Detecting and Mitigating Repaying Attack in Expressive Internet Architecture (XIA)

Beny Nugraha[1], Rahamatullah Khondoker[2], Ronald Marx[2], and Kpatcha Bayarou[2]

[1] *Department of Electrical Engineering, Mercu Buana University, Jakarta, Indonesia*
[2] *Fraunhofer SIT, Rheinstr. 75, Darmstadt, Germany*

Received: October 7, 2014;
Accepted: November 10, 2014

Abstract

Several Future Internet (FI) architectures have been proposed to address the problems of the Internet including flexibility (so called IP bottleneck), host-based addressing (addressing a host rather than the content itself), and security. In the beginning of this article, we survey the security solutions of seven FI architectures, namely XIA, RINA, NENA, SONATE, Mobility-First, NDN, and SONATE, based on literatures, prototypes, and demonstrations. It has been found that none of the architectures can fulfill all of the security goals: confidentiality, authentication, integrity and availability. Further in this article, we focus on eXpressive Internet Architecture (XIA) as it is the most secure and open-source Content-Centric Network (CCN). CCN is claimed by the Future Content Networks (FCN) Group to be the Future Internet. However, XIA does not have any mechanisms to mitigate the replaying attack, thus, this article proposes and implements a solution to mitigate it. Several existing solutions have been analyzed to derive the requirements for the proposed solution. By implementing the proposed protocol, XIA is now able to mitigate all of the reviewed network attacks. The evaluation shows that the proposed solution is more secure and less complex over the existing solutions.

Journal of ICT, Vol. 2, 151–186.
doi: 10.13052/jicts2245-800X.225

Keywords: Replaying Attack, Session Key, eXpressive Internet Architecture (XIA), Future Internet (FI), CCN.

1 Introduction

Current Internet faces challenges such as inability to provide flexibility - changing of protocol in one layer requires another changing of protocol in another layer, and inability to provide intrinsic security - a security mechanism is added to counter a new threat, it is not integrated. The problems arise mainly because of the design principles of the Internet that are hard to be changed (cannot provide flexibility) [1]. Several Future Internet (FI) architectures have been developed to solve these problems. There are two design methods that can be followed for developing an FI Architecture: "clean slate" or "evolutionary". In the clean slate approach, the architecture is designed from the scratch, meanwhile new design components are added to the existing architecture in the evolutionary approach. In the early part of this article, we analyze the security mechanisms of seven future network architectures, namely, eXpressive Internet Architecture (XIA) [2], Recursive Inter-Network Architecture (RINA) [3], Service Oriented Network Architecture (SONATE) [4], Netlet-based Node Architecture (NENA) [5], MobilityFirst [6], NEBULA [7], and Named Data Networking (NDN) [8]. We selected them as they are mature (established in 2009 or 2010), and they have either a demonstration or a prototype or both. NENA and SONATE both use clean slate approach. Meanwhile the other five, XIA, RINA, MobilityFirst, NDN, and NEBULA, all of them use evolutionary approach.

It is indispensable for a newly deployed Internet architecture to fulfil the security requirements. In this article, we discuss the following security goals: confidentiality, integrity, availability, and authentication (defined in Section 2.1). The methodology of the research are, first, specify the threats against each of the security goal (discussed in Section 2.2), second, identify the available security mechanisms of each architecture by analyzing its literatures, prototype, and demonstration (described in Section 2.5) and third, conclude which of the threats can be mitigated by which of the security mechanisms (depicted in Section 2.5).

The rest of this article is organized as follows: the result of the analysis of the FI's security mechanisms is presented and compared in Section 2 and Section 3, respectively. Afterwards, in Section 4, the existing solutions for replaying attack are analyzed to derive the requirements for the proposed solution which is described in Section 5. Based on the derived requirements, the solution is proposed in Section 6. The implementation and evaluation of the proposed solution are described in Section 7 and 8 respectively. Finally, the conclusion and the future work are discussed in Section 9.

2 Methodology for the Survey

The security mechanisms of an FI can be analyzed in one of the two following ways: attack-centric and system-centric [9]. Attack-centric means the attacks on a system (i.e., using attack trees) are modeled, and system-centric means the system itself (i.e., using STRIDE methodology) is modeled. Our methodology is a combination of attack-centric and system-centric approach since it analyzes both the architectures and the attacks, and it is able to provide a better view of the architectures vulnerability to attacks than to follow just one approach. Our methodology works as follows:

1. Defining the network security goals.
2. Specifying attacks that inhibit the network security goals and then analyzing the counter mechanisms for each attack. This is the attack-centric approach.
3. Selection of future network architectures.
4. Analyzing the security solutions of each of the architectures. This the system-centric approach.
5. Matching the counter mechanisms for each attack and the security mechanisms of the future network architectures in order to find out the vulnerability in each architecture.

The details of each item will be discussed as follows:

2.1 Defining Security Goals

The general goal of network security is to give people freedom to enjoy computer networks without fear of compromising their rights and interests [10].

In order to achieve that goal, four specialized goals of network security have been identified. These four goals are the following:

Confidentiality Means that the message that is sent by the sender has to be intended for the receiver only, for the others, this message must be worthless.

Integrity Means that the received message must be the same as the original message.

Availability The services that are accessible by the Internet (i.e., web services, remote machines, networks, etc.) must be available all the time for its authorized users only.

Authentication Only the authorized user is able to send a message and the receiver is able to proof the sender's identity.

2.2 Specifying Threats Against Achieving the Goal

Several network security threats which work against achieving the goal have been identified. They are as follows:

2.2.1 Threats against confidentiality

Snooping and traffic analysis attacks are considered as possible threats against confidentiality. In snooping, the aim of the attacker is to get the database of an authorized user or the packets flowing in a network. The attacker can perform several action to undergo snooping attack, examples of the action are: 1, by using ping-type programs (ICMP ping, TCP ping) to identify active hosts on the network and to further locate potential targets and, 2, by using TCP/UDP port scanning for detecting the target operating system [11]. Snooping attack can be mitigated by having a data encryption mechanism to protect the packets.

In order to perform the traffic analysis attack, the attacker intercepts and examines messages to extract information from the traffic patterns in a communication. The greater the number of packets that can be obtained, the more the information that can be inferred from the traffic. The security mechanism to prevent this attack is to have a mechanism to conceal the identity of the users, therefore, an attacker cannot determine at which point or node he should watch the traffic.

2.2.2 Threats against integrity

Modification and repudiation attacks are threats against integrity.

Modification attack involves deletion, insertion, or alteration of information in an unauthorized manner that is intended to appear genuine to the user [12]. The counter mechanisms for this attack are to hash the message or to have a digital signature, therefore, the receiver will be able check the correctness of the message.

Repudiation is a process in which the sender or the receiver cannot prove that a transaction has taken place between them, either one or both of them can deny that they are sending or receiving the data [12]. Repudiation attack can be mitigated by having a digital signature mechanism in collaboration with a trusted third party to create a non-repudiation message.

2.2.3 Threats against confidentiality

Denial of Service (DoS) attack is a threat against availability. This attack can deny access to information, applications, systems or communications. An example of DoS attack is to flood the traffic with bursts of packets [13]. DoS can be prevented by having a flow control or bandwidth allocation mechanism. Therefore, only the authorized packets that can flow in the traffic.

2.2.4 Threats against authentication

The attacks that are against authentication are: man-in-the-middle, reflection, masquerading, and replaying attack.

The attacker stays in between the sender and the receiver, then observes or modifies the traffic in the man-in-the-middle attack [14]. Man-in-the-middle attack can be prevented by having a digital signature mechanism in order to authorize the real authorized users.

In reflection attack, the attacker has an objective to pretend that he is an authorized user by sending the response from the real authorized user to the target [15]. Reflection attack can also be mitigated by performing a digital signature mechanism.

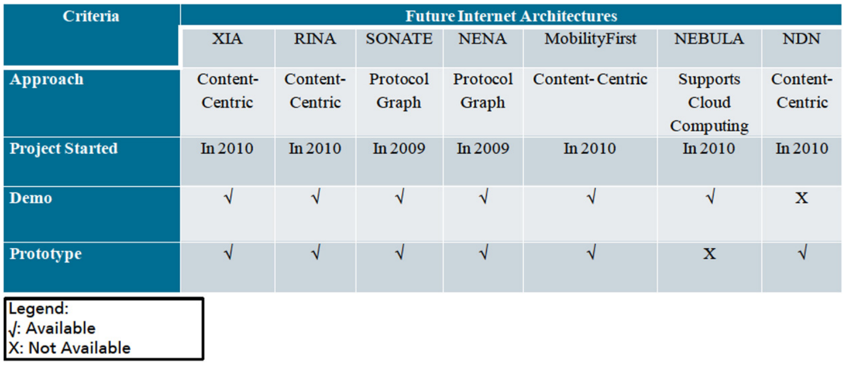
In masquerading, the attacker pretends to be an authorized user of a system in order to gain access to it, and then modifies the [16]. Masquerading attack can be mitigated by having an anonymous connection or having a good user authentication process.

Replaying attack occurs when information is captured and then replayed later, in different session, in order, for example, to gain the trust of other users [17]. Replaying attack can be prevented by having a marker to bind one communication session, example of the marker are session key or random number which will be generated differently each session. By performing this, the messages in one session will be always different than the messages in another session.

To summarize, there are nine attacks to be reviewed in this article. The future network architectures should be able to mitigate all of the attacks intrinsically to fulfill the security requirements.

