Think Smart, Play Dumb: Analyzing Deception in Hardware Trojan Detection Using Game Theory

Tapadhir Das [1,1], AbdelRahman Eldosouky [2], and Shamik Sengupta [2]

[1]University of Nevada
[2]Affiliation not available

November 8, 2023

Abstract

In recent years, integrated circuits (ICs) have become
significant for various industries and their security has
been given greater priority, specifically in the supply chain.
Budgetary constraints have compelled IC designers to offshore manufacturing to third-party companies. When the designer gets the manufactured ICs back, it is imperative to test for potential threats like hardware trojans (HT). In this paper, a novel multilevel game-theoretic framework is introduced to analyze the interactions between a malicious IC manufacturer and the tester. In particular, the game is formulated as a non-cooperative, zerosum, repeated game using prospect theory (PT) that captures different players' rationalities under uncertainty. The repeated game is separated into a learning stage, in which the defender

learns about the attacker's tendencies, and an actual game stage, where this learning is used. Experiments show great incentive for the attacker to deceive the defender about their actual rationality by "playing dumb" in the learning stage (deception). This scenario is captured using hypergame theory to model the attacker's view of the game. The optimal deception rationality of the attacker is analytically derived to maximize utility gain. For the defender, a first-step deception mitigation process is proposed to thwart the effects of deception. Simulation results show that the attacker can profit from the deception as it can successfully insert HTs in the manufactured ICs without being detected.

This paper has been accepted for publication in **IEEE Cyber Science Conference 2020**

Think Smart, Play Dumb: Analyzing Deception in Hardware Trojan Detection Using Game Theory

Tapadhir Das
Dept. of Computer Sci & Engineering
University of Nevada, Reno
Reno, USA
tapadhird@nevada.unr.edu

AbdelRahman Eldosouky
Dept. of Computer Sci & Engineering
University of Nevada, Reno
Reno, USA
iv727@vt.edu

Shamik Sengupta
Dept. of Computer Sci & Engineering
University of Nevada, Reno
Reno, USA
ssengupta@unr.edu

***Abstract*—In recent years, integrated circuits (ICs) have become significant for various industries and their security has been given greater priority, specifically in the supply chain. Budgetary constraints have compelled IC designers to offshore manufacturing to third-party companies. When the designer gets the manufactured ICs back, it is imperative to test for potential threats like hardware trojans (HT). In this paper, a novel multi-level game-theoretic framework is introduced to analyze the interactions between a malicious IC manufacturer and the tester. In particular, the game is formulated as a non-cooperative, zero-sum, repeated game using prospect theory (PT) that captures different players' rationalities under uncertainty. The repeated game is separated into a learning stage, in which the defender learns about the attacker's tendencies, and an actual game stage, where this learning is used. Experiments show great incentive for the attacker to deceive the defender about their actual rationality by "playing dumb" in the learning stage (deception). This scenario is captured using hypergame theory to model the attacker's view of the game. The optimal deception rationality of the attacker is analytically derived to maximize utility gain. For the defender, a first-step deception mitigation process is proposed to thwart the effects of deception. Simulation results show that the attacker can profit from the deception as it can successfully insert HTs in the manufactured ICs without being detected.**

***Index Terms*—Hardware trojans, deception, hypergame theory, game theory, prospect theory, cybersecurity, integrated circuits.**

I. Introduction

The recent years have seen a tremendous and unprecedented growth in technology. Innovations such as the Internet of Things, artificial intelligence, big data, and autonomous vehicles have taken over cyberspace. This, in turn, has led to an increase in usage of electronics in such systems, in particular, integrated circuits (ICs). From the automotive and aerospace industry to the field of consumer electronics, ICs are vital and an integral part of such sectors [1].

Due to the rising cost of manufacturing, many prominent chip makers are outsourcing their designs elsewhere to help reduce the costs [2]. Outsourcing manufacturing of ICs to third party vendors help make manufacturing cost-effective for designers, but it also may introduce serious security risks and threats [3]. Because of their immense use in cyber systems, protecting ICs has gained a lot of attention, recently. One serious threat in the field of IC manufacturing is hardware trojans (HT) [4]. A HT is a malicious design that can be added to the circuitry of an IC, in order to corrupt its functionality.

This research is supported by the National Science Foundation (NSF), USA, Award #1723814.

HTs come with varying degrees of impact to an IC. Some HTs can cause error detection modules to accept inputs that should be rejected while some others can downgrade the performance by intentionally corrupting a device's operational parameters. Certain trojans can leak sensitive data by creating a backdoor for malicious hackers into the IC [5]. Certain trojans can also generate a Denial-of-Service attack by targeting modules to exhaust scarce resources like bandwidth, computation, and battery power [6]. This makes hardware trojans a serious security risk to an IC. Moreover, the impact of HTs is exacerbated when the infected ICs are used in cyber physical systems, e.g., cognitive radios [7], IoT health systems [8], robotics [9], and unmanned aerial vehicles [10], as HTs can facilitate cyber-physical attacks in such systems.

A. Related Work

HTs are designed to be stealthy, meaning that they cannot be easily found, and they might not be activated until certain time, current, temperature, voltage, or logic factors are met [11]. Therefore, once the manufactured ICs are brought back to the designer, testing these circuits for potential security hazards are of foremost priority. There are multiple strategies that can be employed to test the prevalence of hardware trojans in an IC [4] and [12]. In [4], the authors discuss two types of trojan detection techniques: destructive and non-destructive. In destructive techniques, the ICs are de-metallized to check the inner circuits, which is expensive and time consuming. On the other hand, non-destructive techniques involve side-channel analysis and logic testing. In [12], the authors used localized current analysis, on certain portions of an IC, to detect hardware trojans. However, the methods in [4] and [12] rely heavily on the availability of adequate resources for testing, and may be hindered by the lack of resources for effective testing. This raises the need for efficient testing strategies that can detect the most trojans under limited resources.

