SN Computer Science (2022) 3:350
https://doi.org/10.1007/s42979-022-01234-6

ORIGINAL RESEARCH

Ubiquitous Secure Communication in a Future Internet Architecture

Cyrill Krähenbühl[1] · **Adrian Perrig**[1]

Received: 4 October 2021 / Accepted: 25 May 2022

Abstract
In a world with increasing simplicity to store, transfer, and analyze large volumes of data, preserving data confidentiality and integrity of Internet traffic by default becomes more and more important. Unfortunately, a large gap exists between low-security opportunistic encryption and trust-on-first-use (TOFU) protocols, and high-security communication, such as TLS using server certificates or DNSSEC. Our goal is to reduce this gap and provide a base layer for authentication and secrecy that is strictly better than TOFU security. We achieve this by integrating the authentication method PILA into the future Internet architecture SCION. This combines PILA's address-based authentication, which leverages irrefutable cryptographic proof of misbehavior, and the flexibility of SCION's control-plane PKI and its per-AS independent addressing scheme. In this work, two concrete issues of PILA are addressed: (1) the reliance on the hierarchical RPKI which introduces a single global trust root, i.e., a single point of failure regarding the security of PILA, and (2) the necessity of an out-of-band communication to prevent downgrade attacks, which can incur a latency overhead and might be used as a resource exhaustion attack vector. We describe how PILA in combination with SCION mitigates these issues and analyze the security of the system. Finally, we discuss several interesting use cases including the SSH, TLS, and DNS protocols.

Keywords Authentication · SCION · Pervasive encryption · PKI

Introduction

The revelations from the Snowden case have shown that large-scale Internet wiretapping is indeed occurring. For example, in June 2013, the British Government Communications Headquarters (GCHQ) gained access to high-bandwidth fiber-optic cables to collect vast quantities of Internet traffic [1]. These incidents sparked not only the rise of opportunistic encryption protocols such as TCPCrypt [2], but also led to an increased deployment of TLS [3]. These two trends attempt to solve the same privacy problem, but in different ways. Opportunistic encryption improves the minimum achievable security guarantees by encrypting all communication with a low security protocol, while TLS and the Web PKI greatly improve the security of web traffic, arguably one of the most common uses of the Internet.

Compared to a few years ago, when a large portion of web traffic was sent unencrypted, in today's Internet, the most common way to fetch web content is via HTTPS which authenticates and encrypts traffic using the TLS protocol and the Web PKI. This gives users a highly secure authentication system based on certificates issued by certificate authorities (CAs). Security can further be improved by making use of DANE [4] and the DNSSEC PKI. One of the major factors for this development is Let's Encrypt [5], which solved previous deployment issues by providing free and automated server-certificate delivery. However, the usability of TLS is restricted when end-hosts do not have CA certificates; examples include communication modes other than client–server communication, such as peer-to-peer communication, or cases where setting up DNS entries for all devices is infeasible, such as in Internet-of-Things (IoT) settings. Additionally, DNS resolvers, SSH servers, and IoT devices are often identified by their IP address which precludes domain-based authentication.

Opportunistic encryption protocols, such as TCPCrypt or Opportunistic Wireless Encryption (OWE) [6], rely on a

This article is part of the topical collection "Information Systems Security and Privacy" guest edited by Steven Furnell and Paolo Mori.

✉ Cyrill Krähenbühl
cyrill.kraehenbuehl@inf.ethz.ch

[1] Department of Computer Science, ETH Zurich, Rämistrasse 101, Zürich 8092, Zürich, Switzerland

Published online: 28 June 2022

principle called trust-on-first-use (TOFU) [7] by assuming that the initial messages of a protocol have not been altered by an attacker. SSH [8] can be considered a TOFU protocol in the case when the other end-host's public key is not preloaded and the public key fingerprint is not verified during the initial connection [9]. Unfortunately, TOFU protocols only provide secrecy against off-path and passive attackers but not against an active on-path attacker, which can perform a man-in-the-middle (MitM) attack. On the other hand, TOFU protocols have many desirable properties; they are fast, easy to implement, and do not rely on external entities.

In our work, we focus on opportunistic encryption protocols and show how we can improve their security guarantees. Future Internet architectures, such as SCION, provide new opportunities compared to the current Internet.

DRKey [10] is a key distribution protocol in SCION. End-hosts can locally request pair-wise symmetric keys to any end-host in any autonomous system (AS). If two end-hosts simply encrypt their traffic or key exchange messages using these symmetric keys, the security of the communication is already substantially improved compared to a TOFU protocol, since on-path attackers are thwarted. However, the ASes providing these symmetric keys, i.e., the source and destination AS, can still decrypt traffic and perform man-in-the-middle attacks. To prevent such attacks, end-hosts should authenticate each other through asymmetric cryptography; each end host should generate its own private key and its AS should issue a certificate containing the end-host's respective public key.

Pervasive Internet-Wide Low-Latency Authentication (PILA) [11] is an authentication system designed to provide ubiquitous end-to-end authentication by issuing such certificates to end-hosts. PILA is not intended to replace existing strong authentication protocols, such as server certificates based on the Web PKI, but to provide an improved minimum level of security if strong authentication protocols are not available. PILA is incrementally deployable and incurs no computational overhead for intermediate nodes outside the end-hosts' ASes.

In this article, we propose to use PILA in combination with SCION. In comparison to the original PILA proposal, which makes use of the resource public key infrastructure (RPKI) [12] to provide authentication in the current Internet, PILA as described in this article makes use of SCION's control-plane PKI to provide authentication in SCION.[1] SCION's AS-based addressing and decentralized control-plane PKI bring significant improvements to the security and efficiency of PILA. Notably, PILA is improved as follows:

- There is no *single global* trust root which, if compromised, could be used to authenticate arbitrary end-hosts. Instead, each end-host is only required to trust the trust root of the local and remote isolation domain.[2].
- The addressing scheme of SCION allows each AS to use an independent local address space. This property is leveraged to provide in-band downgrade prevention by encoding PILA support in an end-host's chosen local address, reducing the connection setup latency and removing the dependency on external entities for downgrade prevention.

PILA works as follows: In a first step, the set of possible attackers is reduced. We observe that in the Internet, an AS provides a natural candidate to authenticate end-hosts based on their addresses, as the AS already offers connectivity to end-hosts and can thus identify them. PILA in combination with SCION's control-plane PKI reduces the attack surface to only the source and destination AS, since an end-host certificate can only be issued by the respective end-host's own AS.

