THE HAYSTACK PROTOCOL

COOPER DOYLE & THOMAS NOMMENSEN

Abstract. The IOTA Tangle is a distributed ledger which allows for decentralised storage and transfer of data. The Directed Acyclic Graph (DAG) architecture underlying the Tangle creates the potential for feeless transactions between users and statistically indefinite network scalability [IOTA WP]. These features and the inherent decentralised nature of the Tanlge make IOTA an extremely attractive candidate for a anonymous and cryptographically secure communications protocol.

The degree of anonymity of most tools available today is limited by the presence of communications metadata, which can be exploited by eavesdroppers to undermine the security of a network and gather information on users. Despite these issues, the idea of metadata resistance, or that of concealing not only the content, but also the context of communications, is scarcely addressed. In what follows we outline the requirements for a communications system to be anonymous, and present the Haystack Protocol over the IOTA Tangle as a solution.

1. Introduction

Metadata is used ubiquitously in communications as a means of facilitating the routing of messages between nodes on a network, and includes any information that can be used to distinguish a packet. While this method confers many advantages, such as speed and efficiency of the network, the metadata associated with packets can easily be exploited by an opponent in order to trace messages and gather information on users. This data is not always limited to the identities of the sender and receiver, their location, or the time of a conversation. Surveillance of metadata can be used to infer relationships between individuals, groups and even patterns in a user's daily life. So important is the use of metadata in surveillance that the former director of the NSA, Michael Hayden, declared in a debate against Dr. David Cole, the National Legal Director of the American Civil Liberties Union, "We kill people based on metadata".

The vast majority of communications systems available today make use of central servers to route messages, which not only requires *trust* in a central authority, but also renders the system vulnerable to metadata analysis. A malicious eavesdropper who is able to observe traffic data between users and the server can invariably reconstruct the routing table of the network and gather statistical evidence

of communication between clients. More sophisticated systems make use of decentralised, peer-to-peer relaying schemes which obfuscate the identity of senders and receivers, however, most are plagued with issues of reliability, scalability, speed and offline usability.

Here, we present a protocol which uses the IOTA Tangle as a platform for an *anonymous* and *asynchronous* communications network. That is, we require that users need not be online at the same time in order to successfully communicate. In this section we will first introduce the concepts of pseudo-anonymity and degree of anonymity, and explain the benefits of using the IOTA Tangle as a framework. We then present the Haystack protocol and consider a number of attack vectors and vulnerabilities. Finally, we discuss applications and perspectives for the future.

1.1. **Pseudo-Anonymity, Blockchain, and the Tangle.** Blockchain and other consensus structures offer an inherently decentralised platform for transfer of value, storage of data, smart contracts and communications. Most available blockchain technologies feature a public ledger on which all transactions associated with an "address" are readily available to any user. In the case of Bitcoin, anonymity and privacy can only be achieved by keeping a wallet address hidden from the public, however this is an issue since an address must be revealed in order to complete or receive a transfer of value or data. IOTA solves this problem by using virtually innumerable, unique, single-use addresses that are deterministically generated from a cryptographic "seed". In this case, it is the seed and not the individual addresses that correspond to a user's wallet. In order to retrieve a user's balance and fetch any data associated with their transactions, a software wallet can scan the IOTA Tangle for addresses corresponding to the seed. These addresses are cryptographically generated, so that obtaining a seed from a series of corresponding addresses is virtually impossible. Thus, the transaction addresses only make available the information for a single transaction, and do not reveal the entire wallet or user's history. For the sake of private messaging, this means that intercepting one message gives no information about the messages preceding or following it. This is known as *unlinkability*. In addition, by employing public key cryptography and digital signatures, the privacy and authenticity of the message content can be virtually guaranteed.

