NDN-ABS: Attribute-Based Signature Scheme for Named Data Networking

Sanjeev Kaushik Ramani
Florida International University
Miami, Florida
skaus004@fiu.edu

Reza Tourani
Saint Louis University
St. Louis, Missouri
reza.tourani@slu.edu

George Torres
New Mexico State University
Las Cruces, New Mexico
gtorresz@cs.nmsu.edu

Satyajayant Misra
New Mexico State University
Las Cruces, New Mexico
misra@cs.nmsu.edu

Alexander Afanasyev
Florida International University
Miami, Florida
aa@cs.fiu.edu

ABSTRACT

The Named Data Networking architecture mandates cryptographic signatures of packets at the network layer. Traditional RSA and ECDSA public key signatures require obtaining signer's NDN certificate (and, if needed, the next-level certificates of the trust chain) to validate the signatures. This potentially creates two problems. First, the communication channels must be active in order to retrieve the certificates, which is not always the case in disruptive and ad hoc environments. Second, the certificate identifies the individual producer and thus producer anonymity cannot be guaranteed if necessary.

In this paper, we present NDN-ABS, an alternative NDN signatures design based on the attribute-based signatures, to addresses both these problems. With NDN-ABS, data packets can be verified without the need for any network retrieval (provided the trust anchor is pre-configured) and attributes can be designed to only identify application-defined high-level producer anonymity sets, thus ensuring individual producer's anonymity. The paper uses an illustrative smart-campus environment to define and evaluate the design and highlight how the NDN trust schema can manage the validity of NDN-ABS signatures. The paper also discusses performance limitations of ABS and potential ways they can be overcome in a production environment.

CCS CONCEPTS

• **Security and privacy → Security protocols**; • **Networks → Security protocols**; **Network security**; **Network privacy and anonymity**; *Naming and addressing*.

KEYWORDS

Information-Centric Networking, Named Data Networking, Attribute-Based Signatures, Conditional Privacy, Producer anonymity.

ACM Reference Format:
Sanjeev Kaushik Ramani, Reza Tourani, George Torres, Satyajayant Misra, and Alexander Afanasyev. 2019. NDN-ABS: Attribute-Based Signature Scheme for Named Data Networking. In *6th ACM Conference on Information-Centric Networking (ICN '19), September 24–26, 2019, Macao, China.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3357150.3357393

1 INTRODUCTION

Named Data Networking (NDN) [1, 36] architecture is the most prominent realization of the Information-Centric Networking (ICN) vision, where a client (consumer) pulls desired information (Data packet) from the network by sending a named request (Interest packet). NDN, by design, provides inherent security features, such as data integrity and provenance as well as producer's trust assessment, through data signatures and the NDN trust schema [33]. The unique characteristics of name-based data retrieval, leveraging of caching, security, trust have generated interest in the use of NDN as the network layer for applications in new, challenging environments for traditional networking (e.g., autonomous vehicles, edge computing, augmented/virtual reality). However, due to NDN's nascency, several challenges, particularly in security still need to be addressed for large-scale adoption.

In this paper, we study two of such problems and propose an NDN attribute-based signature (NDN-ABS) mechanism as a potential solution. First, is how to verify the signature of a data packet without requiring additional retrieval of certificates, which can be challenging, or impossible in disruptive and ad hoc environments. With the proposed NDN-ABS, as we highlight in Section 4, consumers can verify the signature and ensure integrity and authenticity of a data packet without any additional information, provided they are provisioned with the attribute authority's public parameters (i.e., NDN-ABS trust anchor). A user requests the public parameters from the authority only once and then installs and stores them on his/her device (e.g., in a persistent storage for later reuse). Moreover, the authority's public keys are constant and do not change irrespective of the number of attributes used in the signature generation.

The second problem we are addressing is how to ensure integrity and provenance of the content while preserving anonymity of individual publishers. In other words, a system where neither signatures nor certificates can be used to correlate a set of data to a single entity. By using attribute-based signatures, the producers can sign

ICN '19, September 24–26, 2019, Macao, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6970-1/19/09…$15.00
https://doi.org/10.1145/3357150.3357393

data using attributes of varying granularities, revealing more or less about themselves, as required by the system design. The traditional identity-based signature schemes proposed in [3, 10, 20, 29] cannot provide producer anonymity which is a major contribution of the proposed scheme.

In particular, in our illustrative example of augmented reality application in a smart campus environment (Section 2), mobile producers can sign data merely with "<Affiliation>" and "<Position>" attributes; while edge devices can sign with "EdgeNode-<X>" attributes. In the former case, while data can be attributed to a large group of campus students/faculty/etc., the latter case explicitly identifies the node that produced the data.

Our contributions are five-fold:

- We designed NDN-ABS by integrating ABS signatures as part of NDN protocol operations (Section 4): (a) defined a new signature type; (b) data formats and naming for ABS elements (public parameters of the attribute authority); (c) naming structure for NDN signature key locator, identifying authority and signing policy; and (d) defined how NDN trust schema can validate attribute-based signatures.
- We defined a specific mechanism to ensure time-limited validity of NDN-ABS signatures, approximating validity periods of traditional certificate-based signatures (Section 4.8).
- We created the first comprehensive prototype implementation of ABS signature mechanism, which was proposed by Maji et al. [19] in 2011, but, to the best of our knowledge, did not have a standalone codebase support.[1]
- We evaluated ABS signature performance overhead and proposed potential ways to optimize signing and verification in production environments (Section 6).
- We discussed NDN-ABS in the context of multiple attribute authorities, ABS signature revocation strategies, and NDN-ABS adoption challenges (Section 7).

2 MOTIVATION

The mandate that all NDN data packets are signed makes the proposed signature scheme applicable for many ICN applications. To better illustrate the need for NDN-ABS and highlight the design components, we use a simplified version of an NDN-based augmented reality (AR) application within a university campus environment. In this application, AR-enabled smartphones of students, faculty, staff, and campus guests are publishing video streams, which are being processed (at the edge, in the cloud, or by a peer device) to identify points of interest; the AR application then pulls the identified information and augments the user's view. In essence, the application enables the authoring and experience of rich AR by members and visitors of the campus community while using NDN to leverage networking and security of communication.[2] The use of AR in a smart-campus environment broadens the user's perception of a location with sound, image, and video from multiple sources after having processed them.

One of the expected features of this application is reliance on ad hoc communication that is enabled by ICN/NDN technology. In other words, whenever there is any type of connectivity (phone-to-edge, phone-to-phone), data can be immediately retrieved, analyzed, and results of the analysis made available for retrieval. As mandated by NDN architecture, all data must be secured by signing individual or groups of packets (i.e., aggregated signatures using a manifest mechanism [23]). However, the existing signature mechanisms defined in the NDN packet specification [24] have two distinct disadvantages in our scenario:

- with possibility of ad hoc connectivity, there is no guarantee that keys to verify data (the certificate chains) will still be available after retrieving the data; and
- the corresponding public key of the signature can be used to identify individual data producer, unless the same private key is shared among different users (a dangerous practice).

When using NDN-ABS, these two problems can be effectively solved. For example, the *campus registrar* can act as the attribute authority (AA) to issue attribute secret keys (ska) to authorized entities:[3]

- students, faculty, staff, and other affiliates can receive "[campus-name]", "[title]", "[name]", and other attributes by authenticating with the campus portal or physically going to the registrar's office;
- the edge devices can be configured with "[campus-name]", "[unit-id]", etc., obtained by the responsible personnel; and
- guests can receive "guest" attribute if system verifies that they are physically present on campus.

