The hardware modeling and analysis techniques based on Kripke structure

Da Xiao[1, 2,a,] Yuefei Zhu[1, 2], Shengli Liu[1, 2], Dongxia Wang[3], Ruiqing Xiao[1, 2]

1. Computer network department, Zheng Zhou Information Science University,
450001, Zheng Zhou, He Nan, China
2. State Key Laboratory of Mathematical Engineering and Advanced Computing,
450001, Zheng Zhou, He Nan, China
3. National Key Laboratory of Science and Technology on Information System Security，100101，Beijing, China
([a]shuttle12@163.com)

Abstract.

This paper focuses on two aspects. One is the modeling method of hardware function based on Kripke structure. The other is the hardware design specification formal description method which based on temporal logic CTL. In order to design a hardware modeling method which based on Kripke Structure, Hardware implementation process and behavioral characteristics modeling method were studied deeply. Paper present the basic ideas and realize principles of the hardware Trojan detected by a model test method.

Keyword: Model Testing; Kripke; CTL; Detection of hardware Trojan; Migration Relations;

Introduction

Model Checking is a formal verification method. It try to make a model for the target system by the finite state machine, and use finite state machine to describe or characterize all the features, properties and the properties of the target system. Then, formally describe the desired properties of the target system by temporal logic, and then verify the formula represented by temporal logic for all the status nodes which belongs the finite state machine. Therefore, model checking is a formal verification method with higher efficiency to achieve. Currently, model checking has been widely used in digital hardware design, protocol analysis, complex control system validation and other applications, and application results achieved very remarkable. This is a very effective method which can formally verify the target system function, attributes and behavioral characteristics. This paper will research the principle and method that used in the model checking field of hardware Trojan detection.

© 2015. The authors - Published by Atlantis Press

Kripke structure analysis

Kripke semantics is a relational semantics, which is also called framework semantics. It is a formal semantics of modal logic system. Such a system is Saul Kripke in the late 1950s to the early 1960s established [143]. Respect with classical logic semantics which has emerged earlier, Kripke semantics added inevitability ($\Box$), possibility ($\Diamond$) and some other modal operators. Later Kripke semantics is applied on the field of intuitively logical semantics, and develop into an important logical semantics.

In fact, Kripke structure is another representation of the state transition diagram, it has a powerful expression ability. Moreover, Kripke structure is a mathematical structure, it can strictly defined branching temporal logic semantics. Therefore, Kripke structure is widely used for system modeling.

Definition.1, Kripke structure, Kripke structure expressed by a quintuple, $K =< S, S_0, R, AP, L >$ [144][145]. Among them:

S is the set of states in the system described;

S_0 is the set of the initial states in the system described, S_0 satisfy the relationship of $S_0 \subseteq S$;

R is the transfer relations between states of the system is described;

If the set $Q = S \times S$, then for the relationship of transfer, $q = (s1, s2)$, which between any two states $s1 \in S$, $s2 \in S$, that in the system is described, satisfy $q \in Q$.

AP is the sets of all the atomic propositions and negative atomic proposition, which in the system is described;

$L : S \to 2^{AP}$ is a marker function in the system is described, which describes an arbitrary state of system, $s \in S$, mapped to the set of true atomic propositions in the system $SZ \subseteq AP$, remarked $L(s) = SZ$.

If $L(s) = SZ$, $SZ \subseteq AP$, $ap1$ is the true atomic propositions, that is $ap1 \in SZ$, if $M, s \models ap1$, M is the finite state model of the system is described by a corresponding.

Based on Kripke structure modeling process

Generating the sets of Kripe status. Since the form of the detection object is embedded in the device firmware code , and therefore , the object to be detected can be expressed in the form of instructions. When the model of Kripke structure is constructed for the detection object, first of all, you need to generate the states set of Kripke structure.

Assumes that the selected state element of Kripke does not contain the location of the instruction, so in the description of the tested object model, through the state can not reflect the relations between the order of the instruction sequence. If selected elements do not contain the name of the instruction, it can't distinguish between each specific instructions. In addition, if the Kripke model is set up, the state elements of the tested object selected do not contain the variable information in the instruction, every state in the Kripke structure has established will not be able to reflect the current instruction received by the parameter information, as well as the change of each variable in the instructions execution.

Therefore, in order to be effective for instruction of tested object modeling and testing, this paper builds the Kripke structure state model, the factors of its state construct by the location of the instructions the location, the instructions' name and numerical in the judgment register of instructions three elements of all. The representation is $< instrc\_loc, instrc\_name, var\_value >$.

The generation of Kripke migration relations. In order to quickly detect the hardware Trojan, this paper draws on the traditional code analysis technology, scanning the firmware code by the existing tools and techniques, to realize the extraction of program control flow diagram.

