TORIUM Technical Whitepaper

TORIUM Labs LLC, Memetic Block

Abstract

The Internet began as an open communications network to enable peer-to-peer interconnectivity. However, in the decades since its inception, life has digitized and the majority of internet users have become reliant on a shrinking pool of Internet Service Providers (ISPs) who monitor every search request. The growth of cloud computing, pushing servers into an obscured but virtual setting, and the rise of big data processing by the largest search engines, all together create an environment where internet use is far from private or anonymous.

The Tor network is a widely-used tool for anonymous communication that allows users to browse the internet and access online resources without revealing their identity. Tor has enabled censorship resistance and anonymity for a number of movements, including countless journalists and whistleblowers and, indeed,cryptocurrency itself. However, despite its popularity,the Tor network has faced several challenges hindering mainstream adoption, including insufficient relaysto obscure large traffic, insufficient exit and bridge re-lays, vulnerability to DDoS and malicious exit nodes. In this paper, we present an initiative to promote a moreresilient infrastructure for Tor and mainstream adoption by internet users. We propose a novel recognition mechanism for Tor relays, facilitated by blockchaintechnology and the TORIUM currency, to reward contributors to Tor in an anonymous and equitable way. We then explore with greater detail the implementation of this initiative and mechanisms withinTORIUM to keep the protocol sustainable.

1 Introduction

The concept of anonymous routing dates back to the early days of the internet [1] when users first began to realize the potential privacy risks associated with online activity. One of the earliest solutions to this problem was the introduction of onion routing, a technique that allows users to transmit data anonymously by routing it through a series of encrypted nodes.

Onion routing was first introduced in the mid-1990s by researchers at the United States Naval Research Laboratory, who were looking for a way to protect online communication from surveillance and interception. The basic idea behind onion routing is to encrypt data multiple times, removing each layer of encryption as the data passes through a series of relays or nodes. This process makes it virtually impossible for anyone to trace the data back to its source, as each relay only knows the identity of the previous and next nodes in the chain.

The need for onion routing and other forms of anonymous routing has only grown in recent years, as concerns about online privacy and government surveillance have become increasingly prominent. With the rise of mass surveillance programs and the increasing amount of personal information being shared online, individuals are turning to tools like the Tor network to protect their online privacy and maintain their anonymity.

However, as the Tor network has grown in popularity, it has also faced a number of challenges, including a lack of incentives for individuals to run nodes. Some of the problems is undoubtedly sociological: most people do not feel the need to protect their privacy that way; this is one reason that companies such as Zero Knowledge Systems [2] and Digi cash [3] failed. An-other reason is that remaining anonymous requires the trust of many parties, something that is almost impossible in such a physically distributed system as the internet. There has been previous research trying to tackle the problem, however, they have largely failed due to unanticipated factors at implementation.

This paper acknowledges the mistakes of previous research and combines newer technologies such as the blockchain to create a new recognition-based mechanism for Tor contributors. TORIUM essentially providesa system in which users are distributed rewards in recognition of their contribution to the Tor Network,while keeping the most productive relays obscured.

By further advancing and supporting the Tor ecosystem and network, we create innovative ways tomake it main-stream in a world where privacy issues affect everyone. Furthermore, our approach will address several real-world pain points, including in countries such as Ukraine where political unrest and censorship are rife. By integrating blockchain technology into the Tor net-work, we can ensure that it remains secure and robust, providing a more reliable and incentivized solution foranonymous communication in the digital age.

Note: This whitepaper serves as an evolving document that reflects our continuous development efforts. We are committed to keeping this document up-to-date with the most recent advancements in our project. Due to the dynamic and iterative nature of our development process, it is expected that the final code and implementation may vary from what is currently presented in this paper.

We invite the interested reader to peruse our GitHub repository at

https://github.com/torium-development

2 The Tor Network

To understand the TORIUM Protocol, we must first comprehend the Tor Network. Tor, short for The Onion Router, is software that enables anonymous communication on the Internet. It works by routing internet traffic through a series of servers (nodes) located around the world in a way that makes it difficult to trace the origin of the traffic.

When a user connects to Tor, their data is encrypted and then passed through a selected series of Tor nodes, each of which decrypts a layer of the data, exposing the address of the next node in the chain. This process continues until the data reaches its final destination. Each node in the chain only knows the address of the previous and next nodes, so it's very difficult for any adversary in the middle to trace the data back to its source.

