Est. 1841 | YORK ST JOHN UNIVERSITY

Karthick, Gayathri, Mapp, Glenford, Kammueller, Florian and Aiash, Mahdi (2018) Formalization and Analysis of a Resource Allocation Security Protocol for Secure Service Migration. In: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). IEEE

Downloaded from: https://ray.yorksj.ac.uk/id/eprint/9345/

The version presented here may differ from the published version or version of record. If you intend to cite from the work you are advised to consult the publisher's version: http://dx.doi.org/10.1109/ucc-companion.2018.00058

Research at York St John (RaY) is an institutional repository. It supports the principles of open access by making the research outputs of the University available in digital form. Copyright of the items stored in RaY reside with the authors and/or other copyright owners. Users may access full text items free of charge, and may download a copy for private study or non-commercial research. For further reuse terms, see licence terms governing individual outputs. Institutional Repositories Policy Statement

RaY

Research at the University of York St John

For more information please contact RaY at ray@yorksj.ac.uk

Formalization and analysis of a Resource Allocation Security Protocol for Secure Service Migration

Gayathri Karthick, Glenford Mapp, Florian Kammueller, and Mahdi Aiash
School of Science and Technology
Middlesex University
Email: gk419s@live.mdx.ac.uk, G.Mapp@mdx.ac.uk, F.kammueller@mdx.ac.uk, M.Aiash@mdx.ac.uk

***Abstract*—The advent of virtual machine technology for example, VMware, and container technology, such as Docker, have made the migration of services between different Cloud Systems possible. This enables the development of mobile services that can ensure low latencies between servers and their mobile clients resulting in better Quality of Service. Though there are many mechanisms in place to provide support for mobile services, a key component that is missing is the development of security protocols that allow the safe transfer of servers to different Cloud environments. In this paper, we propose a Resource Allocation Security Protocol for secure service migration. We explore two approaches: in the first approach, the protocol is developed and formally verified by the Automated Validation of Internet Security Protocols and Applications tool. The protocol satisfies the security properties of secrecy and authentication. In addition, nonces are used for replay protection and to ensure freshness. In the second approach, a secure symmetrical session key is used to do the safe transfer and an automatic cryptographic protocol verifier, ProVerif, is employed to verify secrecy, authentication and key exchange.**

***Keywords*—Mobile services, Security protocol, Vehicular Cloud, Avispa and ProVerif.**

I. INTRODUCTION

Cloud computing facilitates the migration of data and services. This gives many incentives for data owners to migrate their data and services to Cloud storage platforms at low cost. In this brave new world, Cloud Providers will actively advertise their Cloud resources to Mobile Service Providers who can use these advertisements to dynamically migrate their servers to these Clouds. In this environment, mobile users will demand to be always connected using heterogeneous networking. Mobile devices will, therefore, have several wireless interfaces including Wi-Fi, LTE, 5G, satellite and Ultra-wideband interfaces. These networks will seamlessly work together using vertical handover techniques [13]. The Y-Comm architecture [12] has been developed to build future mobile systems that can provide seamless communications. Hence, future mobile systems will need to support both mobile users and services. Though extensive work has been done to support mobile users, less work has been done on the development of mobile services. However, there is now increasing interest in this area as there is a need to provide services at the edge of the network, i.e. to support edge-computing. One aspect of the research on mobile services that has been inadequate is support for security. In particular, it is important that servers do not end up being hosted on unsafe Cloud infrastructure which can hamper service delivery to mobile clients and also Clouds do not end up hosting malicious servers which can damage Cloud infrastructure. This paper attempts to address these issues by providing a new security protocol for secure service migration. This security protocol is called the Resource Allocation Security Protocol (RASP) for secure service migration over Cloud infrastructure [15]. The protocol is developed and tested through a simulation study using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool which is used for the analysis of large-scale Internet security protocols and applications. In the second approach, the ProVerif tool, which verifies cryptographic protocols and associated security goals, is employed. Using both techniques, this paper shows that the RASP protocol can be used to securely migrate services to different Cloud environments.The rest of the paper is organised as follows: Section II presents the related work; Section III details the solution approach while Sections IV and V describe our first and second attempts of RASP, evaluation and show the results. Section VI details of the applications of RASP to a Vehicular Testbed. The paper concludes with Section VII.

