6-30-2021

Securing the Internet of Things Communication Using Named Data Networking Approaches

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

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

SECURING THE INTERNET OF THINGS COMMUNICATION USING NAMED DATA NETWORKING APPROACHES

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Sanjeev Kaushik Ramani

2021

To: Dean John Volakis
College of Engineering and Computing

This dissertation, written by Sanjeev Kaushik Ramani, and entitled Securing the Internet of Things Communication Using Named Data Networking Approaches, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Deng Pan

Leonardo Bobadilla

Jean H. Andrian

Sitharama S Iyengar, Co-Major Professor

Alexander Afanasyev, Co-Major Professor

Date of Defense: June 30, 2021

The dissertation of Sanjeev Kaushik Ramani is approved.

Dean John Volakis
College of Engineering and Computing

Andres G. Gil
Vice President for Research and Economic Development
and Dean of the University Graduate School

Florida International University, 2021

DEDICATION

I dedicate this dissertation to my entire family and friends for their unwavering support and motivation over the entire course of this journey in my life.

ACKNOWLEDGMENTS

First, I would like to express my heartfelt gratitude to my advisor and mentor, Dr. Alexander Afanasyev, and co-major advisor Dr. S S Iyengar for all the support and guidance extended to me during the completion of this work. Their encouragement, advice, and training have been the guiding force that has inspired and motivated me to conduct ethical and independent research. Without their support, this research would not have been possible.

I would also like to thank and acknowledge Dr. Jean Adrian, Dr. Deng Pan, and Dr. Leonardo Bobadilla for being a part of my dissertation committee and encouraging and guiding me through my research phase, and for all the valuable feedback during my qualifying examination, proposal defense and other stages.

I would like to take this opportunity to acknowledge the various facilities provided by the Knight Foundation School of Computing and Information Sciences that have aided in the completion of this thesis. I would like to extend my gratitude to all the funding agencies that enabled the completion of this thesis. I would also like to thank all my friends, colleagues, collaborators, and mentors for their support, feedback, and encouragement.

Finally, I would like to thank my entire family for supporting me throughout my life and giving me unconditional love and motivation to pursue my dream here in the United States of America.

ABSTRACT OF THE DISSERTATION

SECURING THE INTERNET OF THINGS COMMUNICATION USING NAMED DATA NETWORKING APPROACHES

by

Sanjeev Kaushik Ramani

Florida International University, 2021

Miami, Florida

Professor Alexander Afanasyev, Co-Major Professor

Professor Sitharama S Iyengar, Co-Major Professor

The rapid advancement in sensors and their use in devices has led to the drastic increase of Internet-of-Things (IoT) device applications and usage. A fundamental requirement of an IoT-enabled ecosystem is the device's ability to communicate with other devices, humans, etc. IoT devices are usually highly resource-constrained and come with varying capabilities and features. Hence, a host-based communication approach defined by the TCP/IP architecture relying on securing the communication channel between the hosts displays drawbacks, especially when working in a highly chaotic environment (common with IoT applications). The discrepancies between the requirements of the application and the network supporting the communication demand a fundamental change in securing the communication in IoT applications.

This research along with identifying the fundamental security problems in the IoT device lifecycle in the context of secure communication also explores the use of a data-centric approach advocated by a modern architecture called Named Data Networking (NDN). The use of NDN modifies the basis of communication and security by defining data-centric security where the data chunks are secured directly and retrieved using specialized requests in a pull-based approach. This work also identifies the advantages of using semantically-rich names as the basis for IoT

communication in the current client-driven environment and reinforces it with best practices from the existing host-based approaches for such networks. We present in this thesis several solutions built to automate and securely onboard IoT devices; encryption, decryption, and access control solutions based on semantically rich names and attribute-based schemes. We also provide the design details of solutions to support trustworthy and conditionally private communication among highly resource-constrained devices through specialized signing techniques and automated certificate generation and distribution with minimal use of the network resources. We also explore the design solutions for rapid trust establishment and vertically securing communication in applications including smart-grid operations and vehicular communication along with automated and lightweight certificate generation and management techniques. Through all these design details and exploration, we identify the applicability of the data-centric security techniques presented by NDN in securing IoT communication and address the shortcoming of the existing approaches in this area.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

Chapter 1

INTRODUCTION

1.1 Background

The *Internet of Things (IoT)* vision conceives a connected world with seamless communication between humans and things [RI17a]. This communication and interconnection of the world's "things" is through the use of a common set of networking technologies. However, a major roadblock in the IoT ecosystem is the high degree of heterogeneity among the plethora of sensors that make the building blocks of these devices. Realizing security in this non-standardized environment is a challenge that has to an extent hampered the growth of IoT device use in daily life compared to what was predicted in the last decade [Nor16]. The existing Internet architecture which is the backbone of IoT communication is built as a host-to-host connectivity model which has showcased poor performance in situations involving multiple interfaces, security regimes, mobility, intermittent connectivity, etc.

The IoT vision has over time shifted the focus of communication to being context and content-aware with millions of smart devices coordinating and working towards achieving their application goals. The traits of IoT devices include mobility, intermittent connectivity, variable interfaces, and capabilities based on the applications they are designed for and thus raise concerns on using the current TCP/IP-based communication for such networks. In this work, we analyze techniques that can be employed in secure communication among resource-constrained devices which constitute $> 70\%$ of the IoT population.

To overcome the fundamental challenges in the ad hoc communication scenario posed by IoT systems, we explore solutions based on the approaches of *Information-Centric Networking* (ICN). Named Data Networking (NDN) which was introduced

as an abstraction of networking is a prominent realization of the ICN-based data-centric approach that is advocated as a part of the future internet architecture funded by the US National Science Foundation. The inherent feature of the communication paradigm is a shift from the five-decade-old Internet Protocol (IP) in advocating a client-driven approach to exchange information. NDN succeeds in providing a secure, trustworthy solution to address the root causes of the fundamental challenges with the use of semantically rich names as the driver of the communication without the need for addresses. This allows the application users and developers to directly interact with the entities in the IoT ecosystem and thus eases the deployment, configuration, and data exchange.

In this work, we understand and build on the NDN-advocated communication model for IoT networks and applications with a specific focus on securing communication. Using the data-centric architecture of NDN we can ensure the IoT nodes can operate in an "off-by-default" state until they explicitly either need or like to share information with other nodes in the network thus conserving their power usage. Data-centric security mandated by the architecture adds to ensure that data communicated in the system is secured (authenticated) and when needed encrypted, thus preventing malicious information from entering and compromising the system. It thus makes it possible to create a system with reduced risk of adversarial activity and is successful in eliminating common attacks like Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), etc., which are highly prevalent in IoT networks [SBL+16]. The flexibility provided by the approach aids in the possibility of extending the NDN security model beyond mere networking to securing the software and hardware components of the system. The secured firmware along with secured updates and messages can be ascertained by ensuring that no information communicated within and across the system is unforgeable and the content producer can

not repudiate the creation of the information. The contents of this thesis are thus organized to discuss our exploration of the fundamental processes involved in the IoT device life-cycle and describe the NDN-based solutions developed for onboarding (secure bootstrapping), authenticating information (signing, verification, and certificate management), and confidentiality and access-control.

1.2 Named Data Networking of Things - supporting IoT Applications

Named data networking (NDN) is a prominent realization of Information-Centric Networking and is conceived to be a future internet architecture that eliminates the reliance on host-based information exchange. NDN provides a pull-based approach for information retrieval with the "consumer" of data creating a specialized packet called the "Interest packet" to request for the desired data chunk. The semantically rich name associated with the interest packet is used to forward this request towards a node that either produces the data or even has a copy of the data that it can send as a response. This requested data is packaged in the form of a "Data packet" that has the same name as the interest packet and the requested content. The names used in the interest and data packets can directly be the application level names which solves one of the fundamental issues with TCP/IP based networking where the application names can not be directly used in the network layer.

In addition to the name and content, the data packet also carries a signature field that has the signature of the content producer. This signature plays a vital role in defining the data-centric security primitives of NDN with the data packet being secured directly as against the channels that are used to deliver these packets as is the case with the IPSec [KS05], SSL/TLS [DR08] that are used in the TCP/IP model.

The signatures on the data ensure that any entity involved in the communication can at any time verify the authenticity of the data packet. As an added advantage, the signatures enable the data packet to be temporarily cached in any of the containers without the loss of any integrity leading to the possibility of content delivery even in conditions where connectivity is highly intermittent. Along with security primitives, the specialized data structures used in NDN communication ensures that data can be transmitted and received over any interface that the device is capable of using (WiFi, Bluetooth, LTE, Zigbee, Zwave, etc.).

Inherent security support, the possibility of direct use of application names, data-centric security, in-network caching of data packets, use of any available interface for communication, etc. solves the fundamental challenges that are posed by highly dynamic and ad-hoc networks like IoT networks. In this thesis, we explore these benefits in detail along with the design of enhancements that can enable secure communication in a connected ecosystem built using IoT devices.

1.3 Research Challenges

IoT devices have fundamental variations to the traditional legacy devices that are supported by the current networks. IoT systems introduce highly resource-constrained devices (power and computation) that operate in adverse non-typical environments (being embedded into objects, immersed or buried), have high mobility, and thus intermittent connectivity along with being highly heterogenous, manufactured by different vendors, etc. The devices thus created may also have minimal or non-existent user interfaces, limiting how users may interact/participate with them for various operations including bootstrapping and configuration. IoT ecosystem thus poses two fundamental challenges for the communication architecture supporting it.

The first being the ways and means to provide local and global inter-connectivity among the different devices. The second relates to securing the devices and communication consistently across all messages and updates, storage, configuration, on-boarding, and enabling effective access control techniques in highly intermittent connectivity.

Extending the second challenge, the inherent questions that arise are: "Is it possible for highly diverse devices from varying vendors to communicate locally and globally without prior knowledge of the existence of other devices?". With the basic blocks of IoT devices being sensor and sensor networks, we also explore to see "Can information in the form of updates/messages be shared and received securely through any interface without establishing secure channels?". The other impending question that is to be addressed in this cyber-era with an explosion of information is: "Can we control and manage what information is read and used by which entity and can data be stored, shared, and processed securely in a highly untrustworthy environment?". IoT devices possessing limited power and computing resources make it inevitable for the need to design a system that ensures their efficient utilization while not compromising on the security and trust aspects. In this dissertation thesis, we thus explore solutions to address the above-mentioned challenges by leveraging the data-centric thinking advocated by NDN.

1.4 Research Objectives

This research aims to design security algorithms and solutions that can be applied in conjunction with the NDN-advocated architectural primitives in securing the communication of IoT devices within a network. The research identifies the fundamental challenges in the current scenario that encompasses the entire IoT device lifecycle

within a network from when it is onboarded to the network to when it is in operation within the network. This research also decreases the traffic in the network and focuses on describing specialized techniques that can be used to automate the tasks while improving the performance of the network.

As a part of this dissertation, we will discuss the design, development, and evaluation of novel techniques that are useful in various applications. The dissertation focuses on the following three thrusts:

- **Automated Secure Onboarding / Bootstrapping**

  Onboarding IoT devices into a network is a critical part of the IoT device lifecycle in ensuring that a device joins the correct network and receives the required cryptographic materials to continue secured communication within the network. This process also ensures that the network is aware of the device that is joining and hence can ascertain the legitimacy of the device. The existing techniques to bootstrap such devices are static or involve the use of increased network resources. The objective of the dissertation in this context is the design of dynamic, automated trust establishment in trivial and complex networks with minimal use of network and devices' resources.

- **Authentication Techniques**

  IoT applications built to use the NDN advocated data-centric security model will exchange data packets that are signed by the producer of the content. Repeated use of keys and certificates by the producers in signing the data packets can lead to in the long run leakage of key information if not secured well and thus provide adversaries with an opportunity to wage targeted attacks. The number of exchanges between the consumer and producer or other network entities also increases drastically with every node trying to verify every data packet. In scenarios where the network connectivity is intermittent, depen-

dence on network entities for verification becomes infeasible. The objective of this thrust is to identify techniques that can reduce network dependence and provide a solution that can prevent exposure of the producer while still being able to verify the authenticity of the exchanged data.

- **Data Confidentiality and Access Control techniques**

  Controlling a device or system's access to resources and cryptographic material is of utmost importance. Unintended access, if provided to an adversary can lead to catastrophic outcomes. In this thrust, we focus on identifying techniques that can conform with the principle of least privileges while following all the security requirements of the NDN-based IoT applications.

1.5 Research Contributions

In this dissertation to address the above-mentioned objectives, we have focused on designing solutions that be used in IoT networks designed with NDN capabilities.

- We designed *NDNViber* which is an automated, dynamic bootstrapping technique for onboarding multiple IoT devices (even when they are placed in highly unapproachable positions) using a vibration channel. NDNViber [RPA20] is designed to use a "Commercial Off the Shelf (COTS)" android phone as the controller to bootstrap multiple IoT devices that are equipped with an accelerometer in the presence of a medium that can conduct vibrations. As a part of this work, we also surveyed the possible channels that can be used for bootstrapping devices and identified the security properties, advantages, and disadvantages of each of the options.
- A rapid trust establishment technique [RA20b] using a request-response approach for short-term trusted communication in vehicular networks is devel-

oped. Using this technique, vehicles that have no prior information about a surrounding vehicle can for a short duration exchange information securely enabling applications like collaborative lane changes, vehicular platooning, etc. This work identifies specialized naming conventions and queries that can be used for computing cognitive and normative trust components and thus compute the transient trust values. This work also discusses the security properties of the proposed approach along with the threats and possible solutions to address them.

- An authentication technique that reduces the dependency on the network by using specialized policies and the attribute-based signature scheme was developed in *NDN-ABS* [RTT+19]. The design details are discussed for an illustrative smart-campus environment where conditional privacy can be achieved along with reduced traffic to the network even while ensuring the NDN trust schema can validate the proposed design. We also discuss the performance limitations and techniques that can be used to overcome them in the production environment.
- We designed *CertCoalesce* [RA20c] which is a method that the IoT devices can use to generate, receive and use multiple valid certificates at once and use either of them to sign the content they produce without any overheads. In contrast to the traditional approaches used for certificate request and retrieval, in CertCoalesce, one request from the device can be used by the certificate issuer to provide "infinite" (a large number) of short-term private keys/certificates with limited storage requirements. The design is based on elliptic curve cryptography and thus also ensures forward secrecy.
- We designed a specialized access control scheme using names in [ZYR+18]. This design supports data confidentiality by ensuring that content is encrypted

on production and the keys required to decrypt it are distributed correctly to the specific entities that are authorized to access the information. Hierarchical names are used to ascertain the granularity of access control and key distribution is handled using the NDN enabled interest/data exchanges. This work also explores the use of an Attribute-Based encryption scheme for improved scalability and performance.

- The use of NDN in vertically securing the various operations of a smart grid is described in [RA20a]. Secure communication among the various stakeholders for the operations including power generation, distribution, consumption, billing, and analytics is discussed. The use of data-centric security, data immutability, and opportunistic in-network caching for enhanced grid operation while providing the consumers with better quality of service (QoS), quality of experience (QoE), and most importantly vertical security is discussed in this work.

1.6 Dissertation Outline

The remaining chapters of this dissertation will focus on the various aspects of the IoT device starting with secure device bootstrapping, authentication techniques, data confidentiality, and access control inspired by NDN. In chapter 2, we discuss the existing work and provide the literature review. Chapter 3 highlights the survey conducted on various bootstrapping techniques and the design of *NDNViber* which is an automated bootstrapping technique using a vibration channel. We also explore the need for rapid transient trust and the NDN-enabled design for achieving this in a vehicular environment while identifying the security properties, challenges, and potential solutions. Chapter 4 describes an authentication technique designed

using attribute-based signature schemes to reduce network overhead and reliance for validation while providing conditional anonymity to the content producer. This chapter also describes the design details of *CertCoalesce* which can be used for requesting a pool of valid certificates in a single request along with the various security benefits it provides. Chapter 5 presents a name-based technique for ensuring access control and data confidentiality which can be enhanced by adopting attribute-based encryption techniques. Chapter 6 describes the use of NDN for vertically securing the various components, communications and operations of the various stakeholders in a smart-grid system. Finally, we summarize the work, identify the limitations, and future directions of research and conclude in Chapter 8.

Chapter 2

Related Work

The commonly used techniques in current IoT architectures and frameworks for communication is Bluetooth [SSKR], and ZigBee [All] with their primary focus being device-to-device connectivity. However, each of these protocols is developed as silos preventing their ability to interoperate with each other [SWA+17]. To interface these technologies and the other internet services, current techniques involve the need for a middleware (working as a translator). This drastically prevents the development and innovation of IoT technologies. Thus, the growing demand is for comprehensive frameworks that can integrate and manage different types of devices and communication technologies to provide a simplified and effective user experience.

While current research on IoT security focuses on solving the fundamental issues of IoT systems. These issues arise due to the architectural inefficiencies of the currently used TCP/IP-based internet architecture. The host-based nature of this approach is more than five decades old and incapable of handling the needs of the edge-based and IoT networks that are moving towards context-aware and content-aware computing. The architectural changes advocated by the data-centric ideas of NDN [ZAB+14] solve a good portion of the fundamental issues. We thus explore the techniques and ideas to enhance the working of the NDN-based IoT communication with a specific focus on vertically securing the communication and reducing the burden on the network.

Security in NDN is data-centric and highly reliant on the names and the signature that is bound to the content in the data packets. The earliest work on securing the network elements using a name-based approach was discussed by Smetters and Jacobson in a PARC Technical Report in 2009 [JST+09]. However, a major part of the research effort that followed focussed on exploring ways to provide a robust and

highly flexible architecture to support all network activities. On a parallel front, a series of research articles specific to security were published which identified the performance benefits of the data-centric security in the presence of DoS, DDoS, and Interest flooding and in general extending the security requirements of the architecture [AMM+13, CCGT13, GTUZ13]. We focus on the publications that discuss these security issues as they form a major chunk of the security challenges in the IoT domain.

The initial implementations of an IoT-like system were discussed in the building management system (NDN-BMS) [SDM+14] where the use of distributed network elements was highlighted. The IoT toolkit that followed under the name NDN-IoT [Ban] showcased the practical benefits of the use of NDN in IoT scenarios. A comprehensive account of the design details and specifications in the adoption of NDN for IoT was later provided by Shang et al. [SBL+16] explaining the name-based networking of "things".

Trust bootstrapping is considered is usually defined as an approach that can be used to assign trust scores for devices and services that are new to the network. This is also an integral part of the trust-building stage of any network wherein the nodes either have very limited or no prior interactions. Existing literature related to this aspect considered assigning pre-determined values as a default to any new entity and then update it either by incrementing or decrementing relative to the entities activities and specified criteria as discussed in [BFL96]. Other existing techniques that are identified utilize the reputation values attached to the devices/entities retrieved from other devices that have interacted with that specific device.

The work in NDN-based IoT device bootstrapping is limited. However, bootstrapping is considered a required aspect and is given emphasis even in the recent works on IoT communication. The initial work on NDN-based IoT device boot-

strapping schemes has the following takeaways:

- assume the presence of symmetric keys that are pre-shared for mutual authentication among the device and controller [CCD16];
- embedding a pre-conceived PKI-based private key directly within the device during its manufacturing [LZW+19] and
- scanning static patterns like barcodes or QR codes to initiate the bootstrapping process.

The above-mentioned techniques and assumptions cater well to a large number of IoT applications but will perform poorly in scenarios with restrictions in terms of resource availability, lack of interfaces, difficulty in accessing the installed location, etc. Moreover, the static nature of the approaches adds to the disadvantages when we need to re-bootstrap the devices leaving them unusable when compromised or misconfigured. The highly human-involved approaches make the existing techniques not very user-friendly and augers the need for a dynamic and user-friendly approach with no or minimal requirement for human intervention.

The work described in [BFL96] can be cited as one of the earliest works on trust modeling. A more proactive approach to assigning new entities with reputation scores is discussed in [Swa08] which highlights the reputation development among peers in an environment conducive for collusion. Malik et.al, in [MB09], describe an approach to realizing community-based bootstrapping. The high dependence on community-based approaches and reliance on third-party entities is a major limitation of these approaches. A user should be able to trust a service before its invocation without requiring the existence of a community that evaluated the service in the past. Feldman et.al, in [FC05] describe another community-based approach to trust computation using a probabilistic approach that aggregates probability of the new node to cheat. The main limitation here is the reliance on other entities to reflect

on a new entity joining the node. In this dissertation, we explore the a design for enabling transient trust with vehicular networks as a use-case that can address these shortcomings.

