

# A Measurement Study on IKEv2 Authentication Performance in Wireless Networks

Zoltán Faigl<sup>†</sup>, Stefan Lindskog<sup>‡</sup>, and Anna Brunstrom<sup>\*</sup>

<sup>†</sup>*Mobile Innovation Center, Budapest University of Technology and Economics, Budapest, Hungary, zfaigl@mik.bme.hu*

<sup>‡</sup>*Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Trondheim, Norway, stefan.lindskog@q2s.ntnu.no*

<sup>\*</sup>*Department of Computer Science, Karlstad University, Karlstad, Sweden, anna.brunstrom@kau.se*

**Abstract**—This paper presents an experimental evaluation of the performance costs of a wide variety of authentication methods over IKEv2 in wireless networks. The studied methods are pre-shared keys (PSK), extensible authentication protocol (EAP) using MD5, SIM, TTLS-MD5, TLS, and PEAP-MSCHAPv2. For the EAP-based methods RADIUS is used as authentication, authorization, and accounting (AAA) server. Two network scenarios, WiFi and UMTS, are considered. The measurement results illustrate the practical costs involved for IKEv2 authentication, and show significant performance differences between the methods.

## I. INTRODUCTION

Next generation wireless networks can be characterized by a heterogeneity in radio access networks (RANs), backbone networks, mobile terminals and applications. They are open in terms of allowing and supporting third-party service providers to deploy and compose application services. They allow mobile users to engage in all kinds of Internet transactions and services with appropriate trust and security relationship management. One of the main issues on the connectivity level is how to seamlessly integrate the heterogeneous RANs and to realize two major functionalities on top of these networks. The two functionalities are authentication, authorization, and accounting (AAA), and mobility support. The implementation of these functionalities must satisfy real-time constraints required by mobile applications and a large terminal base.

The performance cost evaluation of security protocols, such as the Internet key exchange version 2 (IKEv2) protocol [8], contributes to the research of sophisticated security design methods, where the aim is to find the best security configurations that fit to the needs of a given application in a given environment. Different security configurations have different costs in terms of performance and provide different security levels.

There are only a few papers that study the performance costs of the IKEv2 protocol even though it is an important issue in network and upper-layer service design. In [10], Soussi et al. compared the performance costs of IKEv1 and IKEv2 in a National Institute of Standards and Technology (NIST) simulation environment. Springer and Kilmartin [11] compared the performance of IKEv1 and IKEv2 in an OPNET simulation environment. They used pre-shared key (PSK) based authentication methods. PSK-based authentication is,

however, not well suited for large-scale environments due to the scalability problems. In [4], the performance overheads of IKEv2 reauthentications using EAP-TLS was analyzed by the authors.

In this paper, we measure and evaluate the computational and message transfer costs of different IKEv2 authentication methods, and show the resulting authentication delays of the methods in two different scenarios. Our measurement results show that TLS-based methods, such as TTLS-MD5, TLS, and PEAP-MSCHAPv2, can introduce significant delays even in small scale scenarios. The TLS-based methods are also more computationally demanding than the other investigated methods. The cost values obtained from our measurements can be used to estimate the authentication delay also in other network environments, which in turn can be useful in security design or for node dimensioning. The measurement results presented in this paper constitute a subset of the work presented in [5].

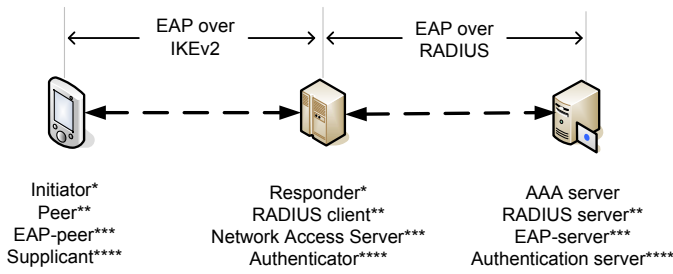
The remainder of the paper is organized as follows. Section II presents the technologies and scenarios investigated in the paper. Section III presents the authentication methods we considered. Section IV describes the measurement results. Section V discusses the results from a high-level perspective. Finally, Section VI draws the main conclusions.

## II. TECHNOLOGIES AND SCENARIOS

Different protocol standards use different terminologies. The naming conventions used to refer to the participants of the protocols are summarized in Fig. 1. In this paper we mainly use the terminology of the IKEv2 standard that is depicted in the first line of the figure. Furthermore, we often refer to the initiator and the responder as peers, when both are considered.

