IPOS: Identity and Privacy Over SCION

Marinos Bernitsas
ECE
marinos@cmu.edu

Maria Lopez
INI
marialop@andrew.cmu.edu

Andrew Sarratori
ECE
newrad@cmu.edu

Yuqian Zhao
ECE
yuqianz@andrew.cmu.edu

Luca Zoia
INI
lzoia@cmu.edu

Abstract

SCION (Scalability, Control, and Isolation On Next-Generation Networks) [21] offers a replacement for the current Internet structure. As an increasing number of activities become digitized, many properties related to security and communication become increasingly desirable. Three important properties that are the focus of this paper include privacy, host identity and user identity. This paper proposes a number of schemes to achieve each property and offers a comparison between their implementation in the current Internet versus SCION. User privacy schemes that have been investigated include the Proxy Sub-TD method, Tor over SCION method and Persona + SCION method. For host identity, SCION is analyzed and proven to achieve this property through its design. User identity has been approached using Kerberized SCION and NamespaceID.

1 Introduction

As SCION is a well researched next generation networking structure, it is a natural starting point for further research into next generation protocols. The scope of this paper is limited to three major security properties: host identity, user identity, and privacy. For the purposes of this paper, privacy is defined as anonymity or masking ones identity. On the other end of that spectrum, host identity is defined as the ability to link data back to a particular machine or host. Finally, user identity is defined as the ability to link data back to a particular individual.

After surveying various identity and privacy protocols from other research, several approaches were decided upon that work and mesh well with SCION. First, an analysis of basic SCION demonstrates its native host identifiability. Next, two user identity protocols are analyzed that use SCION as a base. The first of these two user identity protocols is Kerberized SCION, which incorporates the session keys from Kerberos into SCION to achieve the desired property. The second protocol is called NamespaceID. It verifies and binds a user to a particular host when the user tries to set up a Diffie-Hellman key exchange. Lastly, three privacy protocols are proposed: one takes advantage of a feature inherent to SCION, one adds an application level anonymizer on top of base SCION, and the last also uses SCION as a base but adds another network level protocol on top of it. The first protocol is called Proxy Sub-TD and it hides both source and destination from potential snoopers, just not at the same time. The second protocol is Tor in SCION which uses onion routing as a way of providing source anonymity. Each proxy along the path can only know its previous hop. The last protocol is Persona in SCION. It uses some simple encryption to provide source anonymity and path obfuscation.

2 The State of the Internet

The current Internet offers neither identity nor privacy, as neither of those was a design requirement upon its inception almost 30 years ago.

Host identity is not offered because there is no way of verifying the identity of the host that is sending packets; even worse, that host may not exist or may be another host whose IP address is spoofed. Because there is no integrated way of disallowing a host to spoof the source IP address of the packets it is sending, there are a number of ways to offer such guarantees, such as ingress filtering, but they all introduce extra overhead and are not as effective. It is therefore imperative that any next-generation Internet approach eliminates IP address spoofing.

In the simplest case, user identity is offered by username/password authentication, and other application-level approaches (a more complete list is provided in Figure 1). Since most of the widely-used approaches are application-level protocols, they introduce significant overhead and potential for attacks, which has resulted in neither of the approaches (other than the default username/password authentication) being widely used. The inability of the current Internet to reliably provide user identity has hampered the transition of identity-sensitive services, such as contract-signing and e-government, to

the Internet realm. In addition, lack of user identity allows impunity when performing activities on the Internet, as people may simply claim that their packets were spoofed.

The current Internet does not provide privacy either, although its lack of identity is frequently used as a source of privacy. For instance, since there is no way of reliably linking a real-world identity to a virtual identity, Internet activities can be kept secret. As government and corporate surveillance on the Internet increase [11], it is important for there to be conduits for privately conveying information. There should be a way so that even if the government has control of the Internet - as is the case in several non-democratic regimes - people still have a conduit for privately communicating with the outside world. The emergence of sites like Wikileaks further underline the need for a host to send messages to a server without the server (or other hosts) being able to pinpoint its location simply on the basis of its Internet traffic.

While the current Internet has several applications that provide privacy, such as Proxies, Anonymizers and Tor, they have all been shown to have problems. Proxies require trusting the proxy service provider, while Tor has been shown to suffer from performance and throughput limitations. The need for the future Internet to provide source privacy in an efficient way becomes imperative.

3 Host Identity

3.1 Overview

One of the most problematic issues in the current Internet is that it is not possible to verify that a host, as identified by its IP address, is who it claims to be. Since the IP protocol allows spoofing of the IP source address, it is impossible to accurately verify that a packet was sent by a specific host. For this reason, people are forced not to rely on source address as a means of establishing a host's identity.

Lack of verification of the identity of a host is a serious problem in today's Internet for three primary reasons. First, organizations want to be able to check that hosts accessing their network are who they claim to be. Second, lack of host identity allows attackers to impersonate other hosts to achieve anonymity. Third, host identity allows us to trace attacks to individual hosts, which at least allows us to set filters for those specific hosts as a defensive measure.

Mobility presents additional challenges, because IP addresses are short-term identifiers that change as a device moves to different networks or is multihomed.

3.2 SCION

In SCION, each host is given an endpoint identifier (EID) which, along with the Autonomous Domain Identifier (AID), uniquely identifies it within a Trust Domain (TD). The great advancement of SCION, which comes from its use of Accountable IP [4] is that EIDs and AIDs are self-certifying, since they are the hash of the corresponding public key of that host.

Since it is good cryptographic practice not to re-use the same key for both signing and encrypting, some care should be taken to examine exactly what keys SCION provides us with. For example, the EID could be a hash of both a public key for encryption and a public key for signing, or one key could be derived through the other. Here, we assume that the EID is the hash of at least a public key for signing.

