AVISPA versus AVANTSSAR in the Model Checking of Secure Communication Protocols

Iulian Aciobanitei[1], Roxana-Ioana Guinea[2] and Mihai-Lica Pura[2]
[1]*Cyber Security Advanced Technologies Center of Excellence, Military Technical Academy, Bucharest, Romania*
[2]*Department of Computers and Mathematics, Military Technical Academy, Bucharest, Romania*

Keywords: Model Checking, Communication Protocols, AVISPA, AVANTSSAR.

Abstract: The rapid development of Internet technologies has triggered a tremendous growth in the number of new communication protocols. The scientific community has started to involve formal techniques in their design, like formal verification. To this purpose a series of model checking tools has been developed, some mature enough to be used with confidence in industry. Such tools are AVISPA and AVANTSSAR, the latter one being an upgraded version of AVISPA and targeting the automated validation of distributed services. This paper presents a quantitative comparison between these two tools, from the point of view of secure communication protocols. As expected, the back-ends of the new AVANTSSAR are faster than the ones from AVISPA, but several exceptions have been identified, thus suggesting that there are situations in which AVISPA should be preferred.

1 INTRODUCTION

During the last decades computer networks have known a rapid growth in the advance of their technology, but also in their use, as today they are spread all over the world, in all industry domains and in each and every house. This has triggered a tremendous development of Internet services which use a large set of communication protocols. This set of protocols is so extensive and complex, that it surpasses the human ability to manually analyze and validate their security properties. To assure the needed confidence in the security of Internet communication through these protocols, formal verification tools were needed. They ought to be fully automated, robust and easy to use and ought to provide expressive formal specification languages for security aspects such that they could be easily integrated in the process of development and standardization of a protocol (Vigano, 2006).

Such formal verification tools specially designed for verifying security protocols are (in a chronological enumeration) Casper (used in conjunction with FDR model checker) (Lowe, 1998), ProVerif (Blanchet, 2001), AVISPA - Automated Validation of Internet Security Protocols and Applications (Alessandro Armando, 2005), AVANTSSAR - Automated VAlidatioN of Trust and Security of Service-oriented ARchitectures (Alessandro Armando, 2012), and Scyther (Cremers, 2008). From these, the scientific community has determined AVISPA and its follow-up, AVANTSSAR, to be the needed tools, as they display higher level of scope and robustness with equivalent performance and scalability (Alessandro Armando, 2005). Both AVISPA and AVANTSSAR were analyzed from a qualitative and quantitative point of view by their developers and the benchmarks data are given in the corresponding articles: (Vigano, 2006) for AVISPA and (Alessandro Armando, 2012) for AVANTSSAR, but from the best of our knowledge there were no reports of a comparison between the two.

This paper presents the results of a qualitative and quantitative comparison between AVISPA and AVANTSSAR, using a sample set of representative models for security protocols as well as an analysis of the obtained information. The motivation of such a comparison is the following. AVANTSSAR is a follow-up project of AVISPA. It is based on the same back-end model checking engines that were upgraded and improved. The main difference of the new platform is the fact that while AVISPA was built with the purpose of specifying and verifying security protocols, AVANTSSAR targets the specification and the validation of trust and service-oriented architectures and applications within the Internet of Services paradigm - applications built by compos-

Aciobanitei, I., Guinea, R-I. and Pura, M-L.
AVISPA versus AVANTSSAR in the Model Checking of Secure Communication Protocols.
DOI: 10.5220/0006887905200525
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 2: SECRYPT*, pages 520-525
ISBN: 978-989-758-319-3

ing distributed services, configured and consume in a demand-driven manner (Vigano, 2012). To address the more complex systems that needed to be modeled, the developers of AVANTSSAR have introduced new specification languages, more expressive and powerful than the input language of AVISPA. But AVANTSSAR can still be used to validate simple security protocols, just like AVISPA. So the following question can arise: is the new platform more efficient than the old one when it comes to validate simple security protocols? Or, with other words, the improvements made in AVANTSSAR in order to address the service-oriented architectures have also influenced the efficiency of the tool when it comes to the simple protocols? The answer to this question can help researchers choose one tool or the other when they need to validate a specific type of protocol.