This method is not, however, impervious to metadata analysis. Messages are recognisable by their encrypted ciphertext and metadata, and are thus traceable and identifiable on the Tangle. If an eavesdropper is able to correlate the addresses that two clients attach to and request from, they can gather evidence that two users are in communication, even if they cannot in principle determine the content of the messages. Should the opponent manage to compromise an IOTA node, or construct a malicious node, this process is made even easier. That is to say this

system is "pseudo-anonymous". Schemes exist which circumvent these issues using mixing schemes which obfuscate the mapping from senders to receivers, however these methods often introduce a central point of failure and detract from the decentralised nature of the network, or fail to completely conceal network metadata. A fully decentralised protocol should provide complete anonymity without requiring trust in a central authority. The Haystack protocol over the IOTA tangle aims to address these problems using stochastic relaying methods and a multilayered format-preserving cryptosystem.

1.2. **Degree of Anonymity.** Statistically, the anonymity of a system can be graded according to its 'degree of anonymity', d, as outlined by Diaz et al[]:

$$d = \frac{H(X)}{H_M} \tag{1}$$

Where H(X) represents the entropy of the system given some information corresponding to observations made by an attacker. For a total number of participants N and a probability p_i that a particular message is originated from each this is given by:

$$H(X) = -\sum_{i=1}^{N} p_i \times log_2(p_i) \tag{2}$$

H_M is the maximum entropy of the system, corresponding to a state where each user is equally likely to be sending and receiving. A degree of anonymity approaching unity is therefore attained at high system entropy, where gathered data cannot be used to favour any particular sender.

Metadata can be understood broadly as the degrees of freedom which make packets of data identifiable, excluding the payload itself. For individual encrypted messages, the degrees of freedom are the ciphertext, size, time, sending and receiving addresses, and the geolocation of the message. Each of these provides a potential means of gathering and correlating data from intercepted packets in order to increase the statistical probability p_i that the message originated from the sender and decrease the likelihood for all other participants. This effectively reduces the degree of anonymity d of the system. The most obvious and effective way of combatting this kind of attack is to render the packet invariant to any identifiable degrees of freedom in the message. This network invariance (also known as metadata resistance) effectively obscures individual messages, and means tracking any particular connection is comparable to finding a needle in a haystack.

2. The Haystack Protocol

2.1. **Introduction and Description.** To obfuscate the addresses of senders and receivers in a conversation, the Haystack protocol implements a multilayered encryption and relay scheme. Each relayer in the message trajectory uses their private key to decrypt a layer of encryption and reveal the next relay address. This scheme enforces the requirement that the message ciphertext be distinct between each relay, and ensures that the destination address cannot be decisively determined by any relayer. In addition, by using a hybrid cryptosystem which normalises and preserves the ciphertext length, the packet size is made invariant to the content.

In order to relay messages off other users, participants in the network must first be able to identify active users by their tangle address, and be able to share cryptographic keys even when a user is offline. For this, Haystack implements a Dynamic Public Ledger (DPL) characterised by a unique "public" seed on the IOTA Tangle. Users can upload a unique public address and a public encryption key to the public ledger when they first initialise the service or return from a period of inactivity. In addition, this ledger is dynamic, in that only one address from the public seed is used as the *active* ledger at any given time. Offline contacts may be found by scanning the DPL in reverse order. This method allows for the implementation of a coarse-grain "last seen active" feature.

The decentralised and stochastic nature of the protocol make Haystack immune to many attacks. Since no information is accessible to tie packages to a particular user, there is no way to preference any particular user in assigning probabilities p_i in equation (2). This means that the anonymity of the system approaches unity and the protocol is maximally entropic. In what follows we outline the protocol in detail and provide a technical analysis of the network stability and vulnerability to attack. To preserve the security of the network, numerical values specific to the protocol are not given. We do, however, provide the reasoning behind each.

