12-10-2020

Secure Sharing of Spatio-Temporal Data through Name-based Access Control

Laqin Fan

SECURE SHARING OF SPATIO-TEMPORAL DATA THROUGH NAME-BASED ACCESS CONTROL

by

Laqin Fan

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Computer Science

The University of Memphis

December 2020

Abstract

Named Data Networking (NDN) is proposed as a future Internet architecture, which provides name-based data publishing and fetching primitive. Compared to TCP/IP, the benefits of NDN are as follows. NDN removes the need to manage IP address; NDN provides semantically meaningful and structured names; NDN has a stateful and name-based forwarding plane; NDN supports data-centric security and in-network caching. Name-based Access Control is an access control solution proposed over NDN, which is a content-based access control by encrypting data at the time of production directly without relying on a third-party service(i.e., Cloud storage), utilizes NDN's hierarchical naming convention to express access control policy, and enables automation of key distribution.

As more and more mobile data (e.g., mobile-health data) are generated dynamically and continuously over time and space, data owners often want to share his data with others for data analysis or healthcare, etc. To protect their privacy, they may want to share a subset of data based on their requirements with time and/or space restrictions. An effective and secure access control solution is required to ensure only authorized users can access certain data with fine granularity. Inspired by Named-based Access Control scheme, we take into account the data attributes (time, location) to make access decisions. In this work, we introduce a spatio-temporal access control scheme that allows data owners to specify access control policy and limit data access to a given time interval and/or location area. Specifically, we design a hierarchically structured naming convention to express fine-grained access control policy on spatio-temporal data, we realize a publish-subscribe functionality based on PSync for real-time data stream sharing, we develop a practical spatial-temporal data access control prototype based on NDN codebase. Moreover, we run experiments using Mini-NDN to evaluate the performance of sharing historical data from storage and sharing data in real time.

TABLE OF CONTENTS

List of Figures iv

List of Tables vi

1 Introduction 1

2 Background and Related Work 3
- 2.1 Existing Access Control Model for Sensitive Data 3
- 2.2 Named Data Networking 5
- 2.3 Name-based Access Control 7

3 Design 10
- 3.1 Namespace Design 11
  - 3.1.1 Data Namespace 12
  - 3.1.2 Access Control Namespace 13
- 3.2 Granular Access Control 15
  - 3.2.1 Access Control Policy Specification 15
  - 3.2.2 Temporal Data Access Control 17
  - 3.2.3 Spatial Data Access Control 18
  - 3.2.4 Spatio-Temporal Data Access Control 20
  - 3.2.5 Access Control Policy Enforcement 21
- 3.3 Granular Content Key 25
- 3.4 Access Revocation 27

4 Implementation 29
- 4.1 Class Diagram 29
- 4.2 Real-Time Data Sharing 31

5 Evaluation 33
- 5.1 Security Analysis 33
- 5.2 Performance Analysis 33
  - 5.2.1 Experiments 34
  - 5.2.2 Evaluation Results 35

6 Discussion and Future Work 40

7 Conclusion 41

References 42

LIST OF FIGURES

Figure **Page**

1. NDN Interest and Data Packets 6

2. Forwarding Process at NDN node 7

3. NAC Scheme 9

4. An example of mHealth data sharing 10

5. An example of user data namespace with access control credential namespace 12

6. An example of spatial bound of fitness center 16

7. Temporal Access Control Policy 17

8. Temporal Access Control Naming Rules 17

9. KEKs for Different Dates 18

10. A Sequence of KEKs with Time Intervals 19

11. Spatial Access Control Polic 19

12. Spatial Access Control Naming Rules 19

13. Spatio-Temporal Access Policy 20

14. Spatio-Temporal Access Control Naming Rules 20

15. KEK and KDK Names with Time Interval and Location 21

16. Key Relationship in NAC 22

17. Data Access for Authorized Consumer 22

18. Data Access for Unauthorized Consumer 23

19. An example of access policy 23

20. Key Naming Convention of Consumption Credential 24

21. Data Naming Convention 25

22. CK Naming Convention 25

23. CK Data Naming Convention 25

24. An Example of CK Granularity 26

25. An Example of Access Revocation and KEK Updates 27

26. Class Diagram for Spatio-Temporal Access Control Code Base 30

27. Pub-Sub Workflow in Spatio-Temporal Access Control 32

28. An Example of Mini-NDN topology 34

29. Total Number of CK for Different Granularity 36

30. Sync Delay for Different Granularity 37

31. Data Production Time for Different Granularity 37

32. Data consumption Time for Different Granularity 38

33. Communication Delay for Different Granularity 39

34. Data Retrieval Time for Different Granularity 39

LIST OF TABLES

Table **Page**

1. Notations 8

Chapter 1

Introduction

Nowadays, an enormous amount of spatio-temporal data are collected and stored in different areas, for example, modern smart home systems utilize sensors to collect data and provide services; Mobile health apps and devices generate large amounts of health data for healthcare. This data may contain personal privacy-sensitive information, e.g., activity, medical conditions, and data owners desire to restrict access to their data to ensure only authorized users can have access rights. Therefore, fine-grained access control is required. There already exists solutions proposed in [1], [2] and [3], they are all based on attribute-based encryption for data confidentiality and fine-grained access control. However, they all rely on third-party services to store and share data, and have limitations, e.g., the size of ciphertext increases linearly based on the number of incorporated attributes. There is little attention to controlling the level of data sharing based on when and/or where the data is produced, and no user-controlled data access with end-to-end confidentiality.

Named Data Networking (NDN) [4] is a future Internet architecture, which transfers the communication model from delivering data by identifying destination through IP address to fetching data by giving the data name, and provides data-centric security through securing the data directly instead of securing the channel where the data is transmitted. NDN uses Interest/Data exchange model, data consumer can send an Interest packet containing the name of the desired data to the network, any node in the network has the requested data will return a Data packet with a cryptographic signature along the reverse path back to data consumer. NDN has developed a set of security mechanisms [5], which contains current solutions to data authenticity, confidentiality, and availability, introduces a secure bootstrapping process and access control. Name-based Access Control (NAC) [6] is proposed to provide content-based access control over NDN by leveraging NDN's capabilities.

In order to develop a user-controlled access control with a fine-grained policy and allow users to decide who can access what data at which granularity over time and/or location. we leverage NAC scheme to have end-to-end confidentiality and enable automatic key distribution. In this thesis, we propose a spatio-temporal access control with fine granularity on data attributes, time and/or location. We design a structured naming convention to express fine-grained access control policy for spatio-temporal data, develop a practical spatio-temporal data access control prototype over NDN to support data sharing from storage and data sharing in real time. Finally, we evaluate the performance using Mini-NDN.

The organization of this thesis is as follows. In Chapter 2, we introduce the background briefly, including the existing access control works for fine-grained access control and Named Data Networking. In Chapter 3, we describe how Name-based Access Control works, and introduce PSync protocol. The details of our design are presented in Chapter 4. In Chapter 5, we describe the implementation of our spatio-temporal prototype. Chapter 6 includes the security analysis and the evaluation results. Chapter 7 shows the discussion of current work with privacy concerns and some ideas for future work. In Chapter 8, we summarize the work of this thesis.

Chapter 2

Background and Related Work

2.1 Existing Access Control Model for Sensitive Data

The purpose of Access Control [7] is mediating every request to data and resources maintained by a system, and determining if the request should be granted or denied. Access control can constrain what a user can do directly to the data with a security policy. The policy is authored by data owner to define what should or should not be allowed in an expressive way. Therefore, with access control, data owner could be able to protect the data and information from being disclosed to unauthorized users, and provide data availability to authorized users.

Recently, as more sensitive data is generated and shared, deploying access control techniques for secure data sharing is needed. There are numerous access control solutions proposed, e.g., role-based access control (RBAC) [8], and attribute-based access control, see [1], [2] and [3]. However, they all rely on a third-party trusted services on the Internet to store and share the data.

In paper [8], Sandhu et al. proposed Role-Based Access Control framework, in which permissions are associated with roles and users are associated with roles. There are four basic components: a set of users, a set of roles, a set of permissions and a set of sessions. A user can be a member in an organization, while a role is a title that has a set of permissions to perform some functions in the organization. A permission can be regarded as the right to access a system or an object, where a session is a mapping between a user and multiple roles. In RBAC, a central security administrator is needed to grant or delete access permission. The task of the administrator is to grant and revoke the permissions to the set of particular roles. Once a permission of a role is built within an organization, this permission remains constant or changes slowly. When a new member joins the organization, he/she will be granted permission to an existing role. When a person's functions change, his/her permission of the existing role will be deleted, and a set of permission will be granted. Once the person leaves the organization, all the permission of his roles will be deleted.

The method introduced in [1] supports fine-grained sharing of encrypted data using Key-Policy Attribute-Based Encryption (KP-ABE), with this method, each ciphertext is labeled with a set of descriptive attributes, and each user's private key is associated with an access policy. Specifically, the core algorithm of KP-ABE takes a security parameter as input, and generates a public key and a master key. The encryptor uses the public key for data encryption, and master key is issued by authority to users securely and used for users to generate secret key for data decryption. The inputs of data encryption algorithm include the data, public key, and a set of attributes, while the users use key generation algorithm with access policy and master key as inputs to output a secret key. Then, data decryption algorithm takes ciphertext, secret key as input, and outputs the decrypted data if the set of attributes satisfy the access policy. Through KP-ABE, data is stored in an encrypted form on the Internet by a third-party, such as personal email, data stored on web portal sites Google and Yahoo. Different users are allowed to retrieve and decrypt different pieces of data based on each access policy. The limitation of KP-ABE is the encryptor has no control over who is granted the access to the encrypted data, and he/she can only decide the attribute set for the data, but has to trust some key-issuer to grant access to users.

Paper [2] presents the first construction of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) for access control on encrypted data stored in a trusted server. Unlike KP-ABE, CP-ABE utilizes sets of descriptive attributes to combine with each user's private key, and ciphertext is associated with an access policy. While a third-party server encrypts data using CP-ABE, an associated access policy will be specified. If a user who possesses attributes which satisfy the access policy should be able to decrypt the ciphertext. To be specific, CP-ABE algorithm takes a security parameter as input and outputs a public key and a master key. The data encryption algorithm takes the public key, the data and access policy as input to output the ciphertext. The key generation algorithm generates a secret key for users with master key and a set of attributes associated with the user. Then the user can use the secret key to decrypt the ciphertext if the set of attributes satisfy with the access policy. CP-ABE is conceptually more like traditional role-based access control. Data encryptor can decide who is granted or denied access to the data, but KP-ABE only allows data encryptor to choose the descriptive attributes for the data he/she encrypts, and trusts a key issuer to issue private keys to users for data decryption. Meanwhile, the size of ciphertext in both CP-ABE and KP-ABE increases linearly based on the number of incorporated attributes.

EASiER [3] also employs attribute-based encryption to mitigate access control on data in Online Social Networks (OSNs). Users can define different groups with different attributes and assign keys for the groups to access their data on social networks, such as posts, profiles, etc. Since the groups are dynamic, the attributes need to be changed all the time, this results in key re-generation and data re-encryption frequently. EASiER presents a method for revocation of attributes and users by using a trusted proxy. The proxy updates a new key for each revocation. In their paper, a centralized OSN plays the role of the proxy.

2.2 Named Data Networking

Named Data Networking [4] is proposed as a future Internet architecture, which shifts Internet's communication model from retrieving data by its IP address to fetching data by a name regardless of its address. NDN provides data-centric security to secure the data directly through signing each data packet without relying on security channel where the data is transmitted. The name-based data distribution and data-centric security in NDN make it possible to implement end-to-end confidentiality for data sharing. As follows, we will introduce how the request and reply work in NDN and how trust model works to secure the data directly.

