B. Tech (IT) Major Project Presentation – 2020

Study and Analysis of Authentication Schemes in 6LoWPAN Network

Presented by:

Sonu Saha (BT/IT/1654)

Daisy Baruah (BT/IT/1608)

Amit Deka (BT/IT/1604)

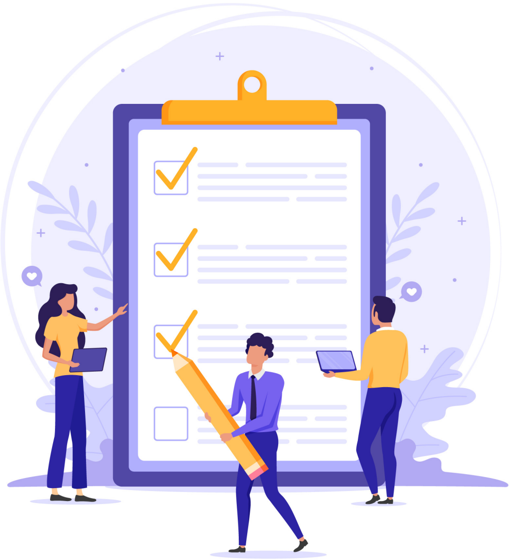
Under the supervision of

Dr. Subhas Ch Sahana

Department of Information Technology

North Eastern Hill Univeristy, Shillong – 22

Contents



Objective

1. Study and Implementation of different Authentication Schemes in 6LoWPAN Network viz.

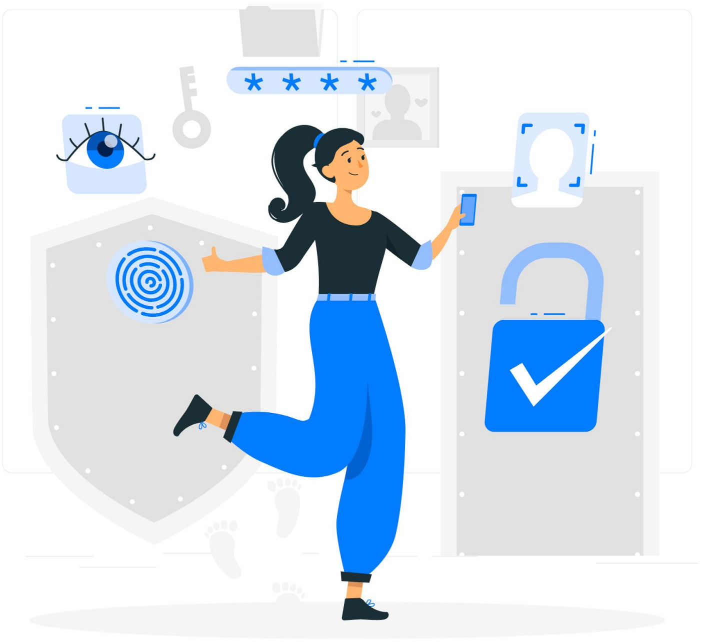
SAKES – Secure Authentication and Key Establishment Scheme

EAKES6Lo – Enhanced authentication and key establishment scheme for M2M communications in the 6LoWPAN networks

S6AE – Securing 6LoWPAN using Authenticated Encryption

2. Performance Analysis of the above mentioned schemes.

Introduction – Authentication



Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system[1].

Two phases of Authentication:

1. Identification
2. Authentication

Example: entering a username and password when we log in to a website.

Introduction – 6LoWPAN

6LoWPAN is an acronym for IPv6 over Low -Power Wireless Personal Area Networks

A 6LoWPAN system is a low-power wireless mesh network where every node has its own IPv6 address allowing it to connect directly to cloud.

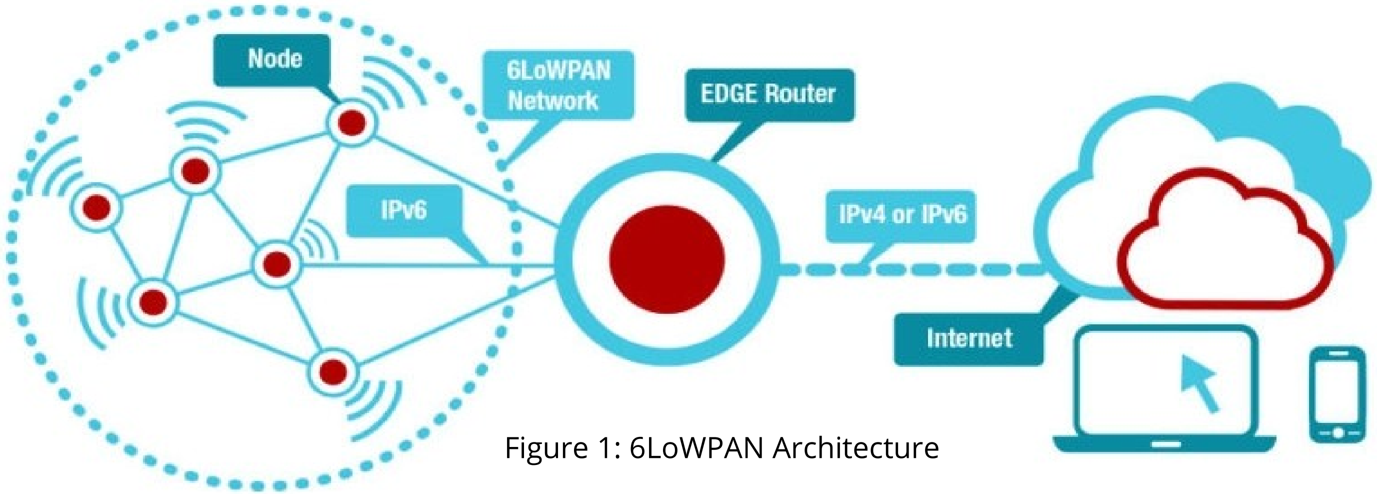


Figure 1: 6LoWPAN Architecture

Introduction – 6LoWPAN (cont.)

Advantages:

1. Mesh Routing
2. Uses open IP standards
3. Offers End-To-End IP Addressable Nodes

Applications:

1. Industrial monitoring
2. Smart Agriculture
3. Home Automation

Authentication Scheme – SAKES

SAKES is an acronym for Secure Authentication and Key Establishment Scheme

This scheme was published in the paper SAKES: Secure Authentication and Key Establishment Scheme for M2M communication in the IP-Based Wireless Sensor Network (6LoWPAN) by **Hassen Redwan Hussen, Gebere Akele Tizazu, Miao Ting, Tackkyeun Lee, Youngjun Choi and Ki-Hyung Kim** in Proceedings of the 2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN).

This protocol consist of **two** phases:

1. **Authentication Phase**
2. **Key Establishment Phase**

Assumption: all nodes which have registered in the network are static.

SAKES – Authentication Phase

The procedures to be followed in the authentication phase:

1. 6ED broadcasts HELLO request to search for neighbour 6LRS.
2. 6ED gets HELLO response from neighbour 6LR.
3. 6ED sends the request message to the 6LBR-AM through the neighbouring 6LR.

   Request Message = (MAC, Ci, ID_{6ED},N_{6ED})

   where MAC = hash(Ci, K6ED6LR, 6ID6ED, N6ED)

   and Ci is ciphered text : (ID6ED, ID6LR, IDServer, N6ED)K6ED6LBR
4. 6LR forwards the request of 6ED to the 6LBR-AM.

SAKES – Authentication Phase (cont.)

Check MAC of the received message

IF success

THEN Create MAC:

MAC = hash(Ci, K6LR6LBR, ID6LR, N6LR)

Send (MAC, Ci, ID6LR, N6LR) to 6LBR-AM.