After successful bootstrapping into the network, the devices will have to be able to successfully communicate and authenticate messages it receives from other entities. In most cases, the traditional signature schemes involve the need for obtaining/retrieving multiple NDN certificates following the *chain-of-trust* for any entity to verify the authenticity of the message. The NDN packet specification [NDNb] provides the details of the current signature scheme and the ways to use it. Key management and network overhead are issues identified with the current scheme. The current scheme also does not provide a way to anonymously publish valid information which may be a requirement for certain applications. Group signatures and identity-based approaches have been discussed by researchers as an option to sign anonymously. Attribute-based signatures (ABS) built on the primitives of attribute-based systems is a variation of digital signatures which is applicable use cases where attributes are used. The general properties of ABS are defined by authors in [LAS$^+$10, MPR11]. In ABS-based systems, which is an extension of the identity-based signature scheme, the signing entity (signer) is provided with a set of attributes combinations of which can be used in the signing process and thus provide unique advantages of producer anonymity. Thus, a scheme involving the use of ABS in NDN could leverage the inherent benefits and provide a way to authenticate the signatures used as a part of the data-centric security concept of NDN. The applicability of identity-based and attribute-based encryption schemes in ICN is explored by Tohru et al. [ANK$^+$15], A M Malik et al. [MBO16], Mihaela et al. [IZS13], and others.

Having explored the attribute-based systems, and authenticated the messages sent in the system, the next important part of the life-cycle of the IoT communication is the manage the access of this information among the entities participating in the system. With the importance of names and naming conventions defined in the NDN architecture [ZAB +14], it is imperative of the many advantages of the names. Manipulation of the naming scheme and the flexibility in scaling to ensure granularity are important pieces that define the specifics of access control policies. However, the complexity of IoT systems and the sheer number can lead to scaling the system as a costly affair. This provided the basis for exploring the use of such ABE schemes in designing and enhancing the automated access control techniques. Extensive research on such access control along with integrity and non-repudiation, particularly ABE and ABS schemes, has neither been performed in the ICN community nor has it been demonstrated with deployments in security earlier. In this dissertation, we comprehensively use the best practices from all these works in defining a robust and highly usable communication scheme for IoT devices based on NDN.

Chapter 3

Automated Secure Bootstrapping

The proliferation of sensors and their use in the Internet of Things (IoT) environment has led to a highly connected environment. These inexpensive and connected devices can function efficiently only if they can communicate with the other entities in the network and this is possible only after they can identify the trustworthy networks and entities. The action of pairing such devices securely is the first and foremost task in the IoT lifecycle which ensures that the devices can trust the information exchanged between them. In IoT terminology, this action is called device onboarding/trust bootstrapping and is the first step of our exploration to secure the overall IoT communication. Bootstrapping is usually a highly cumbersome process, especially in resource-constrained and interface-less devices, which may not be accessible even physically after installation. In this chapter, we discuss in brief the various OOB channels and our proposed design of NDNViber that facilitates automated and secure IoT device bootstrapping. Following this in the latter part of this chapter, we discuss a design for the rapid establishment of transient trust in a vehicular application using the cognitive and normative components in a request-response-based approach.

3.1 Overview of out-of-band (OOB) communication channels

The traditional channels that are used for communication are often filled with attackers eagerly plotting ways to compromise the systems. Thus, we explored the possible auxiliary (out-of-band (OOB)) channels that can be used for pairing. The exploration also focussed on the various properties that these auxiliary channels can

offer with a specific focus on the security of the information being communicated. The main motive to use an OOB channel is for exchanging cryptographic information and thus enhance the systems‘ security. Figure 3.1 depicts a subset of the possible modalities in which OOB channels can be used in an IoT-based smart environment with an emphasis on securely bootstrapping these devices. In this section, we survey such techniques and discuss the security properties they provide along with a note on the vulnerabilities they expose.

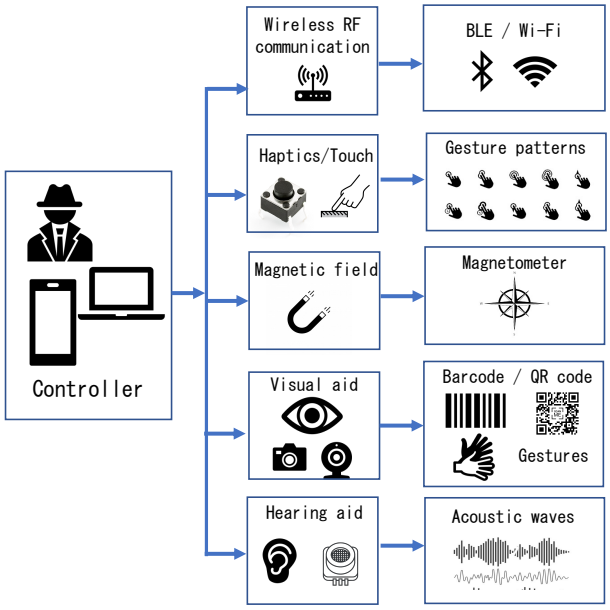

Figure 3.1: Existing OOB communication methods

A brief account of the taxonomy of the OOB channels for secure communication in an IoT environment with examples of applications that use the channel aggregated as a survey is as follows:

Bluetooth Low Energy (BLE) is a common choice for OOB communication. Device Provision Protocol (DPP) [Wi-18] describes a use-case of BLE for secure onboarding of devices. An important vulnerability of using such BLE-based pairing is the large communication range which leads to a possible leak of information to

malicious nodes that are eavesdropping on the channel. Also, the BLE protocol by itself does not provide a proof of possession of bootstrapping keys in the auxiliary channel.

Haptics/Touch Button Enabled Device Association (BEDA) protocol [STU07] describes the typical use of haptics technique with a reliance on physical button press patterns for bootstrapping. This is a very common approach with the patterns translating to the shared secret with usability being a major concern. Visual aids can extend the attack surface with the method having large possibilities of false negatives. Also, the reliance on an interface makes it infeasible in certain IoT applications.

Magnetic field technique Pairing devices using magnetic field values are discussed by Jin et al. [JSZ+15] where smartphones were paired using their magnetometer readings. The magnetometer data, device orientation, and position at that instant of time are recorded and with the addition of ambient noise, a unique correlated message is generated which plays a crucial role in pairing the devices. However, the addition of bulky coils and difficulty in generating stable signals by manipulating the magnetic field reduces their scope in IoT systems.

Visual techniques An account of the visible light-based approach is provided by Kovacevic et al., in [KPČ16]. The main vulnerability of this method of bootstrapping is that a malicious onlooker could view the flashing sequence and inject or interfere with the sequences. Also, the need for specialized light sensors or even cameras in some cases leads to bulky upgrades to devices that are not desired in IoT scenarios.

Audio techniques Modulated sonic frequencies and audio patterns can also be used to perform bootstrapping. Soriente et al. [STU08] discuss a technique where bootstrapping information is exchanged using different codecs that generate audio that is nonsensical to humans. Audio techniques though can be vulnerable to DoS attacks where the attacker can disrupt the communication using noise that interferes and modifies the encoded audio. Another commonly used acoustic technique is the use of ultrasonic frequencies as described by Mayrhofer and Gellersen in [MG07]. Ultrasonic approaches however need a highly controlled environment for effective pairing and can easily be tampered with by third-party devices or physical obstacles in the vicinity.

Vibration techniques Prior attempts to use vibration as a mode of communication for security purposes can be seen in [AS16, KLR+15, SUVA11, DLVZH09, KFR09]. Lee et al. [LRRK18] describe an approach to enhancing the communication rate when using a vibration channel to pair devices. These articles discuss the benefits of the use of vibration for secure communication and its role in alleviating the threats of other techniques. Controlled vibrations can be effectively used if the devices are very close to each other thus creating a reduced attack surface. However, there are trade-offs related to the data rate and the bit errors that can occur due to lack of synchronization[1].

[1]A more detailed account of all the OOB channels along with their merits and demerits for IoT-bootstrapping is discussed in the paper titled "NDNViber" which is published as a part of the IEEE ICC - ICN-SRA Workshop in 2020

3.2 NDNViber: Automated bootstrapping of IoT Devices using vibration based auxiliary channel

Based on the outcomes of the above study, we identified that modulated vibrations can provide a robust and dynamic approach to bootstrapping even highly resource constrained devices which are devoid of interfaces or direct external contact. Thus, we designed the NDNViber approach (Figure 3.2 which uses the semantically rich NDN naming and a specialized encoding method, to provide a solution that uses a vibration-based OOB channel for securely bootstrapping IoT devices. Our prototype implementation involves a commodity (COTS) smartphone as the controller that can bootstrap many small IoT devices that possess accelerometer sensors. The analysis reveals the following advantages of NDNViber:

- real-time generation of initial secret eliminating the need for embedding private-keys or certificates at the time of device manufacturing;
- requirement of the physical proximity of the devices and controller in the order of < 1.5 centimeters [2] thus reducing the attack surface significantly;
- ability to bootstrap multiple devices simultaneously (with the availability of appropriate medium);
- ability to bootstrap devices deployed in inaccessible locations that are hard to reach (e.g., behind a wall along the pipes sensing for any leaks, etc.);
- ease of use of commodity Android phones as controllers without requiring any additional hardware (i.e., via programing using the built-in vibration motors); and

[2]Increasing distance, lowers the intensity of perceived vibrations and thus higher is the probability of erroneous reception.

- no additional (when already built-in) or a meager cost for accelerometers[3], is the only required component on the IoT device side.

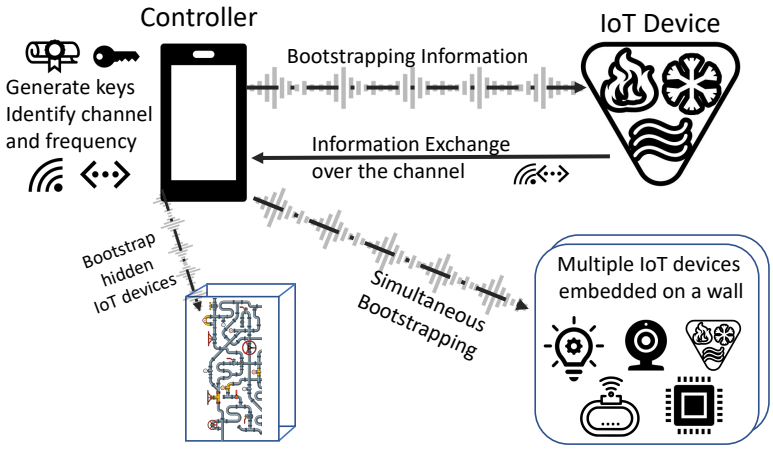

Figure 3.2: NDNViber based secure IoT device bootstrapping

3.2.1 Existing Bootstrapping techniques in ICN/NDN

Initial work on NDN-based bootstrapping schemes either (a) assumes the existence of a pre-shared symmetric key between the device and the controller which is used to achieve initial mutual authentication [CCD16]; or (b) expect the private key of the PKI-based approach to be embedded and installed in the device when it is being manufactured [LZW+19] and the controller scanning either a QR code or other static patterns to initiate the onboarding. While these are extensively being used, the above-mentioned assumptions become a major limitation in smart systems used in futuristic homes. Consider the example of a smart thermostat which includes hundreds of tiny temperature sensors and actuators embedded along the walls of the house, air ducts, windows, and doors. One of the properties of such a thermostat is the fact that it may not have any user interface and be completely inaccessible after

[3]Typical accelerometer sensors cost less than 1 USD

the installation, yet requiring bootstrapping and, potentially, re-bootstrapping (e.g., after upgrading the controller or selling the house). In these cases, it is necessary to have a dynamic and automated approach for bootstrapping.

The goal of NDNViber-based bootstrapping is thus to provide these smart but highly resource-constrained devices with information about:

- which WiFi/BlueTooth/ZigBee network they should be connected to and what are the network credentials;
- the *trust anchor* of the system (a cryptographic certificate of the trusted controller) and any associated trust schemas [YAC+15c];
- the *namespace* under which they can publish data; and
- obtaining a *certificate* so the data created by them can be properly authenticated in the network.

3.2.2 Bootstrapping using NDNViber

The complete NDNViber bootstrapping technique includes four stages (Figure 3.3, three of which include communication over the vibration channel: *pilot sequence* (vibro), *trigger* (hybrid), *anchor* (WiFi/Bluetooth/ZigBee), and *ndncert* (hybrid) exchanges.

The technique is designed to have a *pilot sequence* that can target all devices within the vibro range while acting as a (re-)activation mechanism for the IoT device bootstrapping by waking up the device to actively observe the vibration channel and attempt to decode the modulated messages. After being informed to listen over the channel, the controller initiates the *trigger exchange* constituting an NDN Interest targeting all devices over the vibro channel. This interest includes environment identity information (e.g., the namespace used for the smart house), auxiliary in-

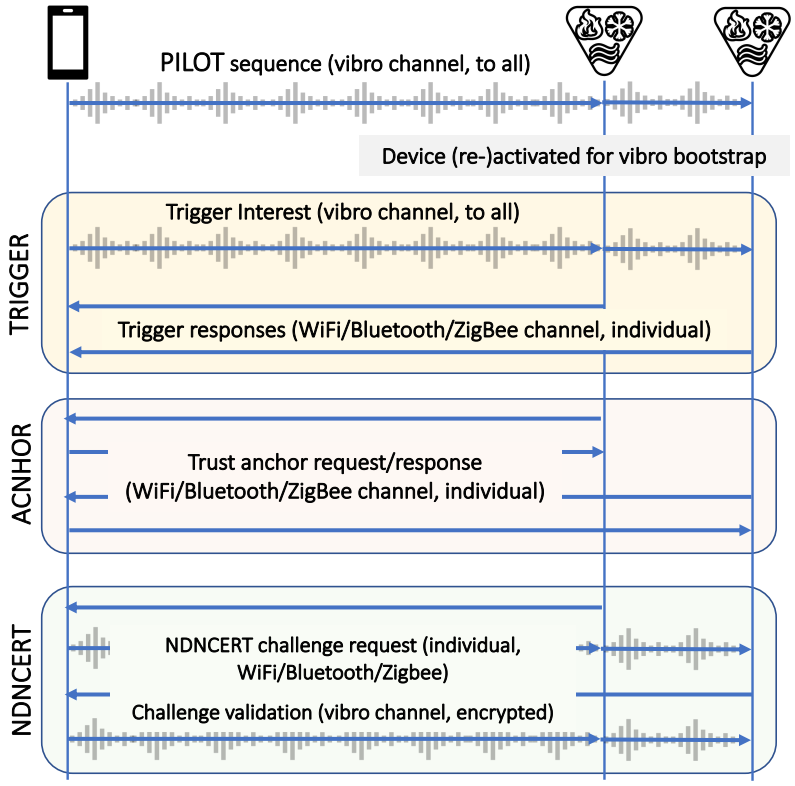

Figure 3.3: NDNViber Bootstrapping Overview

formation (including temporary encryption key), and necessary information for the device to connect to the desired WiFi, Bluetooth, or ZigBee network along with the request for the device identifier. On successful decoding, the IoT devices can connect to the target network and individually respond with the unique device information, including their serial numbers, temporary encryption keys, etc. The *anchor exchange* which follows is realized entirely over the primary networking channel using the information and temporary encryption keys mutually obtained from the trigger exchange. The purpose of the anchor exchanges, initiated by the devices, is to obtain the public key and certificate of the network, i.e., to ensure the device can successfully be authenticated for any future exchanges. Finally, the *NDNCERT* exchanges, again initiated by the individual IoT devices, run the NDNCERT proto-

col [ZYAZ17a, ndna] to retrieve the assigned namespace for the device, generate the corresponding private key, and obtain the NDN certificate for this key/namespace.

While most of the protocol exchanges are realized over the traditional channels (not fully shown in the illustration in Figure 3.3), the key security part: security challenges to ensure vibro-proximity of the device, is done using vibrations. Even though the vibration channel will reach all devices in the range, each response is unique to the challenge-requesting device and is properly encrypted with device-specific keys making it a highly robust approach.

Naming Scheme

To leverage the name advantages of an NDN system, NDNViber directly uses NDN names (more specifically, Interest packets for the named data) as a trigger sequence to initiate (re-)bootstrapping ("TRIGGER"), control sequences to send the system's trust anchor ("ANCHOR") and initiate NDNCERT vibro-challenge ("NDNCERT"). The generalized naming convention NDNViber follows is "/ndnViber/[sequence-type]/[device-name]/[params...]". The details of the components are:

- "/ndnViber" is the prefix name that the device and controller used to identify the communication to be a part of the NDNViber bootstrapping protocol.
- "[sequence-type]" identifies the sequence in progress to being either TRIGGER, ANCHOR, or NDNCERT. This is very important for the exchange of appropriate information especially when we use this technique to bootstrap multiple devices simultaneously.
- "[device-name]" denotes the unique name for the device in the operating environment. Examples are "FIU-PG6/142/duct-temp001", "AA-house/washer-temp002" etc. The granularity can vary based on the number of devices deployed in the vicinity. We can thus have "CPW-SR/kitchen", "FIU-PG6/MeritLab/142" etc.

- "[params...]" here represent the other parameters including the nonces, timestamps, information regarding channels for communication, etc.

Vibration Coding Scheme

It is common for IoT devices and other sensors to go into an idle state when not sensing to save power and the pilot sequence triggers their monitoring of the channel. The pilot sequence initiating the NDNViber approach includes vibrations from the controller for a duration of 250 ms followed by an idle state of 25 ms (these values can be altered based on the sensitivity of the devices and the sensors). NDNViber trigger sequence follows the pilot. The information exchange is performed using a variation of the on-off keying (OOK) technique with the duration being altered instead of the amplitude[4].

Table 3.1: Vibration durations Mapping table

Quad number	00	01	10	11
Decimal Equivalent	0	1	2	3
Vibration Duration (ms)	50	60	70	80

A common trend among most IoT applications is to use a mobile phone as the single controller to manage all the devices and make them highly usable. Thus, our initial version of NDNViber is designed to use a commodity smartphone running Android OS as the controller. Android provides the option to control the vibration motors [vib] while designing applications. In our design, the vibration channel is used only for transmitting the information from the controller to the device(s). The bits of each octet of the packet's wire encoding are grouped and independently converted to a vibration duration value using a simple lookup table as shown in Table 3.1. The mapped durations as seen, start from 50ms because general commodity

[4]OOK is one of the forms of amplitude-shift keying (ASK) modulation used to represent data based on the presence or absence of a carrier

android phones are assembled with vibration motors from different vendors and are thus not precise for vibrations $\leq$ 50ms. To ease the identification of the vibration durations accurately, between subsequent vibrations, an idle period of 20ms is introduced. This idle period helps in (a) determining the end of vibration and (b) providing the receiver enough time to process the received vibrations.

At the receiver end, the device measures the presence of an acceleration value between two idle periods and records them. Acceleration values $\leq 0.1\ m/s^2$ is considered part of the idle period and not a part of the encoded information. The recorded vibrations are then rounded off to the nearest integer value using the "round" function. The mapping table is subsequently used to identify the corresponding equivalence of the decoded value. On completion of the transmission, the device now has the entire information decoded. Figure 3.4 depicts the vibration pattern as detected by the accelerometer associated with the device showing vibrations that are a part of the encoded information and the introduced idle time. The vibration durations in the figure decode to a value of "1032...".

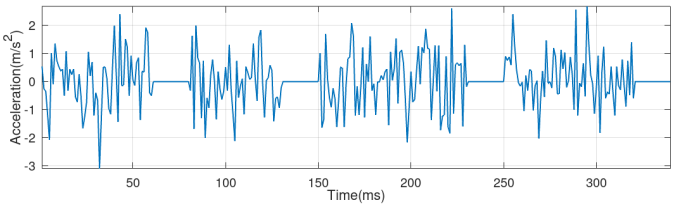

Figure 3.4: NDNViber Encoded Information as detected by the accelerometer associated with the receiver

3.2.3 Evaluation and Discussion

Design Considerations

The use of the duration of vibrations instead of the amplitude of vibration presents the following advantages:

1. All android phones can be used as a controller since the duration is programmable in all versions of android whereas the amplitude can be programmed only in phones running newer versions of android.
2. The transmission error reduces significantly. A carefully selected value for the idle period based on the sensi- tivity of the available controller and device can provide optimal results.
3. The computational load on the resource-constrained de- vice decreases as processing is limited to the round-off function.
4. Time synchronization does not become too critical after the pilot sequence is identified.

Security Properties Analysis

The use of vibrations as a mode for communication ensures that the controller and the device are in very close physical proximity of ¡ 1.5cm. Increasing this distance leads to a drop in the accuracy of reception which leads to the exchange of corrupted information. The drop in received signal power as the controller and device move apart in the absence of a medium (like wood) is exponential. The proximity thus ensures that the probability of an attacker intercepting messages or making any attempts to tamper with the messages is very low.

An attacker can attempt to introduce random and rogue vibration signals to disrupt the system and wage a Denial of Service (DoS) attack. However, the use of

specific naming conventions alleviates the impact of such rogue vibrations as they will not correspond to valid interest or data. The data-centric security leveraged using NDN provides added security. The nonces and the timestamp that are exchanged with the messages also ensure the freshness and authenticity of the devices and controller while also ensuring that replay attacks are thwarted. Every time a device is to be (re-)bootstrapped, NDNViber being a dynamic approach, uses a new set of message exchanges ensuring the completed requests invalid.