In a second step, PILA provides accountability and proof in case either AS misbehaves. This guarantees a strong disincentive to attack, since after a MitM attack is revealed, the attacker is pinpointed by the system. In comparison, in a TOFU or the DRKey protocol, even if an MitM attack is detected, *any* on-path entity or either end-host's AS could have performed the attack, respectively. We can generalize this approach, which we call *trust amplification*, to three principles:

1. **Crude Authentication.** End-hosts are authenticated using a crude, relatively low-security approach which reduces the threat model to few entities.
2. **Accountability.** Misbehaving entities can be detected through cryptographic proof.
3. **Leverage.** End-hosts have means to apply pressure on misbehaving entities in the form of legal recourse, economic detriment, or bad publicity.

The *accountability* of entities trusted in the *crude authentication* in combination with the *leverage*, which disincentivizes misbehavior, inhibits coward attacks, i.e., attacks that are only launched if the attack will not be detected [13].

In summary, existing TOFU approaches reduce the set of possible attackers to only (active) on-path attackers. PILA (and DRKey) provide strictly stronger security properties by further restricting the set of possible attackers to only the ASes of the communicating end-hosts. Compared to DRKey,

[1] Henceforth, PILA refers to the SCION-based instantiation of PILA.

[2] An isolation domain is a grouping of autonomous systems (ASes) as described in "SCION"

PILA furthermore disincentivizes these ASes from misbehaving through cryptographic proof of any misbehavior.

This article is structured as follows. We first provide background on the relevant concepts of SCION and DRKey. Then, we introduce the trust-amplification model, which is a general model to increase the security of an authentication system. Next, we describe PILA, an instantiation of trust amplification in SCION based on the control-plane PKI and opportunistically trusted ASes providing end-host authentication. Finally, we analyze the security of PILA in SCION, present several use cases for PILA, discuss the contribution of this work, and finish with a brief conclusion.

SCION

SCION is a future Internet architecture which provides an alternative to BGP—the standard routing protocol used in the current Internet. In this section, we give a short introduction of the parts of SCION that are relevant for this article. For additional details, please refer to the SCION book [14].

Some of SCION's design principles are quite different from the current Internet. In today's Internet, routing decisions are made by the network nodes. The SCION architecture, however, follows a stateless packet forwarding approach, i.e., there is no per-flow state in routers and thus no opportunity for inconsistent state across the forwarding tables of the globally distributed routers. A forwarding path is defined at the granularity of autonomous systems (ASes), i.e., as a list of ASes including the incoming out outgoing border router. An end-host learns about available network path segments by querying a SCION service in its local network. The end host then creates an end-to-end forwarding path by selecting and combining these path segments according to its own preferences. When sending a SCION packet, the end-host includes the selected end-to-end forwarding path in the packet header, which allows SCION routers along the path to forward the packet accordingly.

On a side note, orthogonal to the properties of the security protocols that we discuss in this article, namely DRKey and PILA, even using a simple TOFU protocol, e.g., an anonymous Diffie–Hellman key exchange, benefits from the path control of SCION. Compared to the current Internet, a SCION end-host knows and controls which intermediate ASes handle its packets and can choose more trustworthy forwarding ASes based on a local preference and thus reduce, or at least clearly know, the attack surface.

Isolation Domains

In SCION, multiple ASes are organized into independent routing planes, called isolation domains (ISDs), which increase both the scalability and security of the network: On the one hand, ISDs enable a separation of the routing protocol into an intra-ISD and an inter-ISD process, which reduces the overall complexity [17]. On the other hand, by isolating the routing process within an ISD from external influence, ISDs limit the effect of misconfigurations and routing attacks.

Independent Address Namespaces

An end-host address in SCION consists of the ISD and AS number and the local end-host address. The ISD and AS number are used to locate the border router of the destination AS and the local end-host address is used to locate the end-host within the internal network of the destination AS. This separation of inter- and intra-AS routing leads to an important property of SCION, namely that each AS can have an independent local address namespace. PILA leverages SCION's addressing system and its independent namespaces to provide more secure and efficient authentication.

Control-Plane PKI

In SCION, all routing messages are authenticated based on a secure but flexible public-key infrastructure (PKI) in which each ISD can independently define its own roots of trust. The roots of trust have the form of a trust root configuration (TRC) which is a mutually agreed policy that is cross-signed by a set of privileged ASes within the ISD. In particular, the TRC policy contains a set of ISD root certificates, which are needed to update the TRC and elect ISD-internal certificate authorities (control-plane CA). Each AS is issued a control-plane AS certificate by such an ISD-internal CA certificate, which in turn is issued by a pool of ISD root certificates. Through this control-plane PKI, the SCION architecture provides strong resilience and security properties. In this article, we show how ASes can issue PILA end-host certificates based on the control-plane PKI.

DRKey

DRKey is a key distribution protocol used in SCION to efficiently derive symmetric keys between any two SCION end-hosts. The DRKey system requires a certificate service in the communicating end-hosts' ASes to derive DRKeys. DRKey works as follows:

- The certificate service of each AS locally generates a secret value.
- Each AS contacts each other AS and establishes pair-wise shared symmetric keys between them.
- Based on these pair-wise shared keys, each AS can derive a symmetric key between any two entities within

the ASes using efficiently implemented pseudorandom functions (PRFs).
- The end-hosts request symmetric keys to other end-hosts from their local certificate service which derives these keys on-the-fly.

As an additional optimization step, if it is known beforehand that one side of the communication is potentially under a higher load and must be able to derive symmetric keys by itself, the certificate service can delegate the key derivation to this entity. This works by providing the entity with an intermediate key in the key hierarchy that can be used to derive the end-to-end keys directly on the entity under load without contacting the local certificate service.

Since all end-to-end DRKeys for end-hosts within an AS can be derived by the certificate service of that AS, the certificate service can decrypt any message encrypted with these keys and forge any MAC created by these keys. Hence, if DRKeys are used to encrypt sensitive user data, the source and destination AS can decrypt these ciphertexts. If DRKeys are used to protect an anonymous Diffie–Hellman key exchange, the source or destination AS can inject their own keying material and forge a valid MAC to perform a man-in-the-middle attack. It is important to note that for the use cases of DRKey, such as preventing spoofing attacks or achieving network connectivity and availability, such attacks are ineffective, since the attacking AS would only cause damage to itself.

Trust-Amplification Model

PILA builds on a trust-amplification model, which is a certificate-based authentication model relying on three key principles: *crude authentication*, *accountability*, and *leverage*. Trust amplification provides a generic model to increase the security of a certificate-based authentication system indirectly by deterring misbehavior of involved certificate-issuing entities. The meaning of misbehavior depends upon the actual system used and typically means equivocating by issuing conflicting certificates. Trust amplification guarantees correct authentication if the certificate-issuing entities selected in *crude authentication* consist of curious-but-cautious (CuBC) attackers, which only launch coward attacks (i.e., attacks that cannot be detected).

