Research Paper

A Comprehensive Security Framework for Heterogeneous IoTs

Dr. Mahdi Aiash

February 8, 2018

Abstract

With Cisco estimating that 50 billion of IoT nodes will be connected to the Internet by 2020, the exponential growth of BGP routing tables and mobility issues will be the two major problems in the Internet. Recently, the Locator Identifier Separation Protocol (LISP) has been proposed as an efficient approach for dealing with the problem of scalable Internet routing. Unfortunately, the anticipated advantage of using LISP as an overlay to support IoTs communications comes with serious security challenges. In IoTs, nodes will be communicating across different networking technologies which deploy different security measures such as authentication and encryption mechanisms. Such heterogeneity increases the attack surface and leaves IoTs' nodes more susceptible to attacks. This highlights the need for a unified security framework for heterogeneous IoTs. This paper provides an overview of our research to address some of the IoTs' security challenges. It proposes a new Node-to-Node (N2N) authentication and key agreement protocol as part of a unified security framework. The underlying protocols of the proposed framework have been formally verified using Casper/FDR, a well-known model checker, and they have been proven to meet a number of desired security properties.

1 Introduction

The explosive increase of the Internet related applications has resulted in continued growth of the size of Border Gateway Protocol (BGP) routing tables, causing severe scalability problems of the Internet. The most fundamental of these is helping to ensure that the routing and addressing systems continue to function efficiently as the number of connected devices increases. The scale of the problem is not expected to ease down especially with the wide spread of the publicly accessible IoTs and multi-homed devices. Although the application of aggregated IP addresses may alleviate the problem to a certain extent, the problem cannot be solved essentially due to the semantics overload of IP address, i.e., IP address represents identity information and location role simultaneously. To deal with the drawback of IP address, the Internet Routing Task Force (IRTF) work group has proposed the idea of locator and identifier separation (also known as the Locator/ID Separation Protocol LISP) [1]. Several implementations to LISP exist at the moment including Cisco Nexus 7000 Series products as well as open source implementations such as OpenLISP [2]. Furthermore, LISP has been proposed as an overlay to support communications in various Next Generation Networks

(NGN) such as the Internet of Things (IoTs) [3] and Information Centric Networks [4].

While LISP sounds a promising solution to address the scalability concerns of the growing Internet, it brings about serious security challenges. Considering the fact that the IoTs' nodes will be communicating over different networking technologies (WLAN, 3G, 4G, etc), this raises a major security concerns in terms of authenticating the devices and securing their communications over these different networks. In our research to address these security challenges, we argue that considering the diversity of technologies that might potentially adopt LISP architecture, there is a need for a unified security framework to address the challenges of IoTs' communications over LISP architecture. Our research initially evaluates the security of the LISP architecture and highlights major security threats against LISP. These threats manifested themselves in the form of poor authentication mechanisms (spoofing attacks, active surveillance) and lack of communication confidentiality (passive surveillance). These threats have been addressed in our previous research in [5] [6]. Building on from these solutions, this paper presents a new authentication and key agreement protocol to secure communications between IoTs' nodes. The proposed protocol integrates with the previous protocols to form a security framework for Node-To-Node (N2N) communication in heterogeneous IoTs.

The rest of the paper is organised as follows. Section 2 introduces the LISP architecture and describes our research to secure the LISP operation. The proposed security framework is presented in Section 3 along with a new AKA protocol for N2N communication. Future work and the conclusion are presented in Section 4.