2.3 Selection of FI Architectures

The selection of future network architectures is done by considering the maturity of the architecture (architectures that established from year 2011 onward will not be considered), the availability of the demonstration, and the prototype as shown in Figure 1. We choose seven architectures to review, they are eXpressive Internet Architecture (XIA), Recursive Inter-Network



Criteria	Future Internet Architectures						
	XIA	RINA	SONATE	NENA	MobilityFirst	NEBULA	NDN
Approach	Content-Centric	Content-Centric	Protocol Graph	Protocol Graph	Content- Centric	Supports Cloud Computing	Content-Centric
Project Started	In 2010	In 2010	In 2009	In 2009	In 2010	In 2010	In 2010
Demo	√	√	√	√	√	√	X
Prototype	√	√	√	√	√	X	√

Legend:
√: Available
X: Not Available

Figure 1 FI Architectural approaches

Architecture (RINA), Service Oriented Network Architecture (SONATE), Netlet-based Node Architecture (NENA), MobilityFirst, NEBULA, and Named Data Networking (NDN) as they are mature (the projects started in 2009 or 2010), and they have either a prototype or a demonstration or both.

2.4 Asking and Receiving Expert's Feedback

After surveying the literatures for each architecture, we got the result described in the following section 2.5. The result of each architecture has been sent to the founders of the architecture for review such as we asked the feedback for the XIA architecture only from the XIA inventors. We received feedback for all of the seven architectures, this step is needed to check whether our analysis is correct or not.

2.5 Analyze the Security Solutions of Every Architecture

The first two architectures that are analysed are SONATE and NENA. The aim of both SONATE and NENA is to use a customized protocol graph (similar to a TCP/IP or UDP/IP network stack) based on the requirements from the application. However, they differ in terms of "when the composition is accomplished?". Whereas in SONATE, the composition is done during runtime of communication association, the composition in NENA is accomplished during the design time of creating new protocols.

2.5.1 SONATE

In SONATE, the services provided by building blocks (the implementation of a protocol or a mechanism like CRC, retransmission, etc.) are selected and composed by a composition algorithm to create a protocol graph during runtime based on the requirements from the application, constraints from the administrator, and networks [4]. Therefore, the security mechanisms in SONATE are depends on the application's requirements. The security mechanisms are [19]:

1. SONATE is able to select building blocks (BB) that:
    a. enable data encryption (i.e., data encryption micro protocol) [19],
    b. provide data authentication (i.e., digital signature, MAC) [19,20], and
    c. provide flow control service [20].
2. Each communication session is bound by one protocol graph [19].

By analyzing the above security mechanisms, we can conclude that SONATE is vulnerable to the traffic analysis attack and the masquerading attack since SONATE does not provide anonymous communication and an attacker only need to know the port and the address of the target to initiate both attacks. SONATE also cannot mitigate the repudiation attack because it does not have a trusted third party to prove a communication between two users has been finished, thus, cannot create a non-repudiation message.

The advantage of SONATE is that it is able to mitigate the other attacks by selecting an appropriate BB to counter the attacks. Example of BB that can be used for counter the threats are data encryption (can be used to prevent the snooping attack), digital signature (can be used to mitigate the modification, man-in-the-middle, and reflection attacks), and flow control (can be used to counter DoS attack).

2.5.2 NENA

In NENA [5], the services provided by building blocks (the implementation of a protocol or a mechanism like CRC, Retransmission, etc.) are selected and composed by a composition algorithm to create a protocol graph called netlet. The composition process run during design time (by a developer or assisted by a software) assuming the requirements from an application, constraints from the administrator and networks. However, the selection of the most appropriate netlet is accomplished during runtime.

The security mechanisms of NENA are:

1. NENA uses secure deployment of protocols. Each protocol has a unique protocol ID [21].
2. NENA has a collaborative attack detection mechanism [22].
3. Similar to SONATE, NENA is able to select a protocol that offers:

a. Data encryption (i.e., data encryption netlet) [5].
b. Data authentication (i.e., digital signature, MAC) [5].

By matching the above security mechanisms with the attacks, we conclude that NENA is not able to prevent the traffic analysis and masquerading attacks since NENA does not provide anonymous connection and an attacker only need to know the port and the address of the target to pretend to be the target. Furthermore, NENA vulnerable to the repudiation attack since it does not have a trusted third party server.

However, NENA is able to mitigate the other attacks by using the secure deployments of protocols, collaborative attack detection, or by selecting an appropriate netlet to counter the attacks. Examples of netlets that can be used for counter the attacks are data encryption and digital signature netlets. Meanwhile, in order to mitigate denial-of-service attack, NENA utilizes the service of collaborative attack detection.

2.5.3 XIA

Whereas in the Internet, an IP address is used to address both the host and the content, XIA uses three principle types of identifiers to retrieve the content: Content ID, Host ID, and Service ID [2]. The content ID, which is the hash of the content, is used to retrieve the content without needing to know its location. The host ID, which is the hash of the host's public key, is used to contact the host that provides the content. The service ID, which is the hash of the service's public key, is used to get the service that provides the content. The security mechanisms in XIA are:

1. The architecture uses Content/Host/Service ID (CID/HID/SID) in order to retrieve the content. CID is the hash of the content, HID is the hash of the Public Key of the Host, and SID is the hash of the Public Key of the service [2].
2. XIA has the LAP (Lightweight Anonymity and Privacy) defence mechanism enables anonymous communication to prevent remote tracking [23].
3. The STRIDE defence mechanism allocates the available bandwidth in a tree-based topology [24]. This mechanism also available in XIA.
4. XIA has the AKI (Accountable Key Infrastructure) defence mechanism which provides a reliable data authentication process [25].

By reviewing the security mechanisms in XIA, we conclude that XIA is able to mitigate all of the reviewed attacks except the replaying attack, because XIA does not have any mechanism to bind one communication session.

For snooping attack, XIA mitigates them by using the public key and private key of a service (SID) to do encryption mechanism. For the other attacks, XIA mitigates them by using a hashed ID (CID/HID/SID) or by using the defence mechanisms provided by SCION architecture (LAP, AKI, or STRIDE defence mechanisms).

2.5.4 RINA

The basic design principle of RINA is that "Networking is only Inter-Process Communication (IPC)" [3] [26]. IPC is a function to allow two communication processes (one in the sender and another in the receiver side) to communicate with each other. Examples of the IPC functions are: locating processes, determining permissions, passing information, scheduling, and managing memory. Process names are used as identifiers. For example, a source application process requests a service using the process name of the destination application. They communicate with each other by utilizing the services of the Distributed IPC Facility (DIF).

The security mechanisms in RINA are:

1. All members in the same DIF must be authenticated first before they can join in [27].
2. Even if the attacker is already inside the DIF, he still needs to scan all of the possible Connection End Point id (CEP-id) of the target, and the probability is 2^{16} (given that the CEP-id is 16-bit) [27].
3. RINA has a SDU protection module that is able to provide security functions such as: encryption function, compression function, and error detection function [28].
4. The CEP-ids in RINA are used to distinguish between the new and the old data connection [27].

By analyzing the above security mechanisms, we acknowledge that RINA is able to mitigate all of the reviewed attacks except the DoS and repudiation attacks. The research to prevent DoS attack from the inside is on going since it is hard to detect. Meanwhile, RINA does not have a trusted third party to prevent the repudiation attack.

For the other attacks, RINA mitigates them by utilizing the function of the SDU Protection Module, the authentication process by IPC Management, and the unique CEP-Id that is assigned to each user. The CEP-Id can also be used to distinguish the old and new communications. The CEP-Id can mitigate the replaying attack by distinguishing the messages from a new session and the messages from an old session.

2.5.5 MobilityFirst

In this architecture, the end-user can request for a service using the Human-Readable Name (HRN). The naming architecture of the MobilityFirst has three identifiers: Network Address (NA), Globally Unique Identifier (GUID), and HRN [5]. It ensures mobility by separating network location information so called NA from its identity so called GUID. Similar with XIA, GUID is the hash of the content itself. MobilityFirst has two mapping services: Name Assignment Services (NAS) and Global Name Resolution Service (GNRS). The NAS binds an HRN with GUID and the GNRS maps GUID to NA. GNRS functions as a content location directory as it dynamically binds the name and the location. When the content is available in more than one locations, GNRS chooses the content for the requester from the nearest location.

The security mechanisms in MobilityFirst are:

1. To retrieve the content, MobilityFirst uses Globally Unique Identifier (GUID) which is assigned to each content as an address [6]. GUID is a result of hashing the content and can be used as a public key for encryption mechanism.
2. MobilityFirst enables frequent routing update using the function of GNRS [29].
3. MobilityFirst uses an integrated protocol that enables self-certifying public key names [30].

By matching the security mechanisms with the attacks, we conclude that MobilityFirst is robust against the snooping attack because it has a mechanism to encrypt the packets by using the name or GUID of the content as the identifier. For the modification attack, MobilityFirst mitigates it by assigning a unique GUID to each content. GUID is a hash of a content, therefore, the receiver can check the correctness of the content. For the man-in-the-middle, reflection, and masquerading attacks, MobilityFirst mitigates them by using a protocol that is able to authenticate a user and having a unique GUID for every content. For the DoS attack, MobilityFirst prevents it by utilizing the function of GNRS to perform routing update. For the repudiation attack, MobilityFirst prevents it by having a non-repudiation message, which is created by the PKI.

However, MobilityFirst cannot mitigate the traffic analysis and replaying attacks due to the following reasons:

1. MobilityFirst cannot mitigate the traffic analysis attack because it does not have a mechanism to enable anonymous communication.
2. The replaying attack is possible to do in MobilityFirst since it does not have a mechanism to bind the messages with the sessions.

2.5.6 NDN

NDN [8] defines two types of packets: one is for request (called interest packet) and another is for reply (called data packet). The interest packet has mainly two fields: content name and nonce (number once). The content name identifies the data to be retrieved and the nonce binds each communication session. The data packet carries both the name and the content of the data, together with the digital signature and signed information.

The security mechanisms in NDN [31] are:

1. An end-to-end encryption can be used in NDN. This is used to encrypt the NDN data.
2. Data packets are signed using the digital signature.
3. The clients in NDN send nonce (Number Once) within the interest packets.

By analyzing the above mechanisms, it can be concluded that NDN is vulnerable to the traffic analysis attack even though NDN is a content-centric network. Being content-centric is not enough to prevent that attack, it needs a mechanism such as a mechanism to conceal the packet, and this mechanism is not provided by NDN. The research for a method to mitigate the DoS attack is on going, therefore, NDN is vulnerable to the DoS attack.

