DCAuth: Data-Centric Authentication for Secure In-Network Big-Data Retrieval

Ruidong Li, *Member, IEEE,* and Hitoshi Asaeda, *Senior Member, IEEE,* and Jie Wu, *Fellow, IEEE,*

Abstract—Big data raises a strong demand on a network infrastructure to support the secure and efficient data retrieval with in-network caching. Information-Centric Networking (ICN) is an emerging approach to satisfy this demand, where big data are ubiquitously cached at the intermediate physical entities (IPEs) in the network and users retrieve the published data from the close copy holders. For the ICN, the unpredictability of users, IPEs, copy holders, and publishers during in-network big data retrievals poses a challenge to design a data-centric authentication mechanism to inhibit the malicious users to flood data requests and prevent the fake data from being cached and provided. However, the existing work only provides the authentications between users and publishers and suffers from the delay enlargement problem. To solve these problems, we design a trust model, namely a suspension-chain model (SCM), which is a trust chain that seamlessly merges certificate authority (CA)-based trust and neighbor-based trust. Based on SCM, we propose the DCAuth integrating certificate collection and packet forwarding, where the suspension certificate chain can be constructed for realizing any authentication to the unpredictable users/IPEs/publishers without accessing servers. Extensive simulations have been conducted to compare DCAuth with the existing work, which shows that delay can be greatly reduced and attacks can be efficiently prevented by DCAuth.

Index Terms—Big data, security, in-network caching, authentication

1 Introduction

Billions of people with mobile devices and small things, such as sensors, actuators, and robots, are generating tremendous amounts of data [1]. This is known as big data, and is characterized by five aspects: volume, variety, velocity, value, and complexity. Big data have attracted wide attention to develop business applications, such as the Internet of Things (IoT). One of the foundations for these services is to efficiently retrieve these big data [2], which is currently designed based on end-to-end communications within the Internet [34]. That is, most services are implemented based on centralized servers/clouds, and big data need to be distributed from the distant server/cloud to users, possibly through similar paths. Because of this, the current big-data retrievals effect large redundant and duplicate traffic, as well as large latency.

To cope with the retrieval of a huge amount of data, the network designs for big data [3] are indispensable, and the architectures supporting in-network big-data retrieval, such as Information-Centric Network (ICN) [5], [6], [36], [37], [38], and Named-Data Network (NDN) [7], have been proposed for content-centric applications. In these networks, big data are cached at the Intermediate Physical Entities (IPEs), such as routers, close to users for reducing delay and redundant bandwidth consumption.

However, in-network big-data retrieval leads the network to be seriously vulnerable to a variety of attacks [9], such as malicious-request attacks [10], [11], [12], and data-poisoning attacks [35], [13], [14]. In a malicious-request attack, adversaries impersonate users to flood data requests (or Interests), thereby causing the network to malfunction. In a data-poisoning attack, adversaries impersonate copy holders or publishers to provide fake data. This form of attack can quickly pollute the IPE caches as the virus spreads, because IPEs cache the fake data, redistribute them, and other intermediate IPEs re-cache them. It finally consumes much in-network caching storage and prevents users from retrieve the correct big data.

Combatting these attacks is much more difficult for in-network big-data retrieval than it is on the Internet, where the users and server(s) providing the data are pre-determined and end-to-end trust is easily established. Unlike on the Internet, the unpredictability with which copy holders provide big data, IPEs cache big data, and users request big data leads to great difficulty in inhibiting malicious-request attacks and data-poisoning attacks. To prevent cache poisoning, users and IPEs need to verify data before storing or caching them. If the data are found to be fake, the copy holder providing the data and the path to retrieve the data should also be discovered in order to disable the further spread of that fake data. To prevent malicious-request attacks, copy holders should verify the identities of the users. That is, the authentication from any user/IPE to the copy holders/publishers or from any IPE to users is required to be provided during the in-network big-data retrieval, which is simply called data-centric authentication in this paper.

The related work on secure in-network big-data retrieval can be classified as Internet Protocol (IP)-based solutions and ICN-based solutions. However, both approaches only provide the authentications between the users and publishers without considering data-centric authentication, and are unable to prevent the malicious-request and data-poisoning attacks. Furthermore, they rely on centralized servers to acquire certificates, thereby increasing authentication delays, which we refer to herein as the delay-enlargement problem.

To solve these problems, we propose a model of data-centric authentication with suspension chain (DCAuth) for secure in-network big-data retrieval. In DCAuth, packet forwarding and suspension certificate chain construction are seamlessly integrated

- *Corresponding author: Ruidong Li is with the Network System Research Institute, National Institute of Information and Communications Technology, Japan.*
  Email: lrd@nict.go.jp.
- *Hitoshi Asaeda is with the Network System Research Institute, National Institute of Information and Communications Technology, Japan.*
  Email: asaeda@nict.go.jp.
- *Jie Wu is with the Department of Computer and Information Sciences, Temple University, USA.*
  Email: jiewu@temple.edu.

to efficiently realize data-centric authentication. To the best of our knowledge, this is the first study to address the issue of trust establishment among unpredictable entities during data acquisition. It can also be widely applicable for secure routing and secure transport during in-network big-data retrieval. The following new properties distinguish the present study from existing works.

- A suspension-chain model (SCM) is proposed as the trust model, where the neighbor-trust-based certificate chain is suspended by certificate authority (CA)-based trust. It fundamentally enables the realization of data-centric authentication.
- Forwarding-integrated hop-by-hop certificate collection together with the adaptive replacement for parts of chain with highly trustworthy certificates is proposed to construct the trustworthy suspension certificate chain based on SCM. It avoids reliance on centralized server(s) for chain construction and solves the delay-enlargement problem.
- DCAuth smoothly extends the authentication from the physical entities to the logic entities. It breaks the barrier between networking and big-data applications, which follows the data-centric approach.

Security analysis shows that DCAuth can satisfy the security design requirements. Extensive simulations show that DCAuth greatly reduces delays compared to the existing PKI-NDN scheme, and it can also efficiently prevent the malicious-request and data-poisoning attacks.

The remainder of this paper is organized as follows. In Section 2, we discuss related work. In Section 3, we provide the problem statement. In Sections 4 and 5, we introduce DCAuth and analyze its security properties, respectively. In Section 6, we provide performance evaluations. Finally, we conclude our work in Section 7.

2 Related Work

In-network big-data retrieval enables data acquisition from nearby caches instead of centralized servers [6], [7]. Recently, ICN has been identified as a promising approach to achieve this. Rather than emphasizing the communications among hosts as in IP networking, communications in ICN are achieved via a pull-based data-retrieval model. With the ICN, IPEs are equipped with cache memories and big data are ubiquitously cached in the network. Each item of big data is signed by the publisher, allowing users to verify the integrity and data-origin regardless of the source (either the publisher or an IPE) that provided the data (or its copy) [7].

However, in-network big-data retrieval with ICN has been identified to suffer from malicious data-request attacks [10], [11], [12] and data-poisoning attacks [35], [13], [14]. In the former, adversaries impersonate users to create a flood of interests, and in the latter, they impersonate copy holders (e.g., IPEs or publishers) to provide fake data. These attacks are severe, because big data are cached in a distributed manner, and copy holders have no way to verify users' identities, and users/IPEs have no way to verify copy holders' identities to avoid caching and spreading fake data.

To inhibit these attacks, data-centric authentication should be provided to support the secure data retrieval, which securely retrieves in-network cached big data from the unpredictable copy holders to the users. It enables data copy holders to provide the data only after successfully authenticating the users and enables users to verify the data and copy holders after receiving the data.

