An Integrated Authentication and Authorization Approach for the Network of Information Architecture

Mahdi Aiash and Jonathan Loo

School of Science and Technology
Middlesex University
London, UK

Abstract

Several projects propose an information centric approach to the network of the future. Such an approach makes efficient content distribution possible by making information retrieval host-independent and integration into the network storage for caching information. Requests for particular content can, thus, be satisfied by any host or server holding a copy. One well-established approach of information centric networks is the Network of Information (NetInf) architecture, developed as part of the EU FP7 project SAIL. The approach is based on the Publish/Subscribe model, where hosts can join a network, publish data, and subscribe to publications. The NetInf introduces two main stages namely, the Publication and Data Retrieval through which hosts publish and retrieve data. Also, a distributed Name Resolution System (NRS) has been introduced to map the data to its publishers. The NRS is vulnerable to masquerading and content poisoning attacks through invalid data registration. Therefore, the paper proposes a Registration stage to take place before the publication and data retrieval stage. This new stage will identify and authenticate hosts before being able to access the NetInf system. Furthermore, the Registration stage uses (cap)abilities-based access policy to mitigate the issue of unauthorized access to data objects. The proposed solutions have been formally verified using formal methods approach.

Keywords: Network of Information, Information Centric Networks, Formal Methods, Authentication, Authorization

1. Introduction

Information-Centric Networking (ICN) is an emerging paradigm envisaged by a growing body of researchers. ICN architectures leverage the role of information as the building block of the Internet architecture as opposed to the current end-host oriented paradigm. ICN architectures have better support for multicast, mobility, and security [1]. In ICN architectures, efficient information dissemination is expected to be supported by dispersing an information item in many network locations using in-network caches and Content Distribution Networks (CDNs) [2].

The Network of Information architecture is an ICN approach developed as part of the Scalable and Adaptive Internet Solutions (SAIL) project [3]. The SAIL NetInf project is centred around a well-defined set of architecture invariants (such as unique naming, location-independence and a strict information-centric service model) and puts particular emphasis on supporting multi-technology /multi-domain interoperability [4]. The project also takes into account developments elsewhere in ICN research (e.g., Content Centric Networking (CCN), Data-Oriented Network Architecture (DONA) and Publish-Subscribe Internetworking Routing Paradigm (PSIRP)) [5] [6] [4].

In NetInf, data objects such as web pages, articles or videos are named and identified using the Uniform Resource Identifier for Named Information (URI-ni) format [7], hence these objects are referred to as Named Data Objects (NDOs). The NetInf architecture is composed of three main components:

- **The Publishers:** These are NetInf nodes acting as source of NDOs and willing to make these objects accessible to subscribers.
- **The Subscribers (or Requesters):** These are NetInf nodes that request specific NDOs.
- **The NetInf System:** This is represented as a network of NetInf routing/forwarding nodes, spanning over the inter-domain topology along which

payload data is delivered. Three types of nodes are needed for the operation of the NetInf system: (1) cache-capable nodes to support the functionality of in-network caching of NDOs (2) Name-Based routers which route and forward NDOs towards subscribers (3) the Name Resolution System (NRS) is a distributed system which is aware of the network locations where an NDO might potentially be available for retrieval

Generally speaking, the operation of the NetInf architecture goes through two stages: the Publication Stage, where publishers publish their NDOs to the NetInf system. The Data Retrieval Stage, where subscribers request specific NDOs from the NetInf system. The requested NDOs will be then forwarded to towards the requesting subscribers. These two stages will be explained in section 2.

Currently, the research concentrates mainly on defining the NetInf overall architecture as well as the structure of the NetInf messages such as the Get-Req/ Get-Resp and Publish-Req/Publish-Resp (more details about these messages in Section 2). The security-related research is still at the stage of defining threat models, highlighting various possible attacks as in [3] and defining basic security measures as part of the URI-ni naming scheme [7]. Therefore, this paper introduces a new approach to address the authentication and authorization issues of implementing the NetInf architecture.

Our main concern here is the security of the Publication Stage, where publishers publish NDOs to the NetInf system. Another major concern is to address the issue of unauthorized access to published NDOs. For a secure publication, two requirements need to be verified namely, the authenticity of publishers and the validity of the published NDOs. Indeed, a malicious node might spoof another publisher ID and publish invalid NDOs. This is very similar to poisoning attacks against Domain Name Server (DNS) or routing tables [8]. To stop such attacks, we need to thwart masquerading threats; therefore, a pre-publication stage, called Registration Stage, is proposed in this paper. During the Registration Stage, both publishers and subscribers need to authenticate themselves with the NetInf system. Therefore, as part of the Registration Stage, we propose

a new authentication protocol based on the ID-Based Cryptography (IBC) [9]. The IBC helps to certify the messages sender as the real owner of the NDO that will update the NetInf system. The main advantage of using the IBC over traditional Public Key Infrastructure is that since the public key will be derived from the nodes' identifiers, IBC eliminates the need for a public key distribution infrastructure, details about IBC are in section 5.2.