The rest of the paper is organized as follows. The second section briefly presents the two tools. Section number three describes the comparison by presenting the comparison methodology, its results and the corresponding interpretation. The final section contains some conclusions and future work directions.

2 PRESENTATION OF THE TOOLS

2.1 AVISPA

AVISPA (Vigano, 2006),(Alessandro Armando, 2005) is a push-button tool for automated validation of Internet security protocols and applications. Protocols to be verified are modeled in HLPSL (High-Level Protocol Specification Language), presented in (Yannick Chevalier, 2004). AVISPA is equipped with web-based GUI and the actual verification process is performed by four different back-ends. AVISPA is working under the assumption of perfect cryptography and the attacker model is the Dolev-Yao intruder, described in (Dolev and Yao, 1983).

As presented in Figure 1, the main components of the AVISPA tool are: (1) the translator which is responsible for the translation from HLPSL to Intermediate Format for protocol specification and (2) the automated validation back-end. AVISPA integrates four different back-ends, as follows:

1. OFMC (On-the-fly Model Checker) is able to perform protocol falsification and bounded verification.
2. CL-AtSe (Constraint-Logic-based Attack Searcher) applies constraint solving and implements redundancy elimination techniques and simplification heuristics.

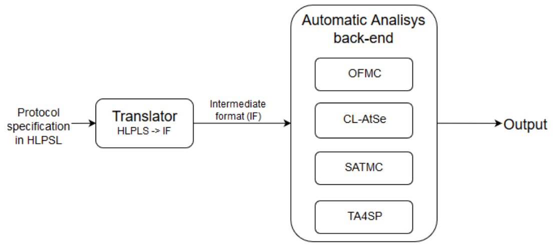

Figure 1: AVISPA Architecture.

3. SATMC (SAT-based Model-Checker) feeds a SAT solver with a propositional formula representing a violation of the security properties of the protocol. Any model found by the SAT solver this way is translated back into an attack.
4. TA4SP (Tree-Automata-based Protocol Analyzer) uses under-aproximation to show if a protocol is flawed and over-aproximation to show if a protocol is safe for any number of sessions. TA4SP uses regular tree languages and rewriting.

The output of specifies if the protocol is safe, the security properties were violated or the problem could not be solved. In case of finding a security flaw in the protocol, AVISPA gives the related attack trace.

2.2 AVANTSSAR

AVANTSSAR is a state of the art tool for security protocols/services validation. As presented in Figure 2 the platform is split into 3 layers: (1) the connectors, (2) the orchestrator, and (3) the validator.

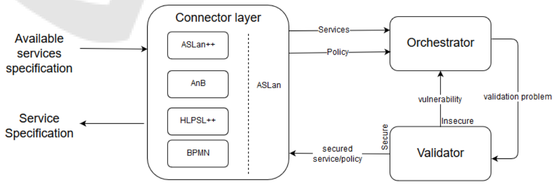

Figure 2: AVANTSSAR Architecture.

The back-end validator uses ASLan (AVANTSSAR Specification Language) as input language. ASLan is a low-level specification language designed for sequential systems. As mentioned in (Alessandro Armando, 2012), it is fully dedicated to specifying trust and security aspects of services, and goes beyond the structure of the static service. Since ASLan is a low-level language, it is difficult to be used by non-experts.

The connector layer expands the usability of AVANTSSAR by assuring the translation from high level specification languages into ASLan and from ASLan back to the high-level specification language used for the new service specified. The following connectors are available:

1. ASLan++ connector - provides translation between ASLan and ASLan++, which has been designed specifying dynamically composed security-sensitive web services and service-oriented architectures (Oheimb and Modersheim, 2011).
2. AnB connector is used in case one prefers extended Alice-and-Bob notation or message sequence charts.
3. HLPLS++ connector translates the HLPSL models (Yannick Chevalier, 2004) and can easily be used by protocol engineers/designers.
4. BPMN + Annotations connector is responsible for the translation of business process standard languages used by practitioners in the field. This avoid forcing the business process practitioners to model the protocol twice.