In addition to attributes, the authority also publishes the public parameters Data packet that acts as a trust anchor, which can be provisioned on the devices during attribute request or through a dedicated bootstrapping protocol [21].

With this initial setup, user and edge devices can start publishing data that can be reliably authenticated. For signing, the application needs to define a policy predicate, e.g., a set of attributes combined with "AND" and "OR" operations (e.g., "[campus-name] AND ([title1] OR [title2])" or "[campus-name] AND [unit-id]"), which, along with the attribute signing keys, can create a verifiable signature. For verification, the receiver just needs to know the producer's claimed policy predicate (which is identified in the data packet itself) and the attribute authority's public parameters (which, as mentioned above, is already knows). This effectively addresses the first problem of traditional signatures (Figure 1).

The specific attributes used in the policy predicate allow producers to reveal or hide identity (actual identity or a pseudonym) of the individual producer. The properties of ABS construction guarantee [19] that (a) signatures are unforgeable; (b) two data packets signed by the same producer and with the same policy predicate cannot be linked to the producer, unless the producer explicitly identified himself in the predicate; and (c) two users cannot combine attribute private keys (*ska*) to create signature with the claim predicate that covers both user's attributes (collusion resistance

[1] Our implementation is based on code by Mauri Miettinen [22], that includes only the basic ABS framework.

[2] If we used HMAC keys along with an on-campus key service, the on-campus keying service will require the authority to give out anonymous identities and keys to an individual for different or all combinations of attributes. With the proposed NDN-ABS scheme, the attributes are distributed once and the user can sign with a combination of the available attributes.

[3] Note that the specific mechanism to determine which attribute can be used by which entity is outside the scope of this paper and NDN-ABS framework.

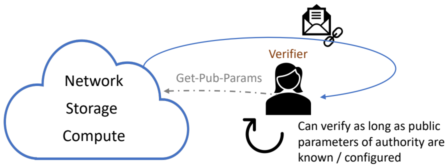
Network
Storage
Compute
Get-Pub-Params
Verifier
Can verify as long as public parameters of authority are known / configured

Figure 1: Ease of verification using NDN-ABS

property). In other words, it is not possible for user U_1 that has ska_a for "[affiliate]" attribute and user U_2 that has ska_r for "[registrar]" to collude and sign with the policy "[affiliate] AND [registrar]". With these essential properties, NDN-ABS provides a way to ensure anonymity of the individual from the edge-computing entity by creation of an anonymity set using the same attributes. This approach is different from the common edge computing services which currently run on edge resources in a sandbox environment wherein data is received from the owner or user of the service using her/his attributes. NDN-ABS thus can realize the desired level of conditional privacy, as illustrated in Figure 2.

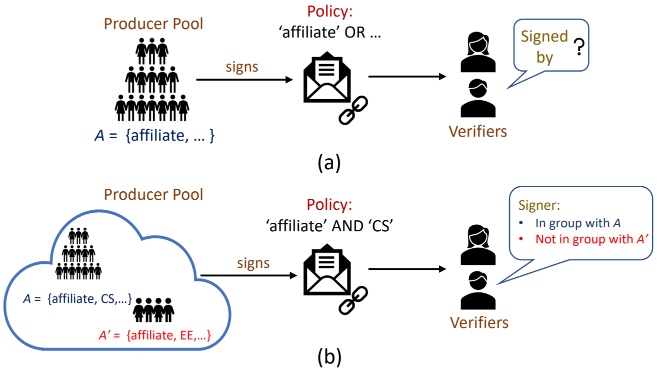

Figure 2: Conditional privacy using NDN-ABS; (a) Verifier unable to decide the identity of the signer; (b) Verifier can identify the group containing the signer

To summarize, the main benefits of NDN-ABS are: (a) the system can be designed to provide conditional privacy wherein the amount of detail revealed about a data-publisher can be controlled using the attributes used to sign the content; and (b) the data can be verified without the need to retrieve any additional information, such as keys in the certification chain.

3 BACKGROUND

3.1 Named Data Networking

NDN is a proposed networking architecture [1] that is designed to use names, which are associated with the content, as the means of retrieval. The NDN design gives the network the capability of retrieving named data in different ways and by treating all involved components, such as storage and computing and network devices in a similar way.

In NDN, data retrieval depends on two types of packets: *Interest* packet is a request for the specific data by name and *Data* packet is the named and secured piece of application data; these packets use the rich NDN naming conventions to aid in data retrieval. Securing NDN's Data packets ensures content integrity, authenticity, and (if encrypted) confidentiality irrespective of how the *Data* packet is retrieved. NDN provides components including *Content Store (CS)*, which acts as temporary data storage to support in-network caching; *Pending Interest Table (PIT)* for NDN's stateful forwarding plane and request aggregation; and *Forwarding Information Base (FIB)*, which acts as a routing table.

3.1.1 Security Advantages of NDN. NDN design has built-in security primitives via cryptographic signatures of the producer on all named data. The consumers (or any intermediate nodes) can verify data integrity and provenance using its name, which provides an essential context for security. NDN's data-centric security allows applications to control data access by using encrypted keys which by themselves are data packets to be retrieved. The immutable nature of data allows its storage in multiple containers without integrity loss and prevents non-repudiation attacks. Secure ways for content sharing and privacy-enhanced routing schemes in ICN based networks has been explored by researchers in [7, 27]. There are various other works that have identified the security advantages that an ICN approach has when adopted in the various applications.

3.2 Attribute-Based Encryption (ABE)

Attribute-Based Encryption (ABE) [8] which enables public key based one-to-many encryption, is a specialized form of identity-based encryption (IBE) introduced by Shamir in [29]. ABE is used for confidentiality rather than integrity and non-repudiation. It is a public key primitive with the potential for realizing scalable and fine-grained access control systems, providing a flexible approach for access rights assignment to an individual or a group of users. In ABE, each secret key is associated with an access structure–termed as a *predicate*–which specifies the type of ciphertexts the key can decrypt. ABE has two variants namely key-policy ABE (KP-ABE), in which the secret keys are generated according to access policy, and ciphertext-policy ABE (CP-ABE), which uses access policies for data encryption. The applicability of identity-based and attribute-based encryption in ICN is explored by Tohru et al. [3], A M Malik et al. [20], Mihaela et al. [10] and others. The use of ABE schemes in automating access control using NDN is discussed in our previous work [38]. Integrity and non-repudiation schemes, particularly ABS schemes, has neither been investigated in the ICN community nor has it been demonstrated with deployments in security earlier.

3.3 Attribute-Based Signature

Attribute-based signature [15, 19] is a variant of digital signatures made applicable for situations involving the use of attributes. ABS is an extension of the identity-based signatures, which generalizes the signing entity (signer) with a set of attributes. The identity-based signature schemes have their own set of advantages but will not be able to provide producer anonymity which is a major achievement of the ABS scheme. ABS and ABE use similar mathematical concepts of bilinear pairing and monotone span programs (Appendix A) to define signature and encryption policies, but their are *substantially* differ in the specific algorithmic steps. Some important terms related to ABS are as follows:

- ***Attribute Authority***: The authority involved in generation of public parameters *pk* and generating and supplying signers with the secret keys *ask* for attribute sets $\mathcal{A}$ that correspond to signers' authorized properties. The campus registrar, in the case of a smart-campus, can work as the attribute issuing authority. In the proposed NDN-ABS scheme, the Attribute authority is required to be online during the generation of the public parameters and when there is a need for re-keying or revocation. In all other instances, the system can work seamlessly even if the attribute authority is offline. [4]
- ***Signer***: The user creates message signature σ with a policy predicate Υ that defined over subset of attributes $\mathcal{A}$ using *ask* obtained from the attribute authority. On a smart-campus, we expect all the users of the AR application to sign the data that they publish and NDN-ABS provides a means to do such signing anonymously yet verifiable.
- ***Verifier***: The users who verify message signatures σ using public parameters *pk* of the authority (e.g., pre-provisioned and trusted) and the policy predicate Υ (e.g., extracted from the message). The users of our AR application and the other intermediate service providers will have to verify the content published by the users for the service to be provided seamlessly.
- ***Policy Predicate*** Υ: A boolean valued logical function that is constructed by combining attributes $\mathcal{A}$ using "`AND`", "`OR`", "`NOT`", and threshold gate operations. The predicate essentially is a logical claim of the signer that it possess a set of attributes. Unless all attributes in the predicate are combined with "`AND`", the claim does not identify which exactly set the signer posses.