In addition, since the last 8 bits of the EID refer to the interface used, the same host can be identified even if it is on multiple interfaces, allowing easy mobility (from a wired to a wireless network). Since the EID stays with the host, it enables us to verify a host that is switching ADs. Some care should also be taken in examining how many EIDs a host can mint. AIP addresses this problem by having the AD provide some signature on the EID that binds it to a specific AD. In this way, an AD has to approve every EID that is minted, allowing it to impose quotas.

AIP allows us to establish host accountability without the use of any additional protocols. This is done on a per-router level by using unicast reverse path forwarding. The protocol ([4] §3.1) requires a host to attest its identity to every router along the way by signing a value provided by each router.

In order to offer host identity in an end-to-end fashion, we do not require the complex per-hop and caching procedures used for accountability. Instead, we show a simple message that a host (Prover) can send end-to-end to a Verifier in order to establish its identity.

3.2.1 Scheme

SCION and AIP assume that there is a secure lookup server where the public key that corresponds to each EID can be found.

By signing the payload with its private key, a prover P can attest its host identity to the verifier V.

$$P \rightarrow V : \{Digest(\text{Payload}), \text{Timestamp}, \text{Service Identifier}\}_{K^{-1}_{EID_P}}$$

If P has the EID it claims to have, it is the only one who has the private key necessary to sign the payload. We also include a timestamp (for freshness) as well as a Service Identifier, to prevent replay attacks for services with similar payloads. An example service identifier could be `secure.bank.com`.

While this exchange only establishes the host identity for a specific payload (e.g. login, wire transfer), further messages in the transaction could be authenticated by including in the signature a Diffie-Hellman half-key along with the payload, and having the verifier return its signed Diffie-Hellman half key. Subsequently, a secret key is setup that enables MACs to be used for future messages. Since each host signs its Diffie-Hellman values, the protocol is safe from man in the middle attacks. Using MACs enables fast authentication, because once the key has been set up, public key cryptography is not used. Figure 3 shows an example of such an exchange for a different protocol.

This scheme allows us to obtain host identity in an end-to-end manner that is lightweight compared to AIP's full host accountability protocols.

3.2.2 Attacker Model

The attacker here is anyone who is trying to spoof a host's identity (EID). This protocol securely identifies a host's EID unless the attacker is in possession of the victim's private key $K^{-1}_{EID_P}$. Note that replaying the identity establishment message will not work, as it has both freshness and is service-specific. As a result, an attacker will be unable to perform a replay attack.

It is important to mention that this scheme does not protect the host from malicious software or other vulnerabilities performing actions under its identity. Once a system is under the control of the attacker, the attacker can perform any actions on behalf of the host. However, since the attacker will be bound to a specific host, it is extremely easy to identify where an attack is coming from. Since TDs in SCION are legally coherent, legal penalties will serve as credible deterrents to attackers.

3.2.3 Evaluation

We have shown how SCION solves the big problem of Host Identity by use of Accountable IP, as a consequence of good design. We have also shown that if we only require end-to-end host identity, one signature is enough. Therefore, in this aspect, it is faster and more efficient than any add-on approach for the current Internet. For example, [12] uses DNSSEC and a complex system of reverse DNS lookups and key management to link a host's IP to a public key. SCION achieves this purely by design. We believe Host Identity is one of the greatest strengths of SCION, as it allows us to establish keys without the need for a PKI or trusted third party. We envision this feature to be the springboard for future innovation and simplification of public key schemes.

4 User Identity

4.1 Overview

User Identity allows us to identify and verify the user establishing a connection. As the current Internet does not provide any means to verify user identity, there are a number of protocols that aim to provide such features. Most approaches for user identity come in the form of application-layer protocols such as OpenID, Single Sign-On, Microsoft Passport, Microsoft Cardspace and OAuth. In Figure 1 we summarize the problems with each of the approaches. This shows that due to the design of the Internet, all approaches that have tried to establish user identity have significant problems that inhibit the provision of services that rely on identity, such as contract signing or e-government services.

As more and more activities transition to the digital realm, the need for a reliable user identity scheme becomes imperative. Below, we explore two approaches that demonstrate whether SCION can provide user identity in a way that is more reliable, more efficient and more scalable than the current approaches.

Our user identity schemes take advantage of the fact that SCION integrates the notion of host identity and build on it in order to link the user to the unspoofable host, thereby bootstrapping user identity on host identity. This allows us to verify user identity more efficiently, using lightweight protocols that should be easier to implement.

4.2 Kerberized SCION

4.2.1 Scheme

We propose an authentication protocol that implements trusted third party and a limited time ticket system called Kerberos v5 [15] for user authentication over SCION.

The basic scheme consists of users, a Kerberos Server that is equipped with an Authentication Server (AS) and Ticket Granting Server (TGS) and Service Providers.

Figure 2 provides a summary of the Kerberos Protocol.

The few assumptions considered in this scenario are that host identification is already provided by SCION (as detailed in section 4) and that the trusted third party is located inside the TD Core. Finally, Kerberos tickets are limited in both time and space. The latter is essential because by definition servers do not have common keys with other realms, and the first one is a security measure to minimize the exposure to the key being compromised.

