A Review on the Verification Approaches and Tools used to Verify the Correctness of Security Algorithms and Protocols

Mohammed Abdulqawi Saleh Al-humaikani[1], Lukman Bin Ab. Rahim[2]
Department of Computer & Information Science, Universiti Teknologi Petronas Bandar Seri Iskandar, Malaysia

***Abstract*—Security algorithms and protocols are typical essential upgrades that must be involved within systems and their structures to provide the best performance. The protocols and systems should go through verification and testing processes in order to be more efficient and accurate. In the testing of software, traditional methods are used for accuracy check-up. However, this could not fulfill the measurement of all the testing requirements. The usage of formal verification approaches in checking security properties considers their best environment to be applied. The available literature discussed several approaches on developing the most robust formal verification methods for addressing and analyzing errors that face systems. This could be during the implantation process, unknown attacks, and nondeterministic adversary on the security protocols and algorithm. In this paper, a comprehensive review of the main formal verification approaches such as model checking and theorem approving has been conducted. Moreover, the use of verification tools was briefly presented and explained thoroughly. Those formal verification methods could be involved in the design, redesign of security protocols, and algorithms based on standards and determined sizes that is decided by these techniques' analysis. The critical analysis of the methods used in verifying the security of systems showed that model checking approaches and its tools were the most used approaches among all the reviewed methods.**

Keywords*—Security algorithms; security protocols; formal verification approaches; model checking; theorem proving

I. Introduction

The number of technology users has increased rapidly on a daily basis all over the globe. This requires developers and researchers to conduct and invent algorithms and protocols that can provide a high level of security. For instance, in [1] stated that security and protection level that can be involved with a variety of information technology platforms include users and users data. Security in the form of security algorithms [2] and protocols [3] can achieve security purposes and provides authentication, data confidentiality, secrecy, and secured communication. One of the well-known security, cryptographic algorithms, and protocols are RSA, SSL/TLS [4]. These protocols and algorithms can be used to provide a level of security and confidentiality for different types of technology's users. However, the level of security offered will be associated with the risk of algorithms / protocols failures, third man attack, or other types of risks that can face users and applications. In [5] mentioned that there are different methods able to ensure the effective performance of these protocols and algorithms by using testing, simulators, or formal verification approaches. The use of simulations to test or measure the security protocols and algorithms' properties is not fully trusted as stated in [6]. Meanwhile, there are some challenges in security protocols and algorithms introduced in the work done by Rosenberg [7] such as the increase number of cyber-attacks. Then, the measurement of results produced by simulators are not the proper tool to evaluate or analyse the properties and in verifying the correctness. Therefore, formal verification methods are considered as one of the approaches that can be used to verify the correctness of high-technologies and complex systems which include security algorithms and protocols [8].

To the authors' best of knowledge there is no available critical review on the methods used for the security verification protocols and algorithms. In this paper, a comprehensive review was conducted in order to describe, compare, and analyse the two formal verification approaches of security and their tools. This included model checking and theorem proving approaches. Hence, this paper would provide a platform for individuals and companies working on the security checking during the early stages of building the security systems.

II. Formal Verification Approaches of Security Aspects

This section provides a review on the selected formal verification approaches that can be used in security aspects. These approaches aim to verify two different types of security methods that are security algorithms and security protocols. Besides, those approaches were classified into their uses and purposes. Among the most popular security verifications methods, two methods can be used to verify the correctness of security algorithms and protocols. These two methods are symbolic model checking [9] and theorem approving methods [10]. In the aforementioned formal verification methods, different tools and approaches are used by the researchers in their experiments. The following sections will explain these two approaches with their tools.

A. Verification Approaches for Security Algorithms

In this subsection, the verification approaches that can be applied on security algorithms and a check on their validity are explained. Besides, the selected verification process for security algorithms was reviewed. Finally, all the selected verification approaches were compared in terms of their differences and usages.

1) Overview on the verification approaches for security algorithms: There are two formal approaches found in the literature to verify the correctness of the security algorithms which are symbolic model checking and theorem approving methods. First, the symbolic model checking is basically a model checking that is implemented as a symbolic representation. Symbolic model checking contains two main approaches; binary decision diagrams (BDD)-based model checking [11] and SAT-based model checking [12]. Symbolic model checking is considered as one of the efficient formal verification approaches that can be implemented in software and hardware systems. In [13, 14] stated that using model checking approach requires to build a model of the system from scratch that will have the desired properties to be verified by using model checkers. Otherwise, the model checker will not be useful to check the absence of error within systems without their models.

