VerSecTis – An Agent Based Model Checker for Security Protocols

Demonstration

Agnieszka M. Zbrzezny
University of Warmia and Mazury
Olsztyn, Poland
agnieszka.zbrzezny@matman.uwm.edu.pl

Andrzej Zbrzezny
Jan Dlugosz University
Czestochowa, Poland
a.zbrzezny@ujd.edu.pl

Sabina Szymoniak
Czestochowa University of Technology
Czestochowa, Poland
sabina.szymoniak@icis.pcz.pl

Olga Siedlecka-Lamch
Czestochowa University of Technology
Czestochowa, Poland
olga.siedlecka@icis.pcz.pl

Miroslaw Kurkowski
Cardinal St. Wyszynski University
Warsaw, Poland
m.kurkowski@uksw.edu.pl

ABSTRACT

We present VerSecTis – a new experimental tool for the verification of timed security protocols' (TSP) modelled by Timed Interpreted Systems (TIS). In addition to the TSP's time-independent properties, our tool can also examine the time dependencies of the TSP's executions on which their security depends. The verification method consists of a new TSPs' modelling method and a translation of the reachability problem for TIS to the Satisfiability Modulo Theories problem. We also deliver nineteen TSPs to verify, and we plan to expand the tool with further protocols.

KEYWORDS

Engineering multi-agent systems; innovative applications

ACM Reference Format:
Agnieszka M. Zbrzezny, Andrzej Zbrzezny, Sabina Szymoniak, Olga Siedlecka-Lamch, and Miroslaw Kurkowski. 2020. VerSecTis – An Agent Based Model Checker for Security Protocols . In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

Nowadays many people spend a large part of their professional and private time online, transferring and processing vast amounts of data. The transmission takes place thanks to using communication protocols, where essential items are security protocols (SP) – short algorithms that allow the correct authentication, key distribution, and data integrity [18]. Since they exist a lot of SP, new ones are still developed [3], and sometimes bugs in their schemes are discovered there is a need for SP's properties verification. For this purpose, many methods of formal modelling and verification of SP, and equally many tools [1, 4, 9, 17] using specific methods and models have been created. However, protocols are still evolving, and scientists are seeing more impacts affecting security, and the need to look for more verification ideas continues. A crucial moment in the SP's history was the introduction of time tickets, which, among others, have prevented SP's upon replay attacks. An important aspect that can be studied here is the time relationships between protocol execution times, network delays and timestamps' lifetimes' values.

The tool that we have implemented combines several modern techniques such as modelling Timed Security Protocols (TSP) by Timed Interpreted Systems (TIS), Bounded Model Checking (BMC) and Satisfiability Modulo Theories (SMT). VerSecTis allows testing protocols not only because of the possibility of an attack by unauthorized taking over confidential information, impersonating of honest users by the Intruder or the Intruder's entry into the interior of communication (Man in the Middle). The main advantage is the possibility to examine many time dependencies during executions of a given TSP. The experimental results showed that the appropriate setting of the times of the importance of individual elements (so-called lifetime), depending on the degree of network load (delay) can deprive the Intruder of the opportunity to launch an attack. The user of our software has the chance to explore and experience these relationships.

To our best knowledge, there have been no formal models using agent techniques to model timed security protocols yet. Furthermore, the analysis of time parameters was carried out only in the initial phase of research [8].

2 BACKGROUND

Models. The central part of the input consists of timed security protocol modelled as a time interpreted system [27] with dense time semantics (Fig. 1). In our approach, each agent and an environment in which agents cooperate constitutes a network of timed automata.
Reachability. For a given model M, the existence of an attack means that a particular state is reachable in the model. This state is reachable if and only if some formula holds in the model M.

3 BENCHMARKS

The tool includes the following TSPs' specifications: the timed version (TV) of Needham Schroeder Public Key Protocol (NSPK) [18] and the TV of Lowe's modification of NSPK (NSPK$_L$) [13, 24];

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

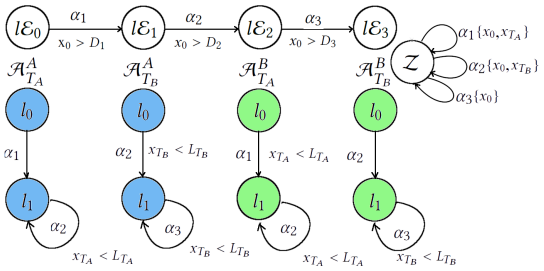

Figure 1: A part of TIS for timed version of Needham Schroeder Public Key Protocol

the TV of Wide Mouthed Frog Protocol (WMF) [6, 26] and the TV of Lowe's modification of WMF (WMF_L) [15]; the TV of Denning-Sacco Protocol (DSC) [10]; the TV of Kao-Chow Protocol (KC) [11, 23]; the TV of Carlsen's Secret Key Initiator Protocol (CSKIP) [7]; the TV of Needham Schroeder Symmetric Key Protocol (NSSK) [18, 26]; the TV of Yahalom Protocol (Y) [6], the TV of Lowe's modification of Y (Y_L) [16], the TV of Paulson's modification of Y (Y_P) [19], and the TV of BAN simplified version of Y (Y_B) [6]; the TV of Woo Lam Pi Protocol (WLP) [25, 26]; the TVs of WLP 1, 2, and 3 (WLP1, 2, 3) [25]; the TV of Andrew Protocol (A) [20] and the TV of Lowe's modification of A (A_L) [14]; MobInfoSec [21, 22].