The trust-amplification model makes several assumptions. First, we assume that there exists a single trust anchor, agreed on by the communicating entities, which provides keys and certificates to ASes. Second, we assume that participating ASes are able to authenticate their end-hosts. Third, end-hosts must have access to an authentic version of the trust anchor (i.e., the TRC of their ISD in SCION). Fourth, rough time synchronization with a precision in the order of a minute is essential for a certificate-based system with certificate lifetimes of several hours.

Crude Authentication

The first step is to significantly reduce the number of entities that can issue certificates for the communicating end-hosts to reduce the attack surface. Such a reduction is only meaningful if the certificate-issuing entities are not omnipotent, i.e., cannot issue certificates for arbitrary end-hosts. Ideally, the entities manage disjoint sets of end-host identifiers which reduce the certificate-issuing entities for an end-host to a single entity. In the trust-amplification model, the relying end-host establishes a trust relation to a certificate-issuing entity of the end-host that is authenticated, based on the following two principles.

Accountability

To increase trust into a certificate-issuing entity, which might initially be untrusted, certificate-issuing entities are held accountable for their actions. In the trust-amplification model, this property is achieved by generating irrefutable evidence that proves the misbehavior of a certificate-issuing entity. Important properties are resilience to slander (cannot forge false evidence) and framing (cannot manipulate an entity to produce false evidence itself), such that evidence is necessarily a result of improper behavior of a certificate-issuing entity.

Leverage

As a third principle, misbehavior must be disincentivized. After detecting misbehavior of a certificate-issuing entity M, other entities must have some form of leverage over M. For end-hosts that are issued certificates by M, leverage could be economic detriments through loss of customers or legal recourse. For other end-hosts, leverage could be a global or local trust rating of certificate-issuing entities based on collected evidence of misbehavior.

Trust amplification is similar to the approach used in certificate transparency (CT) [15], since misbehavior is deterred by providing cryptographic proof thereof. However, with trust amplification, the power of each certificate-issuing entity is restricted to a subset of identifiers (i.e., only the end-hosts within the AS), which is in stark contrast to the omnipotent highly trusted certificate authorities in the Web PKI with CT.

PILA

PILA provides authentication based on the end-host address[3] as an extension to existing protocols, such as TLS or SSH. PILA reduces the attack surface to the end-hosts' ASes, and produces proof of misbehaving ASes that create illegitimate certificates—e.g., to perform man-in-the-middle attacks on their end-hosts. The underlying protocol—which PILA extends to provide authentication for—must have (or must be extended to have) the property that an entity can authenticate itself using an X.509 certificate.

A PILA workflow where an initiator (relying party) *A* authenticates a responder *B*, works as follows. First, *B*'s AS uses its private key from the control-plane PKI (for which it has an AS certificate) to issue a short-lived certificate to *B* over *B*'s public key and end-host address. *B* uses this certificate to, for example, authenticate an SSH or TLS handshake. *A* verifies the authenticity of the handshake, or any signed reply in general, using the certificate chain starting at the TRC of *B*'s ISD. *A* also keeps track of the used control-plane AS certificates and end-host certificates locally or adds them to an append-only log. This retains the irrefutable proof of misbehavior, which can be detected through an out-of-band channel or an external auditor.

ASes as Opportunistically Trusted Entities

In PILA, trust anchors (TRCs) are axiomatically trusted similar to the root key in DNSSEC, but end-hosts interact with few ASes, which are only *opportunistically* trusted. ASes are not omnipotent, as they are identified by a unique AS number and implicitly by an ISD number corresponding to the TRC used as the trust anchor. One significant difference to RPKI-based PILA is that instead of a single trust root at the top of the RPKI hierarchy for all ASes, there is one trust root per ISD, i.e., per group of ASes. Misbehaving and compromised CAs and TRCs in an ISD thus do *not* affect the security of other ISDs. This property ultimately gives an AS more agency, since the AS can choose its ISD, i.e., its trusted entities. PILA uses ASes to bootstrap connection establishment, and then increases the trust placed into these ASes through trust amplification.

Each entity in the Internet is part of at least one AS, which is under the control of a single administrative entity. This facilitates providing a common service that authenticates end-hosts (e.g., using a challenge–response protocol or pre-installed keys and certificates) and issues certificates to end-hosts. Another advantage is the typically close relationship between an end-host and its AS, which allows for a stronger *leverage* in case of misbehavior. Since it is infeasible for an end-host to authenticate each AS by itself (there are ~73000 active ASes in 2022 [16]), the control-plane PKI is used as a trust anchor to authenticate ASes.

Using ASes as opportunistically trusted entities promotes an incremental deployment model of PILA, since there is the immediate benefit of end-hosts within an AS being able to authenticate themselves. The incremental deployment of PILA aligns well with the incremental deployment model of SCION [17].

End-Host-Address-Based Authentication

PILA authenticates end-hosts based on their end-host addresses. The benefit compared to name-based authentication, like domain names, is that all devices participating in Internet-wide communication have an end-host address and can thus make use of PILA. If an AS uses IP addresses as local end-host addresses, end-host certificates can be represented as X.509 resource certificates [18] to be compatible with existing PKI technologies. X.509 resource certificates add several extensions, most notably certificate policies [18] and IP-address and AS-number resources [19], which authorize subdomains to use these resources. End-host certificates are issued by the AS of the end host and contain a single local end-host address or a set of local end-host addresses delegated to this end-host.

The relying end-host constructs the chain of trust based on a given TRC and the control-plane CA and control-plane AS certificates and verifies it as explained in "Control-Plane PKI". In combination with the short-lived end-host certificate, the relying end-host can authenticate received messages.

End-Host Certificates

An end-host requests its certificate ($CERT_E$) from the local AS certificate service. An end-host certificate binds the public key of an end-host to a local end-host address that is unique within the AS. End-host certificates are typically short-lived on the order of hours to allow flexible address assignments without the necessity for revocation. In scenarios where a more dynamic address allocation is desirable, certificates can be issued with lifetimes on the order of minutes if the increase in certificate issuance overhead is acceptable.

