# Foureye: Defensive Deception based on Hypergame Theory Against Advanced Persistent Threats

Zelin Wan, Jin-Hee Cho, *Senior Member, IEEE*, Mu Zhu, Munindar P. Singh, *IEEE Fellow*, Ahmed H. Anwar, and Charles Kamhoua, *Senior Member, IEEE*

## APPENDIX A
## HYPERGAME THEORY

In this section, we briefly discuss hypergame theory which is mainly leveraged to propose the hypergame theoretic defensive deception framework that deals with APT attacks in this work. This section was mainly used in Section 4 of the main paper.

Hypergame theory offers two levels of hypergames that can be used to analyze games differently perceived by multiple players [1]. We adopt first-level hypergames for simplicity. Although hypergame theory applies to multiple players, we consider a game of two players, an attacker and a defender.

### A.1 First-Level Hypergame

Given two players, $p$ and $q$, vectors of their preferences, denoted by $V_p$ and $V_q$, define game $G$ that can be represented by $G = \{V_p, V_q\}$ [1]. Note that $V_p$ and $V_q$ are player $p$'s and player $q$'s actual preferences (i.e., ground truth), respectively. If all players exactly know all other players' preferences, all players are playing the same game because their view of the game is the same. However, in reality, that assumption may fail. Player $p$ can perceive player $q$'s preferences differently from what they are, leading to differences between $p$'s view and $q$'s view. A game perceived by player $p$ based on its perceived preferences about $q$'s preferences, $V_{qp}$, and the game perceived by player $q$ based on its perceived preferences about $p$'s preferences, $V_{pq}$, can be given by

$$G_p = \{V_{qp}\}, G_q = \{V_{pq}\} \tag{1}$$

Hence, the first-level hypergame $H$ perceived by each player is written by $\mathbf{H}^1 = \{G_p, G_q\}$. In a first-level hypergame, analysis is performed at the level of each player's perceived game because each player plays the game based on its belief. Even if the player does not know all outcomes of the game, the outcome can be stable for the player because the player may not unilaterally change its belief. If a game includes an unknown outcome, the unknown outcome is caused by the uncertainty. The stability of an outcome about a game is determined by each player's reaction to the action by the opponent. An outcome is *stable* for $p$'s game if the outcome is stable in each of $p$'s perceived preference vectors, i.e., in each $V_{qp}$. The equilibrium of $p$'s game is determined by the outcome that $p$ believes to resolve the conflict [1].

### A.2 Hypergame Normal Form (HNF)

Vane [5] provides a hypergame normal form (HNF) that can succinctly model hypergames based on players' beliefs and possible strategies of their opponents. HNF is formulated, similar to the normal strategic form in game theory. HNF consists of the following four key aspects: (1) full game; (2) row-mixed strategies (RMSs); (3) column-mixed strategies (CMSs); and (4) belief contexts.

The **full game** is the grid form consisting of row and column strategies, which are associated with the utilities, $ru_{11}, \cdots, ru_{mn}$ and $cu_{11}, \cdots, cu_{mn}$ where $m$ is the number of the row player's strategies and $n$ is the number of the column player's strategies. The full game's grid form $\mathbf{U}$ can be represented by an $m \times n$ matrix with an element $ru_{ij}, cu_{ij}$ for $i = 1, \cdots, m$ and $j = 1, \cdots, n$.

$$\mathbf{U} = \begin{pmatrix} (ru_{11}, cu_{11}) & \cdots & (ru_{1n}, cu_{1n}) \\ \cdots & \cdots & \cdots \\ (ru_{m1}, cu_{m1}) & \cdots & (ru_{mn}, cu_{mn}) \end{pmatrix}, \tag{2}$$

where $R_0$ and $C_0$ denote the full-game strategies by the row and column players, respectively.