4 NDN-ABS DESIGN

4.1 Overview

The functionality of the attribute-based signature scheme depends on the *attribute authority* who is entrusted with the responsibility of distributing attributes to other users (producers, consumers, forwarders, intermediate nodes) involved in the system. In the description, we use a single attribute authority, but our design generalizes to multi-authority systems as well.

Figure 3 depicts a typical ABS scenario where the first step involves generation of public parameters by the attribute authority which are published as data packet(s) and provisioned or retrieved by all parties. However, this process performed only once by the authority and the parameters do not change, except rare "rollover" events similar to root key change in today's DNSSEC. In the subsequent steps, the signers request and retrieve the public parameters and the secret signing key for attributes, e.g., using a modified version of the NDNCERT [37] framework. To verify signatures, the consumers extract information from Data packet's KeyLocator field containing the name of the attribute authority associated with the signature and the claim predicate of the signature.[5] The validity of an NDN-ABS signature ascertains that a *producer*, whose *attributes* satisfy the *predicate*, has indeed signed (endorsed) the message.

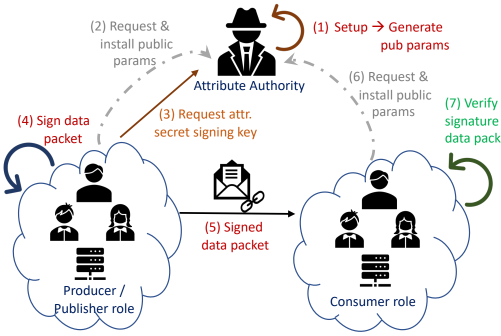

Figure 3: Overview of NDN-ABS

In what follows, we elaborate the design of an NDN-ABS system, which is based on the general ABS scheme.

4.2 Authority Setup

The setup phase is run by the attribute authority using the fields and generators defined based on the concepts of Bilinear pairing (See Appendix A.1). The setup stage results in the generation of the public parameters *pk* along with the secret key *ask*. Algorithm 1 shows the details of a new authority setup in the system. The public parameters are published as an NDN data packet with the name defined in Section 4.6 that can be easily retrieved when needed.

Algorithm 1 NDN-ABS Authority Setup

```
Input:  Authority name.
Output: Public parameters $pk$, Secret key $ask$.
1: Generate $pk$ and $ask$ as defined in the Appendix.
2: Generate name for $pk$ using authority name as input.
3: Publish $pk$ as an NDN data-packet.
```

4.3 Obtaining Attribute Set Secret Key

To sign data, the data producer needs to obtain the attribute set secret key, which can be used to create NDN-ABS signatures. This process can largely mimic the process of obtaining NDN certificates for RSA/ECDSA keys, with the exception that the key itself is being generated by the attribute authority. Specifically, to obtain the key, the producer requests from the attribute authority the key that corresponds to a set of attributes it is entitled to.[6] The authority runs the Secret-Attribute key generation process (Algorithm 2), generates, and returns the secret key. To compute the secret, the authority uses the algorithm described in Appendix A.2. Note that unless the producer provided the original secret key, the generated

[4] Rekeying overhead and other related details are documented in literature and are not discussed as a part of this paper as we are not attempting to solve this and does not have too much relevance to ICN.

[5] The numbers highlighting the steps in the figure are just a representation and the steps pertaining to the generation of public parameters, the request and installation of them are performed only once for an authority and do not repeat every time a data-packet has to be signed or to verify the signature.

[6] To which attributes the producer is entitled is outside the scope of this paper and NDN-ABS design in general.

key can be used to sign data with policy that reflect the new attribute set: old and new secret keys are not compatible by the ABS construction. However, if the key is provided, the authority can generate a key that covers existing and new attributes.

Algorithm 2 Secret signing-key generation

```
Input: Attribute set $\mathcal{A}$, Secret key $ask$.
Output: Secret Signing key $ska$.
if Secret key exists for $\mathcal{A}$ then
    Use components of the existing key.
    Append new information to the key.
else
    Use components of $ask$ to generate the $ska$.
end if
```

4.4 Signing

For the signing process, the data producer generates a claim-predicate policy over the attributes that it has previously obtained from the authority. In the simplest case, the producer can create policy combining all available attributes using "AND" operation (the implementation in our prototype) or use application-defined mechanism to select attributes and combination of "`AND`", "`OR`", "`NOT`", and threshold gates. After that, the producer creates "`SignatureInfo`" for NDN-ABS signature and appends it to the Data packet, formatting key locator field to include name of the attribute authority with which the producer has the relation and the encoded policy as defined in Section 4.6. Finally, the producer uses algorithm in Appendix A.1 to generate signature σ and add it to the packet. Algorithm 3 describes the signing process.

Algorithm 3 Signing a data-packet using NDN-ABS

```
Input: $pk$, $ska$, data packet, Policy $\Upsilon$.
Output: Signature $\sigma$.
Convert $\Upsilon$ into corresponding MSP $M$.
Retrieve data packet corresponding to $pk$ if necessary.
Generate SignatureInfo with KeyLocator information.
Generate the signature $\sigma$.
Append the signature to the data-packet.
```

4.5 Verification

In order to verify data, the consumer needs only two pieces of information: the data packet itself and the public key parameters pk of the corresponding attribute authority. Υ can be directly extracted from the KeyLocator name (see Section 4.6) and pk can be determined (and retrieved if needed) using the attribute authority name extracted from the same KeyLocator. After that, the verifier can simply run the process defined in the Appendix A.1 and determine validity of the signature. Algorithm 4 highlights the steps involved.

4.6 Naming

The NDN-ABS design uses several specialized naming conventions (Figure 4) to name Data packets with the public parameters of an authority and format "`KeyLocator`" name for the "`SignatureInfo`" of the Data packet signature.