The related work on authentication can be classified as IP-based solutions or ICN-based solutions. The IP-based solutions involve authentication over Internet, such as SSL/TLS [15]. These rely on two approaches for trust establishment, namely the certificate authority (CA)-based approach [16], [17], [18] or the web of trust (WoT)-based approach [19], [20], [21], [22], [23]. In the former, a trusted third party, namely the CA, is introduced to issue, manage, and provide certificates. Typical CA-based protocols include X. 509 [16] for securing applications, and DANE [17] and ARPKI [18] for securing networking. In contrast, WoT is based on an "introducer model" that depends on a chain of authenticators [20]. The typical WoT-based protocol is PGP [19], and there are also many variations for ad hoc networks [21], [22].

However, both CA-based and WoT-based approaches rely on entities discovering and retrieving the relevant certificates or certificate chain(s) from the certificate repository(s). Thus, they suffer from the delay-enlargement problem because of such additional certificate acquisitions. In addition, the end-to-end design enables authentication only between users and publishers. However, this design is especially difficult for IPEs, which forward packets, to authenticate publishers or copy holders before caching the data, which makes it easy for fake data to be cached and spread. Furthermore, for WoT-based approaches, trust degrades with the length of the certificate chain, because certificates can become compromised [23]. This can cause authentications to be performed incorrectly.

For ICN-based solutions, there is a set of designs to provide authentications for NDN [24], [25], [26]. The core of the designs is a public-key authentication protocol for NDN, PKI-NDN [24]. It relies on designated hosts to store and provide certificates. In addition, an NDN version of DNSSEC, namely NDNS [25], and an NDN version of domain-based CA design [26] have been developed to assist in the discovery of the relevant designated host(s) and securing routing, respectively. A trust rule for deriving key names from data names [39] is also defined. This set of solutions automates and embeds the authentications in data-centric networking. However, these authentications are restricted between users and publishers, and thus cannot prevent fake data from caching or enable copy holders with fake data to be identified. Moreover, they need to retrieve certificates from designated hosts for authentications and thus suffer from the delay-enlargement problem.

3 Problem Statements

3.1 In-Network Big-Data Retrieval

Consider a system of in-network big-data retrieval with ICN/NDN [7]. The IPEs are equipped with caches, and the data can be cached by the IPEs. The packets for data retrieval are fundamentally *Interest* and *Data*. Each IPE has three main data structures [7]: a forwarding information base (FIB) for forwarding Interests, a content store (CS) for caching data, and a pending Interest table (PIT) for forwarding data. To retrieve big data, a user asks for data by throwing the Interest packet with the data name over the available connectivity. Any entity that receives the Interest and has the big data that satisfies it can respond with a Data packet. This entity can be the IPEs or the publishers.

The system is composed of the entities as follows.

- **Publisher**: an entity that publishes data in the network.
- **User**: an entity that retrieves data from the network.

- **Physical entity**: an entity that communicates using a physical device. This could be an IPE or a publisher node (PN) that hosts applications.
- **Logical entity**: an entity that is involved in an application. This can be an authorizer, a sub-authorizer, or a publisher.

Take a typical big-data service portal, *Travel*, as an application example to introduce the use scenario for in-network big-data retrieval. In this application, the authorizer administrates the portal, *Travel*. It assigns privilege to the sub-authorizers to manage the sub-categories. For example, the authorizer of *Travel* assigns the sub-authorizer, SA_1, the privilege of managing a sub-category, *Travel/CityA*. This sub-authorizer further assigns privilege to another sub-authorizer of managing the sub-category in its service domain, or assigns publication privilege to a publisher. SA_1 might assign a publisher P_X the privilege to publish data in the sub-category *Travel/CityA/Temp*. Therefore, P_X could publish the temperature data of City A on Apr. 13 2018 for a travel service entitled *Travel/CityA/Temp/V20180413* on this service portal.

To retrieve the aforementioned temperature data, user U_X sends out an interest to request data with the name *Travel/CityA/Temp/V20180413*. When an IPE receives this interest, it searches its cache. If nothing is found, the interest is forwarded according to the FIB. Finally, the interest reaches the IPE or PN that holds the relevant big data. The IPEs along this path can cache these big data.

During this procedure, the copy holders (IPEs or publishers) who reply with the big data should authenticate the identity of U_X; IPEs should verify the data that they forward before caching them, and U_X should verify the data it retrieves.

3.2 Adversary Model & Design Requirements

For in-network big-data retrieval, we consider both outsider and insider adversaries. An outsider controls only a set of malicious physical entities (e.g., IPE, PN) or logical entities (e.g., sub-authorizer, publisher). These malicious physical entities are deployed without establishing initial trust, whereas malicious logical entities behave without privileges. An insider controls a set of physical or logical entities that are already part of a trusted application context, as well as optionally a set of entities that are not part of the trusted context.

An adversary can perform the following attacks: (A1) impersonates a copy holder to provide fake data; (A2) impersonates a user to request big data; (A3) compromises entities to provide the authenticable certificates together with fake Interest or data. In attacks A1 and A2, an outsider impersonates entities, whereas in attack A3, an insider uses compromised entities. The goal of these is to flood requests or provide fake data via malicious-request or data-poisoning attacks.

Because of the insiders, the length of the certificate chain is important when using the WoT approach. A shorter chain reduces the number of entities that another entity must trust on the path, and thus increases the trustworthiness of the key [23].

To inhibit the adversaries, we identify the design requirements of the authentication mechanism as follows.

- **Data-centric design**: Any IPE or user can easily authenticate the data, publisher, and copy holder, and any copy holder can easily authenticate users. This is different from the existing authentications only from users to publishers, as IPEs need to verify the data before caching them, copy holders with fake data should be identified quickly to prevent further data provision and spreading, and copy holders need to verify users identities before providing data.
- **Attack prevention**: The proposed protocol should prevent malicious-request and data-poisoning attacks and restrict the influence of any compromised certificate(s).
- **Revocation of compromised certificates**: If a certificate becomes invalid, it should be revoked from use.
- **Availability**: Certificates should be provided anytime and anywhere, even if part of the network is down.
- **Low bandwidth consumption**: The infrastructure should not increase the packet size substantially and should have a negligible impact on bandwidth consumption.
- **Minimal additional delay**: The infrastructure should cause minimal (ideally zero) additional delays (e.g., because of extra certificate requests) to big-data retrieval.

4 DCAuth Designs

Herein we elaborate the design of DCAuth. We start with the trust model design and then present the three phases of DCAuth: 1) initial trust-establishment, 2) data-centric certificate management, and 3) forwarding-integrated authenticable data-retrieval. For the first phase, the CA issues certificates to a set of IPEs to form the highly trusted IPE group (HTIG). Meanwhile, certificates are issued for neighboring entities. In the second phase, IPEs exchange certificates within their neighborhoods, and any compromised entity can be shut down quickly. Finally, the third phase provides a hop-by-hop method for constructing a suspension chain consisting of a physical-entity certificate chain (peCEChain) and a logical-entity certificate chain (leCEChain) for data-centric authentication.

4.1 Trust Model: Suspension Chain Model (SCM)

We propose a suspension-chain model (SCM) as the trust model in DCAuth. The SCM is a flexible series of neighbor-trust-based certificates suspended by CA's trust, which form a suspension chain. Traditionally, WoT-based trust is difficult to merge with CA-based trust, because WoT usually reflects uncertain social relations and thus the suspension points cannot be easily planned. However, for in-network big-data retrieval, the topological neighbors' trust relations are predictable, which makes it feasible to plan the setting of suspension points.

SCM absorbs the merits and limits the demerits for traditional WoT-based and CA-based trust. WoT-based trust provides a certificate to the entity with direct trust relations in a distributed manner, which suffers from the trust degradation problem [23]. In contrast, a CA-based trust provides highly trustworthy relationship with a high maintenance cost for certificates by the centralized CA. In SCM, neighbor-based trust forms the certificate chain to realize data-centric authentication, whereas CA-based trust reduces the length of the chain to solve the trust degradation problem.