5. 6LBR-AM checks the authenticity of the 6ED and the 6LR.

   (a) IF 6LR is registered in the 6LBR-AM

   THEN Send to 6LR:

   N6LBR, [ID6ED,IDServer]Kpriv6LBR, [Kpub,Kpriv]K6LR6LBR

SAKES – Authentication Phase (cont.)

(b) ELSE IF

6LR is not registered in the 6LBR-AM

THEN

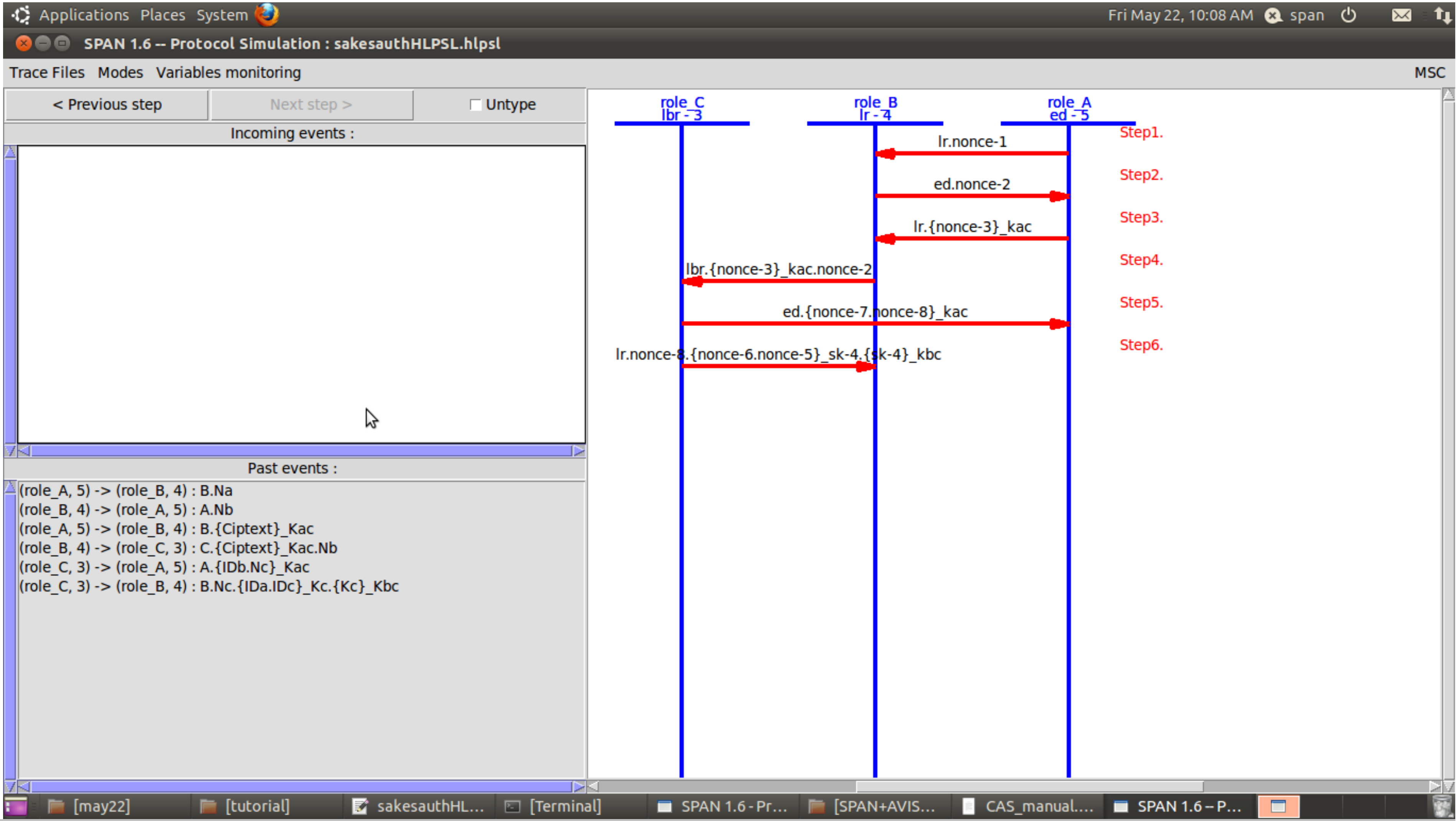
Broadcast the identity of the suspected 6LR to other nodes in the 6LoWPAN

Look for authentic 6LR nearest to the 6ED from the registry of the 6LBR-AM

Decide the correct path to reach the request

Go to (a)

SAKES – Authentication Phase Simulation in AVISPA



SAKES – Key Establishment Phase

The procedures for secure session key establishment between the communicating parties are described with the following pseudo-code:

1. 6LR sends request message of the following form to the server.

   Cipher-text received from 6LBR:

   Ci = (ID6ED, ID6LR, IDServer)KP riv6LBR

   Create MAC:

   MAC = hash(Ci, ID6LR, N6LR, KPub6LR)

   6LR sends (MAC, Ci, 6LRID, N6LR)KPriv6LR to the server.

2. Server checks MAC of the received message

SAKES – Key Establishment Phase (cont.)

IF MAC is valid

Calculate SK based on the received message, prime modulus p and generator g.

Create MAC:

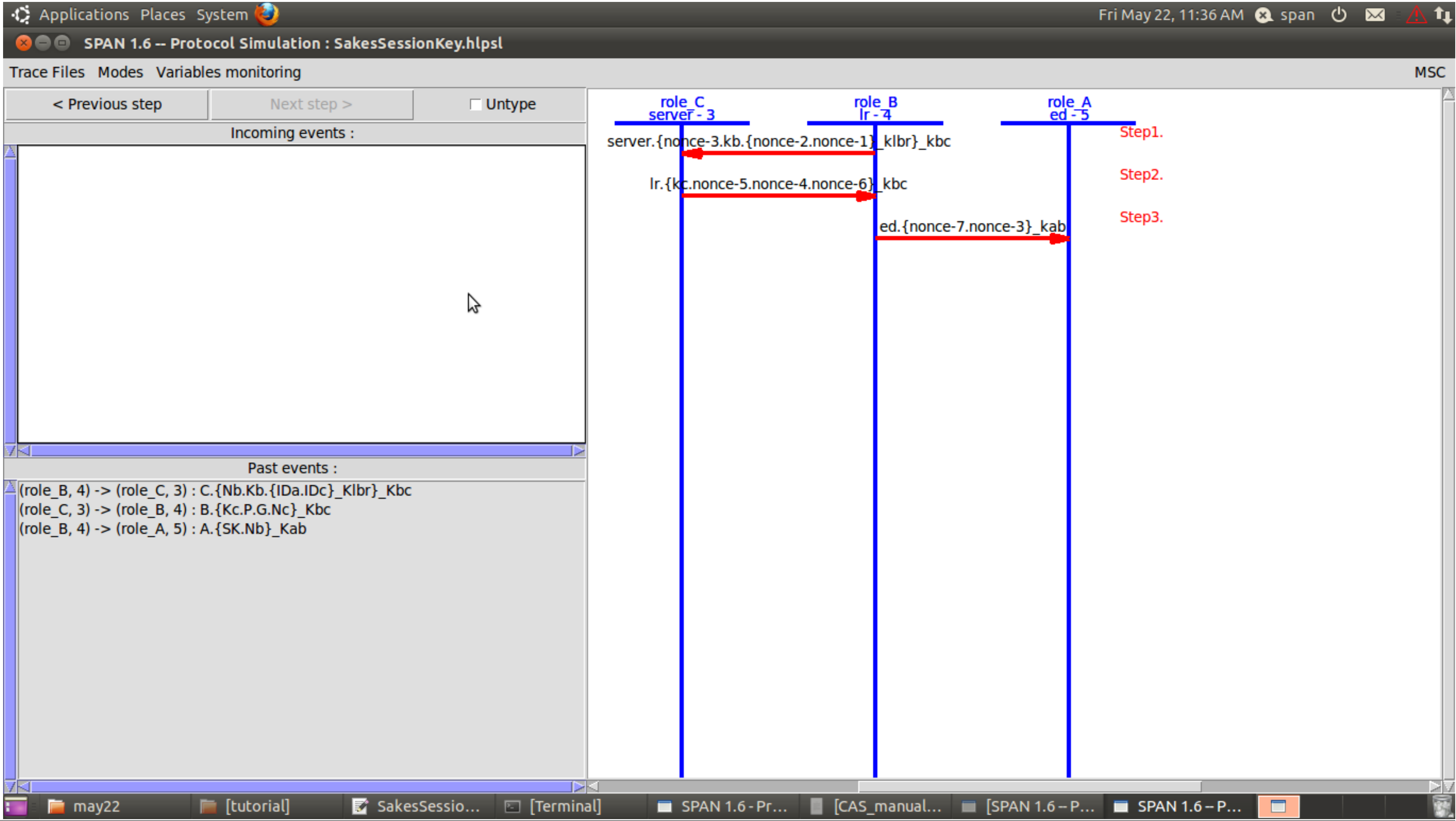
MAC = hash(IDServer, KPubServ, p, g, NServ)