Algorithm 4 Verifying an NDN-ABS signature

```
Input: Data packet.
Output: Signature "Accepted" or "Rejected".
Extract signature $\sigma$ from the data packet.
Extract key locator information from $\sigma$.
Extract policy $\Upsilon$ from the key locator.
Retrieve data packet corresponding to $pk$ if necessary.
Verify the signature.
```

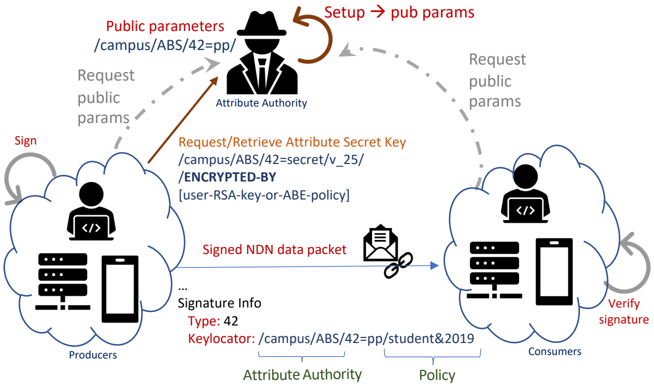

Figure 4: NDN-ABS naming and working

4.6.1 *Public Parameters Data.*

Each attribute authority in NDN-ABS is associated with a dedicated NDN namespace, e.g., "`/campus`". The authority publishes the corresponding public parameters, which are needed to verify signatures that use attributes issued by the authority, as an NDN data packet with the following name:

"`<attr-auth-prefix>/ABS/42=pp/<version>`"

where "`<attr-auth-prefix>`" is authority's namespace, "`/ABS/42=pp`" are the two fixed name components, and "`<version>`" is the version of the generated parameters. For example, "`/campus/ABS/42=pp/_v=1`" represents first version of NDN-ABS public parameters for the main campus authority.

4.6.2 *KeyLocator Name.*

When a producer signs data packet using NDN-ABS, it creates special name for the KeyLocator field of SignatureInfo element, consisting of two components:

"`<name-of-authority-public-params>/42=policy /<claim-predicate-policy>`"

The first part is simply the name of the Data packet that carries authority's public parameters defined in Section 4.6.1, and the second is the claim-predicate policy encoded using Algorithm 5 (the middle "`/42=policy`" acts as a separator).

For example, if a campus student previously obtained the secret key for "student" and "2019" attributes, it can sign data with "student AND 2019" claim predicate policy, which will be encoded as KeyLocator name "`/campus/ABS/42=pp/_v=1/42=policy/student&2019`".

4.6.3 *Attribute Secret Keys.*

As NDN-ABS requires authority to generate keys and deliver them to producers, these keys needs to be packaged as NDN data packets and encrypted. For this, we borrow the name-based access control mechanisms as defined in [38], which

Algorithm 5 Encoding of the attribute policy

```
AttributePolicy = Attribute *ExtraAttribute
ExtraAttribute = BooleanOperator Attribute

Attribute = DIGIT / LCASELETTER / UCASELETTER
DIGIT = %x30-39 ; 0-9
LCASELETTER = %x61-7a ; 'a'-'z'
UCASELETTER = %x41-5a ; 'A'-'Z'

BooleanOperator = "&" / "|"
```

results in the following naming structure for the encrypted attribute secret keys:

"`<name-of-authority-public-params>/42=secret/<version>`
`/ENCRYPTED-BY/[user's-RSA-key-or-ABE-policy]`"

For example, "`/campus/ABS/42=secret/_v=24/ENCRYPTED-BY/[user-key]`" is a 24th version of a secret key (i.e., it does not identify the set of attributes in the name) generated by the campus authority for the identified user.

4.7 Application of Trust Schema

The use of specially formatted KeyLocator name in NDN-ABS signature, allows its use as part of the NDN trust schema validation [33]. In particular, the extended version of the trust schema (our future work) can be programmed not only to validate the structural relationships between the data name and key name (e.g., that data under "`/campus`" namespace is signed with attributes issued by the campus authority), but also include relationship between data name and attribute policy. For example, the schema can define a rule

Data Name	"`(<>*)<AR><video><><>`"
Key Name	"`[@1]<ABS><><42=pp>[@{satisfy(‘‘student’’)}]`"

which requires: (a) the AR video data be signed by an NDN-ABS signature of the corresponding authority, and (b) the claim predicate satisfies policy "student", i.e., the policy is "student" or one of "OR" claims in the policy is "student".

4.8 Signature Validity

Different from RSA and ECDSA signatures, where validity of the signature is determined by the corresponding certificate, NDN-ABS signature require validity be defined in the signature itself (i.e., there is no certificate). Therefore, we define the following two approaches to be used for time-limited signatures:

- a general validity attribute ("valid-until-[date]") and
- time-restricted attributes, such as "student-until-[date]".

The first approach can be used if the attribute authority does not extend the attribute secret keys, i.e., whenever producer requests a set of attributes, all of them are bounded by the defined validity. For example, at different times, the producer may have obtained the secret keys for attribute sets ("student", "valid-until-[date1]") and ("faculty", "valid-until-[date2]"). By the ABS construction, the producer can only sign with the policy that includes attributes from one of the sets and cannot mix and match. Therefore, a correctly behaving authority can ensure the proper validity of created signatures.

The second approach can be used to restrict attributes themselves, in addition or instead of the general validity attribute. In this case, the signature can be considered valid if and only if all the time-restricted attributes are still valid at the time of verification. If the system requires a long-term validity of data signatures, e.g., for data archives, it can require the use of Delorean framework [34] or another blockchain-based assurance that signature was created before the expiration date.

5 ADVERSARY MODEL

The primary motive of any adversary in the NDN-ABS system would be to either

- try to forge a signature with a predicate/policy that does not satisfy her/his assigned attribute set.
- dissect the signature to get hold of the attributes in the predicate/policy to identify the specific individual who signed the message and thus breach the privacy.

NDN-ABS signatures are unforgeable owing to the condition that an adversary will not be able to generate a signature that will satisfy a given predicate if she/he does not possess the attributes (u^*) that satisfies $\Upsilon(u^*) = 1$ [19]. Moreover, a trustworthy attribute authority will not provide the attributes that does not correspond to the said user to be used in the extraction of the *secret signing key*. This argument also provides NDN-ABS with resiliency to collusion attacks. A collusion attack in the context of NDN-ABS can be defined as a situation wherein a group of entities with malicious intent pool their attribute sets and generate predicates that match the one generated by the legitimate signer to sign the data and thus use it to wage an attack.

From the privacy point of view, the claim-predicate rule that is used as a basis for the signature goes by the assumption that as long as the claim is satisfied by the said predicate, the Boolean output is an *Accept / True* and does not reveal anything more about the individual signer. Moreover, the signature takes in the tuple which includes the data packet and the predicate along with the pk and the ska. Thus, even if the adversary manages to get access to the signing secret key, the adversary can not cause much havoc since the signature is independent of everything except the message and the predicate. Related work by Maji et. al [19] that discusses the ABS constructs gives detailed security proofs describing the inherent advantages of using the ABS scheme and is applicable for the NDN-ABS design.

6 EVALUATION

Attribute based signature schemes have been discussed in many works earlier. We implemented a Python library [26], including the algorithms defined in Section 4, to evaluate the performance of NDN-ABS in terms of the time it takes to sign and verify (in milliseconds) as well as the signature size. To run the experiments and evaluate the performance of the implemented scheme, as described in Figure 5, we use various platforms running the implemented NDN-ABS library.

6.1 Signature Cost Per Attribute

Figure 5 depicts the results from 10 runs of the implemented library for experiments with measurements averaged over 64 ABS signatures each time on different platforms. The results depict the mean values; we observed that the standard deviation is very small. The results also show the variation of time for the signing and verification operations for various number of attributes. The time for signing and verification grows super linearly (the growth is quadratic when the policy is generated by combining the attributes using the "AND" operation as is the case in our evaluation setup). The growth is quadratic because the multiplication is done for all rows and columns, and MSP grows in both directions with the "AND" operation (refer to Appendix A.2). As shown, the verification process incurs a higher cost compared to the signing process. Figure 5 also highlights the varying signature size with the increase in the number of attributes and the adjusted overhead per signature. Scaling the number of attributes involved in the signature process results in a bigger predicate, which consequently increases the signature size.