In order to combat the issues revolving around lack of testing resources, a promising approach that is recently being explored is studying the strategic interactions that may take place between an IC manufacturer and a designer (tester). This can help the tester to efficiently use its resources to detect the most trojans. One effective method to carefully study the interactions between agents, in a given situation, is by using game theory. Game theory [13] provides a powerful mathematical tool to study such interactions and enables each

party to achieve its best outcome in light of its opponents' actions. For instance, the works in [14] and [15] attempt to solve this hardware trojan testing scenario using game theory. In [14], the authors develop a "Trust Game" to illustrate the value of both the iterated elimination of dominated strategies and Nash equilibrium solution concepts. This work has been extended in [15] by computing multiple mixed strategy Nash equilibria which allow to effectively identify the optimal testing strategies to detect hardware trojans in a given IC.

The main assumption behind these game-theoretic frameworks is that the players involved are fully rational. However, it was observed that players play irrationally, when facing obstacles or uncertainty, and they tend to deviate from their most rational choices [16]. This phenomenon is best modeled using prospect theory (PT) [17], which can be combined with game theory in the strategic decision making [18] and [19]. In [18], the authors applied PT to a static game to protect drone delivery systems. In [19], PT was used to encapsulate the "user-centric" approach on microgrid power trading.

This use of PT was adopted in literature to study the HT problem under the case of limited rationality [11]. In particular, the authors in [11] formulated and analyzed a hardware trojan game, using a weighting effect of PT, to provide a subjective understanding of the interactions between the attacker and a defender. This application of prospect theory to hardware trojan detection problems, opened new doors into the research behind subjective human perceptions. However, one limitation of such works, i.e., [11], [14], and [15] is that they do not consider the concept of deceit. Deception refers to the act of intentionally behaving in a manner that is not consistent with one's true behavior. The idea of deception is based on the concept of misrepresentation [20]. Misrepresentation of one's true intentions or capabilities are common in real world strategic interactions. The goal is to deceive one's opponent in order to have different perceptions about them. Multiple examples of this interaction can be found in [21] and [22]. For instance, the authors in [22], studied the impact of tactics and deception in chess where a player can deliberately sacrifice pieces at the start to know their opponent's favorite piece and use this knowledge to its advantage during the game. To this end, the problem of deception in HT games is a promising research direction that we are exploring in this paper.

B. Contributions

The main contribution of this paper, is, thus, a multi-level game-theoretic framework to study and model deception in hardware trojan detection. First, we use game theory to model a non-cooperative game between the attacker and the defender, based on available strategies for each player. This game theory model represents the first level of the framework in which players are fully rational and is based on the works of [15]. The next level of the game accounts for the players' rationalities, modeled after [11]. In particular, the players' strategic profiles are weighted according to the players' rationalities. In this level of the game, prospect theory is used to capture the players' strategic deviations and subjectivity under uncertainty.

The whole game is, then, formulated as a repeated game where the static game is played over multiple stages. This resembles the case in which the ICs are returning from the manufacturer in different batches. Due to its limited resources, it is impossible for the defender to test for every single potential trojan on every IC in every batch. Thus, we propose that the defender can learn the attacker's strategies in order to develop strategic testing patterns and then, apply this learning to subsequent stages of the game. Consequently, we break the game down into two separate stages: a learning stage, in which the defender learns, and an actual game stage, in which the defender applies the learning. However, under this scenario, it becomes motivating for the attacker to deceive the defender by playing with a lower rationality during the learning stage, than its true rationality. In a sense, it will be "playing dumb" in order to deceive the defender during the learning stage. That makes the defender think that the attacker has a lower rationality (deception rationality), and, thus its actions are not aligned with its interests. The defender will then focus on that respective attacker strategy profile, which allows the attacker to play at a higher rationality (actual rationality) than what the defender is expecting, during the actual game stage. To the best of our knowledge, this is the first work to consider deception through manipulating the rationality levels in PT.

Subsequently, to capture this misrepresentation and deception in game scenarios, an additional level of game play is presented using "hypergame theory" [20]. Hypergame theory is an extension of traditional complete information games. It is structured as a hierarchical game where certain players have an extended view of the game from their opponents. The opponents may or may not have any idea about this extended view of the game and have a perceived partial view of the same game. In this paper, hypergame theory is used to analytically derive the optimal deception rationality of the attacker, i.e., the lower rationality that the attacker will pretend it has. Finally, we propose a first-step defense technique for the defender to mitigate the effects of the deception, in case it is unaware of the occurrence of the deception. We, then, show through simulations that the attacker can benefit from the proposed deception attack as it can insert HTs into the manufactured ICs without being detected. Simulation results are also used to study the attacker's deception utilities under different combinations of attacker's and defender's rationalities.

The rest of this paper is organized as follows: Section II highlights the system model and problem formulation. Section III demonstrates the first two levels of the game which are the non-cooperative game and the extended game using prospect theory utilities, as well introducing the repeated game scenario. In Section IV, the deception scenario is modeled using hypergame theory and the defense mechanism is introduced. Numerical results and simulations are presented and analyzed in Section V. Finally, conclusions are drawn in Section VI.

II. System Model

Consider an IC manufacturing company, referred to as the "attacker", that has an incentive to attack a designing company,

labeled as the "defender". To minimize the probability of being detected, the attacker will insert a single trojan t from a set of $\mathcal{T}$ trojan types. Each trojan leads to a certain damage and provides a respective utility, V_t, to the attacker if the trojan went undetected. Due to unique operational parameters of each trojan, e.g., voltage, current, access to certain modules, every trojan can only be inserted in a unique partition of each IC.

After getting the designed ICs back from the manufacturing company, the defender's job is to test the ICs for potential threats like hardware trojans. As modern day ICs are extremely complex, it is very resource expensive to test for every kind of potential trojan on every IC. Due to limited testing resources, the defender can only test for a certain subset of trojans per IC, which is a common assumption in literature [11] and [15]. To this end, let $\mathcal{A}$ be tester's subset of trojans that can be tested at a time, such that $\mathcal{A} \subset \mathcal{T}$. This subset will include all different combinations of trojans that can be tested simultaneously, based on the tester's capacity.