Server sends reply message of the form:

(MAC, IDServer, p, g, NServ)KPrivServ to 6LR.

3. On the receipt of the reply message from the server, the 6LR checks the MAC and if successful, it calculates the SK (using Diffie-Hellman (DH) key exchange mechanism) on behalf of the 6ED as the 6ED is resource constrained.
4. 6LR sends SK to the 6ED. 6ED uses SK to communicate with the server.

SAKES – Key Establishment Phase Simulation in AVISPA



Authentication Scheme – EAKES6Lo

EAKES6Lo is an acronym for Enhanced authentication and key establishment scheme for M2M communications in the 6LoWPAN networks

This scheme was published in the paper A Mutual Authentication and Key Establishment Scheme for M2M Communication in 6LoWPAN Networks by **Yue Qiu and Maode Ma,** IEEE Transactions on Industrial Informatics.

This protocol consists of **three** phases:

1. predeployment phase,
2. AKE phase, and
3. handover phase.

EAKES6Lo – Predeployment Phase

Also known as the Registration Phase

The procedures to be followed in the predeployment phase:

1. The node i sends a registration request to the server with its ID and its public key.
2. Server checks if ID is registered or not.

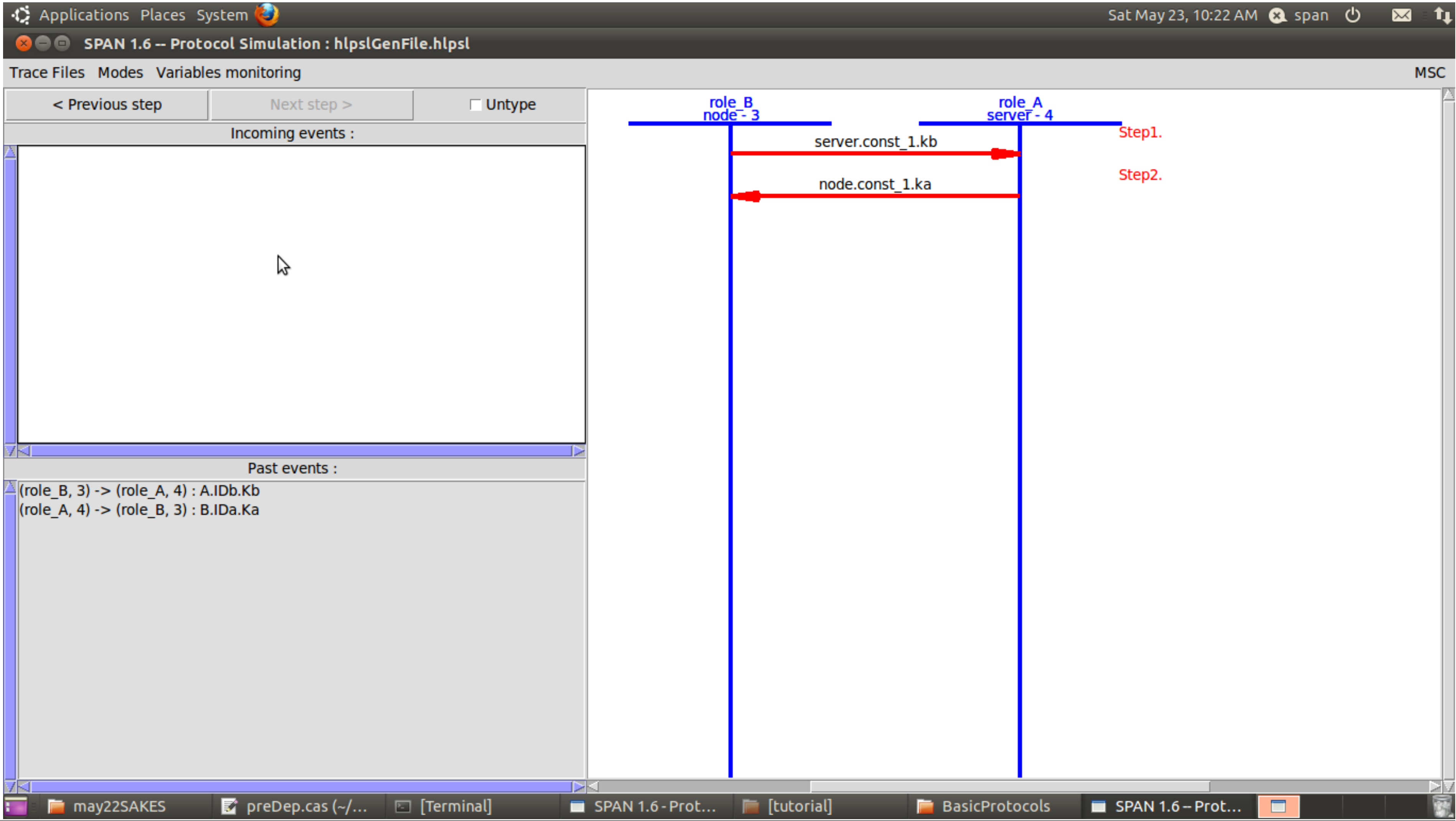
    IF not been registered

        Send its public key and a randomly generated number Si in response.

        Store IDi, Kpubi and Si in the database

(IDi, Kpubi and Si will be further used to produce the symmetric key between the node i and the remote server using Elliptic Curve Diffie-Hellman (ECDH))

EAKES6Lo – Predeployment Phase Simulation in AVISPA



EAKES6Lo – AKE Phase

The procedures to be followed in the AKE phase are as follows:

1. 6LH sends a **Message 1** to 6LR

   Message 1 = {ID6LH,ID6LR,T6LH, (N6LH)K6LH SER} Kpri6LH

2. 6LR checks if the signature is valid and the identity of the router contained in the message.

   IF the checks are passed, forward **Message 1** to 6LER.

3. 6LER checks if the signature is valid.

   IF valid, compare identities of 6LH and 6LR with the registered ID list.

   IF comparison fails, inform other 6LRs and 6LHs, add that 6LR to black list.

EAKES6Lo – AKE Phase (cont.)

IF comparison is passed, **Message 1** is signed with 6LER timestamp and the MAC1 value and sent as **Message 2** to server

Message 2 = Message 1 | {T6LER|MAC1}Kpri6LER

MAC1 = HashHK(T6LER, ID6LH, ID6LR, ID6LER, Kpub6LH, Kpub6LR, Kpub6LER)

4. Server checks if time of transmission < threshold time. IF true,
   a. Retrieve information from the list of registered device.