The Orchestrator receives the input from the connector layer and uses automated reasoning techniques to produce a specification of the target service that is guaranteed to satisfy the specified goals. If the output specification does not meet the security requirements, the validator returns a counter-example to the orchestrator, which generates another target service specification.

The Validator is responsible checking the Orchestrator output. If the orchestration meets the security requirements, the Validator outputs the service specification to the Connectors layer. Otherwise, it provides to the Orchestrator a counter-example which is used to generate another orchestration.

Using AVANTSSAR as presented in (Alessandro Armando, 2008), Alessandro Armando et al. were able to model SAML-based SSO (Single Sign On) protocol used by Google Applications. They discovered a severe security flaw that was unknown at that time and that allowed a dishonest service provider to impersonate a user at another service provider. The attack was reproduced in the actual deployment of Google Application.

3 AVISPA VS. AVANTSSAR

For the comparison, the AVISPA and AVANTSSAR platform validators were run for 29 of the most known security protocols to analyze the performance of the verification. The following data was considered: the running time of the validators for each security protocol and the number of attacks found. All validators provide statistics on the states and transitions analyzed, as well as the running time. When comparing the performance, the criteria has been the total time taken by each validator to check all the 29 protocols (the sum of the times for each protocol).

Table 1: CL-Atse Run Time.

Protocol	CL-Atse - AVISPA					CL-Atse AVANTSSAR				
	A	T1	T2	T3	Tm	A	T1	T2	T3	Tm
AAAMobileIP	NO	0.0	0.0	0.0	0.0	NO	0.1	0.08	0.09	0.09
CHAPv2	NO	0.0	0.0	0.0	0.0	NO	0.04	0.04	0.05	0.043
CRAM-MD5	NO	0.02	0.02	0.02	0.02	NO	0.31	0.33	0.32	0.32
DHCP-delayed-auth	NO	0.0	0.0	0.0	0.0	NO	0.01	0.007	0.008	0.0083
EKE	YES	0.0	0.0	0.0	0.0	YES	0.06	0.067	0.075	0.0673
EKE2	NO	0.0	0.0	0.0	0.0	NO	0.02	0.022	0.024	0.022
h.530	OUT OF MEMORY					NO	0.08	0.11	0.091	0.093
h.530-fix	OUT OF MEMORY					NO	0.11	0.1	0.14	0.116
IKEv2-CHILD	NO	0.02	0.02	0.02	0.02	NO	6.2	6.1	7.7	6.6
IKEv2-DS	YES	0.0	0.0	0.0	0.0	YES	0.03	0.045	0.046	0.04
IKEv2-DSx	NO	5.16	5.14	5.2	5.16	NO	112	117	122	117
IKEv2-MAC	NO	0.0	0.0	0.2	0.06	NO	1.45	1.46	1.16	1.356
IKEv2-MACx	NO	3.36	3.4	3.32	3.36	NO	61	73	71	68.3
ISO1	YES	0.0	0.0	0.0	0.0	YES	0.005	0.006	0.005	0.0053
ISO2	NO	0.0	0.0	0.0	0.0	NO	0.011	0.01	0.01	0.0103
ISO3	YES	0.0	0.0	0.0	0.0	YES	0.016	0.016	0.016	0.016
ISO4	NO	0.0	0.0	0.0	0.0	NO	0.32	0.31	0.3	0.31
Kerb-basic	NO	0.02	0.02	0.0	0.013	NO	6.58	6.42	6.52	6.506
Kerb-Cross-Realm	NO	0.04	0.06	0.04	0.046	NO	755	721	695	723.6
Kerb-Forwardable	NO	0.06	0.04	0.02	0.04	NO	23.2	22.8	22.6	22.8
LPD-IMSR	NO	0.0	0.0	0.0	0.0	NO	0.012	0.012	0.011	0.0116
LPD-MSR	YES	0.0	0.0	0.0	0.0	YES	0.007	0.006	0.007	0.006
PBK	YES	0.0	0.0	0.0	0.0	YES	0.016	0.015	0.014	0.014
PBK-fix	YES	0.0	0.0	0.0	0.0	YES	0.016	0.017	0.018	0.017
PBK-fix-weak-auth	NO	0.06	0.04	0.06	0.053	NO	26.15	26.09	26.18	26.14
SPEKE	NO	0.02	0.02	0.02	0.02	NO	0.157	0.161	0.159	0.159
SRP	NO	0.0	0.0	0.0	0.0	NO	0.036	0.035	0.028	0.033
TLS	NO	0.0	0.02	0.0	0.006	NO	0.11	0.105	0.108	0.107
UMTS_AKA	NO	0.0	0.0	0.0	0.0	-	-	-	-	-
Total time (s)					**8.798**					**8827.929**

The tests were performed on a computer with Intel Core i7-4510U processor, 4 GB of RAM, and an Ubuntu 14.04 operating system.