2 Literature Review

2.1 An Overview of the LISP Architecture

The LISP specifies an architecture and mechanism for replacing the addresses currently used by IP with two separate name spaces: Endpoint IDs (EIDs), used within EID sites, and Routing Locators (RLOCs), used on the transit networks such as the Internet infrastructure. To achieve this separation, LISP defines protocol mechanisms for EID-to-RLOC mapping [8]. Furthermore, LISP assumes the existence of a mapping system in the form of distributed database to store and propagate those mappings globally. The functionality of the mapping system goes through two stages:

- Registration Stage: in this stage, the Map Server learns the EIDs-

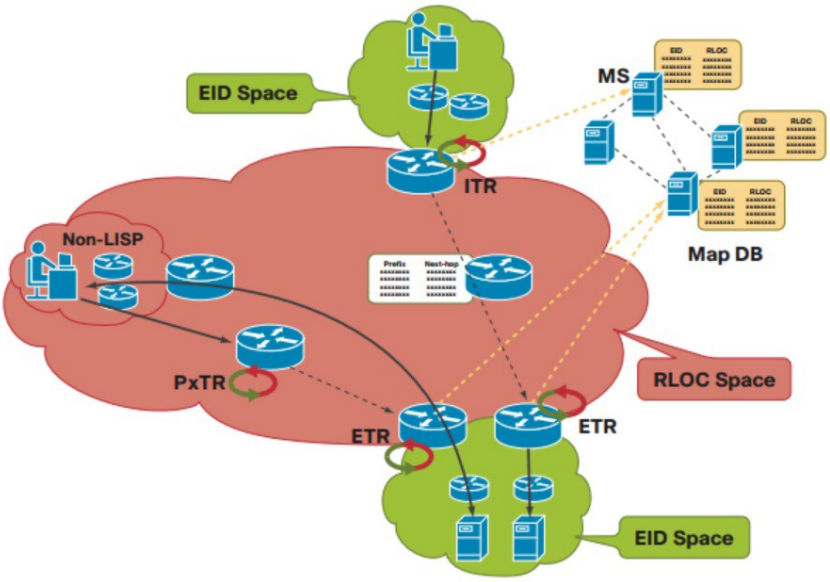

Figure 1: LISP Architecture

to-RLOC mappings from an authoritative LISP-Capable Router and publishes them in the database.

- Addresses resolving Stage: the Map Server (MS) accepts Map-Requests from routers, looks up the database and returns the requested mapping.

The architecture of LISP is presented in Fig 1, three essential components exist in the LISP environment: the LISP sites (EID space), the non-LISP sites (RLOC space), and the LISP Mapping System which comprises Map Servers and databases.

- **The LISP sites (EID space):** they represent customer end-sites in exactly the same way that end-sites are defined today. However, the IP address in the EID space are not advertised to the non-LISP sites, but are published into the LISP Mapping Systems which performs the EID-to-RLOC mapping. The LISP functionalities are deployed on the site's gateway or edge routers. Therefore, based on their roles, two types of routers are defined: firstly, the Ingress Tunnel Routers (ITRs) which receive packets from hosts and send LISP packets toward the Map Server. Secondly, the Egress Tunnel Routers (ETRs) which receive LISP packets from the Map Server and pass them to hosts [9] [1].
- **Non-LISP sites (RLOC space):** it represents current sites where the IP addresses are advertised and used for routing purpose.

- **LISP Mapping Systems:** These are represented by Map Servers (MS) and a globally distributed database that contains all known EID prefixes to RLOC mappings. Similar to the current Domain Name System (DNS), the Mapping systems are queried by LISP-capable devices for EID-to-RLOC mapping.

2.1.1 The Operation of the LISP Components

The functionality of the LISP goes through two stages:

1. **The EID Prefix Configuration and ETR Registration Stage:** As explained in [10], an ETR publishes its EID-prefixes on a Map Server (MS) by sending LISP Map-Register messages which includes the ETR's RLOC and a list of its EID-prefixes. Initially, it has been presumed that prior to sending a Map-Register message, the ETR and Map Server must be configured with a shared secret or other relevant authentication information. Upon the receipt of a Map-Register from an ETR, the Map Server checks the validity of the Map-Register message and acknowledges it by sending a Map-Notify message. When registering with a Map-Server, an ETR might request a no-proxy reply service which implies that the Map Server will forward all the EID-to-RLOC mapping requests to the relevant ETR rather than dealing with them.