### A. Technologies

IKEv2 is a set of protocols and mechanisms designed to perform two functions: creation of a protected environment, which includes authentication of peers that are unknown to each other in advance, and to establish and manage security associations (SAs) between the authenticated peers based on their security policy databases. The IKE protocol has recently been revised and the most up-to-date version is IKEv2. An important improvement in IKEv2, compared to its successor,



\*IKE terminology. \*\*RADIUS terminology. \*\*\*EAP terminology. \*\*\*\*802.1X terminology.

Fig. 1. Reference scenario.

is the ability to provide authentication through the extensible authentication protocol (EAP) [1]. AAA service functionality at the responder side can thereby be delegated to a central AAA server. Besides authentication data, EAP methods can also convey generic configuration data. These features make IKEv2 with EAP attractive for use in next generation mobile networks.

IPsec is used to provide security services at the IP layer and provide a security framework that enables the appropriate selection of security services for a given traffic of IP datagrams. IPsec SAs assure confidentiality, message origin authentication, integrity, and anti-replay protection of the communication. IPsec and IKEv2 together provide cryptographically strengthened service access to the peers.

The RADIUS protocol is used to perform authentication and master key establishment of the initiator by a common authentication server. RADIUS can act as an EAP server and supports a large set of EAP-based authentication methods.

### B. Considered Scenarios

The reference scenario, depicted in Fig. 1, may provide authorization for any service where IPsec connection establishment is required between the peers. Authentication of the participating entities is made, before establishing IPsec SAs, in the following way. The initiator is authenticated by the AAA server using some EAP-based authentication. The AAA server is authenticated to the initiator if the EAP method enables it. The responder and the AAA server are assumed to have a pre-established trust relationship, i.e., they share a secret used for the authentication and integrity protection of RADIUS messages. Moreover, IKEv2 authentication performs the mutual authentication of the peers based on the shared secret.

Two different instantiations of the reference scenario have been used for our measurements. The physical location of the responder and the AAA server was the same in both scenarios. The initiator reached the responder either over an IEEE 802.11g WiFi RAN or over a UMTS R99 RAN. Fig. 2 illustrates the structure of our experimental network.

The considered scenarios represent the cases where IKEv2 is used as a mechanism to provide integrated network access authorization for various RANs. The responder is one or few

hops away from the initiator, and functions as authenticator and enforcement point providing access to the Internet. IKEv2 is also used in IPv6-based mobility services to authenticate the mobility service subscribers, and establish IPsec SAs protecting mobility signaling. In this case, the initiator is the mobile node or mobile router, and the responder is the mobility service provider, i.e., the home agent.

### III. IKEV2 AUTHENTICATION

IKEv2 supports three main authentication method types, i.e., PSK-based, certificate-based, and EAP-based authentications. In all methods, the aim is to authenticate the initiator and the responder to each other. In the rest of this paper, the focus is on PSK and EAP authentication. A typical IKEv2 authentication message flow when PSK is used is illustrated in Fig. 3.

The IKE initialization phase typically consists of two IKE\_SA\_INIT messages and two or more IKE\_AUTH messages exchanged between the initiator and the responder. The aim of the IKE initialization phase is to create an IKE SA pair between the initiator and responder. The IKE\_SA\_INIT messages result in the creation of a shared secret between the two peers using Diffie-Hellman (DH) key exchange. From this shared secret further shared keys are generated. Two of these keys are  $SK_{p,initiator}$  and  $SK_{p,responder}$ . These will be used during the authentication of the peers when calculating authentication data (Auth). The primary aim of IKE\_AUTH messages is to authenticate the peers. The authentication procedure utilizes the  $SK_p$  keys as input in order to join the key material generated at the end of the IKE\_SA\_INIT phase with the identity of the authenticated peers. In case of PSK, two IKE\_AUTH messages are exchanged.

In case of EAP, the number of IKE\_AUTH messages depends on the number of exchanged EAP fragments between the initiator and the authentication server, which in turn is strongly dependent on the specific method chosen for authentication.