Each tool was used to check the following security properties of the selected protocols: secrecy and authentication. Secrecy can be modeled in each of the considered instruments. Authentication cannot be modeled by TA4SP. For the analysis of the obtained results, it is worth to take into consideration the following point: SATMC provides a number of features (e.g. model checking of LTL properties (Armando and Compagna, 2016)) that are not supported by CL-AtSe and OFMC and that are not used in the considered benchmark protocols.

The results obtained are found in Tables 1, 2 and 3, where column A specifies whether a protocol attack has been found, columns T1, T2, T3 represent 3 validator runtimes, and the Tm column represents the average running time. The result of the verification of certain protocols was declared inconclusive by the validator, not specifying whether or not an attack was found. The last line for each table highlights the total running time of the back-ends for the verification of all 29 protocols. Thus is simple to see which one has the best overall performance.

Validators versions used by AVISPA are as follows: CL-Atse - version 2.2-5, OFMC - version of 2006/02/13 (version number is not available), SATMC - version 2.1, TA4SP - version of AVISPA-1.1 (version number is not available).

The same protocols have been verified with the

Table 2: OFMC Run time.

Protocol	OFMC - AVISPA					OFMC AVANTSSAR				
	A	T1	T2	T3	Tm	A	T1	T2	T3	Tm
AAAMobileIP	NO	0.09	0.1	0.1	0.096	NO	0.031	0.035	0.037	0.034
CHAPv2	NO	0.12	0.13	0.13	0.126	YES	0.011	0.013	0.014	0.012
CRAM-MD5	NO	0.11	0.11	0.11	0.11	NO	0.017	0.02	0.021	0.019
DHCP-delayed-auth	NO	0.02	0.02	0.02	0.02	NO	0.008	0.007	0.01	0.008
EKE	YES	0.02	0.02	0.02	0.02	YES	0.026	0.025	0.027	0.025
EKE2	NO	0.02	0.02	0.02	0.02	NO	0.014	0.012	0.015	0.013
h.530	YES	0.23	0.24	0.23	0.23	NO	0.035	0.034	0.036	0.034
h.530-fix	NO	6.77	6.79	7.07	6.87	NO	0.075	0.068	0.072	0.071
IKEv2-CHILD	NO	0.32	0.31	0.31	0.313	YES	0.012	0.011	0.013	0.011
IKEv2-DS	YES	0.07	0.06	0.07	0.06	NO	0.28	0.27	0.3	0.28
IKEv2-DSx	NO	9.57	9.73	9.71	9.67	NO	0.3	0.31	0.33	0.313
IKEv2-MAC	NO	1.84	1.86	1.81	1.83	NO	0.189	0.186	0.185	0.186
IKEv2-MACx	NO	8.61	8.76	8.69	8.68	NO	0.226	0.203	0.284	0.237
ISO1	YES	0.0	0.0	0.0	0.0	NO	0.005	0.007	0.006	0.006
ISO2	NO	0.03	0.02	0.02	0.023	NO	0.011	0.013	0.012	0.012
ISO3	YES	0.01	0.01	0.0	0.006	NO	0.013	0.014	0.011	0.012
ISO4	NO	0.22	0.3	0.31	0.27	NO	0.021	0.019	0.021	0.020
Kerb-basic	NO	1	1.02	0.94	0.98	NO	0.024	0.023	0.024	0.023
Kerb-Cross-Realm	NO	3.07	2.96	2.92	2.98	NO	0.043	0.041	0.042	0.042
Kerb-Forwardable	NO	9.1	6.35	6.31	7.25	YES	0.031	0.029	0.03	0.03
LPD-IMSR	NO	0.03	0.02	0.03	0.026	NO	0.008	0.007	0.01	0.008
LPD-MSR	YES	0.0	0.0	0.0	0.0	NO	0.006	0.007	0.005	0.006
PBK	YES	0.21	0.2	0.2	0.203	YES	0.005	0.009	0.008	0.007
PBK-fix	YES	0.09	0.08	0.09	0.08	YES	0.017	0.019	0.014	0.016
PBK-fix-weak-auth	NO	2.02	2.29	2.21	2.17	YES	0.025	0.033	0.028	0.018
SPEKE	NO	1.68	1.48	1.36	1.5	YES	0.016	0.014	0.013	0.014
SRP	NO	0.06	0.04	0.04	0.046	NO	0.025	0.022	0.021	0.022
TLS	NO	0.14	0.17	0.2	0.143	NO	0.018	0.016	0.017	0.016
UMTS_AKA	NO	0.0	0.0	0.1	0.033	NO	0.008	0.007	0.009	0.008
Total time (s)					**43.755**					**1.503**