NDN is able to mitigate the replaying attack since it has a nonce in its interest packet. This nonce will differentiate the old and the new interest packet. For the modification, repudiation, man-in-the-middle, and reflection attacks, NDN prevents them by performing a digital signature mechanism. For the masquerading attack, NDN prevents it by having a digital signature and unique name for every content.

2.5.7 NEBULA

NEBULA [7] facilitates data centers in a cloud environment to communicate in a reliable way. NEBULA consists of three components: NEBULA Core (NCore), NEBULA Data Plane (NDP), and NEBULA Virtual and Extensible Networking Techniques (NVENT). NCore interconnects the data centers using a reliable routing mechanism. NDP is a data plane that provides flexible access control and security mechanisms. NVENT is a control plane, which responsible for determining paths for the packets to arrive at the destination.

The security mechanisms in NEBULA [32] are:

1. Proof of Consent (PoC) mechanism to authorize a packet and a path.
2. Proof of Path (PoP) in order to make sure that the packet only flows on the authorized path.

3. NEBULA uses token to bind one authorized communication session.
4. There is a consent server in NEBULA that can act as a trusted third party. This server will prove that the communication between two users really took place because the users who want to send a packet will contact it to obtain the PoC.

NEBULA was not designed to mitigate the snooping and traffic analysis attacks. To mitigate these attacks, mechanisms such as end-to-end encryption or onion routing need to be applied on top of NEBULA. NEBULA also cannot prevent the masquerading attack because the attacker can pretend to be the authorized user by getting the users address.

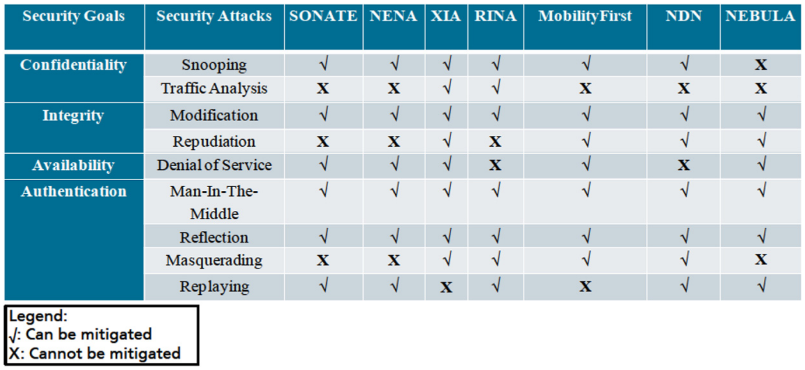
NEBULA is able to mitigate the other attacks by using the function of a PoC and a PoP that reside in the NDP. Moreover, NEBULA is able to prevent repudiation attack because it has a consent server as a trusted third party to create a nonrepudiation message.

3 Survey Results

The result of the comparison of all architectures is shown in Figure 2. It can be seen that SONATE and NENA cannot mitigate the same attacks which are the traffic analysis, masquerading, and repudiation attacks. They cannot mitigate the traffic analysis attack because they do not have any mechanism to enable anonymous communication. SONATE and NENA cannot prevent the masquerading attack since they do not provide anonymous communication and the attacker can get the IP address of the communicating hosts to pretend to be the authorized users. Meanwhile, they cannot prevent the repudiation attack because they do not have a trusted third party.

In summary, none of the studied architectures, whether it is clean slate or evolutionary, can mitigate all of the reviewed attacks. However, it can be seen in Figure 2 that XIA is the most secure, since it is able to handle eight out of nine reviewed attacks. Moreover, XIA is Content-Centric Network (CCN). CCN is claimed by the Future Content Networks (FCN) Group as the Future Internet (FI) [17]. At last, XIA has both a demonstration and a prototype available on github [33], thus, XIA can be considered as the most promising architecture to be deployed into the market.

Furthermore, in this article we choose to focus more on XIA because of the above reasons. Since XIA is still vulnerable to replaying attack, in this article we provide a replaying attack solution which is more secure and less complex than the existing solutions. In order to provide that solution, first we analyze the existing solutions for replaying attack in order to derive the requirements



Security Goals	Security Attacks	SONATE	NENA	XIA	RINA	MobilityFirst	NDN	NEBULA
Confidentiality	Snooping	√	√	√	√	√	√	X
	Traffic Analysis	X	X	√	√	X	X	X
Integrity	Modification	√	√	√	√	√	√	√
	Repudiation	X	X	√	X	√	√	√
Availability	Denial of Service	√	√	√	X	√	X	√
Authentication	Man-In-The-Middle	√	√	√	√	√	√	√
	Reflection	√	√	√	√	√	√	√
	Masquerading	X	X	√	√	√	√	X
	Replaying	√	√	X	√	X	√	√

Legend:
√: Can be mitigated
X: Cannot be mitigated

Figure 2 Comparison of all architectures in terms of handling attacks

for the proposed solution. The analysis of the existing solutions is provided in the following Section.

4 Analysis of the Existing Solutions for Replaying Attack

In order to propose a solution to be implemented in XIA, we reviewed the advantages and disadvantages nine existing solutions that have the mechanisms to prevent the replaying attack (e.g., session keys, random nonce, or timestamp). They are Diffie-Hellmann [34], Lamport's Password Authentication [35], S/Key One Time Password [36], Keung-Siu Protocol [37], Message Binding [38], Timestamp [39], Luo-Shieh-Shien Authentication Protocol [40], Yoon-Jeon Protocol [41], and Tseng-Jou Protocol [42]. The review of each solution is presented in the following Subsections.

4.1 Diffie-Hellman

Diffie-Hellman is a method to compute a unique session key. In order to compute a session key, the sender and the receiver choose two public parameters and generate a new private value in every session. Diffie-Hellman was developed by Whitfield Diffie and Martin E. Hellman and it was published in 1976 [34].

Advantage: This method is considered to be secure if the value of the public parameters, p and g, are chosen properly. Therefore, it is not likely for an attacker to calculate the secret key s= gab mod p. The secret key can be used to prevent replaying attack because only with the correct secret key Alice and Bob can encrypt and decrypt their messages [34].

Disadvantage: Original Diffie-Hellman scheme does not authenticate the communicating users, thus, it is vulnerable to the man-in-the-middle attack [42]. A person in the middle may establish two distinct Diffie-Hellman key exchanges, one with Alice and the other with Bob, effectively masquerading as Alice to Bob, and vice versa, allowing the attacker to decrypt (and read or store) then re-encrypts the messages passed between them.

4.2 Lamport's Password Authentication

Lamport's password authentication is a secure one-time password authentication method that was published by Leslie Lamport in 1981 [35,43]. This method implements a one-time password to protect against eavesdropping. The authentication process is between the user (A) and the server (S).

Advantage: This method is robust against the replaying attack since one session is bound by one password. Furthermore, a system that uses this method will never use a same password even though the system is crash. The system does not require back up to a point where a password already have been used, the system will continue from the point when the system crashed.

Disadvantage: This method is vulnerable to one type of man-in-the-middle attack, called the small n attack (e.g., the attacker impersonates the server).

4.3 S/KEY One Time Password

S/KEY One Time Password is a method that only allows one password ever crosses the network. The secret of a user will never be shared, thus, it prevents from an eavesdropping. This method was published by Neil Haller in 1994 [36].

Advantage: The user's secret pass-phrase never crosses the network at any time, thus, this method is able to prevent an eavesdropper. Assuming that an attacker manages to get hold of a password that was used for a successful authentication. This password is already useless for subsequent authentications, because each password can only be used once in one session, thus, prevents replaying attack.

Disadvantage: This method is vulnerable to a dictionary attack where the attacker is using a list of most possibly used passwords to guess the secret [44].

4.4 Keung-Siu Protocol

This protocol was developed by Stephen Keung and Kai-Yeung Siu in 1995 [37]. Aims of this protocol are to establish a session key while protecting the weak passwords (easy to be guessed by using a list of commonly used passwords) and to prevent off-line password guessing attack (the attacker guesses a password by analyzing the pattern of legitimate user's password). This protocol provides authentication process by using challenge and response messages that allow both users to validate each other.

Advantage: The protocols are immune to the replaying attack because of the following properties:

1. It uses random number that is different in every session. This random number is used to ensure that both hosts are communicating in one session.
2. This method also uses session key that is different in every session. The session key can be used to prevent the replaying attack.
3. This method uses encryption mechanisms so that the attacker is unable to read the message.

Disadvantage: The source and the destination must know the public key server, meanwhile, there is a situation where the public key is difficult to obtain (e.g., in a mobile environment).

4.5 Message Binding

By binding the messages to their correct context (e.g., binding the message to its protocol run), the replaying attack can be prevented. One way of binding the message can be done by including an information in the messages, therefore they are recognized to belong to a certain state of a certain protocol run. Example of information that can be included in the message is a protocol identifier [38].

Advantage: Message binding is able to withstand replaying attack because it has an information that is tagged to the message to bind the message and the protocol run.

Disadvantage: Message binding cannot bind a message and a session, it only binds a message with a protocol run. That means, in a certain point the replaying attack cannot be prevented (e.g., where the same protocol is used in a different session).

4.6 Timestamp

Timestamp is a marker that is used in a message to ensure the freshness of the message [39].

Advantage: The replaying attack is prevented by the use of timestamps. For example, a developer sets the value of $\delta_{\max}$, a constant to limit the difference in timestamp, to 200 milliseconds. If the receiver gets the message and the value of $|\mathrm{T}_{\mathrm{sender}}$ - $\mathrm{T}_{\mathrm{receiver}}|$ is higher than 200 milliseconds, then the receiver will detect the replaying attack and drops the message [39].

Disadvantage: One disadvantage of timestamp is in term of clock synchronization of the two hosts. Synchronization is required to maintain the accuracy and precision of the timestamp. The other disadvantage is, maintaining a list of used timestamps within the current window has the drawback of potentially large storage requirement, and corresponding verification overhead [41].