For CA's trust, the CA assigns certificates to highly trusted IPEs as the suspension points based on CA's trust, which is the pre-trust between these IPEs and CA. According to the CA's suspension trusts, these IPEs then issue certificates to the nearby highly trusted IPEs. The CA selects highly trusted IPEs based on their security properties and locations. For example, a rule is set that the distance between two nearby highly trusted IPEs in the HTIG should be less than the fixed number of hops, such as 5

hops. This rule restricts the chain based on neighbor-based trust to be shorter than 5 hops.

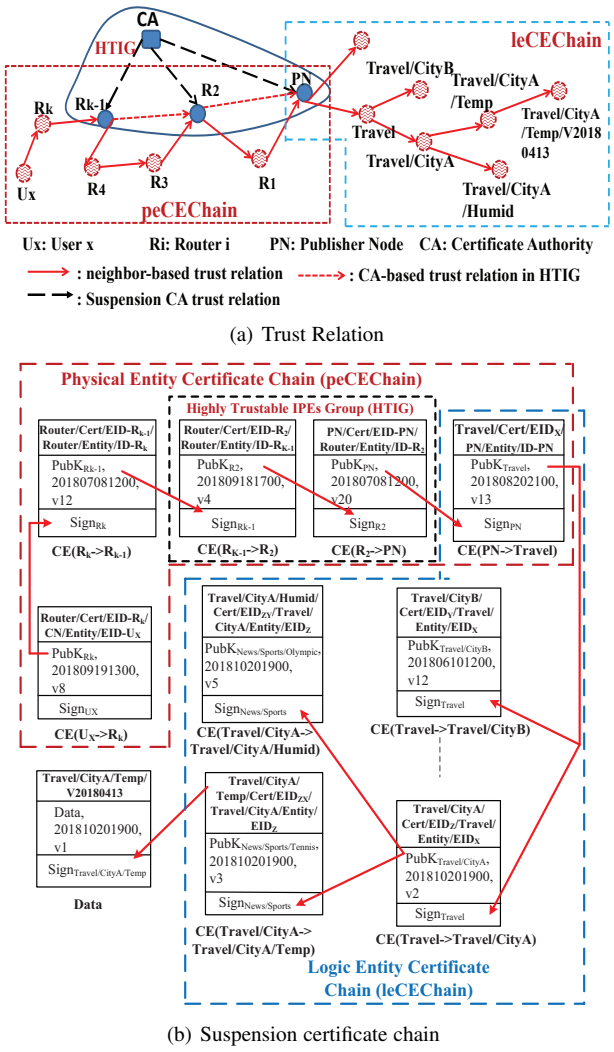
Let $CE(A \rightarrow B)$ represent the public-key certificate issued from A to B. The public key of an entity Z is denoted by $PubK_Z$, and Sig_Z denotes the signature of Z. As in Fig. 1, $CE(R_k \rightarrow R_{k-1})$ is the certificate issued by R_k to R_{k-1} to certify that the public key of R_{k-1} is $PubK_{Rk-1}$. The certificate issued by the authorizer of *Travel* to the sub-authorizer in charge of *Travel/CityA* is $CE(Travel \rightarrow Travel/CityA)$.

Fig. 1 shows an example of the SCM from user U_X to a publisher with the privilege to publish in the category *Travel/CityA/Temp*. Fig. 1(a) illustrates the trust relation and Fig. 1(b) depicts the corresponding suspension chain.

In Fig. 1(a), the neighbor-based trust relations include [U_x trusts R_k], [R_k trusts R_{k-1}], [R_{k-1} trusts R_4], [R_4 trusts R_3], [R_3 trusts R_2], [R_2 trusts R_1], and [R_1 trusts PN], since they are physically connected. For CA-based trusts, R_{k-1}, R_2, and PN are highly trusted physical entities that belong to the HTIG. The CA issues certificates for them, namely $CE(CA \rightarrow R_{k-1})$, $CE(CA \rightarrow R_2)$, and $CE(CA \rightarrow PN)$, which are suspension CA trust relations. The CA distributes these certificates to the IPEs that are close to each other in the HTIG. For example, the CA distributes $CE(CA \rightarrow R_{k-1})$ and $CE(CA \rightarrow R_2)$ to R_{k-1}. If these certificates are verified, they confer CA-based trust relations and issue certificates to each other based on them. In the example shown in Fig. 1(a), R_{k-1} issues $CE(R_{k-1} \rightarrow R_2)$ whereas R_2 issues $CE(R_2 \rightarrow R_{k-1})$ and $CE(R_2 \rightarrow PN)$. These certificates are used to enhance the trust. In Fig. 1(a), the certificates in the HTIG reduce the length of the chain from U_x to PN from 7 to 4, and there are two highly trusted relations among these four, namely $CE(R_{k-1} \rightarrow R_2)$ and $CE(R_2 \rightarrow PN)$. Obviously, this suspension chain is more trustworthy than the pure certificate chain based on neighbor-based trust.

In Fig. 1(b), the suspension chain is composed of two parts, namely peCEChain and leCEChain. The peCEChain describes the trust relations among physical entities, whereas leCEChain illustrates the authorization relations among logical entities. In Fig. 1(b), the trust relations in peCEChain are that [U_x trusts R_k], [R_k trusts R_{k-1}], [R_{k-1} trusts R_2], and [R_2 trusts PN]. The PN hosts the big data application of *Travel*. In leCEChain, the PN authorizes the authorizer to serve the applications of *Travel* on this PN. The authorization relations of *Travel* are as described in the example use scenario in Section 3.1. It is observed that the joint point between peCEChain and leCEChain is the PN. In DCAuth, the PN issues the certificate between itself and the authorizer of *Travel* directly after application installation, which connects the physical entities with the logical entities. The leCEChain is constructed by the PN beforehand, whereas the peCEChain is constructed hop-by-hop during packet forwarding.

Public-key verification is the process for one entity to verify the authenticity of the public key of another entity. User U_X authenticates the public key of the publisher by verifying the suspension chain in the following steps. First, U_X verifies the first certificate by its private key. In Fig. 1, U_X verifies $CE(U_X \rightarrow R_k)$ by verifying Sig_{UX}. If it is correct, it believes that $PubK_{Rk}$ truly belongs to R_k. Subsequently, each intermediate public key is used to verify the next direct associated certificate. This process continues for multiple rounds until the final certificate is verified. Finally, U_X obtains the public key of the publisher.



(a) Trust Relation

(b) Suspension certificate chain

Fig. 1: Typical example for suspension chain model

4.2 Initial Trust Establishment

The initial trust establishment has two components: self-certifiable naming and certificate issuing.

4.2.1 *Self-certifiable Naming*

Self-certifiable naming defines the rule for naming the principals, including entities, keys, and certificates, to enable the entities to be self-certifiable. We merge the hash-based self-certifying names [27] with hierarchical naming as shown in Fig. 2.

The name of a physical entity is defined as EntityType/Entity/EID as in Fig. 2. "EntityType" specifies the type of entity, such as a router. "Entity" is a reserved word to indicate that this name is for an entity. "EID" is set as the hash value of this entity's public key for self-verifiability. The name of the public key for the physical entity is defined as EntityType/PubK/EID, where "PubK" shows that this name is for a public key. "EntityType" and "EID" in the public-key name are the same as in the entity name. This naming rule enables the verification of an entity name by comparing the hash value of the public key and the EID in the entity name. If they are the same, the entity with this name truly holds the public key. This method enables the name to be self-certifiable.