**Row-mixed strategies** (RMSs) are $m$ strategies the row player considers based on its belief of the column player's strategies. A player's subgame is defined as a subset of the full game (i.e., a set of all possible strategies by all players) because the player may limit a number of strategies it wants to consider based on its belief. Therefore, depending on a

• *Zelin Wan and Jin-Hee Cho are with the Department of Computer Science, Virginia Tech, Falls Church, VA, USA. Email: {zelin, jicho}@vt.edu. Email: abdullahzubair@vt.edu. Mu Zhu and Munindar P. Singh are with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695. Email: {mzhu5, mpsingh}@ncsu.edu. Ahmed H. Anwar and Charles A. Kamhoua are with the US Army Research Laboratory, Adelphi, MD, USA. Email: a.h.anwar@knights.ucf.edu; charles.a.kamhoua.civ@mail.mil.*

situation, the player can choose a subgame to play. RMSs for the $\kappa$-th subgame a player perceives are given by:

$$\text{RMS}_\kappa = [r_{\kappa 1}, \cdots, r_{\kappa m}], \text{ where } \sum_{i=1}^{m} r_{\kappa i} = 1, \quad (3)$$

where each probability that a particular strategy $i$ is chosen is estimated by player $p$'s belief based on learning from past experience. Since a subgame consists of a subset of strategies in a full game, if a particular strategy $i$ is not in the subgame $\kappa$, the probability for the row player to take strategy $i$ at subgame $\kappa$ is zero, i.e., $r_{\kappa i} = 0$.

**Column-mixed strategies** (CMSs) are a column player's $n$ strategies, believed by a row player for a $\kappa$-th subgame, which are denoted by:

$$\text{CMS}_\kappa = [c_{\kappa 1}, \cdots, c_{\kappa n}], \text{ where } \sum_{j=1}^{n} c_{\kappa j} = 1, \quad (4)$$

where each probability that particular strategy $j$ is chosen is obtained by player $p$'s observations (or learning) towards $q$'s strategies. Similar to the row-mixed strategies, if strategy $j$ is not in subgame $\kappa$, we set $c_{\kappa j} = 0$.

**Belief contexts** are the row player's belief probabilities that each subgame $\kappa$ will be played and are represented by:

$$P = [P_0, \cdots, P_K], \text{ where } \sum_{\kappa=0}^{K} P_\kappa = 1. \quad (5)$$

$P_0$ is the probability that the full game is played where the full game considers all possible strategies of a player based on the ground truth view of a situation. If the row player is not sure of what subgame $\kappa$ to played due to perceived uncertainty, the unknown belief probability is treated simply for the probability that a full game is played, denoted by $P_0 = 1 - \sum_{\kappa=1}^{K} P_\kappa$.

The row player's belief towards the column player's strategy $j$, denoted by $S_h$, is computed by:

$$S_j = \sum_{\kappa=0}^{K} P_\kappa c_{\kappa j} \text{ where } \sum_{j=1}^{n} S_j = 1. \quad (6)$$

The summary of the row player's belief on the column player's $n$ strategies is represented by $C_\sum = [S_1, S_2, \cdots, S_n]$.

### A.3 Hypergame Expected Utility

The hypergame expected utility (HEU) can be calculated based on EU($\cdot$), and the uncertainty probability perceived by the row player, denoted by $g$, representing the level of uncertainty about what is guessed about a given game. $g$ affects the degree of the EU($\cdot$) of a given hyperstrategy by the row player. HEU for the given row player's strategy $rs_i$ with uncertainty $g$ is given by [4]:

$$\text{HEU}(rs_i, g) = (1-g) \cdot EU(rs_i, C_\sum) + g \cdot EU(rs_i, \text{CMS}_w), \quad (7)$$

where $rs_i$ is a given strategy $i$ by the row player. EU($rs_i, C_\sum$) refers to the row player's expected utility in choosing strategy $i$ when the column player can take a strategy among all available strategies $n$. EU($rs_i, \text{CMS}_w$) indicates the row player's expected utility when choosing strategy $i$ when the column player chooses strategy $w$ which gives the minimum utility to the row player. EU($rs_i, C_\sum$) and EU($rs_i, \text{CMS}_w$) are computed by:

$$\text{EU}(rs_i, C_\sum) = \sum_{j=1}^{n} S_j \cdot u_{ij}, \quad (8)$$

$$\text{EU}(rs_i, \text{CMS}_w) = n \cdot S_w \cdot u_{iw} \quad (9)$$

where $g = 0$ means complete confidence (i.e., complete certainty) in a given strategy while $g = 1$ implies that the row player is completely occupied with the fear of being outguessed (i.e., complete uncertainty) [5].

The calculation of EU($rs_i, \text{CMS}_w$) is based on a pessimistic perspective in that when a player is uncertain, it estimates utility based on the worst scenario. In our work, we consider a realistic scenario in that the player will simply choose a random strategy among strategies in a given subgame. For the defender, when it is uncertain, it will simply play a full game because it does not know what subgame the attacker plays. Note that the utility values will be normalized using min-max normalization method [3].

