Authentication Protocol Runtime Evaluation in Distributed AAA Framework for Mobile Ad-Hoc Networks

Sondes LARAFA, and Maryline LAURENT

***Abstract*—**Access control AAA infrastructures are traditionally used by the service providers so as to charge their subscribers. Given the easiness and the cheapness of MANET deployment and provided that charging is possible, service providers are likely to offer their services over MANET. In previous works [1] and [2], we presented a distributed AAA framework for MANET. We propose to evaluate the runtime of this framework authentication protocol by modeling and simulating typical cases that are fairly representative of the reality and can easily be extended.

I. Introduction

MObile Ad-hoc Networks (MANET) are basically wireless networks where terminals are mobile and contribute themselves to the routing operations of the network. MANET are self-configuring and infrastructure-less networks, with no need for any centralized entities and operators' management.

The easiness of deployment and the resulting financial gain are among the most interesting features of ad-hoc networks. Service providers and network operators are highly likely to take advantage of ad-hoc networks by providing to ad-hoc customers their ordinary and newly-defined services. Therefore access control infrastructures (e.g. AAA: Authentication, Authorization, and Accounting) are a hot topic in this kind of networks as they will help to support subscribers charging.

In our two previous articles [1] and [2], we proposed a theoretical AAA framework that allows a joining node JN to authenticate itself to a group of AAA servers in a MANET. When the authentication succeeds, the servers deliver an Access Token to JN thanks to which the neighboring nodes can check the legitimacy of the JN, before granting access.

An authentication protocol is executed between the JN and several AAA servers during the authentication phase. The aim of this paper is to analyze the runtime inherent to this protocol when the number of AAA servers increases, on one hand, and when the number of hops to these servers increases, on the other hand. We believe that our work is novel because we have not seen such an analysis elsewhere.

The paper is outlined as follows. In the second section, we present the related work. In the third section, we give a reminder of the distributed AAA infrastructure that we detailed in [1] and [2]. The two following sections deal with the evaluation of the runtime using two methods: modeling and simulation. The fourth section is dedicated to the modeling part where we make some hypothesis to establish our model. The fifth section is dedicated to the simulations that serve to validate our model.

Sondes LARAFA, PHD student and Maryline LAURENT, professor, are with TELECOM & Management SudParis, 9 rue Charles Fourier, 91011 EVRY, FRANCE, CNRS Samovar UMR 5157

Email: {Sondes.Larafa, Maryline.Laurent}@it-sudparis.eu

II. Related Works

The most well known research works that considered the distribution of an authentication service to multiple servers proposed distributing the Certification Authority (CA).

COCA [3] defined and simulated a distributed CA architecture for local networks with Ethernet connections at 100 Mbps and focused specially on solutions for some given security attacks that the CA may undergo. It did not analyze formally the time necessary to accomplish client requests for certificates. Moreover the protocol messages went through delegate servers before reaching the CA servers, which increased the number of rounds and so the overhead.

MOCA [4] defined, however, a less complex protocol than that of COCA. Simulations, only, were used to evaluate it.

DICTATE [5] is another solution for requesting certificates to a distributed CA. The defined protocol is based on probabilistic quorum systems and is much more complex than the protocol we propose in this paper. As several rounds are required, a modeling was proposed, however it did not address the protocol runtime.

III. Authentication Protocol within a MANET Distributed AAA infrastructure

A centralized AAA infrastructure is traditionally composed of an AAA server, an AAA client located in a Network Access Server, and a client (a subscriber) which authenticates to the AAA server via the AAA client before accessing to the operator's network. To distribute this architecture and make access control possible in ad-hoc networks, we replace the single server by a group of AAA servers and we place the AAA client directly into the client (subscriber) device. As such, an ad-hoc node (any node from the ad-hoc network) is either an AAA server or an AAA client. AAA clients and AAA servers form the distributed AAA framework.