4.2.2 Identity Offered

Kerberos is a distributed authentication system that has the ability to accurately identify the user making a request instead of by checking a password typed during

User Identity Protocol	Problems
Username+Password	• Can be stolen • Users don't pick hard passwords • Vulnerable to phishing attacks
OpenID, SSO, OAuth, Passport	• Time-consuming procedure (OpenID has 7-steps) • Web-specific • Login interface is vulnerable to spoofing attacks
Kerberos	• 6-step procedure • Requires infrastructure • Single point of failure • Limitations to an Internet-wide deployment
SSL client certs	• Not mobile • Expensive to obtain • Not widely accepted
Two-Factor Authentication	• Expensive to deploy • Losing the device means no access • Useless if private key is leaked (RSA)

Figure 1: A list of User Identity problems experienced in the current Internet

$$A \rightarrow AS : AS\_REQ\ (A, T_{a1}, lifetime_1, TGS) \quad (1)$$

$$AS \rightarrow A : AS\_REP\ (A, T_{a1}, exp\_time_1, \{K_{a\text{-}tgs}, TGS, exp\_time_1, \{Ticket_{a\text{-}tgs}\}K_{TGS}, T_{a1}\}K_a) \quad (2)$$

$$A \rightarrow TGS : TGS\_REQ: \{ Ticket_{a\text{-}tgs}\}K_{TGS} , \{ authenticator_{a\text{-}tgs}\}Ka\text{-}tgs , Ta2, lifetime2, B \quad (3)$$

$$TGS \rightarrow A : TGS\_REP: A, Ta2, exp\_time2, \{ Kab, B, exp\_time2, \{ Ticketab\}KB , Ta2\}Ka\text{-}tgs \quad (4)$$

$$A \rightarrow B : AP\_REQ: \{ Ticket_{ab}\}K_B , \{ authenticator_{ab}\}K_{ab} \quad (5)$$

where a*uthenticator* is composed of client ID and a checksum.

It is possible to do mutual authentication, with an extra message:

$$B \rightarrow A : AP\_REP: \{checksum_2 + 1\ \}K_{ab} \quad (6)$$

Figure 2: Kerberos Protocol

login. The identity is based in a tuple in the form of: $< primary, instance, realm >$

Where primary is the users identifier, the instance is an attribute of the user and the realm is the logical network used to distinguish among different authentication domains.

4.2.3 Attack Model

Despite its strengths and huge deployment, Kerberos has some weaknesses and limitations due to both design deficiencies and environmental issues that have been broadly covered in [5]. Some of the attacks that we need to take into account when deploying this protocol to assure identity over SCION are:

Secure Time Services Authenticators rely on machines clocks but many host use unauthenticated synchronization protocols. This characteristic can be exploited by an adversary but our hypothesis would be that machines already use synchronized ones.

Password-Guessing Attacks In particular, Kerberos is not resilient to password guessing attacks. The attacker can record login dialogs and make a guess confirmed by calculating the public key and using it to decrypt the record answer.

Spoofing Login Kerberos avoid clear-text passwords in the network; however an attacker could record users passwords before employing them in the Kerberos dialog. The disadvantage is that Kerberos protocol makes difficult to counteract with one-time passwords.

Forwarding Tickets Kerberos 5 allows ticket forwarding, introducing an important flaw: cascading trust. That is, a host may be willing to accept forwarded tickets originated in an insecure source. It is an undesirable feature of the newest version of the protocol.

4.3 NamespaceID

In this scheme we establish User Identity by providing *Identity As a Service* (IAaS). Each user is given a unique namespace of the form `user.provider.com`, which we call the *NamespaceID*. The user is responsible for managing and maintaining his namespace, so whenever he switches location he needs to update his namespace with the AID:EID of the host he is currently using. A user can have multiple NamespaceIDs (pseudonymity), for free or for pay, with different levels of due diligence performed to link this virtual identity to a real-world identity. A user may also have multiple AID:EID pairs under his namespace.

This approach borrows from OpenID [18, 17] the notion that each user is assigned a unique namespace, e.g. `user.livejournal.com`. This serves as the user's unique identifier to a service. We also borrow the notion of freedom of the user to select providers, and freedom of providers to offer differentiated levels of security and linkability to the real-world identities of users. As in OpenID, the user can have more than one ID.

This approach also borrows from [12] the notion that a (secure) nameserver can be used as a means of establishing user identity. In [12], the authors establish IPA (IP made Accountable) which uses DNSSEC to bind keys to hosts through reverse lookups. IPA is a rather complicated protocol that offers host accountability. NamespaceID, however, is a simple protocol that uses Forward DNS lookups to bind User Identity. The general idea is that since SCION performs lookups using a secure and trusted nameserver, which returns the AID:EID records under that domain name, we can use the nameserver as a trusted source for retrieving a mapping between a user's ID and the EID of the host(s) he is currently using.

4.3.1 Scheme

Our approach assumes the use of a trusted DNS system, such as DNSSEC. This assumption is shared by both AIP and SCION, therefore our protocol does not modify or break SCION.

Our approach also assumes the user is responsible enough to always keep his DNS records updated with the latest EIDs under his control. We believe this is a reasonable assumption, for three reasons. First, users typically migrate between a small number of machines, creating a small cost per-migration. Second, since interfaces on the same machine differ by only the last 8 bits of the EID, a user who is using one host but migrating across networks/interfaces (e.g. wired/wireless) only needs to update his Namespace once. Third, hosts switching from one AD to another already incur setup overhead per migration.

This procedure establishes user identity through the protocol shown in Figure 3.

Request When the prover is required to establish his identity, he sends his NamespaceID, a Diffie-Hellman half-key (optional), a digest of his payload, a timestamp (for freshness) and a service identifier to the verifier, and signs them with the private key that corresponds to his EID (EID_P) (step 1).

Verification Subsequently, the verifier performs a lookup on the NamespaceID (step 2) and retrieves the record for the user (step 3). First, the verifier checks that the AID:EID of the prover (from the packet header) is listed under the NamespaceID's records. Second, the verifier checks that the signature verifies correctly, the timestamp is within acceptable bounds, the service identifier matches its service and the digest of the payload is correct. At this point, the identity of the packet sent by the prover has been established (step 4).