AVANTSSAR platform validators. Validators versions used by AVANTSSAR are as follows: CL-Atse - version 2.5-21, OFMC - 2012c version, SATMC - version 3.4.

In the depicted tables, the execution time of the validators of the two platforms are compared.

The results obtained after running the verification with CL-Atse validator from Table 1 will be summarized next. The CL-Atse Validator from AVISPA declared 20 out of 29 protocols as safe, 7 as unsafe, and 2 could not be verified. The CL-Atse Validator from AVANTSSAR found 21 out of 29 protocols to be safe (including the two protocols that could not be verified with AVISPA's CL-Atse), 7 to be unsafe (the same as the ones found unsafe by AVISPA's CL-Atse), and one could not be verified.

The result of the validation obtained with OFMC validator from Table 2 are the following: the OFMC Validator from AVISPA has found 21 protocols to be secure and 8 to be not secure. And the OFMC Validator from AVANTSSAR has found 21 protocols to be secure and 8 to be not secure.

Table 3: SATMC Run time.

Protocol	OFMC - AVISPA					OFMC AVANTSSAR				
	A	T1	T2	T3	Tm	A	T1	T2	T3	Tm
AAAMobileIP	NO	0.46	0.48	0.5	0.48	NO	0.24	0.32	0.18	0.24
CHAPv2	NO	0.06	0.08	0.04	0.06	NO	0.18	0.19	0.22	0.19
CRAM-MD5	NO	0.26	0.24	0.28	0.26	NO	0.08	0.0	0.0	0.02
DHCP-delayed-auth	NO	0.04	0.02	0.08	0.046	NO	0.0	0.02	0.02	0.013
EKE	YES	0.04	0.02	0.06	0.04	YES	0.13	0.18	0.12	0.143
EKE2	-	-	-	-	-	-	-	-	-	-
h.530	-	-	-	-	-	-	-	-	-	-
h.530-fix	-	-	-	-	-	-	-	-	-	-
IKEv2-CHILD	-	-	-	-	-	-	-	-	-	-
IKEv2-DS	-	-	-	-	-	-	-	-	-	-
IKEv2-DSx	-	-	-	-	-	-	-	-	-	-
IKEv2-MAC	-	-	-	-	-	-	-	-	-	-
IKEv2-MACx	-	-	-	-	-	-	-	-	-	-
ISO1	YES	0.02	0.02	0.0	0.013	YES	0.06	0.02	0.04	0.04
ISO2	NO	0.67	0.64	0.65	0.653	NO	0.96	0.84	0.82	0.87
ISO3	YES	0.1	0.12	0.1	0.106	YES	0.42	0.4	0.44	0.42
ISO4	NO	1678.9	2158	2318.2	2051.7	NO	109.1	117.4	114.5	113.6
Kerb-basic	Inconclusive	173921	175862	160898	170227	NO	10.86	10.7	10.9	10.82
Kerb-Cross-Realm	Inconclusive	45127	48492	45077	46232	NO	9.28	9.7	9.12	9.36
Kerb-Forwardable	Inconclusive	60296	62824	61311	61477	Inconclusive	71823	53246	84106	69725
LPD-IMSR	NO	0.06	0.08	0.1	0.08	NO	0.14	0.13	0.1	0.12
LPD-MSR	YES	0.0	0.02	0.04	0.02	YES	0.04	0.04	0.02	0.03
PBK	YES	0.24	0.28	0.22	0.246	YES	0.33	0.34	0.44	0.37
PBK-fix	YES	0.1	0.04	0.06	0.06	YES	0.12	0.12	0.12	0.12
PBK-fix-weak-auth	NO	0.3	0.32	0.26	0.29	NO	0.45	0.54	0.44	0.47
SPEKE	-	-	-	-	-	-	-	-	-	-
SRP	-	-	-	-	-	-	-	-	-	-
TLS	NO	1011.1	1128.1	1198.2	1112.4	NO	0.06	0.08	0.04	0.06
UMTS_AKA	NO	0.04	0.02	0.0	0.02	NO	0.04	0.04	0.06	0.04
Total time (s)					**281102.474**					**69861.93**