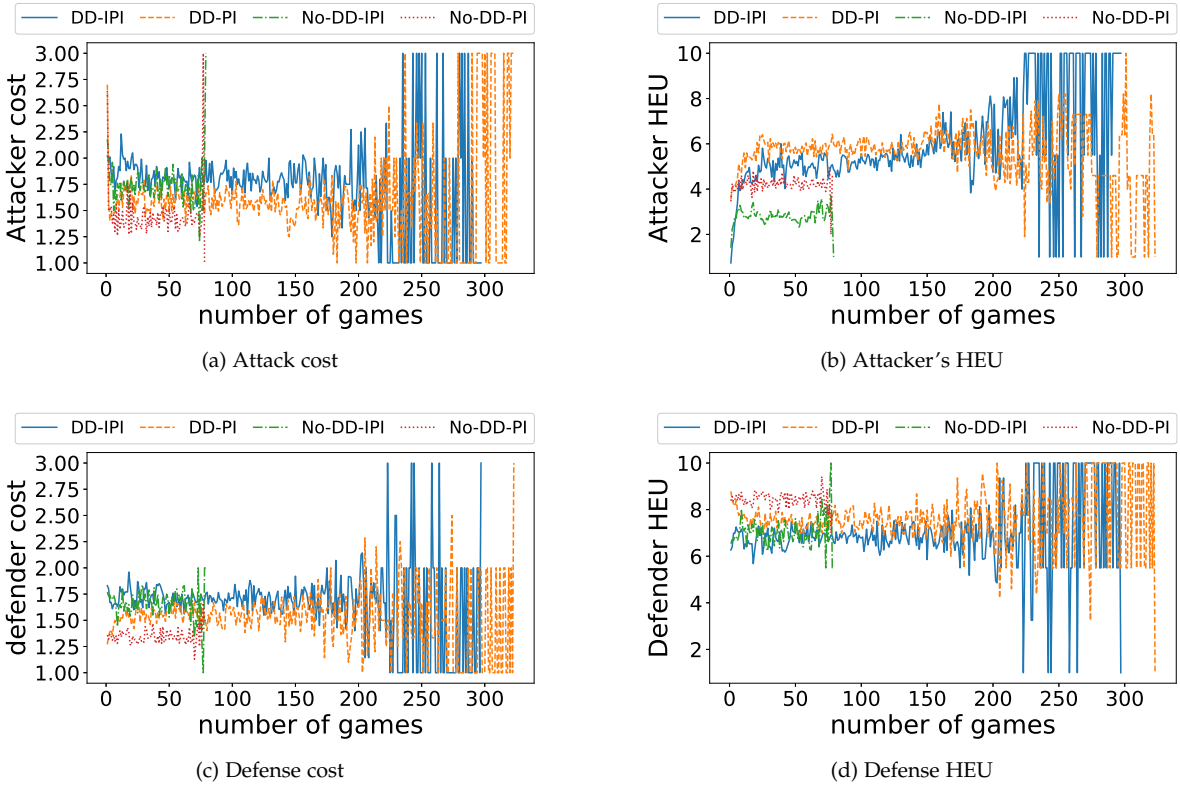
TABLE 1
TABLE OF NOTATIONS

| Symbol | Meaning | Default |
|---|---|---|
| $\text{ac}_k$ | Cost of attack strategy $k$ (main paper) | Table 2 |
| $\text{dc}_k$ | Cost of defense strategy $k$ (main paper) | Table 3 |
| $\rho_1$, $\rho_2$ | Thresholds for SF in Eq. (8) in the main paper | 1/3, 1/2 |
| $N_{\text{LH}}$ | Number of low-interaction honeypots deployed but not activated in a network | 50 |
| $N_{\text{HH}}$ | Number of high-interaction honeypots deployed but not activated in a network | 25 |
| $N_{\text{WS}}$ | Number of Web servers deployed in a network | 25 |
| $N_{\text{DB}}$ | Number of databases deployed in a network | 25 |
| $N_{\text{IoT}}$ | Number of IoT nodes deployed in a network | 450 |
| $N$ | Total number of nodes | 500 |
| $nv_i$ | Total number of security vulnerabilities of node $i$, including encryption (5), software (5), and unknown (1) | 11 |
| $P^r$ | Probability of two nodes being connected in an Erdös-Rényi random network | 0.05 |
| ad | An attacker's deception detectability | $[0, 0.5]$ |
| $\lambda$, $\mu$ | A constant for normalization for the attacker's uncertainty and defender's uncertainty, respectively | 0.8, 8 |
| $P_{fp}$, $P_{fn}$ | Probabilities of false positives and false negatives in the NIDS | 0.01, 0.1 |
| $\text{Th}_{risk}$ | Risk threshold used in Eq. (6) in the main paper | 0.2 |
| $\text{Th}_c$ | The threshold used in $AS_8$ (Data exfiltration) | 150 |
| $\epsilon_1$, $\epsilon_2$ | Increased or decreased percent of a given vulnerability probability by taking attack strategies (i.e., $AS_5$, $AS_7$) or defense strategies (i.e., $DS_1 - DS_3$) | 0.1, 0.01 |
| $P_{fake}$ | Probability the attacker obtains a fake key | $1 - \text{ad}$ |
| $c_{NT}$ | Percentage (%) of edges that are hidden by defense strategy $DS_8$ | 20 |