Authentication If we wish to establish the identity of future packets in the transaction, the protocol allows for an optional step, where authenticated Diffie-Hellman half-keys are sent from the verifier to the prover (step 5). Once the prover receives the half-keys, both parties now have a shared key which can be used in a MAC for authenticating future messages. The message in step 5 is signed by a key K_V which is known to the prover. Since both half-keys are signed, we prevent any man in the middle attacks against the authenticity of future messages.

Namespace Management A bit more care should be taken to examine how the user maintains his namespace. Users have the option to choose how to maintain their namespaces. They can choose to update their EIDs and set timeouts (if they are using public computers, for example), or they can make the records permanent. We assume that the user can securely update his nameserver records using a unique key that is between him and his nameserver. Since this is offered as a service, users can choose their NameserverID provider according to their own criteria.

User Identity providers can offer a number of options to update nameserver records, including smartphones, where the user can scan the EID barcode of a public computer through his phone to update the nameserver record (since the public computer may not be trusted). Of course, this gives rise to the possibility of a tampered sticker.

Another approach is to use one-time records for signing in to public computers. The user may scan the barcode of the public computer to automatically create a one-time record on his nameserver. If a remote attacker tries to impersonate the user, the lookup will fail since the record is deleted after one lookup. By offering Identity As a Service, we encourage providers to innovate on how they can provide the most secure and usable expe-

$$P \rightarrow V : \{\text{user.provider.com}, \{g^s \text{mod} p, g, p\}, Digest(\text{payload}), \text{Timestamp}, \text{Service Identifier}\}_{K^{-1}_{EID_P}} \quad (1)$$

$$V \rightarrow \text{DNS}_{TD_i} : \{lookup \text{ user.provider.com}\} \quad (2)$$

$$DNS_{TD_i} \rightarrow V : \{ \text{ user.provider.com } resolves\ to\ \{AID_j : EID_k, \ldots\}\} \quad (3)$$

$$V : AID_P : EID_P \in \{AID_j : EID_k, \ldots\}, \text{Timestamp} < \text{Threshold}, Digest(\text{payload}) \quad (4)$$

$$V \rightarrow P : \{\{g^s \text{mod} p\}, T, \text{Service Identifier}\}_{K_V^{-1}} \text{ (Optional)} \quad (5)$$

Figure 3: NamespaceID User Identity Protocol

rience for updating a user's namespace.

Example Say a prover with a government-issued NamespaceID (`user.us.gov`) wishes to establish his identity to a verifier (`secure.bank.com`) for online banking. The user first visits the bank's website, where he is required to provide government-issued NamespaceIDs only. The user uses his smartphone to scan the barcode on the host and automatically set a 5-minute timeout or a one-time lookup record. He uses his NamespaceID as his username and the bank-specific password to log in. By use of the optional authentication step, all subsequent messages are authenticated.

If the user only wants to authenticate one message, e.g. a wire transfer, the Diffie Hellman half-key need not be calculated, and step 5 is also eliminated, speeding up the process.

4.3.2 Attack Model

The attacker here is someone trying to impersonate the user. This protocol is secure from impersonation as long as the following conditions are not simultaneously met: a) the attacker has control of the host or possession of its private key, b) the user has included the compromised host in his namespace (or the attacker has hijacked the namespace), c) the attacker has the user's service-specific password.

For example, if a user is on a machine controlled by the attacker updates his namespace to point to the compromised machine and enters his password, the attacker has all the tools necessary to impersonate the user. While one-time records do not allow the attacker to re-establish a new session with the verifier, there is no protection from an attacker who controls the machine to hijack the session and perform his own transactions. This problem is equivalent to someone logging into his bank account and then forced with a gun to perform several transactions.

One way to prevent this is to avoid the use of step 5 and use one-time records for every transaction. If the attacker performs a transaction on behalf of the user, the user will know because the legitimate transaction will fail. In order to avoid ever pointing a Namespace to an attacker-controlled machine, the user can either use remote attestation to verify that the machine is clean before using it, or the private key corresponding to the EID should be stored securely in a location where the attacker cannot directly access it. Last, the identity handshake could be performed on a dynamic root of trust, hence allowing the prover to make all calculations in an isolated environment.

DNS Hijacking Another attack possibility is someone trying to hijack the trust relationship between the NamespaceID provider and the prover, in order to advertise false EIDs (note that the attacker would also need access to a service-specific password to perform an impersonation). This attack can be performed by phishing the user's password or certificate that attests his identity to the NamespaceID provider.

There are a number of impediments to this attack. First, we assume the user will receive some notification every time an entry in his namespace is changed. Second, host accountability in SCION will serve as a deterrent to any attacker willing to do so. Third, since it is much easier to educate users on the security of one service (the NamespaceID service) we believe it is a reasonable expectation that users will be unlikely victims of phishing attacks. Fourth, since the service providers are free to choose how they implement their service, they may use additional security schemes whenever an EID is changed.

Attacking the NamespaceID Provider Another attack is someone trying to break into the NamespaceID provider itself. Since NamespaceID providers are faced with the sole task of providing IAaS (as opposed to the current model where the verifier is faced with the additional burden of storing identities), we assume that they will follow all the necessary safety practices. In addition, since the DNS system is so widely deployed and well-tested, we rely on its security to avoid attacks such as cache poisoning the verifier's DNS server (note that NameserverID records are never cached).

NamespaceID as an Attack Vector Another problem that needs to be addressed is the NamespaceID protocol becoming an attack vector itself, enabling computation DDoS attacks. Three defenses exist here. First, SCION can successfully launch legal deterrents to DDoS attacks. Second, a large chunk of the computation (signing) is performed at the host, making this protocol an unlikely target for DDoS. Third, since it is identity-based, the verifier can quickly flag specific packets and drop them, without performing the DNS lookups or any computation.

