A Holistic Approach Towards Peer-to-Peer Security and why Proof of Work Won't Do

Bernd Prünster[1], Dominik Ziegler[2], Chrisitan Kollmann[3], and Bojan Suzic[4]

[1] Secure Information Technology Center – Austria (A-SIT), Graz, Austria
bernd.pruenster@a-sit.at
[2] Know-Center GmbH, Graz, Austria
dominik.ziegler@tugraz.at
[3] A-SIT Plus GmbH, Vienna, Austria
christian.kollmann@a-sit.at
[4] Institute of Applied Information Processing and Communications (IAIK), Graz University of Technology, Austria
bojan.suzic@iaik.tugraz.at

Abstract. Separation of identity and location is one of the key properties of peer-to-peer networks. However, this separation can be abused to mount attacks against the network itself. Our contribution in this matter is twofold: First, we present a security-first design for P2P networking based on self-certifying identifiers. It provides message authenticity, integrity of routing tables, and authenticated communication, is resistant (and not only resilient) against many typical peer-to-peer-specific attacks, and guarantees uniform identifier distribution. The second aspect of our contribution disproves the often-quoted assumption that proof-of-work-based identifier generation can sufficiently hinder certain peer-to-peer attacks such as the Sybil attack. This finding seriously questions previously proposed proof-of-work-based defence mechanisms and leads to the only conclusion possible: Proof-of-work-based measures to limit arbitrary identifier generation do not stand the test of reality.

Key words: peer-to-peer networks, network security, decentralised routing, authenticated communication, self-certification, proof of work

1 Introduction

Peer-to-peer (P2P) networks like *BitTorrent* [3], for example, enable the creation of logical overlays on top of heterogeneous networks by separating location from identity. Especially in fully decentralised P2P networks, validating the origin and authenticity of data can become a challenging endeavour. Much research effort has since been put into securing such systems [1, 17, 5, 20, 9, 10, 19]. Some proposed improvements succeed in their own domain—usually at the cost of introducing additional complexity and overhead. Others (even recognised ones) do not stand the test of reality, as we have discovered.

We present a security-first approach towards P2P networking based on *self-certifying identifiers*, a concept inspired by *self-certifying pathnames* originally

described by Mazières and Kaashoek [12]. We show that the combination of self-certifying identifiers and regulation of identifier generation (before joining the network) suffices to solve a variety of P2P-specific issues. Contrary to previous proposals [1, 20, 5], we show that this requires no complex governance models, external dependencies, or sophisticated, multi-layered signature schemes.

We provide both a requirements-driven security analysis as well as a performance evaluation of our approach. Our design targets *structured* overlay networks due to their efficient routing, low overhead, and guaranteed round trip times and does not impose any restrictions on possible applications running on top of P2P networks.

The second major aspect of our contribution deals with countermeasures against *Sybil* [4] and *eclipse* [17] attacks. We discovered that the often-made claim that a proof-of-work-based decentralised identifier generation process can defend against such attacks is simply not true under real-world constraints. Section 5 elaborates on this in detail.

The following section explains the problem we try to solve, discusses the shortcomings of previously proposed defence mechanisms and motivates why a holistic approach towards designing secure P2P systems is necessary.

2 Background

The heyday of peer-to-peer research dates back more than ten years, when not only *Kademlia* [11], *Chord* [18], and *CAN* [16] but also unstructured P2P networks were conceived. Back then, the device landscape and usage patterns were dramatically different from today's and security was often not considered a top priority. Still, most P2P-specific attacks have also been long known and some currently deployed systems are based on concepts established at that time. The ease of routing table poisoning or *distributed denial-of-service* (DDoS) attacks in P2P scenarios strongly suggest that security should be considered a first-class feature for networked applications. The following section provides a short summary of key P2P security issues and attacks.

2.1 Well-Known P2P Security Concerns

Routing information in (decentralised) P2P networks is provided by network participants and the correctness of this information is vital. Consequently, *routing table poisoning*—propagating false routing information through the network—constitutes a category of potentially devastating attacks. The *eclipse* attack [17] is one such attack, where attackers deliberately place nodes in close proximity to a target node (or target information) and refuse to respond to queries regarding the target, thus eclipsing it.

The so-called *Sybil* attack [4] describes an adversary generating multiple identities and posing as many unrelated entities to the network. This can be used to mount eclipse attacks. It can also influence consensus mechanisms, due to the

illusion of independent actions, while in reality a single powerful party makes all decisions.

Honest nodes can also be tricked into overwhelming other nodes with queries, routing table updates, or responses, resulting in DDoS attacks. A specific variant of this attack, known as the *backscatter* attack [13], uses forged requests containing false sender information to provoke a vast amount of responses from honest nodes, all targeting a single victim.

Requests can also be involuntarily transmitted to malicious nodes, as soon as *Man-in-the-middle* (MITM) attacks can be mounted. This issue essentially boils down to inaccurate routing information and/or adversaries impersonating other nodes.