AAA servers are chosen and bootstrapped by an offline authority that generates the system key and shares it (by means of threshold cryptography [1], [6], [7]) among the servers. The key-shares are so configured and will be refreshed by this

authority. Clients credentials are also configured by this same authority that also fills their caches with the servers addresses.

An authentication protocol, at the application layer, takes place between an AAA client, e.g. a Joining Node JN, and the group of AAA servers. Both parties authenticate themselves using RSA asymmetric cryptography [8]. During the authentication phase, the JN connects to the AAA servers. Actually, by means of threshold cryptography, it requests authentication to at least a threshold number of them. For the sake of simplicity, we take the threshold number equal to the number of AAA servers in our paper.

Here are the execution steps of the authentication protocol [2]:

(1) JN sends to each server a request for authentication that includes its identity (present in its public key certificate), MSG1: $\{ID_{JN}\}$.

(2) The servers respond with a challenge in the form of a random number [1], MSG2: $\{R_{AAA}\}$.

(3) JN generates a random number R_{JN}. Then it signs, using its RSA private key, both random numbers and the identity of the group of AAA servers (ID_{AAA}). Next, it answers each server sending this signature accompanied by its public key certificate $cert_{JN}$, its random number, and the identity of the AAA service, MSG3: $\{cert_{JN}, R_{JN}, ID_{AAA}, Sign_{JN}(R_{JN}, R_{AAA}, ID_{AAA})\}$.

(4) If the servers succeed to decipher JN's signature and to establish the integrity, each one of them computes a signature piece [7] using its RSA key-share [6] on both random numbers and on the identity of JN (this is one of the threshold cryptography aspects). They also generate an access token T_{JN} for the JN that is sent with the signature pieces accompanied by the public key certificate of the AAA service and the identity of the JN, MSG4: $\{cert_{AAA}, ID_{JN}, Sign_{AAA}(R_{AAA}, R_{JN}, ID_{JN}), T_{JN}\}$

These steps are inspired from the ISO-three way protocol (ISO [9798-3] [9]) that we adapted to our distributed context.

Once the JN successfully validates the integrity of the servers signature pieces (by combining them first [7]), the mutual authentication between the JN and the servers is considered as successful. JN is henceforward authorized to access the network.

So far, authentication and authorization have been addressed in this framework. The accounting function is not yet supported, but, as a hot topic, it will be addressed in future works.

IV. Protocol Modeling for a Theoretical Evaluation of the Authentication Runtime

The present section outlines the reasoning for building a model and computing the runtime of the authentication protocol exposed in section III. It starts by analyzing the events sequence at the nodes from the construction of the first message MSG1 by the JN until its reception by one of the AAA servers, call it AAA_j (cf. Fig.1). Once the runtime of MSG1 with one single server is known, the reasoning simply applies to the other three messages of the protocol, MSG2, MSG3, and MSG4, and for the remaining servers, AAA_1, AAA_2, ..., AAA_n if n is the number of servers.

Our analysis takes into account the possible retransmissions of messages by the MAC layer, and assumes that the DCF technic used is basic DCF ([10], [11]).

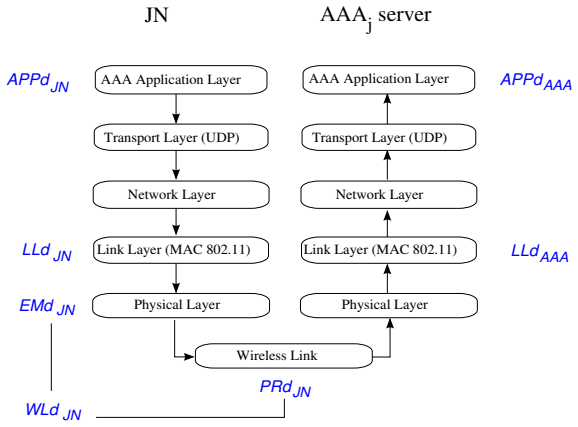

Fig. 1. Events sequence of the first message MSG1

Fig.1 illustrates the events sequence when the JN and the server are one-hop away from each other. At the JN's:

1) AAA Application Layer generates the first message containing the identity of the JN. The delay is $APPd_{JN}$ (Application delay).
2) After going through the Transport and the Network Layers, which takes a negligible time, the resulting packet enters the buffer of the Link Layer. The time spent in the Link Layer is LLd_{JN} (Link Layer delay).
3) During transmission over the Wireless Link, the packet might be subject to collisions or broken routes. Transmission delay (WLd_{JN}) has to take into account the possible packet retransmissions [10], as well as the emission delay (EMd_{JN}) and the propagation delay PRd_{JN}. Considering the speed of electromagnetic waves in the air, the latter is actually insignificant (about 3 μs/km)

At the AAA_j server:

1) Packets coming from the JN are placed in a buffer of the Link Layer. A packet is processed after LLd_{AAA} time.
2) After going through the Network and the Transport Layers, the packet is processed by the AAA Application Layer during $APPd_{AAA}$.

Thereby, the delay d_{1_j} for the first packet generation, transmission to AAA_j and processing is:

$$d_{1_j} = (APPd_{JN} + LLd_{JN} + WLd_{JN}) + (LLd_{AAA} + APPd_{AAA})$$

A. Model Features

From now, we suppose that the computing operations within the nodes (so within JN and AAA_j) are fast enough to neglect the delays $APPd_{JN}$ and $APPd_{AAA}$. We also suppose that there is practically no other packets, except the authentication packets, in the Link Layers of the nodes, so LLd_{JN} and LLd_{AAA} are negligible, too. Thereby:

$$d_{1_j} = WLd_{JN}$$

Fig. 2. Delay on a 3-hop link

Besides, let us consider the case where the JN and AAA_j are not necessarily one hop away from each other but at a number of hops $hops = 3$ (cf. Fig.2); and assume that the routes are fixed between all the nodes, so between JN and AAA_j. Hence the relaying nodes do routing operations into a fixed time supposed equal to zero. Thus:

$$WLd_{JN} = WLd_1 + WLd_2 + WLd_3$$

So, if $hops$ is any number of hops:

$$WLd_{JN} = WLd_1 + WLd_2 + ... + WLd_{hops}$$

and:

$$d_{1_j} = \sum_{k=1}^{hops} WLd_k \quad (1)$$

The delay WLd_k is a positive random variable having as a distribution function F_{WLd_k}. It takes into consideration the prospective retransmissions of a packet as described by the DCF basic access mechanism. The maximum number of retransmissions is equal to seven as defined in the IEEE 802.11 specifications [10]. If p is the probability of retransmission for a packet in the wireless channel, and X the number of retransmissions (X is a discrete random variable that covers the values of the set {1..7}), then:

$$\begin{cases} P(X = i) = p^i(1-p) \ for \ 0 \leq i \leq 6 \\ P(X = 7) = p^7 \\ P(X = i) = 0 \ for \ i \geq 8 \end{cases}$$

The Total Probability Law [12] allows to write the following formula:

$$\begin{aligned} F_{WLd_k}(t) &= P(WLd_k \leq t) \\ &= \sum_{i=0}^{7} P(\{WLd_k \leq t\}|\{X = i\}) \cdot P(\{X = i\}) \end{aligned}$$

and given the binary exponential backoff rules [11]:

$$\begin{aligned} &P(\{WLd_k \leq t\}|\{X = i\}) \\ &= (i+1)DIFS + (i+1) \cdot P(\{EMd \leq t\}) \\ &\quad + \frac{1}{2}\sum_{j=0}^{i-1} 2^j \cdot CW_{min}\theta \\ &\quad + i \cdot ACK\_Timeout + SIFS + EMd_{ACK} \end{aligned}$$

where $DIFS$, θ, $SIFS$, and $ACK\_Timeout$ are DCF timers, CW_{min} is the minimum contention window, EMd_{ACK} is the emission time of an ACK at the byte-rate of 1 Mbps [10], and EMd the emission time of a message of l bytes at the byte-rate λ. Please refer to [11] for further details concerning this formula.