Additional Local Identifiers

In addition to the local end-host address, end-host certificates might contain other (AS-)local identifiers, e.g., a username valid within the AS or a port range for which this certificate is valid; see "NAT Devices". To enable seamless

[3] An end-host address consists of the ISD and AS number and the local end-host address.

transitions between short-lived certificates, an AS issues multiple certificates with overlapping validity times as long as the public key and all identifiers are identical. An AS might refuse to add a local identifier to a certificate to protect itself against framing attacks if it cannot verify the correctness of the end-host's claim of the identifier.

Anycast

SCION introduces a dedicated service-addressing scheme for its control-plane services; the SCION service (SVC) address. An end-host can send a packet addressed to a service address without knowing the actual end-host address of the destination and the border router will locally resolve the service address and send the packet to the service instance. Anycast within an AS is then achieved by running a service from multiple locations in the network associated with a single SCION service address and let the border routers decide which service instance to forward the packet to. In the case that at least two service instances are located in the same AS, the service operator has two options: (a) all service instances can use the same end-host certificate and share a private key; or (b) the service operator requests a single end-host certificate for this AS and issues separate certificates for each service instance. The second solution reduces the impact of a private-key compromise but requires the additional intermediate certificate to be sent during connection establishment.

Certificate Service

All PILA-related functionality within an AS is handled by its certificate service. The certificate service generates and distributes short-lived end-host certificates ($CERT_E$) and, for that purpose, must be able to authenticate end-hosts within the AS. It also provides the control-plane AS certificates to end-hosts that are necessary for the verification of other end-hosts' certificates and signatures.

In PILA, issuing end-host certificates is an automatic process similar to the Automatic Certificate Management Environment (ACME) [20]. End-hosts are authenticated either (a) with a challenge–response protocol, which requires setting up an HTTP server on the client as in ACME [21] or (b) based on a signature of the certificate signing request (CSR) by the end-host.

While the challenge–response protocol is automatic and does not require a pre-existing trust relation between the certificate service and the end-host, the signature-based authentication allows managing certificates from entities other than the end-host. Additionally, ASes allow relying end-hosts to retrieve their local control-plane AS certificate including the certificate chain. The remote control-plane AS certificate including the certificate chain can either be fetched from the local certificate service or the remote end-host. Finally, to prevent downgrade attacks, the end hosts can request proofs from the remote AS whether a given end-host address supports a specific PILA protocol. "Downgrade Prevention" provides a detailed explanation of the different downgrade prevention mechanisms of PILA. These are the corresponding calls an end-host can make:

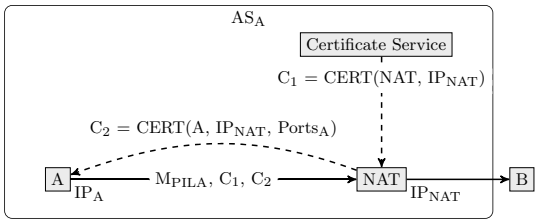

Fig. 1 The NAT device acts as an opportunistically trusted entity

- `getEPCert(local end-host address, [local identifier], public key)` either returns a short-lived certificate $CERT_E$ or an error message.
- `getASCert(ISD and AS number)` returns the control-plane AS certificate for the given ISD and AS number including its certificate chain. The certificate service fetches this information from an AS of the specified ISD if it is not cached.
- `getProof(local end-host address, protocol(s))` returns a signed statement whether the end-host at this local address has been issued an end-host certificate for the given protocol(s).

NAT Devices

In this section, we assume that IP is used as the network-layer protocol within an AS. There are two issues for end-hosts identified by IP addresses that reside behind NAT devices. Since PILA relies on the fact that ASes can identify end-hosts by their local end-host addresses (IP addresses) to distribute per-end-host certificates, a simple mapping of an IP address to a (single) certificate is impossible. The second issue is that due to the NAT device, both end-hosts have a different view of the opposite end-host's IP address, which breaks IP-address–based authentication. To authenticate end-hosts behind NAT devices, end hosts need to be able to use the public IP address of their NAT device as an identifier that is unique within the AS. We present an approach for authenticating between end-hosts with intermediate NAT devices.

The NAT device requests an end-host certificate (C_1) for itself (its public IP address) and acts as an opportunistically trusted entity, distributing certificates (C_2) to its end-hosts, as depicted in Fig. 1. End-hosts then authenticate

themselves by providing the NAT device's certificate (C_1) in addition to their end-host certificate (C_2) and the signed message (M_{PILA}). This requires end-hosts to trust the NAT device in the same way as an AS, i.e., a misbehaving NAT can be detected. The certificates issued to end-hosts behind the NAT device have the same IP address and additionally specify the outgoing port numbers as a local identifier as described in "End-Host-Address-Based Authentication". This allows other end-hosts to authenticate an end-host based on its public IP address and external port number (e.g., an end-host providing a service on a specific port can request a certificate which covers this port). Port numbers are encoded in an X.509 extension in the same way as IP addresses in resource certificates [19].

Multiple sequential NAT devices are supported as well. Each NAT device issues certificates for NAT devices within its local network, which can in turn issue certificates for end-hosts or NAT devices. Each nested NAT device thus requires an additional certificate. This isolates different hosts behind the NAT device and thus simplifies detection of misbehavior if a NAT device issues certificates with overlapping port number ranges for different entities. IPv6 solves IPv4 address shortage, one of the main reasons for the widespread deployment of NAT devices. We expect that with a growing IPv6 adoption, fewer NAT devices would be required and PILA deployment would become easier. In addition, SCION is agnostic to the intra-domain network-layer protocol, which can be chosen independently by each AS and thus facilitates the deployment of IPv6.

Session Resumption

If the underlying protocol supports session resumption, end-hosts can combine the session resumption with a PILA handshake and derive the keying material of the new session based on both sources. TLS 1.3 [22], for example, supports combining pre-shared key and certificate-based authentication to increase the security of a session [23]. The derived keying material is authentic if either the pre-shared key derived from previous keying material or the keying material produced by the PILA handshake is authentic and no secret values were leaked. Since PILA reduces the attack surface to the end-hosts' ASes, authenticated session resumption over different ASes increases the number of ASes that an attacker has to compromise to launch a successful undetected man-in-the-middle attack.

Downgrade Prevention

Whenever an initiator communicates with an unknown responder, an attacker might perform a downgrade attack to reduce the security to a less secure protocol (e.g., a TOFU protocol such as TCPCrypt). In such an attack, an attacker attempts to convince the initiator, i.e., the relying end-host, that either the responder's AS does not support PILA or that the responder does not support a specific PILA-supported protocol. This section presents several possible downgrade prevention approaches, which are further analyzed in "Downgrade attacks".

Signature-Based Approach

In this approach, the relying end-host fetches the necessary proofs for AS and end-host downgrade prevention from the certificate service of the local or remote AS, respectively. A fresh proof is generated and signed for each individual request from an end-host.