Additionally, Tor users further enhance their anonymity due to the way services are provided on the network i.e some websites and services are only accessible through the Tor network. This, in essence, allows users and services to benefit from Tor on both ends. These services use a different routing mechanism, in which the website's address is encrypted and passed through several nodes before reaching its destination, making it very difficult to locate the server hosting the website. Figure 1 provides a simplistic view of the network which the viewer may refer to.

2.1 Encryption within the Tor Network

Tor uses several different encryption methods to protect the privacy and security of its users' data as it passes through the network [4].

Suppose Alice wants to send a message to Bob over the Tor network. The message will pass through multiple Tor nodes (relays) before reaching Bob. Let's call these nodes R1, R2, and R3. Each node has its own public and private key pair.

1. Symmetric Encryption:
   Alice first establishes a symmetric encryption key with each Tor node in the circuit. This is done using a key exchange protocol. Let's call these shared keys K_{A_R1}, K_{A_R2}, and K_{A_R3}.
   Alice encrypts her message using symmetric encryption with K_{A_R3}, then with K_{A_R2}, and finally with K_{A_R1}. This is known as ”onion encryption” because it resembles the layers of an onion.
2. Public-Key Cryptography for Routing:
   Now, Alice needs to create the routing information for each Tor node to know where to send the data next. She creates the routing information for R3, which includes the next hop (Bob) and any necessary metadata. Alice encrypts this routing information using R3's public key.
   Alice then creates the routing information for R2, which includes the next hop (R3) and any neces-

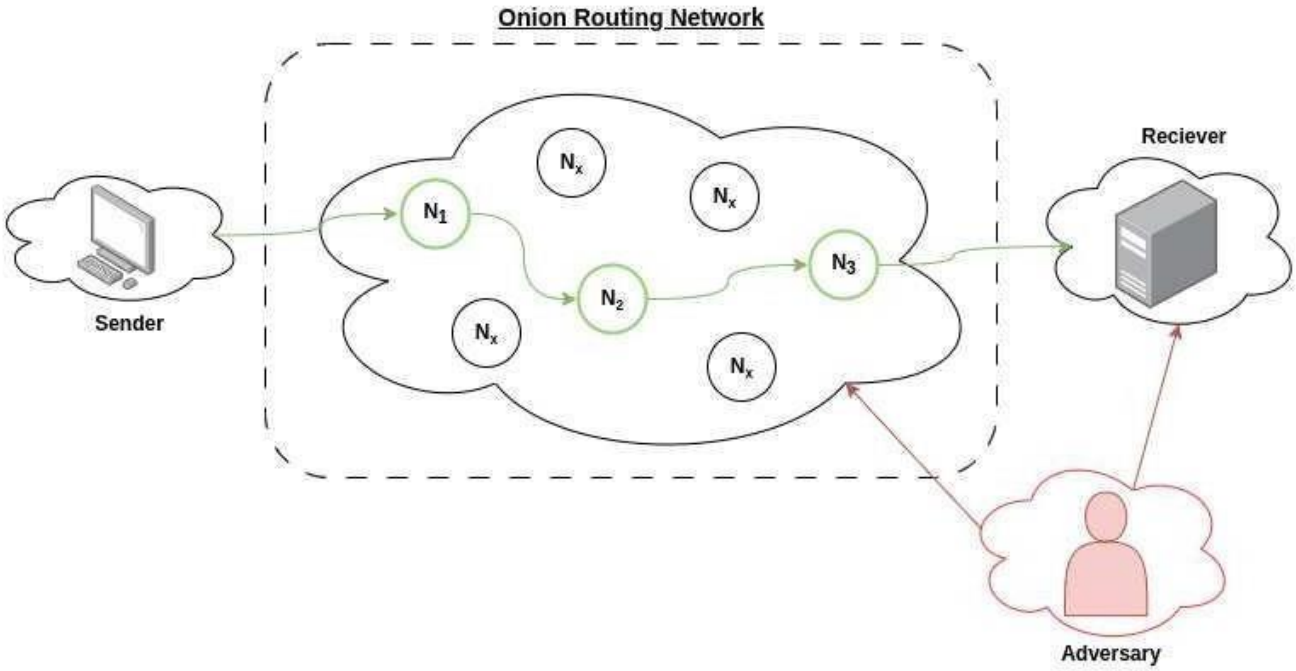

Fig. 1. Example diagram of the Tor Network

sary metadata. She appends the encrypted routing information for R3 and encrypts the entire package using R2’s public key.

