Covert Channels in Combinatorial Games

Philip C. Ritchey
Dept. of Computer Science
Purdue University
West Lafayette, IN
pritchey@cs.purdue.edu

Vernon J. Rego
Dept. of Computer Science
Purdue University
West Lafayette, IN
rego@cs.purdue.edu

ABSTRACT

A general framework for exploiting covert channels in combinatorial games is presented. The framework is applicable to all combinatorial games, including Chess and Go, but is applied to the game of Tic-Tac-Toe for ease of experimental analysis. The security and capacity of the resulting covert channel are analyzed experimentally. By considering the ways in which a passive adversary can attempt to detect and neutralize the usage of the channel, it is shown that the passive adversary cannot distinguish games which contain hidden information from games which do not. It is also shown that, even by enforcing a perfect-play requirement, the adversary cannot reduce the capacity of the channel to zero in order to prevent covert communication. Additionally, the framework is shown to be generalizable to multiplayer games and games without perfect information by identifying covert channels in two other games.

Categories and Subject Descriptors

I.6.8 [**Simulation and Modeling**]: Types of Simulation; D.2.11 [**Software Engineering**]: Software Architectures—*Information Hiding*

General Terms

Theory

Keywords

Covert Channels, Information Hiding, Steganography, Cover Generation Methods, Combinatorial Games, Online Games, Security, Capacity

1. INTRODUCTION

Steganography is the art and science of hiding secrets in innocuous-seeming objects in such a way that the existence of the secret is undetectable to a third party adversary or observer. There is evidence that steganography has been in use for over 2000 years [7]. Today, there are many different ways of hiding information, which Johnson and Kaztenbeisser have grouped into six categories [6]. Of these, cover generation methods have received little attention compared to the other five categories.

The category of cover generation methods is for those techniques which hide information in generated cover-objects, often in the structure of the object. In contrast to other types of steganographic techniques, cover generation methods do not augment an already existing object in order to hide bits of information. The generation of novel objects is not restricted to the field of information hiding, however. Techniques in the field of computer graphics for procedurally generated media are also concerned with generating novel content which mimics some target object, such as trees and cities. When applied to information hiding, the goal is similar: to generate objects which closely resemble the target object. However, cover generation methods for steganography have a requirement that procedurally generated media do not generally have, which is *reversibility*. The generated objects must be deconstructible in order to recover the hidden bits of information. There is a compromise which can be made to mitigate the difficulty of satisfiying this requirement in practice: use a target for which the generation process is fully observable and therefore recordable. In this way, even if the end product is not reversible when taken by itself, the fact that there exists a transcript which recorded the steps taken by the process means that deconstruction is either unnecessary, or made possible when given the transcript.

Such processes can be found in the field of combinatorial games. A combinatorial game is a two-player game with perfect information and no chance moves [3]. Chess and Go are two examples of combinatorial games which have already been identified by other authors as being capable of supporting covert channels. Desoky and Younis [2] and Lange [8] present schemes for covert communcation via chess. Hernandez-Castroa *et al.* present a general methodology for steganography in games and apply it to the game of Go [5]. However, these papers do not address the possibilty that the games can be played naturally (as if by a human) and still transmit information.

The motivation for researching covert channels in games comes not only from the possibility of such channels, but also from the advantages which can be gained by exploiting them. For instance, Murdoch and Zielinski conducted a study which proved that a covert channel in the game of Connect Four could be used to collude with other players for the purposes of winning a league-style tournament [9]. How-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DISIO Workshop 2012, March 19-23, Desenzano del Garda, Italy
Copyright © 2012 ICST 978-1-936968-47-3
DOI 10.4108/icst.simutools.2012.247733

Simutools 2012, March 19-23, Desenzano del Garda, Italy
Copyright © 2012 ICST
DOI 10.4108/icst.simutools.2012.247733

ever, there is no reason to believe that the aim of colluding through a game should be limited to winning a tournament of that game.