The definition of the control flow chart is as follows:

Definition.2, CFG(Control flow graph), $G = (S, E, S0, SX)$ a directed graph to describe the relationship between program instructions executed, among them, S is a collection of nodes in the graph. $E = \{ei =< S_i, S_j > | S_i \in S, S_j \in S\}$, S_0 is a collection of program entry node. SX is a collection of export node, similarly, a program can export only one program, it can also have multiple programs outlet.

When modeling using Kripke model on the target system, through the previous research has identified each state of Kripke model contains the location of the instruction, the instruction's name and the value of discriminant register in instruction, therefore, a node in the control flow graph in Kripke models, may be converted into more than one state. Therefore, when using Kripke model modeling, control flow graph after scanning, every node in the need to control flow graph corresponding instruction in semantic analysis, and according to the state definition described by Kripke model, through semantic analysis of the instruction, determine the state number of currently instruction will converted, then, according to the relationship between control flow of the program control flow graph, generate state relations of migration.

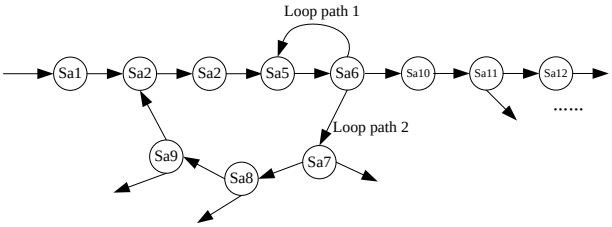

Fig. 1, The Loop path in the process of Migration relation generation

Since the program control flow graph is a directed graph , and therefore may exist cycle path, as shown in Fig.1. In order to avoid the emergence of the state of line cycle, therefore, when generating the migration relationship Kripke model, every generation needs a new state record the current state is derived from the control flow graph in which node. When migrating relationship Kripke model state is continually recorded, Each generates a new state, need to check the current generated by the new state, the corresponding node in the control flow graph is ever seen nodes. If the node has appeared , then considered to have been duplicated nodes, then enters a loop path, At this time show that in the generated current migration relations, this is the end of this node which belong to control flow graph of the current scan, successor node of this node does not need to continue scanning.

The generation of Kripke tag function. Due to the Kripke model is set up on the target system, research and realize the target system's state set and state migration relationship. Especially in the migration relations during the implementation of the target system, Transition diagram of the program expressed by the target system was scanned, moreover, the program transition diagram corresponding to each node semantic analyzes, therefore, in the course of analysis, the state number of each node will be split in the transfer function graph can be obtained, the specific meaning of each state can be obtain. For example, for a status of Kripke model, the status contained a variety of changes of the variable range, the specific command names and other information.

Marking function and the generation process is as follows:

Step1: The target system program scanning, the control flow graph of the target system.

Step2: To analyze of the control flow graph generated by scanning, to analyze each node of control the flow diagrams, according to the generating strategy in the state of Kripke model, mapping out the branch every node in the graph of the state.

Step3: Record for each state in the semantic Kripke model records, especially record for the scope of relevant variables or parameters in each state, the record of this information into an atomic formula form of expression, that is to mark the state function representation.

Step4: All the state tag in function is obtained by this way of enumeration of the state tag in function.

Branching temporal logic CTL

Definition.3, CTL formula syntax, grammar of CTL formula is defined by the following three statements.

（1）All atomic formulas are CTL formulas ;

（2）If the equation φ, ϕ are CTL formula, the following conversion formula , after still CTL formula:

$\neg\varphi$、$\neg\phi$、$\varphi \wedge \phi$、$\phi \wedge \varphi$、$AX\varphi$、$EX\varphi$、$AX\varphi$、$A(\varphi U \phi)$、$E(\varphi U \phi)$

（3）Represented by the formula (1) (2) , after the resulting finite calculation formula are CTL formula.

In the definition, (2) although given some basic arithmetic expression CTL formula , but did not give a CTL path quantifiers and temporal symbols for all possible combinations of operations, this is mainly due to the various other combinations can be converted to the final calculation of formula (2) shown in the logical operation . Including:

$\varphi \vee \phi = \neg(\neg\varphi \wedge \neg\phi)$; $AG\varphi = \neg E(true\, U\, \neg\phi)$

$AF\varphi = A(true\, U\, \varphi)$; $EG\varphi = \neg A(true\, U\, \neg\varphi)$; $EF\varphi = E(true\, U\, \varphi)$

Definition.4, CTL semantics，

$s \mid\!- p \quad iff \quad p \in L(s)$ In the formula, p is an atomic proposition , $L(s)$ is a marker function in the model s ;

$s \mid\!- \neg g \quad iff \quad s \mid\!- g$; $s \mid\!- g \wedge h \quad iff \quad s \mid\!- g \wedge s \mid\!- h$

$s \mid\!- AXg \quad iff \quad \forall\pi(\pi = \{s_0, s_1, s_2, ...\} \wedge (s_0 = s), s_1 \mid\!- g)$

$$s \mid - AXg \quad iff \quad \exists \pi(\pi = \{s_0, s_1, s_2, ...\} \wedge (s_0 = s), s_1 \mid - g)$$

$$s \mid - A(gUh) \quad iff \quad \forall \pi(\pi = \{s_0, s_1, s_2, ...\} \wedge (s_0 = s), \exists i\,(s_i \mid - h) \wedge \forall j(j < i -> s_j \mid - g))$$

$$s \mid - E(gUh) \quad iff \quad \exists \pi(\pi = \{s_0, s_1, s_2, ...\} \wedge (s_0 = s), \exists i\,(s_i \mid - h) \wedge \forall j(j < i -> s_j \mid - g))$$

Hardware Trojan detection system based on Model Checking

Based on the principles of model -based hardware Trojan detection test , detailed design implementation flow hardware Trojan detection systems. As shown in Figure 2 .