Finally, Alice creates the routing information for R1, which includes the next hop (R2) and any necessary metadata. She appends the encrypted package for R2 and encrypts the entire package using R1’s public key.

3. Data Transmission:

   Alice sends the encrypted message and routing information to R1. R1 uses its private key to decrypt the routing information and forwards the encrypted message and remaining routing information to R2. R2 uses its private key to decrypt the routing information and forwards the encrypted message and remaining routing information to R3.

   R3 uses its private key to decrypt the routing information and forwards the encrypted message to Bob.

4. Message Decryption:

   Bob receives the encrypted message from R3. Since he shares the symmetric encryption key K_{A_R3} with Alice, he can decrypt the message and read its content.

In this example, symmetric encryption protects the message content, while public-key cryptography secures the routing information. By combining these two types of encryption, Tor ensures that the message is securely transmitted, and each relay only knows the previous and next hops in the circuit, preserving Alice’s and Bob’s privacy.

3 Previous Research

We refer to previous systems that also attempted to improve the infrastructure and compatibility of the Tor Networks. However all of these fall short in some ways. For instance, Franz et al, [5]have utilized a blind signature electronic cash model to incentivize mixers to operate with honesty. The method of Franz et al. involves dividing electronic payments and messages into small segments and enabling mixes and users to carry out the exchange incrementally, which resulted in a system that was highly inefficient. Additionally, the recipient is compelled to partake in the payment process, which is not desirable as the receiver may lack knowledge or interest in the Tor network.

Tsuen-Wan et al [6] proposed a method to bring incentives into Tor, not via payment but through the support given to the network. The proposed incentive system rewards high-quality Tor nodes with a symbolic ’gold star’, which gives them a priority for traffic in the network. The gold star status is passed on to maintain high priority for connections, while all other traffic in the network is given low priority. This circuit-based priority system ensures that circuits maintain their priority throughout their lifetime. The main problem with this system, however, is that it highlights the most significant relays within the networks, that adversaries can use to their advantage. Taking down or taking controls of these nodes can pose great threats to the network

Significantly, Andreoulakis et al. [7] published a method to incorporate payments into anonymous routing. The paper introduces the concept of adding extra data to the Tor Packets which are hashes of coins payments and also their receipts. A centralized bank located outside the network would provide these coins to the sender, who would then proceed to transfer the coins individually to the next relay. The next relay would subsequently transfer the coins to the subsequent relay and so forth. In this way and because the encryption system works in the same way the standard Tor mechanism works, each relay only knows of its predecessor and successor. The paper also defines key properties that any anonymous payment system should follow:

1. Sender-receiver unlikability, such that even with the cooperation of a third party and the recipient, no one except a global adversary should be able to link the sender and receiver or reveal the path between them. This is critical for ensuring the anonymity and privacy of users within the network.
2. Usable efficiency, which means that the overhead in the packet exchange for the payment scheme and the computational load with additional cryptographic operations will be reasonable and will not significantly hinder the normal functioning of the system. This is important to ensure that the payment scheme does not adversely impact the overall network’s performance.
3. Accountability, meaning that any node attempting to cheat by forging messages or double-spending coins will be detected and expelled from the network. This property is crucial to ensure the integrity and security of the payment mechanism and the network as a whole.

While in theory, this paper seems to have a rigid protocol that almost cannot be broken, it comes with a single Achilles heel. This is the fact that the bank is centralized and exists outside the network. Crucially, it means that relays must rely on the bank to uphold custody of past rewards, instead of having full ownership of them immediately.

TORIUM aims to address these issues in a multi-pronged solution, presented in the paper.

4 System Overview

Relay Registration

1. Relay owners can register to the TORIUM Protocol. The device registers to the network by proving they own both the BNB Keypair and the Tor Keypair.
2. Utilizing master offline keys enables a system for secure communication between the user and the network.