In this paper, the general task of exploiting covert channels in combinatorial games is approached by examining the game of Tic-Tac-Toe. For ease of exposition, a standard framework for studying covert channels, known as the Prisoners' Problem [12] is drawn upon. In Section 2, the intial conditions of Alice, Bob and Wendy's situations and a general methodology of exploiting covert channels in combinatorial games in presented. The methodology is demonstrated in Section 3 by applying it to the game of Tic-Tac-Toe. Experimental results on the capacity and security of the Tic-Tac-Toe stego-system are presented in Section 4. Section 5 mentions other games which have not been covered in the existing literature to which the general theory can be easily applied. Section 6 concludes the paper with a summary.

2. CHANNEL AND METHODOLOGY

Alice and Bob are engaged in a series of communications in which they are playing a simple two-player game, likely more than once. The channel is public but trusted. Wendy is a passive adversary who observes all communucation between Alice and Bob looking for messages which contain secret information. Alice and Bob do not want Wendy to learn their secret, but messages which appear to be encrypted are prohibited on public channels. So, Alice and Bob decide to use steganography to hide their encrypted communication. Messages containing the game data may not be the only communication they send and receive, but such messages are sufficient for Alice and Bob to exchange information covertly. Wendy's task is to distinguish between when Alice or Bob are hiding information and when they are not.

In [5], Hernandez-Castro *et al.* describe an approach to hiding data in games which is built upon in this work. Each player has a set of strategies which specify the move that player should make, given the action sequence leading up to that player's turn. There may be multiple strategies defined for any given action sequence, allowing a player to choose between a set of moves. The set of moves is assumed to be ordered according to the expected payoff of the move.

Under typical playing conditions, each player is expected to respond to the given action sequence with the best move from their set of strategies. However, the selection of a specific move from the set of all possible moves can be used to send information to the other player or an observer. In general, a player could send up to $\lfloor log_2(N) \rfloor$ bits of data per move, where N is the number of possible moves available to the player for that move.

The process of playing a game for the purpose of establishing covert communication can be viewed as a cover generation method. The cover being generated is a sequence of moves in a combinatorial game. An example of the cover generation embedding process for a combinatorial game is shown in Figure 1. At each step, Alice uses her set of strategies to construct a set of moves. To each of these moves, she assigns a bit string and selects the move whose bit string matches the next piece of data she wants to send. By making that move, she signals to Bob what the next piece of secret data is. The board is updated and sent to Bob who can reverse Alice's last move to figure out what she sent.

There are several techniques that Alice and Bob could use to protect their hidden data and help the cover generation

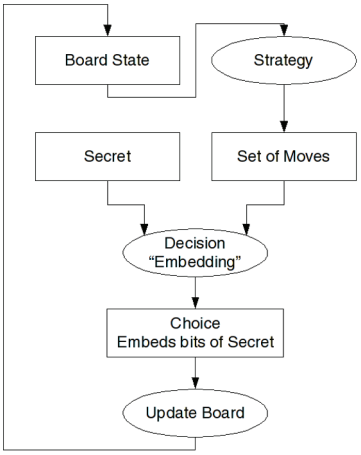

Figure 1: Covert Channels in Combinatorial Games

process to hide the data. One sych technique is to compress and encrypt the data, making it smaller and more random. We can view the cover generation process, as Sallee does in [11], as being a mapping from random bit sequences to valid elements of some cover-space. The hypothesis is that the more random the message appears, the more convincingly diverse the generated objects will be. A second technique is to use a shared random number generator and a seed to randomly permute the move labels at each step. This provides an additional layer of security by acting as a variable length substitution cipher so that any given move does not always encode the same data. Another technique is to use the channel only occasionally, either on a game-by-game or move-by-move basis, which can be accomplished by the usage of the same or another shared RNG and seed. This forces Wendy to decide when individual games or moves are part of the covert channel and when they are not and is similar to the technnique of Chaffing and Winnowing [10].

3. TIC-TAC-TOE

Tic-Tac-Toe is a two-player turn-based pencil-and-paper game played on a 3x3 grid, wherein players attempt to place three marks in a row in order to win. Each player uses marks of a single kind, player-1 uses `X` and player-2 uses `O`, and player-1 usually goes first.