Still, P2P networks can be more resilient than centralised systems, due to their distributed nature. At the same time, many of the just described attacks arise precisely due to the lack of a single entity being able to detect and counter malicious activity. In general, networks lacking built-in defences against *message forgery* are susceptible to various attacks. The following section discusses previously proposed defence mechanisms.

2.2 Previous Work

A variety of hardening mechanisms targeting P2P-specific attacks have been proposed. *S/Kademlia* [1], for example, presents a concept to tackle eclipse attacks using a combination of self-certifying identifiers and mandating a proof-of-work for identifier generation, complemented with disjoint lookup paths. Section 5, however, demonstrates that this solution is not adequate anymore. Moreover, the authors also introduce additional traffic and complexity to node lookups, which we show becomes unnecessary, when redesigning routing procedures based on the full potential of self-certifying identifiers.

Much research has been done to try and conceive decentralised, widely-applicable P2P network designs. At the core of many proposals lie defences against Sybil and eclipse attacks. Some strategies against eclipse attacks assume working Sybil defences in place [17]. Levine, Shields, and Margolin [9] classify more than 90 approaches to defending against Sybil attacks into five categories: trusted certification, no solution, resource testing, recurring costs and fees, and trusted devices—with trusted certification being the only way to actually prevent Sybil attacks. Resource testing (sometimes also continuously, inducing recurring costs) is often proclaimed as a viable solution to deploy Sybil and eclipse-resilient P2P networks. This encompasses computing ability testing (proof-of-work), bandwidth testing, incorporating IP addresses, and storage testing—and always assumes somewhat resource-constrained attackers. Today, computing power, bandwidth, and storage is not distributed evenly. Consequently, imposing high bandwidth, storage, or CPU requirements can actually prevent widespread adoption of a system. Generally speaking, a smaller network is easier to overpower than a lager one. Thus, hindering adoption of a system can make it easier for attackers to gain signifiant influence on the network.

Li et al. [10] explain why social-network-graph-based solutions are only effective in controlled (lab) environments (based on the work by Viswanath et al. [19]) and illustrate why virtually all proof-of-work-based designs are also of limited use under real-world constraints. This also seriously questions the performance of schemes such as *SybilGuard* [20]. Li et al. argue for a "repaired" proof-of-work approach. This concept, however, also relies on assumptions, which simply do not reflect the current device landscape and the current cost of computing resources. Moreover, a clear evaluation of the performance impact of any proposed solution is crucial in order to judge its real-world applicability. Often, however, no such figures are available.

2.3 Open Issues

More important than previously tackled problems are the ones not discussed in this context. Eclipse attacks can be devastating to P2P networks, and Sybil attacks also cannot be dismissed in fully open, decentralised systems. However, countermeasures against one attack must not induce additional churn or significantly increase the overall traffic, as this can easily result in DDoS attacks. Moreover, inducing recurring computational costs or communication overhead to participate in a network is hard to justify with mobile users in mind. Arguing for a proof-of-work-based identifier generation without demonstrating how the difficulty of the problem being solved scales in relation to the overall network size simply fails to provide essential information required to build a working system.

The following section presents the requirements-driven architecture of our approach. We argue that introducing self-certifying identifiers, and imposing cryptographic signatures on all traffic can prevent many forms of attacks on P2P networks without introducing additional requirements or artificial constraints. We then provide a thorough security evaluation of our system in Section 4.

3 Architecture

This section elaborates on an architecture and protocol for implementing a structured P2P network based on a Kademlia-like routing mechanism and state-of-the-art cryptographic primitives.

Our design makes it actually impossible for nodes to produce false information and have the network accept it. It does so by ensuring that origin and contents of all messages are verifiable even without knowing their source. In short, we guarantee that (routing) information regarding a particular node also originated at this node. This effectively ensures that only valid routing information can spread through the network. To satisfy these claims, the following requirements must be fulfilled:

R1 Decentralisation Maintaining the connection to the network must not require central instances.
R2 Self-Certification Node identifiers must be self-certifying in order to prevent identity theft.

R3 Uniform Identifier Distribution Identifier generation must be random, one-way, and unpredictable.
R4 Message Authenticity It must not be possible to alter (routing) messages passed along the network.
R5 Source and Destination Authenticity It must not be possible to spoof sender or recipient of messages.
R6 No Dependency on Gossip No operation must be dependent on information about nodes produced by other nodes.
R7 Routing Information Authenticity All information used for routing must be authentic at any time.
R8 Resiliency Against DoS Attacks The overlay must me resilient against *denial-of-service* (DoS) attacks such as eclipse attacks, and backscatter attacks.
R9 Secure Hash-Function-Based Identifiers Identifiers must be based on secure hash functions.
R10 Low Overhead Memory consumption and computational overhead should be within realistic constraints even for networks consisting of millions of nodes.
R11 Resiliency against Sybil Attacks A single entity must not be able to (automatically) generate a large amount of identifiers.

