Securing Named Data Networks: Challenges and the Way Forward

Elisa Bertino
Purdue University
West Lafayette, Indiana
bertino@cs.purdue.edu

Mohamed Nabeel
Qatar Computing Research Institute
Doha, Qatar
mnabeel@qf.org.qa

ABSTRACT

Despite decades of research on the Internet security, we constantly hear about mega data breaches and malware infections affecting hundreds of millions of hosts. The key reason is that the current threat model of the Internet relies on two assumptions that no longer hold true: (1) Web servers, hosting the content, are secure, (2) each Internet connection starts from the original content provider and terminates at the content consumer. Internet security is today merely patched on top of the TCP/IP protocol stack. In order to achieve comprehensive security for the Internet, we believe that a clean-slate approach must be adopted where a content based security model is employed. Named Data Networking (NDN) is a step in this direction which is envisioned to be the next generation Internet architecture based on a content centric communication model. NDN is currently being designed with security as a key requirement, and thus to support content integrity, authenticity, confidentiality and privacy. However, in order to meet such a requirement, one needs to overcome several challenges, especially in either large operational environments or resource constrained networks. In this paper, we explore the security challenges in achieving comprehensive content security in NDN and propose a research agenda to address some of the challenges.

KEYWORDS

Named Data Networks, Edge Computing, Access Control, Security, Confidentiality, Integrity, Privacy

ACM Reference Format:
Elisa Bertino and Mohamed Nabeel. 2018. Securing Named Data Networks: Challenges and the Way Forward. In *SACMAT '18: The 23rd ACM Symposium on Access Control Models & Technologies (SACMAT), June 13–15, 2018, Indianapolis, IN, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3205977.3205996

1 INTRODUCTION

The current IP based end-to-end Internet architecture [20] designed in 1970's is fundamentally broken as the way we use the Internet has changed drastically over the last four decades. Nowadays, the Internet usage is dominated by content distribution mainly due to video streaming and billions of things connected to the Internet. As a solution, the notion of Named Data Networking (NDN) [61] has been proposed. NDN is a general-purpose, information-centric network architecture [55], that uses names to identify resources in the Internet, similar to the REST architecture [29], and provides native support for content caching at edge nodes. While NDN provides many benefits compared to the traditional network architectures, in order to gain its full potential and make it practical, one needs to address security, efficiency and scalability.

Key requirements for NDN security is to assure that data managed by the system is not tampered with and also that data is kept confidential and only accessed by authorized parties. In addition, privacy is critical.

NDN defines two types of network packets, possessing highly asymmetric properties. Clients send *interest packets*, which contain only a name and a minimal set of additional control fields. Servers respond with *data packets*, which contain the data associated with the name in the corresponding interest. By looking at interest packets of an individual, a malicious party can infer privacy-sensitive information about the individual. Even though today we have a huge body of security and privacy techniques, applying these techniques to NDN, especially when deployed on 5G networks, is challenging due to stringent real time requirements and the scale, and highly dynamic nature of the systems.

In order to provide a comprehensive framework for data security and privacy in NDN, it is critical to address three main requirements. The first requirement focuses on designing efficient digital signature techniques; this is a critical security building block for NDN in order to ensure authenticity and integrity of data packets. The second requirement focuses on access control techniques to allow selective sharing of the data packets with end-to-end encryption enforced. Finally, the third requirement focuses on privacy which is perhaps the most challenging issue.

NDN requires data producers to digitally sign every data packet so that data consumers can verify the data without caring about the locations from which the data packets are delivered. Specifically, a valid digital signature gives data consumers reason to believe that the data was created by a known data producer (authentication), that the data producer cannot deny having transmitted the data (non-repudiation), and that the data was not altered in transit (integrity). However, considering that the data consumers can be mobile devices or small Internet of Things (IoT) devices with limited resources, it is critical to minimize the signature generation/verification latency and the signature size and enhance the efficiency of all operations related with the management of data signature processes.

SACMAT '18, June 13–15, 2018, Indianapolis, IN, USA

ACM ISBN 978-1-4503-5666-4/18/06…$15.00
https://doi.org/10.1145/3205977.3205996

Access control is critical in NDN systems in order to selectively share data among users. Access control has been widely investigated (we refer to [15] for a survey of access control for database systems). At a higher level, an access control system is based on an access control model (such as the discretionary model and the mandatory model [15]). When a discretionary model is adopted, the system uses a set of permissions to decide whether access to a protected resource can be granted. In this paper, we argue that the most suitable model is what we refer to as "name-based access control" model. In addition to differing with respect to access control model, access control systems differ with respect to the enforcement approaches used (e.g.: access control lists, and encryption). The actual enforcement mechanism to be adopted depends in turn from the system architecture and the types of actions to be controlled. A critical issue in designing an access control system for NDN is to select a proper enforcement mechanism. Such a mechanism has to be decentralized, as having to contact some centralized server for access control enforcement is not suitable when there are real-time constraints and does not follow the decentralized distributed architecture of NDN. Further, the mechanism must exploit caching mechanisms to support caching of information needed for access control.

Ensuring privacy for both content producers and content users is a very important as well as a challenging step towards building practical NDN. We identify three key requirements to ensure privacy: (1) communication annonymity, that is, making it difficult for an attacker to trace back to a sender of a message received by a destination; (2) search privacy, that is, hiding the content of the interest packets sent by content users from intermediaries in the NDN infrastructure and attackers; (3) cache privacy, that is, making it difficult for a data consumer to infer information about the content consumption patterns of other consumers in the physical proximity based on cached contents. It should be noted that even if data packets are end-to-end encrypted, the privacy of data packets is ensured only if the privacy of the interest packets is also preserved. The reason is that the content of the interest packet, which includes the name of the content users want to consume, may reveal information about the encrypted data packets even though intermediaries cannot decrypt such packets.

In what follows based on the characteristics of NDN systems, we propose possible approaches to meet the content security and privacy requirements that we have outlined. Further, we critically evaluate the existing solutions proposed for some of the security challenges. We also identify and discuss open research challenges that have to be addressed in order to build practical and secure NDN systems.

The rest of the paper is organized as follows. Section 2 presents an overview of NDN with an emphasis on security. In Section 3, we critically evaluate the existing solutions to address some of the security problems discussed in this paper. We identify challenges in supporting scalable and efficient digital signatures on data packets and discuss possible directions to solve such challenges in Section 4. Section 5 identifies challenges in decentralized access control in NDN and possible solutions to address such challenges. Finally, in Section 6, we discuss privacy requirements in details and challenges in addressing such requirements.