In NDN, the request and reply work at the network layer, an Interest packet as a

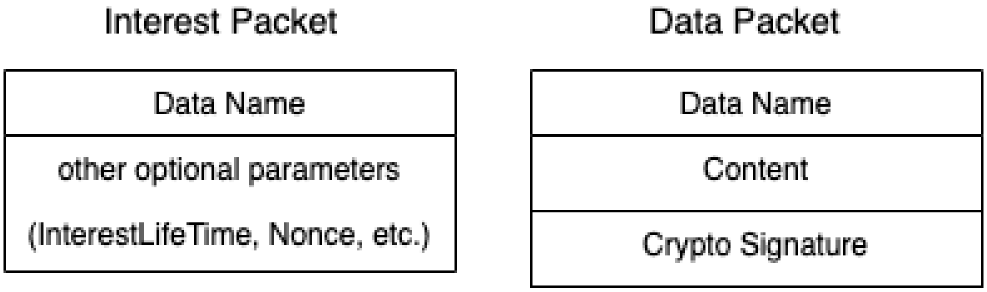


Figure 1. NDN Interest and Data Packets

request contains the requested data name, and fetches a Data packet as a reply back from the original data producer, or in-network cache, see Figure 1.. A consumer sends an Interest with his/her desired data name to the network, NDN routers forward this Interest to the data producer or anyone who has the data. Once a node has the data, the node will conduct a Data packet containing a cryptographic signature. Then, this Data packet follows the reverse path of the Interest back to the consumer. This entire process is enabled by NDN forwarder (Figure 2.). Each NDN router deploys an NDN forwarder which determines whether, where and when to forward the Interest packet. The forwarder contains three basic components, a Content Store (CS), a Pending Interest Table (PIT), and a Forwarding Information Base (FIB). CS is utilized to store the previously retrieved data packets, upon receiving an Interest packet, the forwarder checks if the data in CS satisfies the Interest. If there exists such data matches the Interest, the forwarder returns the data, otherwise, looks up Interest name in PIT. If an existing entry matches the Interest name, the forwarder increases the Interest lifetime as well as records its incoming interface to the entry. If there is no such entry satisfies the Interest, a new entry is created in PIT. To fetch the desired data, that is not cached in CS, the NDN forwarder will forward the Interest to data producer based on the information in FIB as well as the Forwarding Strategy. The Forwarding Strategy could decide to drop the Interest packet when all upstream links are congested, and in NFD [9], the default strategy is BestRoute, which can forward the interest to the route with the lowest cost.

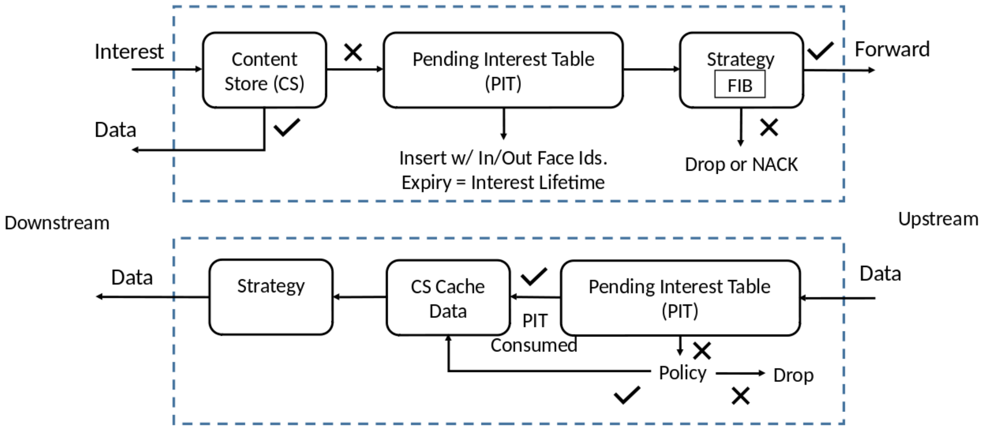


Figure 2. Forwarding Process at NDN node

NDN secures data by requiring data producer to sign each data packet cryptographically. This digital signature ensures data integrity and data provenance. Data consumer can validate the data authentication by verifying the signature without caring about where the data is from and how it is obtained. NDN develops a hierarchical trust model to define trust anchor, and trust rules. Trust anchor is a trusted, self-signed identity in the network by default. Trust rules decide which identity can be used to authenticate another one. The identity is in the form of NDN certificate, which consists of a namespace and a public key. The identity is signed by the trust anchor to prove that it is authenticated and authorized. Moreover, data-centric security enables content-based access control by encrypting the data at the time of production, and distribute the keys as Data packets automatically.

2.3 Name-based Access Control

Name-based Access Control (NAC) [6] is a content-based access control over NDN with end-to-end security, utilizes hierarchical meaningful naming convention to express fine-grained access control policies, and automates key distribution effectively. In NAC scheme (Figure 3.), it assumes trust relationships among entities have been established, each entity can authenticate data packets from other entities. Data owner controls both data production and data access directly through access manager.

Table 1. Notations

KEK	key-encryption key, public key
KDK	key-decryption key, private key
CK	Content Key, symmetric key

In NAC, there are three entities, Access Manager, data Producer, data Consumer. Access Manger represents data owner to specify the access policies, and generate a named public/private key pair as KEK/KDK, see notations in Table 1.. NAC leverages NDN's structured naming convention to express the access control policy in the KEK names. Access Manager encrypts KDK (private key) using authorized consumer's public key, so that only authorized consumers with access permission can decrypt the KDK. Then, KEK and encrypted KDK are published to the network as Data packets to satisfy incoming Interest to fetch. Data Producer follows the access control policy from KEK name to encrypt data, the details are as follows: After fetching KEK, Producer learns the granularity of access control by checking the KEK name, and encrypts the data in that granularity (granularity here is the name prefix of data produced by data producer) using a CK, then the CK is encrypted using that KEK. Data Consumer sends Interest packets to fetch the data he/she desires, once upon receiving the encrypted data, data consumer learns the CK name from the Data packet, and uses CK name to fetch encrypted CK. To decrypt the CK, data consumer fetches KDK from the network after learning KDK name from CK data packet. Once CK is obtained, the encrypted data could be decrypted as well.

In NAC, Access Manager changes the KEKs and KDKs periodically in a short time, that serves for access revocation. The operation of KEK/KDK renewal is transparent to data consumers, since authorized data consumer can fetch the new KDKs by following the naming convention automatically when needed. When a data consumer is reported to be compromised, Access Manager will send a notification to data producer about KEK updates, data producer will generate new CK to encrypted the existing data and fetch the latest KEK for CK encryption. For the data published previously and stored in-network cache, the access cannot be removed.

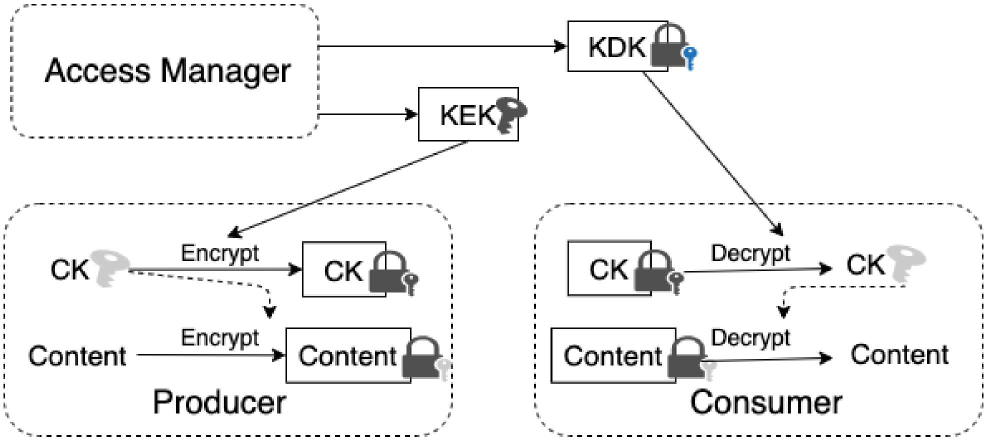


Figure 3. NAC Scheme

Chapter 3

Design

To facilitate the explanation and discussion of our work, we introduce a simple mobile health data sharing as an example, see Figure 4.. Data owner Bob uses a smartwatch to collect his daily activity data, heart rate, sleep patterns, stress, and so on. Those data are collected with the timestamp and location information to record when and where the data are collected. Bob may want to share his data with different people depending on his requirements. For example, Bob wants to share all his activity data generated in the daytime with data science researcher, but except for the data from Bob's house, his physician could access his heart rate data in a particular time period, e.g., between 8 am and 8 pm daily, and his coach may be granted access to his activity data generated only in the fitness center. These requirements contain various restrictions related to time and/or location.

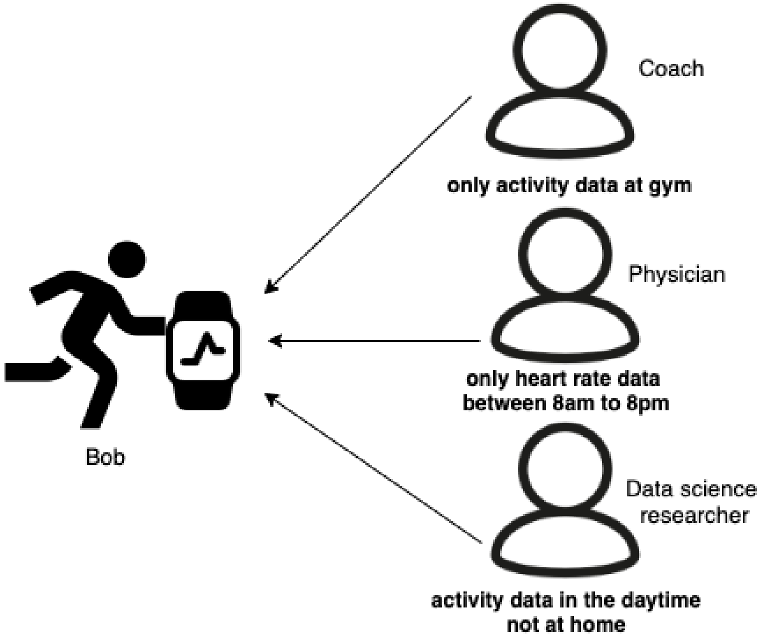


Figure 4. An example of mHealth data sharing

The goal of our work is to design and implement an access control scheme for data owners to share their data with fine granularity over time and/or location. We name the data that contains time and location information spatio-temporal data, and call our access control scheme spatio-temporal access control. This chapter describes namespace design for spatio-temporal access control to specify access control policy over time and/or location. We define access control policy for temporal access control, spatial access control, and spatio-temporal access control. Meanwhile, we illustrate how the policy can be enforced cryptographically and how the keys can be distributed automatically based on NAC model that we use as the cryptography mechanism for contents and keys. We also introduce the granularity of data encryption key (CK), and access revocation.

3.1 Namespace Design

Data is named under a hierarchical naming structure in NDN, this naming structure has been used widely to support application development, routing scalability, automatic data authentication and access control. Our access control namespace design has several requirements: naming data semantically and meaningfully, indicating data ownership, implying data encryption and decryption relationship. Figure 5. shows the namespace for spatio-temporal access control. To distinguish data name with access control namespace, we allocate access control credential namespace in parallel with data namespace. In the naming tree, $< dataowner >$ represents the name prefix of data owner, e.g., "/Alice", "/org/UofM", $< datastream >$ refers to the data type, e.g., health or activity. There are two sub-namespaces under each data stream, "DATA" and "READ", where "DATA" contains sub-namespaces related to data: name of each data point with time and location information, name of the content key used to encrypted the data, and name for data synchronization."READ" contains namespace for access control credentials. Additionally, users' key will be named "/<user-prefix>/KEY/<key-id>", e.g., Alice's key can be named "/edu/memphis/gym/coach/Alice/KEY/%B9%BC%13%80%7F%B2c%11".