If Alice and Bob wanted to play Tic-Tac-Toe, they could do so by sending a piece of paper back and forth between their cells (always through Wendy) and take turns making their marks. They could also use computers and play the game through a digital medium, such as images or text. The specific communication medium is inconsequential, as Alice and Bob can use the game of Tic-Tac-Toe to communicate covertly in any medium. This section will detail how the game of Tic-Tac-Toe can be used as a covert channel and will explore ways in which Wendy can attempt to detect and neutralize the usage of the channel.

3.1 Tic-Tac-Toe as a Covert Channel

The essence of a covert channel in a combinatorial game is that Alice must make a choice of which move she will make during her turn and that her choices can be used to embed

hidden information into the game. When using Tic-Tac-Toe, Alice will encode her secret information as mark locations in the Tic-Tac-Toe grid.

Assume that Alice goes first and that her strategy simply tells her that all moves are equally good so that, during her turn, she can choose to place her mark at any open location. On her first turn, Alice has nine choices of where to place her mark, thus her placement encodes three bits. Bob then makes his move, leaving Alice with seven locations for her second mark. So, with her second move, Alice can send two bits . Again, Bob makes his move and Alice is left with five locations for her mark, allowing her to send two more bits. At this point, Alice has sent seven bits. If she did not win with this move, it is possible that Bob will win with the next move. However, if Bob does not make a winning move during his turn, Alice gets to place another mark in one of the three places Bob has left for her and she can encode one more bit with her final choice. This is the last bit Alice can send since, even if Bob cannot win during his turn and she gets to place her last mark, there will only be one location in which she can place it and thus she cannot encode any additional information.

Alice is guaranteed to be able to send seven bits under this strategy, since neither Alice nor Bob can win the game until Alice has placed her third mark. Furthermore, the exchange between Alice and Bob can be used to simultaneously transfer information in both directions. Assuming that Bob's strategy also lets him play anywhere that is open, his first move has eight choices and encodes three bits. His second choice is from six options, encoding two bits. If he gets a third turn, he will choose from four options and encode another two bits. If he gets yet another turn, his final choice between the two remaining spaces will encode one additional bit. Bob is guaranteed to be able to send five bits, with high likelihood of being able to send seven or eight bits. When not using his channel capacity, Bob can allow Alice to send an eighth bit by forcing a draw since it is always possible for either player, playing perfectly, to force a draw in Tic-Tac-Toe [1].

3.2 Properties of the Channel

The most important properties of an information hiding system are steganographic security, robustness and capacity. The security of an information hiding system is a measure of the indistinguishability of its stego-objects from clean cover-objects. The robustness of the system is a measure of the stego-objects' resistance to tampering. And the capacity of the system is a measure of the amount of secret information the stego-objects can hold.

Three strategies of play are identified and their capacity and security against Wendy, a passive adversary, is analyzed. The first strategy is Random, which allows the player to place their mark in any open grid cell. The next is Greedy, which limits the player to winning or blocking when possible, but is otherwise equivalent to Random. The last is Optimal, which limits the player to those moves which lead to the best outcomes, winning, if possible, or else drawing.

Since the public channel is trusted and Wendy is a passive adversary, Alice and Bob know that what they send to the other arrives unchanged. Therefore, the possibility of Wendy tampering with the stego-objects directly and dropping or forging messages can be ruled out. Thus, robustness becomes a mute point and can be safely skipped, for now.

3.2.1 Security

For Tic-Tac-Toe, steganographic security means that Alice and Bob want Wendy to have as much difficulty as possible distinguishing between games of Tic-Tac-Toe which are hiding information and games which are not. Since, by Shannon's maxim that the enemy knows the system, one way for Wendy to test for hidden information is to assume that all games contain stego and attempt to extract the hidden data. If the extracted data is not garbage, Wendy will mark the games as stego, otherwise, they will be marked as clean. However, if Alice and Bob have compressed or encrypted their data, Wendy will need additional secret information to check if the data she extracted is truly garbage. Additionally, Wendy may also need to determine which games or moves she should use when extracting data, which would again require knowing secret information. If Alice and Bob can keep their keys secret, then Wendy has no hope of using this method for detection.