If we suppose that the emission time necessary to deliver one byte is a positive continuous random variable, following an exponential distribution with parameter λ (the average byte-rate), then the necessary mean time to deliver l bytes is l/λ. Since l indicates the length of an authentication message, l is large enough (cf. table I) to apply the Central Limit Theorem [13]. Thus, the emission time of l bytes is a positive continuous random variable, EMd, following a gaussian distribution of mean l/λ and variance l/λ^2 [13].

Hence, $\forall i \in [0,7], WLd_k|\{X = i\}$ is a positive random variable following a gaussian distribution of mean μ_i and variance σ_i^2 where:

$$\begin{aligned} \mu_i &= (i+1)DIFS + (i+1)\frac{l}{\lambda} + \frac{1}{2}\sum_{j=0}^{i-1} 2^j \cdot CW_{min}\theta \\ &\quad + i \cdot ACK\_Timeout + SIFS + EMd_{ACK} \end{aligned}$$

and

$$\sigma_i^2 = (i+1) \cdot \frac{l}{\lambda^2}$$

Consequently, using the classical erf function [14]:

$$\begin{aligned} &F_{WLd_k}(t) \\ &= \frac{1}{2} + \frac{1}{2}\sum_{i=0}^{6} p^i(1-p)\,\mathrm{erf}\left(\frac{1}{2}\frac{\sqrt{2}(t-\mu_i)}{\sigma_i}\right) \\ &\quad + \frac{1}{2}p^7\,\mathrm{erf}\left(\frac{1}{2}\frac{\sqrt{2}(t-\mu_7)}{\sigma_7}\right) \end{aligned}$$

with a mean μ_{WLd_1} and a variance $\sigma^2_{WLd_1}$ independent from k because all the wireless links are assumed identical. μ_{WLd_1} and σ_{WLd_1} are expressed in terms of $\{\mu_i\}_{0\leq i\leq 7}$ and $\{\sigma_i\}_{0\leq i\leq 7}$ in [11].

Accordingly, the positive random variables $\{WLd_k\}_{1\leq k\leq hops}$ follow the same probability law. Since each transmission of a packet on a specific hop is independent from the transmission of the same packet on another hop, these random variables are independent and the Central Limit Theorem [13] applies again. Hence, given the equality (1), d_{1_j} follows a gaussian distribution of mean $\mu_{d_{1_j}} = hops \cdot \mu_{WLd_1}$ and variance $\sigma^2_{d_{1_j}} = hops \cdot \sigma^2_{WLd_1}$

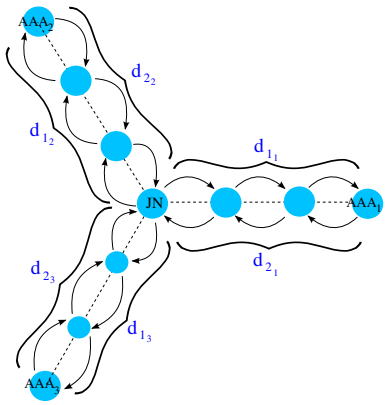

Fig. 3. First round-trip of the authentication protocol ($n = 3$,$hops = 3$)

Now, if d_{2_j}, d_{3_j} and d_{4_j} are respectively the delays for the second, the third and the fourth message of the authentication protocol, d_{2_j}, d_{3_j} and d_{4_j} have similar distribution functions as d_{1_j} (only the length of their corresponding messages differs).