2.2. **The Dynamic Public Ledger.** In order to enable users to share keys and broadcast addresses on the network, the Haystack protocol implements a Dynamic Public Ledger (DPL), corresponding to a public seed on the IOTA Tangle. Clients upload a unique public address and a public encryption key to an address generated from the ledger seed. At any given time, only one address from the public seed is used as the *active* ledger. Transaction timestamps serve to achieve consensus among devices that the ledger has migrated to the next address. This is used as a way to exclude inactive nodes from the network, and allow users to broadcast new public addresses and multi-use encryption keys. The software wallet periodically scans the public seed for the last filled address in order to locate the active ledger.

The device then uploads a public address/key pair to make itself discoverable, and retrieves a list of active addresses to relay messages off.

2.3. **Encryption Scheme.** A critical feature of the Haystack protocol is the ability to employ a multilayered encryption scheme while enforcing a constant bundle size throughout the network. This ensures a high level of cryptographic security while maintaining invariance to message size, but requires the encryption scheme to be format-preserving. Asymmetric schemes do not, in general, conserve information in this way. In order to circumvent this issue, the Haystack protocol uses a hybrid cryptosystem, in which a pseudorandom secret encryption key is generated and associated with each relayer in a given trajectory. The message content is then symmetrically encrypted with each secret key in reverse order of the message trajectory in an iterative fashion. These secret keys are then asymetrically encrypted with the public key of the corresponding relayer, along with the next relay address, and included in the packet as metadata. When a relayer successfully decrypts a secret key with his own private key, this allows him to decrypt one layer of encryption from the message ciphertext and uncover a relay address without revealing any information about the overall trajectory of the message. Finally, at each relay, the final packet including the metadata is stream-encrypted with the public key of the next relayer in order to maximise information security and ensure that the packet, including its metadata, is completely distinct at each bounce.

2.4. **The Protocol.** When a user now decides to send an anonymous message over the IOTA network, the Haystack protocol proceeds as follows:

- The message is first normalised in size and encrypted with the users private key to ensure authenticity.
- If the message is longer than the "normal" it is fragmented and each fragment is treated independently.
- The device randomly selects α lists of β addresses from the public ledger, corresponding to the time-ordered degenerate trajectories that the message will take over the network. One address/key pair in these lists is always the destination address.
- For each address in the trajectory, the program generates an associated pseudo-random *secret key*
- The message is symmetrically encrypted with the destination user's *secret key*.
- The resulting ciphertext is then sequentially encrypted in reverse order using the secret keys associated with each address in the trajectory *preceding* the destination address.
- The secret keys and relay addresses for each bounce are then encrypted with the intended relayers *public* key and included as metadata.

- Finally, the packet is stream-encrypted with the public key of the first relayer in the trajectory and is attached to the first relay address.

Each device on the network acts as a node which periodically scans its active public address as a background process. Messages to be relayed are pulled from the tangle and decrypted. The device then attempts to decrypt each piece of metadata in order to reveal a secret key and relay address. If a relay address is successfully obtained, the message content is decrypted with the associated secret key. An identifier allows the user to determine whether a message has been successfully decrypted. If a message is not recognised, the digest and original metadata are then encrypted with the next relayers public key, located on the DPL, and sent to the next relay address.

The α-fold degeneracy of message trajectories works to ensure fast transmission times and circumvent issues with offline or idle nodes: The message will always arrive first through the path of least resistance. The number of bounces, β, may be increased in order to severely reduce the chance that *all* nodes in a trajectory are malicious, as this is the only case in which an opponent can reconstruct a trajectory. The layered encryption guarantees that only the intended relayer can decrypt the message and determine the following address. In addition, as relayed packets are received from public addresses and sent from private addresses, transactions on the public ledger contain no information linking to the trajectory of a message as illustrated in fig. 1(a). The tangle contains random transactions of identical size from private addresses to public addresses. No information is available to link these transactions. Fig. 1(b) shows the additional transactions effectuated by the relay nodes which pull information from a public address and send from a private address.