Another way Wendy can test for hidden information is to assume that the presence of hidden information will cause the game or move to differ from the expectation for that move or game. That is, Wendy will test whether the moves Alice and Bob make agree with her model of how the game should be played. In this case, Wendy's model is a set of strategies for playing the game which dictate which moves are expected and which are not. If Wendy sees a move which she did not expect she will flag the game as stego. The simplest such model for Wendy is to check for obviously bad moves. To avoid this, Alice and Bob can update their strategy to Greedy, so as to no longer make any obviously bad moves. In response, Wendy can update her model to reflect an expectation of perfect gameplay, possibly on the assumption that perfect gameplay does not leave room for covert communcation. Under Greedy, Alice and Bob will occasionally be forced to make a suboptimal move in order to send some data, which Wendy will catch and use as evidence of the existence of a covert channel. As with Random, Greedy will occasionally cause Alice and Bob to play suboptimally even when not sending data. However, Alice and Bob can further upgrade their strategy to Optimal in order to avoid making the mistakes for which Wendy is looking. Additionally, it turns out that Wendy's assumption that perfect play precludes covert communication is false; Alice and Bob can play perfectly and still have covert channel capacity available.

3.2.2 Capacity

The capacity of the covert channel depends on the play-strategy that Alice and Bob use. Their play-strategy, as shown in the experimental results of Section 4 affects the security that Alice and Bob have against Wendy's analysis. Analytical results on the channel capacity and game length under different strategies are given in Table 1. The means are taken over all possible action sequences (game paths, or histories) for games played under each strategy.

3.3 Algorithms

There are two general functions, Embed and Extract, which are neccessary to carry out communcation over covert channels in combinatorial games. Algorithm 1 contains the embedding process which is carried out on each turn to send data through the channel and Algorithm 2 shows the extraction process which is carried out on each turn to receive

Table 1: Analytical Tic-Tac-Toe Channel Capacity

Strategy	Total Capacity (Min. , Max.)	Mean Capacity (Alice, Bob)	Mean Game Length
Random	(12 , 16) bits	(7.974 , 7.774) bits	8.255 moves
Greedy	(8 , 16) bits	(7.455 , 6.650) bits	8.7398 moves
Optimal	(6 , 15) bits	(7.489 , 4.326) bits	9 moves

data from the channel.

To embed data in a game, the set of best moves are computed for the given board. The size of the returned set is used to compute the number of bits of information which will be consumed in this turn. The set of best moves is randomly permuted using a shared random number generator and seed. The embedding is accomplished by extracting and consuming the correct number of bits from the secret and using them as the index value into the permuted set of best moves. The chosen move is made and a new updated board is returned, along with the updated seed and message.

The first several steps of the extraction process are identical to those of the embedding process: determining the best moves given the board your opponent saw, permuting the set of best moves and determining the number of bits to expect. Comparing the old board to the new reveals which move the opponent selected, the index of which is converted to a binary number containing the correct number of bits. The result is returned as the next fragment of the message being received.

In practice, each turn would consist of an attempt to read data from the channel followed by an attempt to write data to the channel. The determination of when reads and writes contain data or garbage is made external to the algorithms presented here.

Algorithm 1 Embedding process: covert channels in combinatorial games

Input: Board B, Message M, Seed S
Output: Board newB, Message newM, Seed newS
$bm \leftarrow GetBestMoves(B)$
$n \leftarrow \lfloor \log_2(|bm|) \rfloor$
$bm \leftarrow RandomPermute(bm, S)$
$newS \leftarrow RNG(S)$
$m \leftarrow M[1:n]$
$newM \leftarrow M[n+1:end]$
$md \leftarrow toDecimal(m)$
$move \leftarrow bm[md]$
$newB \leftarrow applyMove(move, B)$
return $newB$, $newM$, $newS$