## APPENDIX B
## ADDITIONAL EXPERIMENTAL RESULTS

### B.1 Strategy Cost and Hypergame Expected Utility of the Attacker and Defender

Fig. 1 shows the attack cost, defense cost, and the HEUs of the attacker and defender, respectively, with respect to the number of games played between the attacker and defender when the default setting is used based on Table 1. Note that we only showed the number of games from the 2nd game for meaningful analysis. The main reason of high fluctuations in later games is because only a small number of simulation runs have long system lifetimes, resulting in high variances. We can clearly see the similar trends observed in Fig. 2 of the main paper in that No-DD-based




(a) Attack cost

(b) Attacker's HEU

(c) Defense cost

(d) Defense HEU

Fig. 1. Attack cost and defense cost along with the attacker's hypergame expected utility (AHEU) and the defender's HEU (DHEU). The performance is shown from the second game.

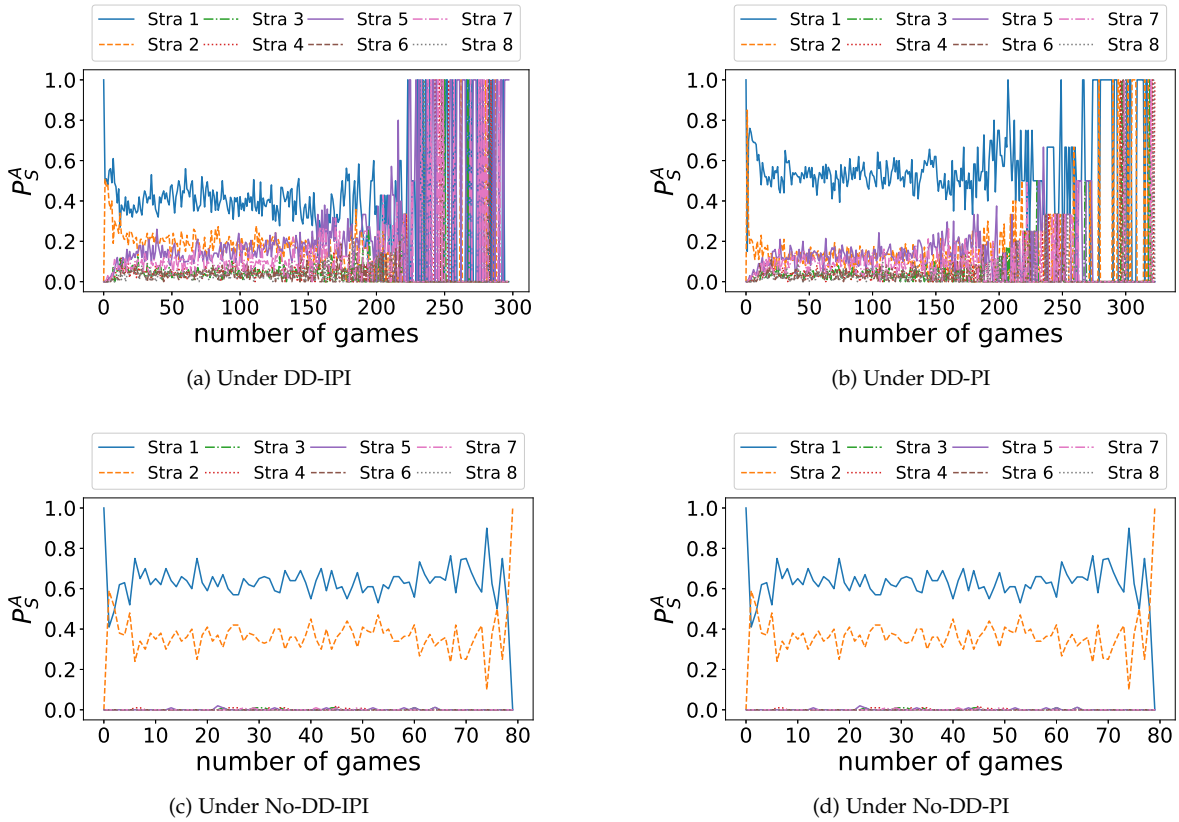
schemes have shorter lifetimes while DD-based schemes show much higher system lifetime, which was measured based on MTTSF in Fig. 4 of this document. Aligned with the trends observed in Fig. 2 of the main paper, in Fig. 1a, we can observe that when DD-IPI is used, the attacker incurs higher cost than other schemes as it is less likely to choose an optimal strategy, which is cost-effective, due to the confusion or uncertainty introduced by DD-IPI. But under DD-PI, by taking the benefit of perfect information (PI) available, the attacker can take a better action incurring less cost. When No-DD is used, the attacker does not have to use various attack strategies that can be useful as an insider attack because it is less likely to be the inside attacker due to the immediate eviction by the NIDS. In addition, there is no chance for the system to intentionally allow them to be in the system for collecting further attack intelligence. Hence, the attacker can use a limited set of attack strategies that do not incur high cost.