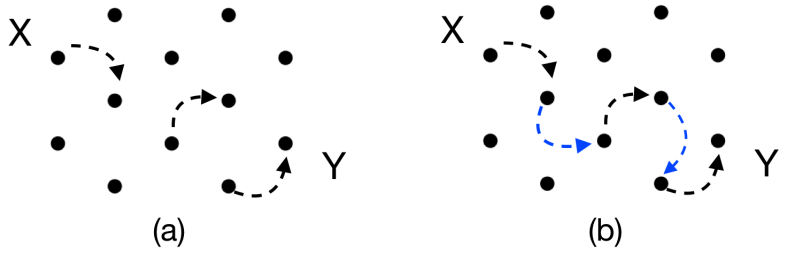

FIGURE 1. **(a)**: Message trajectory on the Haystack network as viewed on the IOTA tangle **(b)**: The same trajectory with the added blue arrows indicating pairs of public receiving addresses and private sending addresses

2.5. **Network Stability and Statistics.** Consider a network where N users send an average of $\bar{f}$ messages in a given unit of time. Each message is assigned α trajectories, with each trajectory containing β bounces. The average probability p that a particular message is relayed off any given user is simply $p = \frac{\alpha\beta}{N}$, and hence the average rate of message relays per user λ is given by:

$$\lambda = p \times N\bar{f} = \alpha\beta\bar{f} \tag{3}$$

Implicit here is the assumption that $N >> \alpha\beta$, such that the probability of appearing twice in a trajectory is small. Notice that this result is independent of the number of users N, and depends only on the rate of real activity on the network $\bar{f}$ and the predetermined system parameters α and β. This means that the activity of any node will retain a constant average as the number of users grows and the network can be said to be stable.

The trajectory of each message on the network is selected at random by the sender's device from the public ledger, thus the probability P of a node bouncing k messages within a fixed time follows a Poisson distribution:

$$P(k) = \frac{\lambda^k}{k!}e^{-\lambda} \tag{4}$$

2.6. **Offline Nodes and the Ping Function.** An important threat to this protocol is the existence of malicious or lazy nodes which can reduce the reliability of the network by failing to relay packets. This is partially addressed by the incorporation of α degenerate trajectories for each message, however a larger numbers of bounces, β, will increase the likelihood that a packet will encounter a point of failure. Given a fraction $p = \frac{N_{off}}{N}$ of offline or malicious nodes the probability of losing a message can be calculated from a simple binomial distribution:

$$P(losing\,a\,message) = (1 - (1 - p)^{\beta})^{\alpha} \tag{5}$$

As expected, a large β increases the probability of encountering an offline user, while a large α, by increasing the degeneracy of message trajectories, reduces the likelihood of a losing a message (recall that $1-(1-p)^{\beta} < 1$ is itself a probability). If we consider that 1 in 100 relayers is idle or malicious and take the network parameters $\alpha = 3$ and $\beta = 5$ in equation (5), we find that only 1 in every 10,000 packets are lost. The threat of offline nodes is further reduced by the time-dependence of the DPL. Inactive users are periodically expelled from the active pool, however this does not in principle prevent a resourceful attacker from spamming the DPL with malicious addresses in order to slow the network.

Conveniently, the Haystack protocol allows for the creation of a "ping" message, which is completely indistinguishable from an actual message, except that the final recipient is the user himself. In this way, clients can test the responsiveness of individual or multiple nodes by creating the appropriate trajectory and listening for the response. The reliability of the network can thus be appraised in real-time by each user, and continuous pinging of addresses on the DPL allows clients to establish a pool of trusted nodes. These relays can also be used to artificially raise the noise level of the system in order to conceal activity, as discussed in the traffic confirmation attack scenario below. Failing to relay messages will lead to reduced anonymity for the user, however the existence of offline nodes sending messages will also increase the noise level without considerably affecting stability as long as other users are able to avoid them.