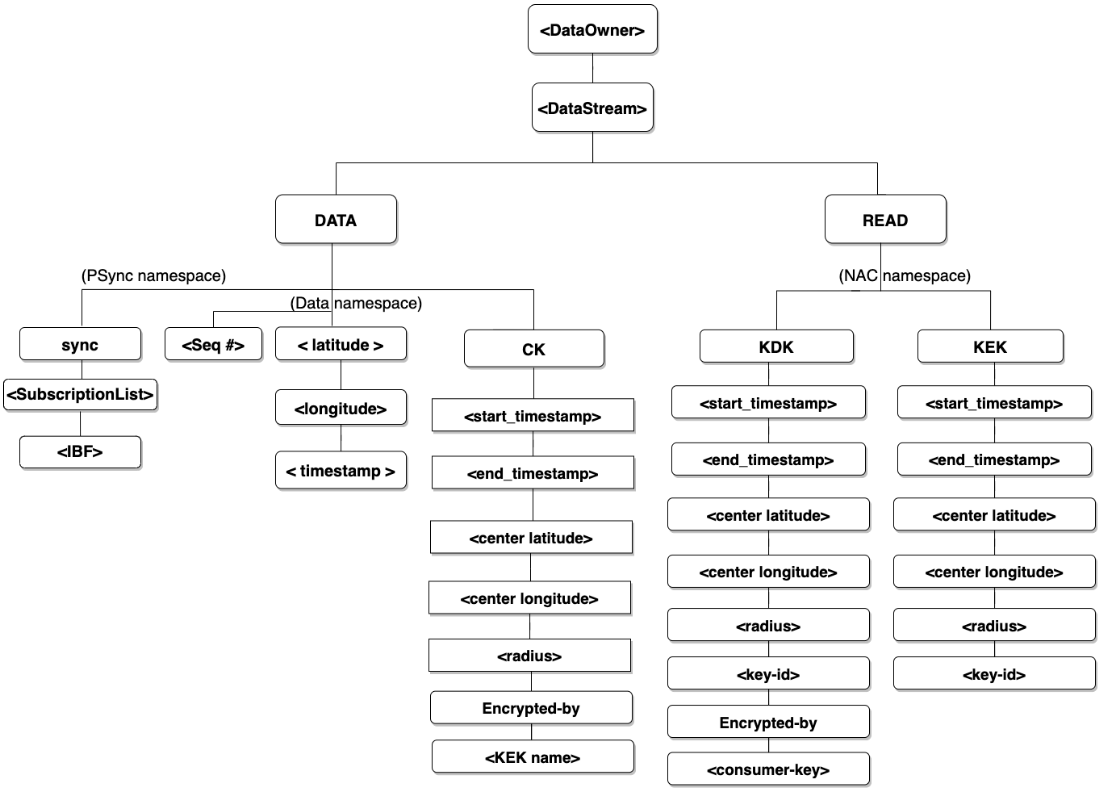


Figure 5. An example of user data namespace with access control credential namespace

3.1.1 Data Namespace

Under ‘DATA” namespace tree, there are three more branches to classify namespace for data, encryption key (CK) and Sync namespace. Each data packet contains a “<timestamp>” name component, “<timestamp>” name component indicates data production or collection time, and is represented in ISO 8601 format, i.e., “YYYYMMDDThhmmss”. To express the location information explicitly in the data name, we name data with two more name components “<latitude>”, “<longitude>”, they are geographic coordinates in decimal format to specify a position of the data point on the Earth’s surface. Data consumers can locate data producer through the data names which are public in the network, therefore, in order to prevent unauthorized consumers from locating the data producer, data owner could hide the sensitive location information without including the “<latitude>”, “<longitude>” name components from data names, or obscure the location information using particular techniques, this will be discussed in Section 6.

For real-time data access control, we deploy PSync [10] in spatio-temporal access control, so that data consumers can subscribe to the name prefixes of the producer's data streams based on their interests. Once data producer has updates, data subscribers could be notified, and then synchorinize the new update. The PSync namespace can be the name of Sync Interest/Data packets, which involves a particular name component "`sync`", and a subscription list "`<Subscriptionlist>`" along with a "`<IBF>`". "`<IBF>`" represents the state of the dataset. To cooperate with PSync, each new data name in the content of Sync Data packet has a "`<Seq#>`", that encapsulates the actual data name. For example, Bob's coach subscribes to Bob's activity data, will periodically sends a Sync Interest "`/Bob/activity/DATA/sync/<Subscriptionlist>/<IBF>`" to data producer. Once a new piece of data is produced at the university gym, a Sync data packet will be sent to Bob's coach. The Sync data packet contains the new data name with a "`<Seq#>`", e.g., "`/Bob/activity/DATA/<Seq#>`". Bob's coach then sends an Interest to retrieve the new data for "`/Bob/activity/DATA/<Seq#>`", the content of the new data contains the actual encrypted data and the actual data name "`/Bob/activity/DATA/35.111287/-89.928124/20201020T123000`".

3.1.2 Access Control Namespace

In our access control scheme, we extend the naming convention for data publication and consumption credentials based on NAC by including additional restritions for data access. As we described in Section 3, there are three types of keys, KEK, KDK, and CK. They play essential roles in enabling fine-grained access control. Under "`DATA`", there is a sub-namespace for "`CK`" that represents the name of content key for data encryption. In our design, each CK can be named with a time period and/or spatial area, the time period needs to cover the timestamp of the data that CK encrypts, and the GPS coordinates of the data has to included in the spatial area indicated in CK name. CK can be changed based on granularity of access control policy. Correspondingly, the time interval and/or spatial area in CK names will be changed as well. The time period in CK name could be one second, one minute, or every two minutes, and spatial area could be at a building level, e.g., gym, shopping mall. We identify the time interval by providing "<start_timestamp>" and "<end_timestamp>" , and create a spatial area using a center point with a radius, this information can be indicated in name components "<center_latitude>", "<center_longitude>" and "<radius>". Besides, CK data name includes additional name components "<ENCRYPTED-BY>" and "<KEK-prefix>", that indicate which KEK is used to encrypt the CK.

Under "READ" name component, there are two sub-namespaces applied to publish data consumption credential, i.e., KEK and KDK. KEK data packet can be named "<dataowner>/<datastream>/READ/KEK/<start_timestamp>/<end_timestamp>/<center_latitude>/<center_longitude>/<radius>/<key-id>", which indicates the data owner, data type, and additional access restritions related to time and location. Silimar to CK namespace, "<start_timestamp>" and "<end_timestamp>" can be used for time interval specification, and "<center_latitude>", "<center_longitude>" and "<radius>" can circle a spatial area for location. Additionly, KDK data name has the same name prefix with KEK, but replaces the name component "KEK" with "KDK", and appending authorized consumer's identity: "<dataowner>/<datastream>/READ/KDK/<start_timestamp>/<end_timestamp>/<center_latitude>/<center_longitude>/<radius>/<key-id>/ENCRYPTED-BY/<consumer-key>", where "<ENCRYPTED-BY>" indicates the KDK is encrypted using the public key of an authorized consumer "<consumer-key>". Each "<consumer-key>" is associated with a certificate that has to be validated by data owner.

3.2 Granular Access Control

Traditional access control mostly focuses on which users are granted access to some dataset, but pays less attention to control data access based on data attributes, time and/or location attributes. In order to protect data owner's sensitive data from being accessed by malicious retrievers, we need to ensure the data retrievers are granted access to the minimum amount of data based on when and/or where the data is produced. Fine-grained access control is required to specify who can access what data as well as at what granularity. In our work, we provide a precise way to specify fine-grained access control policy which encodes access constraints related to data attributes `(time and/or location)`, as well as have automated enforcement of such policy.

3.2.1 Access Control Policy Specification

To achieve truly user-controlled data access, data owner can define access control policy based on their needs. Data owner can specify which subset of the data each data consumer could be able to access based on data types and additional restrictions. Such restrictions could be related to the spatial-temporal attributes of the data, and applied to control data access based on when and where the data is generated. In the above example, we can see that various data consumers are assigned various access rights, e.g., Bob's physician could have right to access his heart rate data generated between 8 am and 8 pm every day, this requires access policy with spatial constraints; His coach Alice may be granted access to activity data that is generated only in the fitness center, this needs access policy to enclose location restriction; Bob grants data researcher access rights to his activity data generated during the daytime, but except the data at his house, this access requirement conbines both time and location restritions. In our access control scheme, we support all three types of policy specification and enforcement.

We conduct a hierarchically structured naming convention (Figure 5.) to express fine-grained access control policy with different restrictions. By following the namespace, we could specify access control policy over time, over location, or combination of time and location. The access policy can determine who is granted access to what data at which granularity. Specifically, “`who is granted access`” can be specified using data consumer’s identity, and data consumer could be named based on their organization, e.g., Bob’s physician could be identified using a name “`/org/baptistdoctors/physician/Dave`”, “`/edu/memphis/cs/Cathy`” identifies an authorized user Cathy from the University of Memphis, she could be the data science researcher. “`what data`” can be indicated using the name prefix of data, such as “`/Bob/activity`”, “`/Bob/steps`”, “`/Bob/heart_rate`”, etc. To deal with “`which granularity`”, we could encode the additional spatio-temporal restrictions of data to specify a time interval, location, or both. The following sections will explain the policy specifications and naming rules for each of them.

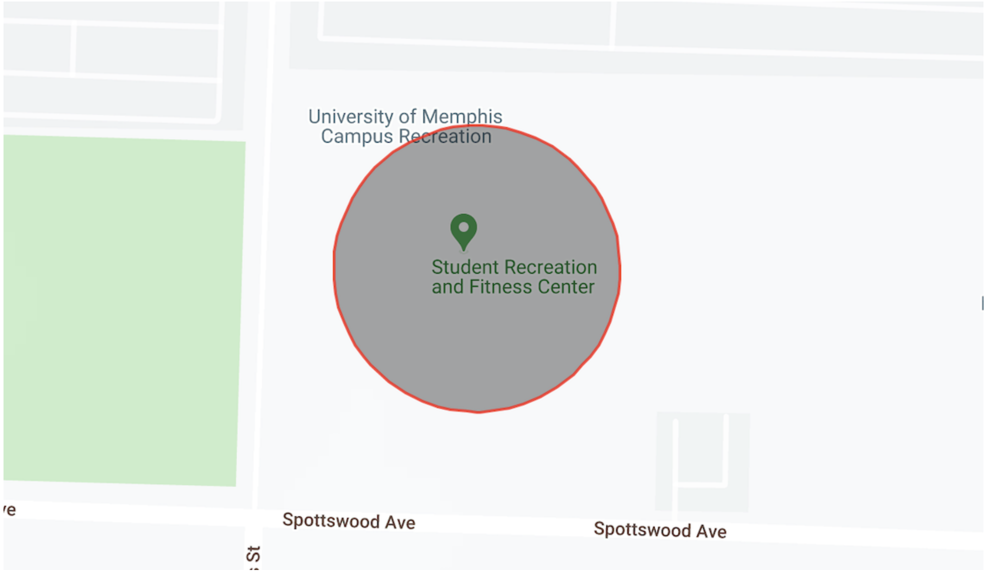


Figure 6. An example of spatial bound of fitness center

The temporal restriction can be specified based on temporal attribute and ranges, temporal range can be represented with a start time and an end time, e.g., “`between 8 am and 12 pm, 09/01/2020`”. The spatial restriction could be based on spatial attribute and a spatial bound. This spatial bound could be at a building level, neighborhood level, or city level, represented using a center point (latitude and longitude

GPS coordinate) and a radius(in meters). e.g., “center_latitude: 35.121196,center_longitude:-89.938124,radius:50” could be used to locate the spatial bound as a circle to represent the fitness center at University of Memphis, see Figure 6..

3.2.2 Temporal Data Access Control

For temporal data access control, users can define access control policies with temporal restrictions to determine data access based on when data is generated. Figure 7. presents the temporal policy structure, where “Data Type” can be filled with name prefix of the data, “User ID” can be the name prefix of data consumer’s identity, and “Schedule” is populated with a list of time schedules. Each time schedule can specify a temporal restriction to control data access by giving “startDate” and“endDate” to state the date constraint, “startHour” and “endHour” to indicate time constraint, and “unit” to repeat the policy. Once temporal access control policy is settled down, data owner can configure it to Access Manager, which can generate data production and consumption credentials. Taking the above mHealth data sharing as an example, Bob wants to share his heart rate data generated “between 8am to 8pm daily from Sept 1, 2020 to Sept 2, 2020” to his physician Dave. We can fill in the policy structure based on the policy description, where “startDate” is “09/01/2020”, “endDate” is “09/02/2020”, “unit:day” to satisfy“daily”. “DataType” would be ‘/Bob/heart_rate”. “UserID” represents data consumer’s identity, here Bob’s physician name is“/org/baptistdoctors/physician/Dave”.