      IF information is not available

      Broadcast a warning message to the 6LoWPAN network that 6LH is an illegitimate device.

EAKES6Lo – AKE Phase (cont.)

IF information is available

i. Verify the signature signed by 6LH and 6LER.

ii. Compare the MAC1 value to the hash of (T6LER, ID6LH, ID6LR, ID6LER, Kpub6LH, Kpub6LR, Kpub6LER) which must be equal.

iii. Derive the nonce N6LH from the encrypted message.

iv. Generate a ticket T to be used in handover phase.

   T = {ID6LH, IP v66LH, Kpub6LH, TSER, TEXP} KpriSER

v. Compose a **Message 3** with the ticket T and a MAC2, send to 6LER

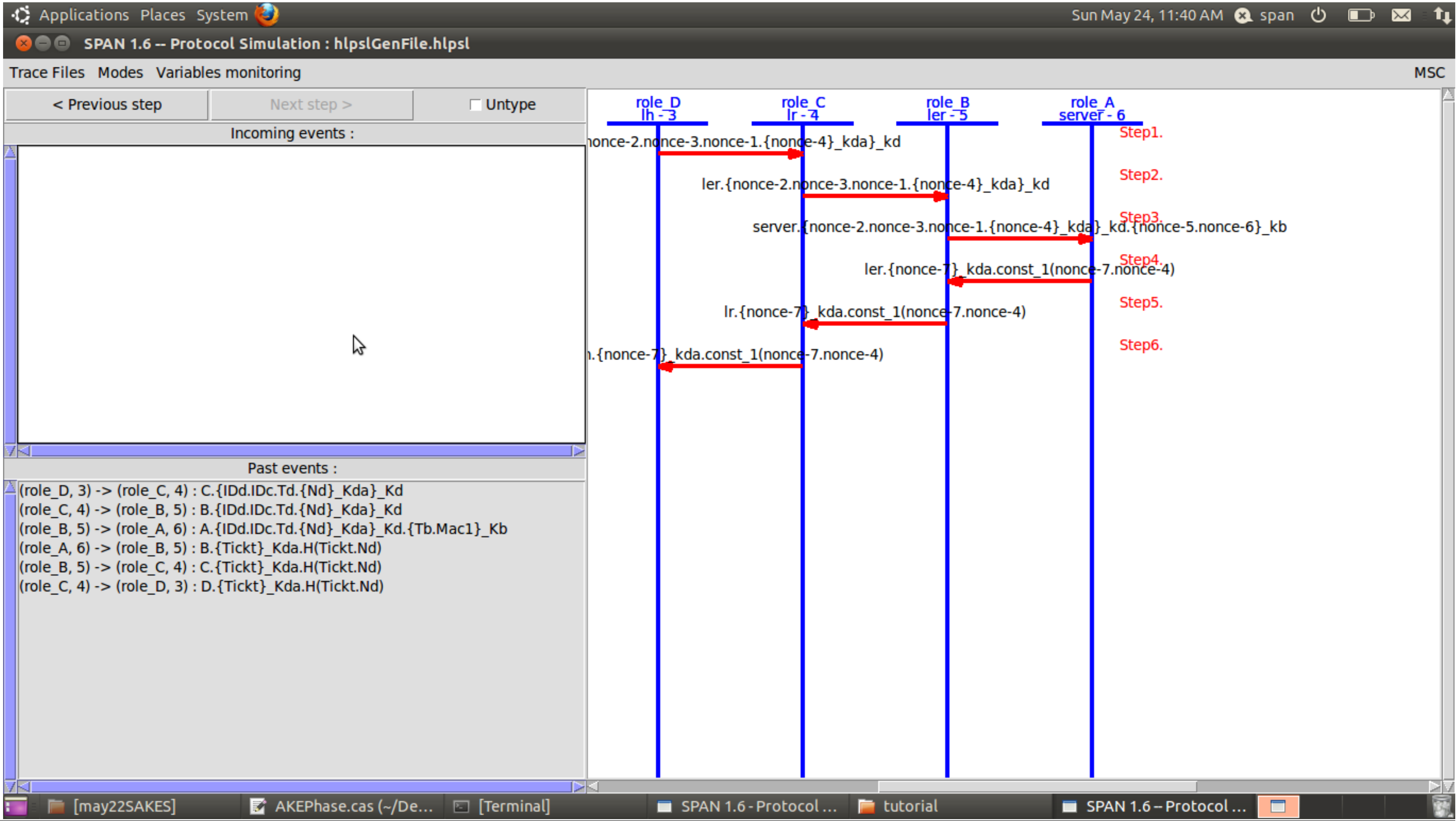
   MAC2 = HashHK (T|N6LH).

EAKES6Lo – AKE Phase (cont.)

5. 6LER verifies the signature of **Message 3** and stores ID6LH and forwards it to 6LR.
6. 6LR verifies the signature of **Message 3** and stores ID6LH and forwards it to 6LH.
7. 6LH checks the MAC2 value of **Message 3,** if valid stores the ticket T to use in handover phase.

The 6LH is successfully authenticated with the remote server and a secure connection between them has been set up.

EAKES6Lo – AKE Phase Simulation in AVISPA



EAKES6Lo – Handover Phase

The procedures to be followed in the Handover phase are as follows:

1. 6LH (the mobile node) sends **Message 1** as request to 6LR2 (new 6LR).

   Message 1 = Ticket | Timestamp | MAC3

   MAC3 = HashHK(Ticket|T6LH)

2. 6LR2 checks the expiry of the Ticket.

   IF Ticket is expired

       Discard the request

   ELSE

       Verify signature and value of MAC3. On verification, forward **Message1** to server.

EAKES6Lo – Handover Phase (cont.)

3. On receiving **Message 1,** server verifies the autheniticity of 6LH and 6LR2.

   IF verified, send notification to 6LR1 (old 6LR) that 6LH is no longer in its area.

   6LR1 checks if 6LH is out of range. If true, deletes the device from member list and sends an acknowledge message (ACK) to the server.

   On receiving ACK from 6LR1, server sends **Message 2** to 6LR2.

   Message 2 = {ID6LH, ID6LR2, TSER, MAC4} KpriSER| {Ns} K6LR2 SER | {Ns}K6LH SER

   MAC4 = HashHK (ID6LH, ID6LR2, TSER, Ns)

   Ns is a random number (*used for establishing a symmetric key shared between the 6LH and 6LR2)*

EAKES6Lo – Handover Phase (cont.)

4. 6LR2 checks if the signature, timestamp and MAC4 is valid. IF valid, it stores the random number and forwards rest of the message as **Message 3** to 6LH

   Message 3 = {ID6LH, ID6LR2, TSER, MAC4} KpriSER | {Ns}K6LH SER

5. 6LH verifies the validity of **Message 3** and stores the random number.