2. **The Address Resolving Stage:** Once a Map Server has EID-prefixes registered by its client ETRs, it will accept and process Map-Requests. In response to a Map-Request (sent from an ITR), the Map Server first checks to see if the required EID matches a configured EID-prefix. If there is no match, the Map Server returns a negative Map-Reply message to the ITR. In case of a match, the Map Server re-encapsulates and forwards the resulting Encapsulated Map-Request to one of the registered ETRs which will return Map-Replay directly to the requesting ITR.

The LISP working group in [1] has defined the structure of all the LISP Packets including the Map-Request, the Map-Notify, the Map-Register and the MAP-Reply. As shown in Fig 2, after the Registration and the Address Resolving Stages, nodes can communicate and share information.

2.2 Verifying Security Protocols using Casper/FDR

Previously, analysing security protocols used to be done using two stages. Firstly, modelling the protocol using a theoretical notation or language such

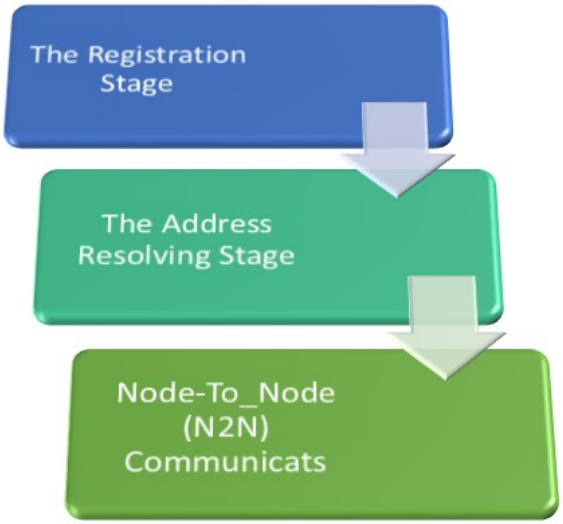

Figure 2: LISP Stages

as the CSP [11]. Secondly, verifying the protocol using a model checker such as Failures- Divergence Refinement (FDR) [7]. However, describing a system or a protocol using CSP is a quite difficult and error-prone task; therefore, Gavin Lowe [7] 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 [12] [13]. The CASPER' s input file that describes the systems and defines the threat model consists of eight headers as explained in Table 1.

2.3 Security of LISP Architecture

The issue of security in LISP architecture and protocol has been investigated in various works such as in [3] [5] [6]. Using the X.805 security standard [3], the authors in [3] investigate the security issues of deploying the Locator/ID Separation Protocol in the Internet of Things. The investigation discovers a number of vulnerabilities that should be considered before moving to the implementation stage. In particular, the papers highlights the need for new mechanisms to enforce Access control, Authentication, Non repudiation, Data confidentiality.

In our research to address some of these challenges, we designed the fol-

Table 1: Casper File Header

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

lowing two protocols to secure the LISP's operational stages namely, the Registration and Address Resolving Stages as described in Section 2.1.1.

ID-Based Authentication Protocol for Securing the Registration Stage: In order to provide a security framework based on LISP architecture, we need first to guarantee that the LISP's operations are secure. Our research in [5] has considered the LISP operation namely the Registration Stage as described in Section 2.1.1. The research argues that one major security threat against the Registration Stage is identity spoofing and masquerading. The research, therefore; proposes a new approach based on the ID-Based Cryptography (IBC) [14]. The IBC helps to certify the messages sender as the real owner of the RLOC that will update the Map Server. The main advantage of using the IBC over traditional Public Key Infrastruc- ture is that since the public key will be derived from the nodes' identifiers, IBC eliminates the need for a public key distribution infrastructure. How- ever, the scheme requires the presence of Trusted Key Generation (TKG) centre/service.