3.2.4 Performance Evaluation

The limited use of vibratory channels for communication and bootstrapping is because of the observed low throughput which is an outcome of (a) commodity smartphones (controller in our scenario) use the vibration motors for providing user notifications and thus response times are not considered too seriously; (b) android phones have different vibration motors and thus the accuracy in terms of the duration of vibration are not precise for values $\leq$ 50ms.

The target devices we consider have low computational capabilities, no interfaces, etc. NDNViber thus employs a simple transformation of the data into vibration durations for encoding data. The time taken to transfer a byte using the encoding scheme described including the 20ms idle period ranges between 260ms to 380ms.

Even though this technique is slower than the other OOB methods, our initial experiments yielded an error ratio[5] of the order of 10^{-9}. Figure 3.5 depicts the vibratory signals sent by the controller (in blue) and vibrations sensed (in red) by the accelerometer in the device. The variations in the duration are because of the

[5]We expect the group bootstrapping to be more erroneous because of the impact the medium has on the communication and is a part of our future work

possible addition of noise in the channel. These received vibrations are rounded off thus eliminating the errors and aiding the correct decoding of the intended message.

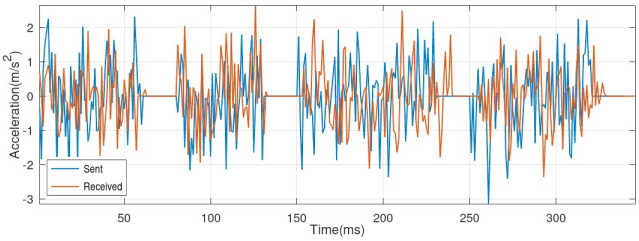

Figure 3.5: Comparison of sent and received vibratory information

An important advantage of using NDNViber is the possibility of being able to bootstrap multiple devices simultaneously. The requirement is the availability of a conducive medium that can transmit the vibrations generated by the controller to the target devices. There are inherent challenges that this brings up like (a) interference among devices; (b) induced passive vibration by the medium affecting the transmitted vibrations; (c) minor degree of acoustic leakage and attenuation; (d) orientation induced errors, etc. However, the use of a controlled environment allows the controller to successfully bootstrap the devices. We intend to explore this extensively in the future.

In the following section, we will discuss a transient trust establishment approach to initiate communication among multiple devices that have very limited or no-prior interaction but will be in contact only for a very short duration for employing the long process involved in building traditional trust values.

3.3 Transient Trust Bootstrapping for NDN-Based Vehicular Networks

Secure communication among entities necessitates the establishment of trust. The frequency of contact and the duration of pairing determine the need for either the traditional computation of trust scores or a transient value that can initiate the communication. Vehicular networks are a specialized IoT application where the communication duration among vehicles is usually very short and the frequency of such communication among the same entities is very low. This to a large extent limits the applicability of the traditional long-standing trust management techniques.

NDN based on the data-centric architecture provides a highly conducive approach to design vehicular networks. The mobility support and request-response type model of communication can be used to design a transient trust establishment scheme that can provide a short-term trust value that can be used for various collaborative applications among the vehicles. This section highlights the use of ideas inspired by the *Swift Trust* model and explores its use in a vehicular communication application design based on NDN. With the proposed design, vehicles in the communication range can quickly make short-term trust decisions for secure publishing, consumption, and processing of data (e.g., to cooperatively analyze the nearby environment for potential safety issues). The proposed design employs a task-oriented method of establishing trust based on request-response communication.

Trust among the communicating entities effectively determines the services or applications that the entities are willing to accept or provide information to. At a high level, trust can broadly be categorized under static/knowledge-based or dynamic/interaction-based trust. The static/knowledge-based trust expects the communicating entities to possess complete or partial knowledge about the other en-

tity, e.g., gained based on prior encounters or from trusted third parties. Dynamic/interaction-based trust involves entities willing to collaborate on a common task without any prior interactions or the involvement of trusted third parties.

In traditional approaches to trust establishment, the communicating parties can rely on pre-existing configurations— pre-configured sets of root Certification Authority (CA) certificates and transitive trust in Public Key Infrastructure (PKI) model—or dynamically build trust relations—using feedback or explicitly setting trust decisions of certificate trustworthiness in Web-of-Trust (WoT) model. In a highly dynamic environment like vehicular networking, these traditional methods may not work. PKI and WoT usually require connectivity to infrastructure which may not be feasible to maintain due to the mobility patterns of the vehicles. The trust relationships thus defined are usually long-standing and not applicable to most vehicular network applications. Existing literature on VANETs [RP09] suggests that in 97% of cases, two vehicles come in communication proximity for less than 10 seconds. In such scenarios, message dissemination by applications and services is highly time-critical and thus needs trust computation on the fly with minimum involvement of external factors.

In this chapter, we explore the applicability of social trust concepts of Swift Trust [MRL13] in designing a transient trust model that can be used in the rapid bootstrapping of trust among vehicles[6]. The problems we aim to solve using this design are

- a technique for entrusting another vehicle (or entity) into performing a task without any prior interactions with it.

[6]The contents of this chapter has been published in IEEE ICC Workshop (ICN-SRA) [RA20b]

- in a vehicular environment, perform complex collaborative tasks like a lane change maneuver with the assistance of the surrounding vehicles
- identify the benefits of using the asynchronous communication model of NDN in successfully passing messages even in the absence of infrastructure support.

The data-centric communication model with the built-in data-centric security primitives offered by the NDN [ZAB+14] architecture provides unique advantages for a large class of inter-vehicular communication scenarios [GPW+13]. In this section, we will be focusing on the opportunity the vehicles are provided with to use any of the available communication interfaces (e.g., WiFi or Bluetooth) to transmit and receive data along with the support for in-network caching that enables robust communication even in intermittent connectivity. Flexible naming eliminates the dependency on mapping systems like DNS, allowing applications to use the network in a semantically meaningful way and using the application names directly for communication in the network. However, the application of NDN still requires a proper bootstrapping of trust to ensure secure production, consumption, and processing of data, which is the research objective we accomplish in this work. Our contributions are thus threefold.

- This is the first attempt to use a design inspired by the swift trust model in computing transient trust scores in a vehicular environment.
- We integrated the benefits of data-centric communication using NDN in the dissemination of important messages.
- We defined a specific mechanism to ensure time-limited validity of the trust scores and possible use of ledgers in cases where trust provenance is desired.

3.3.1 Motivation

Maneuvering through traffic and lane changes is a common sight among moving vehicles. Currently, such operations are totally dependent on the decisions made by the person operating the vehicle. The decisions are made based on the traffic conditions, assessment of the speeds of the surrounding vehicles (moving in the direction of oncoming traffic), proper indications to alert the surrounding vehicles and pedestrians regarding the lane change, etc. The number of parameters involved in the decision-making showcases the important role that the human intellect plays in this context. Even with the human operator, there is a need for a transient trust in order to act as we are still dependent on the actions and indications provided by the surrounding vehicles the operator of which we have to trust.

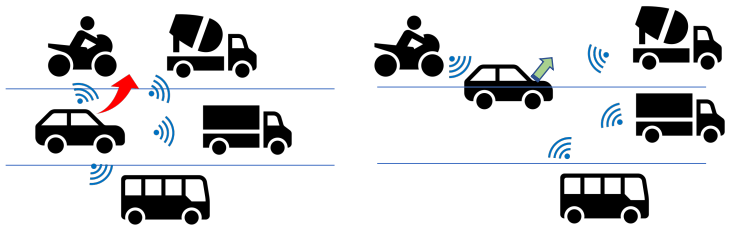

Figure 3.6: Collaborative lane changes

Futuristic vehicles are being designed to operate and make decisions autonomously without any human intervention. This setup will need multiple interactions among the vehicles to successfully perform a complex operation like a lane change maneuver which involves multiple parties. The onus for trust computation and usage shifts to the vehicles from the operators and these operations have to be completed in a very short duration. Any miscommunication or false messages/actions in such a situation can have catastrophic outcomes. The trust values thus computed for a certain vehicle or group of vehicles may not be useful after the completion of the action and thus is highly transient.

We will use this lane change maneuvering as shown in Figure 3.6 as an example to explain our design. This chapter describes our proposal of a model to compute rapid transient trust based on task-oriented concepts.

3.3.2 Background on Swift Trust

The earliest work on introducing Swift trust was introduced by Meyerson [RDH09] to explain the trust paradox in temporary groups. The groups under discussion involve individuals who did not have any prior interaction but need to collaborate to accomplish a common objective. Any temporary team has several common traits, including:

- limited or no previous collaborations. In most cases, the entities may not work again together after the specific goal is achieved
- final goal (is usually very complex to achieve individually), requires entities with varied skillsets
- presence of tight deadlines for meeting the goals and objectives.

Swift trust has two components (a) cognitive and (b) normative (i.e., ideal or standard) components. The cognitive components of swift trust depend on the aggregated opinions of the communicating group about traits that are obvious. These traits could be due to the social identities the entities possess or even self-categorizations. Minimal or no prior interactions make this component of trust computation highly critical as it leads to fostering the initial trusting behavior. The normative components define a set of norms/guidelines that has to be met to enhance the early trust behaviors to be more prominent and less erosive. Setting mutually agreeable norms and meeting them improves the trusting beliefs that one entity has on the other.

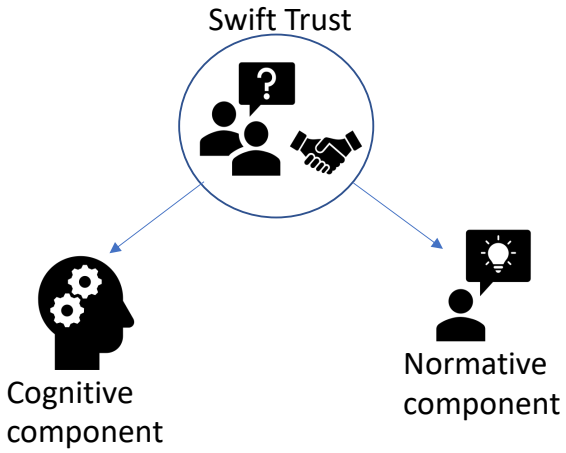

Figure 3.7: Swift Trust model

Swift trust is thus a type of subjective trust model [RDH09] where every node computes trust values of the neighbors/service provider based on its interaction with them. It is different from the objective model where reputation values propagate in a network and entities rely on various forms of transitive trust rules. A basic model of Swift trust designed on the basis of Harwood's Swift Trust Model [Har12] is as depicted in Figure 3.7.

For the scenario depicted in Figur 3.6 to be successfully accomplished, the model is adopted such that the car sends out a signal to collaborate in the lane change. The surrounding vehicles that receive this signal will aggregate the responses to calculate the cognitive trust component. On receiving a response to the request, the car will then provide tasks to the potential collaborators. The tasks are to be strategically designed to be able to receive responses that are within certain guidelines. The responses from surrounding vehicles for this task and the deviation from the defined norms will determine the normative component for computing the trust. The trust scores are the weighted sum of the cognitive and normative components. Once the car has recorded a trust score above a defined threshold from all the surrounding collaborators, the car will commit to doing the lane change.

The detailed design to achieve short-term and rapid transient trust using specialized interest and data exchanges for NDN-based vehicular communication systems is discussed in the following sections of this chapter. This design can be adapted for any situation where such transient trust is required to bootstrap devices and ensure that there can be a trusted and secure long-term communication with the device in the future. According to the design, the communicating entities can at any point in time attempt to re-verify the integrity of the results or even trigger the entire process again to ensure that the communicating devices do not turn rogue in the future. Such a random re-verification can be used in times of suspicion and ensures that the devices in the network abide by the trust principles agreed upon and do not turn malicious after a time interval and is supported by the famous *prisoner's dilemma* concept defined in game-theoretic approaches.

3.3.3 Design Details

In the trust computation, the design focuses on the calculation of both the cognitive and normative components. In our application scenario, we expect the vehicles to be equipped with all the standard recommendations identified for a smart vehicle that is capable of communicating autonomously with its surroundings in the form of V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure) communication modes. The Society of Automotive Engineers (SAE) also recommends that all vehicles periodically (at a frequency of a signal every 0.1 seconds) send out a heartbeat-like Basic Safety Message (BSM) so that the surrounding vehicles can learn of its presence. The BSM has a specialized payload that includes information related to the current state of the vehicle including details of the velocity, acceleration, brake status, and steering angle.

Let us also consider a road transport route to have n lanes with m segments in each lane. We assume vehicle A to be traveling in one such lane. Consider an instance of a time when In any specific area in the lane (i, j) there are K vehicles. The area is defined as $1 < I < N$, $1 < j < M$. Any vehicle k in the lane can receive a message from the preceding or succeeding vehicles, say l where $(l = 1, \cdots, k - 1) and 1 < k < K$.

Each communicating party, (a vehicle) assumes the role of a possible trusted entity with the desire to maintain a good reputation among other vehicles. While occupying this role, they make a promise to the requesting entity to accomplish a task at hand without any malicious intent.

As a part of this mechanism, each vehicle sends out an interest requesting the BSM data packets from its neighbors. The surrounding vehicles on receiving the interest will reply with their BSM data which will contain details about its current state. The car that receives and aggregates these messages, can also verify them as it can compute the relative velocity, acceleration, and other common surrounding parameters and compare them with the received response. The proximity to the correct value will provide a higher cognitive trust score.

The car that desires to perform a lane-change maneuver will formulate some tasks based on the surroundings and the states of the vehicles it received responses from. The car also defines the guidelines for performing the tasks. The car then sends out an *Interest packet* requesting possible collaborators. The vehicles that are willing to collaborate and receive the interest respond with a data packet accepting a possible collaboration. The car then poses the formulated questions to the vehicle. The responses are gathered and the normative scores are computed.

The cognitive component of the trust score is computed using the following equation.

$$T_{cognitive} = (T_A + T_B + T_I)/S \tag{3.1}$$

where

- T_A depicts *Ability based trust* which is derived from either the sensor calculations, the manufacturers' certificate, or other means that highlight the capability of the vehicle.
- T_B depicts *Benevolence based trust* which is based on how prompt and precise the responses are to the questions posed by the requester.
- T_I depicts *Integrity based trust* which is based on the manner in which the remote vehicle handles the interest messages, aggregation of the requests and responses and the prompt retransmission of the time-stamps or nonces.
- S denotes the *self-orientation* referring to the current focus of the car and what is its expectation as a response to the sent interest.

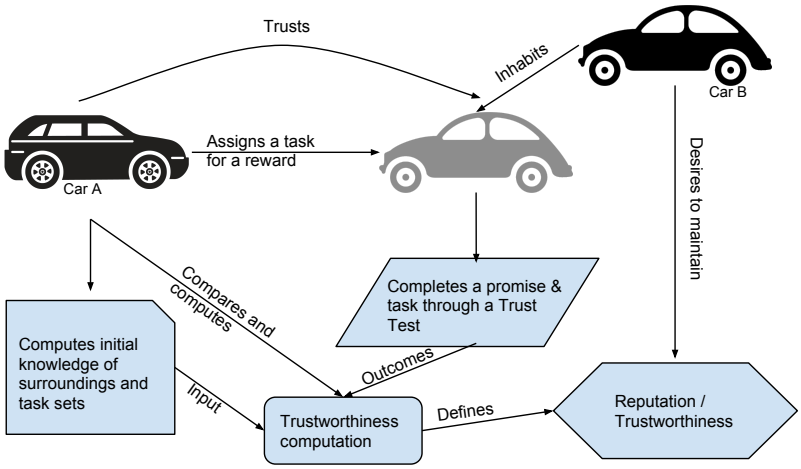

Figure 3.8: Design details

The computation of the normative component of trust involves the risk that the other vehicles are willing to take to accomplish the provided task. The performance of the task as per the set guidelines also plays an important role. The computation

of the normative component is based on the game-theoretic approach involving an incentive to lure the surrounding vehicles to collaborate. A special case of *prisoner's dilemma* involving a donation game is used as the approach here. According to this game, if a vehicle cooperates by performing the task, it is offering the requesting vehicle with a benefit b which is the outcome of the task at a personal computation cost c with b > c. If the vehicle declines to perform the task, it offering nothing.

If T is the temptation to perform the task, R is the reward on completion, P the punishment for not meeting the outcome and S refers to no-loss or no-gain, then based on game-theory, the collaborator will collaborate only if $T > R > P > S$. Once the vehicle is convinced to collaborate and perform the task, the responses received lead to the computation of the normative trust component and is computed using the following formula:

$$T_{normative} = \sum_{i=1}^{k} T_b(i)(\widehat{w}_i) \tag{3.2}$$

where k is the total number of tasks provided to the vehicle b. T_b represents the normalized value of the responses provided by b such that $0 < T_b < 1$. Weights for each task are assigned by the developer depending on the complexity and computation involved in the task. It is to be noted that the value of $T_{normative}$ lies in the range $0 < T_{normative} < 1$.

Once the cognitive and normative components have been computed, the overall transient trust score is given by

$$T_{transient} = T_{cognitive} \circledast T_{normative} \tag{3.3}$$

3.3.4 Simulation Scenario

In this section, we will explain the use of specialized NDN names for fulfilling the communication requirements among the entities while describing possible vehicular networking applications and network messages to compute transient trust as described in the design.

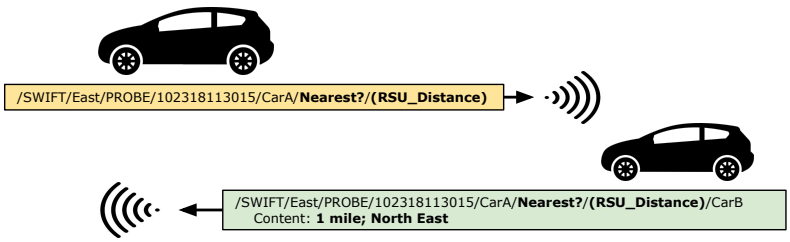

Figure 3.9: Spatio-temporal task: Identifying nearest RSU

As an example for a formulated normative task, we consider a spatio-temporal question requesting the distance of the nearest Road-Side Unit (RSU). Figure 3.9 shows the various exchanges between Car A and Car B. Car A sends out an interest packet with the Probing request asking for the distance of the nearest RSU. The interest packet has various components highlighting the application pertaining to which the message is being transmitted, the direction and the timestamp at which the packet is transmitted, and finally the request shown by "`/Nearest?`". Any car that receives this interest packet and is willing to participate can reply with the appropriate content. Car B, which has received this interest packet, replies with the content that states that the nearest RSU is at a distance of 1 mile in the North Eastern Direction. Based on the guidelines set, Car A computes the value of T_b for Car B.

Similarly, Figure 3.10 depicts another complex task that can be used by the car to compute the normative component of swift trust. Car A in this instance sends out an interest packet requesting the surrounding vehicles to perform a pattern/image

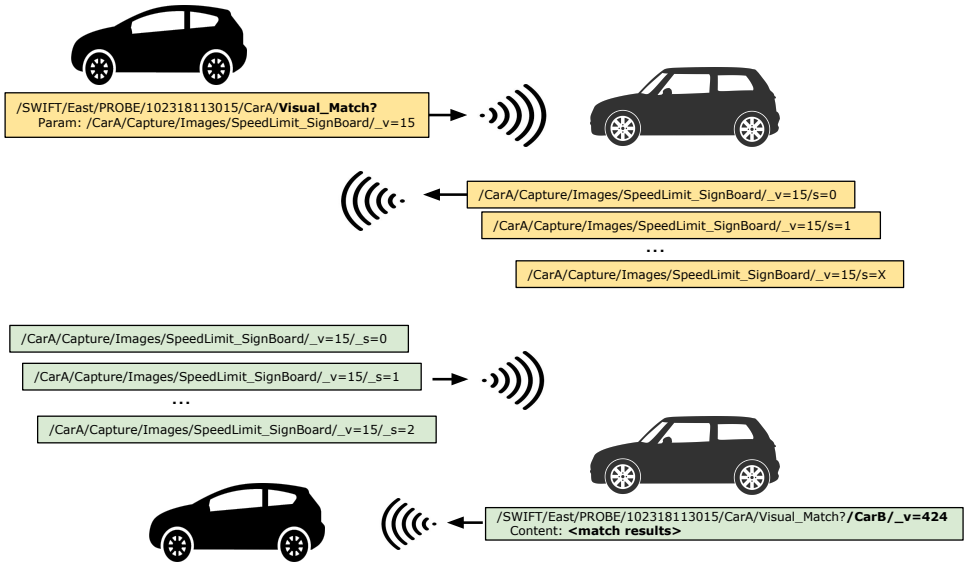

Figure 3.10: Visual Matching task

match of the specific signboard. On receiving such a request, the interested cars respond with interest packets requesting for the image to be matched. Car A sends out the data packet with the image it wants the surrounding cars to work. Car B, one of the cars interested in performing the computation on receiving the image from Car A queries its internal sensors (like a camera) to capture the particular image in question and thus use it for the matching process. Car B then would perform the matching operation and return the result to Car A as a data packet with the content specifying if it is a match or not. Car A could alter the images being sent and based on its knowledge of if the image/pattern should match, can determine the trustworthiness of the Car that responds.