When running SATMC validator for the considered protocols the obtained results were the ones from Table 3. The SATMC Validator from AVISPA could only verify 16 out of the 29 protocols, because it is not supporting some of the functions used in their specifications; from these 16 verified models, 10 were declared safe and 6 unsafe. The SATMC Validator from AVANTSSAR could only verify 18 out of the 29 protocols, not supporting some functions used in their specifications; from these 18 models, 12 were declared safe and 6 unsafe (the same found unsafe by AVISPA's SATMC version).

The following figures provide a graphical comparison between the validators of the two platforms. Figure 3 depicts the fact that AVISPA's CL-Atse validator has achieved better results than AVANTSSAR's CL-Atse, being faster. Both versions have found the same 7 protocols to be unreliable.

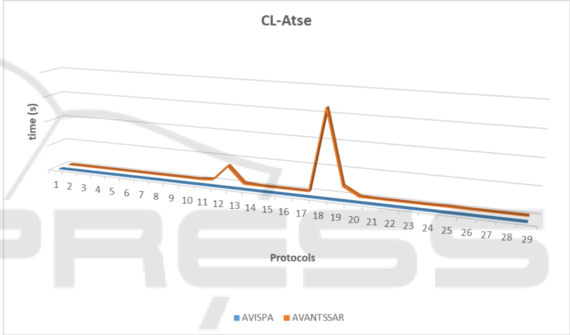

Figure 3: Comparison of CL-Atse back-ends.

Figure 4 compares the OFMC. It can be seen that this time the AVANTSSAR's OFMC was faster in analyzing the protocols than AVISPA's OFMC. Regarding finding an attack on the protocols, both versions had the same results: 21 secure protocols, and 8 not secure.

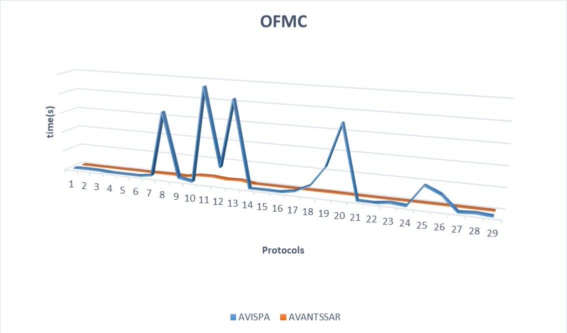

Figure 4: Comparison of OFMC back-ends.

For the SATMC validators, the graphical comparison is depicted in Figure 5. Again, SATMC from AVANTSSAR has a better performance that SATMC version in AVISPA. One can immediately observe that it is the slowest in terms of execution time (by comparison with the other two back-ends, CL-Atse and OFMC) for both platforms, as it has been running a protocol verification for 47 hours within the AVISPA platform. This validator had not supported certain functions defined in the models of the protocols, thus not being able to verify the corresponding models. The same 6 protocols have been declared safe by both validators from the two platforms.