Privacy Another attack we have identified is a privacy attack, where an attacker can periodically query the nameserver for the current EIDs of the user and violate their privacy. Three defenses exit here. First, the EIDs do not necessarily reflect the location of the user, because a) unlike IP addresses, which can be mapped to specific geolocations, EIDs are mobile, so they can remain the same regardless of location; and b) because a user may mint a set of EIDs to confuse an attacker. Second, even if the EIDs do not conceal the user's location, a NamespaceID provider may offer a service whereby only specific user-authorized services may poll for his NamespaceID (e.g. `secure.bank.com`). Third, because of pseudonymity and varying degrees of linkability to a physical identity, the user may give different identities to different services.

4.3.3 Evaluation

Benefits NamespaceID is a relatively clear protocol and straightforward to implement. It does not require extra infrastructure or modifications to SCION. The use of the nameserver provides a standardized approach across the Internet, since every host knows how to communicate with a nameserver. It does not require the installation of any extra software on the client-side. In addition, while AIP [4] defines a set of complicated mechanisms for providing per-hop accountability, in NamespaceID we only rely on two features of AIP: the existence of a nameserver and self-certifying host addresses. Since NamespaceID requires user training for interacting only with a single standardized service to update their EIDs, it is much easier to train users to avoid social engineering and phishing attacks. By integrating into a well-tested and standardized vehicle such as DNS, we remove the security inconsistencies and vulnerabilities of letting the verifier perform the identification. Even if a user's service-specific password is stolen, the attacker would also need to compromise the machine that is under the user's namespaceID as well.

Implications Our approach creates a novel model for the provision of identity, in which providers can choose what services and levels of security they may offer. The only thing that remains standard is the user identity establishment protocol (Figure 3), allowing innovation and product differentiation. As a result, providers can perform different levels of due diligence and offer differentiated levels of linkability of the NamespaceID to a physical identity. For example, a government can offer its citizens government-issued NamespaceIDs in the same way a passport is issued (Estonia already gives its citizens OpenIDs).

We expect that this will enable a variety of applications that are currently not legally respected in the Internet, such as signing legal contracts, running e-government applications etc. It will also enable businesses to better manage users logging in from outside the corporate network, as well as offer a method for syndicated login across different domains.

In addition, NamespaceID allows passwordless authentication for services, by having the verifier check the NamespaceID and host signature, but not require a password. Therefore, a user who switches hosts may update his namespace with the new host and then enjoy access to services without using a password. This much-anticipated feature is made possible because of SCION's guarantees that host addresses cannot be spoofed. However, it is important to mention that a compromised host allows the attacker to impersonate the user to any service using passwordless authentication and launch transactions on his behalf.

Concerns We have identified a number of concerns with NamespaceID that may arise. Our approach is SCION/AIP-specific and therefore may not port well to other next-generation Internet approaches that are not based on self-certifying host names and secure DNS. It requires users to subscribe to a DNS provider, although this service can be free, similarly to how OpenID providers work. NamespaceID creates a one-time cost every time the user changes EID, although this may be streamlined from a user experience standpoint, as outlined in 4.3.1. NamespaceID records cannot be cached, which may require tweaks in the TTLs of the records on the verifier-side (only). We expect this protocol to increase the amount of DNS traffic in the Internet, although nameservers have always been focused on ensuring availability and over-provisioning.

Last, there is the question of whether to trust a NamespaceID that is issued by a nameserver in another TD. This is equivalent to whether one country trusts a form of identification issued by another country. We leave it up to each verifier to establish a list of which providers it accepts. For example, an immigration website may only accept government-issued NamespaceIDs, which is the equivalent of a country accepting passports. This enables foreign citizens to be instantly recognized in an-

other country, without having to re-establish their physical identity.

4.4 Comparison of Schemes

We presented two User Identity schemes, Kerberized SCION and NameserverID.

Kerberos is a robust and scalable system, but its deployment has several limitations that prevent its deployment Internet-wide. It uses a 6-step procedure compared to a shorter 3-step procedure required by NamespaceID. It relies on a trusted third-party and provides a single point of failure. Since nameservers are distributed by design, they are much more resilient to failure.

Kerberos requires extra infrastructure to be deployed, while NamespaceID uses the existing DNS system in SCION, so it is more attractive from a cost perspective. NamespaceID is also a lot better in terms of incremental deployment, since users can receive an instant security improvement regardless of whether they are early adopters or not. In contrast, Kerberos requires a large upfront investment to be made before any benefits can be enjoyed.

Since Kerberos uses only symmetric cryptography, it is faster than NamespaceID, which uses one signature verification and one signature generation (or two if we use the optional step). Since these are one-time costs (per session), we do not expect the user to significantly notice the timing differences.

We believe that Kerberos is better suited for deployment within one or more trusted ADs, since it allows a company more flexibility and control compared to NamespaceID. On the other hand, we believe NamespaceID is a better solution for Internet-wide deployment, with the potential to unleash a wave of innovation.

5 Privacy

5.1 Overview

Privacy has always been a major concern for users. Privacy is related to personal information and internet activity history. In many situations, users want to achieve anonymity, which means preventing third parties from being able to link a user's activities to personally-identifiable information. The following approaches attempt to provide desired levels of privacy by removing that linkability between users and their data. Since all of the following schemes use SCION as a base, the goal is to find a protocol that offers enough privacy while retaining enough of SCION's accountability.

5.2 Proxy Sub-TD

5.2.1 Scheme

As SCION allows sub-TD, so we use a sub-TD as proxy to provide privacy.