After all the tasks have been completed, the vehicle computes the trust scores based on the cognitive and normative component values it has aggregated for a particular vehicle. If the trust score thus computed exceeds a set threshold defined by the developer, the car will transmit an interest to collaborate in a lane change.

The trust scores thus computed for individual vehicles can be stored in a ledger and compared for reciprocity and thus lead to building long-standing reputation values.

3.3.5 Security Properties and Threats

In vehicular communication environments, we encounter malicious nodes that intend to attack and bring down the system/network. Deploying an NDN-based approach handles some of the attacks. However, the content-centric approach raises the possibility of encountering newer and more complex attacks/threats. A list of possible threats/attacks with potential counter-measures is provided below.

Denial of Service (DoS) Malicious vehicles could send multiple queries to spam the network. This makes the network unavailable for legitimate users requesting data at the same time. The outcome is an increase in lost packets and exchange of NACK packets in the network. NDN solutions to counter the DoS and Distributed Denial of Service (DDoS) attacks as highlighted in [CCGT13, GTUZ13]. Other solutions include limiting the number of outgoing interests to a threshold above which the vehicle cannot transmit new interests until their existing interests have been satisfied.

Replay As explained earlier, every vehicle can assume multiple roles depending on the scenario. A vehicle can transmit interest packets with the intention to identify trustworthy neighbors and simultaneously be answering the requests of other neighbors. There is thus a possibility where some of the vehicles replay the responses from other vehicles to gain a trustworthy status. Timestamps along with nonce's and other freshness metrics are possible counter-measures.

Collusion Vehicles with malicious intent can operate alone or with the help and support of other surrounding peers to either wage any or multiple of the attacks mentioned above trying to break down the network. A simple example could be in vehicles trying to falsify information and gain access to the network.

Fake Data Injection (Cheating with Sensor Information) Attackers try to alter their perceived position, speed, direction, etc. to escape liability, notably in the case of an accident. In the worst case, colluding attackers can clone each other and harness the full trust of the target vehicle.

ID disclosure To track the location as in a *Big Brother* like scenario, wherein a global observer constantly monitors trajectory information of targeted vehicles. This data could later be used to profile the user and try to infect the vehicles with malware.

In this chapter, we discussed the need for secure communication among social objects and the requirement of transient trust to initiate communication. We also identified a task-oriented approach to accomplish short-term transient trust among entities that have had no prior communication.[7]

3.4 Summary

In this chapter, we discussed two approaches towards trust-bootstrapping of entities in an IoT network. We surveyed the various options available under the vast number of out-of-band auxiliary channels that can be used for the exchange of cryptographic material required for successful bootstrapping. With NDNViber, we designed a dy-

[7]The security and privacy issues specified in the paper offer many open questions to be addressed and is a motivation for our future work

namic, automated technique to onboard multiple devices simultaneously in devices with no interfaces and even in scenarios where the devices are installed in locations that are unapproachable. As a part of this work, a new encoding technique was introduced that leads to very low errors in the transmitted information. The vibration-based channel used decreases the attack surface significantly and also enables the use of a commodity android device as the controller.

Furthermore, we also discussed a technique for establishing short-term transient trust among vehicles in a vehicular network that can be used to initiate message exchanges and hence play a vital role in securely performing complex collaborative tasks like lane changes, platooning, etc. The proposed request-response-based approach to compute cognitive and normative components for final trust evaluation can be applied for any network involving communication among social objects that have not had any prior interaction.

Chapter 4

Authentication techniques

NDN [ZAB+14, ABR+18], 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 [YAC+15b]. The NDN cryptographic signatures of packets at the network layer ascertain authenticity. However, the traditional RSA and ECDSA public key signatures that are used currently require obtaining the signer's NDN certificate (and, if needed, the next-level certificates of the trust chain) to validate the signatures. This mechanism as defined in the NDN packet specification [NDNb] has two distinct disadvantages/problems (a) the communication channels must be always active in order to retrieve the certificates, which is not always true in IoT systems[1]; (b) the certificate identifies the individual producer and thus producer anonymity cannot be guaranteed if necessary. In this chapter, we study these two problems and propose an NDN attribute-based signature (NDN-ABS) mechanism as a potential solution for the IoT environment wherein devices that have joined the network can produce and authenticate the content they share. We also propose the retrieval of multiple simultaneously valid certificates in the design of CertCoalesce.

4.1 NDN-ABS: Attribute based Signature Scheme for Named Data Networking

To address the issue of network dependence for signature verification and to ensure conditional anonymity of the data producer, we propose a novel signature scheme inspired by the attribute-based signature techniques. With the approach proposed

[1]IoT devices tend to hibernate often to manage the limited power and resources it possess

in NDN-ABS, consumers (IoT devices or controllers) can verify the signature and ensure integrity and authenticity of the produced and communicated data without the need for any additional information from the network, provided they are provisioned with the attribute authority's public parameters (i.e., NDN-ABS trust anchor). Every entity in the system, requests the public parameters from the authority only once and then install and stores them (e.g., in persistent storage for reuse). Moreover, the authority (the controller or data owner) has public keys that are constant and do not change irrespective of the number of attributes used in the signature generation.

NDN-ABS provides a system where neither signatures nor certificates can be used to correlate a set of data to a single entity thus ensuring data integrity and provenance while still being able to preserve the anonymity of individual publishers. A system deployed to use NDN-ABS ensures that neither signatures nor certificates generated and used can be correlated to a set of data from an entity. The data producers have the flexibility to sign data using attributes of varying granularities, revealing more or less (conditional privacy) about themselves, as required by the system design. The traditional identity-based signature schemes proposed in [Sha84, ANK$^+$15, MBO16, IZS13] cannot provide producer anonymity which is a major contribution of the proposed scheme.

In particular, for IoT applications, the devices can sign data merely with "$<$ *Registration* $>$" and "$<$ *Location* $>$" attributes; while the controller can sign with "``Namespace''", "ControllerID" or other corresponding attributes.

Our contributions in developing NDN-ABS [2] are five-fold:

- We designed NDN-ABS by integrating ABS signatures as part of NDN protocol operations: (a) defined a new signature type; (b) data formats and naming

[2]The detailed design and contributions are specified in the article [RTT$^+$19]

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 the time-limited validity of NDN-ABS signatures, approximating validity periods of traditional certificate-based signatures.
- We created the first comprehensive prototype implementation of the ABS signature mechanism, which was proposed by Maji et al. [MPR11] in 2011, but, to the best of our knowledge, did not have a standalone codebase support[3].
- We evaluated the ABS signature performance overhead and proposed potential ways to optimize signing and verification in production environments.
- We discussed NDN-ABS in the context of multiple attribute authorities, ABS signature revocation strategies, and NDN-ABS adoption challenges.

4.1.1 Motivation

The mandate that all NDN data packets are signed makes the proposed signature scheme applicable for many ICN applications apart from use in IoT environments. One of the expected features of IoT applications is its reliance on ad hoc communication that is enabled by ICN/NDN technology. The existing signature mechanisms have these two distinct disadvantages pertaining to IoT scenarios:

- with the 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

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

- the corresponding public key of the signature can be used to identify individual data producers unless the same private key is shared among different users (which is a dangerous practice).

When using NDN-ABS, these two problems can be effectively solved. For example, the smart-home owner can act as an attribute authority (AA) to issue attribute secret keys (ska) to authorized entities[4]:

- controllers, sensors, actuators, devices, network elements, and other entities can receive "[device-name]", "[ID]", "[capability]", and other attributes by registering and bootstrapping into the home network;
- edge devices can be configured with "[home-name]", "[unit-id]", etc., obtained from responsible personnel/entities; and
- guest devices can receive the "guest" attribute if required in the system to verify when they are present in the smart-home site.

In addition to attributes, the authority also publishes its public parameters data packet ("PubParams") that act as the trust anchor certificate, which can be provisioned on the devices during attribute request or through dedicated bootstrapping protocols [MTM18] one of which was described in the previous chapter. With this initial setup, IoT 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., "[house-name] AND ([capability1] OR [capability2])" or "[house-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 parame-

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

ters (which, as mentioned above, is already known). This effectively addresses the first problem of traditional signatures (Figure 4.9).

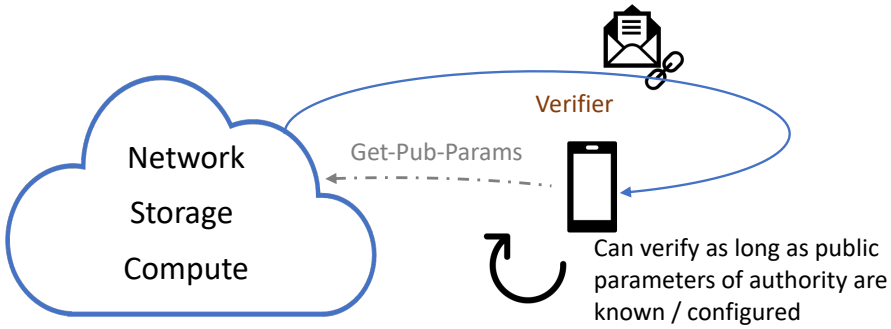

Figure 4.1: Ease of verification using NDN-ABS

The specific attributes used in the policy predicate allow the producers to reveal or hide the identity (actual identity or a pseudonym) of the individual producer. The properties of ABS construction guarantee [MPR11] 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 a signature with the claim predicate that covers both user's attributes (collusion resistance property). In other words, it is not possible for device $D1$ that has ska_h for "[humidity]" attribute and device $D2$ that has $skat$ for "[temperature]" to collude and sign with the policy "[humidity] AND [temperature]". With these essential properties, NDN-ABS provides a way to ensure the anonymity of the individual from the edge-computing entity by the 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 device using its attributes. NDN-ABS thus can realize the desired level of conditional privacy, as illustrated in Figure 4.8.

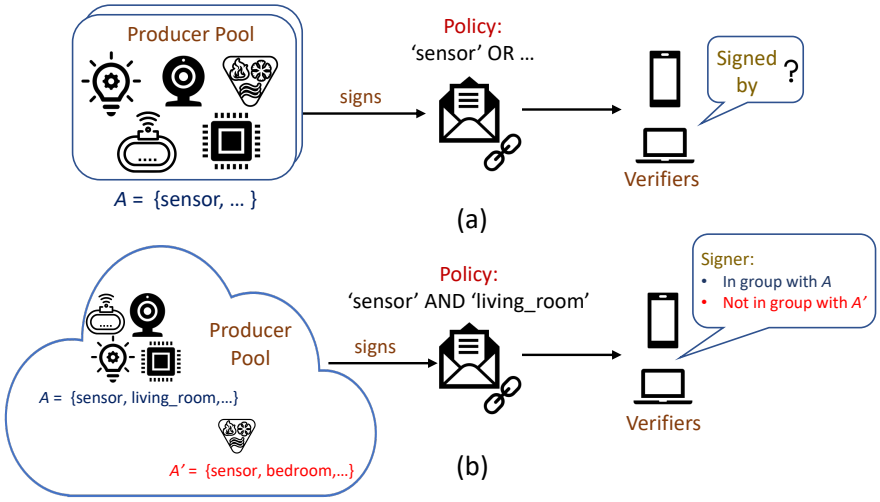

Figure 4.2: Conditional privacy using NDN-ABS: (a) Verifier unable to decide the identity of the signer; (b) Verifier can identify the group containing the signer

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.

4.1.2 Related Work

Attribute-based signature [MPR11, LAS+10] 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 to define signature and encryption policies, but they *substantially* differ in the specific algorithmic steps.

Some important terms related to ABS are as follows:

- ***Attribute Authority (AA)***: The authority is involved in the 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 controller (or homeowner) can act as the AA. In the proposed NDN-ABS scheme, the AA 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 AA is offline[5].
- ***Signer***: The user creates message signature σ with a policy predicate Υ that is defined over a subset of attributes $\mathcal{A}$ using ask obtained from the attribute authority. In the IoT scenario, we expect all the devices 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 IoT devices, controller, and the other service providers and users will have to verify the content published by the devices 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 which is a logical claim of the signer of possessing a set of attributes.

[5]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.

4.1.3 NDN-ABS Design

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.

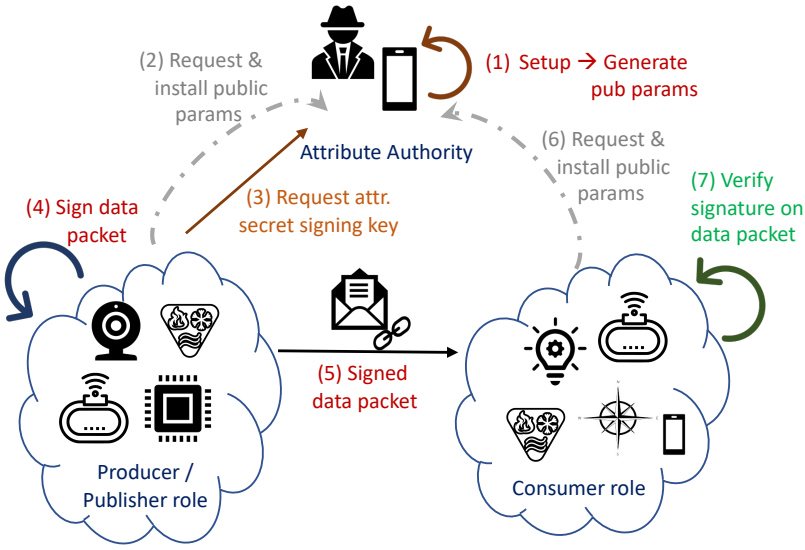

Figure 4.3: Overview of NDN-ABS

Figure 4.3 depicts a typical ABS scenario that begins with the generation of public parameters by the AA. These are further published as data packet(s) and provisioned or retrieved by all parties. However, this process is performed only once by the authority and the parameters do not change, except when rare "rollover" events similar to root key change in today's DNSSEC occurs. In 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 [ZYAZ17b] framework. To verify signatures, the consumers extract information from the Data packet's

KeyLocator field containing the name of the attribute authority associated with the signature and the claim predicate of the signature[6]. The validity of an NDN-ABS signature ascertains that a *producer*, whose *attributes* satisfy the *predicate*, has indeed signed (endorsed) the message. A more elaborate design with all the steps and the associated algorithms is discussed in the paper by Ramani et. al. [RTT+19].

4.1.4 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 $\Upsilon(u^*) = 1$ [MPR11]. Moreover, a trustworthy AA will not provide the attributes that do 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 which is 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.

[6]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.

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 [MPR11] 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.

4.1.5 Evaluation

Attribute-based signature schemes have been discussed in many works earlier. However, we were the first to implement a comprehensive Python library [RA], 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 4.4, we use various platforms running the implemented NDN-ABS library.

Signature Cost Per Attribute

Figure 4.4 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 a 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). As shown, the verification process incurs a higher cost compared to the signing process. Figure 4.4 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.

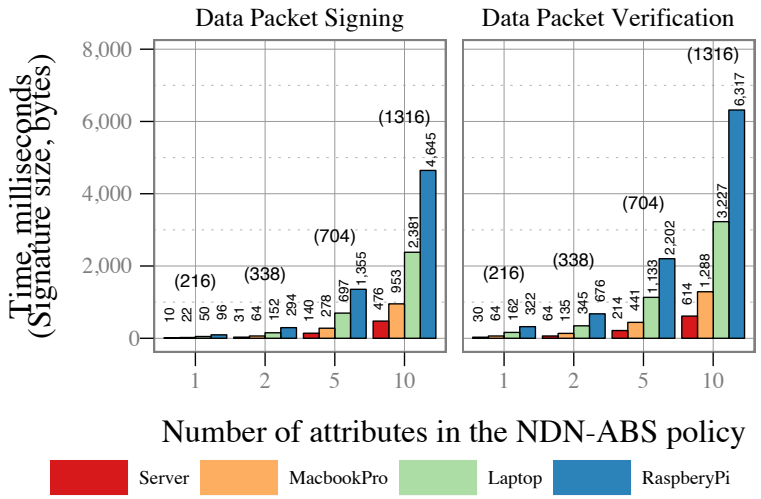

Figure 4.4: Cost for signing and verification using NDN-ABS (Server: Intel i7 4.00GHz, 62.8 GB RAM; Macbook Pro: Intel i9 2.9GHz, 32 GB RAM; Laptop: Intel T2300 1.66GHz, 2.4 GB RAM; Raspberry Pi 3: Raspbian, ARM v7 1.4GHz, 0.9 GB RAM)

As depicted in the results, the NDN-ABS signature scheme incurs a substantial computational cost, especially on limited resource platforms like the Raspberry Pi 3 which is the most commonly used platform for IoT system setup. 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. 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 signa-

ture scheme especially in scenarios where the specific advantages of the scheme are of prime importance.

Performance of the Optimized Signing

To optimize the cost for signing and verification, we ran the experiments on 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 are signed using the proposed NDN-ABS scheme. Policies generated using a varying number of attributes are used in realizing the signatures. For each such generated policy, 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 policy that uses required attributes and is not too long,
- create a manifest with appropriate group size and sign the manifest instead of signing every data packet,
- 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 4.5 shows the mean and confidence intervals for 10 runs of the experiment. The experiments were run with manifests having a varying number of implicit digests. It can be observed that the signing and verification times decrease drastically as the group size increases.

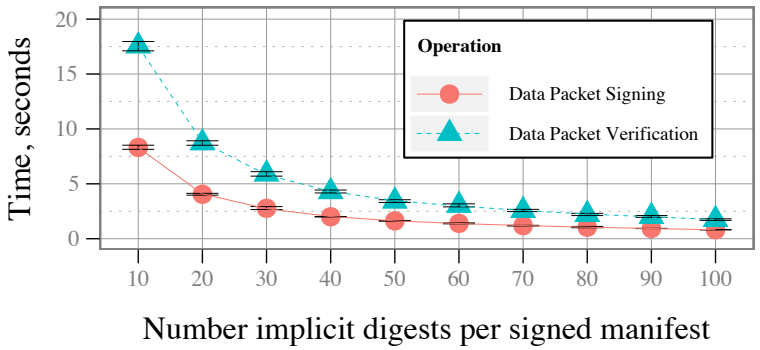

Figure 4.5: Time for signing and verification of manifests with different group sizes

Implementation Optimization

To the best of our knowledge, our work is the first comprehensive prototype implementation of the ABS scheme proposed by [MPR11] with a basic adaptation of the ABS scheme from [Mie] and thus, we do not have any reference to compare with. We 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 [Zeu, SA, JB, Agr, Wan], 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 4.6 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.

Even though the experimentation was in a device with better capability than IoT devices, the results can be consistently extrapolated. 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 costly operations as opposed to often cheap decryption operations except for the *BSWABE (Python)* implementation. We also noticed that the implementation language plays an important role in the efficiency of the system.

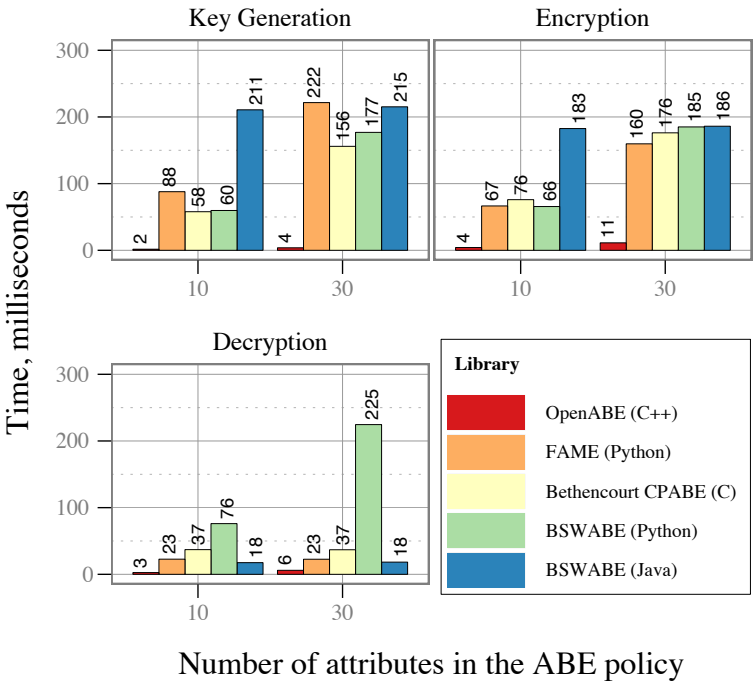

Figure 4.6: Comparison of the cost for keygen, encryption, and decryption for common ABE libraries

Overall, from the results presented above, we observe that the overhead numbers of NDN-ABS can be significantly reduced by a more optimized implementation. We also expect that incorporating hardware acceleration mechanisms can provide a significant boost to the performance and make the NDN-ABS signature an even more viable option to be used in production.