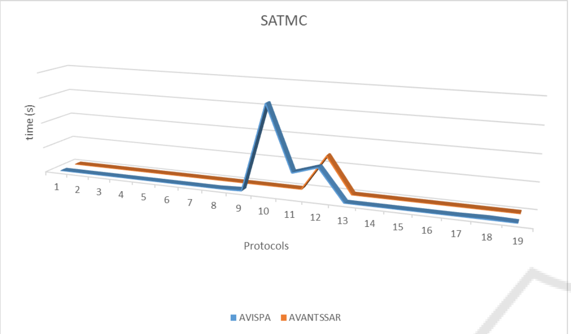

Figure 5: Comparison of SATMC back-ends.

Following the study, OFMC from the AVANTSSAR platform managed to run the verification for all protocols in 1.503 seconds, followed by AVISPA's CL-ATSE with 8.798 seconds, AVISPA's OFMC with 43.755 seconds, AVANTSSAR's CL-Atse with 8827.93 seconds, AVANTSSAR's SATMC with 19.41 hours, and lastly AVISPA's SATMC with 78.08 hours. TA4SP was discontinued in AVANTSSAR and in AVISPA it had the poorest results. The running time was quite small for CL-AtSe and OFMC, and considerably higher for SATMC, for both tools.

At the end of this section, a graphical representation illustrating the relative performance of the different back-ends for each tool is presented.

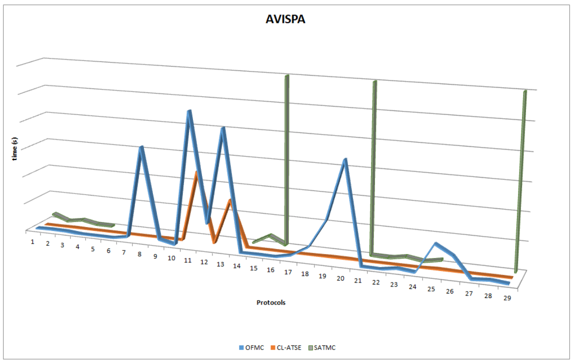

Figure 6: AVISPA's back-ends comparison.

Figure 6 depicts the comparison between the three considered back-ends of AVISPA. The missing segments from the line corresponding to SATMC run times represent the protocols for which SATMC could not be run. In general AVISPA's CL-Atse has the best performance, followed by OFMC, and then by SATMC.

Figure 7 depicts the comparison between the versions of these back-ends that are part of AVANTSSAR. Again, the missing segments from the line corresponding to SATMC run times represent the protocols for which SATMC could not be run. One can see that these are the same as for AVISPA's version of SATMC. In AVANTSSAR, the fastest back-end is OFMC, followed by CL-Atse, and then by SATMC.

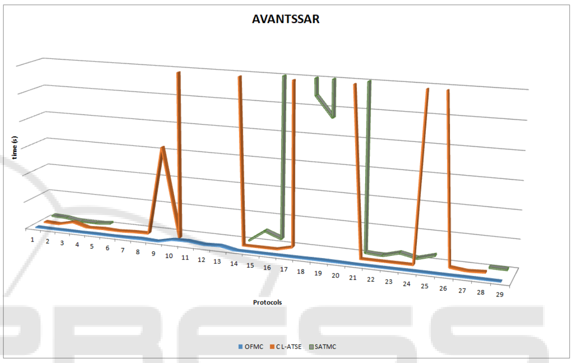

Figure 7: AVANTSSAR's back-ends comparison.

4 CONCLUSIONS AND FUTURE WORK

This paper had presented the results of a quantitative comparison between the validation back-ends from AVISPA and AVANTSSAR, the most mature model checking platforms designed for security protocols/services. AVANTSSAR, being a follow-up of the AVISPA project, the work was motivated by the necessity to identify the fastest of the two tools. Overall, of the three validators common to the two platforms, the AVANTSSAR's OFMC was the fastest, if the total time each validator had finished checking all 29 protocols is considered. The slowest has been AVISPA's SATMC.

But it is interesting to observe that if particular protocols of the 29 analyzed protocols are considered, AVISPA version of CL-Atse had better results that all back-ends of AVANTSSAR: AVISPA's CL-Atse is only surpassed by AVANTSSAR's OFMC for CRAM-MD5, IKEv2-CHILD, IKEv2-DSx, IKEv2-MACx, Kerb-Cross-Realm, Kerb-Forwardable, PBK-fix-weak-auth, and SPEKE. For the other 21 protocols considered, AVISPA's CL-Atse has the best performance.

