SEMOS: A Middleware for Providing Secure and Mobility-aware Sessions over a P2P Overlay Network

Daouda Ahmat, Mahamat Barka, Damien Magoni

SEMOS: A Middleware for Providing Secure and Mobility-aware Sessions over a P2P Overlay Network

Daouda Ahmat[1,2], Mahamat Barka[2], and Damien Magoni[3]

[1] Virtual University of Chad, Chad
daouda.ahmat@gmail.com
[2] University of N'Djamena, Chad
mahamat.barka@gmail.com
[3] University of Bordeaux – LaBRI, France
magoni@labri.fr

Abstract. Mobility and security are major features for both current and future network infrastructures. Nevertheless, the integration of mobility in traditional virtual private networks is difficult due to the costs of re-establishing broken secure tunnels and restarting broken connections. Besides session recovery costs, renegotiation steps also present inherent vulnerabilities. In order to address these issues, we propose a new distributed mobile VPN system called SEcured MObile Session (SEMOS). Based upon our CLOAK peer-to-peer overlay architecture, SEMOS provides security services to the application layer connections of mobile users. Secure and resilient sessions allow user connections to survive network failures as opposed to regular transport layer secured connections used by traditional VPN protocols.

Key words: connectivity, mobility, overlay, P2P, VPN, security.

1 Introduction

Mobile devices and wireless networks have progressively provided increased connectivity for users. However, such extended connectivity often comes at the expense of vulnerabilities to attacks such as eavesdropping. Malicious users can infiltrate public open networks and attack legitimate traffic. Virtual Private Networks (VPN) are offering high security to the network traffic [1]. Traditionally, these systems allow the user to securely and remotely communicate with its Intranet through insecure public networks such as the Internet. Security services provided by these infrastructures are robust against malicious users attacks. However, traditional VPNs fail to support users' mobility. Indeed, network failures automatically break-up secure tunnels and involve a subsequent renegotiation that is then necessary to re-establish broken tunnels. This renegotiation requires expensive computational operations in order to restore tunnels as well as transport-layer and application-layer connections. This phase of negotiation causes not only significant latency but also presents risks of Man-In-The-

Middle (MITM) attacks. Traditional VPNs can not therefore effectively operate in dynamic and mobile environments. In addition, lack of means, disrupted-environments and unsteady networks, due to both poor-quality devices and technical skills deficiency, are pervasive in developing countries, particularly in Africa. In these areas, secured resilient sessions are needed to overcome both security and performance issues inherent to connection disruptions.

In this paper, we propose a new mobile VPN-like system called SEcured MObile Session (SEMOS). This system is designed to support both the mobility and the security of SEMOS entities. We have implemented a prototype and have tested and evaluated it by running experiments upon an emulated mobile network. We have compared its features and performances to three other existing solutions namely HIP, N2N and MOBIKE that propose open source implementations. Results show that our solution provides more features and exhibits better connection resumption performances.

This paper is an extended version of our previous work presented at Wireless Days 2012 in Dublin [2]. This new version includes an exhaustive state of the art, a more detailed description of the SEMOS underlying P2P overlay network called CLOAK and a comparative evaluation of three existing similar and open source solutions (i.e., HIP, MOBIKE and N2N), with respect to features and latency performances. In addition, we have extended MUSeS properties by adding several useful design features. In order to avoid name conflicts, we have also changed the name of our solution from MUSeS in [2] to SEMOS in this paper.

2 Related work

Up to now, several mobile VPN solutions have been proposed in various research papers. In this section, we study some interesting existing solutions that address the design of mobile VPNs. Mobile VPN systems based on both IPsec [3] and Mobile IP [4] have been proposed several times such as in [5], [6], [7] and [8], in order to attempt to overcome the inherent mobility drawbacks of traditional VPNs.

Nevertheless, as explained in [9], many problems arise from the combination between IPsec and MobileIP. In order to overcome these problems, a model has been proposed which is based on the use of two HAs (internal HA and external HA) and two FAs (internal FA and external FA) by Vaarala *et al.* in [10]. However, this model imposes the use of three imbricated tunnels({x-MIP{GW{i-MIP{original packet}}}}).

In addition, an IPsec-based mobile VPN requires n tunnels (n security layers) when there are n IPsec hops between the source and destination entities. Therefore, the imbricated tunnels in such VPN systems have a negative impact on their network performances (i.e., throughput, overhead, etc).