Relay Recognition

1. Utilizing our *Proof of Uptime* system, the network rewards users who have active relays with a significant uptime in the form of the TORIUM token. Rewards are distributed equally to relays that fulfil the *Proof of Uptime* criteria and this avoids recognition of stronger relays problem.
2. Additional mechanisms are developed to provide bonus rewards for bridge relays, exit relays and relays in certain geographies, through subsidiary recognition wallets

Hardware

1. The TORIUM Relay is preconfigured hard-ware to provide relay services to Tor and receive TORIUM recognition, without user configuration
2. The TORIUM Router is a hotspot to allow consumer devices to connect to Tor

TORIUM End-Uses

1. TORIUM tokens will be integrated throughout the protocol, used as a prerequisite for relay registration, and as the primary means to purchase our hardware.
2. TORIUM can be exchanged for TORIUM Hidden- Services, including decentralized web-hosting via Tor-Arweave integration, and blockchain transaction routing.
3. TORIUM tokens facilitate governance and voting throughout the protocol, expended for proposals.

5 TORIUM

Utilizing TORIUM is simple. A user registers and will receive rewards as long as their relay is running. We keep this as simple as possible for our users to increase accessibility to everyone. However, the backend is much more complex. To a layman, we can defragment TORIUM into simple steps:

5.1 Acquiring Hardware

There are two devices that enable users to easily connect to both the Tor and TORIUM Network:

1. The router, is a hotspot device that connects to Tor. This is NOT an entry node itself but rather is used to connect to entry nodes in the Tor network
2. The relay, a Tor relay that is preconfigured with the signing capabilities to work with our *Proof of Uptime system.*

Technical specifications for both types of hardware will be presented in subsequent papers

5.2 Registration

In this section, we cover both the front and backend of the initial registration process that occurs within TORIUM. To comprehend the user registration process we must first understand the Elliptic Curve encryption process.

5.2.1 Elliptic Curves

Elliptic curves play an essential role in modern cryptography as they form the foundation for Elliptic Curve Cryptography (ECC). ECC is a type of public key cryptography that uses the mathematical properties of elliptic curves over finite fields. Compared to traditional public key cryptography systems like RSA, ECC offers similar security levels with smaller key sizes, resulting in lower computational overhead and reduced storage requirements.

An elliptic curve is defined by an equation of the form:

$$y^2 = x^3 + ax + b$$

BNB uses the secp256k1 elliptic curve for its digital signature algorithm, ECDSA. The secp256k1 curve is defined over a prime field F_p, where p is a large prime number. The curve also has a base point G (generator) with specific coordinates (Gx, Gy), and its order n is a large prime number.

Tor uses the Curve25519 elliptic curve for key exchange, signing, and encryption purposes. Curve25519 is defined over a prime field F_p, where p is a large prime number.

With these differences, we notice that they’re not compatible.

5.2.2 Initial Verification

In the initial verification step, we have to require proof from the user they own both keypairs. TORIUM ac-accomplishes this by signing the public keys of each key pair, i.e. the Tor private key is used to sign the BSC public key, then the BSC private key to sign the Tor public key, and broadcast these signatures in the con- tact field of the server descriptors. We shall go further expand on the topic of relay broadcasting in a later section.

The process is presented mathematically as follows:

Let:

T_{priv} represent the Tor private key
T_{pub} represent the Tor public key
E_{priv} represent the BNB private key
E_{pub} represent the BNB public key

To create a signature for the BNB public key using the Tor private key:

$$\Sigma_{\mathrm{TE}} = \mathrm{Sign}(T_{\mathrm{priv}}, E_{\mathrm{pub}})$$

To create a signature for the Tor public key using the BNB private key:

$$\Sigma_{\mathrm{ET}} = \mathrm{Sign}(E_{\mathrm{priv}}, T_{\mathrm{pub}})$$

Furthermore, users should also be utilizing a ”master offline private key”. [8] The concept of a ”master offline private key” pertains to a long-term private key maintained offline, which is not employed directly for signing messages or data in routine communication. Instead, it serves to sign medium-term keys, subsequently utilized for actual communication. The implementation of a master offline private key aims to augment security by mitigating the likelihood of compromising the long-term private key. We utilize this concept within both the context of the Tor Relay and BNB. Within Tor, This key is responsible for signing the medium- term (onion) keys. In demonstrating ownership of the Tor keypair, the medium-term private key, endorsed by the master offline private key, is used to sign the BNB public key. Within BSC, By generating an of- fline BNB keypair (master offline private key) and utilizing it to sign a medium-term BNB keypair, the medium-term private key may then be employed to sign the Tor public key as a component of the *Proof of Ownership* process.