Five authentication methods over EAP are considered here: MD5, SIM, TTLS-MD5, TLS, and PEAP-MSCHAPv2. MD5 [1] is a challenge-response authentication. SIM [7] is an EAP method making use of the GSM SIM module to authenticate the user. Note that EAP-SIM can be used for authentication independently of the access network. In both MD5 and SIM, only the initiator is authenticated. In TTLS-MD5 [6], TLS [2], and PEAP-MSCHAPv2 [9] mutual authentication is provided. In all three methods, the responder is authenticated using certificates. In TTLS-MD5, the initiator is authenticated using MD5, whereas in TLS the initiator is authenticated using a certificate. In PEAP-MSCHAPv2, finally, the Microsoft challenge handshake protocol version 2 [13] is used to authenticate the initiator.

For TTLS-MD5, TLS, and PEAP-MSCHAPv2, we also consider two subtypes, which differ in the certificate chain lengths used for the initiator and AAA server. The certificate chain length represents the number of certificates to verify until the root certificate authority (CA) certificate is verified. The root CA was in all cases the common trusted CA of

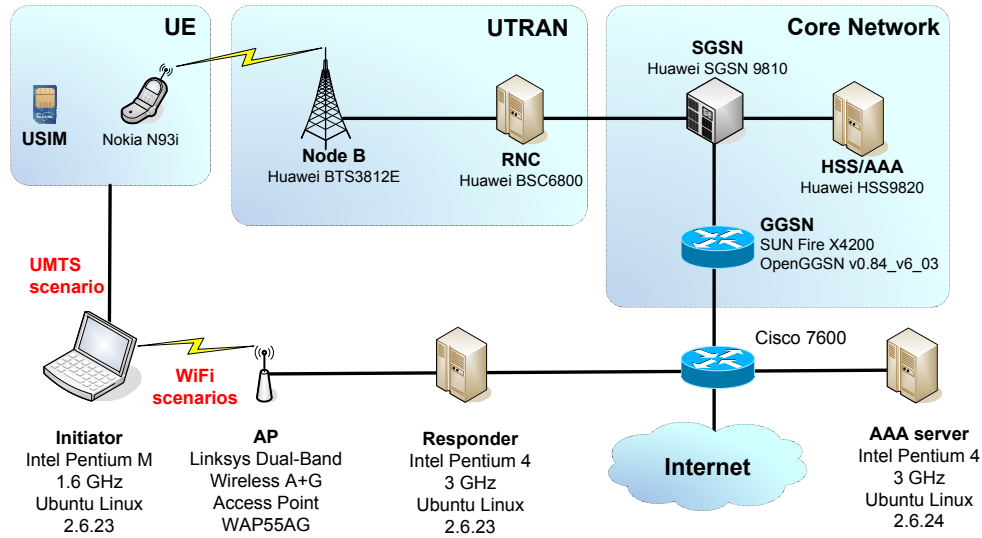


Fig. 2. Measurements scenarios.

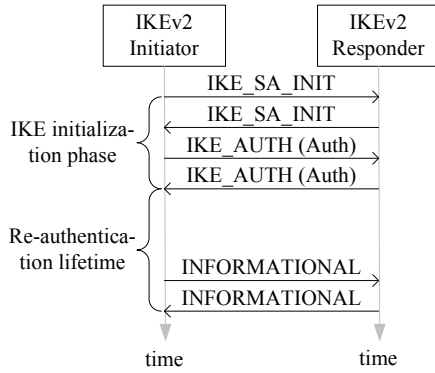


Fig. 3. Message flow of PSK authentication.

the initiator and AAA server. The different certificate chain lengths considered are two and four. In the rest of the paper, TTLS-x.MD5, PEAP-x.MSCHAPv2, and TLS-x-y denote the certificate-based methods, where x and y indicate the lengths of server and client-side certificate chains.

To maintain a high level of security in a communication session, periodic reauthentications are necessary. The IKE reauthentication lifetime controls the frequency of reauthentications of the peers. When peers reauthenticate, the whole IKE negotiation process is repeated. The old IKE SA pair and the IPsec SA pairs negotiated using the old IKE SA are deleted. This is done with the INFORMATIONAL messages. Reauthentication of peers is even more important than rekeying of SAs. We therefore measure the performance aspects of whole reauthentication processes. If only one IPsec SA pair is needed by the security policies and the actual traffic, then the reauthentication process is exactly the IKE initialization phase.