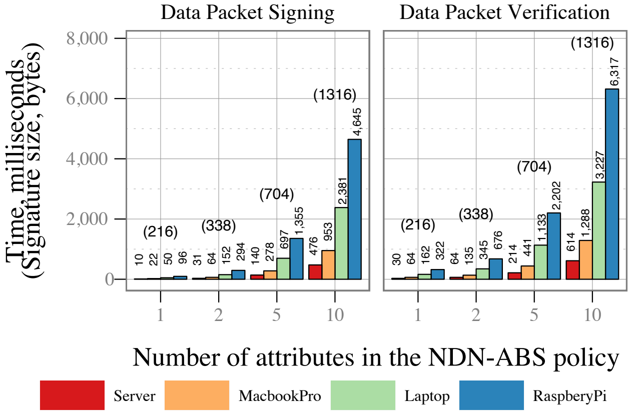

Figure 5: Cost for signing and verification using NDN-ABS
(Server: Intel i7 4.00GHz, 62.8 GB RAM; Macbook Pro: Intel i9 2.9GHz, 32 GB RAM; Laptop: Intel T2300 1.66GHz, 2.4 GB RAM; Raspberry Pi 3: Raspbian, ARM v7 1.4GHz, 0.9 GB RAM)

As can be seen from the results, NDN-ABS signature incurs substantial computational cost, especially on limited resource platforms like the Raspberry Pi 3. In our evaluations (not shown due to space constraints), NDN-ABS signing/verification is at least two orders of magnitude slower than the same operation using RSA. As an indicator, to sign 1 MB data, NDN-ABS takes close to 10 milliseconds while RSA can complete the operation in 0.7 milliseconds when run on the Server. Verification of a similar data size takes about 30 milliseconds with NDN-ABS while it takes only 0.3 milliseconds with RSA. However, with limited policy size (2-3 attributes), aggregated signing (if possible), and future implementation optimizations, we believe NDN-ABS can be a very efficient signature scheme.

6.2 Performance of the Optimized Signing

To optimize the cost for signing and verification, the experiments were run with a test input file of size 10MB. Instead of signing the hash of each packet, hashes of several packets are composed into a manifest–a manifest can have a chosen number of packet hashes. These manifests were signed using the proposed NDN-ABS scheme. Policies generated using varying number of attributes were used in realizing the signatures. For each such generated policies, we ran 10 iterations and observed that the output values were very consistent. Thus, the signer at the time of production can opt for one or many of the following optimization choices (a) a policy that uses required attributes and is not too long, (b) create a manifest with appropriate group size and sign the manifest instead of signing every data packet, (c) hardware acceleration techniques that can provide improved performance. The manifest approach can also be used to amortize the cost. Another approach to reducing the amortized cost will be using a third attribute that encompasses multiple attributes in the policy (e.g., "[attr1-attr2]" instead of "[attr1] AND [attr2]" policy).

Figure 6 shows the mean and confidence intervals for 10 runs of the experiment on a Macbook Pro: Intel i9 2.9GHz, 32 GB RAM platform. The experiments were run with manifests having varying number of implicit digests. It can be observed that the signing and verification times decrease drastically as the group size increases.

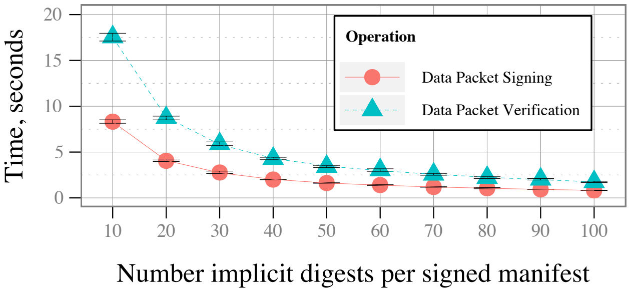

Figure 6: Time for signing and verification of manifests with different group sizes

6.3 Implementation Optimization

To the best of our knowledge, our work is the first comprehensive prototype implementation of the ABS scheme proposed by [19] with a basic adaptation of the ABS scheme from [22] and thus, we do not have any reference to compare with. We thus evaluated the performance of available implementations of attribute-based encryption (ABE) because the underlying computations are similar even though the specific constructs differ.

We compare the existing CP-ABE libraries [2, 11, 30, 31, 35], which have been implemented in various languages showcasing the time taken for *key-generation*, *encryption* and *decryption* operations for scenarios with 10 and 30 attributes. Figure 7 depicts the evaluation results of running the experiments in a standalone system running Ubuntu with an Intel i7 4.00GHz processor and 16 GB RAM.

The results show that *OpenABE* is consistently the most efficient implementation (in C++) across all operations. The common trend shows that the key generation and encryption are the costly operations as opposed to often cheap decryption operations except the *BSWABE (Python)* implementation. We also noticed that the implementation language plays an important role in the efficiency of the system.

Overall, from the results presented above, we observe that the overhead numbers of NDN-ABS depicted in Section 6.1 can be significantly reduced by a more optimized implementation as shown

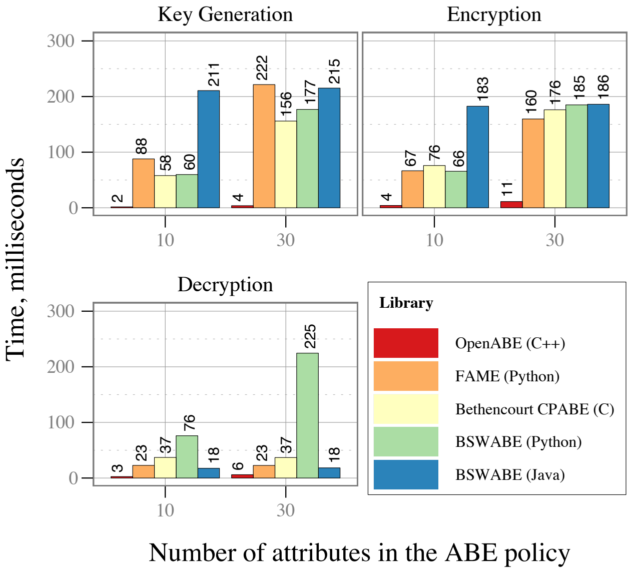

Figure 7: Comparison of the cost for keygen, encryption, and decryption for common ABE libraries

in 6.2. We also expect that incorporating hardware acceleration mechanisms can provide significant boost to the performance and make NDN-ABS signature an even more viable option to be used in production.

7 DISCUSSION

7.1 Multiple Attribute Authorities

Practical applications of using attribute based signatures will involve users receiving attributes from multiple attribute authorities. This also works as a solution for the issue wherein a single attribute authority in the system can be a bottle-neck or a single point of failure when they are compromised. However there may not exist any mutual trust among these attribute authorities and there can be situations where an attribute authority may not be aware of the presence of another.

When working with a system involving multiple attribute authorities, the common challenges are as described below.

(1) How does one ensure that attribute keys for a user received from multiple authorities work correctly.
(2) How does the system or the user manage the many authorities who either move into or away from the system and the state of the attribute keys provided to a user by such authorities.
(3) Cost involved in managing the many keys that will be used by the users and provided by the authorities in the system.
(4) Preventing malicious users from colluding by pooling in attributes and thus making the system robust.