The basic idea is that and AD send a special packet to the proxy TD, whose payload is made up of the real destination and real payload of the packet. The whole payload will be encrypted with the symmetric key shared with the proxy TD. The real source and payload part is encrypted by the symmetric key of the source and destination pair. The proxy TD will decrypt the payload and get real destination and forward the packet.

plaintext	encrypted by the source and proxy symmetric key	
		encrypted by the source and dest symmetric key
header	**payload 1**	**payload 2**
similar to TCP header, with a special field to indicate its forwarding request packet. And the destination is to the proxy sub-TD	real destination	real source + real payload

Figure 4: Forwarding request packet format

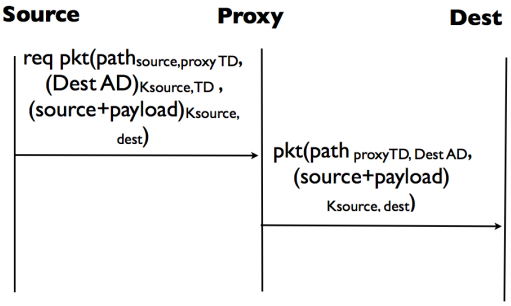

Figure 5: Forwarding process

5.2.2 Privacy Provided

This strategy provides one hop privacy, which means that the traffic can only shows that the source AD sends packets to the proxy sub-TD and proxy sub-TD sends packets to the destination AD. But within the AD, there is no privacy.

5.2.3 Attack Model

Traffic Analysis Attack The first attack can happen when there are not enough incoming hosts and outgo-

ing hosts. If there are m incoming hosts and n outgoing hosts, 1) if the attacker knows that the packet comes from the certain host, the probability of getting the outgoing host is $1/n$; 2) if the attacker knows that the packet goes to the certain host, the probability of getting the incoming host is $1/m$; 3) if the attacker just knows that there is traffic passing the proxy, the probability of getting the correct incoming and outgoing host is $1/(m*n)$. The attacker can also guessing the traffic flow depending on there are certain AD pairs communicating more often. The defense of the attack is to send dummy traffic to avoid traffic analysis.

Sub-TD Compromise Another attack is to compromise the proxy sub-TD. In this case, all the source and destination will be figured out. But the real payload of the packet is still safe, because it is encrypted by the symmetric key of source and destination AD. However, it needs to avoid to the proxy when setting up the symmetric key between source and destination AD.

5.3 TOR over SCION

5.3.1 Scheme

We use the Tor[8] idea for SCION. It is an application layer onion routing protocol. The source AD builds a virtual circuit of encrypted connections through Tor proxies on the network for a TCP stream. The source AD negotiates a separate set of symmetric keys for each hop along the circuit. Every proxy negotiate a circID with the next hop proxy. Every relay encrypted the entire content of the relay packet. So each relay along the way knows only which relay gave it data and which relay it is giving data to.

5.3.2 Privacy Provided

Tor over SCION has stronger forwarding privacy than single proxy, because the attacker needs to compromise all the Tor proxies to trace the real source and destination. It provides privacy per TCP stream. But it also has bigger latency, because it doesn't use short path.

5.3.3 Attack Model

Attacker can compromise the first and the last Tor proxy. Letting them to report the logs. By collecting the log information, the end can reconstruct the Tor path. The log content : (1) its location on the current circuits path (whether it is an entry, middle, or exit node); (2) local timestamp; (3) previous circuit ID; (4) previous AID; (5) previous connections egress interface; (6) next hops AID; (7) next hops ingress interface; and (8) next hops circuit ID. The attacker can link the source and destination.

5.4 Persona+SCION

5.4.1 Scheme

This scheme is an extension of SCION which attempts to add privacy to the packets exchanged across the domain. The goal of this extension is to offer privacy to the host, while maintaining the SCION infrastructure responsible for accountability. The concept proposed by this scheme is analogous to Personas idea of pseudonymity, which allows the user to hide their true identity, until some event is triggered by which a third party can reveal it. ([1], 412)

Two parts of SCION allow for host identification and path tracing, the EID and the Opaque field. The mechanism of this scheme attempts to hide those values as soon as possible to prevent sniffed packets from revealing identifying information. Host identification means a packet can be matched against a specific machine. The EID in SCION is an identifier for the host machine and is normally sent as plaintext, and as such any packet that is sniffed can be matched based on that EID. (Figure 6(A)) The first mechanism is designed to address this issue and has two possible implementations. In the first implementation, the source endpoint AD (AD1) encrypts the packets source EID field prior to forwarding it along the path using its own private key. (Figure 6(B))

When the packet reaches its destination, the destination host can send back replies by reversing the embedded path provided by the forward path. This provides privacy by obscuring the source EID from the destination. Additionally, once the EID is encrypted, the only way to discern the source is to sniff the packets from host machine directly or to compromise AD1. The second implementation for obscuring the source EID provides an additional level of privacy to the first implementation. The idea is that every AD along the path encrypts the current source EID field with its respective private key creating an onion layered encryption that grows with each hop along the path. (Figure 6(C)) Upon receipt of the packet, the destination copies the encrypted source EID field into the destination EID field of the reply packet and sends it back along the path in reverse. As each AD along the path receives the packet it decrypts the destination EID field, and continues relaying the packet back along the path. When the source endpoint AD receives the reply it can decrypt the destination EID field and send the packet to the correct host. Execution of this mechanism requires that a bit be added to the packet header to inform the AD as to whether it needs to encrypt the source EID field or decrypt the destination EID field. One limitation of source EID obfuscation is that by encrypting the source EID, the reverse path must match the forward path to guarantee that the reply packet returns to the correct sender.