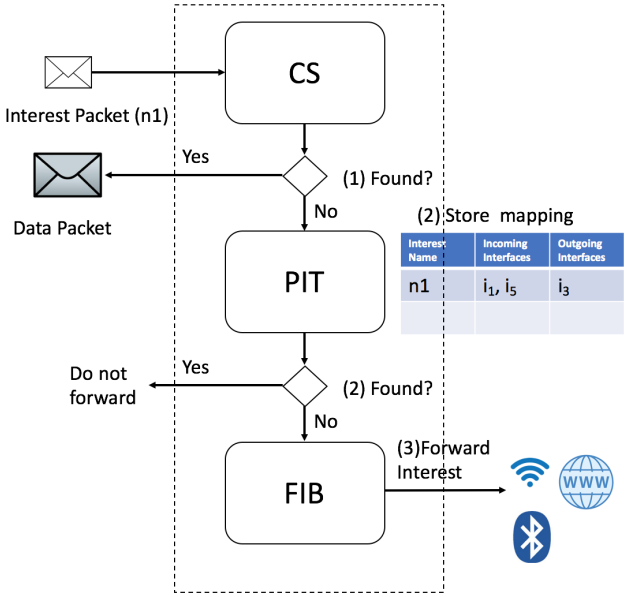

Figure 1: An NDN Node

2 NAMED DATA NETWORKING

NDN [34, 61] is one instance of a more general next generation network architectures called Information Centric Networking [55]. The Internet Research Task Force (IRTF) established an ICN research working group [4] in 2012 in order to further advance the research and standardization of ICN. NDN has its roots in an earlier projected called Content-Centric Networking (CCN) proposed by Van Jacobson. He publicly presented his work at Google Tech talk in 2006 [33].

NDN changes the Internet's communication model from delivering packets to an end host to retrieving content for a given name. The communication in NDN is driven by receivers, i.e. data consumers, based on a pull model. They exchange two types of packets: Interest and Data. Both types of packets carry a name that uniquely identifies a piece of data.

In order to fetch data, a data consumer creates an interest packet adding the name of the data packet it needs and sends it to the network. The routers in NDN use the name in the data packet to push the interest packet towards data producer(s). Once the interest packet reaches a network node that has the requested data packet, signed by the producer's private key, the node pushes the data packet to the consumer following the reverse path that the interest packet took.

Both types of packets carry a name that uniquely identifies an information item in NDN within the given scope and context. NDN names are opaque to the network meaning that NDN does not attribute any meanings to the names. Thus, it allows applications to choose their own naming conventions and evolve independently of the network. However, NDN does assume that names are hierarchically structured. For example, a whitepaper produced by QCRI may have the name /qa/org/qcri/papers/whitepaper.pdf, where '/' separates the name components in text representations similar to

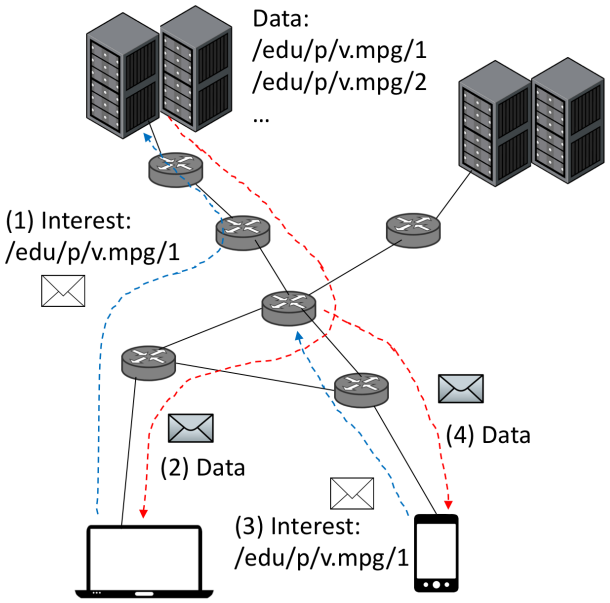

Figure 2: Routing and Caching in NDN

RESTful resources. Large objects, e.g., a video, that cannot be carried as a single component, are segmented into multiple packets. For example, /edu/purdue/cs/keynote.mpg/2 may represent the second segment of a particular keynote video from Purdue University.

As mentioned earlier, names in NDN need to be unique within the scope and context only, very much like the concept of public and private IPs. For example, two organizations may use the same name /iotdevices/smartlights/room1 in their own private context, but if the name is used in the global context, it should be unique universally.

As NDN routes packets based on names, it eliminates two issues in current IP based routing architecture: address space exhaustion, and address management. Unbounded namespace eliminates the address space exhaustion problem. Local address assignment and management are no longer required as NDN does away with IP addresses.

As shown in Figure 1, an NDN node maintains three data structure to move packets around the network: a Forwarding Information Base (FIB), a Pending Interest Table (PIT), and a Content Store (CS). As mentioned earlier, in order to request some data, a consumer sends an interest packet containing the name of the data. These interest packets are forwarded along routes by the FIB in each node. When an interest packet arrives at a node, the node first checks its CS to see if the data is cached; if this is the case, the node returns the data packet to the interface from which the interest packet had been received. Otherwise, the node checks if the data is available in the PIT; if a matching entry is available, the node simply appends the interface of this request to the existing entry and waits for the reply from upstream nodes. If an entry does not exist, the node creates an entry in PIT, and consults the FIB to find the nodes towards data producer(s) in order to forward the request. When a data packet arrives at the node, it follows the reverse path. In order to mitigate Distributed Denial of Service (DDoS) attacks, the node accepts only those data packets which it anticipates to receive as recorded in the PIT. Based on the caching policy, the node may or may not cache the data packet after forwarding it to the corresponding interfaces. Figure 2 shows the typical caching based routing in NDN.

The current connection-based approach to security (Transport Layer Security (TLS)) inexplicably ties the security of the content to trust in the server that stores the content. This approach is widely recognized as a significant problem as the trust that the consumer gets is essentially transient leaving no reliable traces on the content after the connection is over. For example, in order to have some confidence over the data, one always needs to retrieve the data from the original server, not from any intermediaries. NDN is designed to address this issue by moving to a content-based security model instead of a connection-based one. In fact, a central tenant of NDN is that data consumers do not care if a data packet was served from a network node caching the content or from the original producer. Similarly, data producers do not care from where and when data consumers receive the data packet. Thus, the trust in data is decoupled from the time when the data was originally obtained and from the location(s) from which the data was obtained. This content-based design demands mechanisms to validate the integrity, and authenticity of data packets by data consumers and ensure confidentiality.