If the defender successfully detects the presence of a HT, the attacker incurs a fine of F_t where t refers to a trojan type and $t \in \mathcal{T}$. The magnitude of the fine could represent legal consequences for trying to infect an IC with a certain trojan type. Types of legal consequences and fines could range from paying a monetary amount for damages to the termination of the contract between designer and manufacturer [11]. In this model, we limit F_t to monetary fines as in a real-life scenario, it will take the designer a long time to terminate the contract and to shift the production to another manufacturer.

To understand the interactions between the attacker and the defender, we use game theory to study this interactivity. The goal is to mathematically model the interactions, between the players, to find the attacker's strategies of inserting certain trojans within an IC, and use these strategies to help the defender develop appropriate testing patterns.

III. Game Formulation

In this game, each player wants to play its best strategy upon its perception of its opponent's potential strategy. The strategies that are employed by both players end up either corrupting the IC or levying a fine for the attacker. Here, the game will be modeled as a static non-cooperative game. We also consider the case where this game is repeated over time.

A. Static Game

We consider two players: the attacker a and the defender d in a set $\mathcal{N}_P$ such that $\mathcal{N}_P := \{a, d\}$. Let the set $\mathcal{S}$ represent the strategy spaces $\mathcal{S}_d$ and $\mathcal{S}_a$ of the defender and the attacker, respectively. These strategy spaces represent all the possible actions for the players. Let the set $\mathcal{U}$ represent the utility functions of the players U_d and U_a, for the defender and the attacker, respectively. Finally, let the game $\mathcal{G} = \{\mathcal{N}_P, \mathcal{S}, \mathcal{U}\}$.

For the attacker, the strategy space consists of all potential kinds of trojans that can be played in this game, i.e, $S_a = \mathcal{T}$. Here, every strategy in the attacker's strategy space $s_a \in S_a$ refers to a corresponding trojan type $t \in \mathcal{T}$. On the other had, the defender will select a subset of trojan types to test simultaneously per stage of the game, due to limited resources. Let the number of trojans that the defender can test at once be K types. The strategy space of the defender can then be $\mathcal{S}_d$ defined as all the possible subsets of $\mathcal{T}$ with the size of K, i.e, size of $\mathcal{S}_d = \binom{T}{K}$. Here, every subset possibility is present in the defender's strategy space $s_d \in \mathcal{S}_d$.

Players' utilities can be determined using the strategy selection for the attacker s_a and the corresponding strategy selection of the defender s_d such that:

$$U_a(s_a, s_d) = \begin{cases} -F_{s_a} & \text{if } s_a \in s_d, \\ V_{s_a} & \text{otherwise}, \end{cases} \tag{1}$$

where V_{s_a} is the attacker's reward for playing a certain strategy s_a that was not detected by the defender strategy s_d. $-F_{s_a}$ is the attacker's fine for playing a certain strategy that was detected. The magnitude of V_{s_a} reflects the monetary reward gain by the attacker, which also corresponds to the type of damage that the trojan can cause. We notice that the outcome of the game will be either a fine F_{s_a} charged from the attacker and paid to the defender, or the attacker's gain V_{s_a} which is the defender's loss. Thus, the game will feature a zero-sum characteristic and the defender's utility can then be given as:

$$U_d(s_a, s_d) = -U_a(s_a, s_d). \tag{2}$$

Finally, let $P = \{p_a, p_d\}$, represent the objective mixed strategy probability distribution for the attacker and the defender respectively, over their actions.

Next, we study the effect of prospect theory (PT) [17] on players' utilities. According to PT, players deviate from their most rational strategies when faced with uncertainties regarding strategies, or if there are limitations to playing a certain strategy that is not taken into account during the game. Under PT, players have a subjective rationality of the opponent's strategies, and hence, change the expected utilities from an objective to a subjective one. In our game, both players are facing uncertainty when it comes to the opponent's strategies. For the attacker, it is not completely sure of the testing strategy that the defender will employ. Therefore, it may tend to underweight or overweight a particular strategy of their opponent as referred in [11]. The same assumption can be made for the defender. Also, as both players are human, there could be multiple reasons for the subjectivity: company regulations, requirement of more resources to play a certain trojan, not enough resources to test for certain trojans, etc.

In order to capture the deviation from the optimal strategy for each player, a *weighting effect* w is incorporated. Under this weighting effect w, players give a subjective weight to their opponent's strategies for more relevance. The weighting effect depends on the *rationality parameter* $\alpha(0, 1]$, which judges a player's subjective perception based on their objective probability. A rationality of 1 means players are playing with full rationality, i.e., complete objectivity. The weighting effect is defined using the Prelec function [23], as follows:

$$w_i(p_i, \alpha_i) = \exp(-(-\ln p_i)^{\alpha_i}), 0 < \alpha_i \leq 1. \tag{3}$$

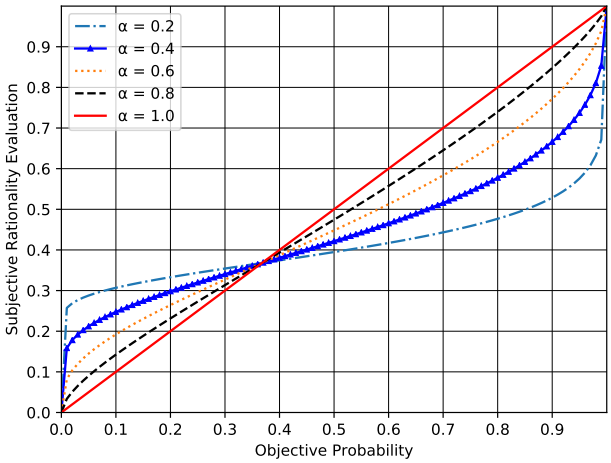

Fig. 1. Player objective probability vs their subjective rationality evaluation of a strategy

Note that, the Prelec function is widely used to model subjectivity when studying rationality [11] and [18]. Fig. 1 shows the impact of α_i on the deviation of a player between their objective probability and corresponding subjective evaluation. Using the Prelec function, the utility for every player can then be updated with respect to their perceived rationality about their opponent's probabilities as follows:

$$U_i^{PT}(p_i, p_j, \alpha_i, \alpha_j) = \sum_{s=S_i} \Big(p_i(s_i) w_i\big(p_j(s_j)|_{\alpha_j}, \alpha_i\big) \Big) u_i(s_i, s_j), \tag{4}$$

where i and j correspond to a player in this scenario.