4.1.6 Discussion

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 bottleneck 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.

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 as discussed in earlier works [LW11, CZWS12, KK18]. This will prevent the users from colluding with other users and perform any malicious activities in the system. 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 [YAC+15b]. In other words, the 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.

Revocation strategies in ABS

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 credentials [PTMW10]. In [BGK08]. This is an aspect we explore in the following section of this chapter which described the details of a Butterfly-key expansion-based ap-

proach for automated generation of multiple certificates and distribution that can support the functionality of IoT systems as well.

4.2 Certificate pools for IoT applications

Named Data Networking (NDN) architecture advocates a data-centric security approach, which relies on the public key signing of the data packet to maintain data integrity and to ensure authenticity. From the perspective of least privilege separation and limiting exposure of individual keys, it is desirable to have certificates that have short-term validity, thus eliminating the need for complex revocation mechanisms. In traditional public-key cryptography techniques, entities generate public/private key pairs, followed by corresponding certificate requests for which the issuers create certificates for each of the public keys. In the case of using short-term validity for a certificate (say, five minutes), each entity will be required to generate a large number of key pairs (288 per day in the example), generate the corresponding requests, and finally store the issued certificates. In the case of IoT networks, such an approach is neither possible nor feasible, because of the highly constrained computation and storage capabilities.

In this section of the chapter, we discuss the design of *CertCoalesce*[7] which is an alternative way to generate, receive, and use multiple certificates simultaneously without incurring major overheads. In CertCoalesce, a certificate issuer only needs to receive a single request with a "master" key pair called the butterfly key, after which it can issue an unlimited number of caterpillar certificates for derivative private/public keys. Therefore, it becomes possible to maintain "infinite" pools of short-term private keys/certificates with very limited storage requirements. More-

[7]The details of the design and contributions are discussed in [RA20c]

over, CertCoalesce preserves forward secrecy (i.e, a compromised key does not reveal other keys/certificates in the pool) and inherits all the strong security assurances of elliptic curve cryptography.

To reduce the scope of the Certificate Issuer and possible key exposure (and damage if it happens), NDN advocates for the use of certificates with a short-term validity, measured in minutes rather than months or years as in today‘s practice. With CertCoalesce, we provide the opportunity to tune this value based on the application requirements. Dealing with certificates having short-validity periods is a challenge in traditional systems, and also becomes completely infeasible on the Internet-of-Things (IoT) use cases because of the following reasons:

- the constrained IoT devices may not have enough capacity to store a variety of private keys to sign produced data (certificates can be stored externally).
- given the predicted explosion of the number of IoT “things”, the number of interactions and exchanges with the certificate issuers can easily overburden the network.

Therefore, to fully realize the NDN vision especially in the IoT environment, we need a fresh look into certificate management and CertCoalesce is one of the outcomes.

The proposed CertCoalesce approach is inspired by the cryptographic constructs [CAM] and is based on existing Elliptic Curve-based public cryptography primitives. The proposed approach can significantly alleviate capacity and network overhead problems that are introduced by the use of traditional approaches to certificate management. CertCoalesce uses concepts of “Butterfly key expansion” which deals with the expansion of a “master” private key into unlimited sets of unlinkable private keys and certificates. With CertCoalesce, an IoT ecosystem will possess an option where the devices can (a) receive multiple derived certificates using a single master public key, (b) select and use certificates from the pool as long as they are valid, (c)

minimize the storage requirements for certificates and keys, and (e) limit communication with certificate issuers. At the same time, the certificate issuers preserve the ability to perform "effective revocation" of malicious devices (which is necessary for the IoT environment) by simply stopping the generation and distribution of the derived certificates inhibiting the device's ability to produce content further in the network.

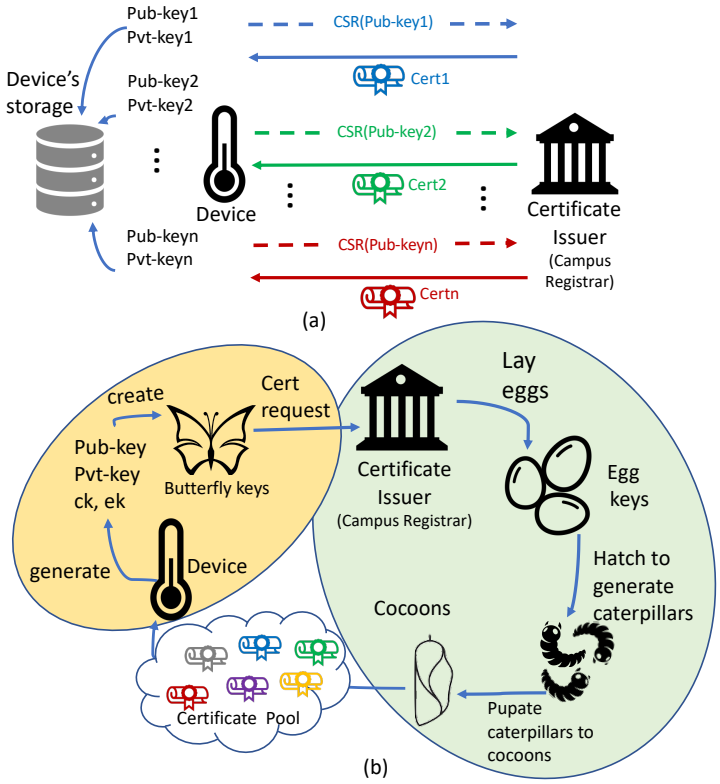

Figure 4.7: Comparison of (a) traditional certificate requisition process (a key-pair for each certificate) (b) CertCoalesce certificate requisition (1 key-pair for a certificate pool)

Following the traditional approach of certificate requisition from the Certificate issuer involves the generation of unique pairs of keys for each certificate. In contrast, the CertCoalesce approach will need a single key pair with the expansion functions to generate an arbitrarily large number of derived certificates. Figure 4.7 highlights

this difference and the limited requirement of communication with certificate issuers in the proposed CertCoalesce approach.

Our contributions in this work towards the objective of enhanced authenticity in NDN based IoT networks are four-fold:

- the design of CertCoalesce for an NDN-based IoT ecosystem by integrating butterfly key expansion with traditional crypto to form a new efficient certificate generation method
- data formats and naming for the crypto-material that is exchanged between the devices and the Certificate Issuer
- define how devices can compute multiple valid certificates with a single specialized private key
- time-limited validity of the certificates and revocation of the invalid certificates by the certificate issuer.

4.2.1 Motivation

To understand the design of CertCoalesce, let us consider an IoT-enabled smart home ecosystem. Such a smart home will be equipped with multiple sensors and actuators [RI17b] that will be involved in the exchange of interest and data packets to perform their designated activities and provide appropriate services. As an example let us consider the smart climate-control system. This system involves the thermostats, the air conditioning system, access to the flow control valves attached to the sprinklers, metering system, control of the lightings, etc. All these devices will communicate amongst most or all of the other devices for the system to seamlessly be able to control the climate in the house. If the devices use the same certificate for all such communications, there is a possibility that the associated public key will be

identified and the corresponding private key becomes vulnerable to being exploited if not stored securely. It is thus a good practice to have multiple certificates used for the various communications which will help in the device being (a) unlinkable, (b) anonymous, etc. while the data exchanged still being valid and verifiable by the consumers.

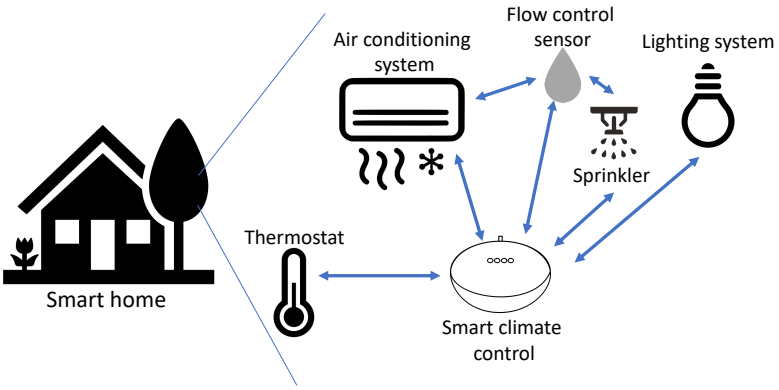

Figure 4.8: Smart home environment showing interactions among devices

Figure 4.8 shows the described smart-home ecosystem with the possible communications among the devices. Also, the thermostat shown here is specifically designed to sense the temperature of its surroundings and propagate the information when requested by the controlling and actuating system or other authorized systems. It is too much of an ask for this small thermostat to be able to hold onto multiple keys and certificates for use and even more for it to be generating public-private key pairs, creating Certificate Signing Requests (CSRs), retrieving certificates, and finally storing all these crypto-material. In comparison, what we expect is a technique like in Figure 4.9, where the IoT devices will be able to request for an arbitrary number of certificates using a generated "master" public-private key pair and use the certificates from the pool to certify its messages. The proposed CertCoalesce design provides this functionality with the certificates in the pool being such that the compromise of one will not reveal the other ensuring forward secrecy and the compromised certificates can easily be avoided and removed from the system with

minimum damage. The certificate issuer in the system will periodically provide the devices with new sets of valid certificates with short validity periods which can be tuned according to the application needs ensuring that the revocation is easier and transparent.

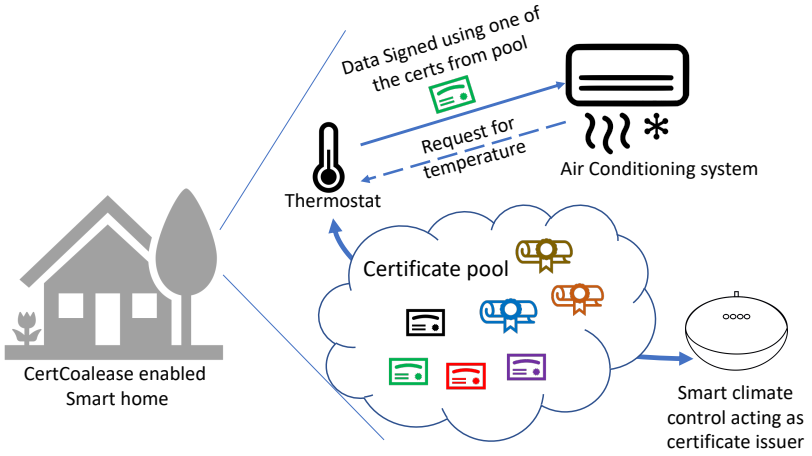

Figure 4.9: Thermostat receiving a pool of certificates by adopting the proposed CertCoalesce scheme

4.2.2 Butterfly Key Expansion

The traditional certification methods of requesting and fetching certificates will be cumbersome in an IoT environment with the need for the creation of thousands or even more public keys for the multiple certificates that are desired. Butterfly keys depict a novel cryptographic construction that can aid in addressing this problem by allowing the certificate requesting devices the liberty to obtain an arbitrary number of valid certificates using a single public key. Also, each of the certificates received will have a unique signing key that is encrypted such that only the legitimate requestor can decrypt and use the certificate. As an important feature, the use of such a method leads to a reduced upload size enabling devices to be able to request certificates even with intermittent connectivity which is common in IoT networks.

The requesting device will be the only entity that can decrypt and calculate the private keys to be used for signing.

The working of butterfly key expansion depends on elliptic curve cryptography [BH19]. A brief account of the working of the Butterfly keys is discussed here with specific design details in Section 4.1.3. The working of butterfly keys involves an agreed-upon elliptic curve base point G which is of the order of l. The devices are expected to generate the master public-private key pair and a pair of AES keys which will provide the expansion functions. The certificate issuing authority (could be one level or multiple levels based on the available resources and application necessities) with the value of the curve point derived from the signing key and a derived expansion function $f_k(l)$ will be able to generate an arbitrary number of derived certificates. The expansion function is a pseudo-random permutation in the integers $modl$ with l being a counter used by the certifier to iterate and generate multiple certificates [8].

4.2.3 CertCoalesce Design

The functionality of the CertCoalesce scheme depends on the *Certificate Issuer* (CI) who validates the certification requests and issues certificates. The other actor in the system is the *certificate requestors* who are highly resource-constrained IoT devices, controllers, and all other entities that will produce data and engage in communication in the given smart-home environment. The actual process for requesting the CertCoalesce certificates can be embedded as part of the NDNCERT [ZYAZ17b] protocol, therefore in the design, we focus on data naming and crypto aspects of the proposal.

[8] More details about this is available in the paper [RA20c]

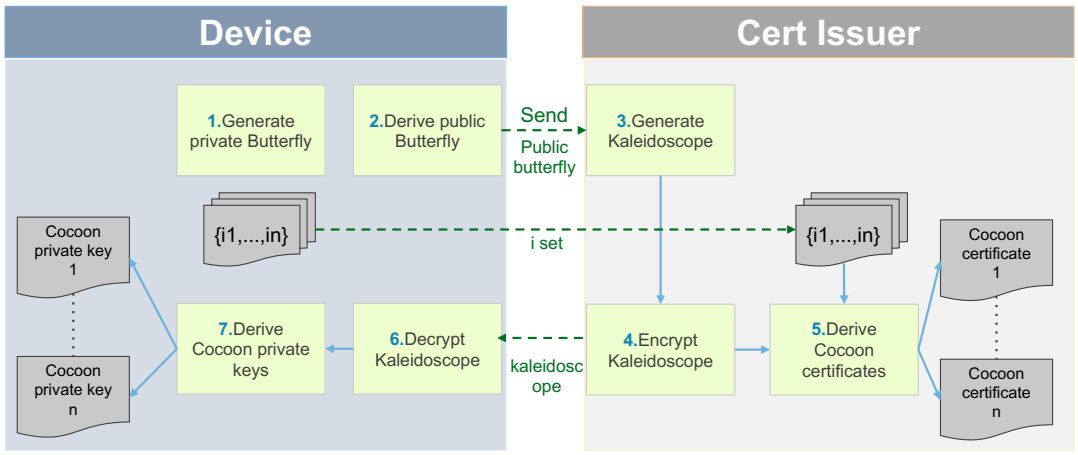

Figure 4.10: Overview of CertCoalesce design

Figure 4.10 depicts a typical certificate request procedure involved in the CertCoalesce operation. The CertCoalesce design of automated certificate issuance involves multiple steps as described below.

Butterfly Key Generation

To request CertCoalesce certificates, the node needs to generate a butterfly key according to Algorithm 1. The butterfly key-pair created acts as a "seed" and is more than just a regular key. It is a combination of ECC signing keys with ECC encryption keys and AES expansion functions. The name of this key follows the standard NDN conventions with a small specialization of the ID part is "`/<identity>/KEY/butterfly-<keyId>`".

Algorithm 1 Butterfly Key

Input: Elliptic curve base point G, f_1, f_{name}.
Output: Caterpillar private key tuple ($keyId$, c_k, a, p) and public key tuple ($keyId$, c_k, A, P)

1: Generate $keyId$
2: Generate key-pair a and the curve point $A = aG$.
3: Generate key-pair p and the curve point $P = pG$.
4: Generate an AES key c_k acting as expansion function for signing keys.

The butterfly private key component is the only key that needs to be stored on the device for a long time and is used to derive private keys of the CertCoalesce pool. However, as a limitation of CertCoalesce, the device cannot derive (and therefore use) the private keys until it receives the "kaleidoscope" from the certificate issuer.

"Laying" Egg Keys

After receiving the certificate request for a Butterfly key and successfully validating the legitimacy of the request using security challenges,[9] generates a set of "Egg keys" as defined in Algorithm 2. Each of the egg keys created is regular ECC public keys. These keys do not have to be stored separately as their names can be expanded from the "`butterfly key + ID inside the pool`". The name of the keys follows the following naming convention "`<identity>/KEY/<key-id>-<ID-in-the-Pool>`"

Algorithm 2 Egg Keys

Input: G, f_1, Butterfly public key tuple ($keyId$, c_k, A, P)
Output: Set of Egg keys B_l
1: "Lay" egg public keys $B_i = A + f_1(c_k, i) \cdot G$

As stated above, the number of the egg keys depends on the use case but needs to be known to the requester, either as a pre-configured parameter or via explicit notification as part of NDNCERT exchanges. For example, if we assume a 5-minute validity of individual certificates and that a single certificate request "covers" a 1-month time frame, then the certificate issuer will generate the egg keys set of size 8640 (12 key pairs per hour $\times$ 24 hours $\times$ 30 days).

[9]Since CertCoalesce is designed to be integrated with NDNCERT, mechanics of this verification is outside the scope of this dissertation and a complete integration is a part of the future work

Hatching Eggs / Generation of Caterpillar Certificates

Using the set of Egg public keys, the certificate issuer can then finally issue Caterpillar certificates for each of the keys (hatching each egg into a caterpillar), as defined in Algorithm 3. Note that though the issuer created Egg keys, it can only create the public keys, as it lacks the knowledge of values corresponding to a and p generated by the device, making it incapable of being able to decipher the private keys.

Algorithm 3 Hatching Eggs into Caterpillars

Input: G, f_{name}, Butterfly public key tuple ($keyId$, c_k, A, P), set of Egg public keys B_l

Output: Set of Caterpillar certificates A_l^c, encrypted secret of the butterfly set C (kaleidoscope), set of derived key ids "`derived(i)`"

1: Generate a random secret c representing butterfly kaleidoscope
2: Hatch each egg into Caterpillar certificate $A_i^c = B_l + c \times G$
3: Derive a set of certificate names "`derived(i)`" $= f_{name}(keyId, i)$
4: Create C by encrypting the secret c with P

Each created certificate will follow the NDN certificate naming conventions [NDN20]:

"`<identity>/KEY/<key-id>-<ID-in-the-Pool>/Coalesce/_version=<XX>`".

Note that we are still exploring options for the most appropriate f_{name}. For example, it can be a SHA256 over the original key id and the sequence number.

Note that CertCoalesce requires that the encrypted secret C (which is the "kaleidoscope" value) is communicated back to the requestor before it can start signing. In other words, the private caterpillar keys cannot be derived until c is known.

The certificate issuer does not need to generate all of the certificates in the set right away. To allow for "revocation", it can only generate certificates on-demand, or pre-generate sets of certificates and publish them (e.g., in NDN repos) in periodic batches, such as every day.

Pupate Caterpillar into Cocoons (Deriving Private Keys)

As soon as the device receives confirmation of the issued Caterpillar certificates, it can immediately pupate the caterpillars or, in regular terms, derive private keys as described in Algorithm 3.

Algorithm 4 Pupate Caterpillar into Cocoons (Deriving Private Keys)

Input: f_1, f_{name}, C, Caterpillar Private key tuple ($keyId$, c_k, a, p), i
Output: Derived Private key $PrvKey(i)$

1: Use p to decrypt C and recover value of c
2: Calculate b_l as $b_i = a + f_1(c_k, i) \cdot G$
3: Derive private key id "`derived(i)`" $= f_{name}(keyId, i)$
4: $PrvKey_{cacoon}(i) = b_i + c$

Note that such derivations can be done on-demand, depending on the need and/or use case. In our example with 5-minute validity per certificate, the device will need to run this derivation a dozen times an hour, which correspondingly increases the value of i. However, there is no additional storage overhead and derivation computation overhead includes only several elliptic curve point calculations.

4.2.4 Security Analysis

The primary motive of any adversary in a system using the CertCoalesce design will be to identify the private keys corresponding to a received set of butterfly public keys in polynomial time. So even when the values of multiple derived certificates like $(A + f_k(1) * G + c_1 * G), (A + f_k(2) * G + c_2 * G), ..., (A + f_k(q) * G + c_q * G))$ are given, it is close to impossible for the adversary to identify $a + f_k(x) + c_r$ for any $1 \leq r \leq q$. The expansion function f_k used is a pseudo-random permutation making it hard for any polynomial-time adversary to be able to distinguish between the various expansion function (f_k) outputs (that are truly random values) and thus be able to recreate the values of c_k, e_k, etc. Based on the underlying constructs and the

derived values, we can concur without any loss of generality that the CertCoalesce design is highly secure given that the elliptic curve discrete logarithm problem is hard to crack in polynomial time [GG16].

Also, the fact that the certificate-issuing authority does not have complete information of the values of the device-generated keys and the recreation of the keys with the available information being hard in polynomial time, ensures that the proposed CertCoalesce design can secure the system against common security attacks[10]. The CI on receiving a single request and the expansion functions can generate sets of simultaneously valid certificates at a constant frequency determined by the activity of the device and the validity of the certificates provide the sets to the devices. Each of the certificates is unique and thus compromise of one of the certificates will not yield access to the other certificates upholding forward secrecy. With the validity of certificates being tunable based on the certificates and the forward secrecy in place, it is easier to publish revocation lists and the device can use other valid certificates in the pool to continue its operations.