3. Hypothetical Attack Scenarios

An important feature of the Haystack protocol is the concealment of information from relayers themselves, and the robustness of the DPL. In principle, an attacker with sufficient computational power could overwhelm the ledger and redirect all traffic through malicious nodes to gather metadata on users. Many of these issues can be solved by running a local instance of the Tangle to conceal activity, however, we consider the case where a client connects to an external IOTA node or where a client's local instance of the Tangle is compromised. Although the protocol conceals most information, an important piece of metadata which can be used to leak information is the specific timing of sends and requests from a node. An attacker can continuously relay off a user and monitor the latency of the response. By using the fact that the latency of traffic streams through a node is influenced by traffic volume, he can then determine the times of relays. If two users are found to be interacting with the network at similar times, an attacker can achieve high confidence that the users are communicating through the service. This is known as a traffic confirmation attack. The continuous 'bouncing' of messages and the packet invariance within the network render the system highly resistant to this kind of analysis, since all users on the service are in contact with the IOTA network in an ongoing way. The signal is, in a sense, shrouded in noise in order to obfuscate activity.

3.1. **Traffic Confirmation Attack.**

Consider now the situation where an eavesdropper is able to probe the activity of a user and attempts to determine whether the user is actively communicating over the network by observing the frequency of interactions with a node. We can determine the posterior likelihood of a hypothesis, H, that a user is actively communicating over the service, given a set of observations D corresponding to a rate of $k + \alpha n$ interactions with the network

in a time δt. Here, for convenience, k is the closest integer to the average rate of packet relays per user λ. Using Bayes' theorem:

(6)
$$p(H \mid D, I) = \frac{p(H \mid I)p(D \mid H, I)}{p(D \mid I)} = \frac{p(H \mid I)p(D \mid H, I)}{p(H \mid I)p(D \mid H, I) + p(\bar{H} \mid I)p(D \mid \bar{H}, I)}$$

The denominator $p(D \mid I)$ represents a normalisation factor and can be obtained by integrating over all hypothesis space. The null hypothesis $\bar{H}$ is the statement that the user is not actively sending over the network. The likelihood $p(D \mid \bar{H}, I)$ then corresponds to equation (4) evaluated at $k + \alpha n$. The prior is chosen to be $p(H \mid I) = \frac{\bar{A}}{N} = r$, the approximate rate of active users sending messages, to reflect an unbiased state of ignorance. This gives:

(7)
$$p(H \mid D, I) = \frac{1}{1 + \frac{1-r}{r}\frac{k!}{(k+\alpha n)!}\lambda^{\alpha n}}$$

As required, the probability function converges asymptotically to 1 for large n. This result makes clear that by imposing the condition $\lambda >> n$, the probability of the hypothesis can be made arbitrarily small. For instance, consider a network with an active participation rate of $r = \frac{1}{20}$ users each sending on average $m = n = 6$ messages an hour. If each message is assigned $\alpha = 3$ trajectories with $\beta = 5$ relays, this corresponds to an average bounce rate of approximately ~ 5 relays per hour for each user, since $\lambda = rm\alpha\beta$. In addition, we consider the continuous and random pinging of active nodes operating in the background, which we choose here to be 30 relays per hour. This makes the effective hourly bounce rate $\lambda_{eff} \sim 65$ relays.

Now, if a participant sends $n = m = 6$ messages in under an hour, an eavesdropper's confidence that the user is actively communicating over the network given the prior of 1/20 is only 37%. This should be contrasted with centralised mixing schemes where a single intercepted message yields a confidence of virtually 100% that a participant is actively using the service. Of course, this scenario corresponds to an ideal case where the average participation rate r and the rate of pinging between nodes are both known and constant, which is not the case in reality. This uncertainty would make it even harder to detect whether a user was actively using the service.