II. RELATED WORK

Service migration has been proposed for many environments and is increasingly being used in Cloud infrastructure. Y-Comm is an architecture that has been designed to build heterogeneous mobile networks [11]. It offers the most functionality and flexibility in terms of communication, mobility, QoS and security. Lately, a new Service-Oriented Architecture [17] that allows services to be managed, copied or migrated to support mobile users, has been proposed. This framework takes into account recent efforts in the area of seamless connectivity across heterogeneous networks, as well as increasing capabilities of Cloud technology. Edge computing, where servers are placed on Clouds close to the edge rather than at the centre of the Internet [16] to reduce delays, has been growing rapidly. Edge computing also enables support for networks in highly mobile environments such as Vehicular Ad-Hoc Networks (VANETs). Though all these mechanisms are promising developments [2], it is necessary to consider security as a key part of the overall design.

III. SOLUTION APPROACH

In this section, we look at protocols for secure service migration [8]. As stated previously, the main issues are fraudulent Cloud providers that attempt to entice service providers to host their services on faulty Cloud facilities resulting in data loss and lack of service as well as misbehaving services which attempt to convince Cloud providers that they are well-behaved

systems leading to mismanagement and abuse of Cloud facilities. In our new approach [10], we are working on secure service migration between commercial Cloud infrastructures. The RASP protocol has been developed to support mobile services that allow the safe transfers of resources to different Cloud environments. The protocol is broken into four stages to clarify the necessary operations involved in secure migration.

1) **Stage1: Advertisement**:
   Cloud CB actively advertises its resources which is picked up by server SA on Cloud CA.
2) **Stage2: Authentication of SA and CB as well as migration request and response**:
   Server SA first requests the Registry to authenticate Cloud CB and the resources it holds. Once it receives the approval from the Registry, it sends a migration request to Cloud CB. Cloud CB then requests the Registry to authenticate server SA and the resources it requires. Once this is verified, Cloud CB sends a positive migration response to server SA.
3) **Stage3: Migration transfer**:
   Server SA sends a message to Cloud CB to begin the migration transfer. Cloud CB begins the transfer and signals server SA when the transfer is completed.
4) **Stage4: Update of New service location to the Registry**:
   The new service SB is now running on Cloud CB and informs the Registry that it has been successfully migrated.

IV. First Attempt

In the RASP protocol [10], the system is moving server SA on Cloud CA to Cloud CB. There are three entities in the protocol: server SA on Cloud CA, Cloud CB, and the Registry.

A. General Notations

1) Server resources: The server (SA) is represented by: Server on Cloud A (SA) = Server_ID, TOS=Server, PKS, Resources.

2) Cloud Facilities:

1) Cloud A(CA) = Cloud_IDA, TOS = Cloud, PKA, Resources.
2) Cloud B(CB) = Cloud_IDB, TOS = Cloud, PKB, Resources.

Each Cloud is uniquely identified by a Cloud_ID and since they are Cloud services, TOS = Cloud; PKA and PKB are public keys for Cloud CA and Cloud CB respectively. In addition, each Cloud will have a number of resources which it actively advertises to servers.

3) The Registry: The Registry is the last key component and is used to verify the identities of all servers on the network. In addition, the Registry knows the services on different Clouds and has the public keys for each service. We represent the public key for the registry as PKR.

4) Nonces and Timestamps: Nonces (NA and NB) are randomly generated numbers which are unforgeable and are used as session tokens, ensuring that requests cannot be repaid by unauthorised personnel. Timestamps TA, TB & Tcomp are used to measure the time taken for the process. TA is the time when the server makes the migration request to Cloud CB. TB is the time when Cloud CB responds to the server request. Tcomp is the time when transfer is completed.