(A)	(B)	(C)
Opaque Field	Opaque Field	Opaque Field
Dest EID	Dest EID	Dest EID
Src EID	E_{Ksh}{Src EID}	Forward: E_{Ki}{EID(i - 1)}
		Reverse: D_{Ki}{EID(i)} -> EID(i - 1)

Figure 6: (A) Original Method, EID in plaintext inside packet. (B) First implementation, EID encrypted using Ksh which is the source hosts secret key. (C) Second Implementation, EID is encrypted by each router along the path.

Forward Path Implementation	Reverse Path Implementation
Op Field(i) = E_{Ki}{Op Field(i - 1)}, Op Field(i)	D_{Ki}{Op Field(i)} -> Op Field(i - 1)
A -> B: packet, Op(A) B: val. Op(B), E_{Kb}{Op(A)} B -> C: packet, Op(B) C: val. Op(C), E_{Kc}{Op(B)||E_{Kb}{Op(A)}} C -> D: packet, Op(C) D: val. Op(D), E_{Kd}{Op(C)||E_{Kc}{Op(B)||E_{Kb}{Op(A)}}}	D: D_{Kd}{Op(C)||E_{Kc}{Op(B)||E_{Kb}{Op(A)}}} D -> C: resp. packet, Op(C) C: D_{Kc}{Op(B)||E_{Kb}{Op(A)}} C -> B: packet, Op(B) B: D_{Kb}{Op(A)} B -> A: packet, Op(A)

Figure 7: Formulaic definition of second mechanism and example exchange

The second mechanism attempts to obscure the path given by the Opaque field values so that a packet cannot be traced across the network. This mechanism extends the forward path validation to function in the reverse direction while obscuring as much of the path as is possible. In traditional SCION, during forward path traversal, the current router uses the previous routers Opaque value to verify its own Opaque value and validate the path in the packet is valid.

DEFINITION:

$$Op(i) = ingress_i||egress_i|| \\ MAC_{K_i}(ingress_i||egress_i||Op(i-1))$$

The current router then passes on its Op along with each of the previous Ops. The path hiding mechanism adds an encryption to this SCION method so that the previous Ops are obscured via the encryption once they are utilized for the path validation. When the current router receives the packet and Ops it validates and encrypts all of the previous Ops since they are no longer required for validation, resulting in a onion encryption where the first Op is encrypted N 1 times and the last AP is unencrypted. Because the last AP is unencrypted, when the destination attempts to traverse the path in reverse, it knows the first AP to which to backtrack. The current router then decrypts its onion layer of the encryption revealing the plaintext $Op(i-1)$. This can be used to validate the current routers Op for path validation. Finally, the router relays the relevant data to the next router in the reverse path. (Figure 7)

5.4.2 Privacy Offered

Implementing one of the variations of the first mechanism offers partial source host privacy. Since the path is not obscured, the packet can be traced back to the first AP but not to the specific host behind that AD. Additionally, this mechanism preserves most of the accountability that SCION provides natively. Implementing the second mechanism also offers some source host privacy through path obfuscation. In SCION, the EID is distinct inside each Trust Domain (TD), but EIDs can repeat across TDs. Since the Ops that define the path are encrypted as the packet advances through the path, the only way to trace a packet to its source is to follow the packet along the entire reverse path. This means that even though the source EID is exposed, as long as the packet crosses more than one TD, someone sniffing the packet wont be able to say for certain from where the packet came. If both mechanisms are utilized together then privacy is increased further.

5.4.3 Attack Model

The attacker model for the Persona + SCION scheme is similar to SCION alone. This means that defenses to attackers in this scheme match those in native SCION. However, the privacy gained by implementing the mechanisms of this scheme is dependent on the encryption protocol used. If the encryption protocols are weak and easily breakable then the attacker can perform any of the attacks on this scheme that are possible in SCION. Compromising an AD is a potential threat in both original SCION and this scheme. If an attacker compromises an AD along the path in this scheme, then he/she could inject packets that duplicate a packet thats already been

handled by the compromised AD. As such a slow-down or temporary-ban message could be a potential solution to this attack. Additionally, due to the encryption of the source EID, a compromised node along the path will still not be able to discern the source host. A compromised node would also be unable to discern any more of the path than the next step same as in original SCION.

5.4.4 Evaluation

This protocol offers several improvements over the current internet design. Pure SCION does not differ in packet overhead in a statistically meaningful way and the proposed changes in this protocol only add at most two encryption operations. This results in a delay of tens of microseconds or less at each hop along the path. As a result, a path would need to exceed a few thousand ADs to add a relevant lag to the path traversal time. Because this scheme is based on SCION, it offers many of the benefits over the current internet that SCION itself does. This protocol is resistant to source spoofing, data-plane attacks, reflection DoS attacks, etc. Additionally, it gains the accountability from SCION that is absent from the current internet. Finally, the privacy added by this scheme is greater than that of the current internet assuming no abuses or proxy use. With proxy use, this scheme is slightly worse than the current internet because the privacy of this protocol is at the AD level which allows the ISP to know which users are sending which packets.

5.5 Comparison of Schemes

The Proxy Sub-TD approach provides least privacy of the three approaches. This stems from its single point of failure, the proxy TD, and its requirement of handling all the traffic from the host that is utilizing it. A compromised proxy TD would reveal all that data passing through it. The Tor over SCION approach requires the sender to negotiate a Tor virtual circuit first. There must be a reliable number of Tor machines serving as relays around the world in order to support this protocol. It provides multiple hops to further increase privacy, however, since it is Tor based, it suffers from the performance issues and other faults inherent to Tor. Persona+SCION has to make changes to base SCION, which is potentially problematic, however, the changes only involve adding encryption schemes to the protocol flow and as such should not offer substantial problems if encountered. It provides the strongest privacy among the three approaches, because it addresses two causes of lack of privacy, however, it also restricts some of the accountability of SCION.