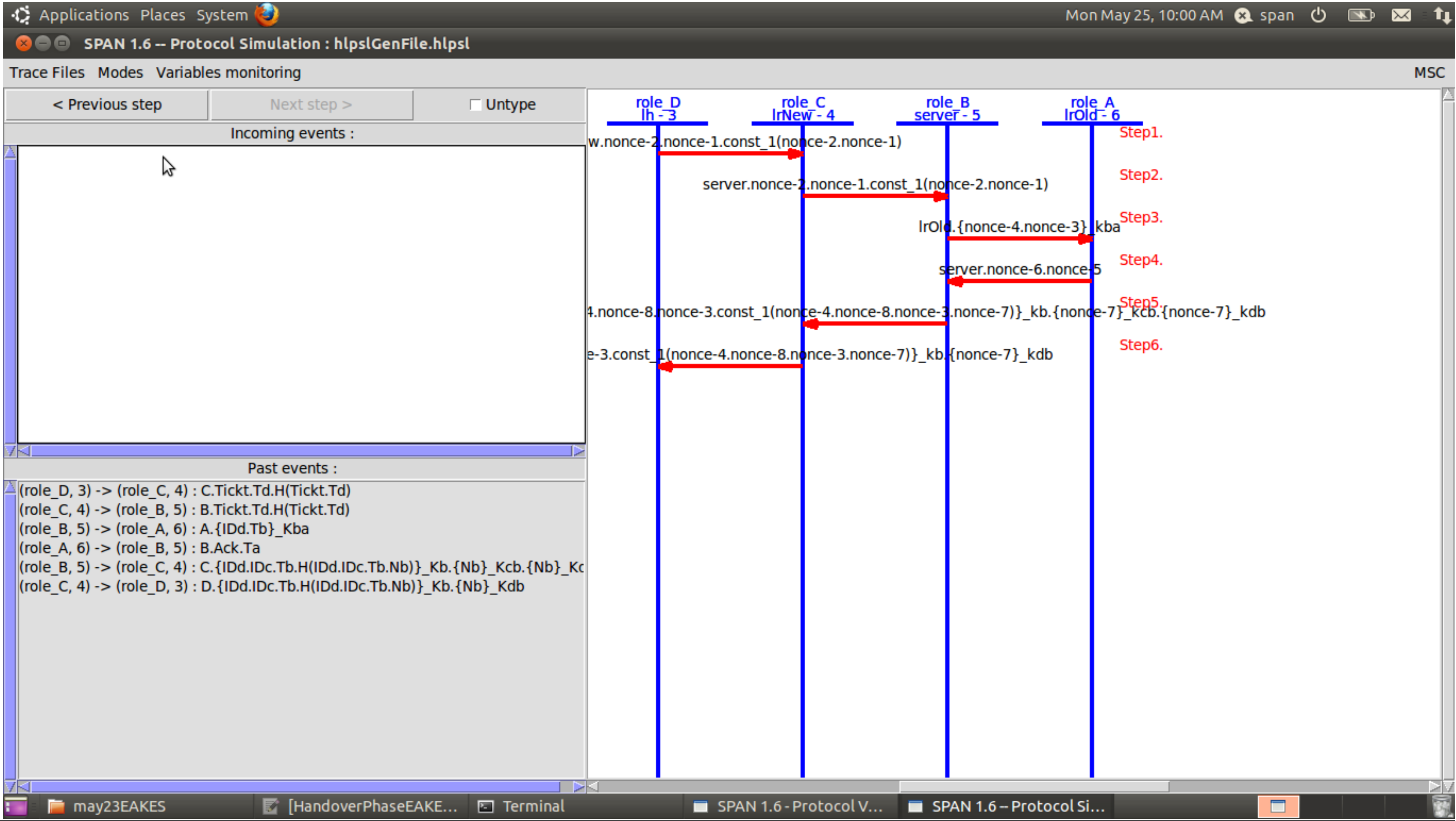
   Now, the symmetric key shared between the 6LH and the 6LR2 can be calculated as follows: K6LH 6LR2

   = Hash(Kpri 6LH * Kpub6LR2,Ns)

   = Hash(Kpri6LR2 * Kpub6LH ,Ns)

   = K6LR2 6LH

EAKES6Lo – Handover Phase Simulation in AVISPA



Authentication Scheme – S6AE

S6AE is an acronym for Securing 6LoWPAN using Authenticated Encryption

This scheme was published in the paper S6AE: Securing 6LoWPAN Using Authenticated Encryption Scheme by Muhammad Tanvee, Ghulam Abbas, Ziaul Haq Abbas, Muhammad Waqas, Fazal Muhammad and Sunghwan Kim, ncbi.nlm.nih.gov

S6AE consists of the three phases:

1. registration phase,
2. the AKE phase, and
3. the handover phase..

S6AE – Registration Phase

This phase deals with the registration of SN before its deployment in 6LoWPAN. CS performs the following operations to register SNs.

1. Calculates the master key Km by computing Km = H(IDcs|rcs), where IDcs is the real identity of CS and rcs is a random number.
2. CS divides Km into four equal chunks of 64 bits, namely Km1, Km2, Km3 and Km4, and computes Kcs = Km1 ⊕ Km2 ⊕ Km3 ⊕ Km4 , where Kcs is a temporary key for CS.
3. Assigns a unique IDsn of 64 bits for SN.
4. Picks a key Ksn of 64 bits for SN.
5. Computes the pseudo-identity SIDsn = IDsn ⊕ Ksn ⊕ Kcs.

S6AE – Registration Phase (cont.)

6. Computes $Hr = H(Km \mid Ksn \mid IDsn)$ and derives security parameter SP1 by computing $SP1 = Hr1 \oplus Hr2 \oplus Hr3 \oplus Hr4$,

   where Hr1, Hr2, Hr3, and Hr4 are four equal chunks of 64 bit Hr.

7. CS stores {IDsn, SP1, Ksn, Kcs, MACsn} into its database and {IDsn, SP1, SIDsn, Ksn, MACcs} in the memory of SN while making use of a secure channel.

   CS also stores SIDsn into 6LDR memory through a secure channel

S6AE – AKE Phase

Two parameters, Sk and AD are generated by the process of Sponge State Generation & Associative Data Generation respectively to be used by the encryption algorithm of ASCON at first two stage.

The procedures to be followed in AKE phase are as follows:

1. SN generates a random number Rs1 and timestamp Tsn.
   a. Computes: X = IDsn ⊕ Rs1 ⊕ SP1 and Y = IDsn ⊕ Rs1; *SP1 is a secret parameter.*
   b. Encrypt (Sk, AD, X, Y) using ASCON to produce cipher text C1 and Tagsn.

   Tagsn guarantees the authenticity and integrity of the ciphertext C1 at the receiving end. Tagsn provides the same functionality as Message Authentication Code (MAC).

S6AE – AKE Phase (cont.)

c. Computes $Z = SIDsn \oplus SIDldr$, where SIDldr is the temporary identity of 6LDR.

d. Construct a message **M1** : {Tsn | Z | (C1 | Tagsn) | R1} and forward it to 6LDR.

2. 6LDR picks out Z from **M1** and

   Compute: $SIDr = Z \oplus SIDsn$.

   IF SIDr == SIDldr (SIDldr is in its memory)

      Construct new message **M2** : {SIDldr | M1} and forward it to 6LAR.

   ELSE

      Abort the AKE process and send an error message back to SN.

S6AE – AKE Phase (cont.)

3. 6LAR checks SIDldr present in **M2.** If registered, 6LAR:
   a. Pick a timestamp Tlar and compute Hlar = H(M2 | SIDlar | Tlar | Klar).
      where Klar is the pre-shared key between 6LAR and CS, and SIDlar is the temporary identity of 6LAR.
   b. Construct a message **M3 :** {SIDlar | Tlar | M2 | Hlar} and forward it to CS.

   If not registered, abort the AKE process and add SIDldr to blacklist.
4. CS checks SIDlar and retrieves secret information(such as Klar) related to 6LAR.

   Check validity of Tlar (should be received within max transmission delay).

   Compute: H'lar = H(M2 | SIDlar | Tlar | Klar).

S6AE – AKE Phase (cont.)

IF H‘lar == Hlar(present in **M3**), CS retrieves **M2** from **M3** and checks validity for timestamp in **M1**. Also, checks whether a valid SIDldr(present in M2) exists in the current list of 6LDR devices.

If verification is True, CS decrypts C1 present in **M1** which generates Tagcs.

