Security Analysis of the Access Control Solution of NDN Using BAN Logic

Yuan Fei  Huibiao Zhu*  Huiwen Wang
Shanghai Key Laboratory of Trustworthy Computing,
School of Computer Science and Software Engineering,
East China Normal University, Shanghai, China

Abstract—Named Data Networking (NDN) is a new promising architecture of information-centric networking. For its caching property, traditional mechanisms of access control can no longer work. Hamdane et al. propose a new access control solution for both closed and open environments. In this paper, we make the very first attempt to formally analyze this access control solution. Inspired by the basic BAN logic which is often used to describe protocols by logical formulas, we present our BAN-like logic by adding some new notions to make it suitable for the access control solution. Using the BAN-like logic, the procedures of the access control solution is idealized in the form of the beliefs of principals. Then the idealized procedures are analyzed under several security goals with a set of logical postulates. Several unsatisfied goals may lead the access control solution to be vulnerable to intruders. We also present the related modification in the implementation of the access control solution. Our study helps to improve security and protect against various attacks for the access control solution.

Keywords—Named Data Networking (NDN), Access Control Solution, BAN Logic

I. INTRODUCTION

Named Data Networking (NDN) [1] is an architecture of Information-Centric Networking (ICN). ICN aims to offer solutions to problems existing in TCP/IP Internet. Nowadays users pay more attention to named content rather than its location. Though TCP/IP Internet has shown great resilience over the years, it cannot support the newly evolving content distribution model successfully. One of the promising candidates of ICN is NDN, which supports multicast of data and adopts the publish/subscribe model. The data producers mean publishers and the data consumers represent subscribers in NDN. When data consumer needs data, it sends out an Interest packet with a required name of the data; according to the name, routers forward the packet over the network; and a Data packet is returned to the consumer when a data produced by the data producer is matched. NDN routers can cache previous forwarded Data packets, which are able to be reused when a matching Interest packet comes. For this caching property, traditional mechanisms of access control, as a way of limiting access to data, can no longer work. Some access control specifications [2], [3] are proposed for NDN. However, each owns several limits. Hamdane et al. [4] put forward a new access control solution to address these limits.

We focus on this solution and analyze it step by step with our BAN-like logic.

The BAN logic [5] was first proposed by Burrows, Abadi and Needham in 1989. It provides a way to formalize the description and analysis of authentication protocols. It has been applied to analyze existing protocols and to find out their flaws [5]. Gaarder et al. [6] introduced new notions based on the basic BAN logic specially for PKCS authentication protocols. In order to reason about cryptographic protocols, Gong et al. [7] added more accurate concepts and definitions to the basic BAN logic. By adding negation, [8] presented the special BAN logic designed for monotonic protocols.

Our BAN-like logic is inspired by the basic BAN logic. We add some new notions to make it suitable for the access control solution. Adopting the BAN-like logic, the procedures of the access control solution in [4] is idealized. Then the idealized procedures are analyzed under several security goals with a set of logical postulates. The results show that some goals could not be archived. It indicates that this access control solution cannot ensure the source of critical keys and data. This may lead the access control solution to be vulnerable to intruders. We give the modification of the idealized procedures and the proofs of them. Meanwhile we also present the related modification in the implementation of the access control solution. Our study helps to improve security and protect against various attacks for the access control solution.

The rest of the paper is organized as follows: Section II briefly introduces the access control solution of NDN and explains how it works. Section III gives the BAN-like logic definitions and the assumptions of the access control solution. In Section IV and Section V, we apply BAN-like logic to analyze the access control solution of write and read operations in a closed environment. Section VI displays the analysis of write and read operation in an open environment. Finally, Section VII concludes and points out the future work.