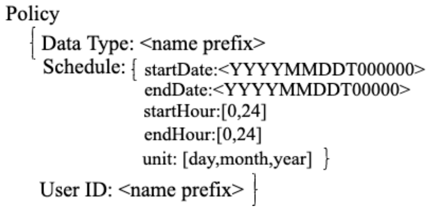


Figure 7. Temporal Access Control Policy

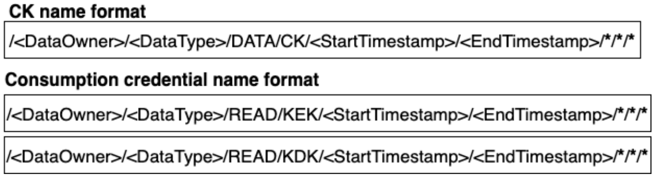
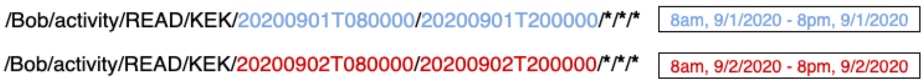

CK name format
/<DataOwner>/<DataType>/DATA/CK/<StartTimestamp>/<EndTimestamp>/*/*/*
Consumption credential name format
/<DataOwner>/<DataType>/READ/KEK/<StartTimestamp>/<EndTimestamp>/*/*/*
/<DataOwner>/<DataType>/READ/KDK/<StartTimestamp>/<EndTimestamp>/*/*/*

Figure 8. Temporal Access Control Naming Rules

Figure 8. lists the naming formats for production and consumption credentials (CK, KEK, KDK) by following the namespace design above, the time period “StartTimestamp” and “EndTimestamp” can be indicated from time schedules. However, in temporal data access control, the access policy is only relative to temporal constraints without spatial information. Our access control naming convention is still applicable without conflicts. To be consistent with the namespace design, we use symbol “⋆” to populate the name components for a spatial bound. Figure 9. shows the KEK names generated based on the access control policy assigned to Dave.



/Bob/activity/READ/KEK/20200901T080000/20200901T200000/*/*/* 8am, 9/1/2020 - 8pm, 9/1/2020

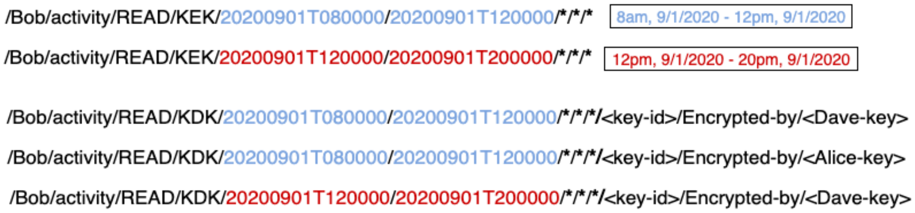
/Bob/activity/READ/KEK/20200902T080000/20200902T200000/*/*/* 8am, 9/2/2020 - 8pm, 9/2/2020

Figure 9. KEKs for Different Dates

In temporal access control, data owner can control access based on temporal limitations. This could lead to overlapping of time intervals. For example, Bob may only want to share his activity data with his coach Alice “between 8am to 12pm, Sept 1, 2020”, but with his physician Dave “between 8am to 8pm, Sept 1, 2020”. In this case, access manager could divide the access control policies into multiple KEKs/KDKs to avoid conflicts, and ensure that consumers can have the appropriate access privileges by getting corresponding KDKs, see Figure 10.. Access manager generates two KEKs for “8am-12pm” and “12pm-20pm” accordingly, and encrypts the KDK between 8am and 12pm using Dave’s key and Alice’s key separately , but only encrypt the KDK between 12pm and 20pm using Dave’s key, so that only Dave can have access permission for that period.

3.2.3 Spatial Data Access Control

In spatial data access control, users can specify access control policy with spatial limitations to control data access based on where the data is produced. Spatial access control policy could be specified using a specific structure as well, see Figure 11., where “DataType” and “UserID” are the same as the temporal access control,“Location”



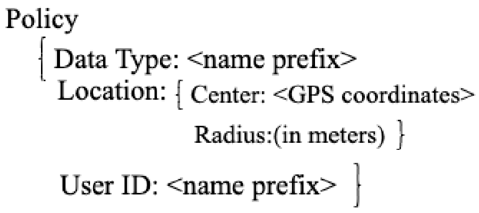
```
/Bob/activity/READ/KEK/20200901T080000/20200901T120000/*/*/*   8am, 9/1/2020 - 12pm, 9/1/2020
/Bob/activity/READ/KEK/20200901T120000/20200901T200000/*/*/*   12pm, 9/1/2020 - 20pm, 9/1/2020

/Bob/activity/READ/KDK/20200901T080000/20200901T120000/*/*/*/<key-id>/Encrypted-by/<Dave-key>
/Bob/activity/READ/KDK/20200901T080000/20200901T120000/*/*/*/<key-id>/Encrypted-by/<Alice-key>
/Bob/activity/READ/KDK/20200901T120000/20200901T200000/*/*/*/<key-id>/Encrypted-by/<Dave-key>
```

Figure 10. A Sequence of KEKs with Time Intervals

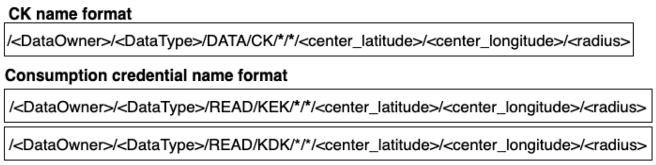
contains a list of spatial bounds to represent various locations, each one is indicated using a radius and latitude and longitude GPS coordinates of a center point. For example, coach Alice is granted access to Bob's activity data produced at gym and sports field. Let's suppose, the gym is the Fitness Center at the University of Memphis, the sports field is Liberty Bowl Memorial Stadium. The spatial bound of the gym can be represented with its center GPS coordinates, "`35.121196,-89.938124`" and an estimated radius "`50m`". And, the spatial bound of the sports field can be located with a center point "`35.1210966,-89.9774275`" and an estimated radius "`100m`".

When we configured this access policy in Access Manager, two pairs of KEKs/KDKs will be generated, the location information will be expressed in KEKs/KDKs name by following the naming rules in Figure 12.. Since in spatial data access control, access control policy has only spatial constraints, we utilize symbol "$\star$" to value the name components related to time interval to keep consistent with the namespace design. The "`<center_latitude>`,"`<center_longitude>`", and"`<radius>`" name components can get filled with the spatial information in the access policy.



```
Policy
  { Data Type: <name prefix>
    Location: { Center: <GPS coordinates>
                Radius:(in meters) }
    User ID: <name prefix> }
```

Figure 11. Spatial Access Control Polic



CK name format

```
/<DataOwner>/<DataType>/DATA/CK/*/*/<center_latitude>/<center_longitude>/<radius>
```

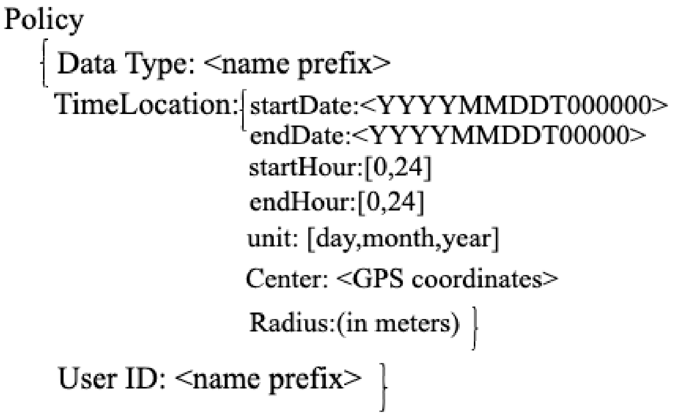
Consumption credential name format

```
/<DataOwner>/<DataType>/READ/KEK/*/*/<center_latitude>/<center_longitude>/<radius>
/<DataOwner>/<DataType>/READ/KDK/*/*/<center_latitude>/<center_longitude>/<radius>
```

Figure 12. Spatial Access Control Naming Rules

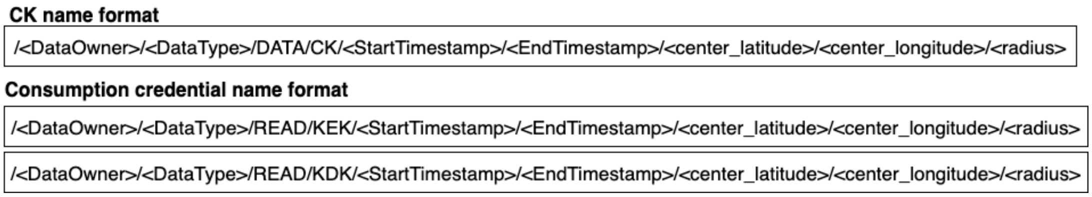
3.2.4 Spatio-Temporal Data Access Control

In spatio-temporal access control, access control policy combines both time and location restrictions. Specifically, users can control access to their data based on when and where the data is produced securely by specifying access control policy with time periods as well as spatial bounds. Figure 13. shows the policy structure, where "`DataType`" and "`UserID`" are the same as the spatial access control. Besides, "`TimeLocation`" is a list of combinations of time schedule and spatial bound. In Bob's mHealth data sharing example, the data researcher Cathy has permission to access Bob's activity data produced at CS department only in the daytime "`between 7:00 am and 6:30 pm`" on Sept 1, 2020. We can set a time schedule based on the time interval, same as temporal access control. Suppose the CS department is Dunn Hall at the University of Memphis, we can represent the spatial area by giving the center GPS point "`(35.121185, 89.938107)`" and a radius "`50m`".



```
Policy
  {Data Type: <name prefix>
   TimeLocation:{startDate:<YYYYMMDDT000000>
                 endDate:<YYYYMMDDT00000>
                 startHour:[0,24]
                 endHour:[0,24]
                 unit: [day,month,year]
                 Center: <GPS coordinates>
                 Radius:(in meters) }
   User ID: <name prefix> }
```

Figure 13. Spatio-Temporal Access Policy



CK name format

```
/<DataOwner>/<DataType>/DATA/CK/<StartTimestamp>/<EndTimestamp>/<center_latitude>/<center_longitude>/<radius>
```

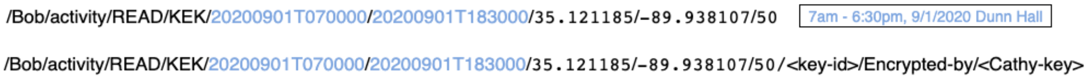
Consumption credential name format

```
/<DataOwner>/<DataType>/READ/KEK/<StartTimestamp>/<EndTimestamp>/<center_latitude>/<center_longitude>/<radius>
```

```
/<DataOwner>/<DataType>/READ/KDK/<StartTimestamp>/<EndTimestamp>/<center_latitude>/<center_longitude>/<radius>
```

Figure 14. Spatio-Temporal Access Control Naming Rules

Our spatial-temporal access control names the credentials by following the naming rules in Figure 14.. Based on the naming rules, Figure 15. shows the KEK/KDK names with location and time interval limitations.



```
/Bob/activity/READ/KEK/20200901T070000/20200901T183000/35.121185/-89.938107/50    7am - 6:30pm, 9/1/2020 Dunn Hall
/Bob/activity/READ/KEK/20200901T070000/20200901T183000/35.121185/-89.938107/50/<key-id>/Encrypted-by/<Cathy-key>
```

Figure 15. KEK and KDK Names with Time Interval and Location

3.2.5 Access Control Policy Enforcement

As described in Section 2, NAC is a content-based access control model in NDN, which we employ to carry out the access control policy in the names of KEK/KDK, and enable the key distribution automatically. Each content is encrypted at the time of production with a content key (CK), this CK is encrypted and published for only authorized users who can access the content. In NDN, keys are also data that can be fetched by a name. If a data point is encrypted using a CK, the CK name is enclosed in the data packet which can be learned and used by users to retrieve the CK in the network. To control access to the CK, named KEK is fetched by data producer to encrypt it. The KDK is encrypted by the authorized user's public key and published to the network. Figure 16. shows the key relationship in NAC. To access the data, data consumer sends interest to access manager to retrieve KDK, and decrypts the encrypted KDK with his/her private key. Then, data consumer could access the CK by decrypting it using the KDK, thus access the data after decrypting it with the CK. Figure 17. is an example showing data access for authorized consumer who can access the data successfully, and for whom not authorized cannot access the data, see Figure 18..