IF Tagcs == Tagsn, CS computes SP1‘.

IF SP1‘ == SP1 (stored in CS database), SN is a legitimate device.

4. IF SN is verified,
   a. CS picks timestamp Tcs, 3 random numbers Rs1, R2 and Rn.
   b. Computes security parameter SP1n, Sk“ and AD to be used in encryption process..
   c. Computes Y1 = Rn$\oplus$Kcs and X1 = Y1$\oplus$Rs1

S6AE – AKE Phase (cont.)

d. Compute session key Kse.

e. Calculate a unique ticket Ticsn and pick expiry time Texp for it.

d. Encrypts (Sk“, AD, SP1n, Rs2) which produces C2 and Tagcs. (*Rs2 = random number*)

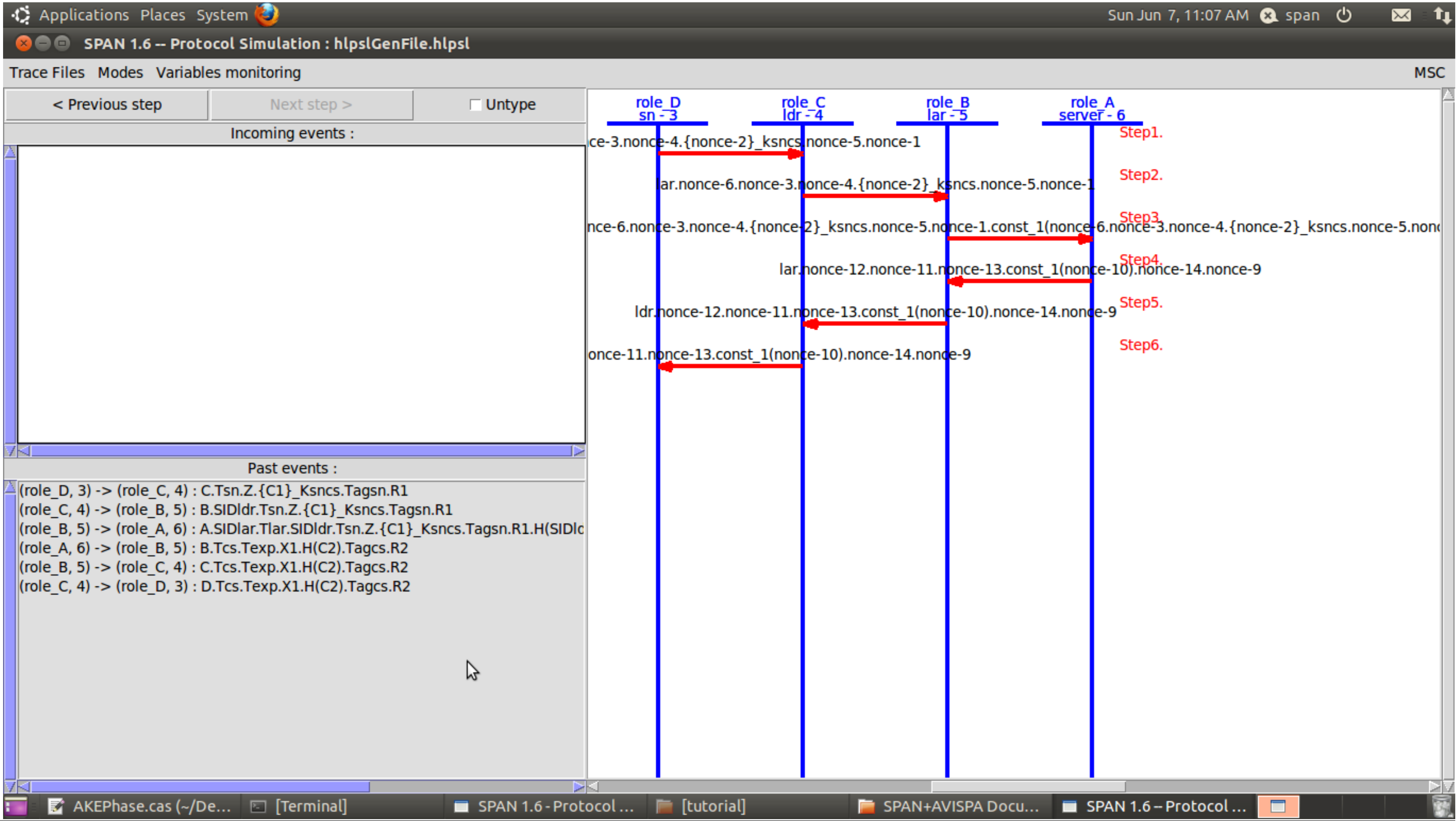
e. Construct message **M4**: (Tcs|Texp|X1|(C2 |Tagcs)|Rs2) and send to 6LAR.

5. 6LAR relays message **M4** to 6LDR and 6LDR relays it to SN.

6. SN checks the validity of Tcs in **M4**. If Tcs is less than maximum time delay, SN decrypts C2 present in M4 which reveals the parameter required to compute a session key and authentication parameter. It also generates a tag, Tagsn.

   If Tagsn‘ == Tagcs and authentication parameter is true, SN is able to compute the key Kse and ticket Ticsn from the revealed text. It stores the Ticsn and Kse in it‘s memory.

S6AE – AKE Phase Simulation in AVISPA



S6AE – Handover Phase

The procedures to be followed in the Handover phase are as follows:

1. SN checks Texp of ticket. If valid, SN picks a timestamp Th and computes Hh.

   Hh = H(Ticsn|Th|SIDsn).

   Construct message **Mh1** : (SIDsn|Th|Ticsn|Hh) and send to 6LDR2 (new 6LDR).

2. 6LDR2 checks integrity of **Mh1** by computing Hh2 = H(Ticsn|Th|SIDsn).

   If Hh2==Hh, stores SIDsn in memory and forward **Mh1** to CS.

3. CS checks integrity of Mh1 by computing Hh3 = H(Ticsn|Th|SIDsn).

   If Hh3=Hh, check if SIDsn exist in database.

   If SIDsn exist in database, check if Ticsn == ticket stored in database.

S6AE – Handover Phase (cont.)

If Ticsn == ticket stored in database, send a message to 6LDR1 to delete SIDsn from its record. 6LDR1 sends an acknowledge message(ACK) back to CS.