Secondly, theorem proving is another approach that can verify the correctness of systems in logic mathematical formulas. The properties of the system are formed as mathematical formulas and solved by finding the proof of presuppositions from the system. Lastly, these selected approaches were reviewed by other authors and will be discussed thoroughly in the next section.

III. Review on the Selected Verification Approaches for Security Algorithms

Schnepf et al. [15] proposed an automated alternative method that can support the verification process of the security in chains at the early stages. The proposed strategy considered control and data planes to be analysed, which contained different security algorithms formed as security functions in chains. First, the security specification of chains were translated into formal models to verify them using verification methods. The security specification translated into formal models using a presented method called Synaptic checker. Two categories supported the verification process and the first category was SMT solver based. The second category was model checking, at which the properties of the security functions were translated to a finite state machine to elaborate on the number of states. After that, the security properties were expressed in the form of temporal logic CTL. Together the FSM states with temporal logic CTL could be verified using the model checking nuXmv. The main benefit of this study was the ability to propose a method that designed as a packet of SDN language. This made it easy to be extended for further formal languages that could check simple invariants properties. Finally, the proposed method needed more improvements to support more complex security functions. Slind [16] presented the verification process of the Rijndael security algorithm. Rijndael block cipher is an algorithm that can encrypt and decrypt data with key sizes of 128, 192, and 256 bits and number of the rounds based on the block sizes. The security algorithm was approved using theorem proving method called HOL. The results showed that the security algorithm was easy to be verified because of the simplicity of its specification. The main advantage of this study was the simplicity of the algorithm when it was coded into SML language, which can be easier to be understood and taught. Besides, the absence of verifying the security rules in case of hybrid security algorithms, or how the difficulties that can face verification process.

Moreover, Keerthi et al. [17] have provided an overview of using formal verification approaches to verify the correctness in the implementation of several security algorithms. The author used formal verification approaches differently based on the desired properties of IoT security algorithms. Cryptographic algorithms, such as ECC and RSA are used as a case study to verify their implementation by using model checking. The cryptographic algorithms were translated to SAT-based model checking in order to verify it by using a model checking approach CBMC (ANSI-C Bounded Model Checking). CBMC checks the verification of cryptographic algorithms with the help of SAT solver to specify the correctness in counter-example whether they failed or verified by showing the fail part, in case of any failure. The benefits of this work were the ability to prove the correctness of two of the most used cryptographic algorithms, by approving the ability of model checking to verify different extendable cryptographic algorithms. On the other hand, it was better to apply multiple cryptographic algorithms to be proved and to confirm the capability of verification approaches.

Furthermore, Arpit et al. [18] have verified one of the cryptographic algorithms called the ElGamal algorithm. ElGamal cryptographic is asymmetric keys based for encryption and decryption. ElGamal cryptographic provided a public key (PK) and secret key (SK) to add enhanced security on the exchanged data between the communicator parties. The author followed specific steps to represent the cryptographic algorithm safety properties and to get them verified. In addition, a transition system was built using a Kripke structure to describe the transitions system of the cryptographic algorithm. The Kripke structure was considered as a model system for the cryptographic algorithm and translated into the form of logic temporal language (LTL). The LTL defined the behaviour or the properties of the model system in the form of formulas. After that, the model system was verified using the model checking. The author has represented all the steps required for LTL syntax and behaviours in the case for any future studies for other cryptographic algorithms. Moreover, the usage of the model checking with the help of LTL showed the simplicity of verifying cryptographic algorithms using tools and formulas together. The drawbacks in this study were that no statistical analysis for future usages was provided which is considered a disadvantage. Chen et al. [19] proposed a formal verification methodology to verify the correctness of a cryptographic algorithm called Curve25519. Curve25519 is high-speed elliptic-curve cryptography that computed up to 18-bit of security keys. The authors have verified the Curve25519 using hybrid methodology consisted of SMT solving and theorem proving tools assistant. The authors believed that this approach could be computed or verified in low-level optimisation for actual cryptographic algorithms and protocols. The tools that have been used for verification were portable assembly, qhasm, the Boolector SMT solver, and the Coq proof assistant. Curve25519 wrote in an assembly language called qhasm using portable assembly qhasm that saved development time for assembly software. This study presented new verification approach methodology that consisted different tools, which can offer low-level optimisation to verify real-world cryptographic algorithms. The disadvantage of using this methodology was because of the translation from a different language into assembly language qhasm, which cannot be accurate for most of the times.