To address the issue of an unauthorized access of NDOs, the paper will introduce an authorization and access control approach based on the (cap)abilities-based access control policy [10] [11]. The (cap)abilities-based access control policy has been used to secure the microkernel of the Valencia's Simple Tasker (VSTa) operating system. The proposed authorization (access control) approach is integrated with the proposed authentication protocol as core components of the Registration Stage. tool [12]. **In summary, the paper's contribution is to introduce an integrated authentication and authorization approach that achieves the following:**

- **To verify the identity of data publishers and subscribers through a novel ID-Based authentication protocol.**
- **To tackle the issue of unauthorized access to published data by using a cap(ability)-based access policy.**

The proposed security measures have been verified using a formal methods approach based on the Casper/FDR. The rest of this paper is organized as follows: the NetInf system is described in Section 2. Section 3 defines the security problem of the Registration Stage of the NetInf. Section 4 describes some related work. The proposed Registration Stage along with the authentication and authorization mechanisms are presented in Section 5. The paper concludes in the conclusion section.

2. An Overview of the NetInf

In NetInf architecture, publishers advertise potential publications in the NetInf system and serve the data contents upon receiving requests. The NetInf sys-

tem acts as a middleman between publishers and subscribers, and is involved in configuring the forwarding path for data delivery [3]. Three pairs of messages have been defined as part of the NetInf architecture:

- The GET-REQ/GET-RESP messages: The GET message is used by a requester to request an NDO from the NetInf network. A node responding to the GET message would send a GET-RESP that is linked to the GET request using the message-Id (msg-id) from the GET message.

- The PUBLISH-REQ/PUBLISH-RESP messages: The PUBLISH message allows a publisher to push the name and a copy of the NDO to the network. A node receiving a PUBLISH message may choose to cache the NDO according to local policy and availability of resources and returns PUBLISH-RESP message, otherwise, it may choose to forward the message to other nodes without sending the response message.

- The SEARCH/SEARCH-RESP messages: The SEARCH message allows the requester to send a set of query tokens containing search keywords. The node that receives the SEARCH message, will either respond if the NDO is in its own cache or forward the SEARCH message.

These messages are supposed to be transported over a Convergence Layer (CL) protocol. As stated in [4], no CL protocol has been defined yet, but any protocol that allows NetInf messages to be passed without loss of information can be used as a NetInf Convergence Layer (NetInf-CL) protocol. These three pairs of message define the transactions of the Publication and Data Retrieval Stages as follows:

1. **The Publish Stage:** Publishers publish their NDOs to the NetInf system by sending the PUBLISH-REQ message to the first hop node which might choose to cache the included information and responds with a PUBLISH-RESP message. Otherwise, it passes the PUBLISH-REQ to the next hop route. A node that caches NDO might update the NRS with the location of the NDO.

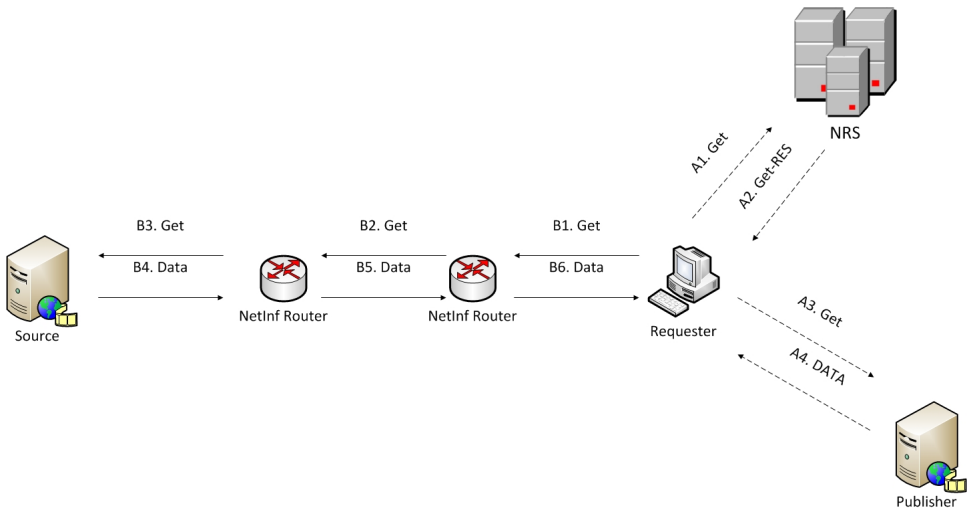

Figure 1: The NetInf Message Flow. The Name Resolution mode (dashed Arrows). The Name-Based Routing (solid Arrows)