3.1 System Model

We have identified the properties necessary to fulfill the previously discussed requirements and present the following descriptive model for decentralised, structured peer-to-peer networks:

- Each node in the network has an identifier independent of its network location.
- Each node generates an *elliptic curve* (EC) key pair.
- Creation of node identifiers must not be automatable (this follows from R11), i. e. it has to include information not automatically obtainable.
- This information must be verifiable offline to enable fully decentralised network operation (following R1). We require a signature over a node's public key created by a trusted authority (cf. [10]), obtainable only after completing a challenge-response procedure, such as *reCAPTCHA*[1].
- An identifier is computed by calculating a cryptographic hash (such as *SHA-3-256* [15], following R9) over the authority-signed public key of a node's key pair (this follows from R2 and R3).
- Communication is message-based.
  - Each message contains a message code, indicating the type of the message.
  - Each message has to be signed by its sender, using the private key corresponding to the sender's identifier (following R4).
  - Each message contains the signed public key corresponding to the signing key (this follows from R5).
  - Each message contains a timestamp.

[1] `https://www.google.com/recaptcha/`

  - Each message contains a cryptographic nonce, called the *communication ID*.
  - Each message contains the identifier of the intended receiver (following R5).
- The authenticity of each incoming message is checked by verifying its signature.
- Nodes cannot "speak for" other nodes (following R6). Instead, information about the location of a node is communicated by forwarding messages originally sent by this node (following R7, messages are contained in routing table entries).
- A receiving node has to check, if it is the indented receiver of a message. This prevents misdirected responses and backscatter attacks (following R8).

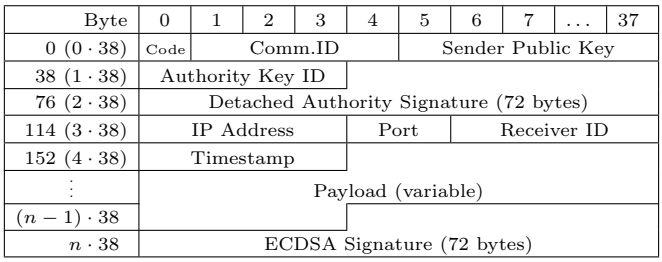

Fig. 1. Message Format

By tying identifiers to the possession of private keys, it becomes computationally infeasible to present a specific identifier to the network without possessing the associated private key—typically, a node will announce its location (IP address) and its identifier to the network upon joining.

Our message format (see Fig. 1) thus mandates that messages include a signed statement about their origin. This statement also doubles as the key, which is used to verify the signature. Consequently, nodes cannot assume a false identity. When queried for some node, the responding node also cannot fabricate false information, since the response is expected to contain the requested information in the form of a statement signed by the entity who originally issued it. Contrary to previous proposals [1, 2], no additional messages and no further overhead are required to verify this information. This assumes certificates reflecting the authority key ID used in a particular message are rolled-out.

To prevent replaying of stale messages, the sequence of messages originating from a node is important. The timestamp is therefore considered *relative* in the sense that it is strictly monotonic increasing on a per-node basis. A message sent from node A after a message sent from node B can therefore still feature an earlier timestamp. In short, no common network time is required.

Finally, the communication ID is necessary to associate incoming responses to previously sent queries. The overhead introduced by this approach is discussed in the following section (in accordance with R10).

3.2 Low-Overhead, High-Performance Self-Certifying Identifiers

We based our work on the general idea presented by the *Host Identity Protocol* (HIP): "In HIP, the public key of an asymmetric key pair is used as the Host Identifier (HI). Correspondingly, the host itself is defined as the entity that holds the private key from the key pair" [14].

Identifiers are based on the *SHA-3* [15] cryptographic hash function and elliptic curve cryptography, enabling compact representations and reducing the overhead introduced by this concept. Signatures are created using the *Elliptic Curve Digital Signature Algorithm* (ECDSA) [8] based on 256-bit keys. By employing point compression, public keys can be represented by as little as 257 bits: 256 bits to represent the x coordinate of the public key and one bit to indicate the sign of the y coordinate. Aligning these 257 bits to byte boundaries still leaves seven bits to encode curve identifiers within a total size of 33 bytes. Using those seven bits to reference the underlying curve allows for a considerable degree of extensibility. Our design directly enables establishing authenticated communication channels, e. g. to perform an *Elliptic Curve Diffie-Hellman ephemeral* (ECDHE) key agreement. This inherently prevents man-in-the-middle attacks.