Based on NAC, we could apply spatio-temporal access control to data at any granularity related to time and location by following the above namespace design (see Figure5.). To enable access control policy enforcement automatically, each piece of data and the relevant keys are named consistently. KEK, KDK, and CK names share the same name prefix with the data that they control access to, and the policy is expressed in KEK/KDK names. Take Bob's activity data sharing as an example, following the naming

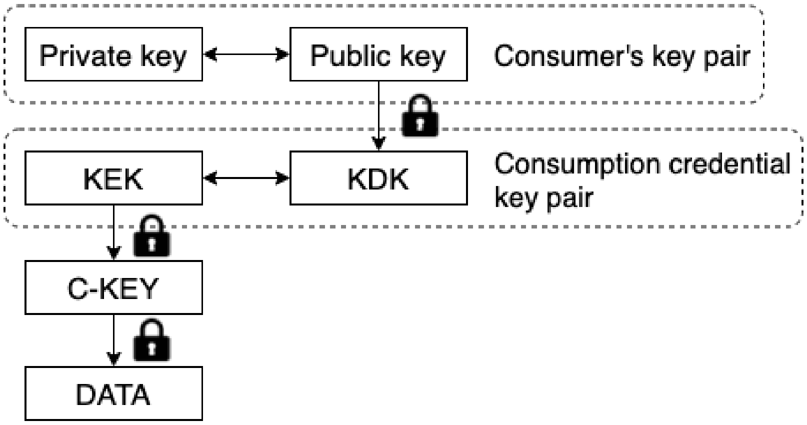


Figure 16. Key Relationship in NAC

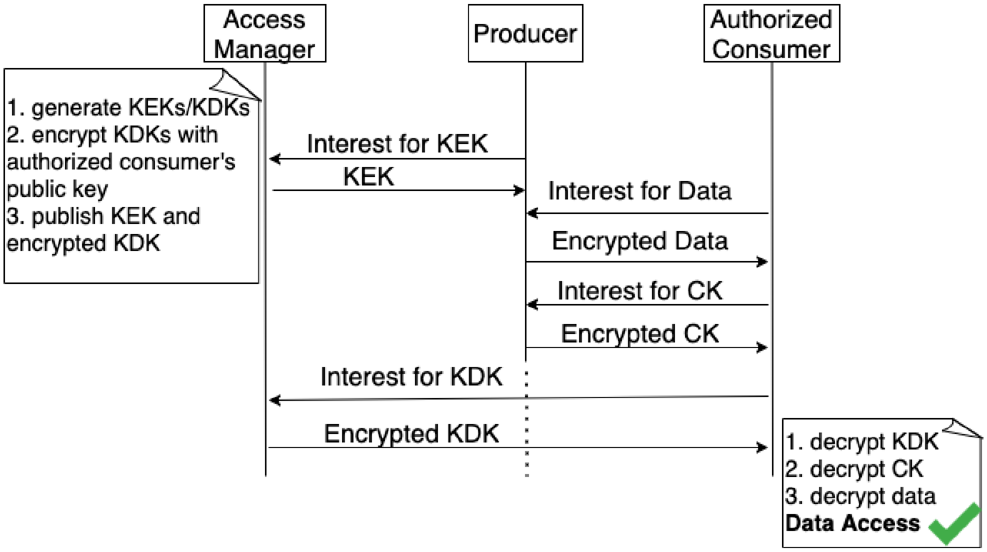


Figure 17. Data Access for Authorized Consumer

convention above, Bob can produce a name prefix "`/Bob/activity/DATA`" for his activity data, and a name prefix "`/Bob/activity/READ`" for consumption credentials. Figure 19. shows an example of spatio-temporal access control policy with descriptive details, including data set, time interval, start date, end date, time unit, spatial area specified by center point of GPS coordinates and radius, and user information identified by data consumer's identity. More Specifically, this access policy can be read as "`between 8am and 12pm`" from "`09/01/2020`" to "`09/05/2020`" every day, user "`/edu/memphis/gym/coach/Alice`" could access Bob's activity data set

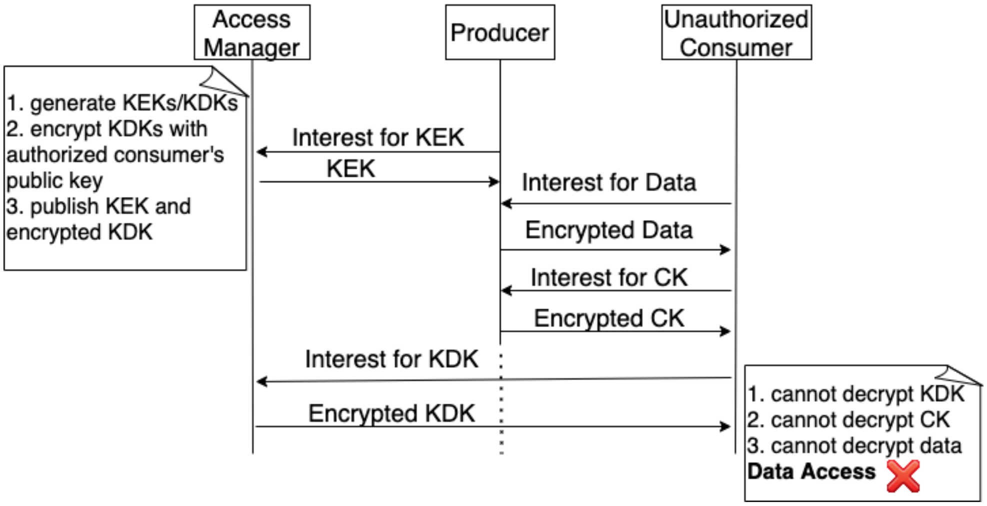


Figure 18. Data Access for Unauthorized Consumer

"`/Bob/activity`" collected from fitness center where the spatial bound can be displayed through "`center(35.121196,-89.938124)`", "`radius(50m)`".

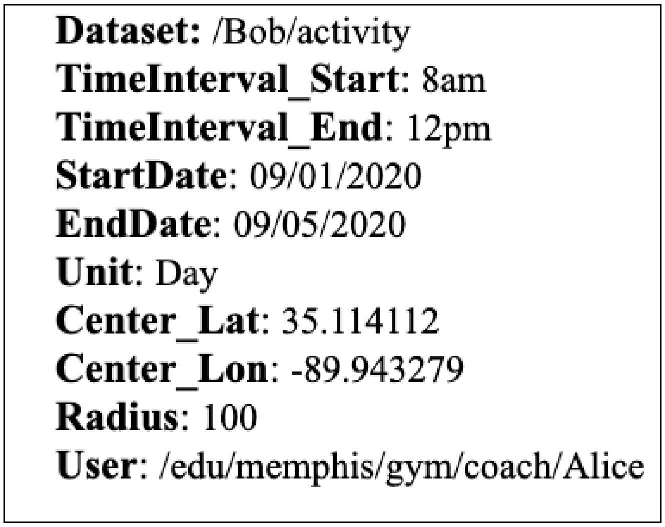
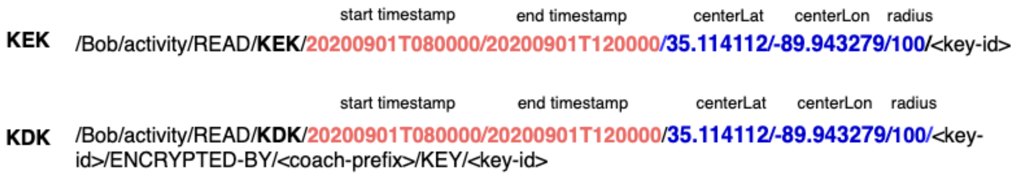

Dataset: /Bob/activity
TimeInterval_Start: 8am
TimeInterval_End: 12pm
StartDate: 09/01/2020
EndDate: 09/05/2020
Unit: Day
Center_Lat: 35.114112
Center_Lon: -89.943279
Radius: 100
User: /edu/memphis/gym/coach/Alice

Figure 19. An example of access policy

When this access rule is assigned to coach Alice. Access Manager serves for data owner Bob to generate named KEK and KDK pairs, Figure 20. shows the naming convention for one pair of the consumption credentials at `01/09/2020`. Access Manager should generate 5 pairs of KEK/KDK for the same time interval, same spatial area, but for different dates from `09/01/2020` to `09/05/2020`. These five consumption credentials can help Bob's coach successfully access Bob's activity data. Both KEK and KDK names share the same name prefix as the data they protect, /Bob/activity/, and additional spatio-temporal constraints are appended to the namespace. KDK name can be inferred from KEK name, but includes consumer's identity, since KDK is encrypted by authorized consumers' public key. After the access policy is issued to coach Alice, Alice can access Bob's activity data generated at fitness center between 8 am to 12 pm from 09/01/2020 to 09/05/2020 every day.



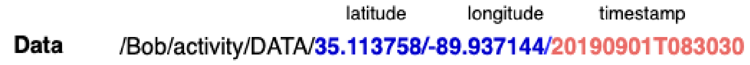
KEK /Bob/activity/READ/KEK/20200901T080000/20200901T120000/35.114112/-89.943279/100/<key-id>

KDK /Bob/activity/READ/KDK/20200901T080000/20200901T120000/35.114112/-89.943279/100/<key-id>/ENCRYPTED-BY/<coach-prefix>/KEY/<key-id>

Figure 20. Key Naming Convention of Consumption Credential

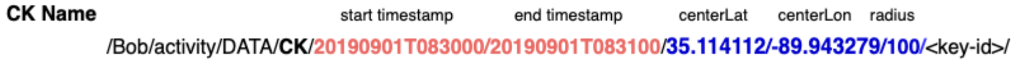
When Bob generates one piece of data somewhere inside the fitness center, at 8:30:30 am, Sept 1, 2020, this data can be named with the current GPS coordinate (latitude, longitude) and current timestamp "20190901T083030", the GPS coordinates are various, but have to be inside the spatial area we circled in the access policy. Figure 21. shows the data namespace including the data generation timestamp and geolocation information. Once the data is generated, we will find out a content key from the local storage, if there does not exist such CK whose time interval and spatial area can cover the timestamp and GPS coordinate of the data, a new CK needs to be generated to encrypt the data. Figure 22. is the CK name with the appropriate time interval from"20190901T083000" to "20190901T083100" and spatial area "Gym" to cover the data. In our work, users can specify CK granularity based on the granularity of access control policy, here CK is changed per minute, see details at Section 3.3. After the data is encrypted using the CK, the encrypted data along with the CK name is published for retrieval. To protect the new CK, data producer requires to fetch appropriate KEK with an Interest name that appends the time interval and spatial area of CK name. After getting the

Interest packet, Access Manager will look up the right KEK with “appropriate” name (see Figure 20.), and return the KEK to data producer. Note that “appropriate” name has to contain time interval and spatial area that cover the ones from the Interest name. Since CK is encrypted with KEK, CK data packet name needs to enclose the KEK name prefix, see Figure 23.. The encrypted CK is published, waiting for data consumer to fetch. Data consumer can learn CK name from encrypted data packet, and use it to fetch CK. Upon receiving encrypted CK, the CK data name can be used to derive KDK name prefix, which is the same as KEK name prefix. Then, data consumer creates an Interest packet with the derived name appending consumer’s public key name to retrieve KDK from access manager, see KDK name in Figure 20..