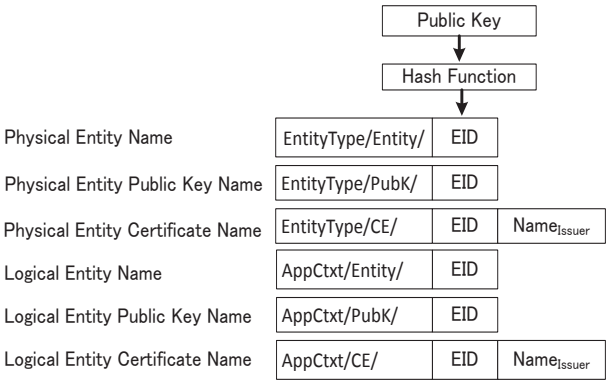


Fig. 2: Self-certifiable names

The name of the certificate for the physical entity is defined as $EntityType/CE/EID/Name_{Issuer}$. In this name, "CE" indicates that the name is for a certificate, and "$Name_{Issuer}$" indicates the name of the certificate issuer, which is introduced to differentiate between certificates from different issuers to the same subject. The logical entities are also associated with the public keys and certificates. To name them, we introduce the application context (AppCtxt) as the hierarchical prefix to describe the category of an entity's authorized privilege, such as *Travel/CityA*. As in Fig. 2, the name of a logical entity is defined as AppCtxt/Entity/EID, where "EID" is the hash of its public key. Similarly, the name of a public key for a logical entity is defined as AppCtxt/PubK/EID, and the name of a certificate is defined as $AppCtxt/CE/EID/Name_{Issuer}$. This naming rule enables the entity to assert name ownership by creating a mapping between a name and the name owner's public/private key pair. This approach embeds the public key into the name, which is self-certifiable.

4.2.2 Certificate Issuing

To issue a certificate, an entity should be convinced that a given public key truly belongs to another entity. Under that situation, it issues a certificate for this entity such that the public key is bound to the entity name by its signature. In DCAuth, certificate issuing is the initial trust establishment among entities having trust relationships, namely as CA-based trust, neighbor-based trust for physical entities, and authorization relationships for logical entities.

CA-based trust between two entities is established using the CA as the "introducer." The CA is managed by the network operator, and provides certificates to the owner's entities and the highly trusted physical entities close to them in the HTIG. The entities in the HTIG then confer CA-based trust relationships and issue certificates to each other.

For neighbor-based trust, a physical entity creates a public key and the corresponding private key locally by itself. It generates the names using the public key based on the self-certifiable naming method. If physical entity B is a neighbor of entity A, B can announce its public key to A with the key name. With self-certifying naming, an attacker cannot take a name created by someone else and send signed packets that appear to come from the owner of that name. However, because names themselves are not certified, an attacker can create a new name and its own public key. This problem can be solved by additional methods, such as challenge/response together with the time-bounded response. Further discussions on this topic are outside the scope of this paper.

In DCAuth, the physical entities pre-keep the certificates and associate certificates with interfaces in FIB. The physical entity knows the next hop of one interface, and it associates the certificate from that entity to itself with the forwarding interface. This mechanism enables the appending of the relevant certificate to the packets as required when forwarding them.

For logical entities, the trust relationship is defined by the application. Each logical entity also generates a public/private key pair by itself. If logical entity A authorizes the right for entity B to manage a sub-category or publish data, A should provide a certificate for the true public key of B. To ensure the public key truly belongs to B, A similarly compares the hash value of B's public key and the EID in B's name.

4.3 Data-centric Certificate Management

We now propose methods to exchange, update, and revoke certificates after the initial establishment of trust.

4.3.1 Certificate Exchange

Certificate exchange allows entities to share the certificates that they issue and hold. Each physical entity has a local repository in which to store certificates securely. In reAuth, the physical entities request and keep all the certificates issued for or by their nearby highly trusted entities in the HTIG and by common neighboring physical entities. Finally, the physical entities hold all the certificates within a two-hop distance and the certificates with nearby highly trusted entities in the HTIG. These certificates are used to construct chains hop by hop, shorten certificate chains, and check the certificates appended by an up-stream entity. This certificate check is performed by the next hop of the entity to check whether the certificate appended by this entity is the same as the one stored by it. If the certificates are the same, the certificate passes the check. Otherwise, this packet will be dropped because of the fake certificate.

The certificates of the logical entities in an application should be stored in the repository of the PN. When data are published, the trust chain from the PN to the publisher can be automatically formed and appended to the packet. Meanwhile, the neighbors of the PN also keep all the certificates of the PN in order to check them during transmissions.

4.3.2 Certificate Update and Revocation

To guarantee its validity, each certificate in the network is issued with a certificate expiration time, after which the certificate is invalid.

For certificate updates using CA-based trust in the HTIG, the subject entity of the certificate requests the CA to issue an updated certificate and provide it to the related nearby highly trusted IPEs, who can further issue update certificates. For certificate updates using neighbor-based trust, the subject entity of the certificate should notify the issuer of its interest in updating the certificate. On receiving this interest, the issuer checks whether this entity has been compromised. It then checks whether the mapping between the name and the public key satisfies the naming rule. If all the checks are passed, the issuer considers whether the public key of the subject entity is still trustworthy, generates an updated certificate, and replies with this updated certificate. If any check is failed, the issuer does not provide a certificate update to that entity.

If one entity no longer wishes to trust another entity, the former may revoke the certificate that it originally issued. Because certificates are issued over a one-hop distance, it is easy to detect the misbehavior of an entity. To revoke a certificate quickly, the revocation is announced over a two-hop distance. The revocation initiator broadcasts the revocation information to all the entities within a two-hop distance. Each entity receiving this information adds the compromised entity or certificate to its blacklist. All the packets from the compromised entities are dropped for a rapid local shutdown.

The operator also maintains a certificate revocation list (CRL). One entity revoking a certificate notifies the CRL of the revocation. If one entity discovers the misbehavior from another entity, it can request to revoke that entity with the evidence. Furthermore, if one entity discovers that a part of the chain is unusable, it reports this part of the chain to the CRL, whereupon the CRL will revoke all the related certificates in that part. All the IPEs retrieve the list of revoked certificates in their spare time without costing busy times.

4.4 Forwarding-Integrated Authenticable Data Retrieval

Here, when forwarding interests and data, we define a forwarding-integrated hop-by-hop approach to construct the suspended chain from unpredictable copy holders to a user for authenticating interest, and from a user or IPE to data for authenticating copy holders or publishers. We let highly trusted IPEs replace the parts of chain with highly trusted certificates induced by CA's suspended trust to enhance trustworthiness. Optimization method to shorten certificate chain is also proposed.

Fig. 3 illustrates an example for data retrievals from PN and IPE, which is based on the typical example scenario in Fig. 1. The steps for data retrieval are as follows.

Step 1: The user issues an interest appended with its signature. It knows its next hop is the IPE to which it is connected. It then appends to this interest the certificate from the next-hop IPE to itself. Finally, it sends out the interest. Take Fig. 3 as an example. User U_x issues an interest to request big data with name as *Travel/CityA/Temp/V20180413*. In the interest packet, $CE(R_k \rightarrow U_x)$ is appended with the interest.

Step 2: When the interest is received by an IPE, such as R_k, …, R_2, R_1 in Fig. 3, it checks whether the previous certificates are correct. If the check succeeds and it belongs to the HTIG, this IPE attempts to find the previous highly trusted IPE to replace the related part of the certificate chain with one highly trusted certificate in the suspended chain. For example, when R_2 receives interest from R_3, it traverses the existing suspended chain and finds that the previous highly trusted IPE is R_{k-1}. Then, R_2 replaces $\{CE(R_2 \rightarrow R_3), CE(R_3 \rightarrow R_4), CE(R_4 \rightarrow R_{k-1})\}$ with $CE(R_2 \rightarrow R_{k-1})$. If this IPE does not belong to the HTIG, it directly finds the interface to the next hop and appends to the interest the relevant certificate from the next-hop IPE to itself. As shown in Fig. 3, R_k checks whether $CE(R_k \rightarrow U_x)$ is correct. R_k can conduct such checks because the certificates are exchanged and kept by the neighboring IPEs. If the check succeeds, this IPE appends $CE(R_{k-1} \rightarrow R_k)$ to the interest. Similarly, R_k, R_{k-1}, R_4, R_3, and R_1 check the previous certificates and append the necessary certificate from the next hop to itself to the interest. Meanwhile, R_2 and *PN* replace part of their certificate paths with a highly trusted one, and then append the necessary certificates.