Table 1. System performance

Number of OPs	ID Gen.	Signing	Verification
100,000	8,385 ms	10,268 ms	31,648 ms
500,000	41,245 ms	50,609 ms	157,823 ms
1,000,000	83,399 ms	101,600 ms	317,111 ms
Average/s	≈12,000 ops	≈9,900 ops	≈3,200 ops

We have evaluated the computational overhead of introducing self-certifying identifiers using a cloud computing instance. On *Hetzner*'s smallest and cheapest CX11[2] cloud instance featuring one virtualised *Intel Skylake Xeon* CPU core, it is still possible to verify and sign thousands of messages each second, as shown in Table 1.

The following section explains our routing protocol in detail, and argues that the overhead introduced is minimal.

3.3 Authenticated Routing Protocol

Our routing protocol is based on the Kademlia *distributed hash table* (DHT) [11]. Therefore, we shall highlight only the differences to the original design.

Apart from the last seen time, the information required to construct routing table entries (node identifier, IP address and port, timestamp, ...), is already contained in messages defined by our design. Therefore, we can keep messages from other nodes as a whole to build up the local routing table. The messages

[2] `https://www.hetzner.com/cloud`

remain small ($\approx$ 300 bytes), thus only inducing a small overhead when compared to the original Kademlia design.

During a node lookup process, all nodes respond with messages contained in their routing tables. Therefore, receiving nodes can verify the authenticity of the messages, as explained in Section 3.2. Additionally, the responding node includes a fresh message, to enable the receiver to update its respective routing table entry. In accordance to [1], nodes must ensure disjoint lookup paths, to prevent eclipse attacks.

Mandating self-certifying identifiers essentially prevents forged messages from being introduced into the network. This way, all information received about the location of nodes is verifiably accurate. Identifiers themselves also cannot be forged, as they cannot be chosen deliberately and require the possession of a private key to be recognised. The following section provides a comprehensive security evaluation of our design.

4 Security Evaluation

In order to verify that our design does indeed provide the proclaimed security features, we performed a *Common Criteria for Information Technology Security Evaluation* (CC)-based [7] security evaluation.

We do not consider attacks on the implementation and assume correctness of all involved cryptographic primitives. We further consider the underlying network infrastructure (the IP network) to operate as expected. As our design features built-in mechanisms to prevent various attacks instead of countering them after the fact, we refrain from discussing *countermeasures* as defined by the CC methodology. Instead, we evaluate *system properties* for countering/preventing threats. Apart from this deviation, the evaluation adheres to the definitions of the CC evaluation methodology and identifies threats aimed at assets, as well as explicit assumptions, and security objectives. We did, however, introduce a separate section regarding threat mitigation. The outcome of this evaluation shows that the interdependence of some objectives, threats, and system properties are not an issue since no residual threats remain.

4.1 Assets

The following assets need to be protected to provide all desired security features:

[**A1**] **Message** Messages form the main building block of our design and are used for building routing tables. This information must be protected from alterations but is not considered confidential.

[**A2**] **Private Key** As the private key is tied to identifier and signature generation, it needs to be kept confidential and must not be shared among entities.

[**A3**] **Identifier** Identifiers uniquely define nodes in the network and are used to contact other participants. Consequently, confidentiality is irrelevant, but uniqueness is imperative.

[**A4**] **Routing Table** The routing table stores all information necessary to contact other nodes. This includes identifiers, IP addresses as well as messages and timestamps.

As our design focuses on providing an overlay network without targeting any specific application, no further assets are relevant.

4.2 Assumptions

Our approach tries to prevent many attacks instead of countering them through tightly-regulated behavioural constraints. It therefore relies only on two single assumptions:

[**AS1**] **Uncompromised Trusted Authority** The trusted authority signing keys operates as intended and is not compromised.

[**AS2**] **Secrecy of Private Key** The proposed design requires messages to be signed using a cryptographic key, which ensures authenticity and integrity of all sent data. Keeping this key secret is thus crucial for the security of the overall system: Attackers obtaining the private key of a specific node can sign arbitrary messages and thus impersonate this node. However, as no purely technological measures exist to keep the private key secret, we therefore assume that appropriate measures are taken to keep the private key secret. The same holds for *[AS1]*.

4.3 Security Objectives

Our design aims at providing the following security objectives:

[**O1**] **Message Integrity** As all information is exchanged through messages, it must not be possible to tamper with messages, without recipients detecting this.

[**O2**] **Uniform Distribution of Identifiers** A uniform distribution of node identifiers must be guaranteed to prevent attackers from being able to generate specific identifiers, e. g. to carry out eclipse attacks.

[**O3**] **Integrity of Routing Tables** Accurate routing tables are integral to operating a P2P network.

[**O4**] **Message Order** To process only up-to-date routing information, nodes shall detect and discard messages which contain information already processed at an earlier point in time.

[**O5**] **Reachability** Assuming working transport, physical, and data link layers, this objective effectively boils down to answering node lookups.

[**O6**] **Authenticated Communication** While establishing authenticated communication channels is not a P2P-specific feature, it is still considered an essential security objective.