To optimize their utilities, players need to consider both their actions as well as their opponents' actions. The solution in this case, is given from game theory as the equilibrium point [13]. Equilibrium solutions, in game theory, are referred to as Nash equilibrium which occur when no player can improve its utility by unilaterally changing its actions. Nash equilibrium can either be pure Nash equilibrium when every player chooses only one action, or mixed-strategy Nash equilibrium which is a probability distribution over the player's set of actions [24]. Here, we focus on mixed-strategy Nash equilibrium and study this equilibrium solution in Section V.

B. Repeated Game

Here, we consider the case where the manufacturer returns the ICs to the designer in multiple batches. Each batch consists of multiple identical ICs. Checking all the ICs in one batch, for all types of HTs, is an unfeasible process for the defender. Therefore a promising approach for the defender is to learn about the attacker's strategies by checking every single IC, for every possible HT, in an initial set of batches to figure out the attacker's probabilistic preferences. Then, this knowledge will be used to test for HTs in subsequent batches. This scenario represents a repeated game [13], which we propose to divide into two separate stages: learning stage and actual game stage.

Note that, under this scenario, players will take actions at the beginning of each stage. These actions will affect their outcomes from all the subsequent batches. Let N be the total number of batches in the game. For every batch in the learning stage, the defender is going to check all the ICs per batch, denoted by C_L. In the game stage, the defender will randomly select a few ICs per batch to test, denoted by C_A such that $C_A < C_L$. In every IC, during game stage, the defender will look for the same number of HTs, given by K. The defender will choose N_L as number of learning batches and N_A to be the number of batches in the game phase, where $N_A = N - N_L$. The utility in the learning stage U_L will be computed by:

$$U_{L_i} = C_L U_i^{PT}(p_i, p_j, \alpha_i, \alpha_j), \tag{5}$$

Similarly, the utility in the actual game stage will be given by:

$$U_{A_i} = C_A U_i^{PT}(p_i, p_j, \alpha_i, \alpha_j). \tag{6}$$

The total utility U_{T_i} of the entire game will be given by:

$$U_{T_i} = U_{L_i} + U_{A_i}. \tag{7}$$

where i denotes a player in this game. The equilibrium of this repeated game is discussed in detail in Section IV-B.

IV. Hypergame Model for Deception

We consider the case in which the attacker wants to exploit the learning stage to deceive the defender. We assumes that the attacker can use its knowledge of K and N to infer the number of learning batches. We notice from Fig. 1 that according to a player's rationality, it will weigh the probabilities in a different order between low and high probabilities. The inflection probability, from Fig. 1, is about 0.37. For instance, a probability of 0.1 will be weighed higher for low rationality levels, e.g., 0.2 than for a rationality of 0.8. Similarly, a probability of 0.7 will be weighed higher for a rationality of 0.8 than for a rationality of 0.2. This gives an incentive for the attacker to misrepresent itself and play a different rationality between the game stages. Thus, the attacker can try to deceive the defender by playing a lower rationality or "acting dumb" during the learning stage, and then plays its actual rationality in the actual game stage. This misrepresentation of one's true tendencies is prevalent in the world of interaction, and capturing it is the focus of this paper.

A. Hypergame Theory

In traditional complete information games, all players are assumed to be aware of all players' strategies, and the respective utilities with these strategies. However, in the case of deception, we assume an additional level of utilities that is available only to the deceiving player. To model this scenario, we will use the framework of hypergame theory [25] which assumes that players involved aren't seeing the same view of the game. In our case, the defender assumes that the attacker is continuing with the same rationality played during the learning stage; in reality, the attacker's rationality has changed. This change in rationality allows the attacker to improve its expected utility without the defender's awareness, giving the attacker an extended view of the game over the defender's perceived view. In this regard, the attacker can use a deception

rationality of α_{a_L} during the learning stage and then α_{a_A}, which is its actual rationality, during the actual game stage such that $\alpha_{a_L} < \alpha_{a_A}$.

B. Hypergame Model

We model the game as a first level hypergame [20] such that the players have different views of the game. Under the considered deception scenario, only the attacker has the complete view of the defender's strategies while the defender has a limited view of the attacker's strategies. Let G_d be the defender's view of the attacker's strategy as follows:

$$G_d = \begin{cases} \alpha_{a_A}, & \text{under no deception}, \\ \alpha_{a_L}, & \text{under deception}, \end{cases} \tag{8}$$

such that under a normal game without deception, the defender will perceive the attacker's actual rationality which will be the same during both game stages, i.e., α_{a_A}. However, under the deception case, the defender will only perceive the attacker's deception rationality during the learning phase and it will not be aware of the rationality change in the actual game. Similarly, the attacker's view of the game can be given by G_a which will equal the defender's actual rationality α_d, i.e., $G_a = \{\alpha_d\}$. The hypergame $\mathcal{H}$ can then be given by all players' views of the game, i.e., $\mathcal{H} = \{G_d, G_a\}$.