2. **The Data Retrieval Stage:** As shown in Fig 1,The NetInf combines two modes for data retrieval:
   (a) The Name Resolution: In this mode, the publisher publishes an NDO using PUBLISH message with a Name Resolution Service (NRS). In this case, a requester will approach the NRS first (using the GET message) which will direct him to the information publisher.
   (b) The Name-Based Routing: In this mode, the GET message will be forwarded hop-by-hop between NetInf nodes until a cached copy of the requested NDO is found or the original publisher is reached.

3. Problem Definition

In NetInf, like other ICN architectures, the primary goal is to retrieve content from the network, regardless of their locations. As described in the previous section, the NetInf architecture has defined the required messages to publish and retrieve NDOs. However, there is no specified approach to secure these messages, rather, security in NetInf is mainly based on object naming scheme.

With the NetInf naming scheme, each NDO is given a unique identifier (ID) with cryptographic properties. Together with additional metadata, the ID can be used to verify data integrity, owner authenticity and several other security properties [13]. The scheme relies on proven mechanisms like cryptographic hashing and public-key certificate chains to reduce the risk of vulnerabilities. In this sense, NetInf's view of security is mainly focused on information security regardless of the security of the underlying transport protocols.

The authors believe that the fact that despite the migration of the predominant usage of the Internet from host-centric to the information-centric model, the underlying content delivery mechanism remains host-centric. As a consequence, some conflicts arise due to the usage of host-centric mechanisms in an information-centric networks, such as content identification and resolution, trust establishment and security. Therefore, we believe in the need for a hybrid security approach that addresses security at both information and infrastructural levels.

As explained in the Introduction section, one serious threat against the NetInf is when a fake publisher registers invalid NDOs with the NSR during the Publication Stage. Obviously, this poisons the whole system, leads to invalid responses to subscribers' requests which is considered as a form of Denial of Service (DoS) attacks. Another threat is when unauthorized users get access to data due to the lack of access control and authorization mechanisms. The solution presented in this paper strives to address these issues by holding publishers and subscribers accountable for their actions and making sure that NDOs could only be published and accessed by identified parties. To achieve this, our approach proposes that publishers and subscribers need initially to go through a Registration Stage where they will be authenticated and given security tokens that define their permissions. The proposed authentication mechanisms in the Registration Stage is based on the ID-Based Cryptography approach, while the proposed authorization mechanism is based on the (cap)ability-based access control policy. After a successful registration, publishers and subscribers could use the NetInf system to publish and request NDOs.

4. Related Work

4.1. ID-Based Cryptography (IBC):

The IBC is a cryptographic scheme was first proposed by Adi Shamir [9]. The scheme enables users to communicate securely and verify each other's signature without exchanging public or private keys. However, the scheme requires the presence of Trusted Key Generation (TKG) centres.

IBC's Operation: Unlike the normal Public Key Infrastructure (PKI) where a TKG randomly generates pairs of public/private keys, each node in IBC chooses its identifier (address or name) as a public key. Practically, any publicly known information that uniquely identifies the node could be used as a public key. The TKG generates the corresponding private key and securely distributes it to the node. When a node (A) wants to communicate with another node (B), node A will sign the message using its private key and encrypt the result with the node B's public key. Upon receiving the message, node B will decrypt the message using its private key and verify the signature using node A's public key. The IBC represents an efficient and easy to implement system which removes some of the overhead encountered in PKI for key management and digital certificate issuance/revocation. However, the security of the IBC is based on the secrecy of the private key. To deal with this issue, the node needs to combine additional information such as timestamps to their identifiers when generating the public key. This procedure will guarantee a periodic update of the public key. However, it introduces a key-management problem where all users must have the most recent public key for the node.

4.2. Authorization and Access Control

Most computer security uses the access control mode shown in fig 2, this model comprises the following elements [14]

- Principals/Subjects: These are the source of access requests.
- Requests to perform operations on objects.