In Fig. 1b, we demonstrated the attacker's hypergame expected utility (AHEU) with respect to the number of games played between the attacker and the defender in the default setting. Under DD-based schemes, when PI is used, higher AHEU is obtained. On the other hand, imperfect information (IPI) hinders the attacker to choose optimal strategies, leading to less AHEU. This was indirectly exhibited that the proposed DD strategies under uncertainty were effective to confuse the attacker. When No-DD is used, PI helps the attacker to make better attack decisions than IPI.

Similarly we demonstrated the defense cost in Fig. 1c and the defender's HEU (DHEU) in Fig. 1d. Under DD-based schemes, in terms of the data availability with respect to the number of games, the system lifetime is observed longer than under No-DD-based schemes with the same reasons explained in the attack cost and AHEU as above. As expected, PI helps the defender to choose better strategies due to no uncertainty perceived than IPI. However, using DD strategies introduces additional cost to achieve better security than using No-DD-based strategies. Hence, DD-IPI incurs minimum DHEU overall. However, note that this doesn't mean DD-IPI has less benefit in system security; rather it implies that there is cost to achieve enhanced security.

### B.2 Probabilities of Attack Strategies

Fig. 2 shows the probabilities of each attack strategy taken, denoted by $P_S^A$, with respect to the number of games played between the attacker and defender under the four schemes. Under all the four schemes, attack strategy 1, $AS_1$ (scanning attack), is dominantly taken. This is because every attacker starts from scanning a target system in the stage of reconnaissance (R) in the cyber kill chain (CKC), which is the first step of the advanced persistent threat (APT) attacks. In addition, due to the presence of the NIDS, when the attacker successfully penetrated into the system, it is highly likely for the attacker to be detected by the NIDS with fairly high detection rate (i.e., $P_{pf} = 0.01$ and $P_{pn} = 0.1$). In addition, only a compromised node or an attacker, which are not detected by the NIDS or passed the risk threshold ($\mathrm{Th}_{risk}$; see Eq. (6) in the main paper), can only remain in the system. Hence, there won't be many insider attackers compared to outsider attackers. Hence, observing the highest probability using scanning attack ($AS_1$) is natural. In




(a) Under DD-IPI

(b) Under DD-PI

(c) Under No-DD-IPI

(d) Under No-DD-PI

Fig. 2. The probabilities of attack strategies under the four schemes. The performance is shown from the second game.

addition, as the attacker tries to get into the system by using social engineering attacks ($AS_2$) exploiting both software and encryption vulnerabilities, it is reasonable to observe the attacker taking $AS_2$ as the second most attack strategy. In Figs. 2a and 2b, when DD strategies are used, $AS_5$ and $AS_7$ are commonly taken. This is because these two attack strategies, $AS_5$ and $AS_7$, relatively incur less cost (1 and 2 for attack costs, respectively). When No-DD strategies are used, it is reasonable to observe that the attacker mainly uses $AS_1$ and $AS_2$ as it does not have many chances to use other strategies due to being detected by the NIDS. When PI is used, the defender also knows more about the attacker due to no uncertainty. This makes the attacker being evicted quicker and more new attackers are likely to attempt to access the defense system. Hence, the attacker uses social engineering attacks ($AS_2$) more frequently in the later games when PI is used than when IPI is used.