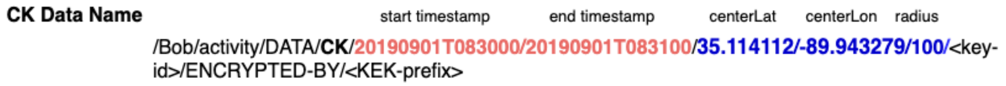
/Bob/activity/DATA/35.113758/-89.937144/20190901T083030

Figure 21. Data Naming Convention



/Bob/activity/DATA/CK/20190901T083000/20190901T083100/35.114112/-89.943279/100/<key-id>/

Figure 22. CK Naming Convention



/Bob/activity/DATA/CK/20190901T083000/20190901T083100/35.114112/-89.943279/100/<key-id>/ENCRYPTED-BY/<KEK-prefix>

Figure 23. CK Data Naming Convention

3.3 Granular Content Key

To enable user-controlled access control, our access control could make users specify different granularities of encryption key. This granularity could be changed depending on time and/or location. In Name-based Access Control mechanism, producer encrypts the data with a symmetric key (CK), this CK is encrypted using KEK (A public key published from Access Manager), only authorized users can retrieve KDK to decrypt the encrypted CK. Once the user gets the CK, then all the data that are encrypted using that CK would be decrypted and exposed to the user.

CK could be modified depending on the granularity of access control policies. For temporal access control, CK granularity follows the granularity of the temporal access control policy. If the minimum unit of temporal access is one hour, the CK can be changed hourly, or finer (e.g., per minute, per second), but needs to be equal or coarser to the data generation rate. For spatial access control, the content key could be changed based on locations at different levels, such as building level, neighborhood level, or city level. For access control policies with both spatial and temporal restrictions, CK granularity is the combination of the granularity of both time and location attributes.

Let's continue taking the spatial-temporal access policy in Figure 19. as an example. The granularity for temporal restriction is in hour level, while granularity for spatial restriction is at a building level. In this example, Bob generates his activity data every second. Based on access control policy and data generation rate, we could generate CK per minute, every two minutes, or per hour. Figure 24. shows an example of three CKs with different granularity, where we fix spatial granularity to the building level. The time interval in CK name indicates the temporal granularity.

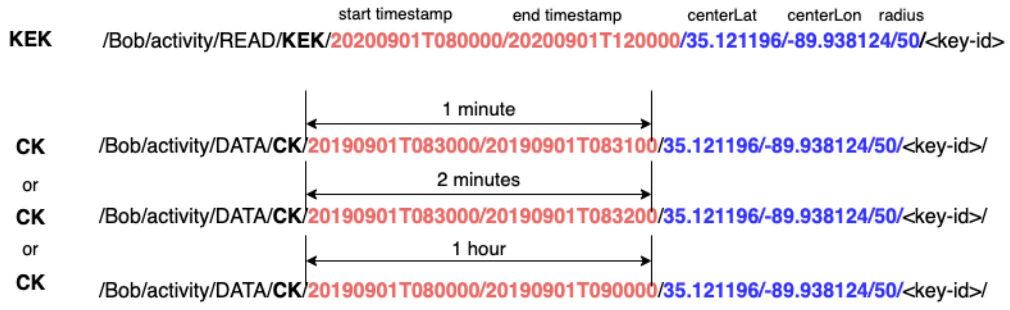

start timestamp
end timestamp
centerLat
centerLon
radius
KEK
/Bob/activity/READ/KEK/20200901T080000/20200901T120000/35.121196/-89.938124/50/<key-id>
1 minute
CK
/Bob/activity/DATA/CK/20190901T083000/20190901T083100/35.121196/-89.938124/50/<key-id>/
or
2 minutes
CK
/Bob/activity/DATA/CK/20190901T083000/20190901T083200/35.121196/-89.938124/50/<key-id>/
or
1 hour
CK
/Bob/activity/DATA/CK/20190901T080000/20190901T090000/35.121196/-89.938124/50/<key-id>/

Figure 24. An Example of CK Granularity

The granularity of the encryption keys (CKs) plays an important role in secure data sharing. For example, in Figure 24., if CK is generated per minute, the time range could be represented in CK name, e.g., "`/Bob/activity/DATA/CK/20190901T083000/20190901T083100/35.121196/-89.938124/50/<key-id>/`", this CK can cover 60 data points produced between"`20190901T083000`" and

“20190901T083100”, if this CK is compromised, those 60 data points would be leaked as well. But if we change CK per second, one CK can handle only one data point. Once this CK is compromised, we only leak one piece of data. Therefore, finer CK granularity is able to protect more data from being exposed.

3.4 Access Revocation

In spatio-temporal NAC, access revocation can be handled through updating KEK/KDK key pairs periodically, the keys renewal is transparent to data consumer, since the keys can be fetched whenever they are needed. For example, physician Dave is assigned access rights to read Bob’s activity data with restrictions “data is produced between 8am to 12pm at Dunn Hall on Sept 1, 2020”, and data researcher Cathy is granted access with restrictions “data is produced between 14pm to 20pm at Dunn Hall on Sept 1, 2020”. With these access policies, Dave and Cathy would be able to get the subset of data they want. In the case that Bob decides not to share his activity data from Dunn Hall with Cathy, for real-time data sharing, we can address this problem by limiting KDK’s effective time and updating KEKs/KDKs in short time period, see Figure 25., KEKs are updated while revoking Cathy’s access. The new KEKs will not be applicable to Cathy, but only for Dave.

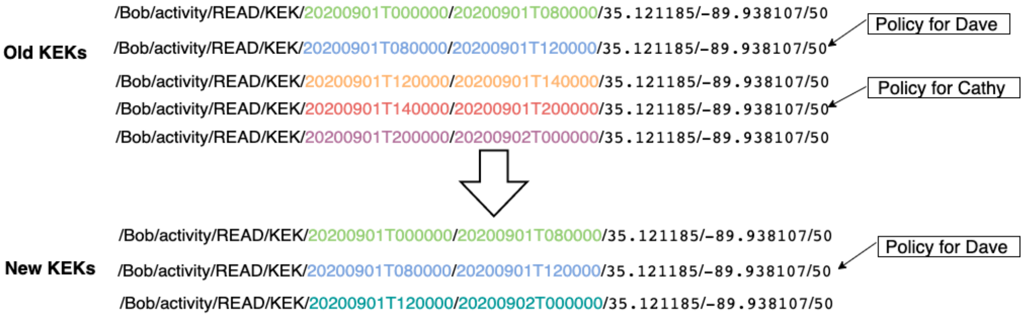

Old KEKs
/Bob/activity/READ/KEK/20200901T000000/20200901T080000/35.121185/-89.938107/50
/Bob/activity/READ/KEK/20200901T080000/20200901T120000/35.121185/-89.938107/50
/Bob/activity/READ/KEK/20200901T120000/20200901T140000/35.121185/-89.938107/50
/Bob/activity/READ/KEK/20200901T140000/20200901T200000/35.121185/-89.938107/50
/Bob/activity/READ/KEK/20200901T200000/20200902T000000/35.121185/-89.938107/50
Policy for Dave
Policy for Cathy
New KEKs
/Bob/activity/READ/KEK/20200901T000000/20200901T080000/35.121185/-89.938107/50
/Bob/activity/READ/KEK/20200901T080000/20200901T120000/35.121185/-89.938107/50
/Bob/activity/READ/KEK/20200901T120000/20200902T000000/35.121185/-89.938107/50
Policy for Dave

Figure 25. An Example of Access Revocation and KEK Updates

However, for historical data sharing from storage, if Cathy has already downloaded Bob’s activity data before removing her access permission, there is no way to prevent data from accessing by Cathy. If Cathy just fetched part of the data, we could prevent the left parts by re-encrypting them using new encryption keys (CKs).

Chapter 4

Implementation

In NAC design, there exists an NAC code library [11], which provides a fine-grained access control scheme based on data-specific granularity. Specifically, users can decide which dataset can be accessed by setting the name prefix of the data as granularity. Based on the existing NAC library and NDN C++ library (ndn-cxx) [12], we develop a spatial-temporal access control code base for application development, applications can control data access based on additional spatio-temporal restrictions on data.

We have implemented a practical prototype of spatio-temporal access control over NDN. This prototype can be applied in many situations: A smart home owner shares sensor data with family members or friends; A personal server collects mobile health data, shares the data with caregivers or others; A Cloud service operated by a research institute shares a big amount of data with data scientists; and possibly others. It can also be deployed for sharing data in real time.

4.1 Class Diagram

Figure 26. shows the class diagram based on our prototype implementation. By summarizing the above designs about controlling access base on spatial-temporal restrictions of data, the code library contains 10 classes. The detailed explanations are as follows.

“`AccessManager`” provides interfaces for applications to insert access control policy and generate KEKs/KDKs. “`Encryptor`” has functions for data encryption, CK generation, KEK retrieval, where we generate a CK with a key length of 256bits and use AES256 for data encryption, RSA for CK encryption. Moreover, it provides an approach to compute a bounding box [13] identified by “`[minLatitude, maxLatitude]`” and “`[minLongitude, maxLongitude]`” after giving a center point and a radius. With this bounding box, applications can determine if a data point with “`[latitude,`

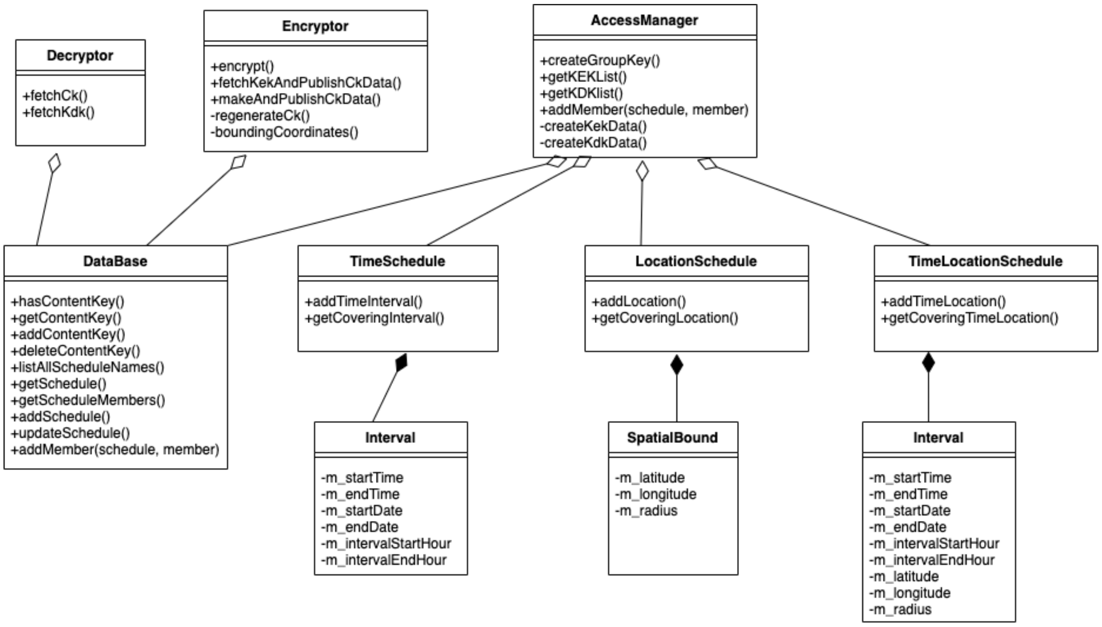


Figure 26. Class Diagram for Spatio-Temporal Access Control Code Base

`longitude]`” is located within a spatial circle, then find out an appropriate CK to encrypt that data point. “`Decryptor`” class can be used by data consumer for data retrieval, and key retrieval. “`DataBase`” provides an SQLite database to store CKs with name, so that CK can be looked up before generating a new CK. The database has interfaces to store time schedules and locations from access control policies, this is useful for later policy updates or access revocation. Other classes, such as “`TimeSchedule`”, “`LocationSchedule`” and “`TimeLocationSchedule`”, provides interfaces to add time schedules, locations, and both respectively.

Users who want to deploy this spatio-temporal acces control scheme, have to run three different applications. A producer application is used for data owner to start an NDN node, producing data and listening to Interests to satisfy. A manager application is used to import access control policies, and generate consumption production credentials. A consumer application works at data consumer’s side, sending Interest to fetch actual data and consumption credentials. All these applications are implemented in C++, and run on

Mac OS or Linux platform. Manger app and producer app could be run on the same node based on user's requirement.