Step 3: This is executed if an IPE holds the requested big data in its cache. Take R_1 in Fig. 3(b) as an example. It checks the

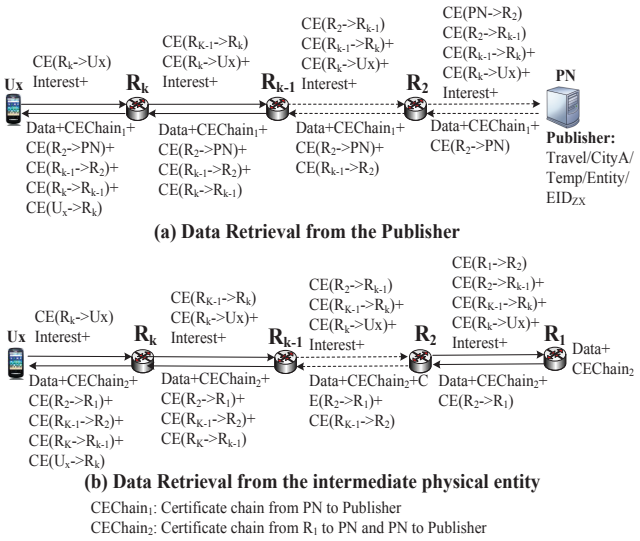


Fig. 3: Data retrievals from publisher or IPE

suspended chain $\{CE(R_1 \rightarrow R_2), CE(R_2 \rightarrow R_{k-1}), CE(R_{k-1} \rightarrow R_k), CE(R_k \rightarrow U_x)\}$ through the process described in the SCM. If the verification succeeds, this IPE replies with the data packet. In the data packet, the suspended chain from this IPE to the publisher, $CEChain_2$ in Fig. 3(b), and the certificate from the next IPE to this IPE, $CE(R_2 \rightarrow R_1)$ in Fig. 3(b), are appended. $CEChain_2$ is cached when these big data are cached in this IPE's memory. This IPE sends this data packet to the interface in reply as specified in the PIT.

Step 4: If the PN receives the interest, it verifies the suspended chain from itself to the user $\{CE(PN \rightarrow R_2), CE(R_2 \rightarrow R_{k-1}), CE(R_{k-1} \rightarrow R_k), CE(R_k \rightarrow U_x)\}$ in Fig. 3(a). If the verification succeeds, the PN discovers the suspended chain from itself to the publisher, $CEChain_1$ in Fig. 3(a), in its storage, and appends this certificate chain with the certificate from the first IPE to itself, $CE(R_1 \rightarrow PN)$, which is late replaced by $CE(R_2 \rightarrow PN)$ in Fig. 3(a). Next, the PN replies with these big data using the reverse path of the interest.

Step 5: After the IPE receives the data packet, it performs forwarding and possibly caching. When the IPE intends to cache the big data, it first caches them in a temporary cache, which is separated from the data cache. Second, it checks the suspended chain from itself to the publisher. Only if the verification passes, these big data can be cached in the data memory and the suspended chain from this IPE to the publisher will be cached along with the data. The verification is performed offline, which does not affect the speed of data retrieval. At the same time, this IPE checks the previous certificate. If the check is successful and it belongs to the HTIG, it will discover the previous entity in the suspended chain belonging to the HTIG. If there is a previous entity, the IPE replaces part of the related certificate path with a highly trusted certificate. Otherwise, the IPE directly finds the interface to the corresponding certificate from the next hop to itself and appends this certificate to the packet, then forwards this packet to the interface.

Step 6: After the user receives the data packet, there should be a certificate chain from the user to the publisher $\{CEChain_2, CE(R_2 \rightarrow PN), CE(R_{k-1} \rightarrow R_2), CE(R_k \rightarrow R_{k-1})$, and $CE(U_x \rightarrow R_k)\}$ in Fig. 3(b). It verifies this suspended chain. If the verification

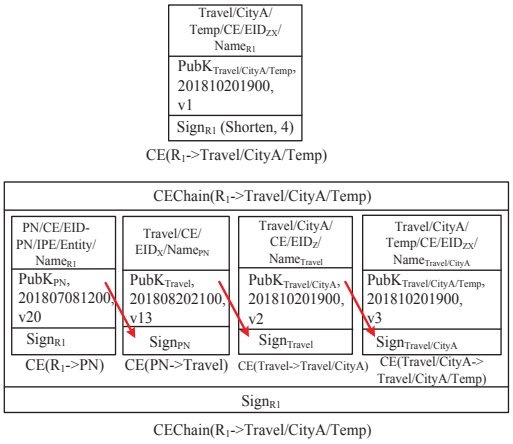


Fig. 4: Shorten certificate chain to one certificate

passes, it believes that it gets the authentic public key of the publisher, and utilizes the key to verify the signature of the data. The user can also verify the copy holder or the IPE on the path.

In DCAuth, to reduce the cost incurred by the suspended-chain construction and verification, the suspended chain can be shortened further by the IPEs and PNs. That is, they can replace the verified suspended chain with a simplified certificate by specifying that this certificate is a shortened one.

We take Fig. 4 as an example to introduce such an optimization. Before caching the big data, R_1 verifies $CEChain(R_1 \rightarrow Travel/CityA/Temp)$, which is composed of four certificates. If the verification succeeds, R_1 believes that it holds the authentic public key of the logical entity that can publish big data with a prefix such as Travel/CityA/Temp. Thus, it issues $CE(R_1 \rightarrow Travel/CityA/Temp)$, as shown in Fig. 4. Meanwhile, the signature information shows that this certificate has been shortened from four certificates. R_1 can also replace the $CEChain_2$ in Fig. 3 with the newly generated $CE(R_1 \rightarrow Travel/CityA/Temp)$. This optimization can lower the cost of suspended-chain construction and verification.

5 Security Analysis

Table 1 summarizes the security properties of DCAuth, the WoT-based approach, and the CA-based approach based on the aspects of data-centric design, attack prevention, revocation, and availability.

In DCAuth, any entity can authenticate any copy holder, and can verify the big data that it retrieves as well. Hence data-centric authentication has been achieved. Authentication from users to publishers can also be realized by the WoT-based and CA-based approaches. However, in those approaches, users cannot authenticate copy holders if they are different from the publishers. Furthermore, IPEs cannot verify big data and copy holders cannot authenticate users. Thus, we mark DCAuth highly for data-centric design, while identifying the WoT-based and CA-based approaches as having low usability in a data-centric environment.

Regarding attack prevention, the malicious-request and data-poisoning attacks mounted by adversaries are rooted in impersonation from outsiders and certificate compromises from insiders. Impersonation can be prevented by the data-centric authentication in DCAuth by dropping packets without valid certificates. For certificate compromises, DCAuth prevents this attack through

TABLE 1: Security Properties

	DCAuth	WoT-based	CA-based
Data-centric design	High	Low	Low
Attack prevention	High	Low	Low
Revocation	Fast, Middle	Low	High with cost
Availability	High	Middle	Low

verification of suspension certificate chains. The copy holders verify the users' identities and revoke them if the number of requests from one verified user exceeds a threshold. The users and IPEs can verify the copy holders' identities and revoke the ones who are found to provide fake data. However, the existing CA-based and WoT-based approaches focus on the authentication from users to the publishers, which cannot effectively address these attacks. Thus, in Table 1, we mark DCAuth as high, whereas the CA-based and WoT-based approaches are low. We also evaluate the performance for attack prevention later in Section 6.4.