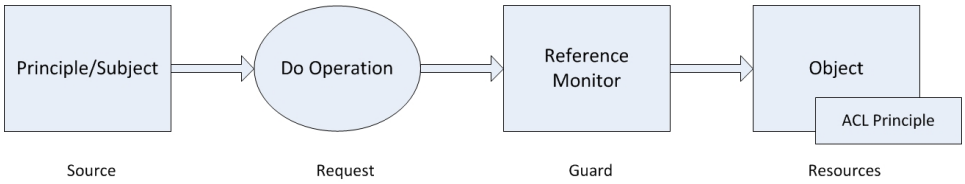

Figure 2: The Fundamental Model of Access Control

- A reference monitor: This is a guard for each object that examines access requests for the object and decides whether to grant it.
- Objects: These represent resources such as files, devices, or processes.

The reference monitor bases its decision on the principal making the request, the operation in the request, and an access rule that controls which principals may perform that operation on the object. To do its work the monitor needs a trustworthy way to know both the source of the request (via authentication process) and the access rule. Obtaining the source of the request is called authentication; interpreting the access rule is called authorization. Thus authentication answers the question "Who said this?", and authorization answers the question "Who is trusted to access this?". Usually the access rule is attached to the object; such a rule is called an Access Control List or ACL. For each operation the ACL specifies a set of authorized principals, and the monitor grants a request if its principal is trusted at least as much as some principal that is authorized to do the operation in the request [14].

In the context of ICNs, access control policies are needed to guarantee that NDOs could be published by authorized sources and the access to these NDOs are only given to authorized subscribers.

4.3. Verifying Security Protocols using Casper/FDR:

Previously, analysing security protocols used to go through two stages. Firstly, modelling the protocol using a theoretical notation or language such as the CSP [12]. Secondly, verifying the protocol using a model checker such as Failures-Divergence Refinement (FDR) [15]. However, describing a system or

Table 1: THE HEADERS OF CASPER'S INPUT FILE

The Header	Description
# Free Variables	Defines the agents, variables and functions in the protocol
# Processes	Represents each agent as a process
# Protocol Description	Shows all the messages exchanged between the agents
# Specification	Specifies the security properties to be checked
# Actual Variables	Defines the real variables, in the actual system to be checked
# Functions	Defines all the functions used in the protocol
# System	Lists the agents participating in the actual system with their parameters instantiated
# Intruder Information	Specifies the intruder's knowledge and capabilities

a protocol using CSP is a quite difficult and error-prone task; therefore, Gavin Lowe has developed the CASPER/FDR tool to model security protocols, it accepts a simple and human-friendly input file that describes the system and compiles it into CSP code which is then checked using the FDR model checker. Casper/FDR has been used to model communication and security protocols as in [16], [17]. The CASPER's input file that describes the systems consists of eight headers as explained in Table 1.

5. The Proposed Solution

As discussed earlier, we propose a new stage to take place before the Publication and Data Retrieval stages. This section discusses our proposal of using the IBC protocol to secure the Registration procedure of the NetInf.

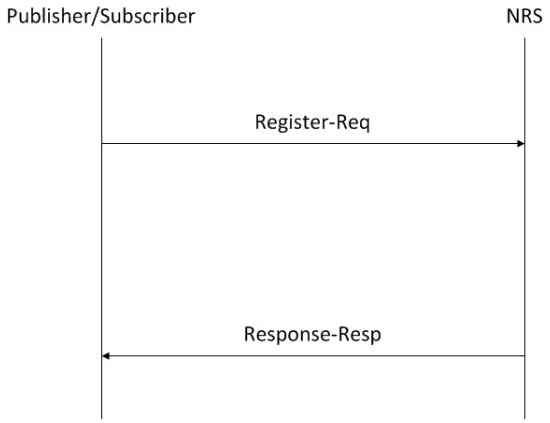

Figure 3: The Registration Stage

5.1. System Definition

In NetInf, data sources publish NDOs by registering a name/locator binding with the a NRS using the Publish message or announcing routing information in a routing protocol. Any NetInf node holding a copy of the NDO can optionally register the copy with the NRS. Subscribers will approach the NRS requesting for a specific NDO, and the NRS will first resolve the NDO into a set of available locators and then retrieve the a copy of the data from best available source.

In order to provide a secure data publication and retrieval, we advocate the need for a registration stage during which both publishers and subscribers need to identify themselves to the NRS and acquire a security tokens that define their privileges and access rights. Two types of security tokens namely, Object and Subject tokens are generated by the NRS. During the Registration Stage, a node needs to disclose its role (publisher, subscriber or both) and after the authentication process, it will receive corresponding tokens (subject, object or both). The security tokens will define security levels for NDOs as for Objects Tokens (ObjToken) and for subscribers as for Subject Tokens (SubToken). The rules of access will be checked and enforced by the NRS which will be acting as a Reference Monitor, more details about the authorization and access control approach is in section 5.3.