In order to address IPsec mobility inherent issues, several improved schemes based on the IPsec architecture have been proposed by Eronen *et al.* in [11], [12], or [13]. Based on security extensions to MOBIKE [11], the solution described in [13] combines secure connectivity and Mobile IPv4. This approach resolves

considerably the issues notified in [9] such as overhead, NAT traversal or mobility problems due to the combination of IPsec and Mobile IPv4. These solutions are however not free of scalability issues and network overhead that they inherit from IPsec and Mobile IP.

The dynamic VPN approach proposed in [14] enables to use alternately IPsec in Full-Mesh mode or in Hub mode with a centralized IPsec Gateway. The first mode is only used when routing problems occur. This architecture extends MOBIKE in order to support dynamic tunnels. However, this model is not designed to support mobility.

Another proposal leveraging MOBIKE is presented by Migault in [15], where they propose an alternative End-to-End security (E2E) architecture based on their own MOBIKEX protocol, which extends the MOBIKE mobility and multihoming features to multiple interfaces and to the transport mode of IPsec. Based on a topology organized in communities, peer-to-peer (P2P) mobile VPN systems have also been proposed such as ELA [16] or N2N [17].

In opposition to most dynamic VPN systems, these systems have the advantage to be fairly scalable and to have ability to communicate across NAT and firewalls. Decentralized P2P VPN are flexible and self-organizing infrastructures that enable users to create their own secure networks upon an untrusted network. N2N and ELA topologies are very similar despite the fact that N2N is based on the OSI layer two whereas ELA is based on the OSI layer three. However the use of super nodes in N2N limits its full scalability as these nodes have a more important role than the other nodes and thus they can weaken the overall strength of the N2N network and may even break its connectivity if they fail. Freelan [18] is a multi-platform and open-source peer-to-peer VPN that abstracts a LAN over the Internet.

The Host Identity Protocol (HIP) [19, 20] is an architecture that provides both mobility and multihoming services. HIP introduces a new name space that enables the separation between the host identity, called Host Identity Tag or HIT, and the host location.

3 System description

3.1 Design

As shown in Figure 1, SEMOS is a communication layer directly used by the applications. SEMOS is using another communication layer called CLOAK which enables the creation and management of a P2P overlay network above any IP network [21]. This system is designed to provide traffic security and session continuity. Applications produce data and send it to the SEMOS middleware. Data is then sliced in packets that are encrypted and authenticated. Unlike IPsec or TLS which rely on IP addresses to define the identity of the communication entities, SEMOS relies on permanent device identifiers called *names* provided by CLOAK. SEMOS can keep a user session active despite a connection disruption and can resume an interrupted secure session.

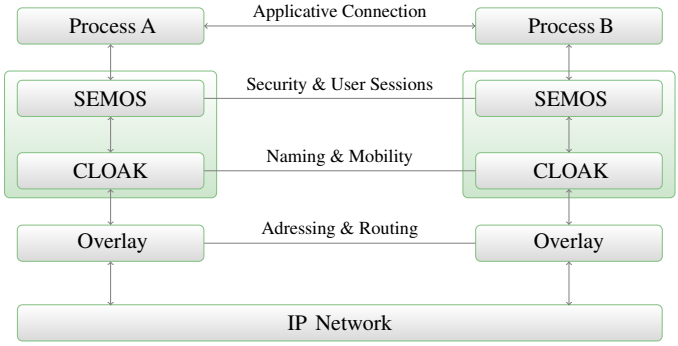

Fig. 1. Architecture overview.

3.2 Overlay

CLOAK is an architecture for building P2P overlay networks which are autonomous and dynamic. A new peer can join any peer already inside the overlay usually by setting up a transport layer connection such as TCP with this peer. The new peer is then offered an overlay address which is unique and dependent on its location in the overlay. These virtual addresses or coordinates are taken from hyperbolic planeand used by a greedy routing algorithm for forwarding the data inside the overlay.

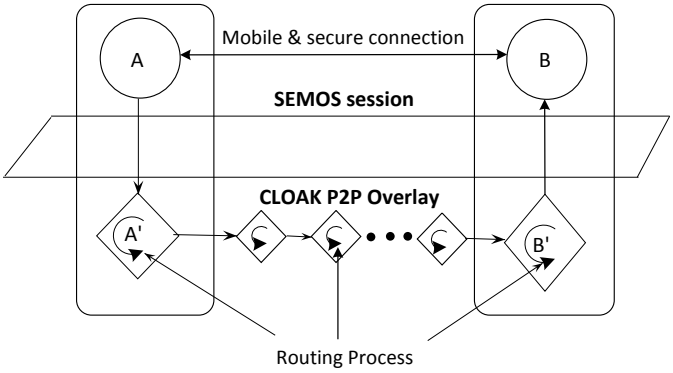

Fig. 2. Routing through the P2P overlay network.