II. ACCESS CONTROL SOLUTION OF NDN

In this section, we introduce the access control solution [4] for NDN. We give the related entities and assumptions. The write operations and read operations in closed and open environments are proposed.
A. Entities and Assumptions

Entities in NDN own two roles: readers and writers. An access control manager (ACM) is introduced to control the management of the access control policy. It creates a key pair \((NK_{\text{encryption}}, NK_{\text{decryption}})\), which is designed for encrypting a symmetric data key \(DK\). These keys are similar as public and private keys, but both of them are secret. \(NK_{\text{encryption}}\) is applied to encrypt the \(DK\) and only be acquired by entities with write privilege. \(NK_{\text{decryption}}\) allows the \(DK\) decryption which can be obtained by any entities. Meanwhile, every entity owns the related public and private key pair. The public writer keys are used to encrypt key pair \((NK_{\text{encryption}}, NK_{\text{decryption}})\), and public reader keys are adopted to encrypt key \(NK_{\text{encryption}}\).

B. Write Operations and Read Operations

We simplify some steps of write and read operations in [4] and only retain the processing related with keys and data. Fig. 1 illustrates four situations of write and read operations in different environments. Because the node playing reader’s or writers role can be connected with a normal node or the ACM, the communication object of Writer\(_i\) and Reader\(_i\) can be ACM or NN. NN is only responsible to transit message to ACM eventually. As a result, Writer\(_i\) and Reader\(_i\) are communicating with ACM essentially.

1) Write operations in a closed environment:
Assuming that Writer\(_i\) knows the name of key pair \((NK_{\text{encryption}}, NK_{\text{decryption}})\) in advance, it sends an Interest packet containing Name \(NK\) to ACM (step a.1). ACM returns a Data packet with all the hash value of public keys that ACM has already known (step a.2). Then Writer\(_i\) recognizes its own hash and transmits it together with Name \(NK\) as a new Interest to ACM (step a.3). ACM recognizes which writer is communicating with it, and uses Writer\(_i\)’s public key to encrypt key pair \((NK_{\text{encryption}}, NK_{\text{decryption}})\). This is added to a new Data packet which is fed back to ACM (step a.4). Then Writer\(_i\) sends the encrypted data DataDK to ACM (step a.5). ACM learns Name \(NK\) and sends the Interest to ACM (step a.6). ACM then uses \(NK_{\text{encryption}}\) to encrypt \(DK\) and produces a new Data packet as a reply (step a.7).

2) Read operations in a closed environment: Reader\(_i\) sends the required Data name to ACM (step b.1). ACM replies Reader\(_i\) with Data packet including Data encrypted with data key \(DK\) (step b.2). Reader\(_i\) knows the name of data key \(DK\) and sends the Interest to ACM (step b.3). ACM uses \(NK_{\text{encryption}}\) to encrypt \(DK\) and creates a Data packet to send back to Reader\(_i\) (step b.4). Reader\(_i\) sends the Interest packet carrying name of key pair \((NK_{\text{encryption}}, NK_{\text{decryption}})\) to ACM (step b.5). ACM returns all the hash value of public keys (step b.6). Reader\(_i\) gives ACM with Name \(NK\) and its hash value of public key (step b.7). ACM uses the related public key to encrypt \(NK_{\text{decryption}}\) to produce a Data packet for ACM (step b.8).

3) Writer operations in an open environment: As the environment is open, Writer is unknown to ACM. As a result, Writer needs to send ACM its hash value of public key. ACM can get the related public key to encrypt information. First, Writer sends Name \(NK\) and the hash value of its public key to ACM (step c.1). As ACM has acknowledged the public key of Writer, it uses the key to encrypt key pair \((NK_{\text{encryption}}, NK_{\text{decryption}})\) (step c.2). Writer sends the encrypted data \{Data\}DK, together with encrypted data key \{DK\} \(NK_{\text{encryption}}\) (step c.3).

4) Read operations in an open environment: Because the environment is open, ACM does not acknowledge the public key of Reader. So it is necessary for Reader to transmit the hash value of its public key to ACM. First, Reader sends an Interest with Data name to ACM (step d.1). ACM responds the encrypted data \{Data\}DK and the encrypted data key \{DK\} \(NK_{\text{encryption}}\) (step d.2). Reader transmits Name \(NK\) and its hash value of public key to ACM (step d.3). Because ACM realizes the public key of Reader, it applies this key to encrypt \(NK_{\text{decryption}}\), which is built into a Data packet and conveyed to Reader (step d.4).