5.3 Tor Relay Broadcasts

TORIUM continuously retrieves data from relay broadcasts which is used for continuous communication. A Tor relay broadcasts the following information:

Relay descriptor: This includes details about the relay, such as its public IP address, port numbers for the OR (Onion Router) and directory services, platform information (operating system and Tor software version), and the date when the descriptor was generated.

Public keys: The relay shares its public encryption keys (both the long-term ”identity key” and the medium-term ”onion key”) so that clients and other relays can encrypt messages sent to it.

Exit policy: If the relay is an exit node (the last relay in the Tor circuit before reaching the destination), it specifies which types of traffic it allows or disallows to exit the network. This is important because some exit nodes may block certain types of traffic to comply with local laws or to reduce abuse.

Bandwidth and uptime: The relay broadcasts its available bandwidth, recent usage statistics, and uptime. This information helps clients and other relays to make informed decisions when selecting relays to build circuits.

Contact information: Optionally, the relay opertorium may include their contact information, such as an email address, for administrative purposes or to report abuse. However, in TORIUM, this field is a necessity as it will contain the EVM address tied to a user.

Flags: The Tor directory authorities assign flags to the relay based on its characteristics, such as whether it’s a Guard (entry) node, an Exit node, a Fast node, or a Stable node. These flags help clients to choose appropriate relays when building circuits.

Fingerprint: The relay’s unique fingerprint, which is derived from its public identity key, helps identify it within the network.

Consensus Weight: A value assigned to Tor relays (nodes) in the Tor network by the directory authorities. It represents a relative measure of the relay’s contribution to the network based on its bandwidth capacity and other factors. Consensus weight is used by the Tor clients to decide which relays to select when building a circuit for their traffic.

The information broadcasted by a Tor relay is published in the Tor network directory, which is maintained by directory authorities. This information is essential for clients to discover available relays and build circuits for secure and anonymous communication. Furthermoses, the next section delves into how exactly this information is used to establish Tor Relays connected to the TORIUM Network.

5.4 Continued communication

Once the user has registered, the user and network need to have a correct authentication flow that is both secure and fast.

In the proposed authentication schema, API endpoints requiring authentication mandate that users sign the corresponding request. This strategy minimizes the

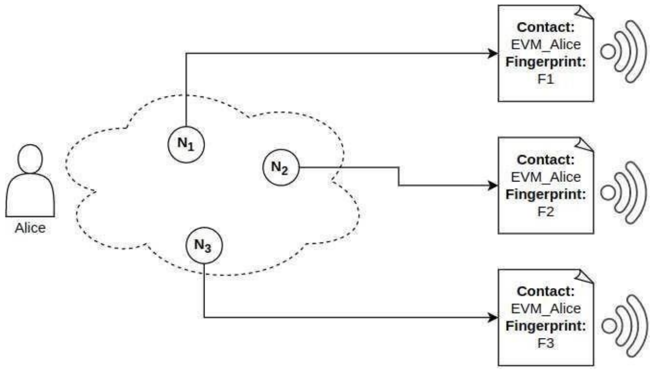

Fig. 2. Alice's Relays

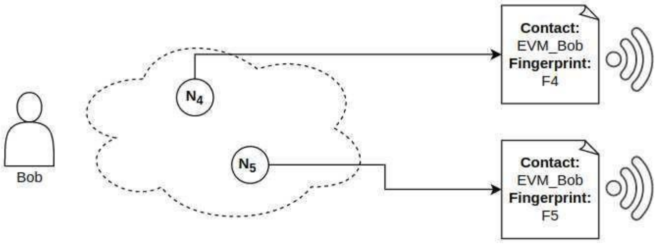

Fig. 3. Bob's Relays

frequency of authentication prompts, thereby fostering a superior user experience compared to other ideas such as JSON Web Tokens (JWTs) for example.

The message structure remains uniform across all authentication endpoints, facilitating the generation of signed ”receipts” that document each user's intended actions. These receipts serve as a record of the user's objectives when interacting with the API.

In the proposed system, the user places their BNB Virtual Machine (EVM) address in the con- tact field of the descriptor associated with their Tor re-lay. This approach enables the user to demonstrate their ownership of both the Tor and BNB keypairs by embedding their EVM address within the Tor relay descriptor. Consequently, when the user signs requests for actions requiring authentication, their EVM address is already included in the descriptor, streamlining the authentication process.