Since the defender is unaware of the deception, its utility in the game will be given by (7). On the other hand, the attacker, as being the deceiver, will have an extended view of the game which represents its outcome from the deception process. The expected utility of the attacker, due to deception, can then be given as the difference between its utilities, in the actual game stage, with and without deception as follows:

$$\begin{aligned} U_a^H = C_A(&U_a^{PT}(p_a, p_d, \alpha_{a_A}, \alpha_d) \\ &- U_a^{PT}(p_a, p_d, \alpha_{a_L}, \alpha_d)). \end{aligned} \tag{9}$$

Equation (9) highlights the deception taking place in this game. After the learning stage, the defender is expecting the attacker to play the strategic profile that corresponds to the deception rationality level α_{a_L}. However, instead the attacker is playing with its actual rationality α_{a_A} which is a higher rationality. The defender only gets to see a partial view of the entire game, which is the game defined in Section III-B. Thus, the defender will check for the trojans corresponding to the probability distributions of the attacker when its rationality is α_{a_L}. In fact, this utility will not reflect the actual status of the game as the attacker will be able to insert other trojans without being detected. The attacker, on the other hand, wants to maximize its utility, which is unknown to the defender. Based on its actual rationality α_{a_A}, the attacker wants to choose the deception rationality α_{a_L} from a set of rationalities $\mathcal{A}_\alpha$ under which its utility in (9) will be maximized. This can be done by solving the following optimization problem:

$$\begin{aligned} \underset{\alpha_{a_L} \in \mathcal{A}_\alpha}{\operatorname{argmax}} \; & C_A(U_a^{PT}(p_a, p_d, \alpha_{a_A}, \alpha_d) \\ & - U_a^{PT}(p_a, p_d, \alpha_{a_L}, \alpha_d)), \end{aligned} \tag{10}$$

which is equivalent to:

$$\underset{\alpha_{a_L} \in \mathcal{A}_\alpha}{\operatorname{argmax}} \; U_a^{PT}(p_a, p_d, \alpha_{a_A}, \alpha_d) - U_a^{PT}(p_a, p_d, \alpha_{a_L}, \alpha_d). \tag{11}$$

Substituting (4) into (11), we get:

$$\begin{aligned} \underset{\alpha_{a_L} \in \mathcal{A}_\alpha}{\operatorname{argmax}} \; & \Big(p_d(s_d) w_d\big(p_a(s_a)|_{\alpha_{a_L}}, \alpha_d\big)\Big) u_a(s_d, s_a) \\ & - \Big(p_d(s_d) w_d\big(p_a(s_a)|_{\alpha_{a_A}}, \alpha_d\big)\Big) u_a(s_d, s_a), \end{aligned} \tag{12}$$

which can be simplified by omitting the common terms as:

$$\underset{\alpha_{a_L} \in \mathcal{A}_\alpha}{\operatorname{argmax}} \; e^{(-(-p_a(s_a)|_{\alpha_{a_L}})^{\alpha_d})} - e^{(-(-p_a(s_a)|_{\alpha_{a_A}})^{\alpha_d})}, \tag{13}$$

which can be further simplified as:

$$\underset{\alpha_{a_L} \in \mathcal{A}_\alpha}{\operatorname{argmax}} \; e^{(-p_a(s_a)|_{\alpha_{a_A}})^{\alpha_d} - (-p_a(s_a)|_{\alpha_{a_L}})^{\alpha_d}}. \tag{14}$$

From (14), an attacker can maximize its utility by choosing a rationality level that maximizes the difference between its probability distribution under this deception rationality and the probability distribution under its actual rationality, when both distributions are weighted with the defender's rationality.

Proposition 1: The Nash equilibrium of the game $\mathcal{G}$ along with the solution of (14) constitute a hyper Nash equilibrium to the game $\mathcal{H}$.

A strategy profile represents a hyper Nash equilibrium *iff* it belongs to the Nash equilibrium profile for each player's perceived game [26]. Since the defender's perceived game is only $\mathcal{G}$, the defender will have the same Nash equilibrium in $\mathcal{H}$ as $\mathcal{G}$. On the other hand, solution to (14) represents the attacker's optimal solution based on its perceived game. Apply this solution to $\mathcal{G}$ will result in an equilibrium strategy for the attacker, as it will represent its best outcome based on its perceived game.

C. Deception Mitigation

The discrepancy between the expected utility of the attacker and its actual utility after deception is a massive incentive for the attacker to continue its deception. Without a proper deception countermeasure, the attacker will continue to deceive on all subsequent batches, to achieve higher payoffs. In the proposed game, as the defender is unaware of the deception, it needs some mechanism to mitigate the impacts of any potential deception, if any, in the actual game stage.

Here, we propose that the defender can run the learning stage again during the actual game stage to update its beliefs about the attacker's rationality. Since the attacker will be unaware of this update, it will keep playing the same strategies and it will incur losses as the defender will be able to detect the trojans. Note that, the premise behind the learning stage is to check every single IC in the batch to form an accurate belief about the attacker's strategies. Since this consumes a lot of time and can delay the production stage, the defender is usually limited by the number of times it can re-run the learning stage. The defender's decision to re-run the learning stage will then be based on the available resources and the

amount of time available. Let T be the total time by which all the batches need to be checked and delivered to the next production stage. Let T_L be the time taken to perform the learning stage on a single batch of ICs. Similarly, let T_A be the time taken to perform the actual game stage, i.e., checking C_A ICs on a single batch of ICs. Similar to III-B, N is the total number of batches that is being tested. Let N_L be the number of batches that the defender will dedicate to the learning stages. Let N_A be the number of actual game batches, such that $N_A = N - N_L$. The maximum number of learning batches can then be given by solving the inequality:

$$T_L \cdot N_L + T_A \cdot N_A \leq T, \tag{15}$$

such that the total time spent in both stages is less than or equal to T, from which the value of N_L will be:

$$N_L \leq \frac{T - T_A \cdot N}{T_L - T_A} = \left\lfloor \frac{T - T_A \cdot N}{T_L - T_A} \right\rfloor. \tag{16}$$

Depending on the value of N_L, the defender can distribute its learning stages uniformly during the extent of this game. This will allow the defender to not be completely dependent on the initial learning stage. The effects of this repeated learning on this game scenario require its own in-depth analysis and is left for future work. One way to model this situation can be by using higher levels of hypergame theory in which the defender is aware that there is a different game being played [20]. The attacker, in return, will not be aware of the deception mitigation, which creates a new view of the game available to the defender only. The equilibrium for all these various views of the game will need to be considered in detail. As previously stated, this requires its own analysis and is left for future work. Here, this mitigation technique can be seen as a first step towards thwarting the effects of deception.