### B.3 Probabilities of Defense Strategies

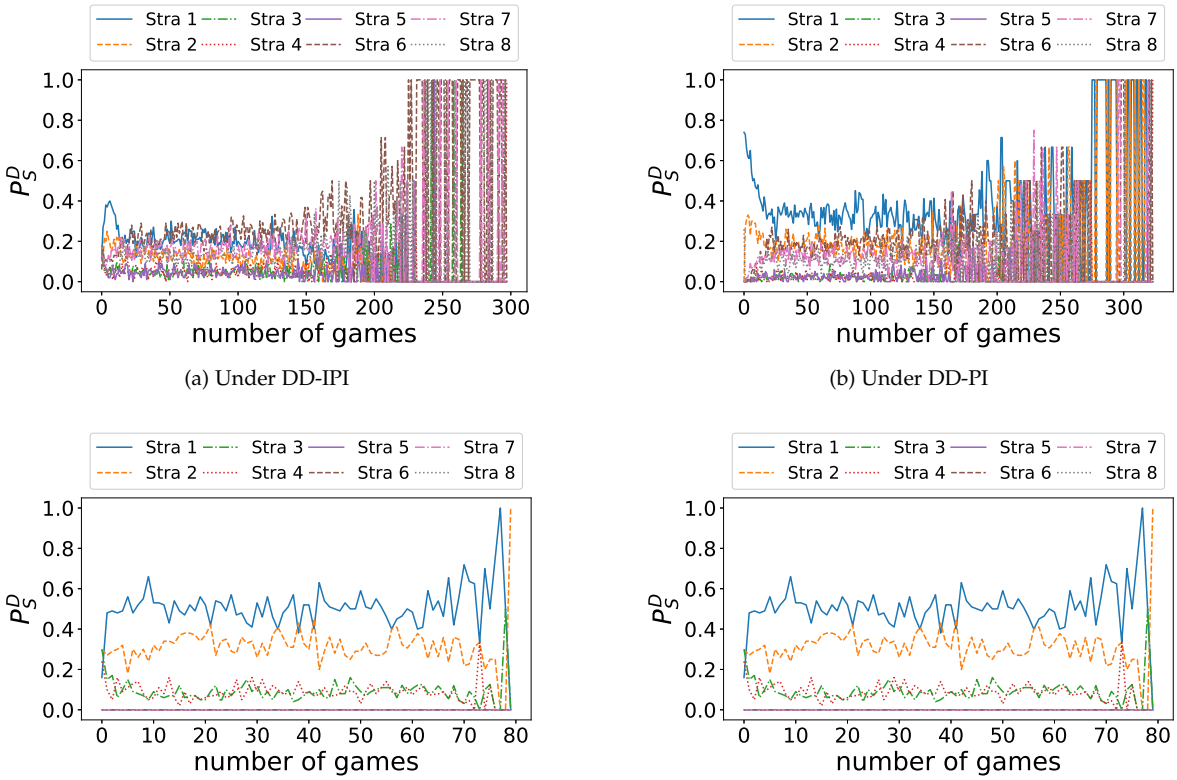
Fig. 3 shows the probabilities of the eight defense strategies ($P_S^D$) used under the four schemes with respect to the number of games played between the attacker and defender. Under all the four schemes, defense strategy 1, $DS_1$ (firewall), is dominantly used as it deals with outside attackers. However, when DD-IPI is used, $DS_2$ (patch management) and $DS_6$ (honey information) were used more commonly compared to other defense strategies. This is because these two defense strategies cost less, which makes the corresponding DHEU higher, ultimately leading the defender to choose $DS_2$ and $DS_6$ more often than other strategies. When No-DD strategies are used, $DS_1$ and $DS_2$ are mainly used while $DS_3$ and $DS_4$ are marginally used.