Considering the system's elements in table 2, the proposed protocol in [5] goes as follows:

Msg1. TKG ← ETR: $\{SK(ETR)\}\{K1\}$
Msg2. TKG → MS: $\{SK(SM)\}\{K2\}$

Table 2: Notation for System Elements

The Notation	Definition
TKG	The Trusted Ticket Granting
SK(ETR), SK(MS)	The Private keys of the ETR, MS, respectively. These keys are derived by the TKG
K1, K2	Pre-shared keys to secure the connections between the TKG and ETR, MS
ETR	The Egress Tunnel Router in the destination EID Space
ITR	The Ingress Tunnel Router in the source EID Space
MS	The Map Server
n1, n2	fresh random numbers
h(m)	Hash value of the message (m)
{m}{K}	The message (m) being encrypted with the key (K)

Msg3. ETR → MS:{Map-Register}{PK(MS)}, {h(Map-Register)}{SK(ETR)}
Msg4. MS → ETR: {Map-Notify}{PK(ETR)}, {h(Map-Notify)}{SK(MS)}

The proposed protocol has been formally verified and has been confirmed to meet a number of desired security features.

A Challenge/Response Protocol for Securing the Address Resolving Stage: After securing the Registration Stage, our research in [6] focus on the Address Resolving Stage of LISP operation (Section 2.1.1). It analyses the security and functionality of the LISP mapping procedure where a LISP-capable router approaches the Map Server with a Map-Request message and expects the required EID-to-RLOC mapping in a Map-Replay message. The work points out several security issues in the protocol such as the lack of data confidentiality and mutual authentication.

To address these issues, the paper proposes a challenge/response authentication protocol that is compatible with the implementation of the LISP. Following the same notations in Table 2, the protocol goes as follows:
Msg1. ITR→ MS: {ITR,N1,MapRequest,K3,h(ITR,N1,MapRequest,K3)}{K1}
Msg2. MS→ ETR:{ ITR,N1,MapRequest,K3,h(ITR,N1,MapRequest,K3) }{K2}
The ITR composes Msg1 and includes a freshly generated secret key (K3) to be used by the ETR to encrypt the Map-Reply packet. This message is forwarded by the MS towards the ETR.