After the initial computation related to the creation of the caterpillar keys, the device will have to only store the values of a and p (the master keys) that will be used in decrypting and deriving the butterfly private key for the certificate that can then be used to sign the data packets. Thus it prevents the need for the device to store multiple public and corresponding private keys which if misplaced can compromise the system completely and is thus dangerous. The devices using the proposed scheme can on the fly select a certificate, decrypt it, extract the required information to derive the private key, and sign data packets. This process reduces the storage burden on devices and safeguards them against possible security infringements.

[10]The underlying construct of the design ensures that even in a system with multi-level certificate issuers, collusion attack is averted.

4.2.5 Evaluation of CertCoalesce

The initial use of Butterfly key expansion as a method to create pseudonym certificates was implemented in the Secure Credential Management System (SCMS) Proof of Concept (PoC) [CAM]. We ran our initial performance analysis experiments on an implemented python variant of the design[11] to identify the time taken for a device to retrieve certificates from the certificate issuer in comparison to the traditional approaches. The comparison involves request generation for 5 and 50 certificates and the total cost involved for the entire certification retrieval process. The CertCoalesce certificate requisition and retrieval process involve a single instance of generation of the public and private key pair and the initial computation of the caterpillar key which is sent for retrieving an arbitrary number of certificates hence is highly cost-efficient in comparison to the traditional approach (see Figure 4.11 (a)).

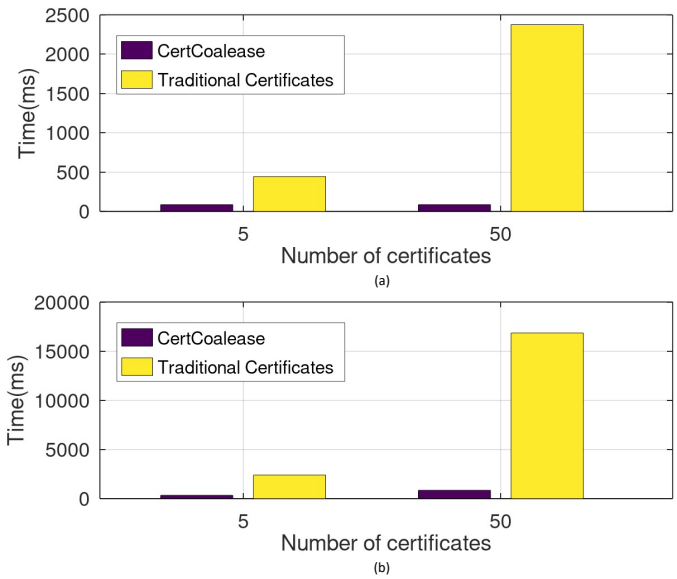

Figure 4.11: Comparison of performance of CertCoalesce: (a) Time taken for generating certificate request; (b) Time for entire process till certificate is received and extracted

[11]At the time of writing this dissertation, we are still exploring better options for the namespace design as a part of the future work in this direction

Figure 4.11 (b) shows the comparison of the time taken for the certificate requisition using the two approaches. The CSR in CertCoalesce design has to be created only once and involves the creation of one pair of public and private keys as against the traditional methods that involve a new pair of keys and a brand new request for all the certificates. The performance of CertCoalesce is thus highly superior in this regard. The design also ensures minimal storage requirements for certificates and keys as the device will have to store only the master keys while the remaining certificates can be stored in a repository. The communication requirement for retrieving these large number of certificates is also reduced considerably as the CI can use the series of cocoon keys to periodically generate derivative certificates and send them to the device for use. Also, the entire process of retrieving the certificate from sending the request to receiving the response consumes way less time than the traditional method.

4.3 Summary

In this chapter, we discussed two approaches towards authentication of messages transmitted within the IoT network. With NDN-based networks requiring that every data chunk be signed at the site of production and the consumers or any intermediate node are given the option to verify the authenticity through the verification of the signatures. We identified the major issues with the existing approaches when adopted to an IoT network which usually witnesses highly intermittent connectivity were related to the high reliance on the network to retrieve verification keys and certificates requiring all the network components to be online always and the possibility of the producer privacy being breached if the certificates are reused multiple times for signing data-chunks.

In this chapter, we proposed two techniques to address the above issues. NDN-ABS is built over the Elliptic curve cryptography and Attribute-based signature primitives to be able to create policies using certain chosen attributes from the producer's list for signing. At the receiver's end, as long as the device has been bootstrapped and has received the trust-anchor information, verification is possible without any dependency on the network. This drastically reduces network overhead. We implemented the first python based library to realize this design and also identified techniques that can provide better performance in the production setting.

To improve on the conditional privacy benefits that NDN-ABS provides and to eliminate the need for CRLs, we developed CertCoalesce which provides a certificate pool of short-term simultaneously valid certificates to choose from to sign the produced data packet. Based on the proposed design, a single master key pair can be used to request multiple certificates (which the issuer can provide periodically as well) and only the initially generated master keys will have to be stored to be able to derive all the certificates in the pool. This technique is built over the security benefits offered by the Elliptic Curve Discrete Logarithmic problem. The certificates in the pool also ensure that the forward privacy is maintained (i.e., the compromise of any of the certificates does not compromise the other certificates and keys in the pool).

Overall, using the above two approaches, we can drastically reduce the network dependence for the signing and verification operations and hence the traffic that is generated for these purposes while being able to provide conditional anonymity to the data producers.

Chapter 5

Data Confidentiality and Access Control

A common and important requirement of IoT and distributed applications is an effective and usable access control solution that can ensure application features and information access to only authorized devices/users. Literature showcases numerous methods that can provide efficient access control [KRD06, Shi07, GPSW06, BSW07, JMB11] have been proposed. Their implementation in the current internet architecture involving TCP/IP protocol stack though is cumbersome and prone to faults especially when distributing access keys. The IP model utilizes DNS services that are offered by third-party service providers for effective key distribution and storage making the system vulnerable and prone to attacks. In an IoT environment, this issue is magnified purely because of the involvement of a large number of entities which complicates the key management process that is vital in providing effective access control facilities.

We thus explore the use of an access control mechanism based on the semantically rich NDN names that can provide content confidentiality and highly fine-grained access control [1]. Name-based Access Control (NAC) scheme, thus is built to provide automated access control using the combination of symmetric and asymmetric cryptography algorithms. The advantages we achieve by using this approach are:

- automatic retrieval of cryptographic information using NDN naming conventions
- effective and flexible namespace design providing fine-grained access control
- support resilient communication and data integrity and confidentiality even with intermittent connectivity using in-network caching.

[1] The content of this chapter was published in MILCOM 2018 [ZYR+18]

5.1 Access Control using names

We assume that the IoT devices have by now been bootstrapped and the messages exchanged by them can be authenticated by the network. The design of using names to perform access control assumes that proper trust relationships have been established among the communicating entities in the system. In our design, we assume the existence of an entity that takes the role of being the owner of the information (data owner), and to better explain the design we will consider the example of a smart-home. The owner of the house thus is considered as the data owner. The data owner is also called the *access manager* and is given the right to define the policies that will be used to identify who has access to which content. The access manager is then responsible for the generation of a key pair which is called the *KEK - Key encryption Key* and *KDK - Key decryption Key.*

The design uses the NDN naming convention to define the granularity of access management with the components in the KEK name determining the prefix of the content that can be encrypted with the key and the corresponding decryption keys (KDK) is given to the appropriate entities that should be able to access the information. In the example depicted in Figure 5.1, the access manager has generated the specific KEK and KDK keys and distributed them to the producer (a smart thermostat) and consumer (air conditioning system) based on the policies defined ensuring that the air conditioning system can access the information published by the thermostat. The working of this system is such that the thermostat on creating data (of the temperature/humidity in a room etc.), encrypts the content with a *content key (CK).* The CK thus generated is encrypted with the KEK that the thermostat received from the access manager. The encrypted content is then pulled by the air conditioning system which then again requests for and receives the en-

crypted CK. The air conditioning system then requests the access manager for the decryption key (KDK) and based on the policy defined, is either given access to the key or not. If the policy authorized the air conditioning system to be allowed to read the content, the KDK is provided which is used to decrypt the encrypted content key to reveal the CK which is then used to decrypt the actual content.

It is important to note that the named policy can be configured or inferred from configuration and data name and the content is not directly encrypted using the KEK As a result, encryption (or decryption) key chain can be established from a producer to a consumer under the control of the access manager.

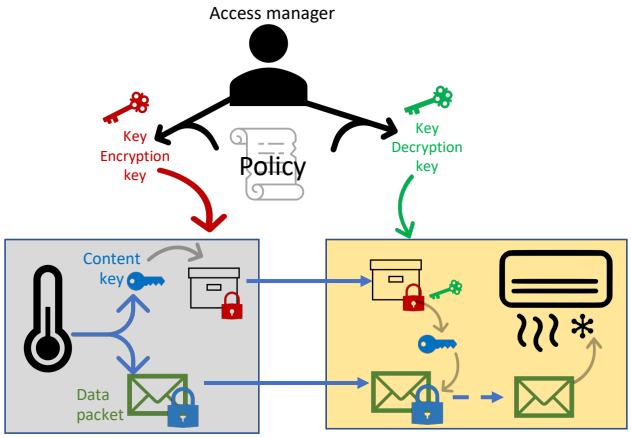

Figure 5.1: NAC Scheme

The hierarchy thus defined does not have to be strictly enforced. In a typical IoT system, the access manager, encryptor, and decryptor could be the roles adorned by the same entity at different points in time. For instance, in a smart home, the controller can act as an access manager, encryptor (when bootstrapping other devices), and decryptor (when working with responses from other controllers or devices).

Fine-grained access control can be achieved by following the KEK name to encrypt the content. The more specific the KEK name is, the fewer Data packets can

be decrypted using the corresponding KDK. NAC also allows the access manager to control the access through KDK distribution. The more KDKs a decryptor can obtain, the more Data packets it can access. The different keys we have discussed thus far are packaged as data packets that can be retrieved from the network using the wel-defined request-response architecture advocated by NDN. Also, the opportunistic in-network caching aids in effective dissemination of the keys even in intermittent connectivity and mobility which is common in IoT scenarios. A detailed account of the entities involved in and the working of NAC is defined in the paper by Zhang et. al [ZYR+18].

5.1.1 Specialized Naming Conventions for access control

A well-defined naming scheme is required for both the retrieval of packets and the automated distribution of keys in the system. Encryptors / Data producers need to fetch the KEK generated by the access manager to encrypt the content key that they used to encrypt the content. The naming convention for KEK and KEK Data packet thus can be in the following format are:

KEK Name = "`/<access_manager_prefix>/NAC/<granularity>/KEK/<key-id>`"

where the access manager prefix field indicates the producer of the KEK, and the granularity is the name prefix of the data that is being produced by the encryptor/producer for which this key is being retrieved, and the key-id is the unique identifier of the key.

The corresponding KDK follows the same naming convention as the KEK and is void of the name component "KEK" which was present in the KEK and is now replaced with KDK:

KDK Name = "`/<access_manager_prefix>/NAC/<granularity>/KDK/<key-id>`"

Because the KDK is encrypted for each authorized decryptor/consumer, the KDK Data packet has additional components:

KDK Data packet Name = "/<access_manager_prefix>/NAC/<granularity>/KDK /<key-id>/ENCRYPTED-BY/<consumer_prefix>/KEY/<consumer_key-id>"

where the *key-id* is the same as its corresponding KEK, and the key identified by decryptor/consumer key-id is the consumer's public key that is used to encrypt the KDK.

CK name and CK Data packet names follow conventions similar to KDK. The CK Data packet name follows the naming convention:

CK Name = "/<content_producer_prefix>/CK/<key-id>"

CK Data packet Name = "/<content_ producer_prefix>/CK/<key-id> /ENCRYPTED-BY/<access_manager_prefix>/NAC/<granularity>/KEK/<key-id>"

Key Generation and Delivery

The access manager is responsible for the generation of the KDK and KEK pairs as in the access control policies with the KEK put in as a part of the data packet. The access policy is next defined to identify the entities that have access to the information and thus should be able to retrieve the corresponding KDK. Once the policies are defined, they are published and the access manager does not have to be online. In-network caches are used in cases when there is intermittent connectivity and the keys have to reach a large number of devices.

The KEK and KDK names following the above-mentioned naming conventions carry the information as to (a) who can access the information, and (b) which set of keys will have to be fetched and used to decrypt and access the content. This

flexibility allows for the granularity to be altered as per the requirement of the system and application that it is servicing.

Content Encryption

Symmetric encryption mechanisms like AES-CBC [FGK03] are used in the production of encrypted content (Encrypted with content key CK). The retrieved KEK provides the encryptors with information as to the granularity to which it can use and encrypts the content with the CK and in turn the CK with the KEK which is then bundled into a data packet that is published.

Content Decryption

Only authorized decryptors should have access to the content and hence should be able to get the CK that was used to encrypt the data. The fetched content, provides information about the KDK to be used and retrieves this information from the network based on the information in the key-locator field, and then uses the CK to finally be able to retrieve the content.

5.2 NAC based on Attribute-based Encryption

The prototype of the above-described access control scheme using names used RSA [ZYR+18] but did not scale well when deployed in scenarios involving a large number of consumers which is a common sight in an IoT ecosystem. Assuming that there are about k devices deployed in the system, and p authorized granularities, to grant access permissions for each of the k devices to all the p granularities will need about $O(p)$ KDKs thus burdening the access manager who will have to produce $O(p)$ pairs of keys (KEK and KDK) and $O(p \times k)$ KDK data packets.

If we consider a situation that demands fine granularity of data access, this number will increase as the number of components in the KEK will increase thus increasing the corresponding KDK count as well. Thus, based on our understanding of the attribute-based system from the previous chapter, we define a Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [BSW07] variant of the access control scheme using names to define *NAC-ABE*. This modification provides better control on defining the access policies even when the number of nodes/devices in the system is large.

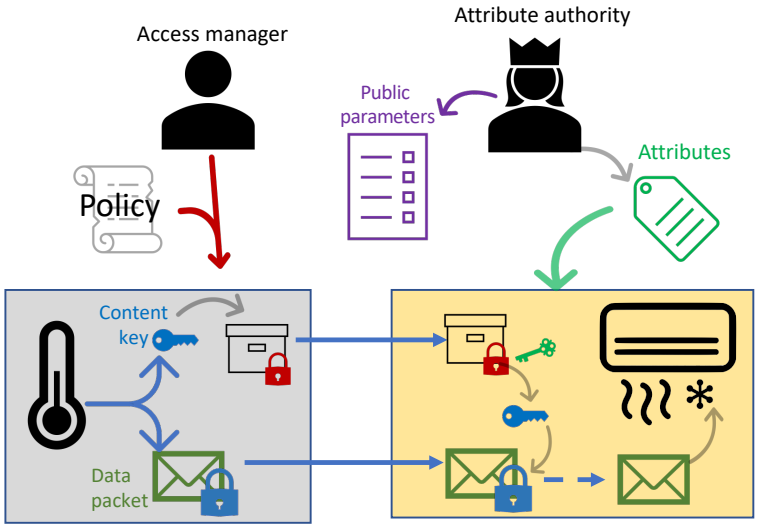

Figure 5.2: NAC-ABE Scheme

Figure 5.2 shows the proposed model wherein the KEK is modeled as an attribute policy and the decryption key is a set of attributes that can satisfy the policy. Thus, the encryptors/producers, encrypt the content-key using a specialized policy that satisfies the attribute list that the consumers/decryptors can retrieve from the attribute authority defined in the system and thus be able to decrypt to reveal the content-key which can further be used to decrypt the content received. We can either have a separate access manager and attribute authority or a single entity performing both roles.

In comparison to the generic NAC model, considering k devices, and possible p granularities, the access manager can define A different attributes to create ABE keys for O(A) times. It is important to note here that A is much smaller than k and the access policies are combinations of these A attributes using "AND", "OR", "NOT", or threshold gates. Since the data consumer's attributes can be issued at one time, the access manager only needs to generate O(k) packets and can be provided to the devices when they are bootstrapped into the network. The public parameters of the attribute authority are also published just once and can at any time be requested for and used in the system. In practice, this process can be greatly improved by issuing attributes in groups or along with identity certificates.

5.2.1 Specialized Naming Conventions of NAC-ABE

The naming convention follows the general naming conventions explained above with minor modifications. The "`/<key-id>`" component of the KEK name is no longer a unique identified but a policy of attributes defined by combining the attributes using common "AND", "OR", "NOT", and threshold gates. Through the name, the consumer/decryptor learns which attribute policy should be used to encrypt the data in granularity.

When a decryptor needs to fetch an authorized attribute from the attribute authority, the decryptor can generate the attribute Interest packet by following the convention.

Attribute Interest Name = "`/<attribute_authority_prefix>/ATTRIBUTE /<attribute name>/ENCRYPTED-BY/<consumer_prefix>/KEY/<consumer-key id>`"

In NAC-ABE, as in any attribute-based system, attributes are expected to be provisioned before the system starts.

5.3 Security Assessment

Our focus here is on threats that are specific to communication confidentiality and access control. NAC is based on NDN, and thus inherits all the security benefits of the data-centric model that NDN advocates. The NDN prowess to mitigate MITM and DoS attacks are studied by researchers and thus is applicable in this context as well.

Eavesdropping Attackers may sniff on the broadcast media or retrieve published Data packets from in-network caches. However, since all the sensitive content (e.g., data, CK, KDK) in NAC are encrypted, even though attackers can collude and aggregate the sensed and published information from the various sensory nodes, they cannot make sense of these ciphertexts.

Device Compromise Attackers may compromise individual devices to gain the data access that is granted to the unit. There is no means to stop a compromised device from accessing the previously published content, but an access control scheme is supposed to revoke the device's privilege as soon as possible in order to prevent further leaks of data. Short-lived KEK-KDK pairs are used in the NAC design to reduce information leakage in cases of compromised devices. Based on the application‘s needs, the access manager may have to take an initiative to notify encryptors to re-encrypt the content using new keys before the old keys expire.

Man-in-the-Middle Attack In NAC, when attackers perform Man-in-the-Middle (MITM) attack and modify the KDK packets, signature verification can throw light on this abnormality to the decryptors.

Denial-of-Service Attack Since all the content Data packets and key Data packets are published in the NDN network, these packets can be cached in cache or dedicated data repositories. NDN and thus NAC's resilience methods to DoS and DDoS attacks are derived from the concepts discussed in [ZVL+18, ASWS15, DWFL13, CCGT13, AMM+13].

5.4 Discussion

Assuming a system with k consumers and p granularities, there are totally d content data packets (each granularity has the same number of data packets) that will be controlled by the access controller. In NAC, we let each decryptor has access right to all d content Data packets. In NAC-ABE, we assume there are totally A attributes and each decryptor has all the attributes. The cryptographic operations are listed in Table 5.1.

Table 5.1: Number of cryptographic operations in the design

Actor/Role	Traditional NAC Scheme	NAC-ABE Scheme
Data Owner/controller	Gen keys + distribute	Gen attributes + distribute
Producer (Encryptor)	Sign data + encrypt	Sign data + encrypt
Consumer (Decryptor)	Decrypt + verify signature	Decrypt + verify claim

In an access control system over TCP/IP, to achieve key distribution, the network configuration (e.g., IP address, DNS name) or the equivalent service invocation (e.g., database query) is linear to the number of keys in the system. In contrast, the network configuration for key delivery in NAC is independent of the number of keys.

5.5 Summary

The existing content sharing techniques rely heavily on third parties to host content and security is enabled using encrypted channels. The channels thus created are not directly between the data producer (or host with the data) and the data consumer but involves multiple containers from various service providers and hence end-to-end data confidentiality is very cumbersome. This is amplified in an IoT environment where we have a large number of producers and consumers in a highly chaotic environment with a protected channel to deliver information not being very secure in ensuring confidentiality and integrity.

In this chapter, we described the use of names for ensuring access control and hence maintaining data confidentiality in NDN-based environment. The proposed design eliminates the need for secure channels or third-party services to deliver data but ensures that the data by itself is secured. We discussed the design with an access manager and a set of rules in the form of a policy that can be used with specialized naming to achieve fine-grained access control. We also introduced the use of attribute-based encryption techniques for better flexibility and scalability in situations where the number of consumers is very large (which is common in the IoT use-cases).

Chapter 6

Application Use-cases

6.1 Vertically securing Smart Power Distribution systems using NDN

Smart grids (SG) is a modern adaptation of power systems that deals with efficient power distribution deals using digital communication to satisfy the demand for power from consumers [KCW+14] and effective regulation and billing for the power generators and distributors. The use of smart grids decouples the strong interconnection between power generation sites and the consumption sites and makes power available as a commodity that can be requested based on the need by any consumer. As a distributor or power generator, the use of SG's the enables pooling of generated power from various sources and distributing and trading is based on the requirements and demand, and localization. Successful deployment of smart grids will provide a hassle-free catering to the worldwide need for power. As a consumer or producer of power, the expectation from such a smart system is to ensure affordable power is available according to the demand with transparent and automated billing options.