6 Future Work

This paper offers a first approach to provide both anonymity and identity by adopting concepts of previous research and using the strongest properties of SCION. The next iteration of this research would involve providing a greater level of anonymity and introducing destination obfuscation while not sacrifcing other properties. Attempting to find the ideal balance of desireable properties for the next generatiopn of protocols is the ideal goal of this research. Simulations would be another natural step forward to assure that these schemes are efficient and to determine possible pitfalls of their deployment. Additionally, future efforts to improve levels of efficiency, security and transparency to the user is another logic step foward.

7 Conclusion

The problems with the current Internet stem from a lack of foresight in designing the protocols to be resistant to abuse. No considerations were made with respect to IP spoofing, DNS hijacking, DoS and DDoS attacks, etc. As the next generation of the Internet draws closer, research is being done to find new protocols to solve the problems of the current Internet. Some of the base protocols discussed in this paper solve or attempt to solve one particular category of problems in the current Internet. SCION provides accountability and host identity at the cost of privacy, Persona offers source privacy while trying to afford some accountability, Kerberos is for authentication, TOR for anonymity, and NameserverID offers user identity.

The next iteration of protocols will need to offer a balance between all the desired security and communication properties. The protocols in this paper meld components of various proposed schemes together to gain properties that do not exist across the individual components. The protocols in this paper achieve some balance between some of the desired properties for the next generation Internet; however, there is still work to do. The goal of this paper is to offer another stepping stone along the path to the solution. These protocols offer insight and ideas for further researchers to reach the ideal solution.

References

[1] A. Agarwala, C. Johns, Y. Mallios, and S. Modi, *Persona: Network Layer Anonymity and Accountability for Next Generation Internet*, IFIP Advances in Information and Communication Technology (AICT) **297** (2011), no. 297, 410–420.

[2] Z. Ahmad, J.L. Ab Manan, and S. Sulaiman, *Trusted Computing based open environment user authentication model*, Advanced Computer Theory

and Engineering (ICACTE), 2010 3rd International Conference on, vol. 6, IEEE, pp. V6–487.

[3] D. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, *Holding the Internet accountable*, ACM HotNets-VI (2007).

[4] David G. Andersen, Hari Balakrishnan, Nick Feamster, Teemu Koponen, Daekyeong Moon, and Scott Shenker, *Accountable Internet Protocol (AIP)*, Proc. ACM SIGCOMM (Seattle, WA), August 2008.

[5] S.M. Bellovin and M. Merritt, *Limitations of the Kerberos authentication system*, ACM SIGCOMM Computer Communication Review **20** (1990), no. 5, 119–132.

[6] M.I. Chehab and A.E. Abdallah, *Architectures for identity management*, Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for, IEEE, pp. 1–8.

[7] R. Dhamija and L. Dusseault, *The seven flaws of identity management: Usability and security challenges*, Security & Privacy, IEEE **6** (2008), no. 2, 24–29.

[8] R. Dingledine, N. Mathewson, and P. Syverson, *Tor: The second-generation onion router*, Proceedings of the 13th conference on USENIX Security Symposium-Volume 13, USENIX Association, 2004, pp. 21–21.

[9] C. Farkas, G. Ziegler, A. Meretei, and A. L ”orincz, *Anonymity and accountability in self-organizing electronic communities*, Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society, ACM, 2002, pp. 81–90.

[10] J. Fontenla, M. Caeiro, M. Llamas, and L. Anido, *Reverse OAuth-A solution to achieve delegated authorizations in single sign-on environments*, Computers and Security.

[11] The Tech Herald, *Themis: Looking at the aftermath of the hbgary federal scandal*, http://www.thetechherald.com/article.php/201112/6951/Themis-Looking-at-the-aftermath-of-the-HBGary-Federal-scandal.

[12] A. Li, X. Liu, and X. Yang, *Bootstrapping Accountability in the Internet We Have*.

[13] X. Liu, A. Li, X. Yang, and D. Wetherall, *Passport: secure and adoptable source authentication*, Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, USENIX Association, 2008, pp. 365–378.

[14] J. Mirkovic and P. Reiher, *Building accountability into the future Internet*, Secure Network Protocols, 2008. NPSec 2008. 4th Workshop on, IEEE, 2008, pp. 45–51.

[15] B.C. Neuman and T. Ts'o, *Kerberos: An authentication service for computer networks*, Communications Magazine, IEEE **32** (1994), no. 9, 33–38.

[16] H.K. Oh and S.H. Jin, *The security limitations of sso in openid*, Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on, vol. 3, IEEE, 2008, pp. 1608–1611.

[17] D. Recordon and D. Reed, *OpenID 2.0: a platform for user-centric identity management*, Proceedings of the second ACM workshop on Digital identity management, ACM, 2006, pp. 11–16.

[18] D. Reed, L. Chasen, and W. Tan, *Openid identity discovery with xri and xrds*, Proceedings of the 7th symposium on Identity and trust on the Internet, ACM, 2008, pp. 19–25.

[19] S. Subenthiran, K. Sandrasegaran, and R. Shalak, *Requirements for identity management in next generation networks*, Advanced Communication Technology, 2004. The 6th International Conference on, vol. 1, IEEE, 2004, pp. 138–142.

[20] Y. Xiao, *Accountability for wireless LANs, ad hoc networks, and wireless mesh networks*, Communications Magazine, IEEE **46** (2008), no. 4, 116–126.

[21] X. Zhang, H.C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D.G. Andersen, *SCION: Scalability, Control, and Isolation On Next-Generation Networks*.

[22] Y. Zhang and J. He, *Identity Information Management in the Network Environment*, (2007).