4 TECHNOLOGY

To verify a chosen TSP, we need carefully prepare the input files. We deliver nineteen benchmarks with prepared files. The directory tree is as follows: each benchmark has its own directory in `protocols` directory, e.g. `nspkt`. In the protocol directory, there are as many subdirectories as many potential attacks are possible, e.g. `Man1, Man2, Lowe1, Lowe2`. Each "attack" directory contains two files: one with the protocol specification and one with a property. The first file e.g. `nspkt_Man1.nta`) is a template file that contains a specification of the environment and all of the agents for specific TSP. The lifetime and delay values are not specified in this file. A file with concrete lifetimes and delays values (e.g. `nspkt_Man1_D2_D4_D8_L3_L10.nta`) is prepared using the template file and the values given by the user (`D1=2, D2=4, D3=8, L1=3, L2=10`). The idea of the modelling method was described in [12, 26], and we adopted it to TIS. The second file (`.efo`) contains a tested formula which is expressed using existential fragment of Computation Tree Logic [2]. This formula expresses the reachability of a local state of an agent and has the form: `EF(propositional_var)`.

The next step is performing BMC [5] algorithm which is implemented in C++ programming language (`bin\smtreach4tis`). The algorithm has three inputs: an `.nta` file, an `.efo` file, and a non-negative natural number k. The reachability problem for MAS modelled by TISs is the question of whether for a given set of target locations, a state with a target location is reachable from some initial state. We assume that a propositional formula describes a set of target locations that express some property. To check the reachability of a state satisfying the property by the BMC method, first, the transition relation of the model is unfolded iteratively to some depth k and encoded as a quantifier-free first-order formula of state variables. Next, the property is translated into an a quantifier-free first-order formula of state variables and the satisfiability of the conjunction of the two above formulae is checked by an SMT-solver. If the conjunction is satisfiable, one may conclude that a path to a target location was found. Otherwise, the value of k is incremented.

All the above algorithms are part of VerSecTis. An implementation along with instructions on how to install the necessary software and the specifications of the tested protocols (`README` file) can be found on GitHub https://github.com/vertisec/VerSecTis/. A movie with an example of usage can be found on Youtube https://youtu.be/nuLVqeqYzP8.

5 EXPERIMENTS

We have implemented our ideas in the VerSecTis tool and then tested on 19 protocols for various time parameters. During this research, we obtained time constraints that show the impact of dependencies between network delays and timestamps lifetimes to the possibility of executing the investigated type of attack upon the protocol. We consider two types of attack: on secrecy (like Lowe's attack) and Man in the Middle (MitM).

Table 1 shows a summary of the series of tests for the lowest lifetime and delay values for which an attack exists. We present time consumption and memory usage for each attack respectively for BMC algorithm and satisfiability testing. The table does not include protocols free of considered types of attacks.

Table 1: Sample experimental results for the protocols with the attacks

Protocol	BMC		SMT		Type of attack
	s.	MB	s.	MB	
NSPK	0.48	2.43	0.9	24.28	MitM
NSPK	0.49	1.61	2.40	30.80	Lowe's
$NSPK_L$	0.49	2.43	2.38	26.11	MitM
DS	0.28	2.57	0.40	23.38	MitM
WLP	0.40	2.54	0.70	29.11	MitM
WLP1	0.54	2.61	1.26	32.54	MitM
WLP2	0.54	2.60	1.26	31.91	MitM
WLP3	0.54	2.61	1.44	32.19	MitM
A	1.80	2.69	12.90	48.93	MitM
A_L	0.98	2.75	3.15	40.44	MitM
MobInfoSec	40.67	16.49	199.01	384.17	MitM

6 CONCLUSIONS

VerSecTis uses an innovative approach to analysing and verifying time properties of security protocols. As a result, it opens the way to study further dependencies. We can analyse subsequent time aspects such as generation, decryption or data encryption. We can add more complex protocols, analyse multiple sessions, and what is essential after adding the parser allow the user to examine their protocols from outside the shared library.

ACKNOWLEDGMENTS

The project financed under the program of the Polish Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in the years 2019 - 2022 project number 020/RID/2018/19, the amount of financing 12,000,000.00 PLN.

REFERENCES

[1] A. Armando and et.al. 2005. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In *In Proc. of CAV 17th Int. Conf., 2005, Edinburgh, Scotland, UK,*. 281–285.

[2] C. Baier and Joost-P. Katoen. 2008. *Principles of model checking*. MIT Press.

[3] M. Bartlomiejczyk, I. El Fray, and M. Kurkowski. 2019. Multifactor Authentication Protocol in a Mobile Environment. *IEEE Access* 7 (2019), 157185–157199. https://doi.org/10.1109/ACCESS.2019.2948922

[4] D. A. Basin, C. Cremers, and C. A. Meadows. 2018. Model Checking Security Protocols. In *Handbook of Model Checking*. 727–762.