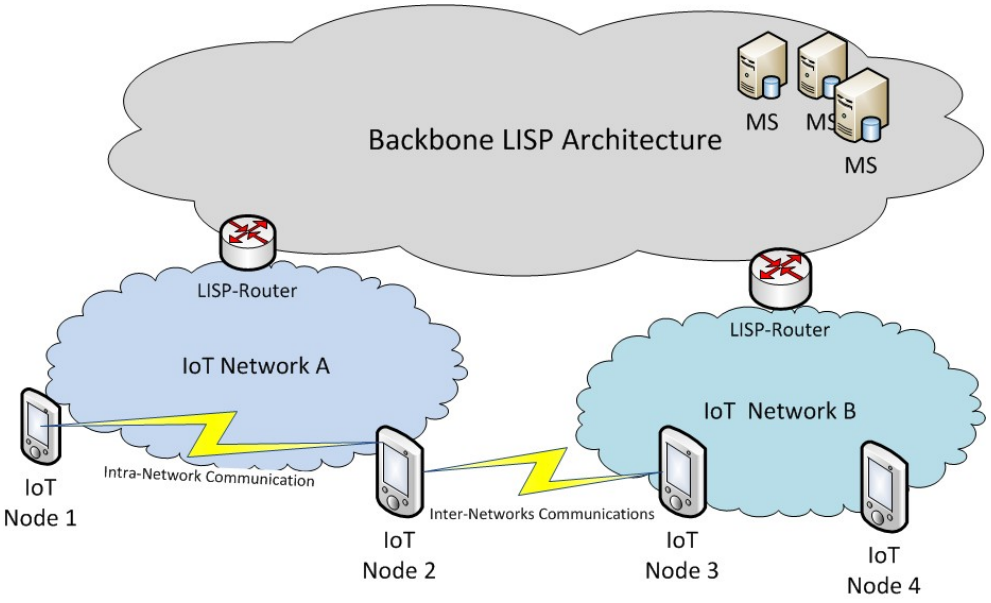

Figure 3: N2N Communication using the LISP Architecture

Msg3. ETR → ITR: {ETR, N1, N2 MapReply,h(ETR, N1, N2 MapReply,)}{K3}

Upon receiving the Map-Request in Msg2, the ETR replies with a Map-Reply message with a challenge nonce (N2). The message is encrypted using the suggested key (K3).
Msg4. ITR → ETR: {N2}{K3}
The ITR returns the challenge (N2) encrypted using the key (K3). The ETR will check the returned challenge to authenticate ITR.

3 The Proposed Protocol for Secure N2N Communications in IoTs

After securing the LISP operational stages, this section describes a new AKA protocol for securing communications between different nodes. These communications could be within the same peripheral network (Intra-Network Communications) or between different IoTs' networks (Inter-Network Communications) as shown on Fig 3.

3.1 AKA Protocol for N2N Communications

The proposed protocol assumes no previous knowledge between the nodes and the LISP network, namely between the nodes and the LISP-Routers. Therefore, assumptions such as pre-shared secrets in forms of encryption or authentication keys will not be possible. This implies that our proposed AKA protocol needs to take this into account, hence, our protocol comprises two stages:

- **Stage 1:** This stage adopts the concept of a Zero-Knowledge Authentication (ZKA) which has been deployed in various security protocols. For simplicity, we consider the communication between Node_i and a LISP-Router (ETR) whereas Node_i and ETR know function f and g, respectively and the two functions are hard to invert and always meet the equation: $g(f(x), f(y)) = f(g(x, y))$, whereas x,y are random numbers of the producer's choice. The proposed ZKA goes as follows:

  Msg1. Node_i → ETR: Node_i, X0, f(X0)
  Msg2. ETR → Node_i: Y0, g(X0, Y0)
  Msg3. Node_i → ETR: f(Y0), f(g(X0,Y0))

  ETR will then calculate g(f(X0),f(Y0)) and compares it to f(g(X0,Y0)) sent by ETR. If the values are the same, B can conclude, with probability 1/m, where m is the size of the range of the function f, that the node is who it claims to be. If a greater level of confidence is desired, ETR can repeat the process with different values of Y and the corresponding g(X0,Y0) to increase the probability to 1/m2, repeat again for 1/m3, etc. After (n) round of the protocol, both entities; the Node and the ETR can agree on a key value as follows: $K = n * g(f(Xn-1), f(Yn-1))$. This protocol will be executed by each node (Node_i) in the network and a key (K_i) will be generated between the ETR and the Node_i

- **Stage 2:** At the end of Stage 1, both communicating entities namely, the ETR and the Node_i will have achieved a high-level of confidence of the identity of the other communicating entity and will have agreed on a symmetric key. The aim of this second stage (Stage 2) is to generate a secret key between any two communicating nodes such as Node_i, Node_j for instance. These nodes are controlled by the same ETR which as has been described in Section 2.3 will act as KDC. Following the terminology in Table 3. The protocol in Stage 2 goes as follows:

Table 3: Notation for System Elements

The Notation	Definition
Node_i, Node_j	The two communicating nodes
ID_i, ID_j	The Device identities such as MAC address
K_i, K_j	Pre-shared keys between the ETR/KDC and Node_i, Node_j respectively. These keys have been generated as per the first stage (Stage 1)
N1, N2	Are randomly generated nonces
Ks	Secret key to be used between Node i and Node _j. The secrecy of this key is the main gaol of the protocol
{m}{K}	The message (m) being encrypted with the key (K)