4.7 Luo-Shieh-Shien Authentication Protocol

This is a protocol to generate session keys with the help of a third party authentication server. This protocol was developed by Jia-Ning Luo, Shiuhpyng Shieh, and Ji-Chiang Shen and was published in 2006 [40].

Advantage: This protocol uses random numbers and session keys. The replaying attack can be mitigated by using the session keys as marker to distinguish the messages in different sessions. Furthermore, there is a mechanism to ensure that both hosts have created the same session key.

Disadvantage: There is a redundant message that increases the complexity of the protocol.

4.8 Yoon-Jeon Protocol

This protocol was developed by Eun-Jun Yoon and Il-Soo Jeon and was published in 2010 [41]. The protocol generates session keys based on Chebyshev polynomial.

Advantage: This protocol is robust against the replaying attack by utilizing the session key to ensure that the messages in one session are different than the messages in another session. Additionally, secure mutual authentication between entities is achieved by using a MAC by each entity. MAC is created by hashing the identity of both users and the Chebyshev Polynomial that is received by each user.

Disadvantage: There is an unused random number N. User A selects large prime number N that is not used in any operation and it is also not used to detect the freshness of the message. The inclusion of an unused random number can increase the complexity of the protocol.

4.9 Tseng-Jou Protocol

This protocol is an improvement of Yoon-Jeon Protocol. Similar to Yoon-Jeon Protocol, Tseng-Jou Protocol uses Chebyshev polynomial as a base to generate session keys. The main improvement is, it provides anonymous identity of the hosts by generating a parameter pseudo identity (PID) on each host. This protocol was developed by Huei-Ru Tseng and Emery Jou and published in 2011 [42].

Advantage: This protocol can mitigate replaying attack by using the session key to ensure the messages are bound to a specific session. It also provides anonymous identity of the host by having parameter PID on each host.

Disadvantage: There is an unused random number N_i. User U_i selects a large prime number Ni that is not used in any operation. To decrease the complexity of the protocol, the used of an unused random number can be avoided.

5 Derived Requirements for the Proposed Protocol

It can be seen in the last Section that all of the reviewed existing solutions have their own problems.

The properties that need to be satisfied by the proposed protocol are:

1. Use of a marker to distinguish the messages in different sessions.
2. Having a process to ensure that both users generate the same session key.
3. Using an encryption mechanism to protect the message, therefore, it is unreadable by the attacker.
4. Utilizing a mechanism to conceal the identity or the address of the sender.

Meanwhile, the properties that need to be avoided by the proposed protocol are:

1. Even though timestamp can be used as a marker, it has a disadvantage in term of clock synchronization between two communicating users. Therefore, timestamp can be avoided to reduce the risk of having synchronization issue.
2. Redundant computation that reduces the efficiency of the protocol.
3. The use of a useless random number that increases the complexity of the protocol.
4. To use several encryption mechanisms that reduces the efficiency of the protocol.

6 The Proposed Protocol

The proposed solution has to satisfy the desired properties and avoid the unwanted ones. Thus, the proposed solution is a complete protocol that provides a mechanism to mitigate replaying attack, provides an encryption mechanism, enables anonymous connection, and provides mutual authentication process. The protocol has the following properties:

1. It has markers in each session in the form of session keys (each host has one session key with length up to 280 bits).
2. The session keys are generated by XOR computation of four random numbers (70 hex per random number). The session keys are used by both users to differentiate the messages in different sessions.
3. Has a mechanism to ensure that the random numbers that are received at the receiver side are correct. This mechanism is needed for both hosts to create the same session key. This is achieved by checking the MAC in each host. The MAC value that is sent by User B has the random numbers that is generated by User A and has been received by User B. If User A finds the difference in the MAC value (e.g., someone is altering the random numbers, or there is an error in the network so that User B cannot obtain the random numbers from User A), then User A will terminate the session.
4. It also has a mechanism to ensure that both users generate a correct session key. This mechanism is needed to detect the replaying attack. This is also achieved by checking the MAC in each host and if each user has verified the MAC, then both users has generated a same session key.
5. Has two times data encryption, therefore, an attacker cannot read the message.
6. It does not have redundant computation and useless random number, thus reduces the complexity and increases the efficiency of the protocol.
7. It generates a parameter that is called Pseudo Identity (PID) to hide the host's identity.

The sequence diagrams of how the protocol works can be seen in Figures 3 and 4.

The proposed protocol works in the following ways:

1. The first assumption before running the protocol is as follows: User A and user B have exchanged their public key to be used for the encryption-decryption mechanisms.
2. User A generates two random numbers n_{A1} and n_{A2}, hashes these two random numbers, then computes pseudo identity (PID_A) to hide his identity. He encrypts his identity (HID_A) and his random numbers with the user B's public key, and then sends it along with the PID_A to user B.
3. User B generates two random numbers n_{B1} and n_{B2}, hashes these two random numbers, then computes pseudo identity (PID_B) to conceal his

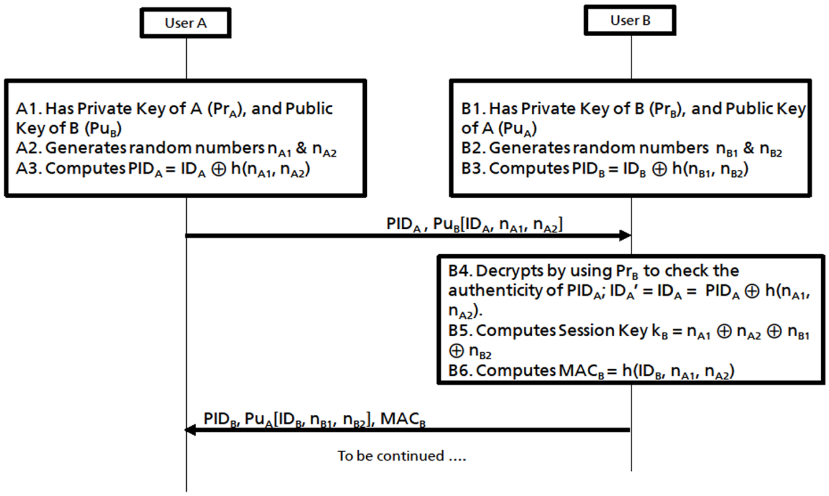

User A
User B
A1. Has Private Key of A (PrA), and Public Key of B (PuB)
A2. Generates random numbers nA1 & nA2
A3. Computes PIDA = IDA ⊕ h(nA1, nA2)
B1. Has Private Key of B (PrB), and Public Key of A (PuA)
B2. Generates random numbers nB1 & nB2
B3. Computes PIDB = IDB ⊕ h(nB1, nB2)
PIDA , PuB[IDA, nA1, nA2]
B4. Decrypts by using PrB to check the authenticity of PIDA; IDA' = IDA = PIDA ⊕ h(nA1, nA2).
B5. Computes Session Key kB = nA1 ⊕ nA2 ⊕ nB1 ⊕ nB2
B6. Computes MACB = h(IDB, nA1, nA2)
PIDB, PuA[IDB, nB1, nB2], MACB
To be continued ....

Figure 3 Sequence diagram of the proposed protocol-top part

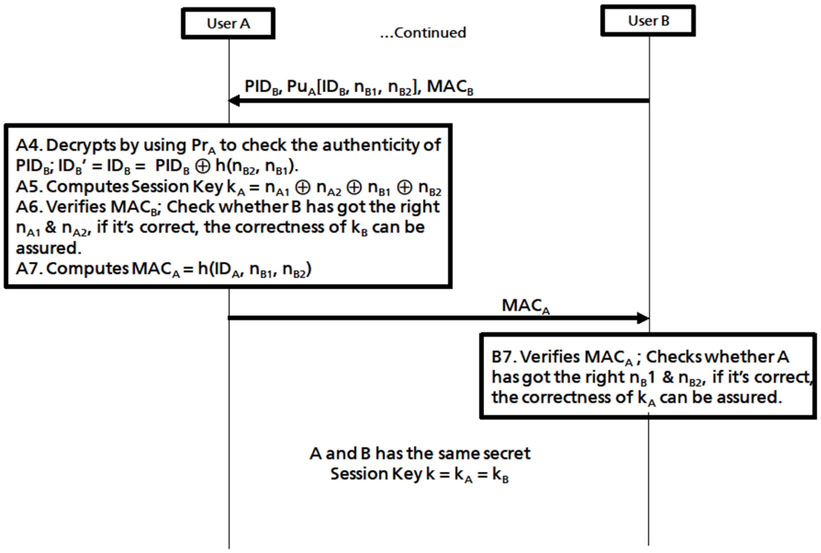

User A
...Continued
User B
PIDB, PuA[IDB, nB1, nB2], MACB
A4. Decrypts by using PrA to check the authenticity of PIDB; IDB' = IDB = PIDB ⊕ h(nB2, nB1).
A5. Computes Session Key kA = nA1 ⊕ nA2 ⊕ nB1 ⊕ nB2
A6. Verifies MACB; Check whether B has got the right nA1 & nA2, if it's correct, the correctness of kB can be assured.
A7. Computes MACA = h(IDA, nB1, nB2)
MACA
B7. Verifies MACA ; Checks whether A has got the right nB1 & nB2, if it's correct, the correctness of kA can be assured.
A and B has the same secret Session Key k = kA = kB

Figure 4 Sequence diagram of the proposed protocol-bottom part

identity. He encrypts his identity (HID_B) and his random numbers with user A's public key. After receiving the message from A, he decrypts the message using his private key, then he authenticates the identity of A, if it is not correct then he will terminate the connection. But if it is correct, he will compute session key k_B and MAC_B. Then he sends his PID_B along with the encrypted message (HID_B, n_{B1}, n_{B2}) and the MAC_B.