Previous solutions addressing these challenges involved defining a unique *Global ID* GID for every user that is embedded in the keys provided to them [6, 14]. This will prevent the users from colluding with other users and perform any malicious activities in the system. To uphold the privacy and not reveal details about the user from the GID, the system would have a central entity called the *Central Authority (CA)* who is trusted by all users. The CA, on receiving the attribute set and GID from the user, recreates a secret key that the user can use for signing the message. Other extensions of these works also specify the use of a pseudorandom function while generating the secret key that the attribute authority uses to sign the attributes for a user and mutually exchange the seed used by the pseudorandom function [13]. The randomization function will prevent malicious users from recreating the secret key of the attribute authority and thus start issuing attributes on behalf of the authority.

The authors also argue that the use of a central authority eases the management of a system having multiple attribute authorities. However, the CA can be a potential point-of-failure and thus demands for a better and robust solution for such a system. The proposed NDN-ABS scheme can work seamlessly in an environment with multiple authorities without the need for the addition of a third party or external entity.

In specific, the public parameters that are generated can by themselves act as authority "certificates". It can either be trusted as a pre-configuration (based on a trust anchor) or signed by a higher layer authority. Such linkage can be easily realized in the NDN-ABS design using the trust schema as described by the authors in [33]. In other words, trust schema does not have to include public parameters of all authorities as trust anchors, but only the higher level ones. The rest can be automatically taken care of during the automated schema-based validation.

7.2 Revocation strategies in ABS

In this subsection, we discuss the existing ABS revocation techniques. The motivation for introducing revocation in the system is two-fold: (i) revoking the compromised private keys and attributes and (ii) revoking the users' attributes that have been terminated. The compromised key and terminated attributes should be identified, eliminated, and potentially replaced with new credentials. The existing attribute revocation techniques, in general, are classified into three groups; time-based revocation, revocation using a trusted third party, and using revocation lists.

Among all, the most common attribute revocation approach is extending users' attributes with an expiration date. Time-based revocation, in general, requires periodic interactions between the users and the authority for obtaining fresh credential [25]. In [4], the authors proposed a timed re-keying mechanism, in which ciphertexts are generated using users' identities and validity periods. This identity-based encryption method requires the user to possess these two attributes' keys for successful decryption. The authors in [16] also leveraged the addition of an expiry time attribute to design a coarse-grained user-revocation, in which a single attribute revocation causes the user to completely lose the access right.

The proposed approaches in the second class leverage a (semi-) trusted third party for attribute revocation. The main advantage of this technique is its capability in instantaneous attribute revocation compared to lazy revocation of periodic or timed-based mechanisms. In [5], the authors used a mediator as the revocation authority, which executes revocation instructions from the attribute authority. In this scheme, the user's secret key is divided into two parts and kept by the mediator and the signer, allowing the mediator to check the signer's revocation status during the signing phase. In [9, 32], the authority, upon a revocation, sends re-encryption keys to semi-trusted proxies. Proxies, in turn, update the valid users' secret keys, which prevents the revoked users, with the old keys, from successfully decrypting the ciphertext. Similar to these approaches, the authors in [28] used a semi-trusted entity to re-encrypt the ciphertext in a way that only non-revoked user will be able to decrypt it.

In the last revocation class, similar to PKI-based certificate revocation list, the authority collects and publishes revocation lists including revoked attributes and users information. Khader in [12] proposed the integration of a revocation table, allowing the verifiers to identify the revoked signers and attributes during the signature verification process.

Some recent initiatives combined the time-based, proxy-based, and list-based revocation methods, intending to optimize the overhead of revocation list distribution and the period between attribute expiry and its revocation. In particular, the authors in [17] used the combination of revocation lists, which is embedded into the ciphertext for instantaneous revocation, and time validity technique to prevent the expired users from decrypting the ciphertext. In a similar work, the authors in [18] used a semi-trusted revocation authority to store and provide the revocation list to verifiers.

7.3 Challenges for NDN-ABS adoption

Attribute-based systems pose a set of new challenges when compared to the traditional PKI system. The challenges we address or discuss in this paper involving the design of a signature scheme based on attributes for Named Data Networking are

(1) The naming scheme that can be used in the design
(2) A scalable approach with regards to manipulating and updating the policies in cases when the number of attributes is variable with frequent new additions
(3) Use of either a multi-attribute authority system or other solutions to alleviate the issues pertaining to a bottle-neck or single-point-of-failure.
(4) A scalable and robust trust schema which is compatible with the defined NDN architecture and design.

As part of our future work, we plan to explore the use of specific naming conventions and details of trust schema invocation as part of the future work, as it goes well beyond the scope of the current paper.

8 ADDITIONAL USE CASE AND FUTURE WORK

Disaster recovery is another compelling NDN-ABS use case where lack of full connectivity due to link failure results in network fragmentation, which severely impacts information communication across disaster affected zones. In such scenarios, NDN's asynchronous communication model can aid in effective and trustworthy data dissemination. However, the absence of reliable infrastructures and connectivity in disaster scenarios negatively impact the existing signature schemes. The existing NDN-based signature verification approaches require the chain of trust (all the certificates in the trust chain leading to the trust anchor) to be retrieved and verified–a practice that is rarely possible in segmented networks. In contrast to these approaches, NDN-ABS eliminates the need for online certificate retrieval and verification, which makes it an attractive choice in such scenarios. Also, with NDN-ABS, the rescue team members can easily generate signatures by combining pre-issued attributes even in the absence of the authorities. Achieving similar capabilities using the existing approaches such as RSA based schemes either require an always online authority or pre-generation and storage of an exponential number of certificates.

For a better illustration, let us assume a major earthquake has struck city "ABC" leading to the region being isolated from its neighboring regions. For effective disaster recovery, first respondents have to be informed and rescue teams should be formed and coordinated on the fly, which requires unanticipated communication. Moreover, victims may also want to communicate with their families and friends. The communicating entities in this scenario are as follow:

(1) Rescue Command Center (RCC): In the logical sense, the RCC determines the set of access privileges for the other entities. RCC acts as the attribute authority (AA) in the system and provides the attributes and attribute secret keys to the other communicating entities.
(2) Forwarders (FW): The drones and first responders' vehicles that act as data mules and carry the information will receive attributes, such as "[drone-A]", "[model]", and "[Registered-name]" from the RCC. The forwarders play a vital role in the exchange of the Interest and data packets and will be actively involved in the verification of signatures.
(3) Units: The rescue team members (on ground or air) and the deployed sensors that produce and consume data will be assigned with attributes, such as "[ground-team-A]", "[flying-squad-B]", and "[chemical-sensor]". These units constantly communicate with the RCC using the forwarders and intend to either receive or transmit messages pertaining to rescue missions.

In what follows, we explain how leveraging NDN-ABS, as described in Section 4, facilitates trustworthy communication. With the initial setup, the units and the RCC can reliably exchange information pertinent to rescue missions. For signing, the application needs to define a policy predicate, which is a set of attributes combined with "AND" and "OR" operations (e.g., "[flying-squad-B] AND [south]" or "[drone-A] AND ([cam-C] OR [chemical-sensor-A])") and use it along with the attribute signing keys to create a verifiable signature. For verification, the consumer requires the knowledge of the producer's policy, which is a part of the published data packet, as well as the attribute authority's public parameters. Even a new first responder arriving at the disaster site, possessing pre-defined attributes, can publish verifiable information and distress messages using the RCC public parameters and the predicate to verify the signature.

In contrast, when RSA based schemes are employed, often the consumers need to retrieve the responder's certificate along with all the certificate in the trust chain for successful verification, which in addition to the communication also involves significant computation during the verification process. Also, in a disaster struck region, the exchange of redress messages is of top priority even if the identity of the data source is unknown as long as it can be verified to be from a trusted source. We intend to explore the additional challenges of this use case in the future.