AS downgrade is prevented by locally keeping a regularly updated list at each AS containing all PILA-enabled ASes. End-hosts then request certificates for a specific AS from their local certificate service, which responds with a signed (possibly empty) list of certificates for this AS.

An AS that supports PILA must additionally provide proof that a service at a given end-host address does not support a specific PILA-supported protocol to assure the relying end-host that its communication is not being downgraded. An end-host sends a request including the end-host address, the PILA-supported protocol, and the current time as a timestamp. The certificate service replies with a signed proof that contains the hash of the request and a (possibly empty) list of certificate entries valid at the requested time. A certificate entry consists of the hash of the certificate and its validity period. The end-host then verifies the signature and that the returned list is empty before falling back to a non-PILA protocol.

Log-Based Approach

While the signature-based approaches for both the AS and end-host downgrade prevention method work well and are easy to implement, they have a large computational overhead due to the signature operation necessary to create each proof. A more elaborate approach that scales better to a large number of requests is organizing AS and end-host certificates in public append-only logs as in certificate transparency. The AS certificate log must provide a globally consistent view of all AS certificates, while the end-host certificate log can also be implemented as a separate log per AS. Each log is accompanied by a verifiable log-backed map [24], which provides a verifiable key-value store that can efficiently derive proofs of presence for a specific key-value mapping and proofs of absence for non-existing keys. The log and the log-backed map only require one signature operation per maximum merge delay (MMD) regardless of the number of requests. The log-backed maps allow end-hosts to fetch an AS certificate for an ISD and AS number and a list of

certificate entries from an ⟨ ISD and AS number, local end-host address, protocol ⟩ tuple.

Self-Verifiable Approach

In this section, we present two novel approaches which do not require any communication with the remote certificate service but require an AS to use a local IP address namespace. These approaches can improve the connection setup time when connecting to distant end hosts with a large RTT and significantly reduce the load on the certificate service, thus improving DoS resilience. In this approach, a relying party decides based on the IP address of the authenticating end-host whether the end-host supports PILA for a specific protocol.[4]

A simple approach is having a reserved range of IP addresses only for PILA traffic. This is applicable to both IPv4 and IPv6, but has the disadvantage that the address space becomes fragmented and thus potentially reduces the usefulness of prefix aggregation. Another disadvantage is that an AS has to reserve different IP address ranges for all possible PILA-supported protocols, which reduces the available address space. And as additional protocols start to support PILA, these reservations reduce the address space even further. This hinders the scalability of the approach, especially for IPv4 in the current Internet, which already experiences a shortage of IP addresses.

If an AS uses IPv6 as its local addressing scheme, then we can eliminate these disadvantages by having the end-host *itself* encode whether it supports PILA in the *variable* bits of the IPv6 address. This approach is inspired by cryptographically generated addresses (CGA) [25] in IPv6 which allow end hosts to encode their public key within the device address, i.e., by replacing the last 64 bit of the address. In PILA, the end host calculates the hash of the ISD and AS number, the fixed bits of the IPv6 address (the network and subnet address and the last 32 bit of the device address), and the supported protocol. The first 32 bit of this hash replace the first 32 bit of the device address (i.e., bits `[64:96]` of the IPv6 address). A relying end-host that connects to an unknown IPv6-enabled end-host using a non-PILA version of a protocol can verify that the address does *not* encode the protocol and that there is no downgrade attack.[5]

If a non-PILA-enabled end-host randomly chooses an IPv6 device address, it can happen that the first 32 bit are identical to the first 32 bit of the previously described hash output. A PILA-enabled end-host connecting to this end-host would not accept a non-PILA version of a protocol and abort the connection by falsely assuming a downgrade attack. However, if we assume that the first 32 bit of the hash output are randomly distributed, such a case is extremely unlikely with a probability of 2^{-32}.

While these two "self-verifiable" approaches are in theory applicable to both the current Internet and SCION, they only seems realistic in SCION, where each AS has an independent local address namespace and can thus easily implement a specific way of addressing without interfering with the global namespace.

Security Analysis

The goal of PILA is to provide an initiator (i.e., relying end-host) with an authentic X.509 certificate of a responder, in the presence of an attacker that can intercept, reorder, modify, and create arbitrary packets. The underlying protocol uses this certificate to derive an authentic key between the initiator and responder (session-establishment protocol) or to verify the correctness of a message signed by the responder (query–response protocol). PILA provides the initiator with an authentic certificate if the responder's AS is honest or a CuBC attacker and the initiator, responder, and global trust anchors are benign and none of these entities are compromised. In mutual authentication, both end-hosts act as responders. The goal of an attacker is to convince the initiator to accept a forged certificate by performing an MitM attack, by downgrading to a non-PILA connection, or by compromising a private key of a certificate in the certificate chain. Additionally, we analyze attacks on AS trust and denial-of-service (DoS) attacks.

MitM Attack

An attacker can perform a MitM attack to impersonate an end-host by providing a forged certificate to the initiator.

For protocols that establish secure sessions, this is done by intercepting the handshake messages and simultaneously creating two separate connections with the initiator and responder. For query–response protocols, the attacker modifies the response and possibly the signature within the response. If the end-hosts resume sessions as described in "Session Resumption", an attacker has to perform the attack on every session resumption to be successful and stay undetected.

Local Responder-Side Attacker. Attackers in the responder's local network are easily detectable, since the

[4] For the AS downgrade prevention, which does not require communication with the remote certificate service, either the signature-based or the log-based approach can be used.

[5] If both self-verifiable addresses and CGA should be supported, the AS could reduce the size of the IPv6 network prefix and let the end-hosts choose the additional variable bits between the shorter network prefix and the 64 bit of the device address to encode the PILA support.

responder can query either the certificate service or the local NAT, see "NAT Devices", and check for duplicate certificates for its identifiers.

Responder-Side NAT or AS Attacker. A malicious AS or a malicious NAT device on the responder side cannot immediately be detected. They do, however, create irrefutable cryptographic proof of misbehavior in the form of conflicting end-host certificates valid at the same point in time. These certificates can be stored locally or published on an append-only log server and later be compared through an out-of-band channel or audited by another entity.

Other Attackers. Other entities, such as a malicious AS or NAT device on the initiator's side or an attacker in the initiator's local network, cannot perform an MitM attack, since they cannot forge valid responder certificates.