4. After user A receiving the message from user B, he decrypts the message by using his private key. Then he authenticates the identity of B, if it is not correct then he will terminate the connection. But if it is correct, he will compute session key k_A and MAC_A. After that, user A will authenticate the MAC_B to make sure that user B got the correct random numbers from him, this means B also has generated a correct session key. After completing all of the checking processes, user A sends his MAC_A to B.
5. User B will authenticate MAC_A to make sure that user A has got the correct random numbers from him, and also to make sure that user A has generated a correct session key.
6. After completing all of the checking processes, user A and user B have the same secret session key ($k_A = k_B$) to be used during their communication.
7. The random numbers and the session keys that are generated by user A and user B are different in every session.

7 Implementation

The protocol is implemented in XIA Prototype in order to prove that the protocol is able to make XIA robust against replaying attack and is able to generate the desired result (secured session keys). In order to simulate how the proposed protocol prevents the replaying attack, a topology is created by using VirtualBox version 4.2.12. The topology can be seen in Figure 5.

It can be seen in Figure 5 that the Attacker is connected to the Router via Ethernet 1 and to the NAT via Ethernet 2. In XIA, The Attacker cannot connect to Host0 and Host1 directly. It is necessary for the Attacker to connect with the Router. Since the HID of the Attacker is given by the Router. The Router is the one that connects the Attacker with Host0 and Host1. Also can be seen in Figure 5 that each host connected via two interfaces, one of them is connected to the Router while the other is connected to NAT. Every hosts need to be connected to NAT in order to give them an internet connection that is used to obtain the XIA Prototype from the github [33].

There are three cases to be used to test the proposed protocol: First, a case when Host0 and Host1 are sending and receiving data without being interrupted by the Attacker. In this case, Host0 and Host1 do not run the proposed protocol applications. Second, when the Attacker is successfully performing the replaying attack. In this case, Host1 authenticates the Attacker as Host0. Third, a case when Host0 and Host1 run the applications for the

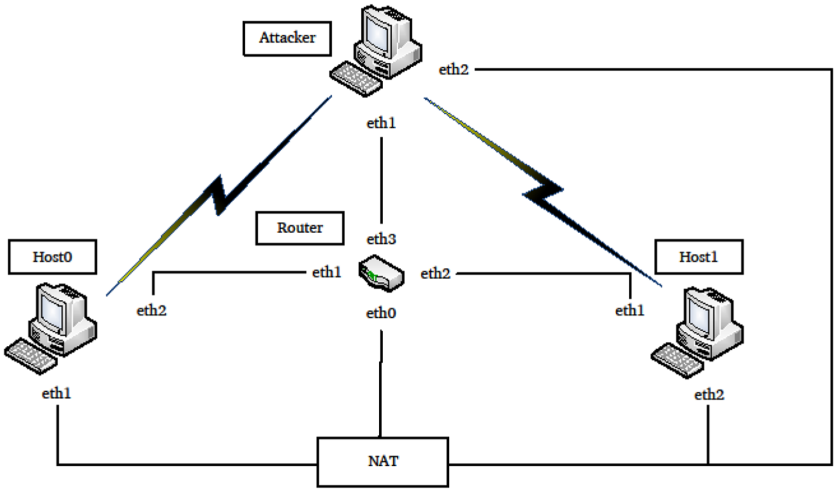


Figure 5 Topology for implementation

proposed protocol before they start exchanging data. This case is used as a proof that the protocol is able to mitigate the replaying attack.

7.1 Common Data Exchange

In this case, Host0 and Host1 are exchanging data without using the proposed protocol. Instead of using the session key as a header of a data, they use their HID. To use the HID as a header, they are exchanging their HID before they start exchanging data. The sequence diagram of this step can be seen in Figure 6. It can be seen that, Host0 and Host1 communicate in one session only. It is assumed that the Attacker is idle.

The result from this case is, each host uses its HID as a header of the message that it wants to send. The HID is used by the receiver to authenticate the sender.

7.2 Replaying Attack Scenario

This case is to simulate the replaying attack. It is assumed that Host0 and Host1 have already exchanged HID. These HIDs are always same in each session. The scenario is, the Attacker captured and saved the data from Host0. To simulate the replaying attack, it is assumed that the previous session has ended and the Attacker replays the data from Host0 to Host1 in the next session. The sequence diagram of this step can be seen in Figure 7.

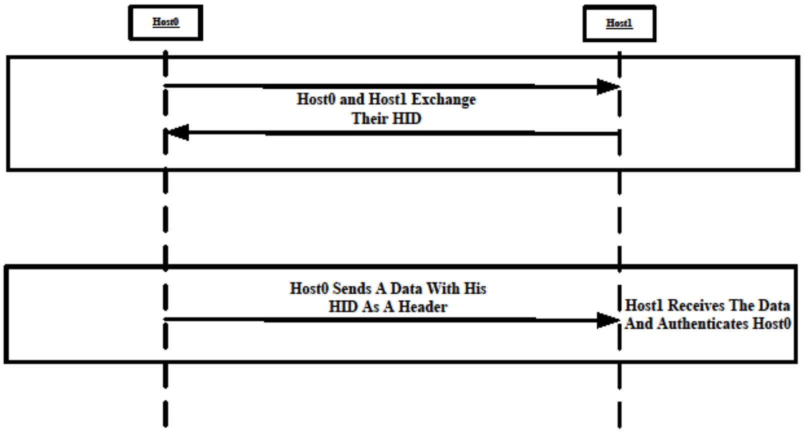

Host0
Host1
Host0 and Host1 Exchange Their HID
Host0 Sends A Data With His HID As A Header
Host1 Receives The Data And Authenticates Host0

Figure 6 Scenario diagram for common data exchange

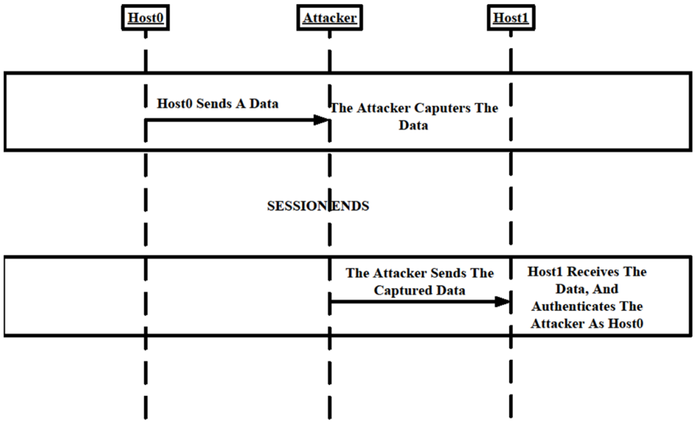

Host0
Attacker
Host1
Host0 Sends A Data
The Attacker Caputers The Data
SESSION ENDS
The Attacker Sends The Captured Data
Host1 Receives The Data, And Authenticates The Attacker As Host0

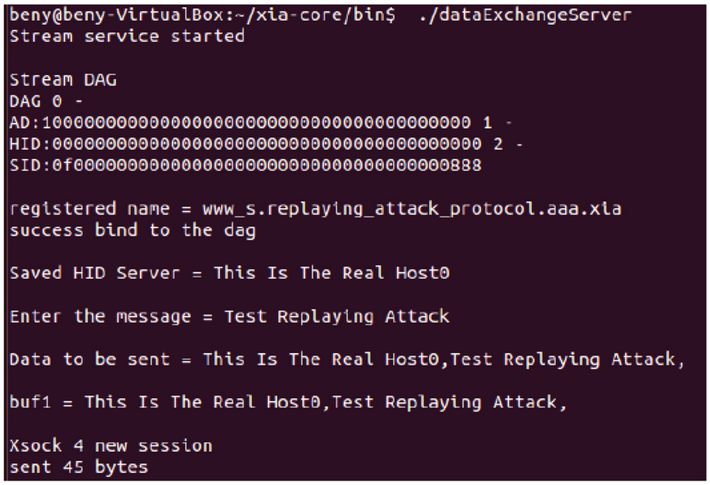
Figure 7 Sequence diagram for replaying attack scenario

The result from this case can be seen in Figure 8 to 11.

Figure 8 represents the following processes:

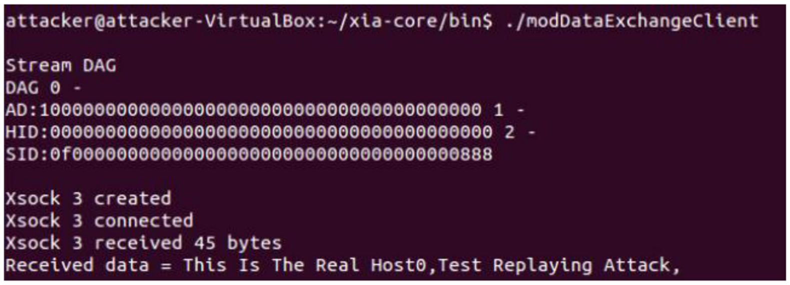
- Host0 used his HID, “This Is The Real Host0”, as the header of the data. The data consists of the HID of Host0 and the message that he wants to send, “Test Replaying Attack”.
- Host0 sent the data, and this data is intended to be sent to Host1.

It can be seen in Figure 9 that the Attacker got the data from Host0 and then save it.