Sensors and sensor networks [RI17c] play an important role in smart grids. Numerous sensory devices are usually deployed to assess the flow of power in the system be it at the site where it is generated, the intermediate locations where the power is transformed using step-up and step-down transformers, the power entrance into a residential house/commercial establishment, etc. Figure 6.1 highlights the flow of power from the generation station(s) to the consumer(s). The Power generation site is composed of various renewable (Solar, Wind, Tidal, etc.) and

non-renewable (Thermal, Atomic, etc.). In a typical smart grid scenario, the power corporation/distributor will have the capability to pool this generated power into a "Power Pool" and based on the request/demand for power give it to the appropriate transmission site(s) that can forward it to the receiving site(s) and finally route it to the consumer(s). Request for a specific quantum of power received by the power grid from the consumer (or smart devices) leads to the power pool delivering it based on availability.

At each of these sites, deployed sensors sense the quantum of power that has been transmitted or consumed and if any of it is misused (by attackers/malicious users) and accordingly generate the bills. The consumer's bill is updated based on the consumption and the producers' share in each transaction is updated (there can be multiple producers involved from multiple sites serving the pool for a transaction). Secure exchange of actuating and transaction information is of utmost importance in smart-grid communication. Traditional communication approaches have proven shortcomings [TBES16, MCHM13] when deployed for smart grids. NDN provides a data-centric approach to communication among the entities and stakeholders in the smart grids and the development of a highly robust grid that is vertically secured from end to end.

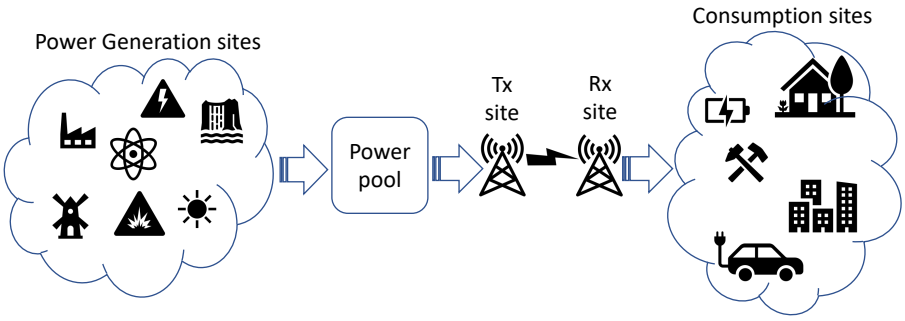

Figure 6.1: Power Transmission overview

The data-centric architecture of NDN ensures that the grid nodes are "off-by-

default" (unreachable) until they explicitly either need or would like to share information with other nodes in the grid. NDN also advocates data-centric security where every single piece of data communicated in the system is secured, authenticated, and when needed encrypted. This inherently prevents "bad" data from entering the system and significantly reduces potential risks by ensuring that data communicated in the system is secured (authenticated) and when needed encrypted preventing malicious information from entering and compromising the system. This helps in the creation of a system with a reduced risk of attack and has been shown to eliminate common attacks like Distributed Denial-of-Service (DDoS) [GTUZ13]. The benefit of NDN is even enhanced by the support it provides not just to the network but for the fact that the security model can be extended to the software and hardware components of the energy system. All the updates that are communicated in the system are secured along with the firmware being secured. The preset of NDN is naming data which ensures that nothing that is communicated within and across the system is forgeable and the content producer can not repudiate the creation of the information.

The current smart grid communications involve the use of host-centric approaches which makes the system incapable of effective sharing of information. The power systems are highly diverse and in most cases are legacy systems that are at times incapable of effectively communicating with the modern smart systems. These interoperability issues and the need for modification of the protocol stack for networking and the added need for container/session-based security are a hassle and challenge to be addressed. These challenges make the trading of power among the users, power corporations, and others complex and at times improbable with this approach. NDN along with providing secure communication among the nodes in the system also provides other benefits due to the flexible naming which ensures seamless

integration among the various applications and devices being used. A subset of the benefits includes optimal paths used for routing information, opportunistic caching at nodes (in-network storage) of the frequently requested information so that power consumed in retrieving the information from the producer is saved. This architecture fits seamlessly in the smart grid system with power being handled efficiently from all sources and even at the consumption end. The access control policies and the trust schema enables trusted and secure information exchange even in a highly distributed setup like that of the power-grids.

In this chapter, we explore the use of NDN in smart-grid communication [1] by showcasing the benefits of the architecture with the power consumption of a smart refrigerator as an example. Our contribution in this chapter is threefold: (a) introduce the use of NDN in smart grid communication, (b) design of specialized naming convention for the exchange of information, (c) identifying security challenges in conventional smart grid communication and provide NDN-based solutions.

6.2 NDN based Smart Power Distribution

Smart grids (SG) are a specialized and well-knit version of Cyber-Physical systems with digital communication being its lifeline. It is a system that is solely based on the communication of information and the other underlying technology related to the generation, delivery, and consumption of energy. The information exchange represents the buying and selling power from the grid to the use of sensors and actuators by the *independent service operators (ISOs)* to manage the power production, distribution, and consumption. Securing the information is thus essential and NDN [ZAB+14] as the prominent architecture of Information-Centric Networking

[1] The content of this chapter was published in IEEE CyberPels 2020 [RA20a]

(ICN) provides the benefits of end-to-end data security. In this section, we discuss the inherent benefits that using NDN based approach can provide to smart-grid communication.

6.2.1 Data-centric security

A smart grid is composed of numerous subsystems which continuously exchange updates and information that can be categorized as

- technical data corresponding to energy profiles
- management information related to billing, pricing, etc.

The are several stakeholders in the system who are interested in different aspects of this information and NDN being a client-driven approach allows them access to specific information using specialized interest packets. These interests follow the NDN stateful forwarding (optimal routes) till it reaches a node with the requested information. The breadcrumb trail is followed to reach the requestor. Thus information is exchanged in the system only when needed and prevents the injection of malicious information that can compromise the systems. It also ensures that only entities that are interested in the data are receiving the information from the grid thus ensuring full end-to-end security.

NDN mandates that every data packet be signed by the producer of the content and bound to the name during production allowing information to be verified and authenticated by any node at any time. Cryptographic keys which are used in encryption and verification are also fetched the same way as data packets. These keys can be a part of a chain of keys that follow the NDN trust schemas [YAC+15a] till a mutually trusted anchor is reached. The sign associated with the data packet also provides a way to be able to store the content in the cache of intermediate nodes

(in-network cache's) so that they can serve future requests for the same data without any compromise on the integrity of the data. Overall, the NDN security model allows for a decentralized and distributed way of secured information exchange with peers and other nodes in the network.

6.2.2 Naming

Names associated with the interest and data packet determine the information being requested and obtained. Application names are directly used across all the networking layers enabling devices with heterogeneous interfaces to obtain data over the said interface. The components in the name also define the granularity to which the communication/information is being sent or fetched.

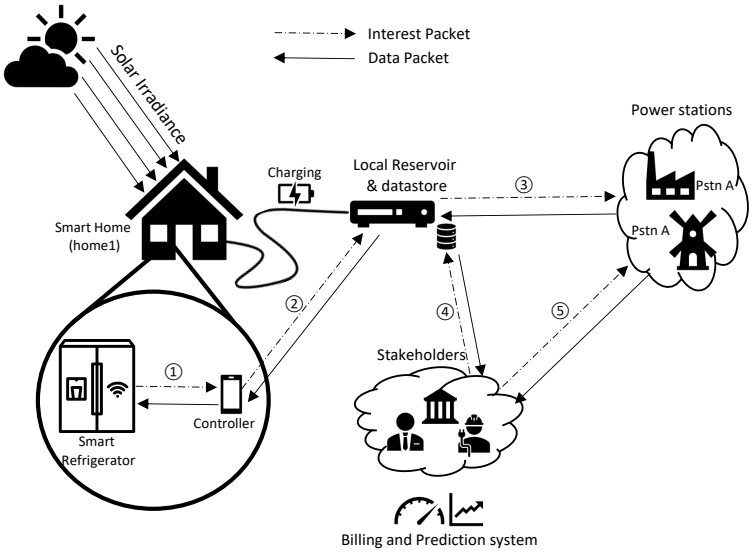

Figure 6.2: A typical scenario of a smart-home connected to the grid

To explain the need and benefits of NDN names, let us consider a smart-home with the Internet of Things (IoT) devices at the culmination end utilizing the requested and acquired power. Let the house also be deployed with solar panels that can generate power when there is adequate solar irradiance. Figure 6.2 depicts the

scenario wherein a smart refrigerator requires additional power to sustain the cooling within. The steps followed in the NDN-based approach is:

1. Based on the need for the compressor, the refrigerator requests the controller for 250 watts of power "`/pndn/home1/sg/refrigerator1/kitchen/250W`".
2. The controller, in turn, forwards this request to the solar power reserve "`/pndn /home1/sg/controller/reservoir/250W`"
3. The solar power reservoir, based on the availability, either sends the requested power and an acknowledgment or further requests the grid "`/pndn/home1/sg /pstnA/200W/1PM`" for the remaining power.
4. The various stakeholders (user/power company), can periodically request for the consumption report "`/pndn/home1/sg/owner/cooling_report/021720/1300`" of the house or each device and use this for the billing information.
5. The stakeholders can also retrieve historical information and use it for predictive analysis "`/pndn/FPL/sg/pstnB/022020/home1`" of the power requirement in the upcoming hour/day/month/year as required.

6.2.3 Routing and Forwarding

Routing and forwarding in NDN are solely based on names eliminating problems faced by the host-based IP networks. The Forwarding Strategy and Forwarding Information Base (FIB) use the longest prefix matching to identify the Multi-path routing are inherently supported by NDN which is missing in IP-based routing and plays a major role in preventing prefix-hijacking. The use of names for routing and the pull-based approach ensures that no targeted attacks on the grid nodes are possible. Even though the names can uniquely identify the information being

requested, the requestor information is opaque to the network, and with the data packet being signed, it allows for the packet to be cached in the router‘s buffer so that it can satisfy an impending request for the same data. Specialized forwarding strategies can be employed that benefit the smart-grid communication making NDN a versatile choice.

The next section describes the security primitives provided by the NDN architecture and how they can be used in ensuring vertical security for the smart grids.

6.3 Vertical Security using NDN

Smart Grids is a sophisticated architecture that is highly reliant on the underlying communication between the various involved entities and nodes and thus is faced with a plethora of security and privacy challenges. These challenges can be broadly categorized based on the need of the application as authentication, authorization, access-control, compromised keys, and certificates, etc. The smart grid system is built around different types of devices from small sensors to large power systems, billing, and financial systems, and hence there is always a risk of not being able to provide end-to-end security leading to the system being vulnerable to attacks and compromise. As an added complication, the interfaces and communication protocols used by the systems can vary drastically based on the capability of the device (IoT based sensors may use ZigBee, ZWave, Bluetooth Low Energy (BLE), etc. and large legacy power systems can use Wifi, cellular links, etc.) which makes the channel-based security prone to attacks. Also, the software and firmware used by the systems can largely influence the security design.

6.3.1 NDN Trust Schema

NDN by design as discussed provides inherent security features, such as data integrity and provenance. NDN's architecture also provides producer's trust assessment, through data signatures and the NDN trust schema.

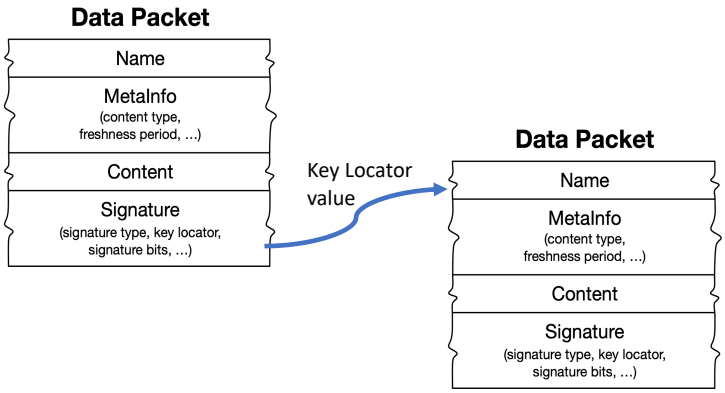

Figure 6.3: A typical NDN Data packet [YAC+15a]

Figure 6.3 depicts a typical NDN data packet. The fields showcase the name that is tightly bound to the content and the signature that is associated with the data. The other important field is the Key-locator that provides information of how the key that is used to sign this data packet can be retrieved and thus the trust chain similar to that in Web-of-Trust (WoT) can be followed to verify the authenticity till the mutually trusted "Trust Anchor" is reached. In the smart grid use-case, let us consider the Smart Grid deployed by vendor PowerCorp has a headquarters and multiple regional offices. Each of these regional offices is in turn responsible for power generation sites (called ProductionSite here) that serve the customers based on the requests received. The trust hierarchy follows the format where the main establishment which is the trust anchor signs the keys of the headquarters using "`/pndn/SG/powerCorp/KEY/...`". The headquarters in turn signs the keys for the regional offices using "`/pndn/SG/powerCorp/HQ/KEY/...`" and these offices sign the

keys of the power generators who finally sign the data they produce. Figure 6.4 describes this process. The smart home (or device in the house) on receiving the data can follow this hierarchical chain and continue to verify the trust anchor to validate the authenticity of the information.

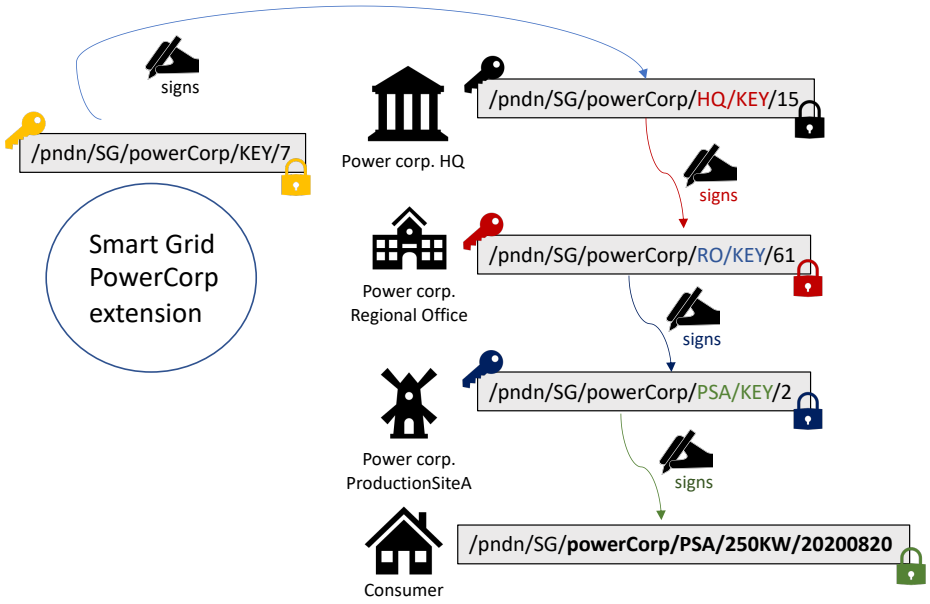

Figure 6.4: Authentication path in a Smart grid application

6.4 Summary

With the understanding of the data-centric benefits of NDN, robust smart grids can be designed and deployed. The paper explored the various advantages of secure communication from the end-user device to the main grid and the ease with which management information can be retrieved, shared, and used. In the future, we propose to extend this exploration and evaluate the complexity concerning time and space in deploying NDN-based smart-grid applications. We shall also build prototypes and Proofs-of-Concept that will be capable of implementing predictive models that can be used to efficiently manage and distribute power in a local and global

setting. The use of centralized repositories or cloud-based storage for ensuring data provenance also opens up security challenges solutions to which will be explored in detail. We are also interested to understand and utilize distributed ledger approaches with NDN for automated maintenance of distribution of power without the involvement of third parties to complete the transactions.

Chapter 7

Future Work

It is the hard reality that virtually all levels of IoT systems today are far from being perfectly safe from attacks. It is thus required for all the information and transactions performed as part of the system to be duly signed with appropriate certificates as advocated by the data-centric security of NDN. However, resource constraints at the device level result in the storage of certificates being a burden that most developers and devices do not welcome in their design. To address this issue, our ongoing research and future work explore the solutions to automate and secure provisioning of computing and storage resources to assist the IoT devices and improve the performance of the system on the whole. To the best of our knowledge, the proposed solution is the first attempt that tries to combine the freedom and flexibility of the data-centric architecture model with the best practices inspired and employed by novel technologies.

7.1 Automated IoT processing and storage provisioning

An IoT environment is a source for enormous amounts of data from the sensing operations performed by the plethora of sensory units. Studies conducted by Dave Evans [Eva11] and supported by Satyanarayanan et al. [SSX+15] predicted this information collection to explode beyond 1.6 zettabytes by 2020. However, most of this information is not directly in the form to be either used or stored for further use. A major concern and aspect that consumes an abundance of resources thus, is the preprocessing and storage requirements. The edge-computing capabilities provide a solution with offloading the computing to the more capable edge servers. We thus are now looking at a time where information flow changes from predominantly being "core-to-edge" to being "edge-to-core" [PAR+18].

In this direction of ongoing and future research, we will explore the data-centric ways advocated by NDN to facilitate the efficient provisioning of storage and processing capabilities. This will ultimately reduce stress on the cellular network links/edge access and provide better context on which information has to be stored or discarded. Mobility of the devices adds to the problem with intermittent connectivity hampering the offloading of information or carry-forward to the next location for continued processing. Provisioning storage efficiently is thus economically more feasible compared to laying more physical components which are not a viable option in devices that are designed to move. The local storage services (either in associated caches, cellular base stations, etc.) can be utilized as a buffer for IoT devices and user-generated data at the edge for either access or pre-processing before synchronization with the cloud.

Smart contracts (SC), is a new technology built to act as a virtual agreement between two entities while utilizing the benefits of an underlying distributed ledger. We intend to effectively utilize these SCs to provide a trusted way to provision storage and processing resources in the IoT environment. The use of SC also eliminates the need for middleware or an external agency to moderate resource availability and usage. SmartProv (the proposed concept), thus provides a set of legal constraints and policies based on SCs to get a consensus among involved parties by defining software code that is executed amid many peering eyes operating as verifiers.

SmartProv thus offers the following benefits:

- Inheritance of trust benefits of smart contracts in building decentralized access to the storage
- Flexibility to define software-based policies to automate resource availability and provisioning

- Orchestration of the benefits of pre-processing data and opportunistic caching that are feasible through the use of data-centric approaches.
- Delegation of processing activities thus reducing the burden on the individual devices.
- Improved performance of the system even in intermittent connectivity and highly reduced network overhead.

Chapter 8

Conclusion

In this dissertation, we have identified the challenges in IoT systems with a specific focus on securing communication. To alleviate the fundamental issues that arise because of the host-centric approach of the current internet architecture, we have employed the data-centric models advocate by Named Data Networking (NDN) which is a prominent architecture under Information-Centric Networking. Along with providing a conducive platform for designing robust IoT networks with the benefits that revolve around the direct use of application names, support for in-network caching, use of multiple interfaces for communication, etc., NDN also offers data-centric security primitives.

In the works described in this dissertation, we have identified novel approaches that can enhance the NDN-enabled IoT networks in being able to automate and securely bootstrap devices, enable transient trust establishment for initial information exchange among entities with minimal or no prior interactions. We have also designed an attribute-based signature scheme with the first comprehensive prototype that can reduce the consumer's dependence on the network for verifying the authenticity of the received data while also providing conditional and tunable privacy to the content producer using elegant policies. To further facilitate the devices (communicating entities) to reduce the need for storage of crypto-material and even generation of certification requests, we propose CertCoalesce for the retrieval of multiple simultaneously valid certificates that can be generated for shot validity periods using a single request. This design also alleviates the issues caused by faulty or incomplete Certificate Revocation Lists (CRLs). Further, it is important to give the correct access control authorization and permissions to consumers for any information, and to enable this we have introduced a name-based access control technique

and extended it to use attribute-based encryption techniques for better flexibility and scalability.

The objectives of the various proposed techniques have been to reduce the overall network overhead and dependence of the devices on the network while being able to provide tunable and conditional anonymity to content producers. The various aspects of NDN architecture including the naming conventions are used in realizing automated bootstrapping techniques, authentication techniques, and access control mechanisms. We also discussed various application scenarios including the bootstrapping of hidden devices, vehicular networks to the end-to-end security of network and the firmware as well as the smart-grid examples. The future work described in this dissertation will introduce a distributed ledger approach to automate and manage resource allocation and use which along with the other objectives realized will aid in the realization of a truly connected ecosystem that is vertically secured.