V. Simulation Results and Analysis

For our simulations, we assume that the attacker has access to 4 kinds of trojans, such that the strategy space of the attacker $S_A = \mathcal{T}$ = {A,B,C,D}. The gain from each trojan is assumed to be: V_A = 1, V_B = 2, V_C = 4, V_D = 12, which corresponds to the magnitude of damage that can be done with that trojan. For the defender, we let $K = 2$, which usually depends on the resources available to the defender. Based on the value of K, the defender's strategy space S_D will consist of $\binom{4}{2}$ = 6 possible testing strategies, i.e., {AB, AC, AD, BC, BD, CD}. Finally, the attacker's fine F = [8,6,2,4] signifies the penalty for the attacker on successful detection. Table I shows the strategies for both players and their corresponding utilities, in the static game. Since the outcomes are alternating between positive and negative for each player, there is no dominant strategy for any player. The mixed strategy equilibrium can then be reached by executing the fictitious play algorithm [11]. This requires initializing the strategic probabilities. Here, we use the same initial probabilities as in [11]. For the attacker, the initial strategic profile p_a = [0.2083, 0.1667, 0.3333, 0.2917] and for the defender, the initial strategic profile p_d = [0.2051, 0.2564, 0.2564, 0.0513, 0.0513, 0.1795].

TABLE I
Static Game Table

		Defender					
		AB	AC	AD	BC	BD	CD
Attacker	A	-8,8	-8,8	-8,8	1,-1	1,-1	1,-1
	B	-6,6	2,-2	2,-2	-6,6	-6,6	2,-2
	C	4,-4	-2,2	4,-4	-2,2	4,-4	-2,2
	D	12,-12	12,-12	-4,4	12,-12	-4,4	-4,4

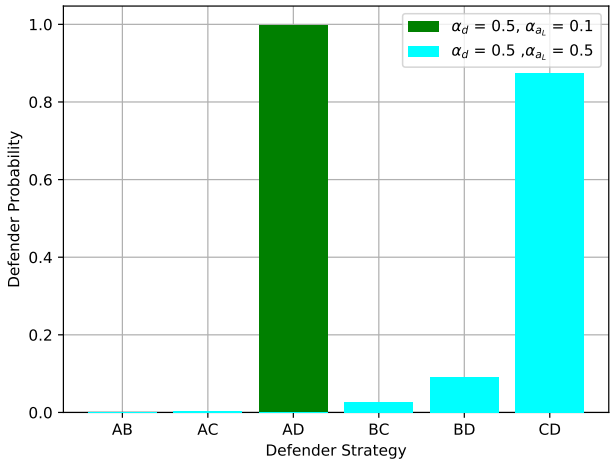

Fig. 2. Defender strategic profile when the defender's rationality $\alpha_d = 0.5$ against attacker's rationalities of $\alpha_{a_L} = 0.1$ and $\alpha_{a_L} = 0.5$.

For the attacker, we let its actual rationality α_{a_A} to be 0.5. Then, we apply the deception problem in (14) to compute the deception rationality, i.e., the rationality the attacker will use in the actual game. The optimal attacker's deception rationality was computed to be of $\alpha_{a_L} = 0.1$. Therefore, the attacker will set $\alpha_{a_L} = 0.1$. Then we run the fictitious play algorithm when $\alpha_d = 0.5$ and when α_{a_L} equals both 0.1 and 0.5. Fig. 2 shows the defender's strategic profile, at equilibrium, for both the attacker's rationalities. From Fig. 2, we can see that when $\alpha_{a_L} = 0.1$, the defender will choose the strategy AD with high probability. However, this changes when the attacker's rationality changes to $\alpha_{a_L} = 0.5$ as the defender will have a more wide probability distribution.

Similarly, Fig. 3 shows the attacker's strategic profile when $\alpha_d = 0.5$ and when α_{a_L} equals both 0.1 and 0.5. We can see that for $\alpha_{a_L} = 0.1$, the attacker chooses the strategy D with high probability. Similarly, when $\alpha_{a_L} = 0.5$, the attacker has a more distributed probability over all their strategies.

We then study the players' utilities calculated from (7) using the equilibrium strategies in Figs. 2 and 3. Fig. 4 shows the players' utilities, for a single batch, in two different circumstances: no deception and deception. During the no deception case, the upper part of Fig. 4, we can see that when the attacker has a rationality of $\alpha_{a_L} = 0.1$, in both the learning stage and the actual game stage, and the defender has a rationality of $\alpha_d = 0.5$, the attacker incurs a utility hit of -3.9614854 per IC. Meanwhile the defender receives a utility of 3.9614854 per IC. Similarly, when the attacker has a rationality of $\alpha_{a_L} = 0.5$, in both stages, the attacker receives

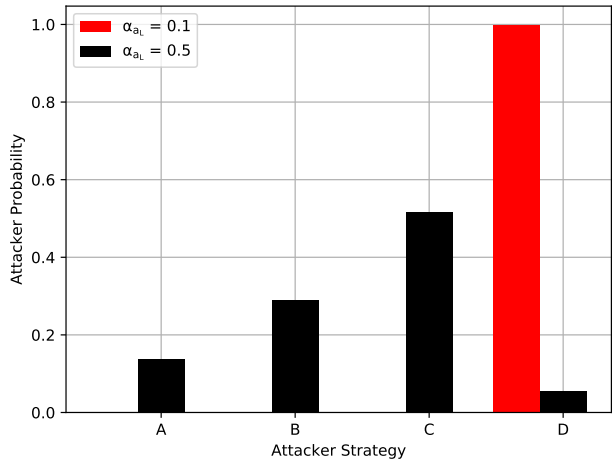

Fig. 3. Attacker strategic profile when the defender's rationality $\alpha_d = 0.5$ against attacker's rationalities of $\alpha_{a_L} = 0.1$ and $\alpha_{a_L} = 0.5$.