A. Verification Approaches for Security Algorithms

In this subsection, the verification approaches used to assist in the security protocol verification and correctness are demonstrated. Besides, selected verification approaches for security protocols were reviewed and all the chosen verification approaches were compared based on their different usages. Formal verification methods / approaches that are usually used to verify the correctness of security algorithms was found to be similar to the formal verification approaches used to verify security protocols [20,21]. However, different tools are used in order to verify the security protocol properties such as SAMTC. The following sub-section briefly highlights the main security approaches of both theorem proving and model checking.

1) Review of the selected verification approaches for security protocols: Armando et al. [22] presented a formal verification approach called SAT-based Model-Checker (SATMC) to verify the correctness of critical security systems. This included security protocols business processes and security Application programming interfaces (APIs). SATMC has involved different verifying security protocols such as the security assertion markup language (SAML) 2.0, single sign-on (SSO) protocol, and OpenID. SATMC is SAT-based model checking that uses LTL formulas to format the properties of the requested protocol for verification purposes with the help of SAT-solver. The authors used ASLan language to model and identify the problems and errors that could face any application with security protocols. This tool was successfully applied in verifying variety of security protocols that is considered as one of the useful tools in supporting model checking methods during the verification process. However, this tool lack to test security aspect sectors and it did not find any statistical analysis for optimising this approach. Paolo and Riccardo [23] have verified the properties of a security protocol called Needham-Schroeder public key authentication protocol. Needham-Schroeder public key authentication protocol was invented in 1978 and it is considered as one of the well-known security protocols. This protocol aimed to provide a level of security by using a trusted key server and public key to establish mutual secured authentication. The authors used the spin model checking approach in this study to verify the security properties of the protocol and to find any known attacks. However, the authors suggested developing a Promela model for the cryptographic protocol to help with the verification of the spin model checker approach. The developed model was built to identify the rules and behaviour of the protocol that need to be checked using the model checker. Lastly, the results show that there were no possible additional attacks that could be detected on the protocol rather than the well-known attack that is called the Lowe’s attack. This study investigated all the details to explain the procedure of the verification for the security protocol, which is considered as an excellent reference for future usages. The drawbacks of this study was the lack to provide a proper comparison between all the security properties such as time synchronisation, secrecy, and equivalences. Moreover, Schaller et al. [24] have proposed a formal verification model that can verify the cryptographic properties of three different security protocols on the network. The three protocols that have been verified in this study were; authenticated ranging, ultrasonic distance-bounding, and TESLA broadcast authentication. A model was formalised with Isabelle and higher-order logic (Isabelle/HOL) theorem proving methods. After that, the rules were defined and modeled to verify the physical characteristics that were found on each cryptographic protocol. The proposed formal model was able to capture the relay attacks, broadcast authentication, and other physical properties that were related to each security protocol. The proposed formal method enabled the capturing and verification for variety of cryptographic properties for three protocols, that are considered as helpful references for future works. Unfortunately, the study has not made any statistical comparison with other formal methods at the same level of functionalities. Cremers [25] had presented an overview of one of the efficient verification tool based on the graphics user interface (GUI) that can verify different security protocols. It is a model checking based tool that uses security protocol description language (SPDL) as an input language to write protocols that need to be analysed. The Scyther tool can identify the possible security protocols properties, attack and generates graphs for each attack based on proposed claims. The properties and attacks of the security protocols are analysed with the assistance of a backward search algorithm method that provides a set of infinite pattern traces to approve the correctness of events that must occur. The advantages of this study was its ability to explain the functionality of the Scyther verification tool in short form.