BIBLIOGRAPHY

[ABR+18] Alex Afanasyev, Jeff Burke, Tamer Refaei, Lan Wang, Beichuan Zhang, and Lixia Zhang. A brief introduction to Named Data Networking. In *Proc. of MILCOM*, 2018.

[Agr] Shashank Agrawal. Ciphertext-policy attribute-based encryption. Online: `https://github.com/sagrawal87/ABE/blob/master/bsw07.py`. Last accessed on Aug 22, 2019.

[All] ZigBee Alliance.

[AMM+13] Alexander Afanasyev, Priya Mahadevan, Ilya Moiseenko, Ersin Uzun, and Lixia Zhang. Interest flooding attack and countermeasures in named data networking. In *2013 IFIP Networking Conference*, pages 1–9. IEEE, 2013.

[ANK+15] Tohru Asami, Byambajav Namsraijav, Yoshihiko Kawahara, Kohei Sugiyama, Atsushi Tagami, Tomohiko Yagyu, Kenichi Nakamura, and Toru Hasegawa. Moderator-controlled information sharing by identity-based aggregate signatures for information centric networking. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, pages 157–166, 2015.

[AS16] S Abhishek Anand and Nitesh Saxena. Vibreaker: Securing vibrational pairing with deliberate acoustic noise. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 103–108, 2016.

[ASWS15] Samir Al-Sheikh, Matthias Wählisch, and Thomas C Schmidt. Revisiting countermeasures against ndn interest flooding. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, pages 195–196. ACM, 2015.

[Ban] Adeola Bannis. Named data network internet of things toolkit (ndn-iott).

[BFL96] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, pages 164–173. IEEE, 1996.

[BGK08] A. Boldyreva, V. Goyal, and V. Kumar. Identity-based encryption with efficient revocation. In *Proc. of ACM Conference on Computer and Communications Security*, 2008.

[BH19] Benedikt Brecht and Thorsten Hehn. A security credential management system for v2x communications. In *Connected Vehicles*, pages 83–115. Springer, 2019.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Proc. of IEEE Symposium on Security and Privacy*, 2007.

[CAM] Security credential management system proof–of–concept.

[CCD16] Alberto Compagno, Mauro Conti, and Ralph Droms. Onboardicng: a secure protocol for on-boarding iot devices in icn. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, pages 166–175, 2016.

[CCGT13] Alberto Compagno, Mauro Conti, Paolo Gasti, and Gene Tsudik. Poseidon: Mitigating interest flooding ddos attacks in named data networking. In *38th annual IEEE conference on local computer networks*, pages 630–638. IEEE, 2013.

[CZWS12] Dan Cao, Baokang Zhao, Xiaofeng Wang, and Jinshu Su. Flexible multi-authority attribute-based signature schemes for expressive policy. *Mobile Information Systems*, 8(3), 2012.

[DLVZH09] Alexander De Luca, Emanuel Von Zezschwitz, and Heinrich Hußmann. Vibrapass: secure authentication based on shared lies. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 913–916, 2009.

[DR08] T. Dierks and E. Rescorla. The transport layer security (tls) protocol version 1.2. RFC 5246, RFC Editor, August 2008.

[DWFL13] Huichen Dai, Yi Wang, Jindou Fan, and Bin Liu. Mitigate ddos attacks in ndn by interest traceback. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pages 381–386. Citeseer, 2013.

[Eva11] Dave Evans. The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper*, 1(2011):1–11, 2011.

[FC05] Michal Feldman and John Chuang. The evolution of cooperation under cheap pseudonyms. In *E-Commerce Technology, 2005. CEC 2005. Seventh IEEE International Conference on*, pages 284–291. IEEE, 2005.

[FGK03] S. Frankel, R. Glenn, and S. Kelly. The aes-cbc cipher algorithm and its use with ipsec. RFC 3602, RFC Editor, September 2003.

[GG16] Steven D Galbraith and Pierrick Gaudry. Recent progress on the elliptic curve discrete logarithm problem. *Designs, Codes and Cryptography*, 78(1):51–72, 2016.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. of ACM Conference on Computer and Communications Security*, 2006.

[GPW+13] Giulio Grassi, Davide Pesavento, Lucas Wang, Giovanni Pau, Rama Vuyyuru, Ryuji Wakikawa, and Lixia Zhang. Acm hotmobile 2013 poster: vehicular inter-networking via named data. In *Proc. of ACM SIGMOBILE Mobile Computing and Communications Review*, volume 17, 2013.

[GTUZ13] Paolo Gasti, Gene Tsudik, Ersin Uzun, and Lixia Zhang. Dos and ddos in named data networking. In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7. IEEE, 2013.

[Har12] William Harwood. *The logic of trust*. PhD thesis, University of York, 2012.

[IZS13] Mihaela Ion, Jianqing Zhang, and Eve M Schooler. Toward content-centric privacy in icn: Attribute-based encryption and routing. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pages 39–40, 2013.

[JB] Brent Waters John Bethencourt, Amit Sahai. Ciphertext-policy attribute-based encryption. Online: `http://acsc.cs.utexas.edu/cpabe/tutorial.html`. Last accessed on Aug 22, 2019.

[JMB11] Sonia Jahid, Prateek Mittal, and Nikita Borisov. Easier: Encryption-based access control in social networks with efficient revocation. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 411–415. ACM, 2011.

[JST+09] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12, 2009.

[JSZ+15] Rong Jin, Liu Shi, Kai Zeng, Amit Pande, and Prasant Mohapatra. Magpairing: Pairing smartphones in close proximity using magnetometers. *IEEE Transactions on Information Forensics and Security*, 11(6):1306–1320, 2015.

[KCW+14] Konstantinos V Katsaros, Wei Koong Chai, Ning Wang, George Pavlou, Herman Bontius, and Mario Paolone. Information-centric networking for machine-to-machine data delivery: a case study in smart grid applications. *IEEE Network*, 28(3):58–64, 2014.

[KFR09] Ronald Kainda, Ivan Flechais, and AW Roscoe. Usability and security of out-of-band channels in secure device pairing protocols. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–12, 2009.

[KK18] Sarmadullah Khan and Rafiullah Khan. Multiple authorities attribute-based verification mechanism for blockchain mircogrid transactions. *Energies*, 11(5), 2018.

[KLR+15] Younghyun Kim, Woo Suk Lee, Vijay Raghunathan, Niraj K Jha, and Anand Raghunathan. Vibration-based secure side channel for medical devices. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2015.

[KPČ16] Tonko Kovačević, Toni Perković, and Mario Čagalj. Flashing displays: user-friendly solution for bootstrapping secure associations between multiple constrained wireless devices. *Security and Communication Networks*, 9(10):1050–1071, 2016.

[KRD06] Sunil Kumar, Vineet S Raghavan, and Jing Deng. Medium access control protocols for ad hoc wireless networks: A survey. *Ad hoc networks*, 4(3):326–358, 2006.

[KS05] S. Kent and K. Seo. Security architecture for the internet protocol. RFC 4301, RFC Editor, December 2005.

[LAS+10] Jin Li, Man Ho Au, Willy Susilo, Dongqing Xie, and Kui Ren. Attribute-based signature and its applications. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 60–69, 2010.

[LRRK18] Kyuin Lee, Vijay Raghunathan, Anand Raghunathan, and Younghyun Kim. Syncvibe: Fast and secure device pairing through physical vibration on commodity smartphones. In *2018 IEEE 36th International Conference on Computer Design (ICCD)*, pages 234–241. IEEE, 2018.

[LW11] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In *Proc. of Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2011.

[LZW+19] Yanbiao Li, Zhiyi Zhang, Xin Wang, Edward Lu, Dafang Zhang, and Lixia Zhang. A secure sign-on protocol for smart homes over named data networking. *IEEE Communications Magazine*, 57(7):62–68, 2019.

[MB09] Zaki Malik and Athman Bouguettaya. Reputation bootstrapping for trust establishment among web services. *IEEE Internet Computing*, (1):40–47, 2009.

[MBO16] Adeel Mohammad Malik, Joakim Borgh, and Börje Ohlman. Attribute-based encryption on a resource constrained sensor in an information-centric network. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, pages 217–218, 2016.

[MCHM13] Ruofei Ma, Hsiao-Hwa Chen, Yu-Ren Huang, and Weixiao Meng. Smart grid communication: Its challenges and opportunities. *IEEE transactions on Smart Grid*, 4(1):36–46, 2013.

[MG07] Rene Mayrhofer and Hans Gellersen. On the security of ultrasound as out-of-band channel. In *2007 IEEE International Parallel and Distributed Processing Symposium*, pages 1–6. IEEE, 2007.

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

[MPR11] Hemanta K Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures. In *Cryptographers' track at the RSA conference*, pages 376–392. Springer, 2011.

[MRL13] Dhiraj Murthy, Atilano Rodriguez, and Jeremy Lewis. Examining the formation of Swift Trust within a scientific global virtual team. In *Proc. of International Conference on System Sciences (HICSS)*, 2013.

[MTM18] T. Mick, R. Tourani, and S. Misra. LASeR: Lightweight authentication and secured routing for NDN IoT in smart cities. *IEEE Internet of Things Journal*, 5(2), April 2018.

[ndna]

[NDNb] NDN Team. NDN packet format specification. Online: `https://named-data.net/doc/NDN-packet-spec/0.3/`. Last accessed on Aug 22, 2019.

[NDN20] NDN Team. NDN Certificate Format Version 2.0. Online: `http://named-data.net/doc/ndn-cxx/current/specs/certificate-format.html`, Last accessed, June 5, 2020.

[Nor16] Amy Nordrum. Popular internet of things forecast of 50 billion devices by 2020 is outdated, 08 2016.

[PAR+18] Ioannis Psaras, Onur Ascigil, Sergi Rene, George Pavlou, Alex Afanasyev, and Lixia Zhang. Mobile data repositories at the edge. In {*USENIX*} *Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[PTMW10] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure attribute-based systems. *Journal of Computer Security*, 18(5), 2010.

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

[RA20a] Sanjeev Kaushik Ramani and Alex Afanasyev. On using ndn to vertically secure smart power distribution. In *2020 IEEE CyberPELS (CyberPELS)*, pages 1–6. IEEE, 2020.

[RA20b] Sanjeev Kaushik Ramani and Alex Afanasyev. Rapid establishment of transient trust for ndn-based vehicular networks. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE, 2020.

[RA20c] Sanjeev Kaushik Ramani and Alexander Afanasyev. Certcoalesce: Efficient certificate pool for ndn-based systems. In *Proceedings of the 7th ACM Conference on Information-Centric Networking*, pages 158–160, 2020.

[RDH09] Lionel P Robert, Alan R Denis, and Yu-Ting Caisy Hung. Individual swift trust and knowledge-based trust in face-to-face and virtual team members. *Journal of Management Information Systems*, 26(2):241–279, 2009.

[RI17a] Sanjeev Kaushik Ramani and SS Iyengar. Evolution of sensors leading to smart objects and security issues in iot. In *International Symposium on Sensor Networks, Systems and Security*, pages 125–136. Springer, 2017.

[RI17b] Sanjeev Kaushik Ramani and SS Iyengar. Evolution of sensors leading to smart objects and security issues in iot. In *International Symposium on Sensor Networks, Systems and Security*, pages 125–136. Springer, 2017.

[RI17c] Sanjeev Kaushik Ramani and SS Iyengar. Evolution of sensors leading to smart objects and security issues in iot. In *International Symposium on Sensor Networks, Systems and Security*, pages 125–136. Springer, 2017.

[RP09] Antony Rowstron and Giovanni Pau. Characteristics of a vehicular network. *University of California Los Angeles, Computer Science Department, Tech. Rep*, pages 09–0017, 2009.

[RPA20] Sanjeev Kaushik Ramani, Proyash Podder, and Alex Afanasyev. Ndnviber: Vibration-assisted automated bootstrapping of iot devices. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE, 2020.

[RTT+19] Sanjeev Kaushik Ramani, Reza Tourani, George Torres, Satyajayant Misra, and Alexander Afanasyev. Ndn-abs: Attribute-based signature

scheme for named data networking. In *Proceedings of the 6th ACM Conference on Information-Centric Networking*, pages 123–133, 2019.

[SA] Melissa Chase Shashank Agrawal. FAME: Fast attribute-based message encryption. Online: `https://github.com/sagrawal87/ABE/blob/master/ac17.py`. Last accessed on Aug 22, 2019.

[SBL+16] Wentao Shang, Adeola Bannis, Teng Liang, Zhehao Wang, Yingdi Yu, Alexander Afanasyev, Jeff Thompson, Jeff Burke, Beichuan Zhang, and Lixia Zhang. Named data networking of things. In *2016 IEEE first international conference on internet-of-things design and implementation (IoTDI)*, pages 117–128. IEEE, 2016.

[SDM+14] Wentao Shang, Qiuhan Ding, Alessandro Marianantoni, Jeff Burke, and Lixia Zhang. Securing building management systems using named data networking. *IEEE Network*, 28(3):50–56, 2014.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of Workshop on the Theory and Application of Cryptographic Techniques*, 1984.

[Shi07] R. Shirey. Internet security glossary, version 2. RFC 4949, RFC Editor, August 2007.

[SSKR] Henrik Snellman, Mikko Savolainen, Jere Knaappila, and Pasi Rahikkala. Bluetooth® 5, refined for the iot.

[SSX+15] Mahadev Satyanarayanan, Pieter Simoens, Yu Xiao, Padmanabhan Pillai, Zhuo Chen, Kiryong Ha, Wenlu Hu, and Brandon Amos. Edge analytics in the internet of things. *IEEE Pervasive Computing*, 14(2):24–31, 2015.

[STU07] Claudio Soriente, Gene Tsudik, and Ersin Uzun. Beda: Button-enabled device association. 2007.

[STU08] Claudio Soriente, Gene Tsudik, and Ersin Uzun. Hapadep: human-assisted pure audio device pairing. In *International Conference on Information Security*, pages 385–400. Springer, 2008.

[SUVA11] Nitesh Saxena, Md Borhan Uddin, Jonathan Voris, and N Asokan. Vibrate-to-unlock: Mobile phone assisted user authentication to multiple personal rfid tags. In *2011 IEEE International Conference on*

Pervasive Computing and Communications (PerCom), pages 181–188. IEEE, 2011.

[Swa08] Gayatri Swamynathan. *Towards reliable reputations for distributed applications*. University of California at Santa Barbara, 2008.

[SWA+17] Wentao Shang, Zhehao Wang, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. Breaking out of the cloud: Local trust management and rendezvous in named data networking of things. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, pages 3–13, 2017.

[TBES16] Eleftherios Tsampasis, Dimitrios Bargiotas, Charalambos Elias, and Lambros Sarakis. Communication challenges in smart grid. In *MATEC Web of Conferences*, volume 41, page 01004. EDP Sciences, 2016.

[vib]

[Wan] Junwei Wang. Java realization for ciphertext-policy attribute-based encryption. Online: `https://github.com/junwei-wang/cpabe/`. Last accessed on Aug 22, 2019.

[Wi-18] Wi-Fi Alliance. Device provisioning protocol specification v1.1, 2018.

[YAC+15a] Yingdi Yu, Alexander Afanasyev, David Clark, KC Claffy, Van Jacobson, and Lixia Zhang. Schematizing trust in named data networking. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, pages 177–186, 2015.

[YAC+15b] Yingdi Yu, Alexander Afanasyev, David Clark, Van Jacobson, and Lixia Zhang. Schematizing trust in Named Data Networking. In *Proc. of ACM Conference on Information-Centric Networking*, 2015.

[YAC+15c] Yingdi Yu, Alexander Afanasyev, David Clark, kc claffy, Van Jacobson, and Lixia Zhang. Schematizing trust in Named Data Networking. In *Proceedings of 2nd ACM Conference on Information-Centric Networking*, September 2015.

[ZAB+14] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, KC Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and

Beichuan Zhang. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.

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

[ZVL+18] Zhiyi Zhang, Vishrant Vasavada, Jonathan Lin, Reddy Siva Kesava K, Peter Reiher, and Lixia Zhang. Producer-assisted pushback. Technical report, Technical Report NDN-0065, NDN, 2018.

[ZYAZ17a] Zhiyi Zhang, Yingdi Yu, Alex Afanasyev, and Lixia Zhang. Ndn certificate management protocol (ndncert). *NDN, Technical Report NDN-0050*, 2017.

[ZYAZ17b] Zhiyi Zhang, Yingdi Yu, Alex Afanasyev, and Lixia Zhang. NDN certificate management protocol (NDNCERT). Technical Report NDN-0050, NDN, April 2017.

[ZYR+18] Zhiyi Zhang, Yingdi Yu, Sanjeev Kaushik Ramani, Alex Afanasyev, and Lixia Zhang. Nac: Automating access control via named data. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 626–633. IEEE, 2018.

VITA

SANJEEV KAUSHIK RAMANI

January 3, 1993	Born, Bangalore, Karnataka, India
2014	Bachelors of Technology Amrita Vishwa Vidyapeetham India
2020	M.S., Computer Science Florida International University Miami, Florida
2016-2021	Ph.D., Computer Science Florida International University Miami, Florida

PUBLICATIONS AND PRESENTATIONS

Ramani, Sanjeev Kaushik, and Alexander Afanasyev., (2020) *On Using NDN to Vertically Secure Smart Power Distribution* IEEE CyberPELS (CyberPELS), pp. 1-6. IEEE

Ramani, Sanjeev Kaushik, and Alexander Afanasyev., (2020) *CertCoalesce: Efficient Certificate Pool for NDN-Based Systems* In Proceedings of the 7th ACM Conference on Information-Centric Networking, pp. 158-160. ACM

Ramani, Sanjeev Kaushik, Proyash Podder, and Alex Afanasyev., (2020) *NDNViber: Vibration-Assisted Automated Bootstrapping of IoT Devices* IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1-6. IEEE

Afanasyev, Alex, and Sanjeev Kaushik Ramani., (2020) *NDNconf: Network Management Framework for Named Data Networking* IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1-6. IEEE

Ramani, Sanjeev Kaushik, and Alex Afanasyev., (2020) *Rapid Establishment of Transient Trust for NDN-Based Vehicular Networks* IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1-6. IEEE

Pal, Pathikrit, G. S. Thejas, Sanjeev Kaushik Ramani, S. S. Iyengar, and N. R. Sunitha., (2019) *A Variation in the Working of Playfair Cipher* 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), vol. 4, pp. 1-7. IEEE

Iyengar, Sitharama S., Sanjeev Kaushik Ramani, and Buke Ao., (2019) *Fusion of the Brooks–Iyengar Algorithm and Blockchain in Decentralization of the Data-Source* Journal of Sensor and Actuator Networks 8, no. 1

Ramani, Sanjeev Kaushik, Reza Tourani, George Torres, Satyajayant Misra, and Alexander Afanasyev., (2019) *NDN-ABS: Attribute-Based Signature Scheme for Named Data Networking* Proceedings of ACM Conference on Information Centric Networking. ACM

Zhang, Zhiyi, Yingdi Yu, Sanjeev Kaushik Ramani, Alex Afanasyev, and Lixia Zhang., (2018) *NAC: Automating access control via Named Data* In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM), pp. 626-633. IEEE

Ramani, Sanjeev Kaushik, and S. S. Iyengar., (2017) *Evolution of Sensors Leading to Smart Objects and Security Issues in IoT* In International Symposium on Sensor Networks, Systems and Security, pp. 125-136. Springer, Cham

Archana, S., G. S. Thejas, Sanjeev Kaushik Ramani, and S. S. Iyengar., (2017) *Image Processing Approaches for Autonomous Navigation of Terrestrial Vehicles in Low Illumination* In 2017 2nd International Conference On Emerging Computation and Information Technologies (ICECIT), pp. 1-6. IEEE

Meda, Nidhi S., Thejas Gubbi Sadashiva, Sanjeev Kaushik Ramani, and S. S. Iyengar., (2017) *Mobile WSN Testbed for Agriculture: Plant Monitoring System* In 2017 2nd International Conference On Emerging Computation and Information Technologies (ICECIT), pp. 1-6. IEEE

Mithil, K. M., G. S. Thejas, Sanjeev Kaushik Ramani, and S. S. Iyengar., (2017) *A Low Cost Multi Sensorial Data Fusion for High Speed Obstacle Avoidance Using 3-D Point Clouds and Image Processing in Self Balancing Robots* In 2017 2nd International Conference On Emerging Computation and Information Technologies (ICECIT), pp. 1-6. IEEE

Nagaraj, Kushal, Thejas Gubbi Sadashiva, Sanjeev Kaushik Ramani, and S. S. Iyengar., (2017) *Image Feature Based Smoke Recognition in Mines Using Monocular Camera Mounted on Aerial Vehicles* In 2017 2nd International Conference On Emerging Computation and Information Technologies (ICECIT), pp. 1-6. IEEE