III. AN INTRODUCTION OF BAN-LIKE LOGIC AND ASSUMPTIONS

In this section, we introduce our BAN-like logic inspired by the basic BAN logic. We add some new notions to make it suitable for the access control solution. Assumptions of access control solution are also presented for further analysis.

A. Statements

There are three sorts of objects in our BAN-like logic: principals, keys, and formulas (also statements). The symbols \(P\) and \(Q\) range over principals; \(K\) ranges over keys; \(X\) ranges over formulas. The following are basic statements.

- \(P \models X\): The principal \(P\) believes the statement \(X\) is true.
• \( P \triangleleft X \): \( P \) sees \( X \), which represents \( P \) has received a message \( X \).
• \( P \vdash X \): \( P \) once said \( X \), which also means \( P \models X \) when \( P \) sent it.
• \( P \models X \): \( P \) governs \( X \), showing that \( P \) has an authority on \( X \).
• \( \models X \): The message \( X \) is fresh.

There is little difference between the keys in the basic BAN logic and the key used here. We add new notations for a better explanation. The key \( K \) represents the public key of asymmetric keys, which can be seen by other agents. Its associated private key \( K^{-1} \) will be secret to any other agents except one agent which owns it. The key pair \((@K,K@)\) denotes an asymmetric key pair, in which \( @K \) is used for encryption and \( K@ \) is devoted to decryption. An agent can only acquire this key pair from the package it received, if it is not the creator of the pair. The symmetric key \( SK \) is applied for both encryption and decryption, which is encrypted and transmitted between agents.

- \( \overset{K}{\rightarrow} P \): The encryption key \( K \) is the public key of \( P \). The matching private key \( K^{-1} \) will be secret to any other principals except \( P \).
- \( \overset{\oplus K}{\rightarrow} P \): The key \( @K \) is the encryption key of the asymmetric key pair \((@K,K@)\), which is acknowledged by \( P \).
- \( \overset{\ominus K}{\rightarrow} P \): \( P \) knows the key \( K@ \), which is the decryption key of the asymmetric key pair \((@K,K@)\).
- \( \overset{\ominus K}{\rightarrow} P \): \( P \) learns that the key \( SK \) is a symmetric key.
- \{\( X \)\}_K: The statement \( X \) is encrypted under the key \( K \). \( K \) can also be replaced by \( @K \) or \( SK \).

B. Logical Postulates

We introduce four categories of postulates and give their explanations.

1. The message-meaning rules are about interpretation of encrypted messages. We postulate the message-meaning rule for symmetric keys as below.

\[
\text{MM1} \quad P \models SK, Q \iff \models SK, Q, P \triangleleft \{X\}_SK
\]

If both \( P \) and \( Q \) believe that the key \( K \) is the symmetric key and \( P \) sees a message \( X \) encrypted under \( K \), then \( P \) believes that \( Q \) once said \( X \).

We also present the message-meaning rule for asymmetric keys as below.

\[
\text{MM2} \quad P \models K, Q \triangleleft \{X\}_{K^{-1}} \iff \models Q \triangleleft X
\]

If \( P \) believes that the key \( K \) is the public key of \( Q \) and sees a message \( X \) encrypted under \( K^{-1} \), then \( P \) believes that \( Q \) once said \( X \).

2. The nonce-verification rule states the check of the freshness of a message.

\[
\text{NV} \quad P \models \not\models X, P \models Q \triangleleft X \iff \models Q \models X
\]

If \( P \) believes a formula \( X \) is fresh and \( P \) believes that \( Q \) once said formula \( X \), then \( P \) believes that \( Q \) believes \( X \).

3. The jurisdiction rule expresses how jurisdiction effects the belief.

\[
\text{J} \quad P \models Q \models X, P \models \models Q \models X \iff \models X
\]

If \( P \) believes that \( Q \) has jurisdiction over \( X \) and \( Q \) believes \( X \), then \( P \) trusts \( X \).