4.2 Real-Time Data Sharing

In our prototype, we develop two more applications with a pub-sub module for data producer and consumer to support real-time data sharing. Data consumer can use the consumer application to subscribe to the data names they need, once the data is updated, they can get a notification, and synchronize the data. This feature is imperative, with real-time data sharing data retrievers can investigate the data immediately and take timely actions. For example, e.g. physician needs to retrieve patients' heart rate data in real time for timely intervention.

In NDN, there are existing developed dataset synchronization solutions, PSync[10], ChronoSync [14], both of them can notify end-users for new data updates and synchronize all the new data. However, PSync supports partial dataset synchronization, that allows end-users to subscribe to a subset of data, and only synchronize the subset of data they need. In our access control scenario, data consumers have to be authenticated before fetching the new data from data producers, and cannot share the data with other consumers for terms of use and data privacy. Therefore, partial dataset synchronization in PSync is applied, which guarantees data consumers can synchronize the new data they subscribe, while data consumers who did not subscribe to the new data cannot be notified.

PSync utilizes an Invertible Bloom Filter (IBF) [15] to represent the latest data names, and uses IBF 's subtraction operation to find out the new data names efficiently. Based on the new data names and subscriptions list, data producer sends a notification to consumer if any new data name is matching with the subscription. Figure 27. presents the workflow of pub-sub module after deploying PSync in our spatio-temporal access control. 1) Data consumer sends a routable Sync Interest to data producer, the Interest name contains a subscription list "`<SL>`" and "`<oldIBF>`", e.g., Bob's activity data is subscribed by someone, "`/Bob/activity`"; 2) Once data producer gets the Sync

Interest, he/she compares the old IBF with current IBF, and discovers the differences between them. When a new data is produced with a name matching with one of the subscription list, data producer sends a Sync reply with new data name to the consumer; 3) Upon receiving the Sync reply, data consumer checks if the new data name belongs to his/her subscription list, if there exists a false positive, that data name will be ignored, otherwise, data consumer sends Interest packet using the new data name as Interest name to retrieve the new data; 4) Data producer returns the new data, which encloses the actual encrypted data packet with the actual data name.

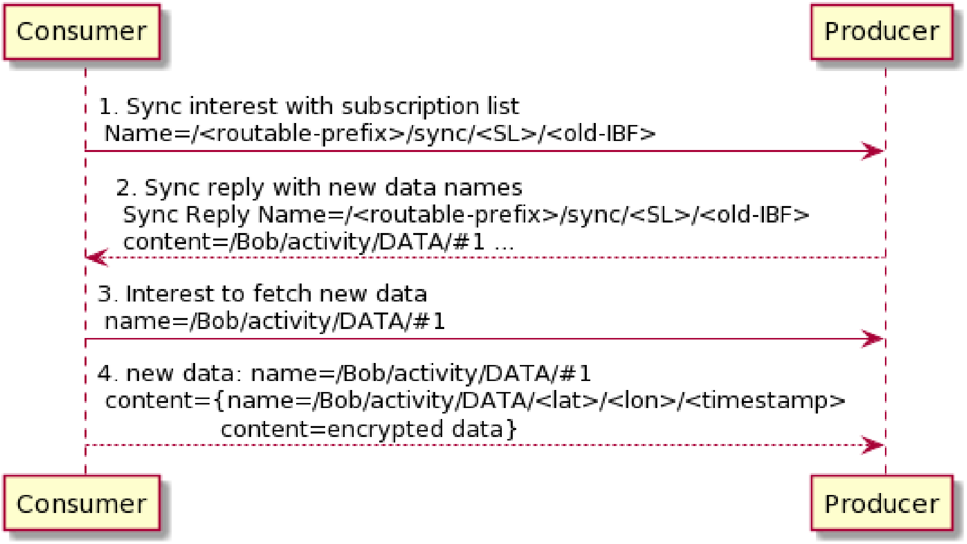

Consumer
Producer
1. Sync interest with subscription list
Name=/<routable-prefix>/sync/<SL>/<old-IBF>
2. Sync reply with new data names
Sync Reply Name=/<routable-prefix>/sync/<SL>/<old-IBF>
content=/Bob/activity/DATA/#1 ...
3. Interest to fetch new data
name=/Bob/activity/DATA/#1
4. new data: name=/Bob/activity/DATA/#1
content={name=/Bob/activity/DATA/<lat>/<lon>/<timestamp>
content=encrypted data}
Consumer
Producer

Figure 27. Pub-Sub Workflow in Spatio-Temporal Access Control

Chapter 5

Evaluation

In this section, we evaluate spatio-temporal NAC by analyzing the security properties and performance.

5.1 Security Analysis

Data consumer compromise: First, NAC executes access revocation periodically in a short time, and renews KEK/KDK key pairs. When a compromised data consumer is reported, he/she will not be granted new access rights. Second, we can specify CK granularity finer than the granularity of the access policy, the finer CK can cover lesser data points, when the CK is compromised, lesser data gets exposed.

Data consumer as attacker: Data can only be accessed by authorized consumers. Data consumer's identity needs to be validated before assigning the access policy through university email address, or phone number, etc. Besides, every sensitive data and keys (CK, KDK) are encrypted.

5.2 Performance Analysis

Regarding the performance of our spatio-temporal NAC, we evaluate data sharing from storage and data sharing in real time by running Mini-NDN [16] experiments. Mini-NDN is an NDN emulation tool that enables experimentation and testing on NDN platform. It is coded based on Mininet [17], and NDN-related libraries released by NDN project, such as NFD, NLSR, and NDN tools. With Mini-NDN, we could build NDN topologies with hundreds of nodes that can be run on a single machine (laptop, local VM, Cloud server, etc) directly to emulate an NDN network on a single system. Those nodes are connected via virtual Ethernet interfaces. We could configure a topology file by populating "`[nodes]`" and "`[links]`" sections based on our requirements. Figure 28. shows a simple Mini-NDN topology configure file and the topology it describes.

```
[nodes]
a: _ radius=0.5 angle=2.64159265359
b: _ radius=0.6 angle=3.64159265359
c: _ radius=1 angle=1.57079632679
d: _ radius=1 angle=4.71238898038
[links]
a:b delay=10ms
a:c delay=10ms
b:d delay=10ms
```

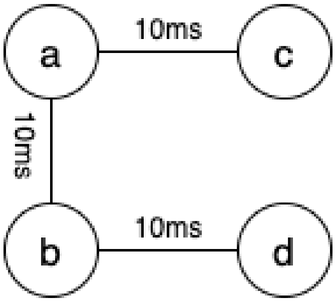


Figure 28. An Example of Mini-NDN topology

5.2.1 Experiments

Data Collection. We generate a sample of GPS data stream through MD2K Could platform (CerebralCortex) [18], which could specify the locations where the GPS data stream is collected. Therefore, the data stream contains a timestamp and GPS coordinates. There are 2000 data points, each data is generated per second.

Experimental Topology. To have a large NDN network for performance evaluation, we create a star topology with 101 nodes and 100 links, where we run data producer and access manager in the center node, and all other nodes are running for data consumers, the delay for each link is in the range between 10ms and 20ms. We use ndnsec [19] toolkit to generate a pub/pri key pair for each node, this can be used for later KDK encryption.

Access Policy. In our experiments, we define one access control policy which is applicable for all data consumers for easy deployment. To be specific, based on the dataset we collect, our policy description is "`access the GPS data stream generated between 8am and 12pm at Dunn Hall on 09/01/2020`".

Evaluation Metrics. In our evaluation, we consider the following metrics for non-real-time data sharing and real-time data sharing.

- **The number of CK.** By specifying different CK granularities, we compare the number of CK that we need to generate for data encryption.

- **Sync Delay.** Data synchronization delay, the time between new data is updated by data producer and notification is received by data consumer .

- **Data Production Time.** When a new data point is generated, through our access control, data producer secures the data by encrypting it using a symmetric key. We measure the runtime overheads of encryption process, it includes CK generation (32-bytes key), KEK retrieval, CK encryption (RSA2048), and data encryption (AES256).

- **Data Consumption Time.** When data consumer receives encrypted data, the data consumption involves CK retrieval, KDK retrieval, CK decryption (RSA2048), and data decryption (AES256).

- **Communication Delay.** The cost for data transmission between data producer and data consumer.

- **Data retrieval time.** It is the total time for data consumer to access each piece of data through spatio-temporal access control.

5.2.2 Evaluation Results

We run our experiments using Mini-NDN by setting up NDN topology. Figure 29. shows the count of CK we need to generate while changing the granularities. In our experiment, we change the CK granularity based on the time attributes, since our data stream is collected per second, we change CK every 1s, every 2s, every 3s, so on and so forth. The tradeoff occurs between finer granularity and the number of CK generated. Coarser granularity generates less CKs, for the same amount of data sharing, this makes one CK cover more data, once the CK is compromised, more data will be exposed. For instance, when the data is collected per second, the CK granularity is 1 hour, one CK can cover 3600 data points, once it is compromised, those data would be leaked as well. If we change the granularity to 1s, then even the CK is exposed to attackers, we prevent more data points from further leakage.

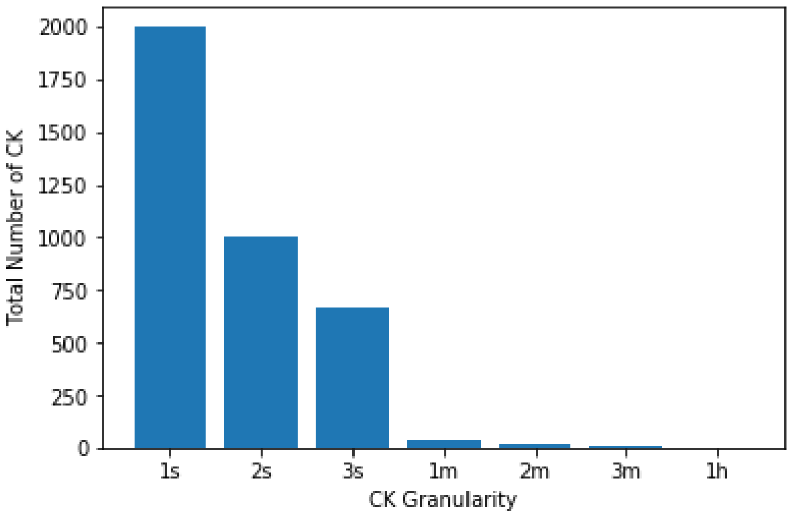


Figure 29. Total Number of CK for Different Granularity

Figure 30. shows the average time cost for each data synchronization with different time-based CK granularities. As CK granularity becomes coarser, the sync delay changes a little, the time difference could be negligible. The average sync delay for each data point is about 15ms.

We measure data production time cost by changing CK granularity over time for real-time data sharing and non-real-time data sharing of 2000 data points and take the average time as a result. As we can see in Figure 31., the data production time for both data sharing has no big difference at each CK granularity, since the process for each data production for sharing data in storage and in real-time is the same. However, as the granularity becomes coarser, the time becomes less, this is caused by the tradeoff of CK granularity. The coarser CK can cover more data, we can save time for symmetric key generation(32-bytes key).