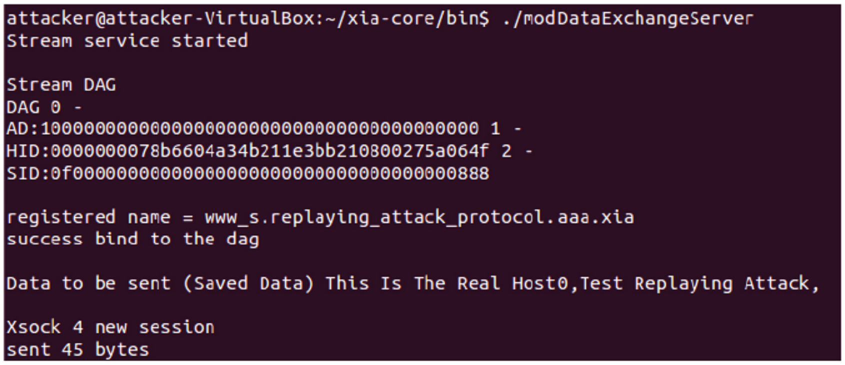
```
beny@beny-VirtualBox:~/xia-core/bin$  ./dataExchangeServer
Stream service started

Stream DAG
DAG 0 -
AD:1000000000000000000000000000000000000000 1 -
HID:0000000000000000000000000000000000000000 2 -
SID:0f00000000000000000000000000000000000888

registered name = www_s.replaying_attack_protocol.aaa.xia
success bind to the dag

Saved HID Server = This Is The Real Host0

Enter the message = Test Replaying Attack

Data to be sent = This Is The Real Host0,Test Replaying Attack,

buf1 = This Is The Real Host0,Test Replaying Attack,

Xsock 4 new session
sent 45 bytes
```

Figure 8 Send the data using HID of host0 as a header



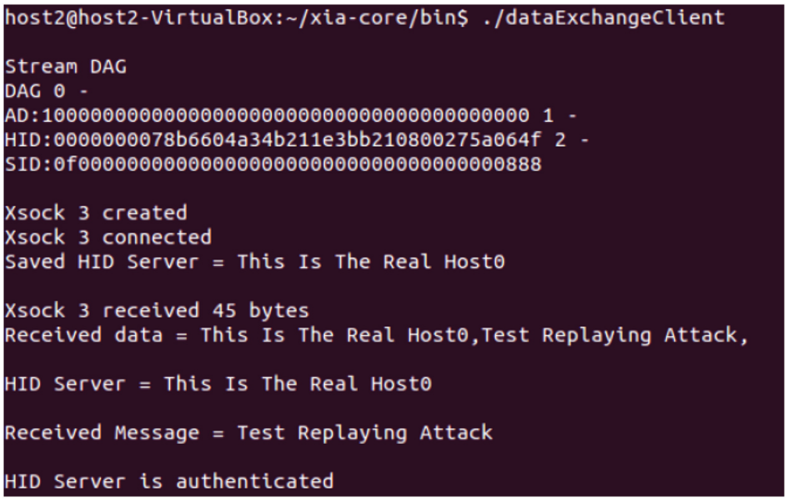
```
attacker@attacker-VirtualBox:~/xia-core/bin$ ./modDataExchangeClient

Stream DAG
DAG 0 -
AD:1000000000000000000000000000000000000000 1 -
HID:0000000000000000000000000000000000000000 2 -
SID:0f00000000000000000000000000000000000888

Xsock 3 created
Xsock 3 connected
Xsock 3 received 45 bytes
Received data = This Is The Real Host0,Test Replaying Attack,
```

Figure 9 The attacker captures and saves the data



```
attacker@attacker-VirtualBox:~/xia-core/bin$ ./modDataExchangeServer
Stream service started

Stream DAG
DAG 0 -
AD:1000000000000000000000000000000000000000 1 -
HID:0000000078b6604a34b211e3bb210800275a064f 2 -
SID:0f00000000000000000000000000000000000888

registered name = www_s.replaying_attack_protocol.aaa.xia
success bind to the dag

Data to be sent (Saved Data) This Is The Real Host0,Test Replaying Attack,

Xsock 4 new session
sent 45 bytes
```

Figure 10 The attacker sends the data in the next session



```
host2@host2-VirtualBox:~/xia-core/bin$ ./dataExchangeClient

Stream DAG
DAG 0 -
AD:1000000000000000000000000000000000000000 1 -
HID:0000000078b6604a34b211e3bb210800275a064f 2 -
SID:0f00000000000000000000000000000000000888

Xsock 3 created
Xsock 3 connected
Saved HID Server = This Is The Real Host0

Xsock 3 received 45 bytes
Received data = This Is The Real Host0,Test Replaying Attack,

HID Server = This Is The Real Host0

Received Message = Test Replaying Attack

HID Server is authenticated
```

Figure 11 Host1Receives the data from the attacker

It can be seen in Figure 10 and Figure 11 that the Attacker managed to perform the replaying attack. The Attacker replayed the data from Host0 to Host1 (as shown in Figure 10) and Host1 authenticated the Attacker as Host0 (as shown in Figure 11).

7.3 Mitigating Replaying Attack by Applying the Proposed Protocol

This case is to simulate how the proposed protocol mitigates the replaying attack. Host0 and Host1 create a session key by running the protocol. This protocol will be run in each session to create a session key that is unique in every session. The sequence diagram of this step can be seen in Figure 12.

The result from this case is, Host1 detects a replaying attack because the session key that is used by the Attacker is different than the session key that were generated by Host0 and Host1. This is because the session keys are different in every session. Once the attack is detected, Host1 terminates the session, and generates a new session key with Host0. The result of replaying attack detection by using the protocol can be seen in Figure 13.

Figure 13 represents the following processes:

- Host1 received the data that was sent by the Attacker. It is assumed that the data was captured by the Attacker in the 1st session, then he sent it to Host1 in the 2nd session.

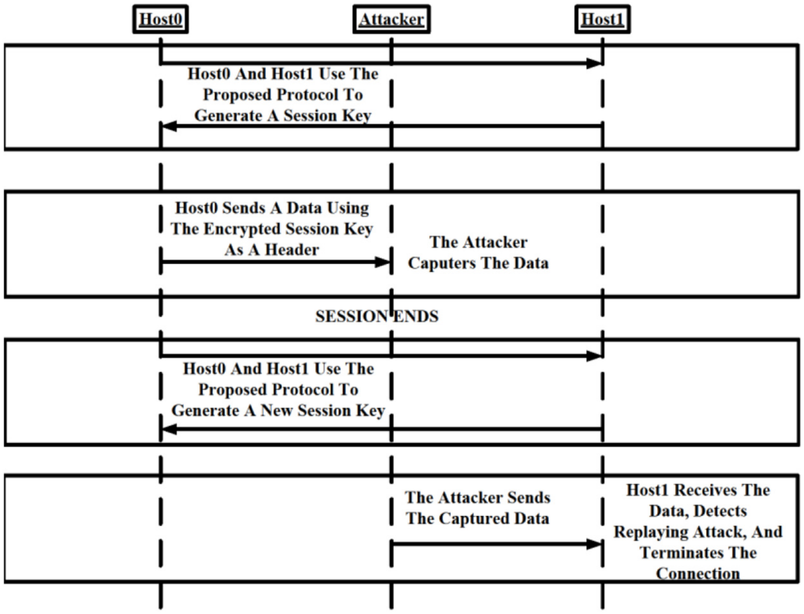
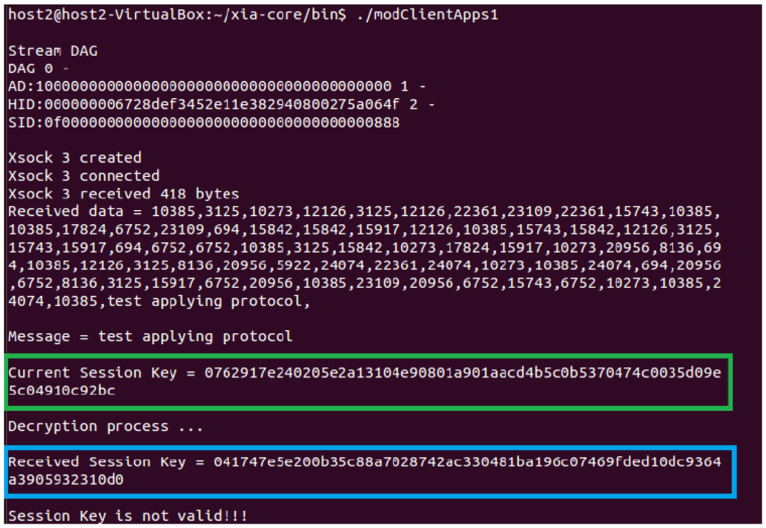

Host0
Attacker
Host1
Host0 And Host1 Use The Proposed Protocol To Generate A Session Key
Host0 Sends A Data Using The Encrypted Session Key As A Header
The Attacker Caputers The Data
SESSION ENDS
Host0 And Host1 Use The Proposed Protocol To Generate A New Session Key
The Attacker Sends The Captured Data
Host1 Receives The Data, Detects Replaying Attack, And Terminates The Connection

Figure 12 Sequence diagram for mitigating replaying attack scenario



```
host2@host2-VirtualBox:~/xia-core/bin$ ./modClientApps1

Stream DAG
DAG 0 -
AD:1000000000000000000000000000000000000000 1 -
HID:00000006728def3452e11e382940800275a064f 2 -
SID:0f00000000000000000000000000000000000888

Xsock 3 created
Xsock 3 connected
Xsock 3 received 418 bytes
Received data = 10385,3125,10273,12126,3125,12126,22361,23109,22361,15743,10385,
10385,17824,6752,23109,694,15842,15842,15917,12126,10385,15743,15842,12126,3125,
15743,15917,694,6752,6752,10385,3125,15842,10273,17824,15917,10273,20956,8136,69
4,10385,12126,3125,8136,20956,5922,24074,22361,24074,10273,10385,24074,694,20956
,6752,8136,3125,15917,6752,20956,10385,23109,20956,6752,15743,6752,10273,10385,2
4074,10385,test applying protocol,

Message = test applying protocol

Current Session Key = 0762917e240205e2a13104e90801a901aacd4b5c0b5370474c0035d09e
5c04910c92bc

Decryption process ...

Received Session Key = 041747e5e200b35c88a7028742ac330481ba196c07469fded10dc9364
a3905932310d0

Session Key is not valid!!!
```

Figure 13 Replaying attack detection

- In order to detect the replaying attack, Host1 separated the encrypted session key and the message.
- Host1 authenticated the session key by comparing the value of "Current Session Key" and the value of "Received Session Key". The value of "Current Session Key" is obtained when Host1 run the protocol application in 2nd session, meanwhile, the value of "Received Session Key" is obtained by decrypting the session key that was sent by Host0.
- When the comparison failed, Host1 sent out an error message. The error message was to inform that the session key is not valid (Host1 detected the replaying attack). Furthermore, Host1 terminated the session because the data that was received by Host1 was encrypted by using different session key than the current session key.