#### IV. MEASUREMENT RESULTS

The measurement results are presented in this section. The results from the authentication delay measurements are provided, followed by the results from the computational cost measurements. Before introducing the measurement results, our experimental setup is described.

##### A. Experimental Setup

In the experimental setup, the initiator side was realized using a laptop with Intel Pentium M 1.6 GHz CPU with fixed CPU frequency. The responder and the AAA server were two desktop PCs with Intel P4 3 GHz CPUs. All computers used the Ubuntu Linux operating system with a 2.6.23 version kernel at the initiator and the responder, and 2.6.24 kernel at the AAA server. The AAA server was realized with freeRADIUS version 1.1.7 with openssl. The IKEv2 daemons at the initiator and the responder was `ikev2-2.0beta1` from the IKEv2 project [12]. The initiator and the responder communicated through IPv6. However, IPv4 was used between the responder and the AAA server. This was due to the fact that IPv6 was not supported by the RADIUS client within the IKEv2 daemon at the responder.

In order to measure the authentication delay, we first established security policies in the IPsec security policy database of the initiator and the responder to protect the ICMPv6 traffic between the initiator and responder. Next, we configured the IKEv2 initiator and responder to establish transport mode IPsec SAs whenever there is traffic requiring protection. At the responder side IKEv2 was set to require IKE reauthentication lifetimes of 8 seconds for the IPsec SA pairs, long enough to ensure that the previous authentication has completed. The EAP authentication methods were configured at the AAA server and initiator side, respectively.

In order to measure authentication delays we triggered ICMPv6 Echo Request messages to be sent from the initiator to the responder every 10 seconds. Every ICMPv6

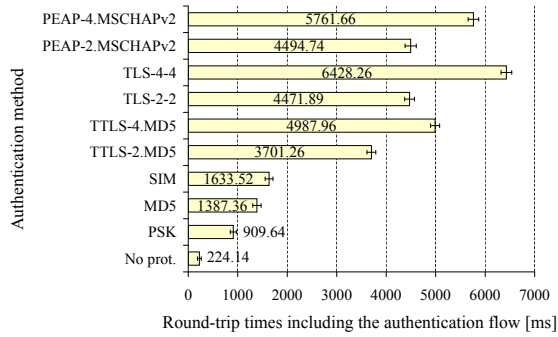


Fig. 4. IKEv2 reauthentication delays in the UMTS scenario.

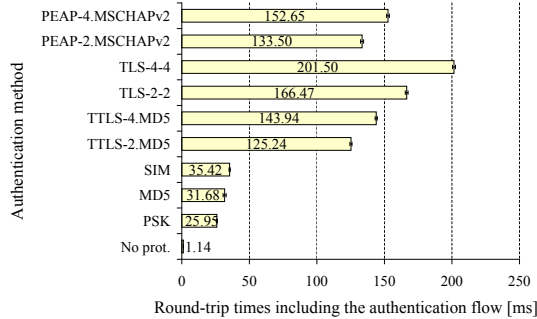


Fig. 5. IKEv2 reauthentication delays in the WiFi scenario.

Echo Request triggers an IKEv2 authentication flow, since the reauthentication lifetime is shorter than the ping interval. The round-trip times of the pings contain the duration of the authentication flow, and the round-trip time of the ping. More than 30 samples of authentication delay for each method in each scenario were measured.

The processor utilization measurements were performed using the Oprofile software [3]. We used Oprofile in timer interrupt mode in our measurements. In this mode, Oprofile collects samples on the processes which utilize the CPUs at each timer interrupt signal triggered by the kernel. The timer interrupt frequency was set to a fixed value of 1000 Hz in the kernel of the nodes. We measured 20 reauthentications within one profile. We repeated the measurements five times for each method. Within one measurement, the profiling was started just before the first authentication flow and stopped after the last authentication flow. The start-up phases of the involved daemons were not measured within the profiles.