9 CONCLUSION

In this paper, we propose and create the first comprehensive prototype implementation of the attribute-based signature scheme. We also integrated this signature scheme to be a part of NDN operations. Our work highlights the benefits of the rich semantics of the NDN naming conventions and how it can be used along with the attributes that an entity can possess and result in successfully exchanging messages. Using a smart-campus, as an example scenario, we explain the design and working of NDN-ABS. The paper also presents a specific mechanism to ensure time-limited validity of the generated signatures. We also evaluated the NDN-ABS signature performance on multiple platforms to identify the limitations and shortcomings. The paper also proposes potential ways to overcome these shortcomings in the production environment.

We evaluate various existing ABE libraries and compare the results to the ABS implementation as the underlying mathematical concepts for both the constructs is similar. We also discuss the usage of multiple attribute authorities and other revocations schemes in the context of NDN-ABS. We observe that attribute based systems provides rich dividends when used in an environment having rich attribute sets with the need for anonymity and unforgeability. Based on the evaluation results, one can notice that the verification process incurs a higher cost compared to the signing process and scaling the number of attributes results in a bigger predicate, which consequently increases the signature size. Hence, attributes that will be involved in the predicate generation have to be wisely chosen during the production phase. More research in this direction is required to well define and harness all the benefits that this scheme has to offer.

10 ACKNOWLEDGEMENTS

This work was supported in part by the US National Science Foundation/Intel grant CNS 1719403, other National Science Foundation grants 1719342, 1345232, 1914635, and EPSCoR Cooperative agreement OIA-1757207. The authors would also like to thank the Graduate and Professional Student Committee and the School of Computing and Information Sciences at Florida International University and the ACM ICN travel grants committee for providing travel support.

A APPENDIX

A.1 Useful Definitions

Definition A.1. **[Bilinear Pairing]** Let $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$ be cyclic (multiplicative) groups of order p, where p is a prime. Let g be a generator of $\mathbb{G}$, and h be a generator of $\mathbb{H}$. Then $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$ is a bilinear pairing if $e(g, h)$ is a generator of $\mathbb{G}_T$, and $e(g^a, h^b) = e(g, h)^{ab}$ for all a, b.

Definition A.2. **[Monotone Span Program] [19]** Let $\Upsilon : \{0, 1\}^n \rightarrow \{0, 1\}$ be a monotone boolean function. A monotone span program for Υ over a field $\mathbb{F}$ is an $l \times t$ matrix M with entries in $\mathbb{F}$, along with a labeling function $a : [l] \rightarrow [n]$ that associates each row of M with an input variable of U, that, for every $(x_1, \ldots, x_n) \in 0, 1^n$, satisfies the following: $\Upsilon(x_1, \ldots, x_n) = 1 \Leftrightarrow \exists\, \vec{v} \in \mathbb{F}^{1\times l} : \vec{v}M = [1, 0, 0, ..., 0]$ and $(\forall i : x_{a(i)} = 0 \Rightarrow v_i = 0)$. That is, $\Upsilon(x_1, \ldots, x_n) = 1$ *iff* the rows of M indexed by $\{i | x_{a(i)} = 1\}$ span the vector $[1, 0, 0, ..., 0]$. Then, l is the length and t the width of the span program, and $l + t$ the size of the span program.

A.2 ABS Construct

ABS.Setup

Choose suitable cyclic groups $\mathbb{G}$ and $\mathbb{H}$ of prime order p, equipped with a bilinear pairing $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$. The Paining group used in the NDN-ABS implementation is "`MNT159`" which represents an asymmetric curve with 159-bit base field. We use "`SHA256`" as the hash function $\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$. Choose value of t_{max} that defines maximum length of the claim predicate. Given the linear growth of the signature size with the length of the claim predicate, a practical value for t_{max} is below 20. Choose random generators: $g \leftarrow \mathbb{G}; \quad h_0, \cdots h_{t_{max}} \leftarrow \mathbb{H}$.

Choose random $a_0, a, b, c \leftarrow \mathbb{Z}_p^*$ and set

$$C = g^c \quad A_0 = h_0^{a_0} \quad A_j = h_j^a \quad B_j = h_j^b \quad (\forall j \in [t_{max}])$$

The master key is $ask = (a_0, a, b)$.

The public key pk is $(g, h_0, \cdots , h_{t_{max}}, A_0, \cdots , A_{t_{max}}, B_1, \cdots , B_{t_{max}}, C)$.

ABS.AttrGen

On input ask as above and attribute set $\mathcal{A} \subseteq \mathbb{A}$ (where $\mathcal{A}$ is a set of strings), use a collision-resistant hash function (we used "`SHA256`") to map attributes to their digests ($SHA256 : \mathcal{A} \rightarrow \widetilde{\mathcal{A}}$). Choose random generator $K_{base} \leftarrow \mathbb{G}$. Set

$$K_0 = K_{base}^{1/a_0} \quad K_u = K_{base}^{1/(a+bu)} \quad (\forall u \in \widetilde{\mathcal{A}})$$

The signing key is then $ska = (K_{base}, K_0, \{K_u | u \in \widetilde{\mathcal{A}}\})$.

ABS.Sign

On input (pk, ska, m, Υ), where m is the message to be signed and Υ is the policy claim predicate First, convert Υ to its corresponding monotone span program $M \in (\mathbb{Z}_p)^{l\times t}$, with row labeling $u : [l] \rightarrow \mathbb{A}$. Also compute the vector $\vec{v}$ that corresponds to the satisfying assignment $\widetilde{\mathcal{A}}$. Compute $\mu = \mathcal{H}(m||\Upsilon)$.

Pick random $r_0 \leftarrow \mathbb{Z}_p^*$ and $r_1, \cdots , r_l \leftarrow \mathbb{Z}_p$ and compute

$$Y = K_{base}^{r_0} \qquad S_i = (K_{u(i)}^{v_i})^{r_0}.(Cg^{\mu})^{r_i} \qquad (\forall i \in [l])$$

$$W = K_0^{r_0} \qquad P_j = \prod_{i=1}^{l} (A_j B_j^{u(i)})^{M_{ij}.r_i} \qquad (\forall j \in [t])$$

The signer does not need to have $K_{u(i)}$ for every attribute $u(i)$ mentioned in the claim predicate, just enough attributes to satisfy the predicate. But when this is the case, $v_i = 0$, and so the value is not needed.

The signature is $\sigma = (Y, W, S_1, \cdots , S_l, P_1, \cdots , P_t)$.

ABS.Ver

On input $(pk, \sigma = (Y, W, S_1, \cdots , S_l, P_1, \cdots , P_t), m, \Upsilon)$, first convert Υ to its corresponding monotone span program $M \in (\mathbb{Z}_p)^{l\times t}$, with row labeling $u : [l] \rightarrow \mathbb{A}$. Compute $\mu = \mathcal{H}(m||\Upsilon)$. If $Y = 1$, then output **reject**. Otherwise check the following constraints:

$$e(W, A_0) \overset{?}{=} e(Y, h_0)$$

$$\prod_{i=1}^{l} e\Big(S_i, (A_j B_j^{u(i)})^{M_{ij}}\Big) \overset{?}{=} \begin{cases} e(Y, h_1)e(Cg^{\mu}, P_1), & j = 1 \\ e(Cg^{\mu}, P_j), & j > 1, \end{cases}$$

for $j \in [t]$. Return **accept** if all the above checks succeed, and **reject** otherwise.