[5] A. Biere. 2009. Bounded Model Checking. In *Handbook of Satisfiability*. 457–481. https://doi.org/10.3233/978-1-58603-929-5-457

[6] M. Burrows, M. Abadi, and R. Needham. 1990. A Logic of Authentication. *ACM Trans. Comput. Syst.* 8, 1 (1990), 18–36.

[7] U. Carlsen. 1994. Optimal Privacy and Authentication on a Portable Communications System. *Operating Systems Review* 28 (07 1994), 16–23.

[8] R. Corin, S. Etalle, P. H. Hartel, and A. Mader. 2007. Timed analysis of security protocols. *Journal of Computer Security* 15, 6 (2007), 619–645.

[9] C. Cremers and S. Mauw. 2012. *Operational Semantics and Verification of Security Protocols*. Springer. https://doi.org/10.1007/978-3-540-78636-8

[10] D. E. Denning and G. M. Sacco. 1981. Timestamps in Key Distribution Protocols. *Commun. ACM* 24, 8 (1981), 533–536.

[11] R. Chow I.-Lung Kao. 1995. An efficient and secure authentication protocol using uncertified keys. *ACM SIGOPS Operating Systems Review* 29(3) (1995), 14–21.

[12] M. Kurkowski and W. Penczek. 2012. Applying Timed Automata to Model Checking of Security Protocols. In *Handbook of Finite State Based Models and Applications*. 223–254. https://doi.org/10.1201/b13055-12

[13] G. Lowe. 1995. An Attack on the Needham-Schroeder Public-Key Authentication Protocol. *Inf. Process. Lett.* 56, 3 (1995), 131–133.

[14] G. Lowe. 1996. Some new attacks upon security protocols. In *Proceedings 9th IEEE Computer Security Foundations Workshop*. 162–169.

[15] G. Lowe. 1997. *A Family of Attacks upon Authentication Protocols*. Technical Report.

[16] G. Lowe. 1998. Towards a completeness result for model-checking of Security Protocols. *Journal of Computer Security* 7, 96 – 105.

[17] C. A. Meadows. 2003. Formal methods for cryptographic protocol analysis: emerging issues and trends. *IEEE Journal on Selected Areas in Communications* 21, 1 (2003), 44–54. https://doi.org/10.1109/JSAC.2002.806125

[18] R. Needham and M. Schroeder. 1978. Using Encryption for Authentication in Large Networks of Computers. *Commun. ACM* 21, 12 (1978), 993–999. https://doi.org/10.1145/359657.359659

[19] L. C. Paulson. 2001. Relations between secrets: Two formal analyses of the Yahalom protocol. *Journal of Computer Security* 9, 5 (2001), 197–216.

[20] M. Satyanarayanan. 1989. Integrating security in a large distributed system. *ACM Trans. Comput. Syst.* 7 (1989), 247–280.

[21] O. Siedlecka-Lamch, I. El Fray, M. Kurkowski, and J. Pejas. 2015. Verification of Mutual Authentication Protocol for MobInfoSec System. In *Computer Information Systems and Industrial Management - 14th IFIP TC 8 International Conference, CISIM 2015, Warsaw, Poland, September 24-26, 2015. Proceedings*. 461–474.

[22] O. Siedlecka-Lamch, S. Szymoniak, and M. Kurkowski. 2019. A Fast Method for Security Protocols Verification. In *Computer Information Systems and Industrial Management - 18th International Conference, CISIM 2019, Belgrade, Serbia, September 19-21, 2019, Proceedings*. 523–534.

[23] S. Szymoniak. 2018. KaoChow Protocol Timed Analysis. In *Advances in Soft and Hard Computing, Proceedings of the International Conference on Advanced Computer Systems, ACS 2018, Miedzyzdroje, Poland, 24-26 September 2018*. Springer, 346–357.

[24] S. Szymoniak, O. Siedlecka-Lamch, and M. Kurkowski. 2018. On Some Time Aspects in Security Protocols Analysis. In *Computer Networks - 25th International Conference, CN 2018, Gliwice, Poland, June 19-22, 2018, Proceedings*. 344–356.

[25] T. Y. C. Woo and S. S. Lam. 1994. A Lesson on Authentication Protocol Design. *SIGOPS Oper. Syst. Rev.* 28, 3 (1994), 24–37.

[26] A. M. Zbrzezny, S. Szymoniak, and M. Kurkowski. 2019. Efficient Verification of Security Protocols Time Properties Using SMT Solvers. In *In Proc. of CISIS'19, Spain, 2019,*. 25–35. https://doi.org/10.1007/978-3-030-20005-3_3

[27] A. M. Zbrzezny and A. Zbrzezny. 2017. Simple SMT-Based Bounded Model Checking for Timed Interpreted Systems. In *Rough Sets - International Joint Conference, IJCRS 2017, Olsztyn, Poland, July 3-7, 2017, Proceedings, Part II*. 487–504. https://doi.org/10.1007/978-3-319-60840-2_35