Finally, El-Madhoun et al. [26] have proposed a new security protocol used in near field communication (NFC) payment systems instead of Europay Mastercard Visa (EMV) security protocols. The new security protocol helped to overcome the vulnerabilities found in the EMV protocol. This protocol was based on an online communication and asymmetric cryptography that allow the connection with an authentication server to execute security functions for NFC transactions. The Scyther tool was used to verify the correctness of the proposed security protocol based on specific claims for verification purposes. The Scyther tool is a model checking-based tool and it supports infinite of traces that help to provide the correctness of the requested claimed. Security protocol description language (SPDL) was also used in this study to write the proposed protocol model into the Scyther tool and this helped to scale the number of claims and tests which simplified the verification procedure. The inputs of the claims provided for the verification were authentication and confidentiality. Then, Scyther will approve the claims and check whether the protocol holds the claims or not. The results showed that the protocol had successfully passed the test and the tool approved the holdings of the claims. In conclusion, this study was able to provide a real example of using the Scyther model checking tool, but there was no statistical comparison of using different verification methods to verify the correctness of the proposed protocol.

IV. Verification Tools

There are various formal verification tools used to verify different security systems. However, an overview is listed below for some of the most popular formal verification tools that have been studied by the authors in previous discussed sections to verify the correctness of different security algorithms and protocols:

New symbolic model verifier (NuSMV) [27] is an extensive model checker of symbolic model verifier (SMV) and it is a formal verification and reliable tool to verify finite state systems. NuSMV is based on binary decision diagrams (BDDs). It interacts with the user by using a textual interface and it needs to implement or input the properties of finite state systems by using computational tree logic (CTL) or logic temporal language (LTL) formulas to help with the verification procedure. Some researchers used this tool as a formal verification tool to verify the properties of security protocol as in Panti et al. [28] who used the NuSMV model checker to verify the security properties of the Kerberos protocol [29]. The authors built a transition states diagram model to represent the Kerberos protocol flow in an understandable method. The security properties of the protocol were expressed by the temporal logic CTL in order to be verified in the NuSMV model checker. Besides, Panti et al. [30] have proposed to use the symbolic model checker NuSMV to verify the correctness of secure electronic transaction (SET) protocol. This protocol provides secure transactions process for the users in the open networks. The protocol was modeled as a transition diagram model and the security properties in the model were described to be verified in the NuSMV model checker. The results showed that there were two different possible attacks found during the verification process and this allowed attackers to attack users of this protocol. Moreover, Massimo and Fausto [31] used NuSMV model checker to be part of the verification process of Andrew Protocol [32]. The author built a model for the protocol by using multiagent finite state machine (MAFSM) to reform the protocol as finite states and converted it to NuSMV language to be verified.

C-bounded model checker (CBMC) [33] is one of the model checkers used to verify the security protocols. It needs two inputs in order to process the verification. The first input is that the system or the program needs to be verified and the second input is the formal specific properties that need to be verified as well. Various studies used this tool in their research experiments. For instance, Keerthi et al. [17] used CBMC to verify the implementation of cryptographic algorithms such as ECC and RSA. CBMC checks the verification of cryptographic algorithms with the help of satisfiability (SAT) solver to specify the correctness in counter-example. The result of the verification process will show two outputs either pass or fail and in the case of failure, the failed parts will be expressed. Sosnovich et al. [34] have proposed a formal approach that helped to automatically discover any security vulnerabilities or attacks that can be found in the network protocol OSPF. The authors modelled the protocol on a concrete model form to expose the desired property to the model checker CBMC. Attacks occurred on OSPF were detected by using CBMC. Satisfiability modulo theories (SMT) [35] solvers use different types of methods to reason about the built theories on SMT solvers. Those solvers check in specific theory to solve satisfiability problems. Schnepf et al. [15] proposed an algorithm that assists to verify the security in chains. The algorithm went through two verification processes; one was model checking and the other one was SMT solvers. The SMT solver was used to check the satisfiability of the model that was designed for the proposed algorithm in order to verify the correctness of the algorithm’s constraints. The algorithm was modelled by using SMTlib input language of SMT solvers for verification purposes. Chen et al. [19] have presented a formal verification approach to verify the correctness of Curve2551 cryptographic algorithm. The formal verification approach consisted of Boolector SMT solver and theorem proving tools assistant such as portable assembly (qhasm). NuXmv is an evolution of NuSMV model checker and it is a new symbolic model checker that works on checking finite and infinite-state transition systems. It was extended from NuSMV with new data types. It provides advanced model checking techniques based on the SMT [36]. Chen et al. [19] verified the correctness of the cryptographic algorithm Curve2551 by using the model checker NuXmv with other verification approaches. The authors expressed the security properties of Curve2551 in the temporal logic CTL to be verified. However, Guanjie and Shigong [37] had verified the properties of one of the famous protocols of radio-frequency identification (RFID) protocol called hash-lock protocol. NuXmv model checker was used to check the security properties of the hash-lock protocol. A model was built for the hash-lock protocol and attack model to help in verifying the hash-lock protocol.