B. Algorithm 1 RASP Protocol

This is our first attempt to implement the service migration framework using the protocol. In this section, we look at when the service moves from Cloud CA to Cloud CB. In this attempt, Stage 1 in the previous Table 1, corresponds to step 1 of the protocol; Stage 2 corresponds to steps 2-7; Stage 3 corresponds to steps 8 and 9 and finally the Stage 4 corresponds to step 10. The protocol is followed in exactly the same way as outlined below.

- **Stage 1**
  1. CB $\rightarrow$ SA : Advertisement (Cloud_IDB, TOS, Resources, PKB)
- **Stage 2**
  2. SA $\rightarrow$ R : Verify_Identity (Cloud_IDB, TOS, PKB, Resources, Server_ID, PKS) PKR
  3. R $\rightarrow$ SA : Message: YES (Cloud_IDB, TOS, PKB, Valid Resources) PKS
  4. SA $\rightarrow$ CB : Migration_Request (Server_ID, TOS, TA, Req_Resources, PKS, NA) PKB
  5. CB $\rightarrow$ R : Verify_Identity (Server_ID, PKS, TOS, Cloud_IDB, PKB) PKR
  6. R $\rightarrow$ CB : Message: Yes (Server_ID, TOS, Valid Req_Resources(services)) PKB
  7. CB $\rightarrow$ SA : Migration_Response (Cloud_IDB, TOS= Cloud, TB, Resources_Granted, NA, NB) PKS
- **Stage 3**
  8. SA $\rightarrow$ CB : Transfer_Migration (Server_ID, Cloud_IDB, Req_Resources, NB) PKB
  9. CB $\rightarrow$ SA : Transfer_Ack (Server_ID, Cloud_IDB, Tcomp) PKS
- **Stage 4**
  10. SB $\rightarrow$ R : Transfer-Comp (Server_ID, Cloud_IDB, TOS, TA, TB, Tcomp) PKR.

C. The full explanation of this migration is given below

- **Step 1** Server SA on Cloud A receives advertisements from Cloud B advertising its resources.
- **Step 2** Server SA checks the validity of Cloud B.
- **Step 3** The Registry authenticates Cloud B.
- **Step 4** Server SA sends a request to migrate to Cloud B.
- **Step 5** Cloud B sends a request to the Registry, to make sure that the server SA is a valid requested service.
- **Step 6** The Registry validates server SA.
- **Step 7** Cloud B signals to server SA that it is OK to migrate the requested service to Cloud B.
- **Step 8** Server SA signals to Cloud B to migrate the service.

- **Step 9** Cloud B signals to server SA that the migration is complete: The service is now started on Cloud B. Hence, new location: (SA $\rightarrow$ SB)
- **Step 10** The new server (SB) on Cloud B signals to the Registry that the migration has been completed.

Figure 1 shows the steps for Cloud-to-Cloud migration of services.

