the aim of minimizing cryptographic windows and attack vectors based on traffic correlation analysis. In the contemporary world of digital interconnection, having the ability to preserve the confidentiality of the data exchange between multiple peers gains significant importance. A secure communication channel must provide meaningful guarantees of confidentiality, integrity, and availability of the data that is being transmitted as well as ensure minimized latencies due to security mechanism overhead.

Existing solutions centered around Virtual Private Networks, mesh-like topologies, and relays share limitations in their ability to avoid traffic concentration points such as entry and public-facing nodes. Thus, it is possible for cryptanalysts to correlate traffic in some cases and potentially compromise the secrecy of communication. Hence, the proposed model aims to avoid such concentration points during traffic transmission by leveraging the capabilities of group routing and decentralized cluster state coordination based on the Replica State Discovery Protocol.

A comparative analysis has been conducted to illustrate the difference in communication topologies between existing solutions and the proposed protocol. At the basis of this research is a theoretical analysis of the proposed model's topology and methods used to facilitate the secure channels' creation and traffic relay mechanisms. Mathematical modeling has been leveraged to formally describe the properties of the proposed protocol to further facilitate its integration into the modern communication systems.

As a result, a model for a distributed management of secure communication channels has been developed and analyzed that reduces the likelihood of attacks based on traffic correlation and thus improves the confidentiality characteristics of the connection between participating peers. A new path routing method based on relay groups has been proposed as well as the traffic dispersion principles related to the outgoing traffic from the mesh network. Secondly, a method of distributed traffic dispersion and coordination has been developed based on the RSDP that allows for avoiding concentration of authority and thus lays the foundation for the decentralized version of building secure communication channels.

Keywords: secure communication channels, network protocols, encryption, security incidents, distributed systems, decentralized systems, distributed network coordination, traffic meshing with RSDP.

Стаття надійшла до редакції 12.11.2025 р.
Прийнято до друку 10.12.2025 р.