Spin tool [38] is a verification tool that is used for verifying the concurrent systems. It is considered as one of the robust model checkers used to verify security protocols. Paolo and Riccardo [23] have proposed to use the spin model checker to verify the security properties of Needham-Schroeder public key authentication protocol. The author built a model in the format of a Promela model for the cryptographic protocol to verify the protocol in the spin model checker. Besides, Li et al. [39] have used the spin model checker tool to verify one of the Ultralight-weight authentication protocols. Rfid authentication protocol with permutation (RAPP) is the proposed ultralight-weight authentication protocol that needs to be verified using the spin model checker. Modelling the ultralight-weight authentication protocol by using a protocol abstract modelling method to verify the authenticity and consistency of the protocol was suggested in this study. In addition, Scyther is a model checking tool that verifies the correctness of the claimed requests with the help of an infinite set of traces. The tool can help the protocol to suspect and analysis any attacks that can occur and identify the protocol behaviour [40]. Moreover, El Madhoun et al. [26] have verified a new proposed security protocol that was used in NFC payment systems instead of

Europay Mastercard Visa (EMV) security protocols. A model was built for the new protocol based on security protocol description language (SPDL) to input it into Scyther tool for verification purposes. The inputs of the claims that were provided for verification are authentication and confidentiality. Cas [25] proposed a new verification approach that was implemented on Scyther tool. The refinement proposed algorithm can verify and provide unbounded verification, falsification, and characterisation correctness of different type of security protocols such as TLS protocols. SATMC or SAT-based model checker [41, 42, 43] is a new formal verification approach that has a flexible platform of SAT-based checker. SATMC can verify the correctness of security sensitive protocols. It proposed a translation method that can help model checker to construct the required security protocols for verification purposes. The proposed method was implemented in SATMC checker using the features of SATMC combined with the proposed method. Besides, Armando et al. [22] had used SATMC as a formal model checker to verify different critical systems such as security systems in business processes and security APIs that include security protocols. The security protocols; SAML 2.0 single sign-on (SSO) protocol and OpenID were used as a case study. Furthermore, the ASLan language was used to model and input the properties of the security protocols into the model checker SATMC and NuSMV.

V. Discussion

In the previous sections, the described formal verification methods and tools have been grouped into two main groups; symbolic model checking, and theorem proving. Several papers have been reviewed based on the formal verification methods and tools, and further elaborated them in Tables I, II and III. Tables I and II explain the most used formal approaches among the two well-known described approaches. It can be observed that model checking methods were adopted by many researchers and it has the highest number of studies. This shows that model checking is the most acceptable tool in verifying security algorithm and protocols with a percentage of 70% compared with other tools as stated in Fig. 1. The percentage of studies that used theorem proving is considered low as this method is hard to be used. Theorem proving usually involves complicated formulas and multiple processes in order to get it done and its percentage was 30% which is considered acceptable as show in Fig. 1.

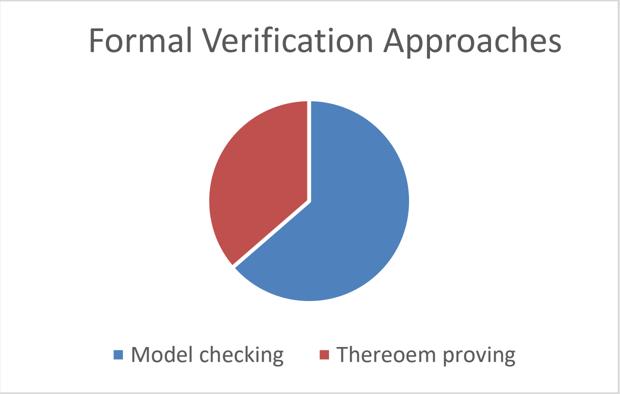

Fig. 1. Formal Verification Approaches.