REFERENCES

[1] Alex Afanasyev, Jeff Burke, Tamer Refaei, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2018. A brief introduction to Named Data Networking. In *Proc. of MILCOM.*

[2] Shashank Agrawal. [n.d.]. Ciphertext-Policy Attribute-Based Encryption. Online: https://github.com/sagrawal87/ABE/blob/master/bsw07.py. Last accessed on Aug 22, 2019.

[3] Tohru Asami, Byambajav Namsraijav, Yoshihiko Kawahara, Kohei Sugiyama, Atsushi Tagami, Tomohiko Yagyu, Kenichi Nakamura, and Toru Hasegawa. 2015. Moderator-controlled information sharing by identity-based aggregate signatures for Information Centric Networking. In *Proc. of ACM Conference on Information-Centric Networking.*

[4] A. Boldyreva, V. Goyal, and V. Kumar. 2008. Identity-based encryption with efficient revocation. In *Proc. of ACM Conference on Computer and Communications Security.*

[5] D. Cao, X. Wang, B. Zhao, J. Su, and Q. Hu. 2012. Mediated attribute based signature scheme supporting key revocation. In *Proc. of International Conference on Information Science and Digital Content Technology*, Vol. 2.

[6] Dan Cao, Baokang Zhao, Xiaofeng Wang, and Jinshu Su. 2012. Flexible multi-authority attribute-based signature schemes for expressive policy. *Mobile Information Systems* 8, 3 (2012).

[7] Nikos Fotiou and George C Polyzos. 2016. Securing content sharing over ICN. In *Proc. of ACM Conference on Information-Centric Networking.*

[8] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. 2006. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. of ACM Conference on Computer and Communications Security.*

[9] J. Hur and D. K. Noh. 2011. Attribute-based access control with efficient revocation in data outsourcing systems. *IEEE Transactions on Parallel and Distributed Systems* 22, 7 (2011).

[10] Mihaela Ion, Jianqing Zhang, and Eve M Schooler. 2013. Toward content-centric privacy in ICN: Attribute-based encryption and routing. In *Proc. of ACM SIGCOMM Workshop on Information-Centric Networking.*

[11] Brent Waters John Bethencourt, Amit Sahai. [n.d.]. Ciphertext-Policy Attribute-Based Encryption. Online: http://acsc.cs.utexas.edu/cpabe/tutorial.html. Last accessed on Aug 22, 2019.

[12] D. Khader. 2007. Attribute-Based Group Signature with Revocation. IACR Cryptology ePrint Archive.

[13] Sarmadullah Khan and Rafiullah Khan. 2018. Multiple authorities attribute-based verification mechanism for Blockchain mircogrid transactions. *Energies* 11, 5 (2018).

[14] Allison Lewko and Brent Waters. 2011. Decentralizing attribute-based encryption. In *Proc. of Annual International Conference on the Theory and Applications of Cryptographic Techniques.*

[15] Jin Li, Man Ho Au, Willy Susilo, Dongqing Xie, and Kui Ren. 2010. Attribute-based signature and its applications. In *Proc. of ACM Symposium on Information, Computer, and Communications Security.*

[16] Y. Lian, L. Xu, and X. Huang. 2013. Attribute-based signatures with efficient revocation. In *Proc. of International Conference on Intelligent Networking and Collaborative Systems.*

[17] J. Liu, T. Yuen, P. Zhang, and K. Liang. 2018. Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list. In *Proc. of International Conference on Applied Cryptography and Network Security.*

[18] W. Lueks, G. Alpár, J. Hoepman, and P. Vullers. 2017. Fast revocation of attribute-based credentials for both users and verifiers. *Computers & Security* 67 (2017).

[19] Hemanta K Maji, Manoj Prabhakaran, and Mike Rosulek. 2011. Attribute-based signatures. In *Proc. of Cryptographers' Track at the RSA Conference.*

[20] Adeel Mohammad Malik, Joakim Borgh, and Börje Ohlman. 2016. Attribute-based encryption on a resource constrained sensor in an Information-Centric Network. In *Proc. of ACM Conference on Information-Centric Networking.*

[21] T. Mick, R. Tourani, and S. Misra. 2018. LASeR: Lightweight Authentication and Secured Routing for NDN IoT in Smart Cities. *IEEE Internet of Things Journal* 5, 2 (April 2018).

[22] Mauri Miettinen. [n.d.]. ABS. Online: https://github.com/Mamietti/ABS. Last accessed on Aug 22, 2019.

[23] Ilya Moiseenko. 2014. *Fetching content in Named Data Networking with embedded manifests.* Technical Report NDN-0025. NDN.

[24] NDN Team. [n.d.]. NDN Packet Format Specification. Online: https://named-data.net/doc/NDN-packet-spec/0.3/. Last accessed on Aug 22, 2019.

[25] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. 2010. Secure attribute-based systems. *Journal of Computer Security* 18, 5 (2010).

[26] Sanjeev Kaushik Ramani and Alexander Afanasyev. [n.d.]. Python Implementation of NDN-ABS. Online: https://github.com/sanjeevr93/PyNDNABS. Last accessed on Aug 22, 2019.

[27] Mariana Raykova, Hasnain Lakhani, Hasanat Kazmi, and Ashish Gehani. 2015. Decentralized authorization and privacy-enhanced routing for Information-Centric Networks. In *Proc. of Annual Computer Security Applications Conference.*

[28] A. Sahai, H. Seyalioglu, and B. Waters. 2012. Dynamic credentials and ciphertext delegation for attribute-based encryption. In *Proc. of Annual Cryptology Conference.*

[29] Adi Shamir. 1984. Identity-based cryptosystems and signature schemes. In *Proc. of Workshop on the Theory and Application of Cryptographic Techniques.*

[30] Melissa Chase Shashank Agrawal. [n.d.]. FAME: Fast Attribute-based Message Encryption. Online: https://github.com/sagrawal87/ABE/blob/master/ac17.py. Last accessed on Aug 22, 2019.

[31] Junwei Wang. [n.d.]. Java Realization for Ciphertext-Policy Attribute-Based Encryption. Online: https://github.com/junwei-wang/cpabe/. Last accessed on Aug 22, 2019.

[32] S. Yu, C. Wang, K. Ren, and W. Lou. 2010. Attribute based data sharing with attribute revocation. In *Proc. of ACM Symposium on Information, Computer and Communications Security.*

[33] Yingdi Yu, Alexander Afanasyev, David Clark, Van Jacobson, and Lixia Zhang. 2015. Schematizing trust in Named Data Networking. In *Proc. of ACM Conference on Information-Centric Networking.*

[34] Yingdi Yu, Alexander Afanasyev, Jan Seedorf, Zhiyi Zhang, and Lixia Zhang. 2017. NDN DeLorean: An Authentication System for Data Archives in Named Data Networking. In *Proc. of ACM Conference on Information-Centric Networking.*

[35] LLC Zeutro. [n.d.]. The OpenABE library. Online: https://github.com/zeutro/openabe. Last accessed on Aug 22, 2019.

[36] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.

[37] Zhiyi Zhang, Yingdi Yu, Alex Afanasyev, and Lixia Zhang. 2017. *NDN Certificate Management Protocol (NDNCERT).* Technical Report NDN-0050. NDN.

[38] Zhiyi Zhang, Yingdi Yu, Sanjeev Kaushik Ramani, Alex Afanasyev, and Lixia Zhang. 2018. NAC: Automating access control via Named Data. In *Proc. of MILCOM.*