For certificate revocation, the CA-based approach uses the CRL to revoke the certificates they issue. Thus, they can achieve the highest revocation, although this incurs costs for synchronizing the CRL. In the WoT approach, the CRL can also be used to revoke the certificates, but the CRL server should first confirm the report on the compromised certificates that are issued among the entities. This process is quite difficult and costly. In DCAuth, such certificates will be shut down by neighbors within two hops for fast revocation if a physical entity is detected to be compromised. CRL is also used for further revocations. The IPEs synchronize their revocation lists in their spare time, thereby causing no disruption to communication. Thus, we mark DCAuth as fast, middle.

As for availability, authentication cannot be performed if the certificate repositories are unworkable in either the common WoT-based approach or the CA-based approach. For the WoT-based approach, some self-organized methods have been proposed for a dynamic environment [21], [22] (e.g., ad-hoc networks) that can enable the availability to some extent. Thus, we mark the CA-based approach as low availability and the WoT-based approach is middle availability. With DCAuth, a certificate chain can be constructed anytime and anywhere only if big data can be retrieved, even if part of the network is down or entities are mobile, as there is a physical path between the two entities. Only if the big data can be retrieved, they can be authenticated. Thus, in Table 1, we mark DCAuth as having high availability.

DCAuth therefore satisfies the requirements for authentication for in-network big-data retrieval from security aspects, which cannot be effectively addressed by the existing WoT-based or CA-based approaches.

6 Performance Evaluations

In this section, we evaluate the performance of DCAuth from the perspectives of cost and attack preventions. For the cost, we compare DCAuth with the PKI-NDN [24], which is a typical authentication mechanism for ICN. With PKI-NDN, users need to retrieve certificates from designated hosts and the authentications are restricted from users to publishers. In contrast, DCAuth seamlessly integrates interest/data-packet forwarding with constructing a suspension chain and data-centric authentication is realized.

For in-network big-data retrieval, users and IPEs have the potential to authenticate users, copy holders, and publishers. Without loss of generality, we investigate the performance when users authenticate the publisher and further data because users are

TABLE 2: Simulation Parameters

Parameters	Value
Simulator	NS3 + ndnSim
Routing	Shortest path routing
Forwarding Strategy	Best performing path selection
Interest generation model	Constant bit rate (CBR)
Interest frequency	100 Interests/sec
Link bandwidth	10 Mbps
One-way link delay	10 ms
Queue size	20 chunks
Interest packet size	200 bytes
Certificate size	200 bytes
Verification time for RSA	0.000013 sec
Verification time for Modified ECDSA	0.0003 sec
Verification time for ECDSA	0.002 sec

equivalent to IPEs. That is, IPEs need to authenticate big data before caching them, which is the same situation as the users.

6.1 Metrics & Experiment Setup

In evaluating the cost of public-key authentication systems, the main metrics are bandwidth consumption and delay [18], whereas for the attack prevention, the main influences on the network is also the bandwidth consumption. Therefore, the following two metrics are considered.

- **Bandwidth Consumption**, defined as the bandwidth (in bytes) consumed in retrieving certificates or consumed by the packet transmissions because of attacks. For the cost of DCAuth, the additional bandwidth consumed in forwarding certificates is measured. In PKI-NDN, interests for certificates are issued by users, and certificates are sent in reply from the relevant designated host(s). Then, the bandwidth consumed in forwarding these packets is measured as the cost in PKI-NDN. For attack prevention, the interests and data packet forwarding effected by attacks are measured.
- **Delay**, defined as the time from the moment that a user sends an interest to the moment that the big data are received and verified by a user. The delay includes the time for interest/data transmission, certificate retrieval, and verifying the copy holder and user.

We simulate DCAuth and PKI-NDN using ndnSIM [28]. The simulation parameters are provided in Table 2. The shortest-path routing and the best-performing-path selection methods [29] are employed. The link bandwidth is set as 10 Mbps, the one-way link delay is set as 10 ms, and the queue size for one node is 20 chunks. The interest packet size is 200 bytes; the size of one certificate is 200 bytes; and the data packet size is 1024 bytes. The interest-generation model for the user is a constant bit rate (CBR) with 100 interests/s, which follows a uniform distribution. The simulation time is 100s. Each simulation is repeated 10 times with different seeds, and the average of the network performance is taken.

We simulate three cryptographic algorithms, namely RSA [30], the elliptic-curve digital-signature algorithm (ECDSA) [40], [31], and modified ECDSA (MECDSA) [32], which is an improved version of ECDSA. These are typical signature algorithms that have different verification speeds. Usually, RSA requires a 1024-bit key to ensure security. It has been claimed that a 192-bit ECDSA key is similar to a 1024-bit RSA key with respect

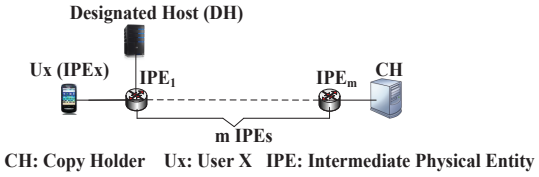


Fig. 5: Topologies with different scales

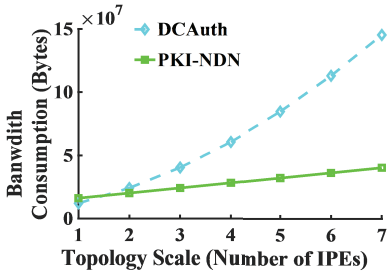


Fig. 6: Bandwidth consumption vs. Topology scale

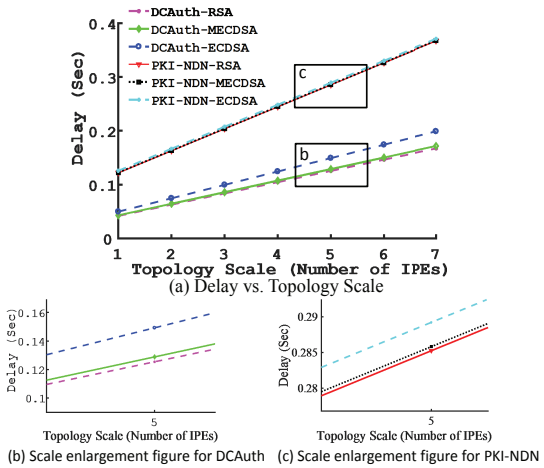
to the level of security they offer [33]. According to the existing evaluations on these algorithms in NDN [32], the verification times for RSA with a 1024-bit key, MECDSA with a 192-bit key, and ECDSA with a 192-bit key are 0.000013 s, 0.0003 s, and 0.002 s, respectively.

In DCAuth, one certificate is appended when forwarding interest/data. In PKI-NDN, when the copy holder receives an interest, it requests the designated host(s) for the relevant certificates. We note that the network scale and the number of requested designated hosts are the key factors that impact the costs of DCAuth and the PKI-NDN. Therefore, we first investigate the influence of network scale on the cost of DCAuth and PKI-NDN (Case 1), and then investigate the impact of the number of requested designated hosts (Case 2). We also explore the attack prevention under a realistic topology, namely Abilene topology (Case 3).

6.2 Case 1: Impact of Network Scale

To evaluate the performance at different network scales, we investigate line topologies for different number of IPEs between the user and the copy holder, as in Fig. 5. In our simulations, there is one copy holder holding the big data and one user acquiring the big data. This user can also be an IPE that authenticates and caches the big data. A total of m IPEs exist between user U_X and the copy holder, where m varies from 1 to 7. One designated host that serves for certificate provision connects with IPE_1 in this case investigation.

We obtain the bandwidth consumption while varying the topology length, as shown in Fig. 6. We observe that the difference in bandwidth consumption between DCAuth and PKI-NDN increases rapidly with distance. This is because the number of certificates that need to be transmitted increases hop by hop until it reaches a copy holder. If the distance increases by one hop, the additional bandwidth consumption for this increase is more than that for the previous hop because all the certificates for the previous hops and the newly collected certificates should be transmitted on this new hop. In contrast, the bandwidth consumption for PKI-NDN increases nearly linearly because the sum of the hops to retrieve the certificates from the designated host by the user and



(a) Delay vs. Topology Scale

(b) Scale enlargement figure for DCAuth (c) Scale enlargement figure for PKI-NDN

Fig. 7: Delay vs. Topology scale

the copy holder increases linearly with the distance between the user and the copy holder.