3 RELATED WORK

NDN security can broadly be classified along two dimensions: infrastructure security and content security. As this paper focuses on the latter, we only provide a summary of the former as we believe that most of the infrastructure security issues can be addressed using the same techniques used to protect traditional IP network infrastructures.

Content based security is at the core of NDN design. NDN specifies that all data packets must be secured by cryptographically signing them. However, designing mechanisms that are efficient, scalable, and usable in order to meet the security requirements of integrity, authenticity, confidentiality and privacy are very much open problems that a few researchers have started to research on. [7, 52, 59]. A issue in meeting these requirements is to establish trust in the keys utilized for enforce the security mechanisms. Diana et al. [52] proposes a PKI based scheme to verify the authenticity and integrity of names associated with contents. They further elaborate on the level of trust one place on the keys used to sign data packets together with their names. Going one step further, Yu et al. [59] proposes the notion of trust schemas that can provide data consumers an automatic way to discover which keys to use for authentication and data producers to identify which keys to use for the signatures. However, it is not clear from their work how data consumers can correctly identify trust schemas or trust anchors, and bootstrap trust. A more serious issue is that there is no provision for key revocation and reflecting key revocation in trust schemas. Afanasyev et al. [7] have investigated how to utilize secure NDNs to replace the current connection based Web infrastructure. They discuss security issues in HTTPS especially with the prevalence of CDNs and HTTPS termination, and propose

a research agenda to solve these issues by using content-based security over NDN. It should be noted that their focus areas are different from what we discuss in this paper. They mainly focus on supporting cryptographic protocols in browsers, key management for data producers and consumers, and establishing trust in keys.

Akin to the routing and lookup infrastructures in TCP/IP based architecture, NDN does require a set of always available services in order to operate across multiple networks. Recently, Afanayev et al. [8] proposed a DNS like name service for NDN to identify the need to look up names. While the proposed system has features similar to DNSSEC [9] and to security extensions of DNS, they show that their design differs from the DNS design mainly due to how NDN operates and NDN caching mechanisms. DNS operates at the application layer, whereas NDN works on names at the network layer itself. Gasti et al. [30] analyze the resilience of NDN to DDoS attacks and identify some new types of attacks specific to NDN, such as interest flooding. Due to the design of NDN, they show that current DDoS attacks such as bandwidth depletion, reflection attacks [46], and black-holding by prefix hijacking [12], are ineffective in NDN systems. As you may recall, NDN supports built-in caching to accelerate content delivery and these caching nodes are an attractive target for attackers. Securing these NDN caching nodes is also an active research area [23, 37].

Privacy protection in NDN [21] is a seldom explored topic. While some features of NDN, such as the lack of source/ destination addresses and cached content retrieval, improve privacy, a closer look at the design choices of NDN reveals a number of open privacy issues: name privacy, cache privacy [5, 6], and certificate privacy are some of the issues that need further attention from the privacy research community.

Content security is not a new topic. In the last couple of decades, numerous research efforts concerning systems and models have focused on content based security, mainly confidentiality. Such efforts include end-to-end encrypted messaging systems [27], encrypted content dissemination [50], encrypted cloud storages [44], encrypted query processing systems (e.g.: CryptDB, DBMask, Monomi, TrustedDB, and Cipherbase), encrypted publish-subscribe systems [24, 40], encrypted web application platforms (e.g.: Mylar, and ShadowCrypt), encrypted email systems [47], computation over encrypted data, and end-to-end integrity over web [36]. While some of the building blocks developed as part of the above systems (e.g. attribute based group key management [41]) can be applied to secure NDN systems, most of the cryptographic techniques utilized in these systems are either known be broken under practical threat models or too inefficient to meet performance and response time requirements. While content based security, especially encryption, satisfies the necessary security requirements, encryption comes with a cost, namely broken functionality. The above systems utilize a new class of algorithms that try to strike a balance between these two conflicting goals of security and functionality. They are collectively called property preserving encryption schemes: searchable encryption schemes [17, 53], order preserving encryption schemes [38], and format preserving encryption schemes [13]. Most of the above property preserving schemes are known to leak information to various degrees based on auxiliary information available and broken under honest-but-curious threat model [19, 45].

4 DIGITAL SIGNATURES

A valid digital signature gives data consumers reason to believe that the data was created by a known data producer (authentication), that the data producer cannot deny having transmitted the data (non-repudiation), and that the data was not altered in transit (integrity). However, considering that the data consumers can be mobile devices or small IoT devices with limited resources, it is critical to minimize the signature generation/verification latency and the signature size and enhance the efficiency of all operations related with management of data signature processes. In particular the design of a suitable digital signature for NDN should follow three different orthogonal strategies: (1) adopt the most efficient signature scheme(s); (2) devise scalable and efficient strategies for the management of information required by the adopted scheme(s) (such as public encryption keys); (3) support the concurrent execution of multiple authentication operations. Adoption of these strategies is particularly critical for NDN deployed over 5G networks in that in these networks connection times are extremely short. In what follows, we discuss approaches that can be adopted and extensions to these approaches to meet scalability and stringent time requirements.

Signature Schemes: Over the past decades, various digital signature schemes have been devised. NIST recommends the RSA signature algorithm, the Digital Signature Algorithm (DSA), and its elliptic curve variant ECDSA as the digital signature standards and specifies their parameters for various security levels [2]. Each such algorithm has its unique properties. The RSA signature algorithm has the fastest signature verification time, but the slowest signature generation time. DSA is slower in verifying, but faster in signing than the RSA signature algorithm. A DSA key of the same strength as the RSA signature algorithm generates a smaller signature. The elliptic curve-based algorithms have moderate signature generation/verification time (see the details in [3]). However, compared with the RSA algorithm, the size of ECDSA signatures/keys is much smaller than the size of the RSA signatures/keys. Recently, Bernstein et al. [14] developed the Edwards-curve Digital Signature Algorithm (EdDSA) using a variant of Schnorr signature based on Twisted Edwards curves. It is designed to be faster than existing digital signature schemes without sacrificing security. EdDSA is included in OpenSSH and GnuPG. Therefore, when designing the security framework of NDN, it is crucial to adopt the best digital signature scheme according to the specific NDN application scenario. For efficiency, we should also consider signature aggregation. Given n signatures on n distinct messages, by different n users, it is possible to aggregate all these signatures into a single signature [18]. Aggregate signatures may be useful for reducing the size of data signatures in NDN. To minimize signing cost, Merkle Hash Trees [39] can be used to aggregate many data contents and sign them all together.