4. EXPERIMENTAL RESULTS

Nine parameter sets for Alice and Bob are made up of a choice between three game-playing strategies and a choice between three embedding techniques. The three game-playing strategies Alice and Bob can use are Random, Greedy and Optimal. Under the Greedy strategy, it is still possible for Alice or Bob to send data when forced to play "smart", such as in the case of Tic-Tac-Toe when a player can win or block in more than one location, providing a choice and thus capacity for covert communication. The Optimal strategy uses the Negamax algorithm [4] to compute the set of best moves as those which maximize the current players pay-off, which is the negation of the opposing players pay-off since Negamax assumes that the game is zero-sum. Whenever there is more than one best move, a player has the opporuntity to send hidden information through the channel. The three embedding techniques that Alice and Bob use are Every Game, Random Games and Random Moves. The Every Game technique causes Alice and Bob to attempt to send data on every move of every game, in order to maximize the utilization of the channel. Using the Random Games technique, Alice and Bob will only send (and expect to receive) data during games chosen at random by a random number generator and seed shared by both Alice and Bob. Every move of a chosen game is used for data transfer. When using the Random Moves technique, Alice and Bob only send and receieve data on moves chosen at random by their shared random number generator.

Algorithm 2 Extraction process: covert channels in combinatorial games

Input: Board oldB, Board newB, Seed S
Output: MessageFragment m
$bm \leftarrow GetBestMoves(oldB)$
$n \leftarrow \lfloor \log_2(|bm|) \rfloor$
$bm \leftarrow RandomPermute(bm, S)$
$newS \leftarrow RNG(S)$
$move \leftarrow extractMove(oldB, newB)$
$md \leftarrow find(bm == move)$
$m \leftarrow toBinary(md)$
return m

4.1 Capacity Results

Figure 2 contains the mean embedding capacity results for different values of $P(embed)$, the likelihood that a game or move is chosen to be used for transmitting data, for each of the different embedding techniques when applied to Tic-Tac-Toe. Only the Random Games and Random Moves techniques are shown because the Every Game technique does not depend on $P(embed)$ and is equivalent to the capacity measured when $P(embed) = 1$, which is shown on both plots.

In the result for the Random Games technique we can see that Alice, playing randomly for full utilization of the channel, has the highest capacity followed by Bob playing randomly, both being between seven and eight bits per game when $P(emebed) = 1$. Next, Alice playing Greedy and Alice playing Optimal have very similar capacities, just above six bits per game at most. Greedy Bob sends just under five bits per game while Optimal Bob sends just under three bits per game. The embedding capacity for each player increases linearly with the value of $P(embed)$. The results for the Random Moves technique, which are very similar to those

of the Random Games technique, shows that the embedding capacity is not much affected by the choice between Random Games and Random Moves. One reason for the lack of effect is that embedding in $p\%$ of games is, on average, equivalent to embedding in $p\%$ of moves.

4.2 Security Results

Figure 3 displays the results of the security tests for the two embedding techniques, Random Games and Random Moves. In these tests, the different parameter sets for Alice and Bob's stego-system are pitted against two Wendys, each with a strategy for determining which games are suspected of containing hidden information, and Wendy's mean error rate is recorded. Wendy makes an error whenever she flags a clean game as dirty (false positive) or a dirty game as clean (false negative). Alice and Bob's security is inversely related to Wendy's error rate; the more errors Wendy makes, the more secure Alice and Bob's covert channel is because Wendy has more difficulty distinguishing clean exchanges from those containing hidden information. The Greedy Wendy makes her decision by checking for moves in which Alice or Bob failed to make an obviously good move when they had the chance. If a game contains such a move, it is classified as suspicious. The Optimal Wendy makes her decision by checking for non-optimal play, that is any move which is not in the set of best moves given the state of the game.