Msg1. Node_i → Node_j: ID_i, N1
The first node (Node_i) initiates the communication with the other node (Nod_j) by sending its identity (ID_i) and a fresh nonce (N1).
Msg2. Node_j → KDC: ID_j, {ID_i, N1, N2}{K_j}
The node (Node_j) receives msg1, and composes msg2 towards the KDC. The message is encrypted using the key K_j. Upon recipient of the message, the KDC realizes the intention of the Node_j to communicate with Node_i. The KDC decrypts msg2, authenticates Node_j and composes msg3.
Msg3. KDC → Node i: {ID_i, N1, Ks}{K_j},N2, {ID_i, Ks}{K_i}

In this message, the KDS, generates a secret key (Ks) to be used between the communicating nodes (Node i and Node _j).
Msg4. Node i →Node j : {ID_i, N1, Ks }{K_j}, N2 }{Ks}
On receiving msg3, Node i extracts part of the message {ID_i, N1, Ks}{K_j} and passes it to Node j. It also includes a challenge (N2) encrypted with Ks.
Msg5. Node_j →Node i: {N1 ||N2 }{Ks}
After receiving msg4, Node_j extracts the secret key (Ks) and uses it compose a response (N1||N2) as in msg5.

3.1.1 Formal Analysis Using Casper/FDR

To formally analyse the proposed solution, we simulate the system using Casper/FDR tool. For conciseness, only the #Protocol Description, the #Specification and the #Intruder Information headings of the protocol de-

scription are described here, since the rest are of a less significance in terms of understanding the verification process.

The #Protocol description heading defines the system and the transactions between the entities.

```
#Protocol description
0.  -> Node_i :  Node_j, KDC
1.  Node_i -> Node_j :  ID_i, N1
2.  Node_j -> KDC : ID_j, {ID_i, N1, N2}{K_j}
3.  KDC -> Node_i :  {ID_i, N1, Ks}{K_i},N2, {ID_i, N1, Ks}{K_j}%w
4.  Node_i -> Node_j :  w%{ID_i, N1, Ks}{K_j}, {N2}{Ks}
5.  Node_j -> Node_i:  {N1 ||N2}{Ks}
```

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 lines starting with the Agreement define the protocol's authenticity properties.

```
#Specification
Secret(Node_i, Ks, [Node_j, KDC])
Secret(Node_i, K_i, [KDC])
Secret(Node_j, K_j, [KDC])
Agreement(Node_i, Node_j, [N2])
Agreement(Node_j, Node_i, [Ks])
```

The # Intruder Information heading specifies the intruder's 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 the intruder knows the identity of the participants.

```
#Intruder Information
Intruder = Mallory
IntruderKnowledge = {node_i, node_j, Mallory, kdc , id_i, id_j, n1}
```

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 4.

3.1.2 Security Analysis

As stated in [15], it is desired for AKA protocols to meet certain security properties. Therefore, a list of these properties will be used to analyse the proposed AKA protocol in this paper.

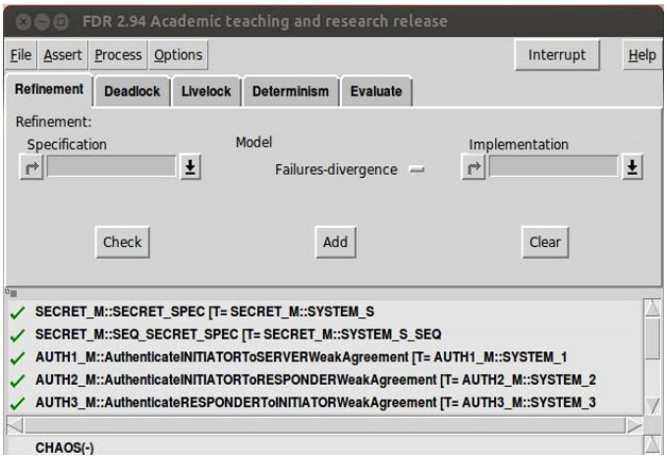

Figure 4: The FDR Formal Verification

- **Mutual Entity Authentication:** This is achieved when each party is assured of the identity of the other party.