The importance of this result becomes evident when one imagines attempting to correlate the already highly stochastic communication patters of two or more clients interacting over the network, as the uncertainty related to each is compounded. In addition, this attack requires an *a priori* suspicion that two clients are in contact. One clear limitation to this obfuscation method is the allowable

effective bounce rate λ_{eff}, which must be tuned according to the available bandwidth of the network and the capabilities of client devices by varying the protocol parameters. As more nodes become available and the quality of client devices increases, this limitation will become less significant. In particular, if every client on the service runs a full node, the network capabilities will grow exponentially.

3.2. **Attacks on the Dynamic Public Ledger.** Consider now the possibility of an attacker attempting to spam the Dynamic Public Ledger. By broadcasting dummy addresses a user can decrease the efficiency of the network since this is equivalent to introducing offline nodes, or dead-ends in message trajectories. While this can damage the reliability of the network, the effectiveness of such an attack is curbed by the Tangle's resistance to spam attacks. On the IOTA tangle, Proof of Work (PoW) is network-bound. Combined with bandwidth constraints this gives a reasonable assumption that it will be hard to occupy a big portion of the EM spectrum (due to necessity for radio-modules that are spatially wide-spread). Additionally, the Haystack protocol can implement its own PoW scheme in order to validate addresses on the DPL. This can raise the computational power and wait-time required to successfully access the network arbitrarily high, to a point where spam attacks on the ledger are unreasonably computationally expensive for a large number of users. Since the ledger is dynamic, malicious addresses cannot be accumulated on the ledger over an indefinite amount of time.

If we consider the choices $\alpha = 3$ and $\beta = 5$ in equation (5) as an example, we find that controlling 50% of nodes on the network prevents approximately 90% of all messages from arriving at their destination. This issue is addressed by the introduction of the ping function, which allows users to assess response times and identify effective relayers. Furthermore, this type of attack is simply not effective enough to be worthwhile. Not only because communication over the network is still possible, but also because even with so much control, no information about senders or receivers of messages can be gathered, nor can the content of the encrypted messages be revealed. Since each *sending* address of the IOTA network is unique for each relay, the intrinsing *unlinkable* nature of the protocol means trajectories can never be reconstructed. In addition, as the network is scaled and the number of users increases, the computational resources required to achieve such network dominance will be unreasonably large.

4. Practical Considerations and Outlook

4.1. **Beyond Messaging.** Although the previous sections are predominantly focused on private and anonymous messaging, the concepts are readily extended to streams of encrypted data. Consider a server running the Haystack protocol.

Clients could access data on the server through the decentralised network in near complete anonymity and with a high level of security. Such a system would resemble the Tor protocol, with the distinction that the network does not require volunteers to run nodes, only users, since the protocol operates over the decentralised IOTA tangle. One important consideration here is management of the size of the Tangle supporting the network. Since each data packet is repeated through the network and stored, if users are sharing large files the Tangle will become unreasonably large without frequent snapshotting and pruning. Another consideration is the speed of the network, which is in principle at a disadvantage when compared to protpcols such as Tor based on TCP transfers. The theoretical confirmation time of the Tangle is approximately 2 milliseconds, although it currently stands at around 10 seconds. This transaction time decreases with scale, so that with more nodes and more transactions the network becomes faster. The speed of the network could therefore, with the appropriate infrastructure, approach the real-world speed of Tor.

4.2. **Post-Quantum Encryption.** While the current implementation of the Haystack protocol is not quantum invariant, the scheme can easily be extended to incorporate post-quantum encryption. The symmetric encryption system employed to encrypt content is already quantum-invariant, however the asymmetric cryptography used to encrypt addresses and secret keys is not. This can easily be solved by implementing the Merkle signature scheme, NTRUEncrypt, or the McEliece cryptosystems, however the size of the required public keys is in megabytes, and thus threatens to weigh significantly on the network and on user devices. As the quality of available technology improves, this will become less of a problem.

4.3. **About the Creators.** The Haystack protocol is developed by the Consensus group, a group of physicists based in Sydney, dedicated to anonymity, information security and decentralisation.

Contents