Likewise, AVISPA's OFMC is faster than AVANTSSAR's OFMC for IKEv2-DS, ISO1, ISO3, and LPD-MSR. The reason is the fact that AVISPA's OFMC had found these protocols to be unsafe, which means that when an attack was found, the computation of the state space was interrupted. On the other hand, AVANTSSAR's OFMC computed the entire state space, as it could not found any attack.

This indicates that there are situations in which using the old AVISPA platform is a better choice than using the new AVANTSSAR, at least for simple, non-distributed communication protocols.

Figure 8 contains a graphical view of the presented conclusions. To not overload the graphics, the data for the two versions of SATMC was not shown (being the slowest back-ends for both tools).

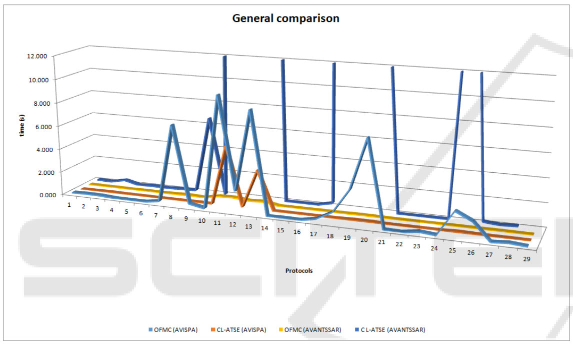

Figure 8: General comparison of back-ends.

It remains to conduct an in depth study of the back-ends and to identify the cause of the observed exceptions. On one hand this will be helpful to identify on which type of security protocols AVISPA should be preferred, and, on the other hand, it would suggest new improvements to the AVANTSSAR platform to surpass AVISPA in all cases.

ACKNOWLEDGEMENTS

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0272 / Avant-garde Technology Hub for Advanced Security (ATLAS), within PNCDI III.

REFERENCES

Alessandro Armando, e. a. (2008). Formal analysis of saml 2.0 web browser single sign-on: breaking the saml-based single sign-on for google apps. In *6th ACM workshop on Formal methods in security engineering*. ACM.

Alessandro Armando, Wihem Arsac, T. A. e. a. (2012). The avantssar platform for the automated validation of trust and security of service-oriented architectures. In *18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 267–282. Springer-Verlag Berlin Heidelberg.

Alessandro Armando, D. Basin, Y. B. e. a. (2005). The avispa tool for the automated validation of internet security protocols and applications. In *17th International Conference on Computer Aided Verification*.

Armando, Alessandro, R. C. and Compagna, L. (2016). Satmc: a sat-based model checker for security protocols, business processes, and security apis. *International Journal on Software Tools for Technology Transfer*, 18(2):187–204.

Blanchet, B. (2001). An efficient cryptographic protocol verifier based on prolog rules. In *14th IEEE Computer Security Foundations Workshop (CSFW)*, pages 82–96. IEEE.

Cremers, C. J. F. (2008). The scyther tool: Verification, falsification, and analysis of security protocols. In *20th International Conference on Computer Aided Verification*.

Dolev, D. and Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208.

Lowe, G. (1998). Casper: a compiler for the analysis of security protocols. *Journal of Computer Security*, 6(1–2):53–84.

Oheimb, D. V. and Modersheim, S. (2011). Aslan++ - a formal security specification language for distributed systems. In *Formal Methods for Components and Objects*. Springer Berlin Heidelberg.

Vigano, L. (2006). Automated security protocol analysis with the avispa tool. *Electronic Notes in Theoretical Computer Science*, 155:61–86.

Vigano, L. (2012). Automated validation of trust and security of service-oriented architectures with the avantssar platform. In *International Conference on High Performance Computing and Simulation (HPCS)*.

Yannick Chevalier, e. a. (2004). A high level protocol specification language for industrial security-sensitive protocols. In *Workshop on Specification and Automated Processing of Security Requirements (SAPS)*. Austrian Computer Society.