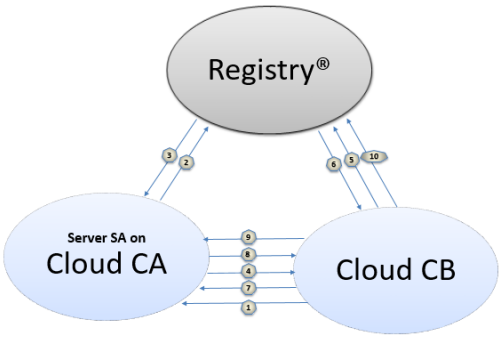

Figure 1: First Attempt Migration between SA to CB

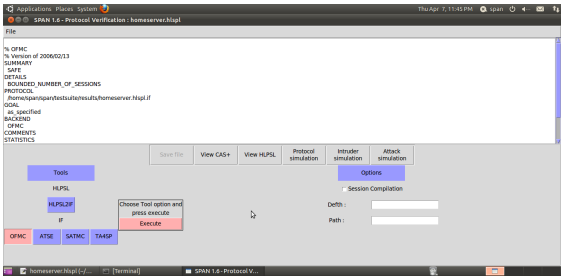

Figure 2: OFMC: Cloud A to Cloud B

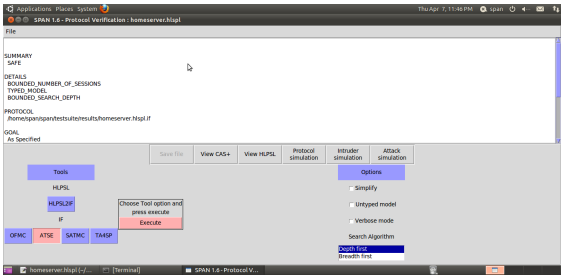

Figure 3: ATSE: Cloud A to Cloud B

D. Evaluation of the First Attempt

There are important observations to be made with this protocol. Firstly, nonces are used to protect the interaction between server SA and Cloud CB to which the server wants to migrate. This is done using an extended approach based on the Needham Schroeder protocol [3]. In step 4, the server uses NA as part of the transfer request. In step 7, Cloud B replies to the server using NA and NB as part of the transfer response. In step 8, server authorises the transfer of the service to Cloud B using NB.

E. AVISPA Results

In our first attempt, AVISPA [4] was used to analyse the protocol specified in the previous section. AVISPA provides automated validation of Internet security protocols and applications. Here, we represent the formal specification and verification which is carried out using the role-based language, HLPSL and the AVISPA model checker which automates the model checking of the protocol and uses the parameters necessary for the formal verification of RASP. The first step was to validate the specification using the OFMC back-end tool of AVISPA, and the second step was to use the ATSE back-end. In our protocol, AVISPA outputs SAFE from OFMC Figure 2 and ATSE Figure 3. Hence both protocols show SAFE [10] so the expected result is accomplished.

Using the AVISPA tool, we are able to show that the protocol specified is safe against imposter attacks in normal operation. However, in order to create a complete security framework for Cloud Systems [9], we also need to look at verification of the cryptographic protocols. ProVerif is used for automatically analysing the security of cryptographic protocols. In our second attempt, we used the ProVerif tool to fully verify the interaction for secure service migration including the security properties such as the services, nonces, private keys, session keys, signatures, encryption and decryption mechanisms. Compared to AVISPA, ProVerif is a more expressive tool. Furthermore, AVISPA is sufficient for abstract protocols but ProVerif may help to model more of the Cloud infrastructure into the protocol thus allowing more extensive properties to be detailed and proved. Hence, ProVerif allows us to better define what is being attacked and hence what needs to be protected. ProVerif also features efficient automatic reasoning tools and is therefore potentially able to verify specific properties such as Cloud resources.

V. SECOND ATTEMPT

In our second attempt, all the stages of the proposed protocol remain the same but implemented in a different way and tested by ProVerif [5]. ProVerif is an extension of Pi-calculus with cryptography. It follows Dolev-Yao models [3] of cryptographic operation. ProVerif is capable of proving reachability properties, correspondence assertions and observational equivalence. It has a large variety of protocol structures (event and the queries, private channels, rewrite rules, etc.) and modelling primitives used by cryptographic protocols (encryption, decryption, digital signatures, etc.). ProVerif′s internal abstraction uses queries to verify the security properties and attempts to prove that there is a state in which the security properties are known to the attacker or not [6]. If the results of the queries are True, then the security property cannot be derived by the attacker. Additionally, it verifies the properties of the security protocol to prove secrecy (strong/weak), authentication and observational equivalence for an unbounded number of sessions using an unbounded message space.