Downgrade Attacks

In this section, we analyze the three downgrade prevention approaches explained in Downgrade Prevention. In a *downgrade attack*, an attacker attempts to convince the initiator connecting to an unknown responder that the responder's AS does not support PILA or that the responder does not allow the desired PILA-supported protocol. However, care must be taken that the downgrade prevention approaches do not introduce an additional *DoS vector* where a non-PILA-enabled end-host is prevented from communicating with a PILA-enabled end-host.

Signature-Based and Log-Based Approaches. Both the signature-based ("Signature-based Approach") and log-based ("Log-based Approach") approaches prevent downgrade attacks, since an attacker is not able to forge valid signatures for bogus statements which claim that a PILA-enabled end-host does not support PILA. Replaying a (potentially different) out-of-date statement is prevented by the time stamps within the statements and due to the assumption of time synchronization (see 3). For the same reason, an attacker cannot use an out-of-date statement which claims that a non-PILA-enabled host supports PILA as a DoS vector, since this statement will be rejected by the relying end-host.

Self-verifiable Approaches. We separate between the two self-verifiable address approaches explained in Self-Verifiable Approach: address range reservation and IPv6 address encoding.

If an AS reserves an IP address range for PILA-enabled traffic, then an attacker can neither downgrade (since the relying end-host can locally check whether the remote end-host is within the IP address range) nor use it as a DoS vector (since only PILA-enabled end-hosts are assigned to this IP address range).

For the self-verifiable IPv6 address encoding approach, an attacker cannot perform a downgrade attack since the two communicating end hosts will perform the same *deterministic* computation to verify whether the end-host has encoded PILA support in the IP address. Regarding a potential DoS vector, we consider two attackers: an on-path attacker which can and an on-path attacker which cannot influence the network prefix of the IPv6 address of an end-host. We assume the worst case, where the attacker can predict the device address that will be chosen by the end-host. The attacker's goal is to make the non-PILA-enabled end-host choose an IPv6 address that indicates PILA support.

- If the attacker cannot influence the network prefix and thus cannot impact the final IPv6 address chosen by the non-PILA-enabled end-host, the probability of a DoS for the non-PILA-enabled end host remains unchanged from the case without any attacker (2^{-32}).
- If the attacker can influence the network prefix and predict the device address, then the attacker could potentially fabricate a network prefix, such that there is a hash collision on the leftmost 32 bit of the device address. This would prevent the non-PILA-enabled end-host from communicating with a PILA-enabled end-host. However, it is very likely that an attacker with the capability of controlling the routing within the AS can simply drop unwanted traffic, which is in comparison a much stronger and more effective attack.

Private Key Compromise

The severity of a compromised private key depends on the entity and the lifetime of the certificate belonging to this key.

Key compromises of entities in the SCION control-plane delegation chain are relatively easy to detect if abused, since there would be ASes with multiple valid certificates for an ISD and AS number with different public keys. AS key compromises are similarly easy to detect but only allow forging signed PILA messages within the compromised AS. End-host key compromises are less severe, as end-host certificates are short-lived. In RPKI-based PILA, a compromised trust root impacts the authenticity of *all end hosts*. In comparison, a compromised (ISD) trust root in SCION-based PILA only impacts the authenticity of *end-hosts within this ISD*. Additionally, a single (or a few) compromised control-plane CAs can be removed from the set of trusted CAs by updating the trust root configuration (TRC) which specifies all control-plane CAs.

Attacking AS Trust

Attackers might attempt to reduce the trustworthiness of an AS. Slander, i.e., accusing a benign, uncompromised AS of having issued incorrect certificates, is not possible in

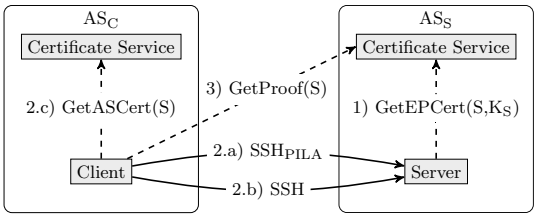

Fig. 2 SSH$_{PILA}$ connection establishment

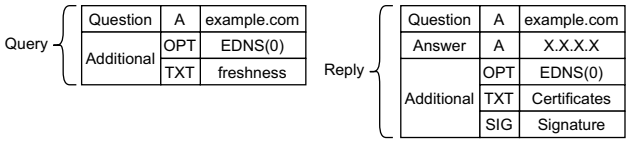

Fig. 3 DNS messages exchanged between client and its recursive DNS$_{PILA}$ resolver. The columns depict the DNS query section, resource record type, and resource record value

PILA, since an attacker does not possess the AS's private key and thus cannot forge certificates. An attacker might try to frame an AS by requesting incorrect certificates. Incorrect certificates could be certificates for local end-host addresses already assigned by the AS or certificates containing another end-host's local identifier. ASes prevent such framing attacks by verifying the correctness of an end-host's claim of identifiers as explained in "End-Host-Address-Based Authentication".

Resource-Exhaustion Attacks

Resource-exhaustion attacks attempt to overwhelm computational, storage, or network resources of an end-host or certificate service. An AS deploying a certificate service can perform ingress filtering to limit external DoS attacks and locate end-hosts performing DoS attacks in its own network. End-hosts can deploy typical DoS countermeasures for transport or application layer protocols; for example, DNS cookies [26].

Use Cases

We present three use cases for PILA, which cover a remote login protocol (SSH), a query–response protocol (DNS), and a general session-establishment protocol (TLS).

SSH

The SSH protocol allows clients to establish an encrypted and possibly authenticated session and open a terminal on a remote machine. The SSH protocol is frequently used to connect to machines identified by their local end-host, i.e., IP, address and thus is well suited to use PILA as a baseline for security. After initially connecting to a remote machine, SSH associates the machine's address and its public key, and thus, all subsequent connections are authentic if the initial connection was authentic. If a client either pre-loads these associations on their machine or verifies the fingerprint (hash of the remote machine's public key), then the authenticity of the initial connection is guaranteed. In many cases, a client simply accepts the provided fingerprint or fails to detect a difference between two long hexadecimal fingerprints [9, 27]. In these cases, PILA mitigates attacks by providing an association between local end-host addresses and public keys of hosts in PILA-enabled ASes when connecting to a remote machine from a client for the first time.

In SSH$_{PILA}$, instead of directly authenticating a server by its public key, servers are authenticated by their end-host certificates. This requires a slight change in the SSH handshake message format. Instead of adding its public key to the final SSH handshake message, a server adds its end-host certificate which contains the public key used during the SSH handshake.