Table III shows the most used tools that have been suggested by various researchers in order to help in the process of verifying security protocols and algorithms. The tools that have been used were NuSMV model checker that was used by two different researchers in the past sections. Moreover, NuSMV model explore the properties of system model using LTL and CTL. However, different researchers used different tools in the past section to assist the verification of their approaches or systems. They used spin tool, SATMC checker tool, NuXmv updated model checker of NuSMV, CBMC checker, Scyther tool and SMT solvers.

TABLE I. Comparison of Symbolic Model Checking Approaches

Ref.	Main idea	Formal verification method
[15]	A method that was designed as a packet of SDN language which makes it easy to be extended for further formal languages to check simple invariants properties	The security properties were expressed to form temporal logic CTL. The FSM states together with temporal logic CTL could be verified using the model checking nuXmv.
[17]	An overview of using formal verification approaches to verify the correctness in the implementation of serval security algorithms such as ECC and RSA.	Verified using model checking approach CBMC (ANSI-C Bounded Model Checking).
[18]	Verified one of the cryptographic algorithms called the ElGamal algorithm	Built a transition system using a Kripke structure to describe the transitions system of the cryptographic algorithm and translated it into the form of logic temporal language (LTL). This was also checked using NuSMV
[22]	Verified the correctness of critical security systems that included security protocols business processes and security APIs.	Used ASLan language to model and identify the problems and errors that could face the application that included security protocols and was verified using SATMC mode checker.
[23]	Verified the properties of a security protocol called Needham-Schroeder public key authentication protocol.	Developed a Promela model for the cryptographic protocol in order to help with the verification in the spin model checker approach.
[25]	Presented an overview of one of the most efficient GUI based tools that can verify different security protocols.	The properties and attacks of the security protocols were analysed with the help of a backward search algorithm method that provided a set of infinite pattern traces to approve the correctness of events by using the Scyther model checker
[26]	Proposed a new security protocol that was able to overcome the vulnerabilities found in the EMV protocol.	Used protocol description language (SPDL) to write the proposed protocol model into the Scyther tool and helped to scale the number of claims and tests.

TABLE II. COMPARISON OF THEOREM PROVING APPROACHES

Ref.	Main Idea	Formal Verification Method
[15]	A method that was designed as a packet of SDN language. This make it easy to be extended for further formal languages that can check simple invariants properties	Uses SMT solver to check the satisfiability of the model that made out the proposed algorithm in order to verify the correctness of the algorithm's constraints. The algorithm modelled by using SMTlib input language of SMT solvers for verification purposes.
[16]	Verified the security properties of the Rijndael security algorithm.	Approved the security algorithm by using theorem proving method called HOL.
[19]	Verified the correctness of a cryptographic algorithm called Curve25519.	Verified the Curve25519 by using hybrid methodology that consisted of SMT solving and theorem proving tools assistant, qhasm, and Boolector SMT solver.
[24]	Verified the cryptographic properties of three different security protocols on the network, such as authenticated ranging, ultrasonic distance-bounding, and TESLA broadcast authentication.	Formed models of the protocols with Isabelle/HOL formalization theorem proving methods. Besides, the rules were defined and modeled based on the physical characteristics found on each cryptographic protocol.

TABLE III. COMPARISON OF THEOREM PROVING APPROACHES

Ref.	Tool	Formal verification type
[28,30,31]	NuSMV	Model checking
[17,32]	CBMC	Model checking
[15,19]	SMT solvers	Theorem proving
[16,37]	NuXmv	Model checking
[23,39]	The spin checker	Model checking
[24,26]	Scyther tool	Model checking
[22,35]	SATMC	Model checking

VI. FUTURE WORK

This paper illustrated two main verification methods to verify the correctness of security algorithms and protocols. The two main methods are model checking and theorem proving. Moreover, this paper presented the verification tools that used along with verification methods; tools such as, NuSMV, CBMC and SMT solvers. Future work can be done on verification methods that can be used to verify the different types of algorithms and protocols.

VII. CONCLUSION

This paper presented a comprehensive literature review of the most used formal verification methods and approaches used to verify and analyse the correctness of security properties for the cryptographic protocols and algorithms. Various studies were extensively reviewed that involved the security protocols, algorithms and formal verification methods. Those studies were classified based on two main formal verification types; model checking and theorem proving. The reviewed literature was explained and analyzed in general based on the formal verification approaches, proposed approaches, builds models for verification, advantages, and disadvantages.