4. On receiving ACK, CS computes new session key Ksen.

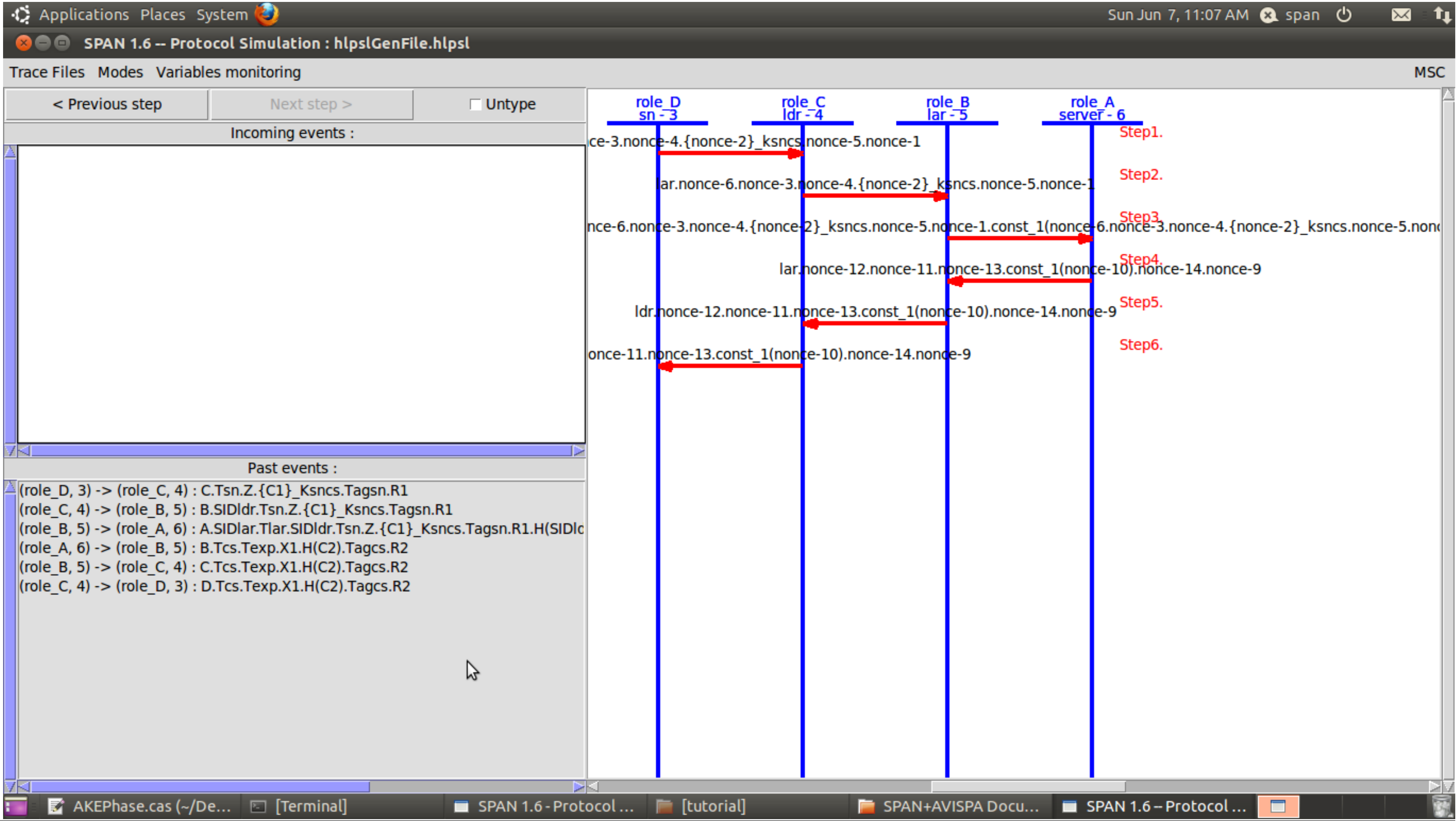
   Construct a new message **Mh2**: (SIDsn|Ch|Tagcs), (*where Ch is an encrypted text and Tagcs is the tag obtained on encryption)* and send it to 6LDR2.

5. 6LDR2 checks SIDsn from **Mh2**. If SIDsn exists in memory, forward **Mh2** to SN.

6. SN decrypts Ch present in Mh2, which reveals the parameter required to compute a new session key Ksen and authentication parameter. It also generates a tag Tagsn.

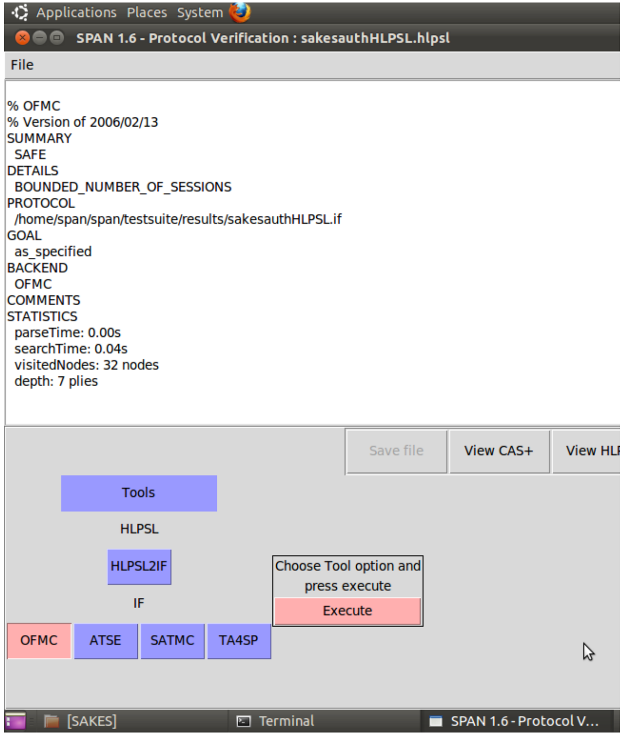
   If Tagsn == Tagcs and authentication parameter is true, replace the old key with Ksen and updates the expiry time Texp of the ticket.