Figure 2 shows the SSH$_{PILA}$ connection establishment with dashed lines indicating situational or periodic operations. The server periodically requests an end-host certificate from its AS's certificate service (1) and includes it in the final handshake message. The client initiates an SSH$_{PILA}$ handshake (2.a) which might be dropped by a non-SSH-$_{PILA}$ server. Concurrently to this handshake, to reduce the latency, the client initiates a regular SSH handshake in case the server does not support SSH$_{PILA}$ (2.b) and fetches the server's AS certificate if it is not cached (2.c). When the final handshake message is received, the client validates the received end-host certificate using its TRCs (trust anchors) and the server's AS certificate. If the SSH$_{PILA}$ handshake fails, but the server's AS supports PILA, the client additionally requests an explicit proof that the server does not support SSH$_{PILA}$ from the server's AS (3). In the case of using self-verifiable downgrade prevention "Self-Verifiable Approach", the third step is omitted. It is important to note that the client does not complete the regular handshake until the explicit proof is received to prevent leaking information such as login credentials to a MitM attacker.

DNS

In a large-scale survey by Chung et al. in 2017 [28], 88% of all DNSSEC-enabled recursive DNS resolvers returned supposedly DNSSEC-verified responses, without actually verifying the certificate chain. If the certificate chain from the DNS root certificate is not verified before an entry is cached, then DNSSEC does not provide any security and is vulnerable to the same attacks as regular DNS, while

providing a false sense of security. The issue seems to be that the resolver is not easily held accountable for the validity of the returned DNSSEC entries. DNS_{PILA} solves this issue by holding resolvers accountable for their DNSSEC responses. Auditors can use the signed DNS_{PILA} responses to prove recursive DNS resolvers's misbehavior (serving unverified DNSSEC-enabled responses) by verifying the DNSSEC responses themselves.

DNS_{PILA} adds freshness to DNS queries and returns signed DNS replies including necessary certificates as shown in Fig. 3. DNS is a prime example of the benefits of PILA as it is (1) unauthenticated, (2) interception and redirection of requests are widespread [29], and (3) DNS servers are mostly identified by their local end-host, i.e., IP, addresses. It is important to note that DNS_{PILA} operates between the client and resolver, unlike DNSSEC, where authoritative nameservers publish DNSSEC entries which are distributed by resolvers. In comparison to DNS over HTTPS (DoH) and DNS over TLS (DoT) which provide secrecy and authenticity, DNS_{PILA} provides authenticity and non-repudiation. DNS_{PILA} is thus to some extent orthogonal to DoH and DoT and could be encapsulated within DoH or DoT to additionally provide secrecy.

A client adds the following resource records (RR) to the DNS query: An EDNS(0) RR_{OPT} [30] to allow payloads larger than 512 B for transmitting the necessary certificates and a TXT record (RR_{TXT}) containing a random value to prevent replay attacks. The server detects PILA support by checking for a DNS_{PILA} RR_{TXT}. If DNS_{PILA} authentication is requested, the server adds an RR_{TXT} with the required certificate (chain) to the response. The server then adds the signature which is calculated over both the query and response in the form of an RR_{SIG} record analogous to a SIG(0) [31] signature. The client checks each field in the response and verifies the RR_{SIG} using the end-host certificate.

There is a privacy concern for disclosing malicious DNS servers as it reveals the browsing behavior of a user. A way to circumvent the privacy implications is by sending a denouncing DNS_{PILA} response to an auditor in a privacy-preserving way, e.g., via TOR.

TLS

We define PILA for TLS (TLS_{PILA}) as an example of a secure session-establishment protocol. Our goal is to achieve a baseline of security for persistent connections, which requires authenticating services by default, regardless of whether the service is identified by a domain name or a SCION address (i.e., ISD and AS number and a local address). An end-host uses TLS_{PILA} if neither a `TLSA` resource record nor a Web PKI certificate is available and the service is identified by a SCION end-host address.

In TLS_{PILA}, TLS 1.3 is modified to use PILA end-host certificates instead of certificates signed by the Web PKI and verify the SCION address instead of the domain name of the opposite end-host. Apart from the handshake, TLS_{PILA} is analogous to SSH_{PILA} except that since TLS requires a certificate, there is no fallback mechanism and thus no `GetProof` request.

Discussion

In this section, we discuss the differences between the RPKI-based PILA [11] and SCION-based PILA including the contributions of this work.

In SCION-based PILA, the trust root comprises the TRC of the isolation domain of both end-hosts. This ensures that a compromised control-plane PKI entity only impacts the security of end-hosts within the same isolation domain. The severity of private key compromises is thus significantly reduced compared to the RPKI-based PILA which relies on a global hierarchical structure with a single trust root.

Additionally, SCION-based PILA has a more flexible trust model, since an AS can join any isolation domain, e.g., in case an isolation domain does not conform to the desired standards of the AS. However, there are two caveats. First, it often makes most sense for an AS to join the same isolation domain as its providers and creating a separate isolation domain incurs administrative overhead that small ASes might not be willing to shoulder. This can limit the effective choice of isolation domain. Second, an end-host is often also limited in the number of ISP, and thus the number of isolation domains, to choose from. In December 2020, for example, only roughly 50% of the US population have broadband access with at least 25 Mbit/s with more than 2 providers [32].

Regarding performance, the novel self-verifiable downgrade prevention presented in "Self-Verifiable Approach" ensures that there is no out-of-band communication which can delay the connection setup or cause significant overhead at the certificate service. SCION simplifies the deployment of such self-verifiable downgrade prevention through its per-AS independent local address space. In the current Internet with its prefix-based routing architecture, addresses have to be unique in a global namespace which impedes the repurposing of the address semantics as it might conflict with other techniques such as prefix aggregation or subnetting.

Finally, a practical advantage of an SCION-based PILA is the simplicity of deploying PILA. SCION already deploys a certificate service per AS which can easily be extended to issue short-lived PILA certificates. The control-plane PKI is an integral part of SCION and allows the issuance of such certificates without major changes to the protocol. In comparison, RPKI

is still not ubiquitously deployed (although RPKI coverage has been steadily increasing over the last years [33]).

Conclusion

PILA in combination with SCION enables ubiquitous authentication for a wide range of devices and provides improved security properties compared to TOFU approaches and existing security protocols in SCION. This is achieved through the trust-amplification model in combination with SCION's flexible control-plane PKI. A variety of communication protocols, such as session-establishment (SSH and TLS) and query–response protocols (DNS), benefit from PILA. We show how PILA can leverage the inherent security properties of SCION and believe that it will further improve the security ecosystem of SCION by providing a strong, efficient, and easily deployable security baseline for many applications.