CLOAK was originally defined in our paper [21], which presented the protocols and modules of the architecture with details and reported simulation results upon static and dynamic networks concerning routing success ratio, path length and stretch, as well as DHT performances. The DHT scheme defined by CLOAK

is fully explained in our paper [22]. The Figure 2 shows how a packet is routed over CLOAK.

3.3 Permanent identifiers details

The CLOAK system provides a lookup service based upon distributed tables that consist of pairs defined as $< \mathrm{ID}_{\mathrm{CLOAK}}, @_{\mathrm{overlay}} >$.

Figure 3 describes the mobility mechanism provided by CLOAK. When a CLOAK node joins the network, it stores the pair $< \mathrm{ID}_{\mathrm{CLOAK}}, @_{\mathrm{overlay}} >$ into the DHT, where $\mathrm{ID}_{\mathrm{CLOAK}}$ is its permanent identifier and $@_{\mathrm{overlay}}$ is its virtual temporary address. Thus, after moving within the network or leaving the network, when the node reconnects to the network, it will update the pair $< \mathrm{ID}_{\mathrm{CLOAK}}, @_{\mathrm{overlay}} >$ by replacing its old address $@_{\mathrm{overlay}}$ by its new address that determines its current location. Only $\mathrm{ID}_{\mathrm{CLOAK}}$, which is a permanent identifier, is used to identify the endpoints of a session in order to keep sessions active despite connection disruptions.

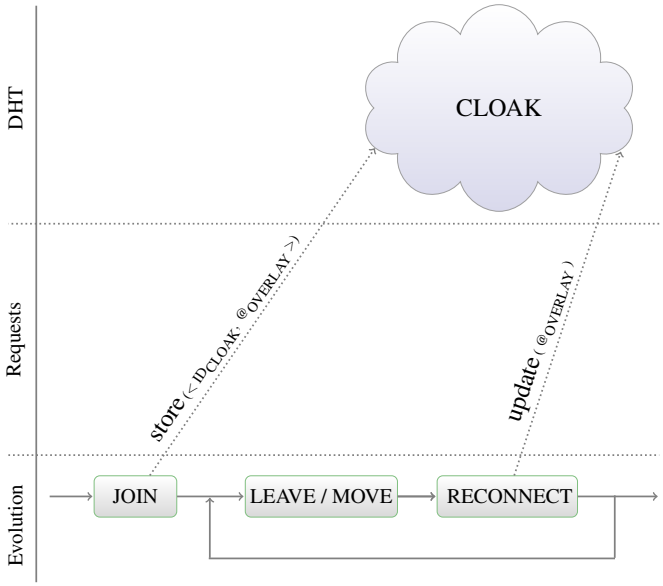

Fig. 3. Mobility scheme based upon the CLOAK DHT.

3.4 Architecture

As shown in figure 4, the SEMOS system is based on four main components:

– The Session Security Module (SSM) manages security services such as confidentiality, integrity and protection against replay attacks;

– The Session Reliability Module (SRM) is designed to provide fault-tolerant secure user sessions; In a nutshell, SRM offers user session continuity service that survives despite connection disruptions caused by lower layers failures;
– The Local Connection Manager (LCM) handles multiple local TCP connections in order to support several simultaneous user applications;
– The Port Forwarder (PF) enables to forward destination ports into local ports in order to intercept and process traffic and finally to send it.

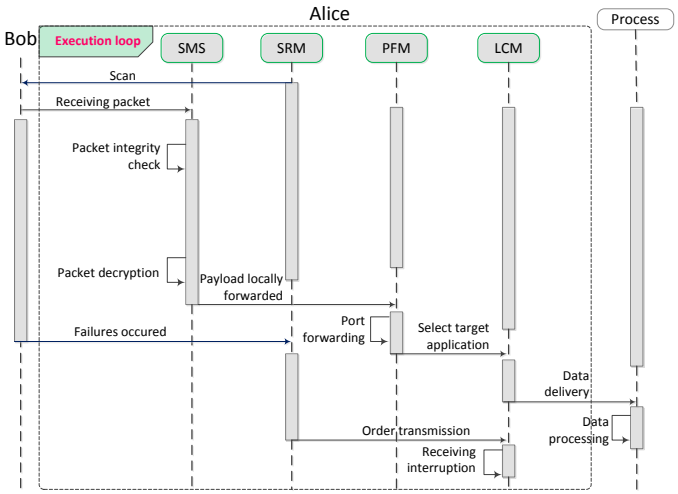

Fig. 4. Interaction between SEMOS components when receiving data.