4. The seeing rule describes the situation when a principal sees a formula encrypted with different kinds of keys.

\[
\text{SEE1} \quad P \triangleleft \{X\}_K, P \models K \iff P \triangleleft X
\]

If \( P \) sees \( X \) encrypted with a public key \( K \) and processes the corresponding private key \( K^{-1} \), then \( P \) is considered to have seen \( X \).

\[
\text{SEE2} \quad P \triangleleft \{X\}_{K^{-1}}, P \models K \iff P \triangleleft X
\]

If \( P \) sees \( X \) encrypted with a private key \( K^{-1} \) and owns the corresponding public key \( K \), then \( P \) is regarded as seeing \( X \).

\[
\text{SEE3} \quad P \triangleleft \{X\}_K, P \models KW \iff P \triangleleft X
\]

If \( P \) sees \( X \) encrypted with an encryption key \( K \) of a key pair \((@K,K@)\) and has the decryption key \( K@ \), then \( P \) is thought of as seeing \( X \).

\[
\text{SEE4} \quad P \triangleleft \{X\}_SK, P \models SK \iff P \triangleleft X
\]

If \( P \) sees \( X \) encrypted with a known symmetric key \( SK \), then \( P \) sees \( X \).

C. Assumptions of access control solution

We refer principals to the Access Control Manager (ACM), readers and writers, which are presented by symbols \( M, R_1,...,R_n \) and \( W_1,...,W_n \). \( KW_1,...,KW_n \) and \( KR_1,...,KR_n \) denote the public keys of writers \( W_1,...,W_n \) and readers \( R_1,...,R_n \) respectively. \( KW_1^{-1},...,KW_n^{-1} \) and \( KR_1^{-1},...,KR_n^{-1} \) represent the corresponding private keys. \((@NK,NK@)\) is the asymmetric key pair produced by ACM. \( M.\, SDK \) and \( M.\, Data \) is the symmetric key and data created by writers.

To analyze the access control solution, we first list the following assumptions.

- \( A_1: M \triangleright W_i \triangleright R_j \implies K_j \triangleright W_j \triangleright R_j \)
- \( A_2: W_i \triangleleft SDK \implies W_i \)
- \( A_3: M \models @NK \implies M \)
- \( A_4: M \models NK@ \implies M \)
- \( A_5: W_i \triangleright R_j \implies M \models (@NK, NK@) \)
- \( A_6: M \models R_j \implies W_j \models SDK \)
- \( A_7: M \models R_i \implies W_j \models Data \)
- \( A_8: W_i \models \not\models SDK \)
- \( A_9: W_i \models \not\models Data \)
- \( A_{10}: P \triangleright NK@ \implies P \models (@NK, NK@) \)
- \( A_{11}: P \models NK@ \implies P \models N\models P \)
- \( A_{12}: P \models (@NK, NK@) \implies P \models (@NK, NK@) \)
- \( A_{13}: P \models (@NK, NK@) \implies P \models \not\models P \)
- \( A_{14}: P \models (@NK, NK@) \implies P \models N\models P \)
- \( A_{15}: P \models SDK \implies P \models \not\models SDK \)
- \( A_{16}: P \models SDK \implies P \models SDK \)
- \( A_{17}: P \models Data \implies P \models Data \)

Assumptions \( A_1-A_4 \) are about the keys initially known to the principals. Assumptions \( A_5-A_7 \) describe that \( M \) is trusted by \( W_i \) to make key pair \((@NK,NK@)\) and \( M \) trusts that \( W_i \) can produce data key \( SDK \) and data \( Data \). Assumptions \( A_8 \) and \( A_9 \) indicate that \( W_i \) believes data \( Data \) and data key \( SDK \) are fresh. Assumptions \( A_{10}-A_{16} \) present the situation when a principal sees different keys. The principal \( P \) could be replaced by \( M, R_i \), and \( W_i \), \( P \) learns the related key which is also considered to be fresh. \( A_{17} \) shows when \( P \) sees data \( Data \) it also confirms the freshness of it.
IV. ANALYSIS OF WRITE OPERATION IN A CLOSED ENVIRONMENT

In this section, we introduce the specific write operation in a closed environment. Then we apply our BAN-like logic to this procedure. Several security goals are listed to be proved.

A. Write operation in a closed environment

We give the write operation procedure in a closed environment according to Fig.1(a) as below.