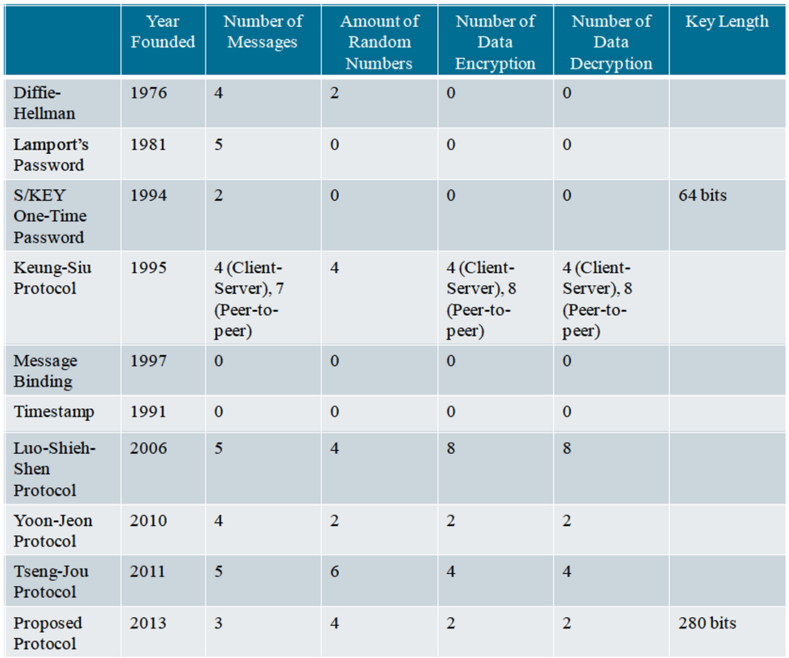
From the above figures can be concluded that the new protocol is able to detect replaying attack. In the next section the evaluation of the new protocol is presented.

8 Evaluation

This Section presents the evaluation of the proposed protocol by comparing it with the existing solutions as shown in Figure 14. The evaluation can also be seen in [46].

It can be seen in Figure 14 that:

1. The proposed protocol generates a session key with a length of 280 bits. The session key is never exchanged between hosts, therefore, the Attacker needs to guess the session key if he wants to carry out an attack (e.g., replaying or modification attacks). The possibility for the attacker to guess the session key is 2^{280}.
2. It has three messages to be exchanged while running the protocol. This amount of message is smaller than the other solutions that have four (Diffie-Hellman, Keung-Siu Protocol, and Yoon-Jeon Protocol) or five messages (Lamport's Password Authentication, Luo-Shieh-Shen Authentication Protocol, and Tseng-Jou Protocol).
3. It has four random numbers. The random numbers are used to generate the session key, a possibility for an attacker to guess four random number is $10^{70} \times 4$ (given 10 possibilities in one digit) and it is larger than a possibility to guess two random numbers (used in Diffie-Hellman and Yoon-Jeon Protocol), which is $10^{70} \times 2$. Furthermore, it is less complex than the solution that has six random numbers (Tseng-Jou Protocol).



	Year Founded	Number of Messages	Amount of Random Numbers	Number of Data Encryption	Number of Data Decryption	Key Length
Diffie-Hellman	1976	4	2	0	0	
Lamport's Password	1981	5	0	0	0	
S/KEY One-Time Password	1994	2	0	0	0	64 bits
Keung-Siu Protocol	1995	4 (Client-Server), 7 (Peer-to-peer)	4	4 (Client-Server), 8 (Peer-to-peer)	4 (Client-Server), 8 (Peer-to-peer)	
Message Binding	1997	0	0	0	0	
Timestamp	1991	0	0	0	0	
Luo-Shieh-Shen Protocol	2006	5	4	8	8	
Yoon-Jeon Protocol	2010	4	2	2	2	
Tseng-Jou Protocol	2011	5	6	4	4	
Proposed Protocol	2013	3	4	2	2	280 bits

Figure 14 Comparison of the proposed protocol with existing solutions

In addition, none of these random numbers are useless like in Yoon-Jeon and Tseng-Jou Protocols.

4. It has two times data encryption and decryption, therefore, it reduces the complexity than the other solutions that have four (Tseng-Jou Protocol and Keung-Seu Protocol between client and server) or even eight times data encryption-decryption (Keung-Seu Protocol between two clients and Luo-Shieh-Shen Authentication Protocol).

9 Conclusion and Future Work

The eXpressive Internet Architecture (XIA) is an open-source Content-Centric Network (CCN) which has potential to be standardized in future as CCN is claimed by the Future Content Networks (FCN) Group to be the Future Internet (FI). However, XIA lacks mechanism to mitigate replaying attack. Therefore, a solution for replaying attack has been proposed and implemented in this article. Nine existing solutions such as Diffie-Hellmann, Lamport's Password Authentication, S/Key One Time Password, Keung-Siu Protocol, Message

Binding, Timestamp, Luo-Shieh-Shien Authentication Protocol, Yoon-Jeon Protocol, and Tseng-Jou Protocol have been analyzed to derive the requirements for the proposed protocol. Based on the derived requirements, the solution has been developed.

The protocol has been implemented in XIA prototype and has been proven to be able to mitigate the replaying attack. The proposed protocol has the following properties: First, There is a unique session key for each host in every session. Second, there is a checking process to ensure that the session key that is generated at each host is the same. Third, it has mechanisms to encrypt the messages and to conceal the identity of the hosts.

The proposed protocol has been evaluated to have more advantages over the reviewed existing solutions. It is more secure by having session key with length of 280 bits. Moreover, it is less complex as none of the random numbers used in the protocol are worthless. By applying the proposed protocol, XIA is now able to mitigate all of the reviewed attacks.

According to the current standard [45], a session key with a length of up to 280 bits is secure. In the future, when 280 bits is not enough, the size of the session key can be extended.

10 Standardization Candidate

The standardization of the future network architectures is a responsibility of the ITU-T Study Group 13 (SG 13) [47]. The SG 13 consists of several groups, and Focus Group Future Networks (FG FN) is one of them, and its aim is to collect FI architectures and technologies to be standardized [48]. The new protocol that is presented in this paper can be standardized to be used in XIA, as it causes XIA to be robust against all of the reviewed attacks. Furthermore, the protocol is more secure and more efficient than others replaying attack solution, thus, the protocol can also be standardized for the current Internet network.

References

[1] Anja Feldmann, “Internet Clean-Slate Design: What and Why?,” in *SIGCOMM Computer Communication Review*. 2007, pp. 59–64. Volume 37, Number 3, ACM.

[2] Ashok Anand, Fahad Dogar, Dongsu Han, Boyan Li, Hyeontaek Lim, Michel Machado, Wenfei Wu, Aditya Akella, David Andersen, John

Byers, Srinivasan Seshan, and Peter Steenkiste, "XIA: An Architecture for an Evolvable and Trustworthy Internet," in *Proceedings of the tenth ACM Workshop on Hot Topics in Networks (HotNets-X)*. 2011, pp. 1–32. Article No. 2, ACM.

[3] John Day, Ibrahim Matta, and Karim Mattar, "Networking is IPC: A Guiding Principle to a Better Internet," in *Proceedings of the 2008 ACM CoNEXT Conference. 2008*, pp. 1–6. Article Number 67, ACM.

[4] Bernd Reuther and Paul Müller, "Future Internet Architecture - A Service Oriented Approach," in *In IT - Information Technology*, Volume 50, Number 6, 2008, pp. 1–7.

[5] Denis Martin, Lars Völker, and Martina Zitterbart, "A flexible framework for Future Internet design, assessment, and operation," *Journal Computer Networks: The International Journal of Computer and Telecommunications Networking*, pp. 910–918. Volume 55 Issue 4, March 2011.

[6] Ivan Seskar, Kiran Nagaraja, Sam Nelson, and Dipankar Raychaudhuri, "MobilityFirst Future Internet Architecture Project," in *Proceeding of: AINTEC '11, Asian Internet Engineering Conference*, 2011, pp. 1–3.

[7] Robert Broberg, Matthew Caesar, Douglas Comer, Chase Cotton, Michael J. Freedman, Andreas Haeberlen, Zachary G. Ives, Arvind Krishnamurthy, William Lehr, Boon Thau Loo, David Mazires, Antonio Nicolosi, Jonathan M. Smith, Ion Stoica, Robbert van Renesse, Michael Walfish, Hakim Weatherspoon,and Christopher S. Yoo, "The NEB-ULA Future Internet Architecture," *Lecture Notes in Computer Science*, pp. 1–24. Volume 7858, 2013.

[8] Lixia Zhang, Deborah Estrin, Jeffrey Burke, Van Jacobson, James D. Thornton, Diana K. Smetters, Beichuan Zhang, Gene Tsudik, KC Claffy, Dmitri Krioukov, Dan Massey, Christos Papadopoulos, Tarek Abdelzaher, Lan Wang, Patrick Crowley, and Edmund Yeh, "Named Data Networking (NDN) Project," pp. 1–26, 2010.

[9] Rowan Klöti, "OpenFlow: A Security Analysis," M.S. thesis, Swiss Federal Institute of Technology Zurich, 2013.

[10] Jie Wang, *Computer Network Security: Theory and Practice*, Higher Education Press, 2009.

[11] Ltd Hangzhou H3C Technologies Co., "Attack Prevention Technology White Paper," 2008.

[12] Emmett Dulaney, *CompTIA Security+ Study Guide*, Wiley, Indianapolis, 4th edition, 2009

[13] Jelena Mirkovic, Sven Dietrich, David Dittrich, and Peter Reiher, *Internet Denial of Service: Attack and Defence Mechanisms*, Prentice Hall, 2005

[14] Mark Ciampa, *Security Plus Guide to Network Security Fundamentals*, Cengage Learning, 3rd edition, 2009.

[15] Ling Dong and Kefei Chen, *Cryptographic Protocol: Security Analysis Based on Trusted Freshness*, Springer, 2012.