4.4 Threats

We have identified the following threats based on known attacks aimed at peer-to-peer networks. Table 2 illustrates how these map to security objectives.

[T1] Eclipse Attack Being able to generate specific identifiers, or identifiers close to a specific node enables eclipse attacks. This directly violates *[O2]*, and *[O5]*.
[T2] Impersonation An adversary able to assume the identity of a specific node, can send arbitrary messages which then appear to be originating from the original node. This can lead to disruptions, violating *[O3]*, and *[O5]*; impersonation can be a consequence of violating *[AS2]*.
[T3] Message Forgery An attacker could try to alter messages originating from other nodes in the network or compose messages which contain inaccurate information, violating *[O1]*, *[O3]*, and thus *[O5]*.
[T4] Replay Attacks Replay attacks are defined as re-sending previously captured messages to disrupt the network and violate *[O4]*. This can lead to compromising *[O3]*, and *[O5]*.
[T5] Routing Table Poisoning Routing table poisoning as a general attack describes deliberately inserting false information or malicious nodes into routing tables. This directly conflicts with *[O3]*, leading to compromising *[O5]*.
[T6] Sybil Attacks Sybil attacks can disrupt a network in various way, from a DoS point of view, this violates *[O5]*. More generally speaking, a successful Sybil attack can lead to fatal disruption of the whole network.
[T7] Backscatter Attacks The backscatter is a DDoS attack and violates *[O5]*.
[T8] Man-in-the-Middle Attacks When trying to establish a communication channel, an attacker could try to fool a node into contacting their node instead of the intended target. This directly violates *[O5]* and *[O6]*.

4.5 System Properties

This section illustrates how most threats can be mitigated by one or more system properties. The security features of our system are the result of a combination of properties. Because of this, we first define all relevant system properties and separately elaborate on how threats are inhibited. Table 3 provides an overview of the results obtained as part of this security evaluation, illustrating which combinations of properties prevent certain threats. Our design features the following properties:

[P1] Public-Key-Based, Self-Certifying Identifiers Identifiers are created by calculating a cryptographic hash over the authority-signed public key of a node's EC key pair. These form the basis for countering a variety of threats.
[P2] Signed Messages As explained in Section 3, this makes it possible to unequivocally determine the origin and authenticity of a message.
[P3] One-Way Derivation of Identifiers Using an unbiased hash function as final step during identifier generation ensures uniform identifier distribution.
[P4] Message Timestamp Message timestamps enable stale messages originating from a specific node can be detected.
[P5] Recipient Declaration in Messages By mandating messages to declare the intended recipient, incoming messages addressed to some other node can be detected and discarded.

4.6 Threat Mitigation

Typically, a combination of system properties is required to prevent a threat. Table 3 shows exactly how each threat is mitigated by a combination of system properties (and *[AS1]*).

Table 2. Mapping of threats to security objectives

	T1	T2	T3	T4	T5	T6	T7	T8
O1	○	○	✗	○	○	○	○	○
O2	✗	○	○	○	○	○	○	○
O3	○	✗	✗	✗	✗	○	○	○
O4	○	○	○	✗	○	○	○	○
O5	✗	✗	✗	✗	✗	✗	✗	✗
O6	○	○	○	○	○	○	○	✗

Table 3. Mapping of threats to system properties preventing them

	T1	T2	T3	T4	T5	T6	T7	T8
P1	✗	✗	○	○	✗	○	○	✗
P2	✗	✗	✗	○	✗	○	✗	✗
P3	✗	○	○	✗	✗	○	○	○
P4	○	○	○	✗	✗	○	○	○
P5	○	○	○	○	○	○	✗	○
AS1	✗	○	○	○	○	✗	○	○

All threats are effectively countered, given a trusted authority regulates identifier generation. During normal operation, the system is fully decentralised. This, in conjunction with the system performance discussed in Section 3.3, demonstrates the applicability of our system under real-world conditions. At the same time, our design does not introduce additional complexity, such as auditing mechanisms, or external dependencies which were previously proposed as defences against eclipse attacks, e. g. [17, 5]. This clearly presents an improvement over existing systems. Still, many (even widely-used) P2P designs suggest that fully decentralised identifier generation and satisfactory attack resilience are possible based on proof-of-work approaches—a claim we shall refute in the following section.