In order to develop digital signature schemes that are very efficient in terms of response time, one possible approach is to select the most efficient representative digital signature techniques and combine them with pre-computation techniques. Such strategy is based on the observation that the signature aggregation operation for some signature schemes is several magnitudes of times faster than that of their signature generation. One can leverage this observation to shift off-line expensive operations of the signature

generation phase. That is, we off-line compute a set of signatures on the bit-structures of the hash output domain. Later, these pre-computed signatures very efficiently. An approach based on the combination of pre-computation techniques and signature aggregation protocols has been recently proposed [57].

However, as the most efficient scheme depends on the specific scenario, it is important to develop scenarios for different applications, such as augmented reality, and IoT systems, and identify the most effective scheme(s) for each scenario. Based on these scenarios, one can design and implement a multi-schema digital signature service that can support different signature schemes for different applications.

Scalable Infrastructure for Signature Management: In 5G NDN networks, even small devices (such as IoT devices) can be data producers and, thus, must have their own private/public keys (as most common schemes are based on public-key cryptography). Considering the large number of such devices, a scalable key management scheme is thus required. The traditional Certificate-Authority-based Public Key Infrastructure (CA-PKI) is not well-suited for NDN since it can be a single-point-of-failure problem due to its centralized nature. Therefore, it is important to design a distributed/scalable PKI with an appropriate key management scheme. In order to address such an issue, one possible approach is to utilize the blockchain technology so that the role of the traditional CAs (i.e., binding an ID with a public key by signing them) is replaced with the proof-of-work of the blockchain networks. One can also consider a hybrid approach combining CA-PKI with the blockchain-based PKI or PGP Web-of-Trust [1] in order to support various NGN applications. However in order to achieve very small response times for retrieving information authentication information from blockchain, it is critical not only to adopt the most efficient blockchain technologies, but also to investigate caching strategies that can further improve such approaches and conduct experimental assessment of the various approaches.

Concurrent Execution of Signature Operations:
Many NDN applications, especially in the context of 5G networks, may require the same device to transmit/receive data from many different devices within a very short time. It is thus critical that the device be able to simultaneously execute many different signature operations (e.g. verifying a digital signature, or generating a digital signature). To address this issue, a possible approach is to use hardware-based acceleration techniques, such as techniques based on the use of GPU [56] available for example on systems-on-chip of vehicles.

5 ACCESS CONTROL

At a higher level, an access control system is based on an access control model (such as the discretionary model and the mandatory model [15]). When a discretionary model is adopted, the system uses a set of permissions to decide whether access to a protected resource can be granted.

Permission Specification: A permission typically consists of three components: (subject-specification,
object-specification, action-specification). For example, the permission (Bob, ND, Read) states that user Bob can read the data packet with name ND. Many variations exist with respect to such specifications. We now explore each of the items in this specification below.

Object Specification: It specifies the object(s) to which a subject is granted a given action. There are many variations for specifying objects. For example, the object can be specified by name (name-based access control) or by content (content-based access control). We argue that name-based access control combining the name of the packet and its namespace is the most suitable for NDN as every object has a unique name within a given namespace. Further, it is consistent with the name based matching of interest packets to data packets in NDN.

Subject Specification: Even more options are possible for subject specification; the most notable being: user-ids, roles (as in the popular RBAC model [48] which has been standardized by NIST [28]), and attribute-based denotations (as in the ABAC [60] model adopted by the XACML standard [15]). Out of all the above subject specification options, ABAC is the most expressive and flexible specification which can support fine-grained access control in NDN.

Action Specification: It indicates which actions a subject can perform on a given object. Depending on the application, context, and/or the system, various action specification schemes are utilized. For example, create, read, update and delete (CRUD) operations are commonly supported by persistent storage systems and PUT, GET and UPDATE operations are frequently supported by RESTful web APIs. The most important operation in NDN is the "read" action by which data consumers can read the protected data according to the permissions in the system. Thus access control should be optimized for "read" action by data consumers and "write" action by data producers. One may extend the system to support other actions such as "update" and "delete" at the expense of additional mechanisms in place.

Enforcement Mechanism: In addition to differences with respect to the access control model, access control systems differ with respect to the enforcement approaches. Well known approaches include access control lists, and encryption. The actual enforcement mechanism to be adopted depends however from the system architecture and the types of actions to be controlled. Since NDN follows a decentralized link-to-link communication model compared to a connection based one, it is important to choose an enforcement mechanism whose reference monitor is not centralized. Especially in the context of NDN over 5G networks, having to contact some centralized server for access control enforcement is not suitable when there are real-time constraints. Further, the enforcement mechanism must be able to exploit caching mechanisms in NDN to support caching of information needed for access control. Thus, we argue that an encryption based enforcement mechanism can satisfy these enforcement requirements that the NDN specification demands. In addition to controlling access to data packets, encryption enforces confidentiality of data packets at content level which current transport level security protocols such as HTTPS fail to support due to inherent limitations of the current Internet architecture. For example, today, many CDNs and middleboxes intercept HTTPS traffic in order to make routing and content optimization decisions violating HTTPS's model of an end-to-end encrypted connection. With encryption at content level, confidentiality of data packets

is guaranteed irrespective of how those packets arrived at data consumers.

Designing and developing an efficient and flexible access control system for NDN that meets the above mentioned access control requirements with strict performance guarantees and scalability is challenging. In what follows we discuss a possible approach to address the access control challenge.

Encryption based Access Control System for NDN: As mentioned earlier, one suitable enforcement strategy is based on encryption. An important design decision one needs to make is which cryptosystem to utilize to encrypt data packets. There are two main choices: public key cryptosystems (PKC) and symmetric key cryptosystems (SKC).