Message 1. Writer$_i$ → ACM : name$_{NK}$
Message 2. ACM → Writer$_i$ : name$_{NK}$, H(K$_1$),..,H(K$_n$)
Message 3. Writer$_i$ → ACM : name$_{NK}$, H(K$_i$)
Message 4. ACM → Writer$_i$ : name$_{NK}$, H(K$_i$), {NK$_c$, NK$_a$}$_K_i$
Message 5. Writer$_i$ → ACM : name$_{Data}$, {Data}$_D$K
Message 6. ACM → Writer$_i$ : name$_{DK}$
Message 7. Writer$_i$ → ACM : name$_{DK}$, {DK}$_N$K$_c$.

Writer$_i$ sends the name of key NK to ACM. ACM returns name$_{NK}$ together with the hash values of K$_1$..K$_n$, denoted by H(K$_1$),..,H(K$_n$). Then Writer$_i$ recognizes the correct hash values of K$_i$ in the hash value sequences and feedbacks it to ACM with name$_{NK}$. After receiving the package, ACM returns a new message added with the corresponding key pair (NK$_c$, NK$_a$). Writer$_i$ uses its special data key DK to encrypt the data and sends it to ACM for storing. ACM can learn the name of data key DK and send it to Writer$_i$. Finally, Writer$_i$ returns the message of DK encrypted by key NK$_c$ with name$_{DK}$. This can lead ACM to acknowledge data key DK, which is used to decrypt message {Data}$_D$K.

B. Analysis of security goals of asymmetric key pair

In order to idealize the procedure, we abstract all the forwarding encrypted messages, modify the forms of keys, and change the expression of formulas in our BAN-like logic as below.

M$_1$: $W_i \triangleleft \{(@NK,NK@)\}$\_$_{KW}$
M$_2$: $M \triangleleft \{Data\}$\_$_{SDK}$
M$_3$: $M \triangleleft \{SDK\}$\_$_{W}$

We hope the procedure should archive several security goals when distributing (@NK,NK@), Data and SDK. Considering (@NK,NK@), there are three authentication goals described in formulas: $W_i \triangleleft (@NK,NK@)$, $W_i \triangleleft M \triangleleft (@NK,NK@)$ and $W_i \triangleleft (@NK,NK@)$. These mean that $W_i$ should see key pair (@NK,NK@). $W_i$ also believes that $M$ believes the key pair, and $W_i$ believes the key pair respectively.

The first goal can be proved easily. Applying the seeing rule SEE1 to message M$_1$ and assumption A$_1$ yields

S$_1$: $W_i \triangleleft (@NK,NK@)$.

Now we hence proved that the procedure has achieved the first goal. But we cannot keep carrying forward, as the next two goals need more information. The crux of proving the two remaining goals is to achieve $W_i \triangleleft M \vdash (@NK,NK@)$ using the message-meaning rule MM$_2$. Hence, we need a pair of asymmetric keys for the application of this rule. We assume that $M$ owns a public key $K_M$ and a corresponding private key $K_M^{-1}$ used for its signatures. As a result, we do the modification to the idealized procedure. Two new assumptions are added as below.

A$_1'$: $W_i \triangleleft \{K_M\} \rightarrow M$
A$_2'$: $M \triangleleft \{K_M\} \rightarrow M$

As the key pair (NK$_c$, NK$_a$) is produced by $M$, message M$_1$ is changed as below.

M$_1'$: $W_i \triangleleft \{(@NK,NK@)\} \_$_{KW}$

Again applying SEE$_1$ to message M$_1'$ and assumption A$_1'$ yields

S$_2$: $W_i \triangleleft \{(@NK,NK@)\} \_$_{KW}$.

Using the seeing rule SEE$_2$ to S$_2$ with assumption A$_1'$ also produces

S$_1$: $W_i \triangleleft (@NK,NK@)$.

Then employing the message-meaning rule MM$_2$ to S$_2$ and assumption A$_1'$ gets

S$_3$: $W_i \triangleleft M \vdash (@NK,NK@)$.