Interruption detection: When a connection fails, SRM has the ability to detect and handle it (see Figure 5). The destination's unreachability detection is made possible by an acknowledgment mechanism that is based on the exchange of *request-reply* messages.

When there is a packet loss, detected by a lack of acknowledgment or a wait timeout, the communication will be temporarily suspended. Each peer concerned by an interrupted session attempts to reestablish the connection with the corresponding peer. It sends *Re-Hello* packets (see Figure 7 that describes Re-Hello packet structure) at regular time intervals to attempt to restart the disrupted session.

Session Identifier: When a mobile SEMOS entity wishes to communicate with a remote corresponding node, these two peers compute together, in the *handshake* phase, a unique value in order to singly identify each session following the same principle as with a socket connection. When *Alice* sends a *Hello* message (see Figure 6 that describes the Hello packet structure) to *Bob* to start the communication (*handshake phase*), *Alice* generates a partial session ID (*pSessID-a*) and sends it to *Bob*. After receiving *pSessID-a*, *Bob* also generates a partial

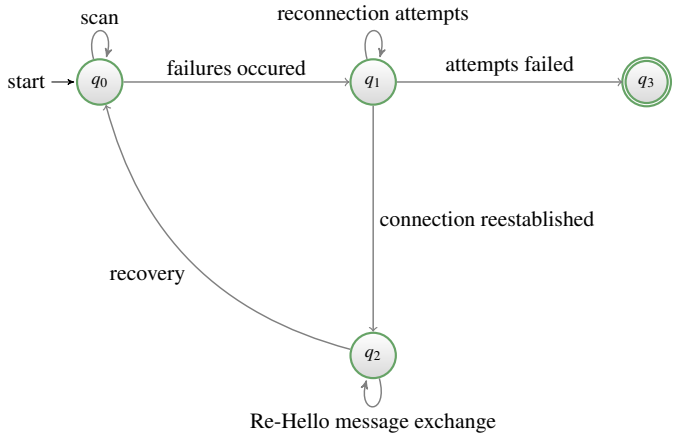

Fig. 5. Failure detection and recovery mechanism.

Hello Packet
- Header
  - Routing Header
    - Src ID = $\mathrm{ID}_{\mathrm{CLOAK}}$Src
    - Dst ID = $\mathrm{ID}_{\mathrm{CLOAK}}$Dst
  - Additional Header Fields
    - Packet Type = flag 0
    - RC P = Source port
    - DST P = Destination port
    - IsConnectionOK() = Yes / No
- Payload % partial session ID
  - PSess ID = uuid() $\odot$ SRC Port

Fig. 6. Hello packet structure.

Re-Hello Packet
- Header
  - Routing Header
    - Src ID = $\mathrm{ID}_{\mathrm{CLOAK}}$Src
    - Dst ID = $\mathrm{ID}_{\mathrm{CLOAK}}$Dst
  - Additional Header Fields
    - Packet Type = flag 2
    - IsResumptionOK()=T/F
- Payload
  - Sess ID = SEMOS ID
  - SEQ NBR = Last packet ID

Fig. 7. Re-Hello Packet Structure

session ID (*pSessID-b*) and sends it to *Alice*. Each corresponding user determines the final session ID (SessID) by concatenating the partial session IDs: SessID = pSessID − a $\odot$ pSessID − b, where $\odot$ represents a concatenation according to lexicographic order.

The communication, identified by a SessID, can be therefore started between the two actors. Precisely, a partial session ID (pSessID) is essentially computed by a combination between a source port number (psN) and a peer ID (pID): pSessID = psN $\odot$ pID.

Secure Mobility: In our model, the concept of secure mobility is a way to allow mobile users to keep an established secure session active when the underlying system connection fails until a subsequent new connection is started. In other words, when a user changes its location as shown in Figure 8 or when its connections are disrupted by network failures, the user session survives as long as necessary to resume the communication.