5.2. The Proposed Authentication Protocol:

As shown in Fig 3, and based on the notations in Table 2, the secure Registration Stage using the IBC goes as follows:

`Msg1. TKG→ Pub : {SK(Pub)}{K1}`

`Msg2. TKG → NRS : {SK(NRS)}{K2}`

The TKG provides the two communicating parties (Pub, NRS) with their private keys SK(Pub), SK(NRS) in messages 1 and 2. These messages are encrypted using the pre-shared secret keys K1, K2, respectively.

`Msg3. Pub → NRS : {Reg-Req}{PK(NRS)}, {h(Reg-Req)}{SK(Pub)}`

The Pub sends a Register-Request (Reg-Req) packet which includes information about the node role (Pub or Sub) and a one-time message ID in Msg3. The content of this message is encrypted using the NRS's public key (which is publicly known) and digitally signed using the private key of the Pub.

`Msg4. NRS → Pub : {Reg-Resp, ObjToken}{PK(Pub)}, {h(Reg-Resp, ObjToken)}{SK(NRS)}`

Upon receiving msg3, the NRS will use its private key SK(NRS) to decrypt the message and then verify the signature using the Pub's public key PK(Pub). Finally, the NRS will hash the included Reg-Req and compare the result with the received signed value. Only if the two values are equal, the NRS composes a Register-Response (Reg-Resp) packet as msg4 which includes the received message ID (Msg-ID) and an Object Token (ObjToken). This message is encrypted using the Pub's public key and digitally signed using the NRS's private key. The Pub will check the included Msg-ID and only when the check succeeds, the Pub authenticates the NRS and accepts the token. The protocol's steps are shown in Fig 4.

It is worth to point out that the same proposed protocol should be used for Registering subscribers before accessing NDOs. The only difference in this case

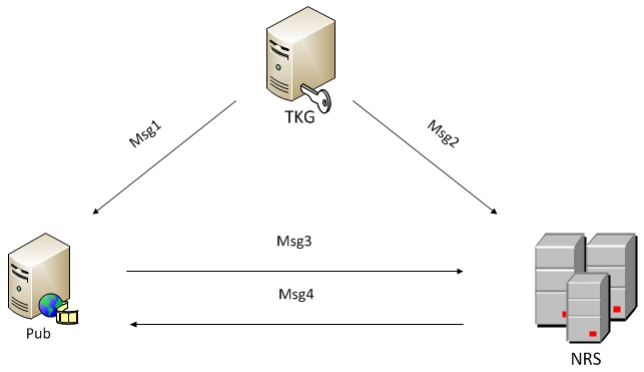

Figure 4: The Proposed Security Protocol

will be the use of Subject Token (SubToken) instead of the Object Token. At the end of the Registration Stage, the NRS will have a list of all authorized subscribers and publishers.

5.2.1. Formal Analysis Using Casper/FDR:

To formally analyse the proposed solution, we simulate the system using Casper/FDR tool. The eight headings of the simulated system are described below.

The #Free Variables section defines the variables and functions that are used in the protocol. The term "Free Variables" refers to the fact that these variables will be represented by instances of actual values when running the protocol. For instance, the variables na, nb, seq2, n1 are of type Nonce. The functions PK and SK return an agent's public key and private key, respectively. These functions will be defined later in the #Functions. The "InverseKeys" keyword defines the keys that are inverses of one another like PK and SK.

```
#Free variables
Pub, NRS : Agent
na, nb, seq2, n1 :  Nonce
MID: MessageID
SubToken:  SubjectToken
ObjToken:  ObjectToken
```

Table 2: Notation

The Notation	Definition
TKG	The Trusted Ticket Granting
SK(Pub), SK(NRS)	The Private keys of the Pub, NRS, respectively. These keys are derived by the TKG
K1, K2	Pre-shared keys to secure the connections between the TKG and Pub, NRS
Pub	The data source or the publisher of NDO
NRS	The Name Resolution Service which holds the name/location binding for NDOs
ObjToken	A security token will be attached to the published NDO
SubToken	A security token will be attached to the subscribers
MID	Message-ID
h(m)	Hash value of the message (m)
{m}{K}	The message (m) being encrypted with the key (K)

```
PK: Agent → PublicKey
SK: Agent → PrivateKey
K1, K2:  PreSharedKey
TKG: Server
m,m2, Ack:  Messages
InverseKeys = (PK,SK), (K1, K1),(K2, K2)
h :  HashFunction
EIDPre:  EIDPrefix
hash1:  HashValues
```