A. General Notations

Table 1 represents Algorithm 2

Notation	Explanation
CA	Cloud A
CB	Cloud B
SA	Server SA on Coud A
SB	New service on Cloud B
Ksc	Symmetric session key
pkC/skC	public and private key pairs of Cloud B
pkS/skS	Public and Private key pairs of server SA
pkR/skR	Public and private key pairs of The registry
Ns	Nonce of SA
Nc	Nonce of CB
Advc	Advertisements
ResSA	Requested Resources of SA
ResCB	Resources of CB
M_Reqc	Request for migration
M_Trfs	Transfer of migration
M_Ackc	Acknowledgement of migration
sign	Signature/signed by the Registry
aenc	Asymmetric Encryption
enc	Symmetric Encryption

Table I: Algorithm 2

B. Algorithm 2: RASP for Migration between server on Cloud A to Cloud B

As we stated earlier, the Algorithm 1 and Algorithm 2 remain the same but in Stage 3, we create a secure session key (Ksc) [1] which is used for sessions between server SA and Cloud CB. In addition, the same key will be used for encryption and decryption. Server SA generates the session key (Ksc), to start the migration request (M_Reqc). By using the session key (Ksc), Cloud CB begins the transfer (M_Trfs) and signals server SA when the transfer is completed (M_Ackc). In this second attempt, Stage 1 corresponds to step 1 of the protocol; Stage 2 corresponds to steps 2-7; Stage 3 corresponds to 8-11, and finally, Stage 4 corresponds to step 12. The RASP protocol is followed in exactly the same way an outlined below and Fig. 4 shows the steps for Cloud-to-Cloud migration of services.

- **Stage 1**
  1. CB $\rightarrow$ SA : Advc (CB, ResCB)
- **Stage 2**
  2. SA $\rightarrow$ R : (SA, CB, ResCB)pkR
  3. R $\rightarrow$ SA : sign((CB, pkC, ResCB), pkS)
  4. SA $\rightarrow$ CB : aenc ((Ns, SA , ResSA), pkC)
  5. CB $\rightarrow$ R : (CB, SA , ResSA)pkR
  6. R $\rightarrow$ CB : sign((SA, pkS , ResSA), pkC)
  7. CB $\rightarrow$ SA : aenc((Ns, Nc, CB), pkS)
- **Stage 3**
  8. SA $\rightarrow$ CB : aenc((Nc, Ksc), pkC)
  9. CB $\rightarrow$ SA : enc ((M_Reqc, SA), Ksc)
  10. SA $\rightarrow$ CB : enc ((M_Trfs, ResSA), Ksc)
  11. CB $\rightarrow$ SA : enc ((M_Ackc), Ksc)
- **Stage 4**
  12. SB $\rightarrow$ R : aenc((SB, CB), pkR)

C. The full explanation of this RASP protocol is given below

- **Step 1** Server SA receives advertisements from Cloud B advertising Cloud B′s resources with its identity CB.
- **Step 2** Server SA checks the validity of Cloud CB with the Registry.
- **Step 3** The Registry authenticates server SA.
- **Step 4** Server SA sends a migration request to Cloud CB.
- **Step 5** Cloud CB sends a request to the Registry to make sure that server SA′s request for resources is valid.
- **Step 6** The Registry replies back to Cloud CB.
- **Step 7** Cloud CB sends the migration response back to server SA.
- **Step 8** Server SA generates the session key(Ksc)to start the migration request.
- **Step 9** Cloud CB sends the migration initialisation request.
- **Step 10** Server SA does the migration transfer.
- **Step 11** Cloud CB sends an acknowledgement to server SA.
- **Step 12** Service on Cloud B (SB) updates the Registry on its new location.