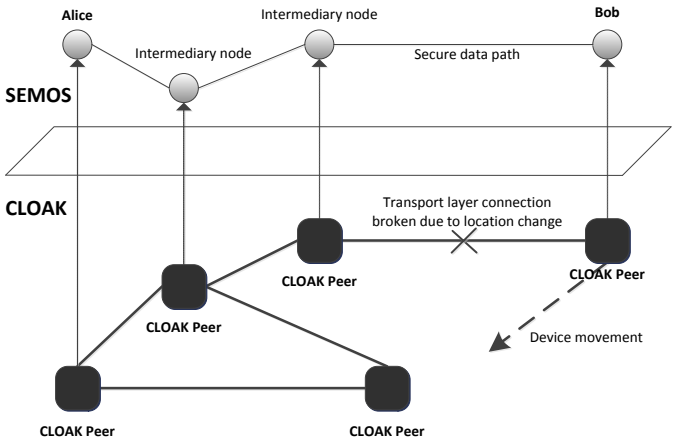

Fig. 8. User mobility with ongoing secure session.

4 Experiment

In order to assess those four VPN technologies in a mobility scenario, we have used a tool called Network Emulator For Mobile Universes (NEmu), developed by Vincent Autefage [23]. Furthermore, NEmu is an emulator that enables the creation of virtualized roaming or mobile devices which is needed in our scenario. The virtual machines run on top of QEMU using the Debian OS. QEMU virtual hosts are interconnected by virtual switches emulated by the Virtual Network Device (VND) software. In order to emulate roaming or mobile devices, NEmu uses the Network Mobilizer (*nemo*) software module.

Table 1. Experiment results.

System	Location change	Average latency	Maximum latency
HIP	20 s	16 s	21 s
MOBIKE	20 s	13 s	19 s
N2N	20 s	49 s	92 s
SEMOS	20 s	4 s	11 s

We have used a minimal FTP-like application based on the OpenBSD version of *nc* in order to experiment on the four different mobile VPN solutions presented. Table 1 provides the average results of ten experiments for each of the four systems. Location change duration showed in the second column is the same for all systems, it is defined as a parameter in NEmu. The third column shows the average latency for each system over ten rounds of experiments. These results show that SEMOS has the smallest average and maximum values for the latency due to communication resumption. Besides, failures do not cause any packet loss despite the long duration of disruption.

5 Conclusion

In this paper, we have presented a new solution called SEMOS for the simultaneous security and mobility of end-to-end connections. SEMOS is based on our P2P overlay system called CLOAK which provides dynamic naming, addressing and routing services. SEMOS builds upon CLOAK to provide secure and resilient sessions to the applications run by mobile users. In other words, user sessions can be paused for any period of time and can be restarted without the cost of renegotiating security parameters. In addition, SEMOS makes disruptions at lower layers completely transparent to the user applications and to their remote corresponding peers. The secured mobility and roaming of SEMOS user communications is thus ensured. The SEMOS middleware is designed to support several user applications simultaneously.

We have implemented a prototype of SEMOS as a middleware inserted between user applications and our CLOAK P2P overlay middleware. Currently, applications using the SEMOS API to implement secure and mobile sessions, must be recompiled. Devices running these applications must also run the CLOAK middleware and belong to a CLOAK overlay. SEMOS is implemented in C and can be downloaded at [24].

We have compared its features to three existing similar solutions: HIP, MOBIKE and N2N. We have also evaluated the performances of SEMOS and the corresponding implementations of the other solutions in an emulated dynamic network environment by using NEmu. We have shown that HIP is the closest solution for providing all the functionalities offered by SEMOS. We have also observed that the latency resumption of SEMOS was the smallest of the four evaluated implementations, which places it as an interesting alternative to existing solutions.

Our future work will consist in completing and improving our current implementation as well as integrating a key management scheme suitable for dynamic networks to provide the users with a means to exchange or generate secret keys in-band. To this end, we have designed a key generation technique based on the Shamir's secret sharing method and using multiple disjoint paths in the overlay, for creating a secret key without requiring certificates such as in Diffie-Hellman [25]. We are currently evaluating this technique and we plan to insert it in the negotiation phase of a SEMOS connection.