The results showed that majority of the reviewed studies used model checking tools and approaches to verify their works and experiments. NuSMV is one of the formal verification tools that was used frequently by many researchers and this verification tool used LTL and CTL to format models for verification process. Therefore, the result of the discussion of the reviewed studies showed that there is still a gap in security aspects that could be studied intensively in the future studies.

ACKNOWLEDGMENT

The authors acknowledge the generous financial support provided by Universiti Teknologi PETRONAS (UTP) using the Graduate Assistantship Scheme (GA). The authors would like to express their gratitude and appreciation to UTP for the generous support.

This research presented in this paper is part of a research project funded under the Fundamental Research Grant Scheme (FRGS) - FRGS/1/2018/ICT01/UTP/02/2.

REFERENCES

[1] Y. Sun, J. Zhang, Y. Xiong, G. Zhu, "Data security and privacy in cloud computing", Int. J. Distrib. Sens. Networks, Vol. 10, No. 7, 2014.

[2] G. Singh, S. Supriya, "A study of encryption algorithms (RSA, DES, 3DES and AES) for information security", Int. J. Comput. Appl., Vol. 67, No. 19, 2013.

[3] A. W. Roscoe, "Intensional specifications of security protocols", Proceedings of the 9th IEEE Computer Security Foundations Workshop, 1996.

[4] T. Dierks, E. Rescorla, "The transport layer security (TLS) protocol version 1.2", RFC 5246, August 2008.

[5] P. Bjesse, "What is formal verification?", ACM SIGDA Newsletter, Vol. 35, No. 24, 2005.

[6] Q. Zhang , L. Cheng , R. Boutaba, "Cloud computing: state-of-the-art and research challenges", J. Internet Serv. App, Vol. l, No 1, 2010.

[7] J. Rosenberg, "Security in embedded systems", Rugged Embed. Syst. Comput. Harsh Environ., Vol. 3, No. 3, 2017.

[8] T. Coffey, R. Dojen, T. Flanagan, "Formal verification: an imperative step in the design of security protocols", Comput. Networks, Vol. 43, No. 5, 2003.

[9] J. Edmund , M. Clarke , O. Grumberg , D.A. Peled , "Model Checking", MIT Press, 1999.

[10] S.A. Cook, "The complexity of theorem-proving procedures", Proceedings of the third annual ACM symposium on Theory of computing, 1971.

[11] F. Al-Saqqar , J. Bentahar , K. Sultan , W. Wan , E. K. Asl, "Model checking temporal knowledge and commitments in multi-agent systems using reduction", Simul. Modell. Pract. Theory, Vol. 51, No. 2, 2015.

[12] W. Farn, "Formal verification of timed systems: a survey and perspective", Proc. IEEE, Vol. 92, No. 8, 2004.

[13] S.P. Miller, M.W. Whalen, D.D. Cofer, "Software model checking takes off", Commun. ACM, Vol. 53, No. 2, Feb. 2010.

[14] Alur, Rajeev, D. Dill, "Automata for modeling real-time systems", Proc. International Colloquium on Automata Languages and Programming, 1990.

[15] N. Schnepf, R. Badonnel, A. Lahmadi, S. Merz, "Automated verification of security chains in software-defined networks with synaptic", Proc. IEEE Conference on Network Softwarization (NetSoft), 2017.

[16] K. Slind, "A verification of rijndael in HOL", NASA conference publication, 2002.

[17] K. Keerthi, I. Roy, A. Hazra, and C. Rebeiro, “Security and fault tolerance in internet of things”, Springer International Publishing, 2019.

[18] Arpit, A. Kumar, “Verification of elgamal algorithm cryptographic protocol using linear temporal logic” 2011 Int. Conf. Multimed. Technol. IEEE, 2011.

[19] Y. Fang. Chen et al., “Verifying Curve25519 Software”, Pro. of the 2014 ACM SIGSAC Conference on Computer and Communications Security, 2014.

[20] Marrero, Will, E. Clarke, J. Somesh, “Model checking for security protocols”, Carnegie-mellon univ pittsburgh pa dept of computer science, No. CMU-CS-97-139, 1997.