The #Processes heading defines each involved agent in the protocol as a CSP process. The keyword "knows" defines the knowledge that the agent in question is expected to have at the beginning of the protocol run. In our system, INITIATOR, RESPONDER and SERVER are the names of the process representing the Publisher, the Name Resolution Service and the Trusted Ticket Granting, respectively. The values within the brackets and after the "knows" keyword define the agents' initial knowledge.

```
#Processes
INITIATOR(Pub, NRS,TKG, K1,nb, m, MID) knows PK(Pub), PK(NRS), SK(Pub)
RESPONDER(NRS,TKG, K2, m2, SubToken, ObjToken) knows PK(Pub), PK(NRS),
SK(NRS)
SERVER(TKG, Pub, NRS, K1, K2, na) knows PK, SK(Pub), SK(NRS)
```

The #Protocol description heading defines the system and the transactions between the entities. It is worth pointing out that for security simulation we need to explicitly define the security parameters. Therefore, we mention the security-related contents such as the message ID (MID) and the object Token (ObjToken) in msg 3, 4. Where (m) and (m1) refer to Register-Request and Register-Response packets, respectively.

```
#Protocol description
0.  -> Pub :  NRS, TKG
1.  TKG -> Pub :  {SK(Pub)}{K1}
2.  TKG -> NRS : {SK(NRS)}{K2}
```

```
3.  Pub -> NRS : {m,Pub, MID}{PK(NRS)}, {h(m, Pub, MID)}{SK(Pub)}%z
[decryptable(z, PK(Pub))]
4.  NRS -> Pub :  {m2, MID, ObjToken}{PK(Pub)}, {h(m2, MID, h(SubToken))}{SK(NRS)}%w
[decryptable(w, PK(NRS))]
```

The security requirements of the system are defined under the # Specification heading. The lines starting with the keyword **Secret** define the secrecy properties of the protocol. The `Secret(NRS, MID, [Pub])` specifies the MID as a secret between the Pub and the NRS. The lines starting with **Agreement** define the protocol's authenticity properties; for instance `Agreement(NRS, Pub, [MID])` specifies that, the NRS is correctly authenticated to the Pub using the message ID (MID). The `WeakAgreement(Pub, NRS)` assertion could be interpreted as follows: if Pub has completed a run of the protocol with NRS, then NRS has previously been running the protocol, apparently with Pub.

```
#Specification
Secret(NRS, MID, [Pub])
Secret(NRS, ObjToken, [Pub])
WeakAgreement(Pub, NRS)
WeakAgreement(NRS, Pub)
Agreement(NRS, Pub, [MID])
```

The #Actual Variables section defines the types of variables, used in the actual system. These are defined in a similar way to the #Free Variables.

```
#Actual variables
pub, nrs, Mallory :  Agent
Na, Nb, Seq2, N1 :  Nonce
mID: MessageID
k1, k2:  PreSharedKey
tkg:  Server
InverseKeys = (k1, k1),(k2, k2)
EIDpre:  EIDPrefix
M, M2, ack:  Messages
haash1:  HashValues
```

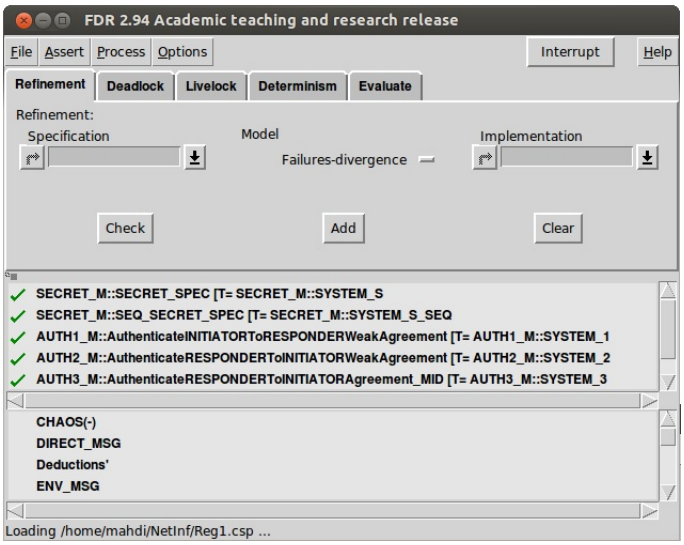

Figure 5: The Verification Result

```
subToken:  SubjectToken
objToken:  ObjectToken
```