5 Proof of Work and Attack Resiliency

This section illustrates why proof-of-work-based defence mechanisms against eclipse and Sybil attacks fail to achieve their goals. We use our system's model as a baseline (i. e. worst case for attackers), replace regulated identifier generation with a proof-of-work approach and evaluate the difficulty and cost of certain attacks. The results clearly show that proof-of-work in itself has to be questioned. Assuming an unbiased hash function, the following network properties can be defined:

$k \equiv$ system-wide replication factor
$b \equiv$ identifier length in bits
$S = 2^b \equiv$ size of identifier space
$c \equiv$ cost of generating a single identifier
$N \equiv$ actual number of nodes in the network
$d = \frac{S}{N} \equiv$ average shortest distance between nodes

Generally speaking, an (eclipse) attack targeting a single node requires the operation of *enough* attacker-controlled nodes *close* to the victim node. The relevant notion of *closeness* in this context amounts to malicious nodes being closer to the target than the closest honest node. Considering that any given distance to other nodes can occur only once from a given node's point of view, the probability of generating an identifier, which is closer to the target node than the closest neighbour, can be estimated as $\frac{1}{N}$ (following Eq. 1, given a sensible identifier space such as 2^{256}).

$$p(\text{close}) = p(\text{dist} < d) = \sum_{i=1}^{d-1} p\left(\text{dist} = i\right) \;=\; \sum_{i=1}^{\frac{S}{N}-1} \frac{1}{S} \;\approx\; \frac{S}{N} \cdot \frac{1}{S} \;=\; \frac{1}{N} \qquad (1)$$

As our design ensures disjoint lookup paths, k malicious nodes will need to be placed and operated close to an attack target to successfully eclipse it [1]. The probability $p(\text{atk})$ of obtaining k close identifiers in n trials can be calculated according to Eq. 2: Generating identifiers can be modelled as a Bernoulli process (following the first term of Eq. 2). However, the probability of generating an identifier multiple times (which would not increase the chance of a successful attack) also needs to be compensated for (following the second term of Eq. 2).

$$p(\text{atk}) = \underbrace{\left(1 - \sum_{i=0}^{k-1} \left(\binom{n}{i} \underbrace{\left(\frac{1}{N}\right)}_{p(\text{close})}^{i} \left(1 - \frac{1}{N}\right)^{n-i}\right)\right)}_{\text{probability to generate } \geq k \text{ close IDs}} \cdot \underbrace{\left(\prod_{i=1}^{k} \left(\frac{S}{N} - i + 1\right)\right) \left(\frac{N}{S}\right)^{k}}_{p(\text{alldifferent})} \qquad (2)$$

Considering the benchmark results from Section 3.2 and a price of 0.4 ct/h to achieve this performance[3], a single instance can process $\approx$ 2,400 messages (request–response workflows) per second. A single lookup will lead to $\mathcal{O}(k)$ many messages to be processed. As k is typically ≤ 20, the running costs for operating k malicious nodes can thus be virtually neglected, if the goal is to eclipse a single target.

More crucial is the cost of generating *enough close* identifiers. The plots shown in Figure 2 illustrate the success probabilities of generating enough nodes to eclipse a single target for multiple network sizes, based on 256-bit identifiers. We can observe that the probability scales almost linearly with the number of generated

[3] According to pricing information available at `https://www.hetzner.com/cloud`

identifiers between 0.1 and 0.9. Generating approximately $k \cdot N$ identifiers already results in a success probability of >50%, while generating $2k \cdot N$ suffices to succeed almost certainly as further illustrated in Figure 3.

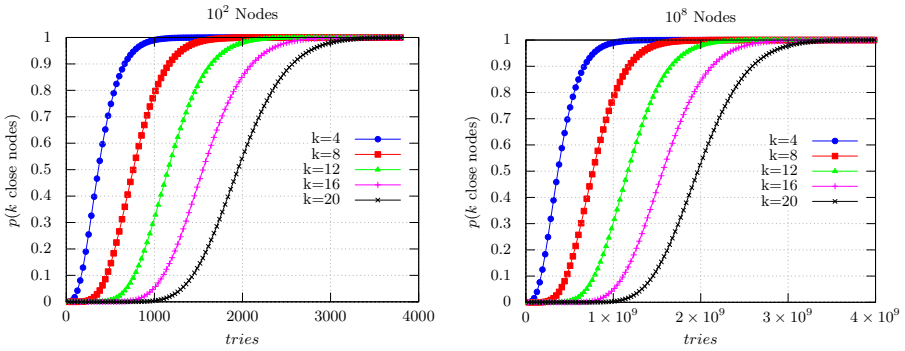

Fig. 2. Probability of generating k identifiers close to a target for different network sizes

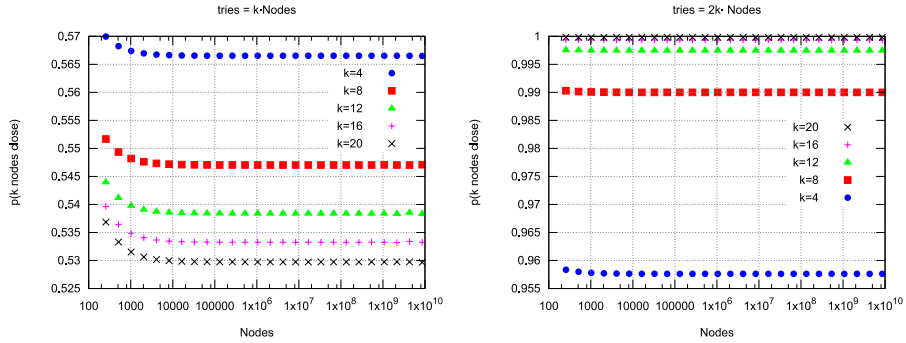

Fig. 3. Probability of generating k identifiers close to a target after $k \cdot N$ and $2k \cdot N$ tries for different network sizes