  Casper provides no direct specification to model this property. However, considering the protocol transactions, we argue that this requirement could be met to a certain extent in our protocol. This is achieved by including the IDs (ID_i, ID_j) in the exchanged messages, these IDs will be encrypted later on using the keys K_i and K_j which should be known to trusted Nodes that have successfully passed stage 1 of the proposed protocol.

- **Mutual Key Authentication:** This is achieved when each party is assured that no other party aside from a specifically identified second party gains access to a particular secret key.

  The mutual authentication between the nodes is based on the secrecy of the Ks. We got Casper to check this using the Secret(Node_i, Ks, [Node_j, KDC]) assertion check.

- **Mutual Key Confirmation:** This requirement means that each party should be assured that the other has possession of a particular secret key.

  This is the main purpose of the Stage 1 of the proposed protocol; the more the rounds of the protocol, the more confident the nodes become of the identities of each other and the secrecy of the derived key.

- **Key Freshness:** a key is considered fresh if it can be guaranteed to be new and not reused through actions of either an adversary or authorized

party.

This feature is related to the key derivation functions, and hence there is no direct function with Casper to simulate this feature. However, the fact that Casper does not detect any attack on the secrecy of the secret key (Ks) implies that key freshness is not violated.

- **Unknown-Key Share:** In this attack the two parties compute the same session key but have different views of their peers in the key exchange. In other words, in this attack an entity A ends up believing that it shares a key with B; although this is the case, B mistakenly believes the key is instead shared with an entity E ≠ A. This feature has been simulated using the Agreement assertions in the #Specification section

4 A Security Framework based on LISP Architecture

While the protocols described in Section 2.3 address the security within the LISP architecture, the proposed protocol in Section 3.1 provides measures for authentication and key management between communicating nodes in the peripheral networks. These three protocols form the underlying mechanisms of our proposed network-level, security framework for secure IoTs communications as shown in Fig 5. However, in order to provide a comprehensive solution, it is crucial for all these protocols and security measures to integrate and work together.

Therefore, as a proof of concept, we will show how these protocols could integrate. The sequence below shows the messages of the proposed protocol discussed above then in **Bold** it shows the actual message(s) when integrated with the protocol of [5]. It is important to point out that this is only Stage 2 of the protocol. We assume (for simplicity) that Stage 1 has already run and consequently, the keys K_i and K_j have been derived. Furthermore, since it is a proof of concept, we will consider intra-networks communications (both Nodes (Node_i and Node_j) are within the same peripheral network).

Msg1. Node_i → Node_j: ID_i, N1
Msg 1.1: Node i → ITR: {ID i, N1}{K i}
Msg 1.2: ITR → Node j: {ID i, N1}{K j}
Msg2. Node_j → KDC: ID_j, {ID_i, N1, N2}{K_j}
Msg3. KDC → Node_i: {ID_i, N1, Ks}{K_j},N2, {ID_i, Ks}{K_i}
Msg4. Node_i → Node_j : {ID_i, N1, Ks}{K_j}, {N2}{Ks}

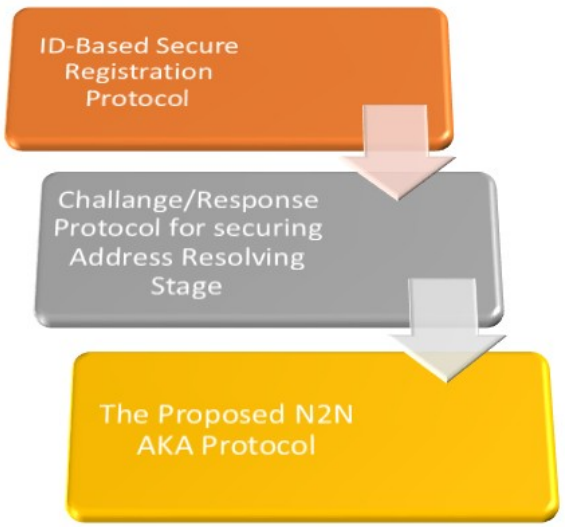

Figure 5: The Proposed Framework

Msg 4.1: Node i → ITR : *{ID i, N1, Ks}{K j}, {N2}{Ks}*
Msg 4.2: ITR → Node j : *{ID i, N1, Ks}{K j}, {N2}{Ks}*