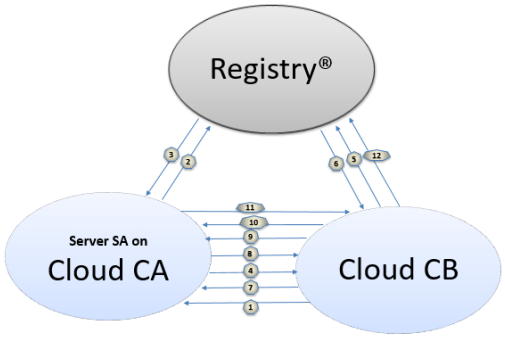

Figure 4: Second Attempt Migration between SA to CB

D. Stages in detail

1) Stage 1: Cloud CB advertises (Advc) its resources which are picked up by server SA on Cloud A. Server SA and Cloud CB do not know anything about each other and there is no prior communication between SA and CB. In the first step, the server SA receives an advertisement from Cloud B advertising its resources (ResCB) with its identity CB.
1. CB $\rightarrow$ SA : Advc (CB, ResCB)

2) Stage 2: Both entities, server SA and Cloud B, must be authenticated to each other. Firstly, on receiving an advertisement from CB, SA checks the validity of Cloud B with the Registry. So it sends its identity SA, Cloud B′s identity, CB, and the resources at Cloud B (ResCB) by using the public key of Registry.The Registry will be able to verify the validity of any Cloud resources. The Registry replies back to server SA, saying that Cloud CB is a valid Cloud and encrypts the response with the public key(pkS) of SA and signs the response with its private key. Therefore, this digital signature

ensures that the Registry is the originator of the message. SA generates and sends a fresh nonce (Ns), its identity SA, requested resources (ResSA) and encrypts it using the public key of Cloud B, (pkC). Nonces are used to protect this session between the server SA and the Cloud B to which the server wants to migrate [7]. When CB receives this message, it decrypts the message using its private key (skC). Cloud CB verifies the request made by SA by forwarding the requested resources of SA (ResSA), the identity of SA and its identity, CB, to the Registry by using the public key (pkR) of Registry. The Registry replies back to Cloud CB, it validates server SA and encrypts the response with Cloud CB′s public key and signs it with its private key. Finally, Cloud B sends a message to server SA that it can migrate to Cloud B, it acknowledges Ns and generates a new nonce, Nc.

2. SA → R : (SA, CB, ResCB)pkR

3. R → SA : sign((CB, pkC, ResCB), pkS)

4. SA → CB : aenc ((Ns, SA , ResSA), pkC)

5. CB → R : (CB, SA , ResSA)pkR

6. R → CB : sign((SA, pkS , ResSA), pkC)

7. CB → SA : aenc((Ns, Nc, CB), pkS)

3) Stage 3: In Stage 3, server SA generates a fresh symmetric session key (Ksc), pairs it with the nonce generated by CB (Nc), encrypts the message by using Cloud Bs public key (pkC) and sends to CB. Cloud B initiates the migration request (M_Reqc), pairs it with server SA, and encrypts it using the generated symmetric session key (Ksc). Once the migration request from CB to SA is received, SA is able to decrypt it using the shared key (Ksc) and starts the migration transfer (M_Trfs) with the requested resource (ResSA) by using the symmetrical session key (Ksc). Once the migration transfer is completed, CB sends an acknowledgement (M_Ackc), to server SA which is encrypted using Ksc.

8. SA → CB : aenc((Nc, Ksc), pkC)

9. CB → SA : enc ((M_Reqc, SA), Ksc)

10. SA → CB : enc ((M_Trfs, ResSA), Ksc)

11. CB → SA : enc ((M_Ackc), Ksc)

4) Stage 4: In Stage 4, the service (SB) updates the Registry on its new location and hence SA → SB.

12. SB → R : aenc((SB, CB), pkR)

E. Evaluation of the Second Attempt

In our second attempt, by using symmetric session key (Ksc), the requested service is transferred to the new location CB. As we mentioned in the first attempt, nonces are used to protect the session between server SA and Cloud CB. However, in the second attempt, the symmetrical session key is used to do the actual transfer. Hence, this is a more secure mechanism than using nonces for the actual data transfer.

F. ProVerif Results

The results show that RASP can preserve the secrecy, authentication and key exchange of the service migration mechanism. We verified Stage 2 and Stage 3 using the ProVerif tool and present the output of the program which is given in Figure 5 below. The security properties were specified in the input language to check whether it is derivable by the attacker or not.