When mapping these figures to a network consisting of N nodes and the aforementioned prices of cloud computing resources, the actual cost of generating enough identifiers to eclipse a single node can be calculated. Matching the performance figures to current pricing leads to an estimated cost of 1 ct for generating 100 million identifiers. Consequently, the proof-of-work can be scaled to match different attacker budgets following Eq. 3, and Eq. 4.

$$cost_{\text{base}} = p(\text{atk} > 95\%) = \frac{2k \cdot N}{100\text{M}} \text{ [ct]} \tag{3}$$

$$\text{len}(\text{PoW-prefix}) = \log_2 \left\lceil \frac{B}{cost_{\text{base}}} \right\rceil \text{[bits]} \qquad B \ldots \text{Budget [ct]} \tag{4}$$

Increasing the cost of identifier generation without keeping in mind the time it takes for legitimate users to generate identifiers can lead to identifier generation taking too long for the typical, honest user. Therefore, it is more sensible to calculate the cost of an attack, given the amount of time honest users are willing to wait during identifier generation. Eq. 5-6 formalise this optimisation problem.

$$\max \quad cost_{\text{base}} \cdot factor = cost_{\text{atk}} \tag{5}$$

$$\text{s.t.} \quad \tfrac{1}{12000} \cdot 2^{\lceil \log_2 factor \rceil} \leq t_{\text{IDgen}} \tag{6}$$

To put these relations into context, we set $b = 256$, and $k = 20$ for a network consisting of $N = 1\text{M}$ nodes, while allowing a single identifier generation to take 1, 5, and 15 minutes, respectively. The results are depicted in Table 4 and scale linearly wrt. N. Clearly, hardening a fully decentralised P2P network solely using a proof-of-work based identified generation is not a viable solution, even though it is often praised as such [1, 2, 9]. This does raise some concerns, considering that our design imposes even tighter constraints than *S/Kademlia*, for example, and already restricts attackers as much as possible. Other designs, e. g. *SybilControl* [10], which mandate nodes to periodically solve proof-of-work challenges, also do not help against targeted attacks, as the cost of operating k nodes is still negligible.

At the other end of the spectrum are long-running, global attacks. The limiting cost factor in this case is not generating identifiers, but operating enough nodes for a prolonged period of time. Considering the performance figures, and a price of 0.4 ct/h, a single instance can process $\approx$ 1,200 messages per second[4]. Assuming 10 messages per second for a single operating node amounts to a cost of ≈ 33€/h to operate a million nodes. The time it takes to actually become resident in all nodes' routing tables is dependent on the churn rate and costs scale accordingly. Introducing an artificial overhead to increase running costs, as proposed by *SybilControl* is a questionable practice, especially considering the current device landscape, where mobile devices (like smartphones) with limited battery power are predominant[5] when it comes to Internet usage. In summary, we believe that essentially draining honest users' batteries to defend against adversaries is not a viable defence against long-running attacks.

The only other (noteworthy) decentralised alternatives to centralised identifier generation are based on the graph of the nodes' social network. As mentioned in Section 2, these approaches also do not stand the test of reality. Furthermore, countermeasures such as those against eclipse attacks inside the Bitcoin network [6], are either already part of our system, or highly specific to the layout of the Bitcoin network, and target P2P networks with low churn rates. Therefore, P2P networks in general are still susceptible to eclipse and Sybil attacks, especially when targeting mobile users, who can cause considerable churn rates. This effectively leaves centralised identifier generation as the only viable alternative.

[4] Each message requires two signatures to be verified

[5] `https://www.statista.com/statistics/306528/share-of-mobile-internet-traffic-in-global-regions/`

Table 4. Attack costs and work factor for prefix lengths based on the time allowed for identifier generation

	1 min.	5 min.	15 min.
len(PoW-prefix)	20 bits	22 bits	23 bits
$factor$	1,048,576	4,194,304	8,388,608
$cost_{\text{atk}}$	€ 4,194.30	€ 16,777.22	€ 33,554.43

6 Conclusion

This work presented a holistic approach towards P2P network security based on self-certifying identifiers. A comprehensive common-criteria-based security analysis as well as a performance evaluation demonstrate real-world applicability. Our authenticated routing protocol harnesses the full potential of self-certifying identifiers without introducing additional overhead or imposing complex behavioural constraints. All in all, this presents an advancement over existing designs, as network operation itself is kept as simple as possible while at the same time successfully defending against a variety of critical P2P attacks.