References

1. T. Berger. Analysis of current vpn technologies. In *The First Int'l Conf. on Availability, Reliability and Security, ARES 2006.*, page 8 pp., 2006.
2. Daouda Ahmat and Damien Magoni. Muses: Mobile user secured session. In *5th IFIP Wireless Days Int'l Conf.*, Dublin, Irland.
3. S. Kent and K. Seo. Security Architecture for the Internet Protocol. IETF RFC 4301, 2005.
4. C. Perkins. IP Mobility Support for IPv4. IETF RFC 3344, 2002.

5. Jim Binkley. An Integrated IPsec and Mobile-IP for FreeBSD. *Technical Report*, pages 01–10, 2001.
6. T. Braun and M. Danzeisen. Secure mobile IP communication. *26th IEEE Conf. on Local Computer Networks*, pages 586–593, 2001.
7. Heesook Choi, Hui Song, Guohong Cao, and T. La Porta. Mobile multi-layered ipsec. *24th Joint Conf. of the IEEE Computer and Communications Societies*, pages 1929–1939, 2005.
8. R. Ruppelt, A. Pelinescu, C. Constantin, J. Floroiu, D. Sisalem, and B. Butscher. Building ALL-IP Based Virtual Private Networks in Mobile Environment. *Int. Work. on Informatic and Mobile communication over wireless LAN: Research and applications*, 2001.
9. F. Adrangi and H. Levkowetz. Problem Statement: Mobile IPv4 Traversal of Virtual Private Network Gateways. IETF RFC 4093, 2005.
10. S. Vaarala and E. Klovning. Mobile IPv4 Traversal across IPsec-Based VPN Gateways. IETF RFC 5265, 2008.
11. P. Eronen. IKEv2 Mobility and Multihoming Protocol (MOBIKE). IETF RFC 4555, 2006.
12. V. Devarapalli and P. Eronen. Secure Connectivity and Mobility Using Mobile IPv4 and IKEv2 Mobility and Multihoming (MOBIKE). IETF RFC 5266, 2008.
13. M.M. Karbasioun, M. Berenjkub, and B. Taji. Securing mobile IP communications using MOBIKE protocol. *Int'l Conf. on Telecommunications*, 2008.
14. K. Ishimura, T. Tamura, S. Mizuno, H. Sato, and T. Motono. Dynamic IP-VPN architecture with secure IPsec tunnels. *Symp. on Information and Telecommunication Technologies*, 2010.
15. D. Migault, D. Palomares, E. Herbert, Wei You, G. Ganne, G. Arfaoui, and M. Laurent. E2e: An optimized ipsec architecture for secure and fast offload. In *Int'l Conf. on Availability, Reliability and Security*, pages 365–374, 2012.
16. S. Aoyagi, M. Takizawa, M. Saito, H. Aida, and H. Tokuda. ELA: A Fully Distributed VPN System over Peer-to-Peer Network. *Symp. on Applications and the Internet*, pages 89–92, 2005.
17. L. Deri and R. Andrews. N2N: A Layer Two Peer-to-Peer VPN. *Int'l Conf. on Autonomous Infrastructure, Management and Security*, pages 53–64, 2008.
18. J. Kauffmann. The freelan secure channel protocol. `https://github.com/ereOn/libfscp/blob/1.0/fscp.txt`, 2011.
19. R. Moskowitz and P. Nikander. Host identity protocol (hip) architecture. IETF RFC 4423, 2006.
20. Andrei Gurtov. *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. Wiley, 2008.
21. T. Tiendrebeogo, D. Ahmat, D. Magoni, and O. Sié. Virtual connections in p2p overlays with dht-based name to address resolution. *Int'l Journal on Advances in Internet Technology*, 5(1):11–25, 2012.
22. T. Tiendrebeogo, D. Ahmat, and D. Magoni. Reliable and scalable distributed hash tables harnessing hyperbolic coordinates. In *5th IFIP Int'l Conf. on New Technologies, Mobility and Security*, 2012.
23. V. Autefage and D. Magoni. Network emulator: a network virtualization testbed for overlay experimentations. In *17th IEEE Int'l Work. on Computer-Aided Modeling Analysis and Design of Communication Links and Networks*, pages 38–42, 2012.
24. Daouda Ahmat. SEcure MObile Session. `http://www.labri.fr/~magoni/cape/`.
25. D. Ahmat, D. Magoni, and T. Bissyandé. End-to-End Key Exchange Through Disjoint Paths in P2P Networks. In *ICST Trans. on Security and Safety*, volume 2-3, pages 1–15, 2015.