The functions used by the agents in the #Free Variables section will be defined under the #Functions heading.

```
#Functions
symbolic SK, PK
```

The # Intruder Information heading specifies the intruder identity, knowledge and capability. The first line identifies the intruder as Mallory, the intruder knowledge defines the Intruder's initial knowledge, i.e., we assume that the intruder knows the identity of the participants, all public keys, its own private key and can fabricate Register-Request and Register-Response messages.

```
#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Pub, NRS, Mallory, PK ,SK(Mallory), M, M1}
```

After generating the CSP description of the systems using Casper and asking FDR to check the security assertions, no attack was found against the proposed solution as shown in Fig 5.

5.2.2. Security Analysis:

Despite the fact that no attack has been discovered against the proposed solution in section 5.2.1, this result needs to be considered carefully. The formal verification result is based on the system defined in 5.1. In this system, it is assumed that the Pub knows the authoritative NRS in its network or domain. In a very similar way to the current Domain Naming System (DNS), where clients are preconfigured with the authoritative DNS server. However, we simulated the case when the Pub is not sure of the identity of its authoritative NRS. The following attack against the Secret(NRS, MID, [Pub]) and Agreement(NRS, Pub, [MID]) is discovered.

1a. TKG $\rightarrow$ *I_Pub :* $\{SK(Pub)\}\{K1\}$

1b. I_TKG $\rightarrow$ *Pub :* $\{SK(Pub)\}\{K1\}$

2a. TKG $\rightarrow$ *I_NRS :* $\{SK(NRS)\}\{K1\}$

2b. I_TKG $\rightarrow$ *NRS :* $\{SK(NRS)\}\{K1\}$

3a. Pub $\rightarrow$ *I_Pub :* $\{M, Pub, MID\}$ $\{PK(Mallory)\}$, $\{h(M, Pub, MID)\}\{SK(Pub)\}$

3b. I_Pub $\rightarrow$*NRS :* $\{M2, Pub, MID\}\{PK(NRS)\}$, $\{h(M, Pub, MID)\}\{SK(Pub)\}$

4. NRS $\rightarrow$ *I_Pub :* $\{M2, MID, objToken\}$ $\{PK(Pub)\}$, $\{h(M2, MID, h(subToken))\}$ $\{SK(NRS)\}$

The intruder knows MID

Where the notations I_ NRS, I_Pub and I_TKG represent the case where the Intruder impersonates the NRS, Pub and TKG, respectively. This is an active Man-in-the-Middle attack; the Intruder intercepts and replays messages 1 and 2. Since the Pub is not sure of the identity of the NRS, the intruder manages to impersonate the NRS and fools the Pub to use its (rather than the NRS's) public key to encrypt message 3a. Consequently, the message ID will be compromised, and the Pub will run the protocol mistakenly believing it is with the NRS, while in reality it is with the Intruder. As a consequences of this attack, the intruder will be able get the name/location binding at the publication stage and mix them in away to deny subscribers from getting the requested data and hence launch a DoS attack.

There are two ways to stop such attack: firstly using an out-of-band approach in which the Pubs should be pre-configured to use the an authoritative NRS in its domain or network. This could be simply achieved during the network configuration in a similar way to configuring the default DNS server or the default gateway in a network. Secondly, by requesting the NRS to explicitly identify and authenticate itself to the Pub via providing a digital certificate that could be verified by a trusted third party such as the TKG or a Certificate Authority (CA).

5.3. The Authorization and Access Control

During the Registration Stage, once a party (subscriber or publishers) is authenticated, the NRS will generate a security token. Two types of tokens are generated: Object Tokens, attached with the published NDOs and Subject Tokens attached with subscribers. These tokens define objects and subjects abilities. An ability is represented as a dot-separated sequence of numbers, called a label. So, an ability is a string $.i_1.i_2.i_3.....i_n$ for some value n where $i_1, i_2, i_3....., i_n$ are integers. Examples of abilities are .1.2.3, .4, or 10.0.0.5. Upon successful registration, both NDOs (objects) and subscribers (subjects) will be given labels (abilities). Access for an NDO is given if the NDO's label is a prefix of the subscriber's label. For instance, an NDO with a label ".3" could only be accessed by subscribers with abilities like ".3.1", ".3.2.3", ".3.1.2"...etc. This way, whenever an authenticated subscriber requests an NDO, he needs to present the right label that confirms his right to access the NDO.