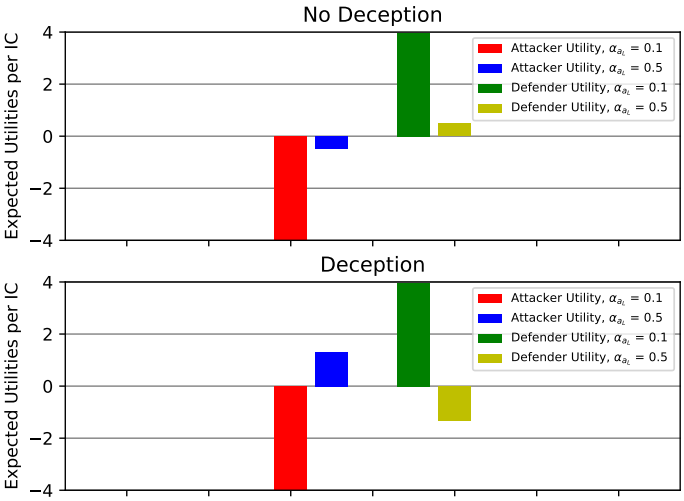

Fig. 4. The attacker's and the defender's utilities under no deception ($\alpha_{a_L} = \alpha_{a_A}$) and deception ($\alpha_{a_L} \neq \alpha_{a_A}$).

a utility hit of -0.4984252 per IC and the defender receives a utility of 0.4984252 per IC. This shows that the attacker will incur a negative outcome when it plays the same rationality in both game stages, no matter whether it played high or low rationalities. This is because, the defender randomizes over its actions and it will be able to detect the hardware trojans with high probability.

On the other hand, under deception, the attacker plays a lower rationality level α_{a_L} during the learning stage. The defender will receive the attacker's strategies corresponding to $\alpha_{a_L} = 0.1$ and thus, it will play the corresponding profile. However, in the actual game stage, the attacker will change its strategy to its actual rationality of $\alpha_{a_A} = 0.5$. This case is shown in the lower part of Fig. 4. In this case, the defender will receive a utility hit of -1.3090019 per IC while the attacker receives a utility gain of 1.3090019 per IC, which represents the attacker's success in deceiving the defender.

The accumulated utility of the attacker from the hypergame model, as defined in (9), is shown in Fig. 5. This utility

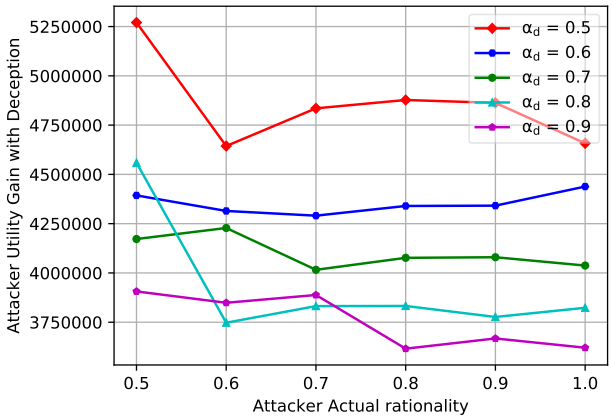

Fig. 5. Attacker's utility gain under different defender's rationalities α_d when the optimal deception rationality is played for each type.

represents the outcome of the attacker when playing its actual rationality in the actual game stage and "playing dumb" in the learning stage. Since this utility depends on both the attacker's rationalities and the defender's rationality, in Fig. 5, we study different scenarios for the players' rationalities and how they affect the attacker's accumulated utility. From Fig. 5, we can see that the attacker's utility gain will always be higher when faced with a defender with a lower or equal rationality. For instance, an attacker with actual rationality of 0.5 will gain the highest utility facing a defender with rationality of 0.5. The same is observed for an attacker with actual rationality of 0.6 versus defenders with rationalities of 0.5 and 0.6. Interestingly, if the defender's rationality is higher than 0.6 , the attacker will still achieve a positive utility gain, but the order becomes less predictable. For instance, an attacker with actual rationality of 0.5 will achieve higher utility against defender of rationality of 0.8 than 0.6. The same happens for the attacker's rationalities of 0.6 and 0.7, as the attacker will achieve a higher utility when the defender has a rationality of 0.9 compared to 0.8.

Another interesting finding in Fig. 5 is that an attacker with higher actual rationality does not, necessarily, achieve a higher utility than an attacker with a lower rationality. For instance, an attacker with actual rationality of 0.6 fares best when facing a defender of 0.7 rationality. This corroborates the importance of the hypergame model to enable the attacker achieving its maximum utility by calculating the deception rationality, which may differ for each actual rationality.

Finally, Fig. 6 studies the utility of different attacker types, different on their actual rationalities α_{a_A}, for their choice of the deception rationalities α_{a_L}. Each curve represents an attacker type and extends from the lowest possible rationality up to the attacker's actual rationality. We can see that the deception rationality of $\alpha_{a_L} = 0.1$ achieves the highest utility gain of any attacker with actual rationality greater than 0.3. Another important finding from Fig. 6, is that an attacker with lower actual rationality, e.g., 0.3 will not benefit much from the deception. Similarly, an attacker with high rationality will maximize its utility if it plays a deception rationality lower than or equal to 0.3. This corroborates the findings from Fig.

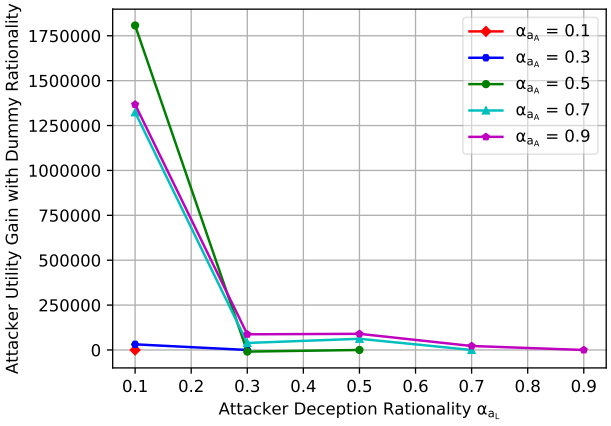

Fig. 6. The utility gain of each attacker's type when they play different deception rationalities α_{a_L} in the learning stage.