The cases of Greedy Wendy against Random Alice and Bob, Optimal Wendy against Random Alice and Bob and Optimal Wendy against Greedy Alice and Bob all have error rates which decrease linearly with $P(embed)$. When no embedding is taking place, at $P(embed) = 0$, Wendy has a high error rate because she is mistaking Alice and Bob's poor play for covert communication, resulting in mostly false positives with a few true negatives whenever Alice and Bob get lucky to stumble into what appears to be a clean game. When $P(embed)$ is near 1, Wendy has a low error rate because her propensity for flagging games as dirty is matched by the proportion of games which do contain hidden information. The opposite trend occurs for the cases of Greedy Wendy against Greedy Alice and Bob, Greedy Wendy against Optimal Alice and Bob and Optimal Wendy against Optimal Alice and Bob. In these cases, a low value of $P(embed)$ results in Wendy having a low error rate because Alice and Bob's games are well-played and do not confuse Wendy. As the embedding frequency increases, so too does Wendy's error rate. This is because the games are not looking any less well-played despite the increasing frequency of covert transmission. When Alice and Bob are sending bits on every move of every game, Wendy's error rate is 100% because the games still look well-played according to her decision making process.

In Figure 3, the Random Moves technique has interesting results for security. We see the same general trend of Wendy's error rate decreasing as the rate of transmission increases when Wendy has the better model and increasing when Alice and Bob have an equivalent or better model. However, the relationship is no longer linear. This is because Wendy is making her classification based on games and not on a move-by-move basis. At $P(embed) = 0.5$, only half of the moves are used to send data but they occur in almost every game, which is equivalent to every game being used to send data at a lower rate by half. This means that when Wendy flags a game as being suspected of transmitting data, she does not know which or how many moves were used to transmit data. Thus, Wendy's correct classifications are mostly useless since Wendy cannot hope to extract the hidden message and cannot drop the messages or change them without alerting Alice and Bob to her tampering.

The security tests were revised to allow Wendy to make her classification on a move-by-move basis and the results are shown in Figure 4. The results for both the Random Games technique and the Random Moves technique are very similar to each other but differ greatly from the game-by-game analyis in the three cases where Wendy has a better model of gameplay than do Alice and Bob. The other three cases, where Alice and Bob have a model which is at least as good as Wendy's, produce much the same results as they did in game-by-game analysis. Under move-by-move analysis, Wendy's error rate increases with $P(embed)$, except in the case of Optimal Wendy against Random Alice and Bob which is just about 50% regardless of the value of $P(embed)$. This is because most of the moves Alice and Bob make seem completely normal to Wendy, except when Alice and Bob are playing Random, in which case half of their moves appear suspicious and the other half are inadvertantly played perfectly.

The likelihood that Wendy considers a given move to be suspicious depends only on the two models of gameplay being pitted against each other. From the plots, we can see that Greedy considers about 30% of Random's moves to be suspicious and Optimal considers nearly 50% of Random's moves, and approximately 20% of Greedy's moves, to be suspicious. Greedy does not consider its own or Optimal's moves to be suspicious, and similarly neither does Optimal consider its own moves to be suspicious.

These results show that Wendy's advantage over Alice and Bob is dependent entirely on the relative quality of her model of the cover-object space compared to that of her opponents. When Alice and Bob have a model of the cover-object space which is as good as or better than Wendy's model, then they can sneak hidden information past her without any problems. When Wendy has the better model, Alice and Bob are easily defeated.

5. OTHER GAMES

The combinatorial games of Chess, Go, and Connect Four were mentioned above in Section 1 and have already been dealt with in previous research. The framework presented in this paper is applicable to these games and supports schemes which are substantially similar to those presented in the previous work on this topic. The framework can also be generalized to be applicable to multiplayer games and games without perfect information. This section will briefly explain the exploitation of covert channels in two other games which are popular games to play online.