S6AE – Handover Phase Simulation in AVISPA

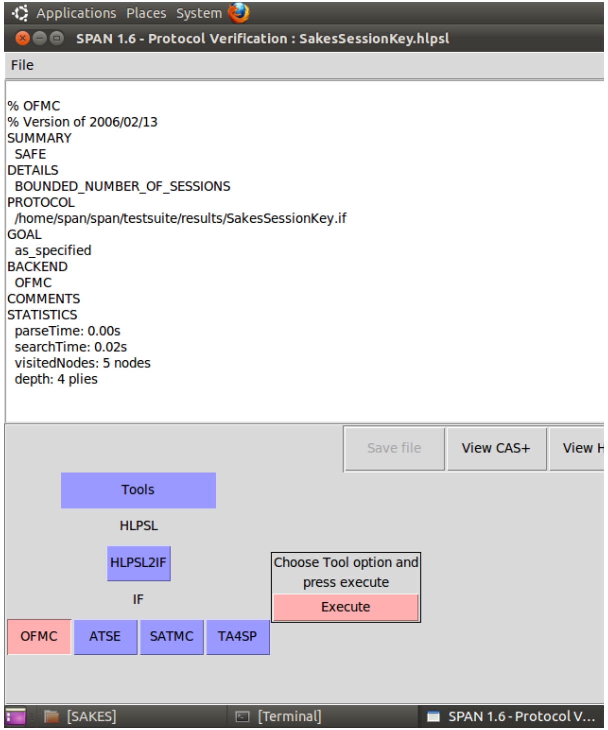


AVISPA OFMC Back-End Simulation Result SAKES Scheme

Authentication Phase

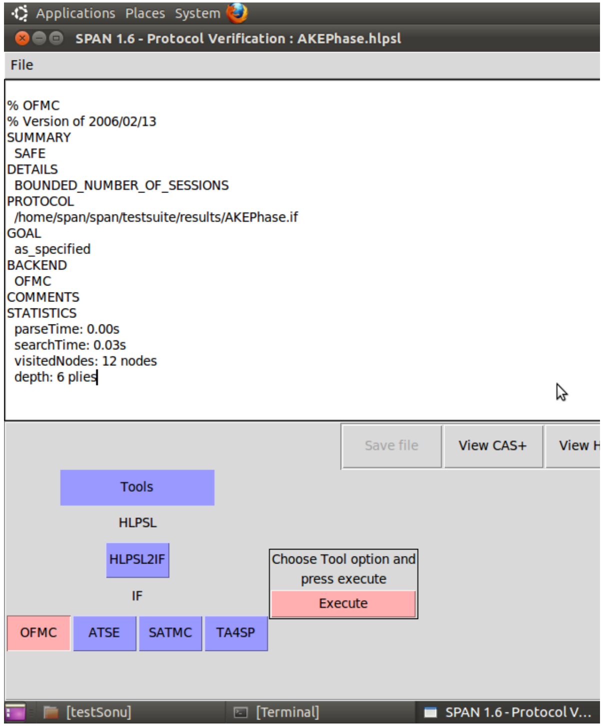


Key Establishment Phase

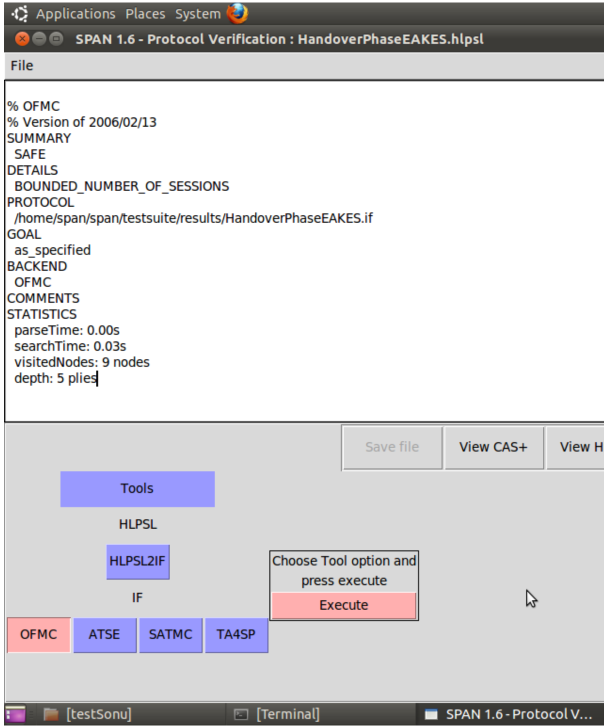


AVISPA OFMC Back-End Simulation Result EAKES6Lo Scheme

AKE Phase

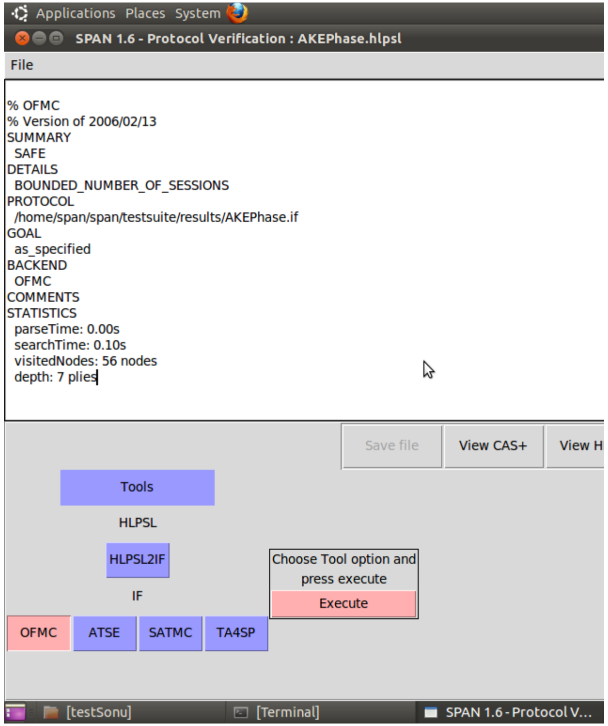


Handover Phase

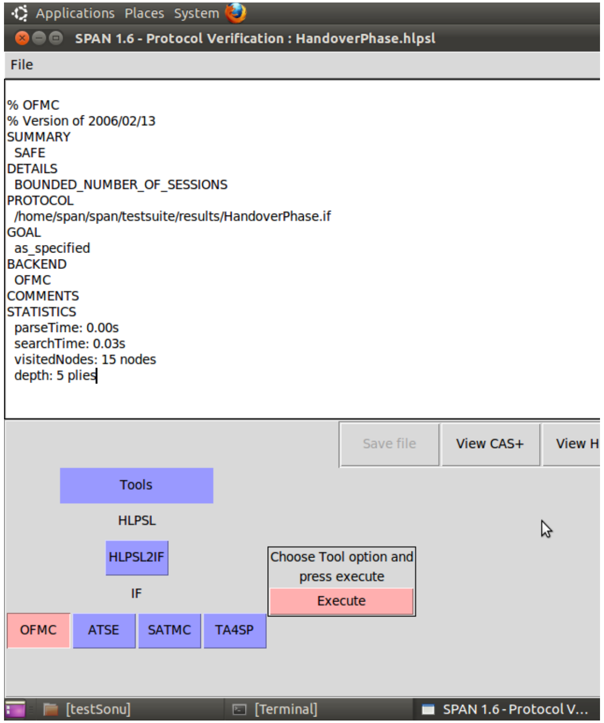


AVISPA OFMC Back-End Simulation Result S6AE Scheme

AKE Phase



Handover Phase



Performance Analysis

	SAKES	EAKES6Lo	S6AE
1. Type of Cryptographic System used for key exchange process.	AES-CTR-128bits, SHA-256, and Diffie-Hellman (DH) key exchange.	AES-CTR-128bits, SHA-256, and ECDH160	SHA-256 and ASCON
2. Storage Overhead (Parameters required to be stored by SN and CS)	Total in SN: 272 bytes Total in CS: 272 bytes	Total in SN: 88 bytes Total in CS: 80 bytes	Total in SN: 50 bytes Total in CS: 58 bytes
3. Computational Overhead	Total time ≈ 58.6044 ms	Total time ≈ 17.2494 ms	Total time ≈ 0.6643 ms

Performance Analysis (cont.)

	SAKES	EAKES6Lo	S6AE
4. Mobility	Only static nodes	Both static and mobile nodes	Both static and mobile nodes
5. Handover Phase (for mobile nodes)	Not Applicable	Time reqd.: 11.9366 ms	Time reqd.: 0.2544 ms
6. Communication Overhead	SN → 6LDR: 688 bits 6LDR → SN: 2176 bits Total: 358 bytes	SN → 6LDR: 672 bits 6LDR → SN: 784 bits Total: 182 bytes	SN → 6LDR: 496 bits 6LDR → SN: 528 bits Total: 128 bytes

Conclusion

6LoWPAN is a providential technology having a vital share in IoT and is commonly deployed in a variety of applications.

In this project, we studied and analyzed the schemes for authentication and key establishment in 6LoWPAN.

From the analysis, it is found that S6AE is more effective than SAKES and EAKES6Lo in terms of computational, storage, and communication overhead.

The schemes are implemented in the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool for formal security verification and are found to be safe.

Future Work

As a future work, the available schemes can be worked on to improve the existing schemes or develop a new solution.

The existing schemes SAKES could be improved to introduce mobile nodes,

EAKES6Lo could be improved to be less computationally expensive and

S6AE could be improved to be resistant against jamming attack.

References

1. Computer Security Resource Center (https://csrc.nist.gov/glossary/term/authentication)
2. ElectronicNotes. What is 6lowpan - the basics (https://www.electronicsnotes.com/articles/connectivity/ieee-802-15-4-wireless/6lowpan.php)
3. 6LoWPAN (https://en.wikipedia.org/wiki/6LoWPAN)
4. Hussen, H.R.; Tizazu, G. T. M. L. T. C. Y. K. K. Sakes: Secure authentication and key establishment scheme for m2m communication in the ip-based wireless sensor network (6lowpan). 2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN), (2013)
5. Qiu, Y.; Ma, M. A mutual authentication and key establishment scheme for M2M communication in 6lowpan networks. IEEE Trans. Ind. Inform, (2016).
6. Muhammad Tanveer, Ghulam Abbas, Z. H. A. M. W. F. M. and Kim, S. S6AE: Securing 6lowpan using authenticated encryption scheme. ncbi.nlm.nih.gov (2020).
7. 6LoWPAN Tutorial – A Wireless Extension of the Internet : Texas Instrument (https://www.youtube.com/watch?v=zzoZNG_NB_c&t=50s)

Thank You :)

Special Thanks to Freepik for the vectors