### B. Authentication Delay Measurements

Fig. 4 and Fig. 5 present the reauthentication delays of the methods in case of the UMTS and WiFi scenarios, respectively. The figures represent the mean and the 95% two-sided symmetric confidence intervals of the obtained authentication delays. Within the UMTS scenario the effect of the number of message transfer rounds due to the high message transfer delay of UMTS is noticeable. The ordering of methods in Fig. 4 is almost reflecting Fig. 6, which illustrates for each method the number of authentication messages that are transferred between the nodes. Two exceptions can be seen, i.e., the delays of the

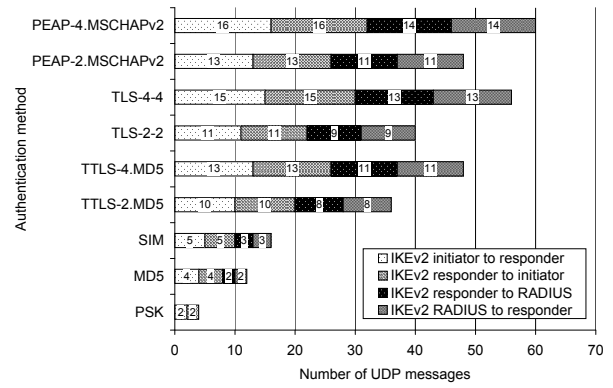


Fig. 6. Number of authentication messages in one authentication flow.

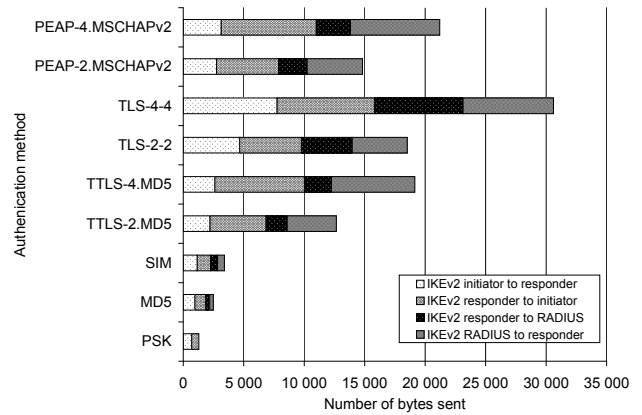


Fig. 7. Number of bytes sent in one authentication flow.

TLS-2-2 and TLS-4-4 methods are larger than expected if only looking at Fig. 6. As the bandwidth for UMTS is limited, the message transmission delays and thus the size of the messages, see Fig. 7, also become important for the authentication delay. The WiFi scenario has small transfer delays and high data rate. Thus, the computational delay also contributes considerably to the total delay in this scenario.

It is interesting to investigate the absolute values of the reauthentication delays in the two scenarios. Values from 25 ms to over 6 seconds can be seen. These values can be considered during security design. The WiFi scenario illustrates the minimum delay case, since here the participants are all located on a small site with very small transfer delays and high data rates. Already in that case the reauthentication delays of certain methods exceed 150 ms (see Fig. 5). Furthermore, the differences between the considered authentication methods are large, with the TLS-based methods being several times slower than PSK, MD5, and SIM.

### C. Utilization Measurements

Fig. 8, Fig. 9, and Fig. 10 present computational cost values of one authentication flow at the initiator, responder, and AAA server, respectively. The cost values were derived from the samples taken by the profiler at each one of the nodes. The mean cost of an authentication flow of a given method was

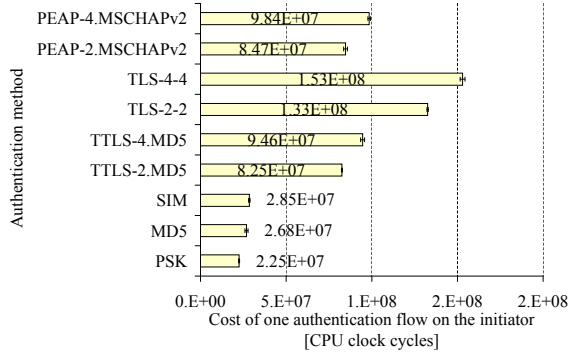


Fig. 8. Cost of one authentication flow at the initiator.

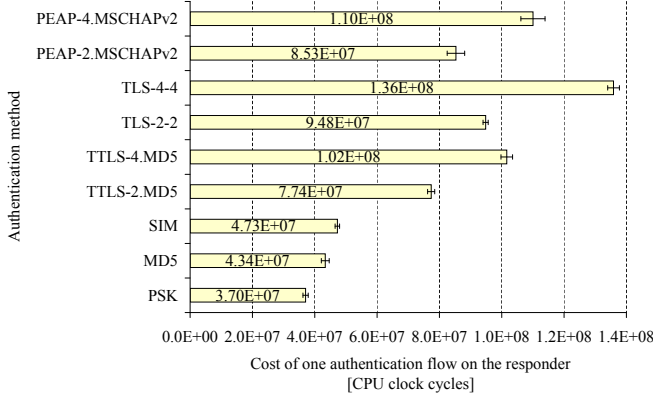


Fig. 9. Cost of one authentication flow at the responder.