### B.4 TPR of the NIDS With Respect To the Number of Games

Fig. 4 shows the TPR of the NIDS with respect to the number of games played between the attacker and defender. As expected, since DD-based schemes allow the defender to learn additional attack intelligence which is considered in the NIDS, this naturally leads to the improvement of the TPR in the NIDS.

## Appendix C
## Representations of Modeled Attack and Defense Strategies, HEUs, and NIDS using the Mind Map

For Figs. 5-8, we demonstrated the Mind Maps on how attack and defense strategies are designed, how AHEU and DHEU are estimated, and how the NIDS operates in the given system. These Mind Maps are based on Eqs. (4)-(9) in this supplement document and Eqs. (9) and (10) in the main paper. For Fig. 8, we described the workflow on how the NIDS operates in the considered system where each number refers to an execution step. We demonstrated these Mind Maps in order for readers to clearly follow the algorithms used in this work for easy reproducibility. For the demonstrated Mind Maps, we used the Mind Maps tool called the MindNode [2].




(a) Under DD-IPI

(b) Under DD-PI

(c) Under No-DD-IPI

(d) Under No-DD-PI

Fig. 3. Probabilities of defense strategies under the four schemes. The performance is shown from the second game.

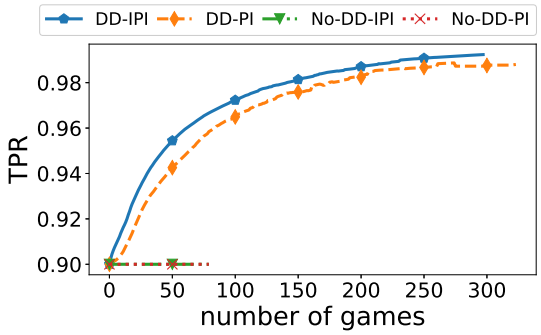



Fig. 4. True positive rate (TPR) of the NIDS.

## REFERENCES

[1] N. M. Fraser and K. W. Hipel, *Conflict Analysis: Models and Resolutions*. North-Holland, 1984.

[2] I. GmbH. MindNode. [Online]. Available: https://mindnode.com/

[3] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Elsevier, 2011.

[4] R. Vane, *Hypergame theory for DTGT agents*. AAAI, 2000.

[5] ——, "Advances in hypergame theory," in *Proc. AAMAS Workshop on Game-Theoretic and Decision Theoretic Agents*, 2006.

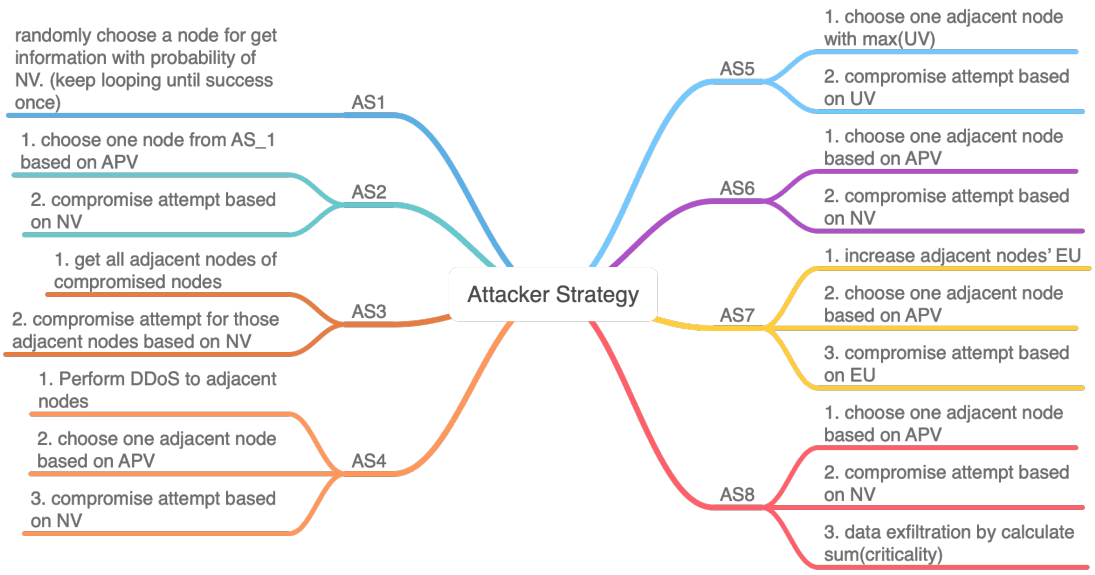



Fig. 5. Modeling Attack Strategies.