Each data consumption time is similar as well, see Figure 32., since the process for each data consumption for both sharing data in storage and in real-time is the same. However, the same tradeoff occurs at data consumer side, when data consumer decrypts a

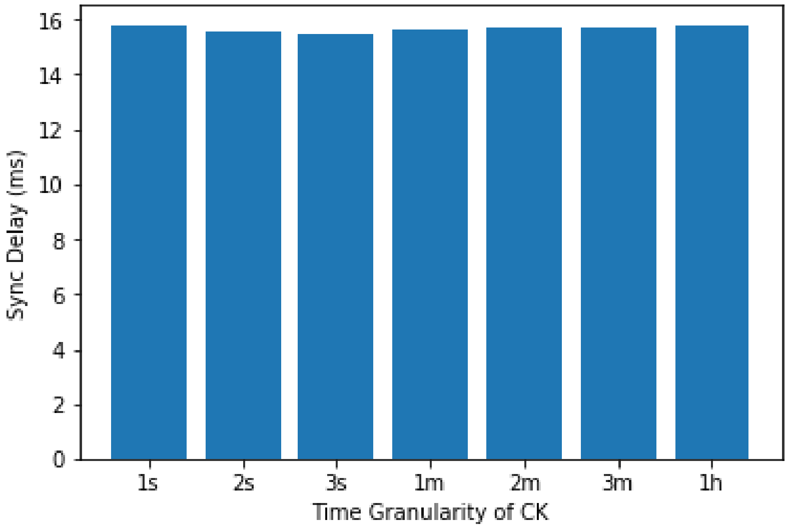


Figure 30. Sync Delay for Different Granularity

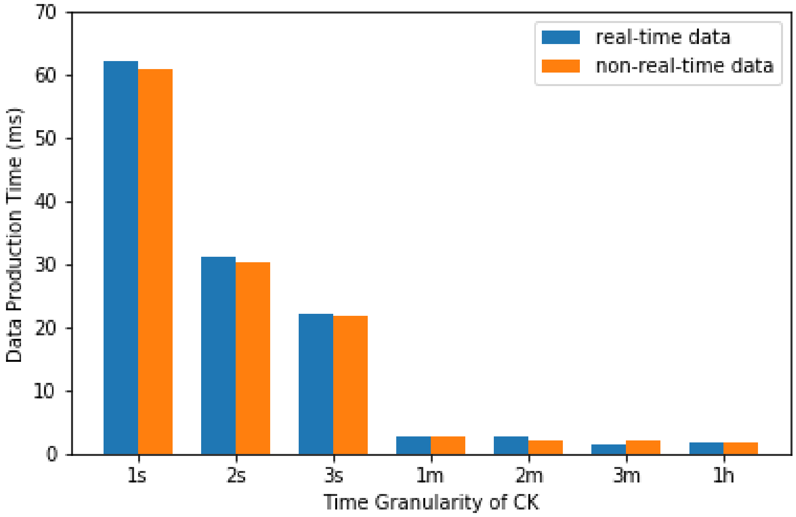


Figure 31. Data Production Time for Different Granularity

CK, if that CK is coarser, it can be used to decrypt more data, we save the time for new CKs and KDKs retrieval, as well as CK and KDK decryption. This makes data decryption more efficient. Figure 33. shows the transmission cost between consumer and producer, the average time for each piece of data is about 30ms. Figure 34. shows the result of data retrieval cost, the average time for accessing 2000 data points. For real-time data sharing, it takes more time to access each data point than sharing historical data from storage due to the sync delay. Overall, using coarser CK can provide more performance impacts than finer CK, but if coarser CK gets compromised, more data will be exposed as well.

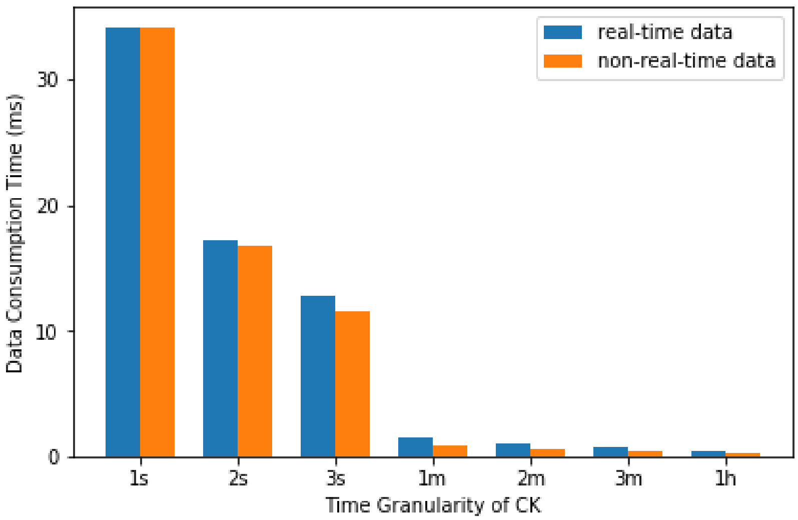


Figure 32. Data consumption Time for Different Granularity

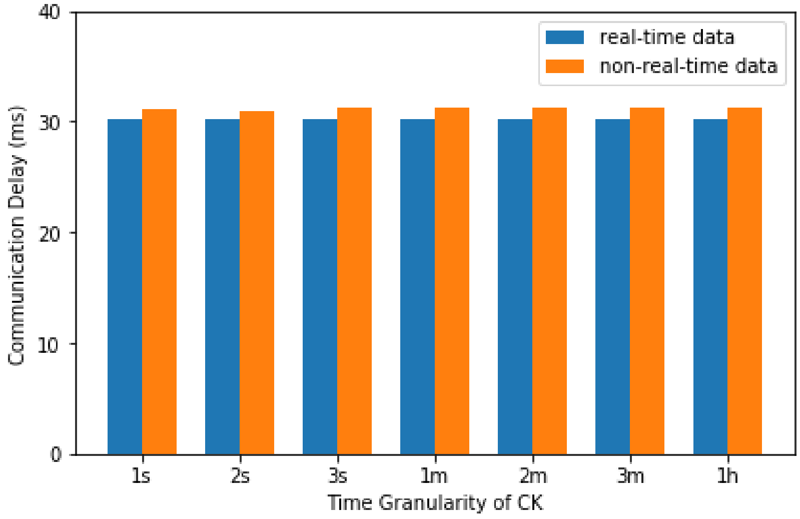


Figure 33. Communication Delay for Different Granularity

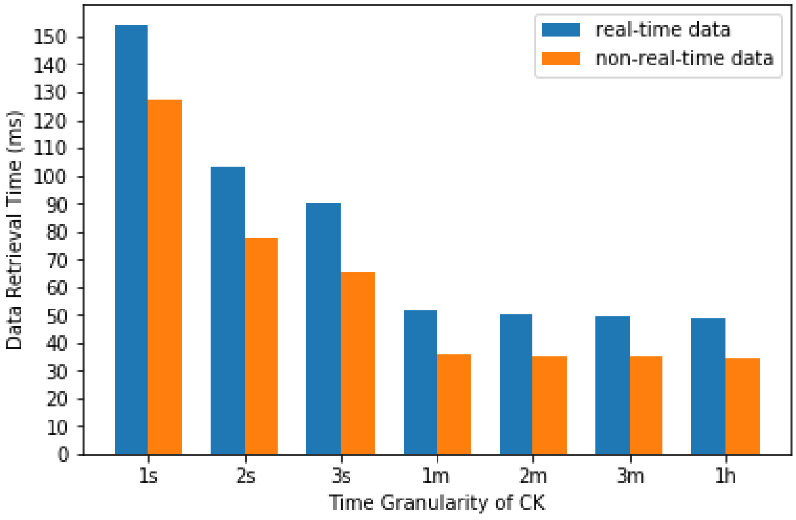


Figure 34. Data Retrieval Time for Different Granularity

Chapter 6

Discussion and Future Work

First, in our design of the naming convention, we include user's location information as name components in the namespace of KEK, KDK to express access control policy, and in the CK namespace to specify location-based granularity. The location information may be sensitive and personal, e.g., user's home address, and in NDN, data is fetched by giving the data name, so attackers can sniff the data packets and extract the location information of the user. To prevent location information exposure, we can obfuscate the name by using an encryption function or hash function, only authorized users are able to derive the real data name. Some name obfuscation solutions are proposed [20]. We leave the development of name obfuscation as future work.

Second, in our prototype implementation, we specify a spatial area/location by providing a center point and a radius in access control policy manually, this is configured by entering the center GPS coordinate and radius. A user-friendly and usable interface is needed for data owners to geo-locate a spatial bound through a map (e.g. Google map) directly, and specify a certain data type, data consumer, time interval through the interface, which can convert the input information into access control policy internally. In addition, the location constraint may be continuous, e.g. a path from location A to location B. We will figure out a way to express such constraints.

Third, in our performance evaluation, we only focus on changing the CK granularity over time, e.g., 1s, 2s, 1m. The location granularity is fixed at a building level. In real world, mobile data could be generated in a neighborhood, city, or country level. Meanwhile, we will expand multiple access policies in the experiment. We leave the experimentation and evaluation work for exploration in future work.

Chapter 7

Conclusion

As an enormous amount of sensitive data sharing is a growing trend, the design and development of a secure access control scheme for spatio-temporal data sharing is necessary. To protect the data privacy, and ensure only authorized users can retrieve the data. In this paper, we design a spatio-temporal access control scheme based on existing Name-based Access Control over NDN, and develop a practical access control prototype for deployment and evaluation.

Our spatio-temporal NAC provides a hierarchically structured naming convention to describe fine-grained access control policy over time and/or location, which enables user-controlled access control on the level of the data sharing, not just on who can access the data. This work leverages the existing content-based access control scheme and synchronization solution in NDN, to allow data owners to share existing data from storage as well as share data in real time. We evaluate a preliminary prototype through security analysis and running Mini-NDN experiments, the result shows that our spatio-temporal NAC can achieve data sharing efficiently and securely.

REFERENCES

[1] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” *in Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, 2006.

[2] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute- based encryption,” *in Proceedings of IEEE Symposium on Security and Privacy*, 2007.

[3] S. Jahid, P. Mittal, and N. Borisov, “Easier: Encryption-based access control in social networks with efficient revocation,” 2011, pp. 411–415.

[4] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, “Named Data Networking,” *ACM SIGCOMM Computer Communication Review*, July 2014.

[5] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev, and L. Zhang, “An Overview of Security Support in Named Data Networking,” *IEEE Communications Magazine*, vol. 56, pp. 62–68, November 2018.

[6] Z. Zhang, Y. Yu, S. K. Ramani, A. Afanasyev, and L. Zhang, “NAC: Automating Access Control via Named Data,” in *IEEE Military Communications Conference (MILCOM)*, 2018.

[7] P. Samarati and S. de Capitani di Vimercati, “Access Control: Policies, Models, and Mechanisms,” vol. 2171, October 2001, pp. 137–196.

[8] P. S. Sandhu, E. J. Coynek, H. L. Feinsteink, and C. E. Youmank, “Role-Based Access Control Models,” vol. 29, February 1996, pp. 38–47.

[9] “Nfd developer’s guide,” http://named-data.gitlab.io/TR-NDN-0021-NFD-dev-guide/ndn-0021-nfd-guide.pdf, (Accessed on 10/11/2020).

[10] M. Zhang, V. Lehman, and L. Wang, “Scalable Name-based Data Synchronization for Named Data Networking,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, May 2017.

[11] “named-data/ndn-cxx: ndn-cxx: Ndn c++ library with experimental extensions,” https://github.com/named-data/ndn-cxx, (Accessed on 10/11/2020).

[12] “named-data/name-bases-access-control: Ndn named-based access control,” https://github.com/named-data/name-bases-access-control, (Accessed on 10/11/2020).

[13] J. P. Matuscheck, “Finding Points Within a Distance of a Latitude/Longitude Using Bounding Coordinates,” in *Technical Report*, 2011.

[14] Z. Zhu and A. Afanasyev, “Let’s chronosync: Decentralized dataset state synchronization in named data networking,” in *2013 21st IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2013, pp. 1–10.

[15] M. T. Goodrich and M. Mitzenmacher, “Invertible bloom lookup tables,” in *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2011, pp. 792–799.

[16] NDN Project Team, “Mini-NDN GitHub,” https://github.com/named-data/mini-ndn, (Accessed on 10/11/2020).

[17] B. Lantz, B. Heller, and N. McKeown, “A network in a laptop: rapid prototyping for software-defined networks,” in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, 2010, p. 19.

[18] MD2Korg, “CerebralCortex,” https://github.com/MD2Korg/CerebralCortex, (Accessed on 10/11/2020).

[19] NDN Project Team, https://named-data.net/doc/ndn-cxx/current/manpages/ndnsec.html, Title = ndnsec, (Accessed on 10/11/2020).

[20] C. Ghali, M. A. Schlosberg, G. Tsudik, and C. A. Wood, “Interest-Based Access Control for Content Centric Networks,” September 2015.