PKC based approaches can be mainly classified into three groups: (1) traditional PKI based schemes such as RSA, (2) Proxy Re-Encryption (PRE) [11] schemes, and (3) Attribute Based Encryption (ABE) [16, 31] schemes. However, such schemes have several weaknesses: they cannot efficiently handle the addition and/or the revocation of subjects, and policy changes; they require to keep multiple encrypted copies of the same key; they incur high computational cost. On the other hand, SKC based schemes, such as AES and Blowfish, are orders of magnitude faster than PKC based schemes and thus are the preferred to cryptosystem for NDN. However, SKC schemes have their own limitations. One needs to consider the challenge of how to generate a minimal number of keys to enforce the access control policies, how to enforce the access control policies over encrypted data, how to efficiently manage keys especially in a dynamic environment where new users join and existing users leave frequently, how to scale the key management scheme to a large number of names, data producers and data consumers, and how to efficiently deliver the keys to data consumers.

In a SKC based system, each data packet D is encrypted with a symmetric key K. Subjects authorized to read D receive the symmetric key for decrypting D, whereas the non-authorized subjects do not receive such key. Therefore, even if a non-authorized subject gets a copy of D (for example by intercepting messages transmitting D), it would not be able to decrypt D. Note that different data packets may be encrypted with different symmetric keys, depending on the authorizations associated with each data. The adoption of such a strategy however requires a mechanism to distribute the symmetric keys to the authorized users. An out-of-band communication channel is required to do so. One research challenge is thus to devise possible options to efficiently deliver the keys to data consumers. One approach is to utilize a PKC scheme, such as a traditional PKI scheme or an attribute-based encryption (ABE) scheme, to encrypt the symmetric keys using the public keys of data consumers. Another approach is to utilize a hybrid approach where a minimal number symmetric keys are delivered to data consumers utilizing a PKC scheme and then utilize the same underlying SKC scheme to deliver the remaining keys. As discussed in Section 4, a scalable and distributed PKI infrastructure is thus critical and is an open research challenge.

While the above scheme works well in a static environment where users and authorization policies are predefined and do not change over time, it is unable to efficiently handle user and authorization policy dynamics. A possible solution is to adopt and extend

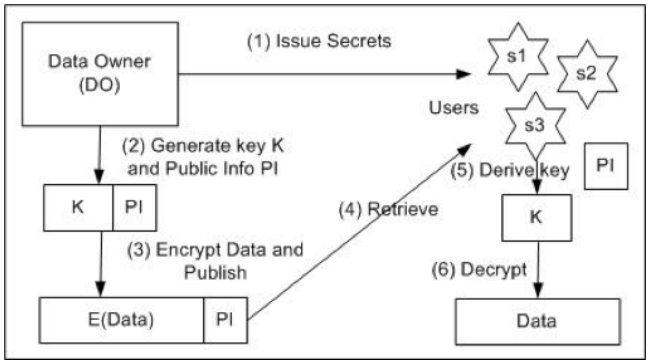

Figure 3: Broadcast Group Key Management Scheme

an approach based on our earlier work [41, 44]. Our previous approach is based on the idea that instead of directly distributing to the authorized subjects the symmetric keys for decrypting the data packets, one can allow the subjects to dynamically derive the keys at the time of decryption.

As shown in Figure 3, the basic idea of such an approach, referred to as *Broadcast Group Key Management* (BGKM), is to generate and distribute secrets to users based on their identity attributes (such as user-id, role, etc.) and later allow them to derive actual symmetric keys based on their secrets and some public information. Having only the public information in data packets does not allow data consumers to derive the underlying key used to encrypt the data packet. The ability to derive the key depends on whether a given data consumer satisfies the authorization policy encoded in the public information. A key advantage of the BGKM scheme is that adding users/revoking users or updating the permission can be performed efficiently and only requires updating the public information attached to subsequent data packets.

The BGKM scheme satisfies several requirements: minimal trust, key indistinguishability, key independence, forward secrecy, backward secrecy, and collusion resistance with minimal computational, space, and communication cost. We now provide a brief technical description of this primitive. The key idea of BGKM is to hide the symmetric data encryption key into a public data structure that is generated as a function of the secrets of the authorized users. These secrets map one on one to the attributes that data consumers possess (e.g. driver license, age, and the role played at work). Therefore, only users that have those secrets can extract the key. The two basic operations of BGKM are: generation of the public information hiding the key; and key derivation to get at the data decryption key. Below, we provide a high-level technical presentation of the adapted BGKM solution for NDN. We refer the reader to [44, 51] for additional technical details and proofs.

Let the data producer be DP [1] and a set of data consumers $DC_i, i = 1, 2, \ldots, n$.

paramgen It generates the parameters required to initialize. DP takes a security parameter ℓ. DP chooses an ℓ-bit prime number q, a positive integer $N \geq n$ which represents the maximum allowed

[1]There can be many data producers, but for simplicity of presentation only one is considered.

number of group members, and a cryptographic hash function

$$H(\cdot) : \{0,1\}^* \rightarrow \mathbb{F}_q,$$

where $\mathbb{F}_q$ is a finite field with q elements, which can be represented by $\{0, 1, \ldots, q-1\}$ with modular arithmetic. *DP* sets the keyspace $\mathcal{KS} = \mathbb{F}_q$. $param = \langle \mathcal{KS}, N, H(\cdot), \rangle$. where *param* consists of all public parameters.

secgen It generates secrets for each *DC*. For each $1 \leq i \leq n$, *DP* chooses a random bit string $s_i \in \{0,1\}^*$ as a secret for each DC_i, and sends s_i to DC_i. *DP* saves these s_i together with the group's membership information locally. Without loss of generality, we also assume that $s_i \neq s_j$ for $i \neq j$. In practice, an s_i is chosen long enough (e.g., ≥ 80 bits) so that guessing becomes infeasible.

keygen It generates public information (*PI*) embedding data decryption key. *DP* picks a random $K \in \mathcal{KS}$ as the shared group key. *DP* chooses N random bit strings $z_1, z_2,$
$\ldots, z_N \in \{0,1\}^*$. *DP* creates an $n \times (N+1)$ $\mathbb{F}_q$-matrix

$$A = \begin{pmatrix} 1 & a_{1,1} & a_{1,2} & \ldots & a_{1,N} \\ 1 & a_{2,1} & a_{2,2} & \ldots & a_{2,N} \\ 1 & a_{3,1} & a_{3,2} & \ldots & a_{3,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_{n,1} & a_{n,2} & \ldots & a_{n,N} \end{pmatrix},$$

where

$$a_{i,j} = H(s_i || z_j), 1 \leq i \leq n, 1 \leq j \leq N. \quad (1)$$

DP then solves for a nonzero $(N+1)$-dimensional column $\mathbb{F}_q$-vector Y such that $AY = 0$. Note that such a nonzero Y always exists as the nullspace of matrix A is nontrivial by construction. Here we require that *DP* chooses Y from the nullspace of A uniformly randomly. *DP* constructs an $(N+1)$-dimensional $\mathbb{F}_q$-vector which we call an *access control vector*

$$ACV = K \cdot e_1^T + Y,$$

where $e_1 = (1, 0, \ldots, 0)$ is a standard basis vector of $\mathbb{F}_q^{N+1}$, v^T denotes the transpose of vector v, and K is the pre-chosen shared group key. *DP* lets $PI = \langle ACV, (z_1, z_2, \ldots, z_N) \rangle$, and broadcasts *PI* via the broadcast channel.