When detect hardware Trojans, first of all, you need to define the function of the target system. Function of the target system have many forms , For example describing the function of the target system, target system function specification, or target system design requirements and the like. Try to express it as normalized function and nature of the target system, finally, described the functions by the corresponding logical formulas.

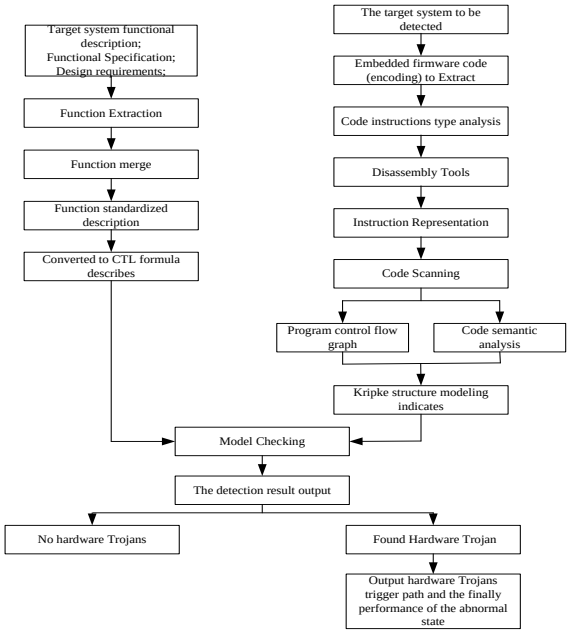

Fig. 2, Hardware Trojan detection principle based on model checking

In detection process, first, extract the firmware code, then the code is scanned, through the current maturity for such architectures scanning tool, Can effectively extract the control flow graph of the code. Then based on the control flow grap, semantic analysis the embedded firmware code, get the semantics of each node in control flow graph corresponding, Provides status set migration relations and marking functions and other related information for the generation based modeling Kripke structure.

According to the program control flow graph and the semantic analysis results, achieve to expresst the model of target system in Kripke structure. Then, model cheching CTL formulas and models described by Kripke structure previously. Entire testing process can be automated execution , according to the results of the implementation of the final output the results. If Kripke structure with each state represented on the target system modeling are met CTL formula, that indicates that the target system is a model to be detected CTL formulas, detection of the target system found no hardware Trojans. If you find during testing have established Kripke structure model, a state does not satisfy the CTL formula. That indicates that the path from the initial state to trigger and perform a complete hardware trojan execution path .

Summary

The key issue of hardware Trojan detection is testing and modeling the target system's function, behaviors and attributes , and, the desired properties of the target system expressed by the temporal logic .The paper do a useful exploration research of this crucial issue.

Acknowledgment

This work is supported by the National Science-Technology Support Plan Project of China (No. 2012BAH47B01), by the Natural Science Foundation of China (No. 61271252), and by the Municipal Science and Technology Innovation Team Project of Zhengzhou (No. 10CXTD150).

References：

[1] Jaghoori MM, Sirjani M, Mousavi MR, Efficient symmetry reduction for an actor-based model[J].Chakraborty, G.(ed.)ICDCIT 2005. Springer, Heidelberg,2005, 3816:494-507.

[2] Xiao D, Zhu Y F. A Research on Detection Algorithm of Failure-Type Hardware Trojan[A].Multimedia Information Networking and Security (MINES) [C], 2012 Fourth International Conference on. IEEE, 2012: 918-921.

[3] Sergei S., Christopher W. Breakthrough Silicon Discovers Backdoor in Military Chip [A]. In: Proceedings of 14th International Workshop on Cryptographic Hardware and Embedded Systems [C], Leuven, Belgium, 2012:23-40.

[4] Santos J., Carlos M., Yunsi F. Designing and implementing a Malicious 8051 processor [A]. In: Proceedings of 2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems [C], Austin, TX, 2012:63-66.

[5] Bastian S. Model checking of software for microcontrollers [J]. ACM Transactions on Embedded Computing Systems (TECS).2010, 9(4):1-27.

[6] Johannes K., Stefan K., Christian S., Helmut V. Proactive Detection of Computer Worms Using Model Checking[J].IEEE Transactions on Dependable and Secure Computing, 2010,7(4): 424-438.