As BAN logic defaults to using Modus Ponens (MP) rule, adopting MP to S$_1$ and assumption A$_2'$ obtains

S$_4$: $W_i \triangleleft \{(@NK,NK@)\}$.

Utilizing the nonce-verification rule NV to S$_3$ and S$_4$ obtains

S$_5$: $W_i \triangleleft M \vdash (@NK,NK@)$.

Applying the jurisdiction rule J to assumption A$_5$ and S$_5$ yields

S$_6$: $W_i \triangleleft (@NK,NK@)$.

We can conclude that the original formulas can only archive the first goal $W_i \triangleleft (@NK,NK@)$. After our modification, the next two goals $W_i \vdash M \vdash (@NK,NK@)$ and $W_i \vdash (@NK,NK@)$ can also be satisfied. Hence, in the implementation of the access control solution in Fig.1 (step a.4), the data domain of Data packet should be changed from $\{\{NK_{encryption}, NK_{decryption}\}\}K_{pub-writer}$ to $\{\{NK_{encryption}, NK_{decryption}\}\}K_{symmetry}^{-1}K_{pub-writer}$.

C. Analysis of security goals of data key

The first series of goals should be archived are about data key SDK: $M \triangleleft SDK$, $M \vdash W_i \vdash SDK$ and $M \vdash SDK$.

Applying the seeing rule SEE$_4$ to message M$_3$ and assumption A$_5$ yields

S$_7$: $M \vdash SDK$.

In order to push further, we need to obtain the formula $M \vdash W_i \vdash SDK$ concerning with the message-meaning rule MM$_2$. As a result, we need to utilize the private key $KW^{-1}$ to encrypt SDK first. We also make a little change to the idealized procedure. Message M$_3$ is altered as below.

M$_3'$: $M \triangleleft \{\{SDK\}K_{symmetry}^{-1}\} \_$_{W}$

Utilizing the seeing rule SEE$_3$ to message M$_3'$ and assumption A$_4$ gains

S$_8$: $M \triangleleft \{SDK\}$\_$_{KW^{-1}}$.

Using the seeing rule SEE$_3$ to S$_8$ with assumption A$_1$ produces

S$_7$: $M \vdash SDK$.

Applying the message-meaning rule MM$_2$ to S$_8$ and assumption A$_1$ yields

S$_9$: $M \vdash W_i \vdash SDK$.

Adopting the MP rule to S$_7$ and assumption A$_15$ gets

S$_10$: $M \vdash \{SDK\}$.
Utilizing the nonce-verification rule NV to S9 and S10 gains
\[ S11: M \equiv W_i \equiv SDK. \]
Applying the jurisdiction rule J to S11 and assumption A6 yields
\[ S12: M \equiv SDK. \]
By this point, we have figured out the the formal proof of three goals for data key DK. As a result, in the implementation of the access control solution in Fig.1 (step a.7), the data domain of Data packet should be changed from \{DK\}NK_encryption to \{(DK)\}_{\text{pri-writer}}\cdot NK_encryption.

### D. Analysis of security goals of data

We focus on the other three goals for Data which represents the real data the writer wants to publish. They are \( M \prec Data, M \equiv W_i \equiv Data \) and \( M \equiv Data \). These denote that M should see Data, M also believes that \( W_i \) believes Data, and M believes Data respectively.

Similarly, we can easily apply the MP rule to assumption A15 and S7 to get
\[ S13: M \equiv SDK. \]
The seeing rule is also adopted to S13 and message M2 to gain
\[ S14: M \prec Data. \]
For further proof, we are required to gain the formula \( M \equiv W_i \triangleright Data \) springing from the message-meaning rule MM2. So we choose to apply the private key \( KW_{i}^{-1} \) to encrypt SData first. Message M2 is changed as below.
\[ \text{M2}: M \equiv \{\{\text{Data}\}_{KW_{i}^{-1}}\}_{SDK}. \]
Utilizing the seeing rule SEE4 to message M2’ and S13 gains
\[ S15: M \equiv \{\text{Data}\}_{KW_{i}^{-1}}. \]
Applying the seeing rule SEE2 to S15 and assumption A1 yields
\[ S14: M \equiv Data. \]
Employing the message-meaning rule MM2 to S15 according to assumption A1 gets
\[ S16: M \equiv W_i \triangleright Data. \]
Using the MP rule to S14 with assumption A16 produces
\[ S17: M \equiv \triangleright \{\text{Data}\}. \]
Applying the nonce-verification rule NV to S16 and S17 yields
\[ S18: M \equiv W_i \equiv Data. \]
Adopting the jurisdiction rule J to assumption A7 and S18 gets
\[ S19: M \equiv Data. \]
Now we have settled the formal proof of the three goals of data Data. As a result, in the implement of the access control solution in Fig.1 (step a.5), the data domain of Data packet should be changed from \{Data\}DK to \{(Data)\}_{\text{pri-writer}}\cdot DK.