[21] Subramanyan, Pramod, S. Ray, S. Malik, "Evaluating the security of logic encryption algorithms", In Proc. 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2015.

[22] A. Armando, R. Carbone, L. Compagna, “SATMC: a SAT-based model checker for security protocols, business processes, and security APIs”, Int. J. Softw. Tools Technol. Transf., Vol. 18, No. 2, 2016.

[23] P. Maggi and R. Sisto, “Using SPIN to verify security properties of cryptographic protocols”, International SPIN Workshop on Model Checking of Software, 2002.

[24] P. Schaller, B. Schmidt, D. Basin, and S. Capkun, “Modeling and verifying physical properties of security protocols for wireless networks”, Proc. IEEE Comput. Secur. Found. Symp., 2009.

[25] C. Cremers, “The Scyther tool: verification, falsification, and analysis of security protocols”, International Conference on Computer Aided Verification, Vol. 5123, 2008.

[26] N. El-Madhoun, F. Guenane, G. Pujolle, “An online security protocol for NFC payment: Formally analyzed by the scyther tool”, Proc. of the 2nd Conf. Mob. Secur. Serv., 2016.

[27] A. Cimatti, E. Clarke, F. Giunchiglia, M. Roveri, “NUSMV: A new symbolic model checker,” Int. J. Softw. Tools Technol. Transf., Vol. 2, No. 4, 2000.

[28] M. Panti , L. Spalazzi , S. Tacconi, ”Using the NuSMV model checker to verify the kerberos protocol”, Proc. of the International Conference on Simulation and Multimedia in Engineering Education, Vol. 34, 2002.

[29] Abdelmajid, T. Nabih, et al., "Location-based kerberos authentication protocol", Proc. of the 2010 IEEE Second International Conference on Social Computing, 2010.

[30] M. Panti, L. Spalazzi, S. Tacconi, “Verification of security properties in electronic payment protocols”, in Proc. ACM SIGPLAN IFIP WG, 2002.

[31] M. Benerecetti, F. Giunchiglia, “Model checking security protocols using a logic of belief”, International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Vol. LNCS, No. 1785, 2000.

[32] M. Burrows, M. Abadi, R.M. Needham, “A logic of authentication”, ACM Transactions on Computer Systems, Vol. 8, No. 1, 1990.

[33] C. Baier, C.T. Eds, “Tools and algorithms for the construction and analysis of systems”, 21st International Conference TACAS, 2015.

[34] O. Grumberg, A. Sosnovich, G. Nakibly, “Finding security vulnerabilities in a network protocol using parameterized systems”, In Proceedings of CAV, 2013.

[35] O. Demir, W. Xiong, F. Zaghloul, J. Szefer, “Survey of approaches for security verification of hardware / software systems”, IACR Cryptol. ePrint Arch., 2016.

[36] R. Cavada et al., “The NUXMV symbolic model checker”, International Conference on Computer Aided Verification, Vol. 8559, 2014.

[37] G. Yuan, S. Long, “Formal verification of RFID protocols using nuXmv”, Proc. of the 10th IEEE Int. Conf. Anti-Counterfeiting, Secur. Identification (ASID), 2017.

[38] G. J. Holzmann, “The spin model checker”, IEEE Trans. Softw. Eng., Vol. 23, No. 5, 1997.

[39] Li, Wei, et al, “Formal analysis and verification for an ultralightweight authentication protocol RAPP of RFID”, National Conference of Theoretical Computer Science, 2017.

[40] G. Goss, J. Hartmanis, J. Van Leeuwen, “Stabilization, safety, and security of distributed systems”, proc. of the 14th International Symposium, Vol. 9, No. 3. 2012.

[41] A. Armando, R. Carbone, L. Compagna, “SATMC: a SAT-based model checker for security critical systems”, Proc. of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS’14), Vol. 8413, 2014.

[42] S. P. Johnson, G. W. Wilson, Y. Tang, K. S. Scott, “SATMC: Spectral energy distribution Analysis Through Markov Chains”, Monthly Notices of the Royal Astronomical Society, Vol. 436, No. 3, 2013.

[43] A. Armando, R. Carbone, L. Compagna, "SATMC: a SAT-based Model Checker for Security-critical Systems", International Conference on Tools and Algorithms for the Construction and Analysis of Systems, 2014.