Funding Open access funding provided by Swiss Federal Institute of Technology Zurich. We gratefully acknowledge support from ETH, ZISC, and the European Union's Horizon 2020 research and innovation programme under Grant Agreements No 825310 and 825322.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. MacAskill, E., Borger, J., Hopkins, N., Davies, N., Ball, J.: GCHQ taps fibre-optic cables for secret access to world's communications . 2013 https://perma.cc/EP5N-V4U8
2. Bittau A, Hamburg M, Handley M, Mazières D, Boneh D.: The case for ubiquitous transport-level encryption. In: Proceedings of the USENIX Security Symposium . 2010 https://doi.org/10.5555/1929820.1929855
3. Let's Encrypt: Percentage of Web Pages Loaded by Firefox Using HTTPS. https://letsencrypt.org/stats/ 2022
4. Barnes R. Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE). RFC Editor, Fremont, CA, USA. 2011. https://doi.org/10.17487/RFC6394.
5. ISRG: Let's Encrypt. https://letsencrypt.org
6. Harkins (Ed.) D, Kumari (Ed.) W. :Opportunistic Wireless Encryption RFC Editor, Fremont CA USA. https://doi.org/10.17487/RFC8110 2017
7. Wendlandt D, Andersen D.G, Perrig A.: Perspectives: Improving SSH-style host authentication with multi-path probing. In: Proceedings of the USENIX Annual Technical Conference 2008. https://doi.org/10.5555/1404014.1404041
8. Ylonen T, Lonvick (Ed.) C.: The Secure Shell (SSH) Protocol Architecture. RFC Editor, Fremont, CA, USA. Updated by RFC 8308 2006. https://doi.org/10.17487/RFC4251
9. Gutmann P.: Do users verify SSH keys? Login **36** 2011
10. Rothenberger B, Roos D, Legner M, Perrig A.: PISKES: Pragmatic Internet-scale key-establishment system. In: Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIACCS) 2020. https://doi.org/10.1145/3320269.3384743
11. Krähenbühl C, Legner M, Bitterli S, Perrig A.: Pervasive Internet-wide low-latency authentication. In: Proceedings of the International Conference on Computer Communications and Networks (ICCCN) 2021. https://doi.org/10.1109/ICCCN52240.2021.9522235
12. Lepinski M, Kent S.: An Infrastructure to Support Secure Internet Routing. RFC Editor, Fremont, CA, USA 2012. https://doi.org/10.17487/RFC6480
13. Liu B, Chiang J.T, Haas J.J, Hu, Y.-C.: Coward Attacks in Vehicular Networks. ACM SIGMOBILE Mobile Computing and Communications Review **14**(3) 2010. https://doi.org/10.1145/1923641.1923654
14. Perrig A, Szalachowski P, Reischuk R.M, Chuat L.: SCION: A Secure Internet Architecture, 1st edn. Springer International Publishing, Switzerland 2017. https://doi.org/10.1007/978-3-319-67080-5
15. Laurie B, Langley A, Kasper E. Certificate Transparency. RFC Editor, Fremont, CA, USA. 2013. https://doi.org/10.17487/RFC6962.
16. Bates T.: CIDR Report. 2022 https://perma.cc/Y9BS-E5FV
17. Krähenbühl C, Tabaeiaghdaei S, Gloor C, Kwon J, Perrig A, Haussheer D, Roos D.: Deployment and scalability of an inter-domain multi-path routing infrastructure. In: International Conference on Emerging Networking EXperiments and Technologies (CoNEXT) 2021. https://doi.org/10.1145/3485983.3494862
18. Huston G, Michaelson G, Loomans R.: A Profile for X.509 PKIX Resource Certificates. RFC Editor, Fremont, CA USA. Updated by RFCs 7318, 8209 2012. https://doi.org/10.17487/RFC6487
19. Lynn C, Kent S, Seo K.: X.509 Extensions for IP Addresses and AS Identifiers. RFC Editor, Fremont, CA, USA 2004. https://doi.org/10.17487/RFC3779
20. Barnes R, Hoffman-Andrews J, McCarney D, Kasten J. 2019 Automatic Certificate Management Environment (ACME). RFC Editor, Fremont, CA, USA.https://doi.org/10.17487/RFC8555
21. Shoemaker RB. Automated Certificate Management Environment (ACME) IP Identifier Validation Extension. RFC Editor, Fremont, CA, USA. 2020. https://doi.org/10.17487/RFC8738.
22. Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3. RFC Editor, Fremont, CA, USA 2018. https://doi.org/10.17487/RFC8446
23. Housley R. TLS 1.3 Extension for Certificate-Based Authentication with an External Pre-Shared Key. RFC Editor, Fremont, CA, USA 2020. https://doi.org/10.17487/RFC8773
24. Eijdenberg A, Laurie B, Cutter A. Verifiable Data Structures 2015. https://perma.cc/7ZU3-B22V
25. Aura T. Cryptographically Generated Addresses (CGA). RFC Editor, Fremont, CA, USA. Updated by RFCs 4581, 4982 2005. https://doi.org/10.17487/RFC3972
26. Eastlake 3rd D, Andrews M. Domain Name System (DNS) Cookies. RFC Editor, Fremont, CA, USA 2016. https://doi.org/10.17487/RFC7873
27. Dechand S, Schürmann D, Busse K, Acar Y, Fahl S, Smith M. An empirical study of textual key-fingerprint representations. In: Proceedings of the USENIX Security Symposium 2016. https://doi.org/10.5555/3241094.3241110

28. Chung T, Van Rijswijk-Deij R, Chandrasekaran B, Choffnes D, Levin D, Maggs B.M, Mislove A, Wilson C. A Longitudinal, End-to-End View of the DNSSEC Ecosystem. In: Proceedings of the USENIX Security Symposium 2017. https://doi.org/10.5555/3241189.3241291
29. Liu B, Lu C, Duan H, Liu Y, Li Z, Hao S, Yang M. Who is answering my queries: Understanding and characterizing interception of the DNS resolution path. In: Proceedings of the USENIX Security Symposium 2018. https://doi.org/10.1145/3340301.3341122
30. Damas J, Graff M, Vixie P. Extension Mechanisms for DNS (EDNS(0)). RFC Editor, Fremont, CA, USA 2013. https://doi.org/10.17487/RFC6891
31. Eastlake 3rd D. DNS Request and Transaction Signatures (SIG(0)s). RFC Editor, Fremont, CA, USA 2000. https://doi.org/10.17487/RFC2931
32. Federal Communications Commission: Compare Broadband Availability in Different Areas. https://perma.cc/U9W2-4XSF 2020
33. NIST: RPKI Monitor. https://rpki-monitor.antd.nist.gov 2022