keyder This method derives the data decryption key based on a set of secret a user possesses. Having s_i and *PI*, DC_i computes $a_{i,j}, 1 \leq j \leq N$, as in formula (1) and sets an $(N+1)$-dimensional row $\mathbb{F}_q$-vector

$$v_i = (1, a_{i,1}, a_{i,2}, \ldots, a_{i,N}).$$

DC_i derives the group key as $K' = v_i \cdot ACV$.

update It updates *PI* to reflect the user dynamics of leaving and joining. *DP* runs the *keygen* phase again with respect to the current group users, creates a new group key $\hat{K}$ and random $\hat{z}_i, 1 \leq i \leq N$, and broadcasts $\widehat{PI} = \langle \hat{X}, (\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_N) \rangle$ via the broadcast channel. A current *DC* derives the shared group key by following the same procedure specified in the *keyder* phase.

An important practical issue is how to support expressive access control policies over encrypted data. This requires encoding an access control policy into the above PI data structure. One possible approach is that that an access control policy is represented as an access tree, where each node in the tree can be represented using an instance of BGKM. Data consumers who have secrets to climb up the tree from leaf nodes all the way up to the root node are able to get the key to decrypt the associated data packets. Therefore, an interesting direction is to adapt and extend such an approach supporting expressive access control policies over encrypted data [41] in order to provide expressive policies over NDN.

Another interesting extension is based on the fact that names in NDN follow a hierarchical structure. An important consideration is whether we can leverage this fact to enhance the BGKM based access control mechanism with respect to the number of secrets that need to be shared. A key challenge is to design meaningful and efficient schema mappings similar to the concept of trust schemas [59] in order to identify hierarchically ordered data packets and the corresponding keys.

With the scale of operations, data producers may find it challenging to keep up with the access control requirements. Another important practical consideration is that how to efficiently delegate some of the access control enforcement functionality to intermediaries without compromising confidentiality. One possible approach is to utilize a two layer encryption approach where data producers enforce a coarse grained access control over data and intermediaries enforce fine grained access control over data. One research direction is thus to extend our previous approach on delegated encryption based access control on cloud based systems [42] and adapt a similar approach in NDN in order to reduce the load on data producers.

In order to make the scheme practical, one must overcome usability challenges involved in using encryption based access control mechanisms. The mechanism should be as transparent as possible to users of the system. One possible direction in this regard is to utilize browser based proxies, similar to current password managers, to hide the complexity of key management, encryption and decryption operations from users.

6 PRIVACY

As we mentioned in the introduction, comprehensive privacy in NDN requires combining different techniques that we discuss in what follows.

Anonymous Communication: Lack of such a mechanism makes it easy for an attacker to trace back to a sender of a message received by a destination, and vice versa. One possible approach is to utilize a network anonymizer that supports anonymous communications; a very well known example of such an anonymizer is represented by Tor [25]. An anonymizer makes it much more difficult for an attacker to trace back to the sender of a message received by the destination.

Starting from the notion of Onion routing, on which Tor was implemented, several other network anonymizers have been proposed. However, most existing anonymizers have scalability and performance issues. Recent approaches, like LAP [32] and [49] have addressed performance issue, but at the expense of reduced security. A recent scheme, HORNET, by Chen et al. [22] has addressed the problem of high performance and stronger security. In particular, HORNET uses only symmetric key encryption, which enhances efficiency. Scalability in HORNET is ensured by the fact that HORNET routers do not keep per-flow state or perform computationally expensive operations. A major issue in applying HORNET to NDN

is that it requires the client to know in advance the IP address of the destination, whereas in a NDN the client only needs to know the searched data name. One research direction is to investigate the use of HORNET in the context of NDN to identify additional issues, and design caching approaches that can allow the client to determine the possible paths leading to a given data.

Private Searches: Anonymous communication is not sufficient to ensure privacy because by looking at the interest packets and/or data packets of the requested data, an attacker may combine this information with other available information and link back the request to a specific subject. In other words, a network anonymizer only prevents an attacker from knowing from which IP address a given data was requested, not which data was requested. Further, setting up a communication anonymizer requires setting up a set of anonymizing routers from the client which must choose the path to follow and setting up such a network anonymizer may not be always possible. Therefore, a complementary mechanism to actually hide the actual data, in both interest packets and data packets, is required.

Efficient and effective countermeasures must be taken to prevent the data consumer's interests from being inferred according to the submitted interest packets. Several approaches could be adopted, including techniques for private-retrieval [58], but such techniques are not scalable and efficient for use in NDN systems, especially over 5G networks. One alternative approach is the use of the cover file notion proposed by Arianfar et al. [10]. Under such an approach a file F of interest is split into different chunks and the content of each chunk is randomized by combining it, through an exclusive OR operation, with another chunk. The latter can be a chunk from the same file F or from another file, referred to as cover file. This approach also includes a strategy for generating secure names for the various transformed chunks. A major issue in the use of this approach is that data users need to know the cover blocks used for generating the blocks of the file to be retrieved. The approach by Arianfar et al. does not indicate how such information is transmitted to the data users. To address this issue one possible approach is the use of encryption-based access control mechanisms, such as the one, described in the previous section. Information about the cover blocks for a given file would be encrypted and the encryption key made available only to authorized users. An additional research direction is the design of techniques by which one can select more than one blocks of actual interest by retrieving more chunks than needed in order to support a level of plausible deniability. We note however that the solution of Arianfar et al. does not work when one of the authorized users is a malicious party trying to infer privacy sensitive information about other authorized users. To address such scenario, one can explore privacy-preserving publishing techniques [40, 43].