Fig. 7(a) shows how the delay changes with network scale, and Figs. 7(b) and 7(c) show the scale-enlargement figures of rectangles b and c in Fig. 7(a). From Fig. 7, we see that the delay of PKI-NDN is much higher than that of DCAuth. However, the delay difference between DCAuth and PKI-NDN decreases with verification time because DCAuth is more influenced by verification time than is PKI-NDN.

From Fig. 7(a), we see that the delay increases with both distance and verification time. The delay increases nearly linearly because the sum of the hops for certificate retrieval from the designated host increases linearly in PKI-NDN. A similar situation occurs in DCAuth. From Figs. 7(a) and 7(b), we see also that the delay increases faster with higher verification time than with lower verification time in DCAuth. This is because more certificates need to be verified as the number of hops increases. From Fig. 7(c), we see that the delay difference remains nearly constant with verification time, although more verification time means more delay.

6.3 Case 2: Impact of Number of Requested Designated Hosts

The requested designated hosts denote the designated hosts that need to be accessed to retrieve the relevant certificates. Obviously, the number of requested designated hosts influences the performance of PKI-NDN, whereas it does not influence DCAuth. We assess here the relative change in performance within the number of requested designated hosts.

We simulate the realistic Abilene topology as illustrated in Fig. 8. There are ten copy holders and ten users connected in the network. There are five designated hosts in the network, and each of them connects with one IPE. We vary the number of requested designated hosts for certificates from one to five.

We obtain the bandwidth consumption as shown in Fig. 9, where we observe that it is independent of the number of designated hosts in DCAuth. In contrast, the bandwidth consumption increases with this number in PKI-NDN. We see also that the bandwidth consumption of PKI-NDN exceeds that of DCAuth for one or two requested designated hosts under the Abilene-topology scenario. Although DCAuth incurs a relatively high bandwidth

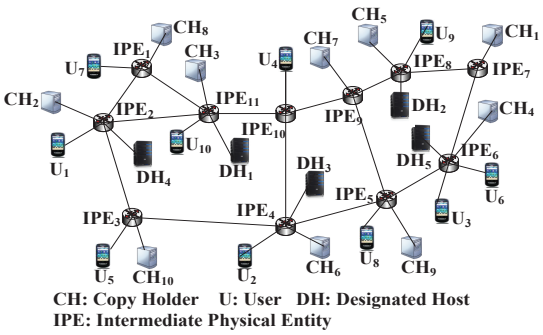


Fig. 8: The Abilene topology

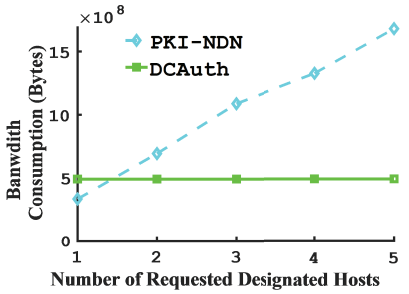


Fig. 9: The impact from number of requested designated hosts on bandwidth consumption

cost with relatively few requested designated hosts, PKI-NDN incurs an even higher bandwidth cost if the number of requested designated hosts exceeds a threshold.

Fig. 10 shows how the delay changes with the number of requested designated hosts. Similarly, the delay of DCAuth is independent of the number of requested designated hosts, whereas the delay increases with this number in PKI-NDN. We find that the delay of retrieving data through PKI-NDN is much greater than that of DCAuth, and the difference between them increases rapidly with the number of requested hosts.

6.4 Case 3: Attack Prevention under Abilene Topology

We investigate the performance of attack prevention under the Abilene topology with ten copy holders and ten users as shown in Fig. 8. We define the ratio of the adversaries as the number of adversaries over the total number of normal entities with the same type. We vary this ratio from 10% to 50% to investigate the bandwidth consumption.

For the malicious-request attack, an adversary sends the malicious Interests to the network with 500 packets/s. For the data-poisoning attack, the adversary(s) provides fake data to the users. Both outsiders and insiders are considered. The outsiders do not hold legal certificates, whereas the insiders hold compromised certificates. To prevent the malicious-request insiders, we let copy holders revoke the users (insiders) if the number of requests from the same verified user exceeds 10. To inhibit the data-poisoning insiders, we enable the revocation of the copy holders (insiders), if the user finds that the big data it retrieves are fake regardless of whether the data passed the verification.

We obtain the bandwidth consumption, shown in Figs. 11 and 12, by the malicious-request and data-poisoning attacks, respectively. Both Figs. 11 and 12 show that the bandwidth

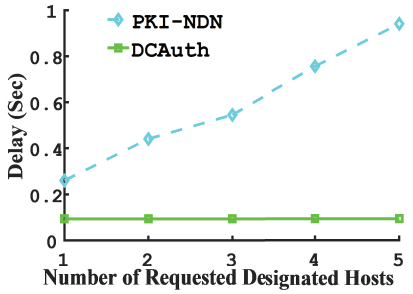


Fig. 10: The impact from number of requested designated hosts on delay

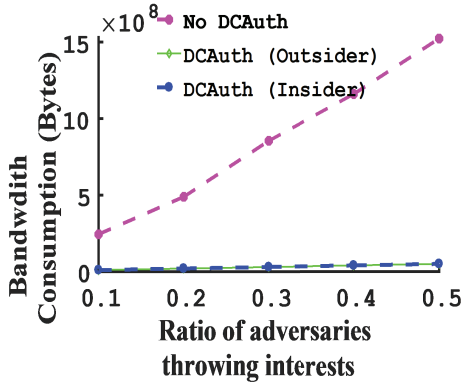


Fig. 11: Bandwidth consumption under Interest flooding attack

consumption due to these attacks increase drastically with the ratio of adversaries if there is no DCAuth. PKI-NDN and the other existing authentication schemes without DCAuth only focus on the authentication from the users to the publishers, which cannot prevent the malicious-request and data-poisoning attacks (both outsiders and insiders). The proposed DCAuth scheme can reduce the bandwidth consumption caused by these attacks to a lower situation, though there is still some bandwidth consumption. This bandwidth consumption is effected by the first one-hop transmissions to flood malicious interests or provide fake data or by the first several rounds of big-data retrieval and authentication to discover adversaries.

As in Figs. 11 and 12, it is observed that the cost for preventing outsiders is similar to that for insiders for both attacks in DCAuth. In further detail, the insiders cause slightly more bandwidth consumption than the outsiders. It is because around 10 data retrievals are tolerated to discover malicious-request insiders, and users need to retrieve and authenticate the fake data in order to discover data-poisoning insiders. In contrast, the outsiders can be prevented by the first-hop certificate checks from neighbors, as they do not have valid certificates.

Therefore, we see that DCAuth achieves a minimal additional delay without incurring additional certificate retrievals and its bandwidth consumption is acceptable for performance while efficiently preventing malicious-request and data-poisoning attacks for in-network big-data retrieval.

7 Conclusions

In-network big-data retrieval is vulnerable to malicious-request and data-poisoning attacks. To prevent such attacks, we

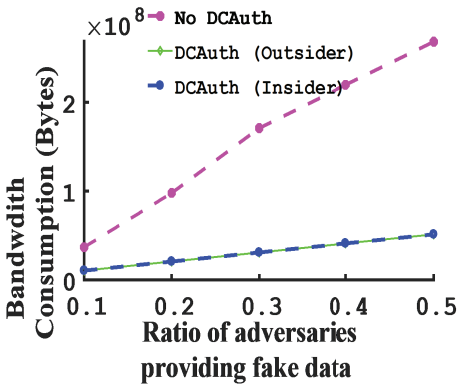


Fig. 12: Bandwidth consumption under fake data attack

proposed DCAuth, which provides data-centric authentication with merging CA-based trust and neighbor-based trust. It enables authentication among entities including users, IPEs, copy holders, and publishers, regardless of their unpredictability. Extensive simulations have been conducted, and show that DCAuth can reduce the delay for certificate collection compared to PKI-NDN and can prevent malicious-request and data-poisoning attacks efficiently.