Let $d_{12_j} = d_{1_j} + d_{2_j}$ and $d_{34_j} = d_{3_j} + d_{4_j}$ be the delay for respectively the first and the second round-trip of the protocol through the server AAA_j, and n the number of AAA servers. The delays $\{d_{12_j}\}_{1\leq j\leq n}$ (respectively $\{d_{34_j}\}_{1\leq j\leq n}$) are different for each server (because the transmissions on the links between the JN and the AAA servers can not be exactly the same for each link and at any moment), however they follow the same probability law. Also, suppose that the number of hops between the JN and each server is the same i.e. equal to $hops$ (cf. the example of Fig.3 when $hops = 3$ and $n = 3$).

During the first round-trip and for threshold cryptography reasons, JN has to wait for all the servers responses before triggering the second round-trip with all the servers. There are two possible approaches to compute the delay to achieve the first round-trip with all the servers:

1) "Max" model approach: JN waits the maximum of the $\{d_{12_j}\}_{1\leq j\leq n}$.
2) "Sum" model approach: JN waits the sum of the $\{d_{12_j}\}_{1\leq j\leq n}$.

In the real case, it waits the maximum. But because the network simulator NS-2 is unable to simulate parallel events and rather executes them one after another [15], it actually waits the sum. So the delay $D_{12_{sum}}$ to achieve the first round-trip with all the servers is:

$$D_{12_{sum}} = \sum_{j=1}^{n} d_{12_j}$$

As $\{d_{12_j}\}_{1\leq j\leq n}$ follow the same probability law and that the expected value (or mean [13]) is a linear operator:

$$\begin{aligned} E(D_{12_{sum}}) &= E(\sum_{j=1}^{n} d_{1_j}) + E(\sum_{j=1}^{n} d_{2_j}) \\ &= n \cdot (hops \cdot \mu_{WLd_1}) + n \cdot (hops \cdot \mu_{WLd_2}) \\ &= n \cdot hops \cdot (\mu_{WLd_1} + \mu_{WLd_2}) \end{aligned}$$

Similarly the delay $D_{34_{sum}}$ to achieve the second round-trip with all the servers verifies:

$$\begin{aligned} E(D_{34_{sum}}) &= E(\sum_{j=1}^{n} d_{3_j}) + E(\sum_{j=1}^{n} d_{4_j}) \\ &= n \cdot (hops \cdot \mu_{WLd_3}) + n \cdot (hops \cdot \mu_{WLd_4}) \\ &= n \cdot hops \cdot (\mu_{WLd_3} + \mu_{WLd_4}) \end{aligned}$$

If D indicates the total delay for a successful authentication, then:

$$\begin{aligned} E(D) &= E(D_{12_{sum}}) + E(D_{34_{sum}}) \\ &= n \cdot hops \cdot (\mu_{WLd_1} + \mu_{WLd_2} + \mu_{WLd_3} + \mu_{WLd_4}) \end{aligned}$$

where E(D) is the expected value of the total delay i.e. the authentication protocol runtime. Its expression is not given here intentionally because it is quite long and complex. We simply draw its profile in the next section. You can find more computing details in [11].

TABLE I
PARAMETER VALUES USED IN THE MODEL

Parameter	Value
1st message length ($l1$)	287 bytes
2nd message length ($l2$)	32 bytes
3rd message length ($l3$)	1593 bytes
4th message length ($l4$)	1925 bytes
byte-rate (λ)	11 Mbps
SIFS	10 μs
DIFS	50 μs
SlotTime (θ)	20 μs
CWmin	32
ACK_Timeout	334 μs
ACK message length	304 bits
ACK emission time (EMd_{ACK})	304 μs
retransmission probability (p)	0.1

B. runtime Evaluation

The runtime $E(D)$ depends on the parameters summarized in the table I. The probability of retransmissions is supposed fixed here.

The number of servers is $n \in \{1, ..., 6\}$ and the number of hops is $hops \in \{1, ..., 10\}$. The spread technic we employed is DSSS. The length of messages were indicated according to their content (cf. section III) and following the example given in [16].