In all the privacy approaches that we outlined, search patterns are revealed to intermediate NDN routing nodes, especially if network anonymization is not utilized. Oblivious RAM (ORAM) [54] is one of the best tools we have today for hiding such access patterns. Although ORAM is currently slow, an open challenge is how to build faster schemes specifically for NDN routing. If developed, such a technique can be added on top of existing privacy and security techniques proposed for NDN to hide the access patterns in NDN routing.

Cache Privacy: One important NDN feature is router-side content caching. While it helps to reduce congestion and improve throughput/latency, it can leak the interests of data consumers to nearby curious or malicious users. The most effective counter measure against cache sniffing attacks has been to randomize the caching strategy [5, 35]. Such approaches delay response time and increase congestion. Further, the privacy they provide is not well defined. It is thus an open challenge to support caching privacy with concrete privacy guarantees, such as the one defined by the differential privacy model [26] without degrading the response and the throughput.

7 CONCLUSIONS

NDN is a promising next-generation content-centric network architecture proposed for content distribution. Unlike current connection based security which is patched on top of the TCP/IP protocol stack, NDN takes a clean-slate approach to incorporate security from the beginning. However, achieving the security goals of NDN networks is challenging due to performance requirements, the scale and the highly dynamic behavior of users and content. In this paper, based on the characteristics and design goals of NDN, we identify security and privacy requirements, and propose possible directions towards meeting those requirements. There may well be other practical issues related to content security that we may have overlooked. We believe this research agenda will serve as a basis to identify other practical requirements as well as help make secure NDN a reality by addressing the challenges discussed in this paper.

ACKNOWLEDGEMENTS

This work was supported by the NSF grant CNS-1719369, and Intel as part of the NSF/Intel ICNWEN program.

REFERENCES

[1] https://www.gnupg.org/gph/en/manual/book1.html. (????).
[2] Digital Signature Standard (DSS). http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf. (????).
[3] eBACS: ECRYPT Benchmarking of Cryptographic Systems. http://bench.cr.yp.to/results-sign.html. (????).
[4] 2006. IETF ICN Working Group. https://goo.gl/wVHPWg. (2006).
[5] N. Abani and M. Gerla. 2016. Centrality-based caching for privacy in Information-Centric Networks. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*. 1249–1254.
[6] G. Acs, M. Conti, P. Gasti, C. Ghali, and G. Tsudik. 2013. Cache Privacy in Named-Data Networking. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*.
[7] Alexander Afanasyev, J. Alex Halderman, Scott Ruoti, Kent Seamons, Yingdi Yu, Daniel Zappala, and Lixia Zhang. 2016. Content-based Security for the Web. In *Proceedings of the 2016 New Security Paradigms Workshop (NSPW '16)*. ACM, 49–60.
[8] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin, and L. Zhang. 2017. NDNS: A DNS-Like Name Service for NDN. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. 1–9.
[9] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. Resource Records for the DNS Security Extensions. https://tools.ietf.org/html/rfc4034. (2005).
[10] Somaya Arianfar, Teemu Koponen, Barath Raghavan, and Scott Shenker. 2011. On preserving privacy in content-oriented networks. In *2011 ACM SIGCOMM Workshop on Information-Centri Networking, ICN 2011, Toronto, Canada, August 19, 2011. Proceedings*. 19–24.
[11] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. 2006. Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage. *ACM Transactions on Information System Security* (2006), 1–30.
[12] Hitesh Ballani, Paul Francis, and Xinyang Zhang. 2007. A Study of Prefix Hijacking and Interception in the Internet. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '07)*. ACM, 265–276.