Figure's 2 and 3 may give the reader a simplified and different perspective. Alice and Bob are both user's registered to TORIUM and they both own relays. An EVM Address (broadcasted in the Contact field) represents a user while a Tor fingerprint (broadcasted in the Fingerprint field) represents a Tor relay. The fingerprint is calculated by taking a cryptographic hash of the relay's Ed25519 public key. Specifically, the Tor network uses the SHA-1 hashing algorithm to compute the fingerprint. This system allows users to have multiple relays connected to TORIUM and each relay is still uniquelydistinguishable.

6 Proof of Uptime

Uptime is defined as the duration for which the relay has been continuously operational and accessible within the Tor network. A higher uptime indicates that the relay has been consistently available and functioning without significant interruptions or downtime.

Uptime is an important metric for both clients and directory authorities in the Tor network. Clients consider relay uptime when selecting relays for building circuits, as a higher uptime often implies greater stability and reliability. Directory authorities also use uptime, along with other factors, to assign flags (such as the ”Stable” flag) to relays, which influences the relay selection process for clients.

To create criteria for *Proof of Uptime* we propose using the Consensus Weight field relayed from the server descriptors. This weight as stated before is decided by the Directory authorities who have decided this protocol. Directory authorities are special nodes in the Tor network responsible for collecting information about relays, such as their IP addresses, public keys, and bandwidth capacities. They use this information to create a ”network consensus” document, which is a list of all known relays along with their consensus weights.

The consensus weight is influenced by several factors, including:

1. Relay's bandwidth capacity: Relays with higher bandwidth capacity will generally have a higher consensus weight since they can handle more traffic.
2. Uptime: Relays that have been online and stable for a longer period will likely have a higher consensus weight.
3. Exit policy: Relays that allow more traffic to exit the network (e.g., have a more permissive exit policy) may be assigned a higher consensus weight.

4. Performance measurements: Directory authorities may take into account the relay's performance in terms of latency, reliability, and other metrics when assigning consensus weights.

Using this information, a minimum threshold is able to be decided for a current *Proof of Uptime mechanism*. It is also important to note that the Direct Authority Protocol is centralized. Rather TORIUM takes this approach and creates an alternative decentralized solution.

7 Addressing Security Concerns

7.1 Descriptors

The relay descriptors in the Tor network are not encrypted; however, they are digitally signed to guarantee integrity and authenticity. Relay descriptors contain public information about each relay, which is essential for the Tor network to operate effectively. The public identity key of a relay is used to sign its descriptor, and directory authorities and clients utilizing the relay subsequently verify this signature.

Although relay descriptors are not encrypted, the Tor network is specifically designed to deliver anonymity and privacy for its users. As data is transmitted through the Tor network, it undergoes encryption in layers and passes through a sequence of relays, constituting a circuit. We can think of TORIUM building on top of this system.

Relay descriptors serve the primary function of providing the required information for clients to discover and choose relays when constructing circuits. As they do not carry sensitive information, their content remains unencrypted. Nevertheless, the digital signature we have created affirms the authenticity of the information and ensures that it has not been subjected to tampering.

7.2 Registration

The dashboard will be accessible from Tor and will not require any specific IP, nor will IP be tracked. As opposed to a JWT typically used in web 2.0, each registration action can be signed as it happens, removing a potential vector for compromise. The TORIUM Protocol will create initiatives for existing Tor relay providers to gain the initial TORIUM tokens required to lock and register, as well as the relevant gas token, without needing to link an EVM wallet to an exchange or an identity, and instead continue to accumulate recognition without trace.

8 Recognition Tokenomics

This section outlines the source of TORIUM tokens for relay recognition, for the medium term - where TORIUM circulating supply will remain inflationarywith a view to rapid adoption; and the long term, whereTORIUM will stabilize through its utilities to balance inflow and out-flow of tokens.

8.1 Pre-allocated Rewards

10% of the TORIUM BSC token supply is reservedfor relay rewards. It is vested over two years with weekly unlocks, and will be amortized for three-day periods and sent to qualifying relays in the network. The majority of these tokens will be distributed equally be- tween all relays with verified uptime of 80% (the periodlength and required uptime will be subject to governance decisions) or more, to their registered EVM wal- lets. A proportion of the recognition will be reserved to recognize relays fulfilling additional network-critical functions as outlined in section 7.