With the proposed protocol in section 5.2, labels are generated by the NRS so subscribers can not promote themselves to access other NDOs. Furthermore, to maintain the integrity of the labels and making sure they have not be tampered with, labels are integrated in a security tokens (SubToken, ObjToken) which are hashed and digitally signed by the NRS. Additionally, the security tokens are time stamped and have expiry date after which new tokens are needed. When generating the token, it should be noted that no Subtokens have a validity period longer than that of the corresponding ObjToken. Using the time stamp and the

expiry time will minimize the risk of a both active and passive replay attacks.

6. Conclusion

Building a scalable information-centric architecture involves several challenges. This includes the development of an information model and a naming framework which support efficient information dissemination with improved security properties. The NetInf is a promising architecture for data dissemination and retrieval that is based on the Publish/Subscribe model. In this model, publishers publish their data (through the Publication stage) to the NRS system which then launch these data to subscribers upon request (through the Data Retrieval Stage). This papers explains how the Publication Stage might be vulnerable to masquerading and content poisoning attacks which might happen when an unauthenticated node publishes invalid data to the system. The paper also highlights the issue of an authorized access to published data. To address these challenges, an integrated authentication and authorization approach is proposed in the paper. While the proposed authentication protocol is based on the IBC protocol and achieves mutual authentication between publishers and the NetInf system, the proposed authorization approach is based on cap(ability) access policy. The proposed approaches have been formally verified using formal method approach.

References

[1] N. Fotiou, F. Marias, C. Polyzos, Access control enforcement delegation for information-centric networking architectures, in: Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking, ICN '12, ACM, New York, NY, USA, 2012, pp. 85–90. doi:10.1145/2342488.2342507.
URL http://doi.acm.org/10.1145/2342488.2342507

[2] T. Sipat, Z. Al-Qudah, R. Michael, Content delivery networks: Protection

or threat?, in: ESORICS, Lecture Notes in Computer Science, Springer, 2009, pp. 371–389.

[3] T. Edwall, The network of information: Architecture and applications, Tech. rep., SAIL Scalable and Adaptable Internet Solutions (2013).
URL http://www.sail-project.eu/wp-content/uploads/2011/08/SAIL_DB1_v1_0_final-Public.pdf

[4] D. Kutscher, S. Farrell, E. Davies, The netinf protocol, Tech. rep., Network Working Group (2013).
URL http://tools.ietf.org/id/draft-kutscher-icnrg-netinf-proto-01.txt

[5] K. Teemu, C. Mohit, C. Byung-Gon, E. Andrey, K. Hyun, S. Scott, S. Ion, A data-oriented (and beyond) network architecture, in: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '07, 2007, pp. 181–192. doi:10.1145/1282380.1282402.

[6] X. Zhang, T. Niu, F. Lao, Z. Guo, Topology-aware content-centric networking, in: SIGCOMM, 2013, pp. 559–560. doi:10.1145/2486001.2491729.

[7] H. Baker, R. Stradling, S. Farrell, D. Kutscher, B. Ohlman, The named information (ni) uri scheme: Optional features, Tech. rep., Network Working Group (2012).
URL http://tools.ietf.org/html/draft-hallambaker-decade-ni-params-03

[8] M. Gregg, Certified Ethical Hacker, Que Publishing, 2006.

[9] A. Shamir, Identity-based cryptosystems and signature schemes, in: CRYPTO 84 on Advances in cryptology, Springer-Verlag, 1985.

[10] D. Gollmann, Computer Security, 3rd Edition, Wiley, 2011.

[11] H. C. Chen, A multi-issued tag key agreement with time constraint for homeland defense sub-department in nfc environment, Journal of Network and Computer Applications (2014) 88–98.

[12] G. Lowe, P. Broadfoot, C. Dilloway, M. Hui, Casper: A compiler for the analysis of security protocols, Oxford (2009).

[13] C. Dannewitz, J. Golic, B. Ohlman, B. Ahlgren, Secure naming for a network of information, in: INFOCOM, 2010, pp. 1–6. doi:10.1109/INFCOMW.2010.5466661.

[14] C. Paquet, Implementing Cisco IOS Network Security, Cisco Press, 2009.

[15] Formal Systems, Failures-divergence refinement: fdr2 user manual and tutorial (1993).

[16] M. Aiash, A formal analysis of authentication protocols for mobile devices in next generation networks, Concurrency and Computation Practice and Experiencedoi:10.1002/cpe.3260.

[17] M. Aiash, A novel security protocol for address resolving in the location/id split architecture, in: Network and System Security NSS2013, Vol. 7873, 2013, pp. 68–79.