Msg5. Node j → Node i: *{*N1 ||N2*}{*Ks*}*

Msg 5.1 : Node j → ITR: *{{N1 ||N2}{Ks}}{K j}*
Msg 5.1 : ITR → Node i: *{{N1 ||N2}{Ks}}{K i}*

5 Conclusion and Future Work

In order to accommodate the increased number of communicating devices in the era of IoTs, the LISP protocol has been proposed by the IETF and Cisco research center to enhance the Internet's scalability. The security challenges of this new deployment has been discussed in the literature. Major concerns were related to authentication and traffic confidentiality. Therefore, this paper presents a security framework in the form of AKA protocols to address these security threats. Our security framework mainly focuses on achieving nodes authentication and maintaining communications' confidentiality and integrity between nodes, hence it protects against both passive and active traffic surveillance. Other desired security properties such as access control, privacy as well as upper-layers (application layer) security integration are beyond the scope of this paper and shall be considered as part of our on-going research.

References

[1] Farinacci, D., Fuller, V., Meyer, D., Lewis, D.: Locator/ID Separation Protocol (LISP). Internet-Draft , November 13, 2012.

[2] Dung P, Stefano S, Damien S.: The OpenLISP control plane architecture. IEEE Network, Volume: 28 Issue: 2, 2014.

[3] Ali R, et al.: Supporting communications in the IoTs using the location/ID split protocol: A security analysis. The Second International Conference on Future Generation Communication Technology (FGCT). 2013

[4] Mohammad M, et al.: Supporting Communication in Information Centric Networks Using the Location/ID Split Protocol and Time Released Caching. International Conference on Cloud Computing, 2015.

[5] Mahdi A.: Securing Address Registration in Location/ID Split Protocol using ID-Based Cryptography. 11th International Conference on Wired/Wireless Internet Communications WWIC 2013.

[6] Mahdi A.: A Novel Security Protocol for Resolving Addresses in the Location/ID Split Architecture. The 7th International Conference on Network and System Security (NSS 2013).

[7] Lowe, G., Broadfoot, P., Dilloway, C., Hui, M. L.: Casper: A compiler for the analysis of security protocols, 1.12 ed., September 2009.

[8] Locator/ID Separation Protocol (lisp)Working Group. http://datatracker.ietf.org/wg/lisp/charter/

[9] Cisco. Locator/ID Separation Protocol Architecture. http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/locator-id-separation-protocol-lisp/white paper c11-652502.html

[10] Farinacci, D., Fuller, V.: LISP Map Server Interface. Internet-Draft , March 4, 2012.

[11] Goldsmith, M., Lowe, G., Roscoe, A.W., Ryan,P., Schneider, S.: The modelling and analysis of security protocols, PEARSON Ltd, 2010.

[12] Loo. J, Aiash. M.: Challenges and solutions for secure information centric networks. Journal of Network and Computer Applications. Volume 50, April 2015.

[13] Aiash. M, Loo. Jonathan.: An integrated authentication and authorization approach for the network of information architecture. Journal of Network and Computer Applications. Volume 50 Issue C, April 2015.

[14] Shamir,A., Identity-based cryptosystems and signature schemes, in Proceedings of CRYPTO 84 on Advances in cryptology. New York, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 4753.

[15] A. Menezes, P. van Oorschot, S. Vanstone,Handbook of Applied Cryptography, CRC Press, Boca Raton, FL, USA, 1996.