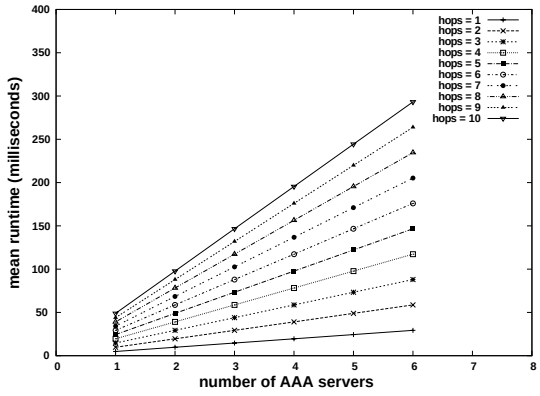

Fig. 4. Sum model: runtime with number of AAA servers varying between 1 and 6

Fig.4 depicts the evolution of the runtime $E(D)$ when the number of AAA servers and the number of hops increase. As expected, $E(D)$ increases when n rises and when $hops$ rises. The form of the curves when n increases and when $hops$ increases is roughly a line segment. The values range is between about 0.01 sec for $n = 1$ and $hops = 1$ and 0.29 sec for $n = 6$ and $hops = 10$.

V. PROTOCOL SIMULATION FOR A PRACTICAL EVALUATION OF THE AUTHENTICATION RUNTIME

We used the simulator NS-2. In the simulation, nodes were placed on concentric circles of the same center: the joining node JN. Servers are on the outermost circle of radius $100 \cdot hops$ meters. They are placed in such a way that angles are equal between them. Relaying nodes are at the intersection

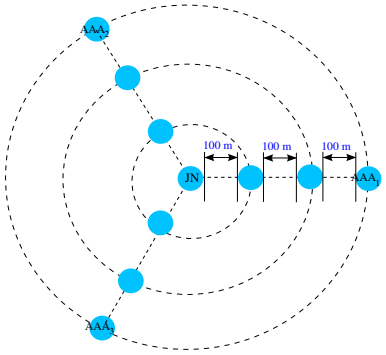

Fig. 5. Nodes placement in simulation flat-grid ($n = 3$, $hops = 3$)

of the lines joining the JN to the servers with the circles of radius $r \in \{100, ..., 100 \cdot (hops - 1), 100 \cdot hops\}$ (cf. Fig.5)..

The routing protocol used is AODV and the communication range of the nodes is 120 meters. Each value in Fig.6 was measured 100 times.

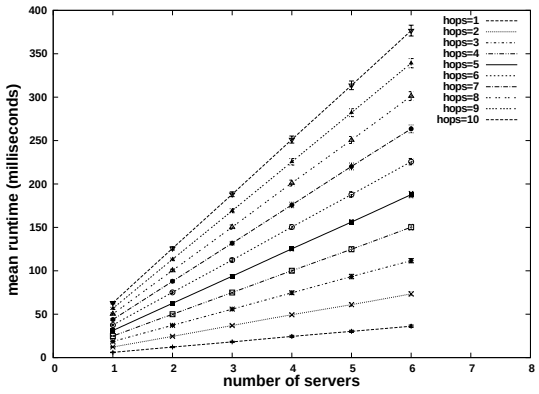

Fig. 6. Motionless simulation: runtime with number of AAA servers varying between 1 and 6

The similarity between Fig.6 and Fig.4 is striking. The range of values is slightly larger for the simulation. This difference is due to the accumulated delays of transmissions and receptions of the packets between the layers of the nodes. In the model, we supposed that these delays were negligible. However in the simulations, we realized, after processing NS-2 log files, that these delays represent about 30% of the total runtime for each value of the Fig.6 . So if we add 30% to each runtime value of the Fig.4, we will obtain approximatly the values of Fig.6. Besides, it is also to be noted that the probability of retransmission depends in fact on the number of nodes and their distribution in the network and on the amount of traffic. Its value is not fixed as we supposed in the model. But our simulations demonstrated that, in our case, its value is always less than 0.1 and the fact that it is not fixed has no impact on the shape and the values of Fig.4.