1 about the inflection point of 0.37 and its effect on flipping the order of a player's strategies. Note, in Fig. 6, the optimal deception rationality for all the players is 0.1. However, this is not a general case in deception scenarios but rather the solution of (14), which depends on the game's other parameters.

VI. Conclusion

In this paper, we have proposed a novel framework for deception in hardware trojan detection systems. Prospect theory has been used to model the basic players' utilities, without deception, in order to account for different players' rationalities. We then formulated a repeated game in which the defender learns about the attacker's strategies in the learning stage, and then applies this learning knowledge to the subsequent actual game stage. The incentive behind deception has been carefully discussed which is built on the premise of the Prelec function's effect on flipping the order of evaluation between low and high probabilities. We have then formulated an extended view of the game using a hypergame level in which the attacker has a complete view of the game while the defender has a partial perceived view of the game. The hypergame model allows the attacker to optimally determine its deceiving rationality level to maximize its utility gain. We have also proposed a first-step defense against the deception by allowing the defender, while resources permit, to repeat the learning stage in order to mitigate the effects of deception. Finally, we have tested the proposed framework using simulations and the results have shown that the attacker can successfully insert hardware trojans without being detected. Results have also shown the attacker's gain in utility under different combinations of rationalities of the attacker and the defender. For future work, we will focus on building more rigorous defense mechanisms using moving target defense and higher levels of hypergame theory.

References

[1] M. Haselman and S. Hauck, "The future of integrated circuits: A survey of nanoelectronics," *Proceedings of the IEEE*, vol. 98, pp. 11–38, jan 2010.

[2] B. N. Hwang, T. T. Chen, and J. T. Lin, "3PL selection criteria in integrated circuit manufacturing industry in Taiwan," *Supply Chain Management*, vol. 21, no. 1, pp. 103–124, 2016.

[3] J. Dofe, Q. Yu, H. Wang, and E. Salman, "Hardware security threats and potential countermeasures in emerging 3D ICs," in *Proceedings of the ACM Great Lakes Symposium on VLSI, GLSVLSI*, 2016.

[4] R. S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware trojan: Threats and emerging solutions," in *Proceedings - IEEE International High-Level Design Validation and Test Workshop, HLDVT*, 2009.

[5] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy Hardware: Identifying and Classifying Hardware Trojans," *Computer*, vol. 43, pp. 39–46, October 2010.

[6] T. Boraten and A. K. Kodi, "Mitigation of denial of service attack with hardware trojans in noc architectures," in *2016 IEEE international parallel and distributed processing symposium (IPDPS)*, pp. 1091–1100, IEEE, 2016.

[7] S. Sengupta, K. Hong, R. Chandramouli, and K. P. Subbalakshmi, "Spideradio: A cognitive radio network with commodity hardware and open source software," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 101–109, 2011.

[8] A. Eldosouky and W. Saad, "On the cybersecurity of m-health iot systems with led bitslice implementation," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6, IEEE, 2018.

[9] T. Brodeur, P. Regis, D. Feil-Seifer, and S. Sengupta, "Search and rescue operations with mesh networked robots," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 6–12, IEEE, 2018.

[10] A. Eldosouky, A. Ferdowsi, and W. Saad, "Drones in distress: A game-theoretic countermeasure for protecting uavs against gps spoofing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2840–2854, 2020.

[11] W. Saad, A. Sanjab, Y. Wang, C. A. Kamhoua, and K. A. Kwiat, "Hardware Trojan Detection Game: A Prospect-Theoretic Approach," *IEEE Transactions on Vehicular Technology*, 2017.

[12] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic, "Hardware Trojan detection and isolation using current integration and localized current analysis," in *Proceedings - IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2008.

[13] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge university press, 2012.

[14] J. Graf, "Trust games: How game theory can guide the development of hardware Trojan detection methods," in *Proceedings of the 2016 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2016*, 2016.

[15] C. A. Kamhoua, H. Zhao, M. Rodriguez, and K. A. Kwiat, "A Game-Theoretic Approach for Testing for Hardware Trojans," *IEEE Transactions on Multi-Scale Computing Systems*, 2016.

[16] X. D. He and X. Y. Zhou, "Portfolio choice under cumulative prospect theory: An analytical treatment," *Management Science*, 2011.

[17] N. C. Barberis, "Thirty years of prospect theory in economics: A review and assessment," 2013.

[18] A. Sanjab, W. Saad, and T. Başar, "Prospect theory for enhanced cyber-physical security of drone delivery systems: A network interdiction game," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2017.

[19] L. Xiao, N. B. Mandayam, and H. Vincent Poor, "Prospect theoretic analysis of energy exchange among microgrids," *IEEE Transactions on Smart Grid*, 2015.

[20] N. S. Kovach, A. S. Gibson, and G. B. Lamont, "Hypergame theory: a model for conflict, misperception, and deception," *Game Theory*, vol. 2015, 2015.

[21] D. Sklansky, *The theory of poker*. Two Plus Two Publishing LLC, 1999.

[22] A. Petrovic, I. Markovic, V. Koprivica, and B. Bokan, "Tactics factors in chess: Theoretical-empirical aspects,"

[23] D. Prelec, "The Probability Weighting Function," *Econometrica*, vol. 66, no. 3, p. 497, 1998.

[24] A. Eldosouky, W. Saad, and D. Niyato, "Single controller stochastic games for optimized moving target defense," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2016.

[25] P. G. Bennett, "Hypergames: Developing a model of conflict," *Futures*, 1980.

[26] Y. Sasaki, N. Kobayashi, and K. Kijima, "Mixed extension of hypergames and its applications to inspection games," in *Proceedings of the 51st Annual Meeting of the ISSS-2007, Tokyo, Japan*, vol. 51, 2007.