### V. ANALYSIS OF READ OPERATION IN A CLOSED ENVIRONMENT

In this section, the specific read operation in a closed environment is presented. We adopt our BAN-like logic to this procedure and analyze it with some important security goals.

#### A. Read operation in a closed environment

We demonstrate the read operation procedure in a closed environment according to Fig.1(b) as below.

- **Message 1.** Reader_j → ACM: name_{Data}.
- **Message 2.** ACM → Reader_j: name_{Data}, \{Data\}_{DK}.
- **Message 3.** Reader_j → ACM: name_{DK}.
- **Message 4.** ACM → Reader_j: name_{DK}, \{DK\}_{NK_e}.
- **Message 5.** Reader_j → ACM: name_{NK_e}.
- **Message 6.** ACM → Reader_j: name_{NK_e}, H(K_1)...H(K_n).
- **Message 7.** Reader_j → ACM: name_{NK_e}, H(K_i).
- **Message 8.** ACM → Reader_j: name_{NK_e}, H(K_i), \{NK_{K_e}\}.K_i.

Reader_j sends the name of Data to ACM. ACM returns Data encrypted with data key DK. In order to decrypt it, Reader_j sends name_{DK} to ACM to request for data key DK. ACM gives Reader_j with DK encrypted with key NK_e. As a result, Reader_j still needs to deliver name_{NK_e} to get decryption key NK_{K_e}. ACM returns name_{NK_e} together with the hash values of K_1...K_N, denoted by H(K_1)...H(K_N). Then Reader_j recognizes the correct hash values of K_i in the hash value sequences and feeds it back to ACM with name_{NK_e}. Finally, ACM returns a new message added with NK_{K_e} encrypted with public key K_j. NK_{K_e} is the corresponding decryption key of encryption key NK_{K_e}, which is applied to decrypt \{DK\}_{NK_e}.

#### B. Analysis of security goals of data

We also idealize the read operation procedure as below.

- **M4:** R_i \prec \{Data\}_{SDK}.
- **M5:** R_i \prec \{SDK\} \triangleright SDK.
- **M6:** R_i \prec \{NK_{K_e}\} \triangleright R_i.

Three security goals described in BAN formulas need to be proved: \( R_i \triangleleft Data \), \( R_i \equiv W_j \equiv Data \) and \( R_i \equiv Data \).

Applying the seeing rule SEE2 to message M6 and assumption A1 yields
\[ S20: R_i \triangleleft NK_{K_e}. \]
Using Modus Ponens (MP) rule to S20 with assumption A9 produces
\[ S21: R_i \equiv NK_{K_e}. \]
Utilizing the seeing rule SEE3 to message M5 and S21 gains
\[ S22: R_i \triangleleft SDK. \]
Employing the Modus Ponens (MP) rule to S22 and assumption A15 gets
\[ S23: R_i \equiv SDK. \]
Applying the seeing rule SEE4 to message M4 and S23 yields
\[ S24: R_i \triangleleft Data. \]
Now the first goal has been proved. To prove the remaining goals, we also change message M4 slightly. The modification reason is similar with the one in Section IV-D. Supposing that the Data is produced by writer W_j, it first is encrypted by W_j’s private key KW_{j}^{-1}.

- **M4’:** R_i \prec \{\{Data\}_{KW_{j}^{-1}}\}_{SDK}.