These findings are of great importance because they prove that our model is valid. They also prove that the authentication protocol is scalable for different numbers of servers and different numbers of hops. They would remain valid if the model was computing the maximum of the message delays through the servers rather than their sum: the maximum is indeed at most equal to the sum.

VI. Conclusions and Future Works

In this paper, we analyzed the runtime resulting from the authentication of a joining node by a distributed AAA framework within a mobile ad-hoc network. The built model demonstrates that when routes are already established, the runtime increases as the number of servers rises and as the number of hops rises too. This value doesn't exceed 380 milliseconds for a maximum of 6 servers and 10 hops. The undertaken simulations validated our model and hence showed that the investigated protocol is scalable when the routes are already established.

In the future, dynamic scenarios with multiple joining nodes will be treated to generalize these results. Later, a trade-off has to be found between the number of AAA servers to use and the maximum runtime to tolerate.

Acknowledgment

We are thankful to ANR (Agence Nationale de la Recherche) financing the TLCOM MobiSEND project.

We are also thankful to the CEREGMIA department of the UAG (Universite des Antilles et de la Guyane) for their collaboration during the years 2007 and 2008.

References

[1] S. Larafa, M. Laurent-Maknavicius, and H. Chaouchi, "Light and distributed AAA scheme for mobile ad-hoc networks," in *Proc. First Workshop on Security of Autonomous and Spontaneous Networks (SETOP 2008)*, Loctudy, France, october 2008, pp. 93–103.

[2] S. Larafa and M. Laurent-Maknavicius, "Protocols for distributed AAA framework in mobile ad-hoc networks," in *Proc. Workshop on Mobile and Wireless Networks Security (MWNS 2009)*, Aachen, Germany, May 2009, pp. 75–86.

[3] L. Zhou, F. B. Schneider, and R. Van Renesse, "COCA: A secure distributed online certification authority," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 329–368, 2002.

[4] S. Yi and R. H. Kravets, "MOCA: Mobile certificate authority for wireless ad hoc networks," in *The Second Annual PKI Research Workshop (PKI)*, 2003.

[5] J. Luo, J.-P. Hubaux, and P. T. Eugster, "DICTATE: Distributed certification authority with probabilistic freshness for ad hoc networks," *IEEE Trans. Dependable Sec. Comput.*, vol. 2, no. 4, pp. 311–323, 2005.

[6] A. Shamir, "How to share a secret," in *Communications of the ACM*, vol. 22, November 1979, pp. 612–613.

[7] V. Shoup, "Practical threshold signatures," in *EUROCRYPT*, 2000, pp. 207–220. [Online]. Available: http://link.springer.de/link/service/series/0558/bibs/1807/18070207.htm

[8] J. Jonsson and B. Kaliski, "Public-key cryptography standards (PKCS) #1: RSA cryptography specifications version 2.1," RFC 3447, February 2003. [Online]. Available: http://tools.ietf.org/html/rfc3447

[9] ISO [9798-3]. [Online]. Available: http://www.iso.org/iso/fr/search.htm?qt=9798-3&published=on&active_tab=standards

[10] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Standard 802.11, June 1999.

[11] S. Larafa and M. Laurent-Maknavicius, "Distributed AAA framework for MANET: performance analysis," TELECOM & Management SudParis, Tech. Rep. 09009-LOR, August 2009.

[12] D. Zwillinger and S. Kokoska, *CRC Standard Probability and Statistics Tables and Formulae*. CRC Press, 2000.

[13] M. Kendall, A. Stuart, and J. Ord, *The Advanced Theory of Statistics*. Wiley, 2009, vol. 1.

[14] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1964.

[15] UC Berkeley, LBL, USC/ISI, and Xerox PARC. (2009, January) The NS manual (formerly ns notes and documentation). [Online]. Available: http://www.isi.edu/nsnam/ns/ns-documentation.html

[16] Creating a rogue CA certificate. [Online]. Available: http://www.win.tue.nl/hashclash/rogue-ca/