[16] Hamid Jahankhani, David Lilburn Watson, Gianluigi Me, and Frank Leonhardt, *Handbook of Electronic Security and Digital Forensics*, 2010

[17] Hannes Gredler and Walter Goralski, *The Complete IS-IS Routing Protocol*, Springer, 2004.

[18] Future Internet Assembly (FIA) Future Content Networks (FCN) Group, "Technical Report. Why do we need a Content Centric Future Internet?," pp. 1–23, 2009.

[19] M. Rahamatullah Khondoker, Abbas Siddiqui, Bernd Reuther, and Paul Müller, "Service Orientation Paradigm in Future Network Architectures," in *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2012)*, 2012, pp. 346–351.

[20] Paul Müller, Bernd Reuther, and Markus Hillenbrand, "Future Internet: A Service-Oriented Approach - SONATE," in *Würzburg Workshop on Visions of Future Generation Networks (EuroView2007)*, 2007, pp. 1–35.

[21] Oliver Hanka and Hans Wippel, "Secure Deployment of Application-Tailored Protocols in Future Networks," in *Proceedings of the Second International Conference on the Network of the Future (NoF 2011)*, 2011, pp. 10–14.

[22] Thomas Gamer and Hans Wippel, "A Collaborative Attack Detection and its Challenges in the Future Internet," in *Proceedings of the Joint ITG, ITC, and Euro-NF Workshop "Visions of Future Generation Networks" (EuroView)*, 2010, pp. 1–2.

[23] Hsu-Chun Hsiao, Tiffany Hyun-Jin Kim, Adrian Perrig, Akira Yamada, Samuel C. Nelson, Marco Gruteser, and Wei Meng, "LAP: Lightweight Anonymity and Privacy," in *Proceedings of the IEEE Symposium on Security and Privacy*. 2012, pp. 506–520, IEEE Computer Society.

[24] Hsu-Chun Hsiao, Tiffany Hyun-Jin Kim, Sangjae Yoo, Xin Zhang, Soo Bum Lee, Virgil Gligor, and Adrian Perrig, "STRIDE: Sanctuary Trail Refuge from Internet DDoS Entrapment," in *Proceedings of the ACM Symposium on Information, Computer and Communications Security (ASIACCS)*. 2013, pp. 415–426, ACM.

[25] Tiffany Hyun-Jin Kim, Lin-Shung Huang, Adrian Perrig, Collin Jackson, and Virgil Gligor, "Accountable Key Infrastructure (AKI): A Proposal for a Public-Key Validation Infrastructure," in *Proceedings of the 22nd*

international conference on World Wide Web. 2013, pp. 679–690, International World Wide Web Conferences Steering Committee.

[26] Eleni Trouva, Eduard Grasa, John Day, Ibrahim Matta, Lou Chitkushev, Patrick Phelan, and Miguel Ponce de Leon Steve Bunch, "Is the Internet an unfinished demo? Meet RINA!," in *TERENA Networking Conference*, 2011, pp. 1–12.

[27] Gowtham Boddapati, John Day, Ibrahim Matta, and Lou Chitkushev, "Assessing the Security of a Clean-Slate Internet Architecture," in *Proceedings of the Seventh Workshop on Secure Network Protocols (NPSec)*. 2012, pp. 1–6, 20th IEEE International Conference Network Protocols (ICNP).

[28] Jeremiah Small, "Patterns in Network Security: an Analysis of Recursive Inter-Network Architecture Security Module Efficiency," M.S. thesis, Boston University, 2012.

[29] Feixiong Zhang, Kiran Nagaraja, Yanyong Zhang, and Dipankar Raychaudhuri, "Content Delivery in the MobilityFirst Future Internet Architecture," in *Sarnoff Symposium (SARNOFF)*, 35th IEEE, 2012, pp. 1–5.

[30] MobilityFirst Project Team, "MobilityFirst: A Robust and Trustworthy Mobility-Centric Architecture for the Future Internet," Tech. Rep., 2010.

[31] Paolo Gasti, Gene Tsudik, Ersin Uzun, and Lixia Zhang, "DoS and DDoS in Named-Data Networking," 2012, pp. 1–10. Volume abs/1208.0952.

[32] Jad Naous, Michael Walfish, Antonio Nicolosi, David Mazieres, Michael Miller, and Arun Seehra, "Verifying and Enforcing Network Paths With ICING," in *Proceedings of the Seventh Conference on emerging Networking EXperiments and Technologies*. 2011, pp. 1–12. Article No. 30, ACM.

[33] XIA Project Team, "XIA Prototype," https://github.com/XIA-Project/xia-core/wiki, 2013, [Online; Accessed on 01-August-2013].

[34] Whitfield Diffie and Martin E. Hellman, "New Directions in Cryptography," *Journal IEEE Transactions on Information Theory*, pp. 644–654. Volume 22 Issue 6, 1976.

[35] Leslie Lamport, "Password Authentication With Insecure Communication," *Magazine Communications of the ACM*, pp. 770–772. Volume 24 Issue 11, 1981.

[36] Neil Haller, "The S/KEY One-Time Password System," in *Proceedings of the Internet Society Symposium on Network and Distributed Systems*, 1994, pp. 151–157.

[37] Stephen Keung and Kai-Yeung Siu, "Efficient Protocols Secure Against Guessing and Replay Attacks," in *Proceedings, Fourth International Conference on Computer Communications and Networks*, 1995, pp. 105–112.

[38] Tuomas Aura, "Strategies against Replay Attacks," in *Proceedings of the 10th IEEE workshop on Computer Security Foundations CSFW'97*, 1997, pp. 59–68.

[39] Cdric Adjih, Daniele Raffo, and Paul Mhlethaler, "Attacks Against OLSR: Distributed Key Management for Security," *2nd OLSR Interop/Wksp.*, pp. 1–7, 2005.

[40] Jia-Ning Luo, Shiuhpyng Shieh, and Ji-Chiang Shen, "Secure Authentication Protocols Resistant to Guessing Attacks," *Journal of Information Science and Engineering*, pp. 1125–1143. Volume 22 No. 5, 2006.

[41] Eun-Jun Yoon and Il-Soo Jeon, "An efficient and secure Diffie Hellman key agreement protocol based on Chebyshev chaotic map," *Communications in Nonlinear Science and Numerical Simulation*, pp. 23832389. Volume 16, Issue 6, 2010.

[42] Huei-Ru Tseng and Emery Jou, "An Efficient Anonymous Key Agreement Protocol Based on Chaotic Maps," in *IEEE 13th International Conference on High Performance Computing and Communications (HPCC)*, 2011, pp. 752–757.

[43] Tsuji Takasuke, "A One-Time Password Authentication Method," M.S. thesis, Graduate School of Engineering, Kochi University of Technology, 2002.

[44] Sung-Ming Yen and Kuo-Hong Liao, "Shared authentication token secure against replay and weak key attacks," in *Information Processing Letters*. 1997, pp. 77–80. Volume 62 Issue 2, Elsevier North-Holland, Inc.

[45] Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid, "Recommendation for Key Management Part 1: General (Revision 3)," pp. 1–147, 2012.

[46] Beny Nugraha, Rahamatullah Khondoker, Ronald Marx, and Kpatcha Bayarou, "A Mutual Key Agreement Protocol To Mitigate Replaying Attack In eXpressive Internet Architecture (XIA)", in *ITU Caleidoscope Academic Conference*, pp. 233–240. 2014.

[47] "ITU-T SG13: Future networks including cloud computing, mobile and next-generation networks," http://www.itu.int/en/ITU-T/studygroups/2013-2016/13/Pages/default.aspx, Online; accessed 06-Dec-2013.

[48] "ITU-T FG FN: Focus Group on Future Networks (FG FN)," http://www.itu.int/en/ITUT/focusgroups/fn/Pages/Default.aspx, Online; accessed 06-Dec-2013.

Biographies

Beny Nugraha received his dual masters degree – International Master Degree Program from Bandung Institute of Technology (Indonesia) and Hochschule Darsmtadt (Germany) in 2013. In order to finish his Master Degree in Germany, he received a scholarship from the Indonesian Directorate General of Higher Education. Currently, he is a lecturer at the department of Electrical Engineering in Mercu Buana University located in Jakarta, Indonesia. His research is mainly about network security, currently he is focusing on the security of Future Internet Architectures and cloud computing.

Since 2010, **Rahamatullah Khondoker** has been working towards his PhD degree on "Description and Selection of Communication Services for Service Oriented Network Architectures (SONATE)" at the University of

Kaiserslautern in Germany. He was awarded from Ericsson, Germany in the year 2008 and from the FIA Research Roadmap group in October 2011. Currently, he is affiliated with the Fraunhofer SIT located in Darmstadt, Germany. He worked with the DFG project (PoSSuM), BMBF projects (G-Lab, G-Lab DEEP, Future-IN), and EU projects (PROMISE, EuroNF). Currently, he is focusing on the security of Future Internet Architectures, Software-Defined Networking (SDN), and Network Function Virtualization (NFV).

Ronald Marx is the deputy head of the "Mobile Networks" at the Fraunhofer Institute for Secure Information Technology (SIT). He received his diploma in computer science at the Technical University of Darmstadt (TUD). Since 2005, he was involved in numerous projects, as project staff and project manager. His work focuses on the security aspects in next generation networks (NGN), the mobility and identity management and voice over IP communications.

DR. KPATCHA BAYAROU received his Diploma in electrical engineering/ automation engineering in 1989, a Diploma in computer science in 1997, and his Doctoral degree in computer science in 2001, all from the University of

Bremen in Germany. He joined the Fraunhofer Institute for Secure Information Technology (Fraunhofer SIT) in 2001. He is the head of the "Mobile Networks" department that focuses on Cyber Physical Systems and Future Internet including vehicular communication. Dr. Bayarou managed several EU and nationally funded projects and published several conference papers related to security engineering of mobile communication systems, mobile network technology, and NGN (Next Generation Networks).