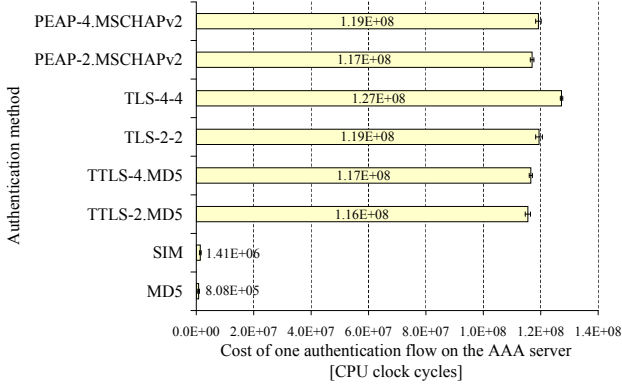


Fig. 10. Cost of one authentication flow on the AAA server.

calculated using (1).

$$C_{\text{comp}} = \frac{S f_{\text{CPU}}}{n f_{\text{timer interrupt}}} \quad (1)$$

where  $f_{\text{CPU}}$  is the CPU frequency,  $f_{\text{timer interrupt}}$  is the timer interrupt frequency of the kernel,  $S$  is the collected number of samples, and  $n$  is the number of authentication flows within one profile result, i.e., 20 in our case.

Equation (1) results in an absolute cost metric that is characteristic for a given node and the processor architecture used. It gives the computational cost in terms of number of CPU clock cycles, and is valid for any CPU speed. Assuming

that the processor is otherwise idle, the computation time of an authentication flow on a given node can be estimated by dividing the cost  $C_{\text{comp}}$  with  $f_{\text{CPU}}$ . The profiling results were obtained for nodes with a single CPU using one timer interrupt flow for profiling. Computational time should be calculated differently in case of parallel processing.

The computational costs at the initiator (see Fig. 8) show that the TLS methods are computationally most demanding for the initiator. In those cases the initiator must perform public key signature operations on part of the messages in the handshakes. Besides that effect, we can see that the computational cost values correlate to a certain level with the number of bytes sent between the initiator and the responder. The number of bytes influence the amount of symmetric encryptions required.

The computational costs at the responder (see Fig. 9) are mainly depending on the number of bytes to encrypt, since the responder has mainly a relaying role. It also takes part in DH public value computation in the IKEv2\_INIT\_SA phase, but that cost is the same for every method.

The costs on the AAA server (see Fig. 10) are approximately  $10^8$  for all the TLS-based methods, and  $10^6$  in case of MD5 and SIM. That is because the AAA server has to perform TLS-server related jobs, i.e., asymmetric cryptographic operations. In case of the PEAP-MSCHAPv2 and TTLS-MD5 methods, which have two phases, the AAA server has to encrypt and decrypt the second phase messages within the TLS session. For the TLS methods, the server has to perform signature verifications. TLS-4-4 has moderately higher cost than TLS-2-2, since it needs to perform signature verifications on a four-tier, instead of a two-tier, certificate chain of the client. MD5 and SIM basically do few symmetric cryptographic operations, such as hashing.

The AAA server-side computational cost might be important from the aspect of node dimensioning from a provider perspective. The results show that TLS-based authentications are typically 100 times more demanding.

## V. DISCUSSION

Although reauthentications are not expected to be made very often, i.e., in the order of minutes at maximum, real-time applications may experience problems during handovers when IKEv2 is used for network access service authorization. The satisfaction of users may depend on the experienced authentication delays. Already in our small scale scenario, some of the IKEv2 methods introduced delays that are too large to provide fully seamless reauthentications for real time applications.

The computational costs of the methods could be considered in node dimensioning or security design. Knowing the costs of one authentication flow and the arrival rate of reauthentication processes, the required computational capacity can be calculated. This could be interesting for AAA providers. Furthermore, given the computational speed, we can estimate the computational times required in the nodes. This can also

be important in the estimation of the authentication delays experienced by the users.