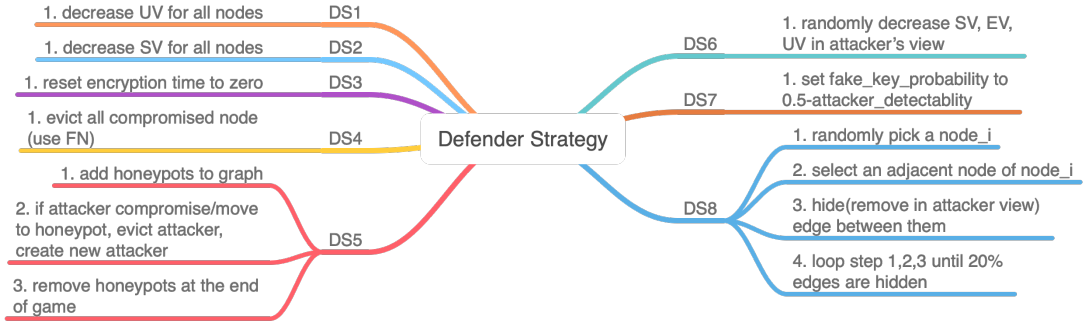



Fig. 6. Modeling Defense Strategies.

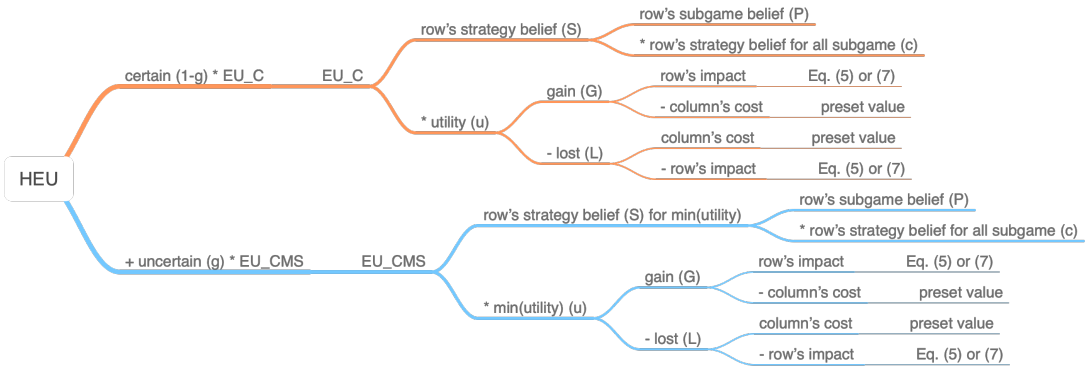



Fig. 7. Modeling HEU.

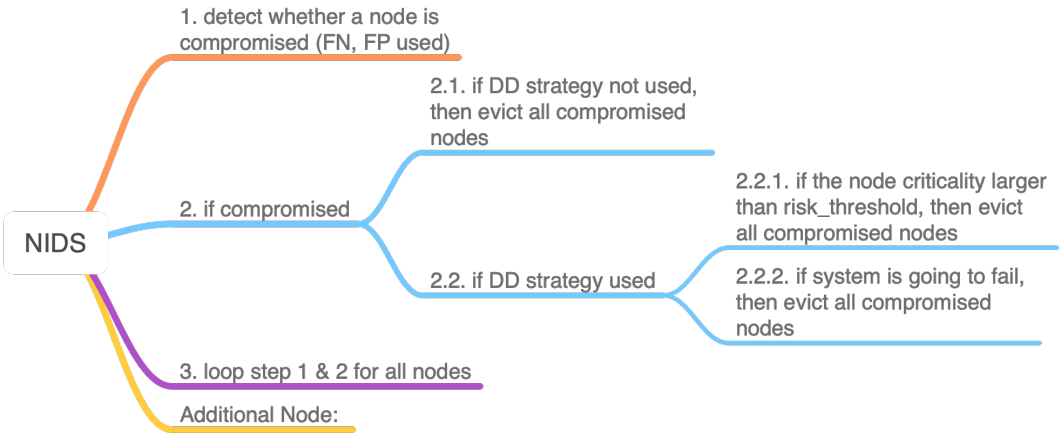



Fig. 8. Modeling NIDS.