References

[1] N. Khan, I. Yaqoob, I. Hashem, Z. Inayat, W. Ali, M. Alam, M. Shiraz, and A. Gani, "Big Data: Survey, Technologies, Opportunities, and Challenges," *The Scientific World Journal* , Vol. 2014, 2014.

[2] H. Yin, Y. Jiang, C. Lin, Y. Luo, and Y. Liu, "Big data: Transforming the design philosophy of future Internet," *IEEE Network*, vol. 28, no. 4, pp. 14-19, July 2014.

[3] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, "Networking for Big Data: A Survey," *IEEE Communications Surveys and Tutorials*, Vol. 19, Issue 1, pp. 531-549, 2017.

[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, 50(7), pp.26-36, 2012.

[5] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, 50(7), pp.26-36, 2012.

[6] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," *ACM SIGCOMM 2007*, Aug. 2007.

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," *ACM CONEXT'09*, Rome, Italy, Dec. 2009.

[8] H. Cui, X. Yuan, Y. Zheng, and C. Wang, "Enabling secure and effective near-duplicate detection over encrypted in-network storage," *The 35th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2016)*, Apr. 10-15, San Francisco, CA, USA.

[9] E. AbdAllah, H. Hassanein, and M. Zulkernine, "A survey of security attacks in information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 17, issue 3, 2015.

[10] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," *IFIP Networking*, pp. 1-9, 2013.

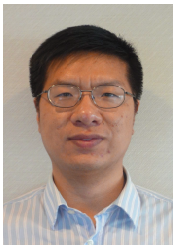
[11] A. Compagno, M. Conti, P. Gasti, and G. Tsudik, "Poseidon: mitigating interest flooding ddos attacks in named data networking," *IEEE LCN'13*, pp. 630-638, Oct. 2013.

[12] T. Nguyen, R. Cogranne, and G. Doyen, "An optimal statistical test for robust detection against interest flooding attacks in ccn," *IFIP/IEEE International Symposium on Integrated Network Management (INM)*, pp. 252-260, 2015.

[13] C. Ghali, G. Tsudik, and E. Uzun, "Network-layer trust in named-data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, number 5, Oct. 2014.

[14] D. Kim, S. Nam, J. Bi, and I. Yeom, "Efficient content verification in named data networking," *ACM Conference on Information-Centric Networking*, pp. 109-116, 2015.

[15] T. Dierks, and E. Rescorla, "The transport layer security (TLS) protocol," *IETF RFC 5246*, Aug. 2008.
[16] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R Housley, and W Polk, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) prole," *IETF RFC 5280*, May 2008.
[17] P. Hoffman and J. Schlyter, "The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: TLSA," *IETF RFC 6698*, Aug. 2012.
[18] D. Basin, C. Cremers, T. Kim, A. Perrig, R. Sasse, and P. Szalachowski, "Design, Analysis, and Implementation of ARPKI: an Attack-Resilient Public-Key Infrastructure," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, DOI: 10.1109/TDSC.2016.2601610, 2016.
[19] P. Zimmermann, "The official PGP user's guide," *MIT Press*, 1995.
[20] N. Li, W. H. Winsborough, and J. C. Mitchell, "Distributed credential chain discovery in trust management," *Journal of Computer Security*, vol. 11, no. 1, pp. 35-86, Feb. 2003.
[21] K. Hamouid, and K. Adi, "Self-certified based trust establishment scheme in ad-hoc networks," *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, Turkey, May 7-10, 2012.
[22] S. Capkun, L. Buttya, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, issue 1, pp. 52-64, 2003.
[23] A. Ulrich, R. Holz, P. Hauck, and G. Carle, "Investigating the OpenPGP web of trust," *The 16th European conference on Research in computer security*, pp. 489-507, Sept. 2011.
[24] Y. Yu, "Public key management in named data networking," *NDN Technical Report, NDN-0029*, 2015.
[25] A. Afanasyev, "Addressing operational challenges in named data networking through NDNS distributed database," *PhD thesis*, UCLA, 2013.
[26] A. Hoque, S. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: named-data link state routing protocol," *The 3rd ACM SIGCOMM workshop on Information-centric networking*, pp. 15-20, 2013.
[27] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in content-oriented architectures," *The 1st ACM SIGCOMM workshop Information-centric networking*, Aug. 2011.
[28] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," *NDN Technical Report, NDN-0005*, http://named-data.net/techreports.html, 2012.
[29] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *SIGCOMM Computer Communication Review*, vol. 42, no. 3, pp. 62-67, Jun. 2012.
[30] R. Rivest, A. Shamir, and L, Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, issue 2, pp. 120-126, Feb. 1978.
[31] K. Katiyar, K. Dutta, and S. Gupta, "A survey on elliptic curve cryptography for pervasive computing environments," *International Journal of Computing Applications*, vol. 11, no. 10, pp. 41-46, Dec. 2010.
[32] A. Ali, "Comparison and evaluation of digital signature schemes employed in NDN network," *International Journal of Embedded systems and Applications(IJESA)*, vol.5, no.2, June 2015.
[33] Federal Office for Information Security, "Technical Guideline TR-03111 Elliptic Curve Cryptography Version 2.0," Germany, 2012.
[34] J. Saltzer, D. Reed, and D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, 2 (4), pp. 277-288 1984.
[35] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "Dos and ddos in named data networking," *In the IEEE ICCCN'13*, pages 1-7, July 30-Aug. 2, 2013.
[36] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: Line Speed Publish/Subscribe Inter-Networking," *Proceedings of the ACM SIGCOMM 2009*, pp. 195-206, Aug. 2009.
[37] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: seeing the forest for the trees," *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, 2011.
[38] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker, "Less pain, most of the gain: incrementally deployable icn," *Proceedings of the ACM SIGCOMM 2013 conference*, pages 147-158, 2013.
[39] Y. Yu, A. Afansyev, D. Clark, K. Claffy, V. Jacoboson, and L. Zhang, "Schematizing trust in named data networking," *In ACM Conference on Information-Centric Networking*, pp. 177-186, CA, USA, Sept. 2015.
[40] A. Johnson, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *In Certicom research*, 2001.



Ruidong Li is a researcher with the Network System Research Institute, NICT. He received a bachelor in engineering from Zhejiang University, China, in 2001. He received a doctorate of engineering from the University of Tsukuba in 2008. He received the Best Student award from the Graduate School of Systems and Information Engineering, University of Tsukuba in 2007. He serves as co-chair for the young researcher group in the Asia Future Internet Forum (AsiaFI). His current research interests include future networks, information-centric network, internet of things, network security, and wireless networks. He is a member of the IEEE and IEICE.

Hitoshi Asaeda is a research manager with the Network System Research Institute, NICT. From 1991 to 2001, he was with IBM Japan, Ltd. From 2001 to 2004, he was a research engineer specialist at INRIA Sophia Antipolis, France. He was a project associate professor at Keio University, where he was during 2005-2012. His research interests include information-centric networking (ICN), network coding, high-quality streaming, and large-scale testbeds. Since 2012, he has been chairing the ICN working group in the Asia Future Internet Forum (AsiaFI). He has also been actively working in the IETF standards body. He holds a Ph.D. from Keio University. He is a senior member of IEEE and IEICE, and a member of ACM.

Jie Wu is the Associate Vice Provost for International Affairs at Temple University. He also serves as Director of the Center for Networked Computing and as Laura H. Carnell professor. He served as Chair of Computer and Information Sciences from 2009 to 2016. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Service Computing and the Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.