Results in Terminal:proverif/opam/system/bin/proverif1.98pl1 docs/Nee/CB toSA_SecuirtyProtocol.pvgrep ”RES”.

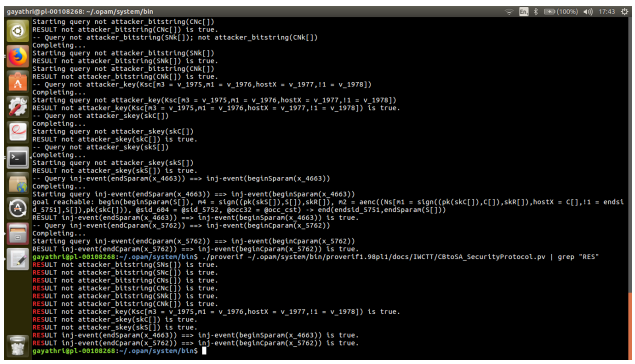

Figure 5: Second Attempt results using Proverif

1) Nonces are secured and not derived by the attacker:

- RESULT not attacker_bitstring (SNs []) is true.
- RESULT not attacker_bitstring (SNc []) is true.
- RESULT not attacker_bitstring (CNs []) is true.
- RESULT not attacker_bitstring (CNc []) is true.

2) The session key is not derived by the attacker:

- RESULT not attacker_key (Ksc [m3 = v_1975, m1 = v_1976, hostX = v_1977! 1 = v_1978]) is true.
- RESULT not attacker_bitstring (SNk []) is true.
- RESULT not attacker_bitstring (CNk []) is true.

3) Private keys of SA & CB is not derived by the attacker:

- RESULT not attacker_skey (skC []) is true.
- RESULT not attacker_skey (skS []) is true.

4) Authentication SA to CB and CB to SA is true:

- RESULT inj-event (endSparam (x_4663)) → inj-event (beginSparam (x_4663)) is true.
- RESULT inj-event (endCparam (x_5762))→ inj-event (beginCparam (x_5762)) is true.

VI. APPLICATION TO VEHICULAR TESTBED

The rapid growth in the number of vehicles on the roads has created a plethora of challenges for road traffic management authorities such as traffic congestion, increasing number of accidents, air pollution, etc. Over the last decade, significant research efforts from both the automotive industry and academia have been undertaken to accelerate the deployment of the Wireless Access in Vehicular Environments (WAVE) standard based on a Dedicated Short Range Communication (DSRC) among moving vehicles (Vehicle-to-Vehicle, V2V) and roadside infrastructure (Vehicle-to-Infrastructure, V2I) [14]. This

network is called a VANET network and is characterised by high node speed, rapidly changing topologies, and short connection lifetimes. VANETs are realised by the deployment of Roadside Units (RSUs) located along the transport infrastructure and Onboard Units (OBUs) in the vehicles or worn by pedestrians or cyclists. Based on the protocol presented, which was verified using Proverif, VANET systems would be the best option on which to implement the Resource Allocation Security Protocol. Hence, the protocol will be tested on a VANET testbed developed by Middlesex University on its Hendon Campus.

VII. CONCLUSIONS AND FUTURE WORK

This paper has presented a new Resource Allocation Security Protocol (RASP) for server migration between commercial Cloud environments. In our first attempt, using AVISPA tool, we showed that the protocol is safe under normal operation; the present protocol critically prevents impersonation attacks either by rogue Cloud infrastructure hoping to sneer valid services or by malicious servers wanting to inflict damage on Cloud Infrastructure. In our second attempt, using ProVerif, we showed that the proposed protocol succeeds in three significant security properties namely: secrecy, authentication of SA to CB and CB to SA, and secure symmetric key exchange. More work is being done to analyse intruder attacks to show that the proposed protocol is secure against active and passive attacks. We are also exploring how this protocol can be enhanced to prevent man-in-the-middle attacks.