The second key finding presented in this work concerns proof-of-work-based strategies, often claimed to be effective against Sybil and eclipse attacks. We have demonstrated that such proposals simply do not hold up to real-world constraints, given the current prices of cloud computing power. This raises some concerns, since even proposals questioning proof-of-work as an adequate defence mechanism tend to simply propose different proof-of-work approaches.

The only other (significant) approaches towards tackling the problem of decentralised identifier generation are based on the graphs of nodes' social networks. It has been shown, however, that these mechanisms assume certain properties, which are not present in real-world P2P networks. This leads to identifier generation based on a trusted authority being the only defence that is known [4] to work.

References

[1] I. Baumgart and S. Mies. "S/Kademlia: A practicable approach towards secure key-based routing". In: *2007 International Conference on Parallel and Distributed Systems*. Dec. 2007, pp. 1–8.

[2] Juan Benet. *IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3)*. July 2014. URL: `https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf` (visited on 07/04/2017).

[3] Bram Cohen. *The BitTorrent Protocol Specification*. Oct. 11, 2013. URL: `http://www.bittorrent.org/beps/bep_0003.html` (visited on 04/24/2017).

[4] John R. Douceur. "The Sybil Attack". In: *Peer-to-Peer Systems*. Ed. by Peter Druschel, Frans Kaashoek, and Antony Rowstron. Vol. 2429. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2002, pp. 251–260.

[5] R. Fantacci et al. "Avoiding Eclipse Attacks on Kad/Kademlia: An Identity Based Approach". In: *2009 IEEE International Conference on Communications*. June 2009, pp. 1–5.

[6] Ethan Heilman et al. “Eclipse Attacks on Bitcoin’s Peer-to-Peer Network”. In: *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C., USA: USENIX Association, Aug. 2015, pp. 129–144.

[7] International Organization for Standardization. *ISO/IEC 15408-1:2008 Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model*. Geneva, Switzerland, Jan. 15, 2014.

[8] Don Johnson, Alfred Menezes, and Scott Vanstone. “The Elliptic Curve Digital Signature Algorithm (ECDSA)”. In: *International Journal of Information Security* 1.1 (2001), pp. 36–63.

[9] Brian Neil Levine, Clay Shields, and N. Boris Margolin. *A Survey of Solutions to the Sybil Attack*. Tech. rep. 2006-052. Amherst, MA: University of Massachusetts Amherst, Oct. 2006.

[10] Frank Li et al. “SybilControl: Practical Sybil Defense with Computational Puzzles”. In: *Proceedings of the Seventh ACM Workshop on Scalable Trusted Computing*. Raleigh, North Carolina, USA: ACM, 2012, pp. 67–78.

[11] Petar Maymounkov and David Mazières. “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric”. In: *Peer-to-Peer Systems*. Ed. by Peter Druschel, Frans Kaashoek, and Antony Rowstron. Berlin, Germany: Springer, 2002, pp. 53–65.

[12] David Mazières and M. Frans Kaashoek. “Escaping the Evils of Centralized Control with Self-certifying Pathnames”. In: *Proceedings of the 8th ACM SIGOPS European Workshop on Support for Composing Distributed Applications*. Sintra, Portugal: ACM, 1998, pp. 118–125.

[13] David Moore et al. “Inferring Internet Denial-of-service Activity”. In: *ACM Transactions on Computer Systems* 24.2 (May 2006), pp. 115–139.

[14] R. Moskowitz, P. Nikander, and T. Henderson. *Host Identity Protocol*. RFC 5201. Apr. 2008. URL: `http://www.rfc-editor.org/rfc/rfc5201.txt` (visited on 05/04/2017).

[15] National Institute of Standards and Technology. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. FIPS PIB 202. Aug. 4, 2015.

[16] Sylvia Ratnasamy et al. “A Scalable Content-addressable Network”. In: *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM ’01. San Diego, CA, USA: ACM, 2001, pp. 161–172.

[17] Atul Singh et al. “Defending Against Eclipse Attacks on Overlay Networks”. In: *Proceedings of the 11th Workshop on ACM SIGOPS European Workshop*. EW 11. Leuven, Belgium: ACM, 2004.

[18] Ion Stoica et al. “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”. In: *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM ’01. San Diego, CA, USA: ACM, 2001, pp. 149–160.

[19] Bimal Viswanath et al. “An Analysis of Social Network-based Sybil Defenses”. In: *Proceedings of the ACM SIGCOMM 2010 Conference*. New Delhi, India: ACM, 2010, pp. 363–374.

[20] Haifeng Yu et al. “SybilGuard: Defending Against Sybil Attacks via Social Networks”. In: *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM ’06. Pisa, Italy: ACM, 2006, pp. 267–278.