The computational costs of authentication methods might be high in some application scenarios in next generation wireless networks. All IKEv2 authentication methods rely on asymmetric cryptographic primitives, such as RSA encryption and decryption, calculation of DH public values and the generation of DH secrets. IKEv2 with PSK applies at least one DH key exchange in the initialization phase. The considered TLS-based methods perform a number of RSA signature verifications depending on the length of the certificate chains, and sign part of the handshakes. In order for the use of IKEv2 to be feasible, the mobile devices thus need to have a computational capacity in which asymmetric cryptographic operations can be made in a usable time frame.

## VI. CONCLUDING REMARKS AND FUTURE WORK

This paper has presented an experimental evaluation of the performance costs of different IKEv2 authentication methods. Performance cost values obtained from real measurements are an important complement to theoretical analysis or simulation environments, and capture all the overhead involved in the system. The measurement results can also be used as input parameters for analytical modeling of authentication processes.

The ping statistics results describe the effect of reauthentications on the application level. The results show that TLS-based methods can introduce significant delays even in small scale scenarios. The computational cost measurements show that the TLS, TTLS-MD5, and PEAP-MSCHAPv2 methods are also significantly more computationally demanding than MD5, and SIM at the AAA server.

In the future, we would like to follow two main research directions. The first is to find appropriate security design methods that take into consideration the environmental characteristics, application preferences and the needs of the participants. The second research objective is to develop security solutions for next generation wireless networks for scenarios where the existing solutions do not perform well. Additional performance evaluations of security solutions in large-scale scenarios are

also needed. Such studies would help to identify areas where more research is required.

## ACKNOWLEDGEMENT

The first author has been co-funded by the OPTIMIX project, which is an EU ICT project. The work conducted by the second author has been financially supported by the research council of Norway. The authors are grateful to the Mobile Innovation Center in Budapest, Hungary for the possibility of making measurements on its testbed.

## REFERENCES

- [1] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. RFC 3748: Extensible authentication protocol (EAP), June 2004.
- [2] B. Aboba and D. Simon. RFC 2716: PPP EAP TLS authentication protocol, October 1999.
- [3] W. E. Cohen. Tuning programs with OProfile. *Wide Open Magazine*, 1:53–62, 2004.
- [4] Z. Faigl, S. Lindskog, and A. Brunstrom. Analyzing IKEv2 performance when protecting mobile IPv6 signaling. In *Proceedings of 4th International Symposium on Wireless Communication Systems (ISWCS 2007)*, pages 390–395, Trondheim, Norway, October 16–19, 2007.
- [5] Z. Faigl, S. Lindskog, and A. Brunstrom. Performance evaluation of IKEv2 authentication methods in next generation wireless networks. *Security and Communication Networks*, 2009. To appear.
- [6] P. Funk and S. Blake-Wilson. RFC 5281: Extensible authentication protocol tunneled transport layer security authenticated protocol version 0 (EAP-TTLSv0), August 2008.
- [7] H. Haverinen and J. Salowey. RFC 4186: Extensible authentication protocol method for global system for mobile communications (GSM) subscriber identity modules (EAP-SIM), January 2006.
- [8] C. Kaufman. RFC 4306: Internet key exchange (IKEv2) protocol, December 2005.
- [9] A. Palekar, D. Simon, J. Salowey, H. Zhou, G. Zorn, and S. Josefsson. Protected EAP protocol (PEAP) version 2, October 2004. Work in progress.
- [10] H. Soussi, M. Hussain, H. Afifi, and D. Seret. IKEv1 and IKEv2: A quantitative analyses. *Proceedings of World Academy of Science, Engineering and Technology*, 6:194–197, June 2005.
- [11] B. Springer and L. Kilmartin. Performance of the Internet key exchange protocol for securing VoIP networks. In *Proceedings of the Irish Postgraduate Telecommunications Symposium*, Dublin, Ireland, October 2003. <http://www.dspru.nuigalway.ie/site/publication/96/>.
- [12] J. Vucak, L. Jelenkovic, and M. Golub. Implementation of EAP authentication into IKEv2 protocol. In *Proceedings of the Information Systems Security, MIPRO 2007*, pages 173–176, Opatija, Croatia, May 21–25, 2007.
- [13] G. Zorn. RFC 2759: Microsoft PPP CHAP extensions, version 2, January 2000.