Adopting the seeing rule SEE4 to message M4’ and S23 produces
\[ S25: R_i \prec \{\{Data\}_{KW_{j}^{-1}}\}. \]
Applying the seeing rule $\text{SEE2}$ to $S_{25}$ and assumption $A_1$ yields

$$S_{24}: R_i \triangleleft \text{Data}.\quad \Box$$

Utilizing the Modus Ponens (MP) rule to $S_{24}$ and assumption $A_{16}$ gains

$$S_{26}: R_i \models \triangledown \text{(Data)}.\quad \Box$$

Using the message-meaning rule $\text{MM2}$ to $S_{25}$ with assumption $A_1$ produces

$$S_{27}: R_i \models W_j \triangleright \triangledown \text{Data}.\quad \Box$$

Employing the nonce-verification rule $\text{NV}$ to $S_{26}$ and $S_{27}$ gets

$$S_{28}: R_i \models W_j \models \text{Data}.\quad \Box$$

Applying the jurisdiction rule $J$ to $S_{28}$ and assumption $A_6$ yields

$$S_{29}: R_i \models \text{Data}.\quad \Box$$

So far, we have proved three goals of data $\text{Data}$. So in the implement of the access control solution in Fig.1 (step b.2), the data domain of $\text{Data}$ packet should be changed from $\{\text{Data}\}DK$ to $\{\{\text{Data}\}K_{pri-writer}\}DK$.

VI. ANALYSIS OF WRITE AND READ OPERATION IN AN OPEN ENVIRONMENT

In this section, we show the specific read and write operation in an open environment. The BAN-like logic is applied to them. Then we analyze them with some important security goals.

A. Write operation in an open environment

Similar with Section IV-A, write operation procedure in an open environment in 1.(c) can be idealized as blew.

$$\begin{align*}
M_1: \quad & W \triangleleft \{\@NK,NK\}\triangledown w \\
M_2: \quad & M \triangleleft \{\text{Data}\}\triangledown DK \\
M_3: \quad & M \triangleleft \{\text{SDK}\}\triangledown NK \\
\end{align*}$$

As ACM can figure out key $K_w$ when it received the hash value of it, we can infer that

$$A'3: M \models \triangledown K_w \rightarrow W.\quad \Box$$

There are six goals needing to be proved: $W \triangleleft \{\@NK,NK\}$, $W \models M \models \{\@NK,NK\}$, $W \models \{\@NK,NK\}$, $M \triangleleft \text{Data}, M \models \triangledown W \equiv \text{Data}$ and $M \models \text{Data}$. With the help of assumption $A'3$, we can also prove them in almost the same way as we do in Section IV.

B. Read operation in an open environment

Like Section V-A, read operation procedure in an open environment in 1.(d) can be idealized as blew.

$$\begin{align*}
M_4: \quad & R \triangleleft \{\text{Data}\}\triangledown DK \\
M_5: \quad & R \triangleleft \{\text{SDK}\}\triangledown NK \\
M_6: \quad & R \triangleleft \{\text{NK}\}\triangledown K_r \\
\end{align*}$$

We can also deduce a new assumption.

$$A'4: M \models \triangledown R\rightarrow R.\quad \Box$$

Three goals should be verified: $R \triangleleft \text{Data}, R \models W_j \models \text{Data}$ and $R \models \text{Data}$. With assumption $A'4$, we can demonstrate these goals using the similar way in Section V.

VII. CONCLUSION AND FUTURE WORK

In this paper, we described a new approach to reasoning about an access control solution of NDN. Our work, which is inspired by the BAN logic, is a special logic to analyze different operation procedures in the access control solution. We idealized the procedures using the BAN-like logic and set several security goals to analyze them. These unsatisfied goals indicate that this access control solution cannot ensure the source of critical keys and data. This may lead the access control solution to be vulnerable to intruders. Then we did some improvement for the unsatisfied goals and made the new access control solution archive the important security goals. This method leads us in identifying mistakes and suggesting corrections. Our study helps to improve security and protect against various attacks for the access control solution. As for the future work, we would like to apply this method to other access control solutions of NDN.

Acknowledgement. This work was partly supported by Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of Things (No. ZF1213).

REFERENCES