[13] Mihir Bellare, Thomas Ristenpart, Phillip Rogaway, and Till Stegers. 2009. Format-Preserving Encryption. In *Selected Areas in Cryptography*. Springer Berlin Heidelberg, 295–312.
[14] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. 2012. High-speed high-security signatures. *J. Cryptographic Engineering* 2, 2 (2012), 77–89.
[15] Elisa Bertino, Gabriel Ghinita, and Ashish Kamra. 2011. Access Control for Databases: Concepts and Systems. *Foundations and Trends in Databases* 3, 1-2 (2011), 1–148.
[16] J. Bethencourt, A. Sahai, and B. Waters. 2007. Ciphertext-Policy Attribute-Based Encryption. In *2007 IEEE Symposium on Security and Privacy (SP '07)*. 321–334.
[17] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. 2004. Public Key Encryption with Keyword Search. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, Christian Cachin and Jan L. Camenisch (Eds.). Springer Berlin Heidelberg, 506–522.
[18] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. 2003. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*. 416–432.
[19] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. 2015. Leakage-Abuse Attacks Against Searchable Encryption. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 668–679.
[20] V. Cerf and R. Kahn. 1974. A Protocol for Packet Network Intercommunication. *IEEE Transactions on Communications* 22, 5 (1974), 637–648.
[21] Abdelberi Chaabane, Emiliano De Cristofaro, Mohamed Ali Kaafar, and Ersin Uzun. 2013. Privacy in Content-oriented Networking: Threats and Countermeasures. *SIGCOMM Computer Communication Review* 43, 3 (2013), 25–33.
[22] Chen Chen, Daniele Enrico Asoni, David Barrera, George Danezis, and Adrian Perrig. 2015. HORNET: High-speed Onion Routing at the Network Layer. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*. 1441–1454.
[23] Mauro Conti, Paolo Gasti, and Marco Teoli. 2013. A Lightweight Mechanism for Detection of Cache Pollution Attacks in Named Data Networking. *Computer Networking* 57, 16 (2013), 3178–3191.
[24] Giovanni Di Crescenzo, Brian Coan, John Schultz, Simon Tsang, and Rebecca N. Wright. 2014. Privacy-Preserving Publish/Subscribe: Efficient Protocols in a Distributed Model. In *Proceedings of the 8th International Workshop on Data Privacy Management and Autonomous Spontaneous Security*, Joaquin Garcia-Alfaro, Georgios Lioudakis, Nora Cuppens-Boulahia, Simon Foley, and William M. Fitzgerald (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 114–132. DOI: http://dx.doi.org/10.1007/978-3-642-54568-9_8
[25] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. 2004. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium, August 9-13, 2004, San Diego, CA, USA*. 303–320.
[26] C. Dwork, F. McSherry, K. Nissim, and A. Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of the Third Conference on Theory of Cryptography*. Springer-Verlag, 265–284.
[27] Ksenia Ermoshina, Francesca Musiani, and Harry Halpin. 2016. End-to-End Encrypted Messaging Protocols: An Overview. In *Internet Science*. Springer International Publishing, 244–254.
[28] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. 2001. Proposed NIST Standard for Role-based Access Control. *ACM Transactions on Information Systems Security* 4, 3 (2001), 51.
[29] Roy Fielding. Architectural styles and the design of network-based software architectures. https://goo.gl/tDx9JZ. (????).
[30] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang. 2013. DoS and DDoS in Named Data Networking. In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*. 1–7.
[31] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. 2006. Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*. ACM, 89–98.
[32] Hsu-Chun Hsiao, Tiffany Hyun-Jin Kim, Adrian Perrig, Akira Yamada, Samuel C. Nelson, Marco Gruteser, and Wei Meng. 2012. LAP: Lightweight Anonymity and Privacy. In *IEEE Symposium on Security and Privacy, SP 2012, 21-23 May 2012, San Francisco, California, USA*. 506–520.
[33] Van Jacobson. 2006. A new way to look at networking. https://goo.gl/VGwkUu. (2006).
[34] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. 2009. Networking Named Content. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '09)*. ACM, 1–12.
[35] M. Jakobsson and S. Stamm. 2007. Web Camouflage: Protecting Your Clients from Browser-Sniffing Attacks. *IEEE Security Privacy* 5, 6 (2007), 16–24.
[36] N. Karapanos, A. Filios, R. A. Popa, and S. Capkun. 2016. Verena: End-to-End Integrity Protection for Web Applications. In *2016 IEEE Symposium on Security and Privacy (SP)*. 895–913.
[37] D. Kim, J. Bi, A. V. Vasilakos, and I. Yeom. 2017. Security of Cached Content in NDN. *IEEE Transactions on Information Forensics and Security* 12, 12 (2017), 2933–2944.
[38] Kevin Lewi and David J. Wu. 2016. Order-Revealing Encryption: New Constructions, Applications, and Lower Bounds. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1167–1178.
[39] Ralph C. Merkle. 1987. A Digital Signature Based on a Conventional Encryption Function. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*. 369–378.
[40] Mohamed Nabeel, Stefan Appel, Elisa Bertino, and Alejandro Buchmann. 2013. Privacy preserving Context Aware Publish Subscribe Systems. In *Network and System Security - 7th International Conference, NSS 2013, Madrid, Apain, June 3-4, 2013. Proceedings*. 465–478.
[41] Mohamed Nabeel and Elisa Bertino. 2014. Attribute Based Group Key Management. *Trans. Data Privacy* 7, 3 (2014), 309–336.
[42] M. Nabeel and E. Bertino. 2014. Privacy Preserving Delegated Access Control in Public Clouds. *IEEE Transactions on Knowledge and Data Engineering* 26, 9 (2014), 2268–2280.
[43] Mohamed Nabeel, Ning Shang, and Elisa Bertino. 2012. Efficient Privacy Preserving Content Based Publish Subscribe Systems. In *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies (SACMAT '12)*. ACM, New York, NY, USA, 133–144. DOI: http://dx.doi.org/10.1145/2295136.2295164
[44] Mohamed Nabeel, Ning Shang, and Elisa Bertino. 2013. Privacy Preserving Policy-Based Content Sharing in Public Clouds. *IEEE Trans. Knowl. Data Eng.* 25, 11 (2013), 2602–2614.
[45] Muhammad Naveed, Seny Kamara, and Charles V. Wright. 2015. Inference Attacks on Property-Preserving Encrypted Databases. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 644–655.
[46] Christian Rossow. 2014. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *In Proceedings of the 2014 Network and Distributed System Security Symposium, NDSS*.
[47] Mark D. Ryan. 2014. Enhanced certificate transparency and end-to-end encrypted mail. In *In Network and Distributed System Security Symposium (NDSS). Internet Society*.
[48] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. 1996. Role-based access control models. *Computer* 29, 2 (1996), 38–47.
[49] Jody Sankey and Matthew K. Wright. 2014. Dovetail: Stronger Anonymity in Next-Generation Internet Routing. In *Privacy Enhancing Technologies - 14th International Symposium, PETS 2014, Amsterdam, The Netherlands, July 16-18, 2014. Proceedings*. 283–303.
[50] N. Shang, M. Nabeel, F. Paci, and E. Bertino. 2010. A privacy-preserving approach to policy-based content dissemination. In *Proceedings of the IEEE 26th International Conference on Data Engineering*. 944–955.
[51] N. Shang, M. Nabeel, F. Paci, and E. Bertino. 2010. A Privacy-Preserving Approach to Policy-Based Content Dissemination. In *ICDE '10: Proceedings of the 2010 IEEE 26th International Conference on Data Engineering*.
[52] Diana Smetters and Van Jacobson. 2009. Securing network content. (2009).
[53] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. 2000. Practical Techniques for Searches on Encrypted Data. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society.
[54] Emil Stefanov and Elaine Shi. 2013. ObliviStore: High Performance Oblivious Cloud Storage. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 253–267.
[55] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos. 2014. A Survey of Information-Centric Networking Research. *IEEE Communications Surveys Tutorials* 16, 2 (2014), 1024–1049.
[56] Attila Altay Yavuz. 2014. An Efficient Real-Time Broadcast Authentication Scheme for Command and Control Messages. *IEEE Transactions on Information Forensics and Security* 9, 10 (2014), 1733–1742.
[57] Attila Altay Yavuz, Anand Mudgerikar, Ankush Singla, Ioannis Papapanagiotou, and Elisa Bertino. 2017. Real-time digital signatures for time-critical networks. *IEEE Transactions on Information Forensics and Security* 12, 11 (2017), 2627–2639.
[58] Sergey Yekhanin. 2010. Private Information Retrieval. *Communication of ACM* 53, 4 (2010), 68–73.
[59] Yingdi Yu, Alexander Afanasyev, David Clark, kc claffy, Van Jacobson, and Lixia Zhang. 2015. Schematizing Trust in Named Data Networking. In *Proceedings of the 2Nd ACM Conference on Information-Centric Networking (ACM-ICN '15)*. ACM, 177–186.
[60] E. Yuan and J. Tong. 2005. Attributed based access control (ABAC) for Web services. In *IEEE International Conference on Web Services (ICWS'05)*. 569.
[61] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. Named Data Networking. *SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.