Governance will later determine reduction in rewards over longer periods of time in accordance with overall market capitalization, to prolong the inflationary period of TORIUM as necessary.

8.2 Protocol Inflows for TORIUM

TORIUM will flow back into the protocol througha number of mechanisms, to build reserves to allow the rewards to sustain themselves indefinitely. TORIUM must be locked by the BSC wallets whitelisting a relay on signup, checked by a smart contract, which prevents naive DDoS attempts. Whitelisting can also be done on behalf of other wallets, to allow the mass onboarding of existing relay groups or non-technical users with TORIUM hardware. In addition, under the TORIUM governance framework, which will open up certain protocol and reward decisions, TORIUM must bepaid to the protocol to make proposals, and TORIUM holders can vote on proposals, lending decision-making value to the token. Crucially, the router and relay hardware are purchased using the TORIUM token

8.3 Utility Inflows for TORIUM — Provisional

The TORIUM Protocol creates a framework to recognize messages associated with ECDSA key pairs such as BSC, and curve25519 for Tor, Near or Solana, in- Ter operably. This engenders a wealth of opportunities to enable Tor Hidden Services for ECDSA uses, such as BSC transactions and Web3. Products and services built on top of the TORIUM protocol could use the TORIUM token to have messages (or transactions) be recognized. While in its infancy, thepotential for services driven by TORIUM are being explored heavily.

8.4 Utility-Recognition Balance

In the long term, the inflow of TORIUM through utilities, hardware and governance will be exactly balanced by the outflow of recognition rewards, ending the inflationary period for circulating supply and self-sustaining the protocol for the long term.

8.5 Note: Recognition over Rewards

It is important to note that TORIUM is more than a financial or tokenized reward for computation; it rep- resents the broader concept of recognition. The TORIUM team is dedicated to promoting internet anonymity and censorship resistance holistically, providing TORIUM rewards for education, promotion and research in areas where they are most needed.

9 Conclusion

In this paper, TORIUM Labs LLC presents their continually-updated proposal to grow Tor capacity and adoption. This study has undergone an exploration of the Tor network and prior attempts to build incentives on top of that, outlining a multi-pronged solution that can add value to the Tor network immediately and can scale up to encompass it. The cryptographic implications of secure and private relay signup and recognition has been addressed, as well as the incorporation of blockchain through the cryptocurrency TORIUM. While the long-term tokenomics to ensure the longevity of the protocol have been explored, the TORIUM team currently remains focused on expansion and adoption by Tor re-lays, be it with our own hardware users and the wider community dedicated towards internet anonymity.

References

[1] Chaum, D. L., 1981. “Untraceable electronic mail, return addresses, and digital pseudonyms”. *Commun. ACM,* 24(2), feb, p. 84–90.

[2] Back, A., Goldberg, I., and Shostack, A., 2001. “Freedom 2.1 security issues and analysis”.

[3] Chaum, D., 1992. “Achieving electronic privacy”. *Scientific American,* 267(2), pp. 96–101.

[4] Dingledine, R., Mathewson, N., and Syverson, P., 2004. “Tor: The second-generation onion router”. In SSYM’04: Proceedings of the 13th conference on USENIX Security Symposium.

[5] Franz, E., Jerichow, A., and Wicke, G., 1998. “A payment scheme for mixes providing anonymity”. In Trends in Distributed Systems for Electronic Commerce, W. Lamersdorf and M. Merz, eds., Springer Berlin Heidelberg, pp. 94–108.

[6] “Johnny” Ngan, T.-W., Dingledine, R., and Wallach, D. S., 2010. “Building incentives into tor”. In Financial Cryptography and Data Security, R. Sion, ed., Springer Berlin Heidelberg, pp. 238–256.

[7] Androulaki, E., Raykova, M., Srivatsan, S., Stavrou, A., and Bellovin, S. M., 2008. “Par: Payment for anonymous routing”. In Privacy Enhancing Technologies, N. Borisov and I. Goldberg, eds., Springer Berlin Heidelberg, pp. 219–236.

[8] Patidar, R., and Bhartiya, R., 2013. “Modified rsa cryptosystem based on offline storage and prime number”. In 2013 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1–6.

10 Disclaimers

TORIUM is not a subsidiary of Tor, nor is it endorsed by Tor. Any opinions or views expressed by the Tor team are not associated, nor a direct reflection of the team at TORIUM.