REFERENCES

[1] M. Aiash, G. Mapp, A. Lasebae, J. Loo, and R. Phan, "A Formally Verified AKA Protocol For Vertical Handover in Heterogeneous Environments using Casper/FDR," *EURASIP Journal on Wireless Communcations and Networking (Open Springer)*, April 2012.

[2] M. Aiash, G. Mapp, A. Lasebae, R. Phan, and J. Loo, "Integrating Mobility, Quality-of-Service and Security in Future Mobile Networks," *Electrical Engineering and Intelligent Systems: Lecture Notes in Electical Engineering*, vol. 130, pp. 195–206, 2013.

[3] R. M. Amadio and W. Charatonik, "On name generation and set-based analysis in the dolev-yao model," in *International Conference on Concurrency Theory*. Springer, 2002, pp. 499–514.

[4] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani *et al.*, "The avispa tool for the automated validation of internet security protocols and applications," in *International conference on computer aided verification*. Springer, 2005, pp. 281–285.

[5] B. Blanchet *et al.*, "Modeling and verifying security protocols with the applied pi calculus and proverif," *Foundations and Trends® in Privacy and Security*, vol. 1, no. 1-2, pp. 1–135, 2016.

[6] T. Chen, F. Kammüller, I. Nemli, and C. W. Probst, "A probabilistic analysis framework for malicious insider threats," in *International Conference on Human Aspects of Information Security, Privacy, and Trust*. Springer, 2015, pp. 178–189.

[7] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018.

[8] Q. Duan, Y. Yan, and A. V. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Transactions on Network and Service Management*, vol. 9, no. 4, pp. 373–392, 2012.

[9] K. Kammueller, G. Mapp, S. Patel, and S. Abubaker, "Engineering Security Protocols with Model Checking - Radius-SHA256 and Secured Simple Protocol," in *Proceedings of the 7th International Conference on Internet Monitoring and Protection*, 2012.

[10] G. Karthick, G. E. Mapp, F. Kammueller, and M. Aiash, "Exploring a security protocol for secure service migration in commercial cloud environments," 2017.

[11] G. Mapp, M. Aiash, A. Lasebae, and R. Phan, "Security Models for Heterogeneous Networking," in *International Conference on Security and Cryptography (SECRYPT)*, July 2010.

[12] G. Mapp, F. Shaikh, M. Aiash, R. Vanni, M. Augusto, and E. Moreira, "Exploring Efficient Imperative Handover Mechanisms for Heterogeneous Networks," in *Proceedings of the International Symposium of Emerging Ubiquitous and Persuasive Systems, Indianapolis, USA*, August 2009.

[13] G. Mapp, F. Shaikh, M. Aiash, R. P. Vanni, M. Augusto, and E. Moreira, "Exploring efficient imperative handover mechanisms for heterogeneous wireless networks," in *2009 International Conference on Network-Based Information Systems*. IEEE, 2009, pp. 286–291.

[14] V. V. Paranthaman, A. Ghosh, G. Mapp, V. Iniovosa, P. Shah, H. X. Nguyen, O. Gemikonakli, and S. Rahman, "Building a prototype vanet testbed to explore communication dynamics in highly mobile environments," in *International Conference on Testbeds and Research Infrastructures*. Springer, 2016, pp. 81–90.

[15] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[16] F. Sardis, G. Mapp, J. Loo, and M. Aiash, "Dynamic edge-caching for mobile users: Minimising inter-as traffic by moving cloud services and vms," in *Proceedings of the Conference on Advanced Information Networking and Applications, AINA 2014, Victoria, Canada*, May 2014.

[17] M. Sardis, G. E. Mapp, J. Loo, M. Aiash, and A. Vinel, "On the Investigation of Cloud-based Mobile Media Environments with Service-Populating and QoS-aware Mechanisms," *IEEE Transactions on Multimedia*, January 2013.