The first game is Dots and Boxes, which, like Tic-Tac-Toe, is a turn-based pencil-and-paper game played on a grid. However, it can be played by more than 2 players and the grid for Dots and Boxes is very much larger than the grid for Tic-Tac-Toe. A typical Dots and Boxes grid may be larger than 9x9, where Tic-Tac-Toe is only 3x3. Also, instead of making moves in the cells, moves are made at cell boundaries, the goal being to completely surround a grid cell in order to claim it for oneself. The winner of the game is the player with the most boxes of their own at the end of the

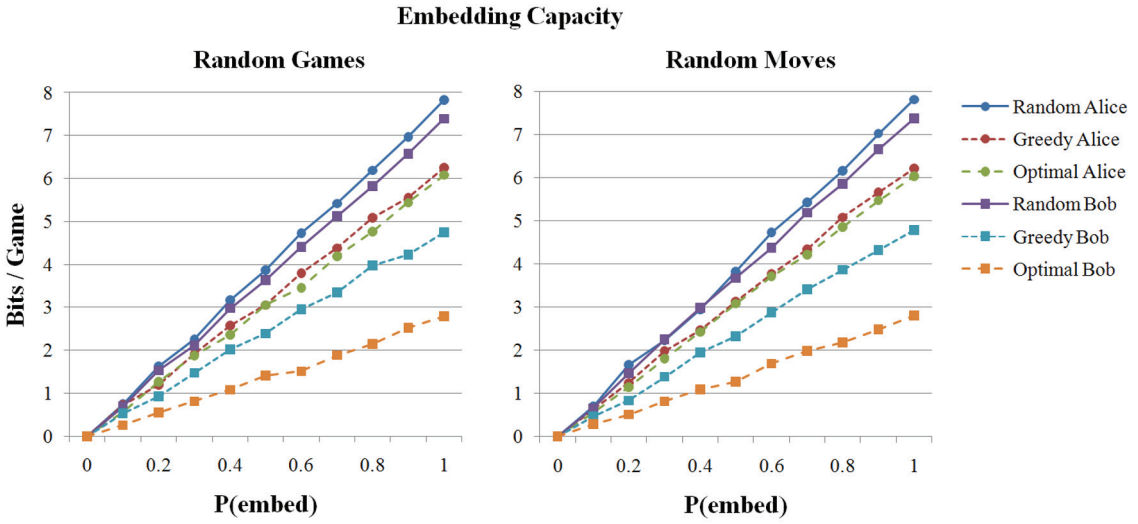

Figure 2: Embedding Capacity experimental results

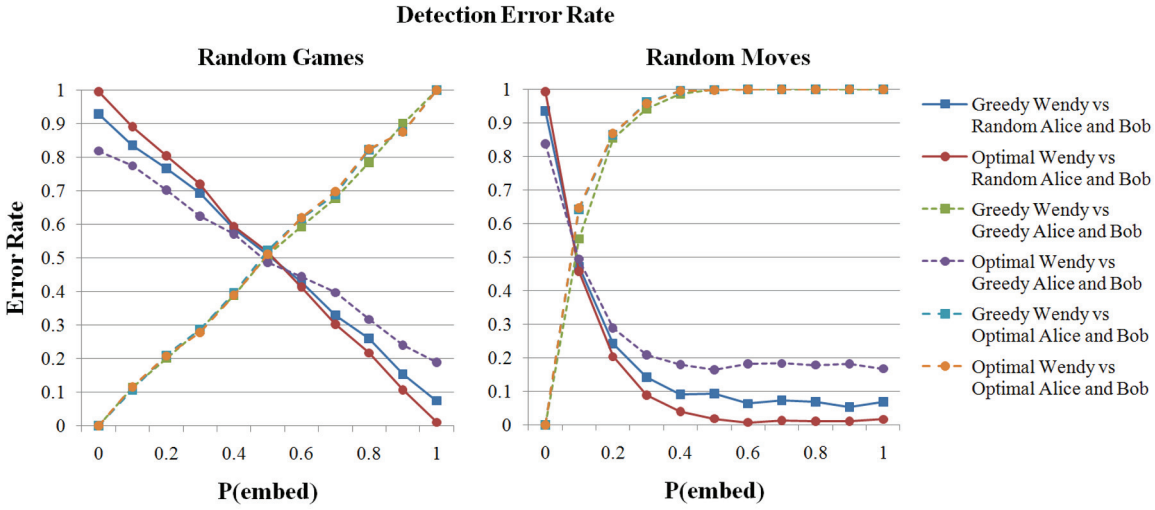

Figure 3: Detection Error Rate experimental results

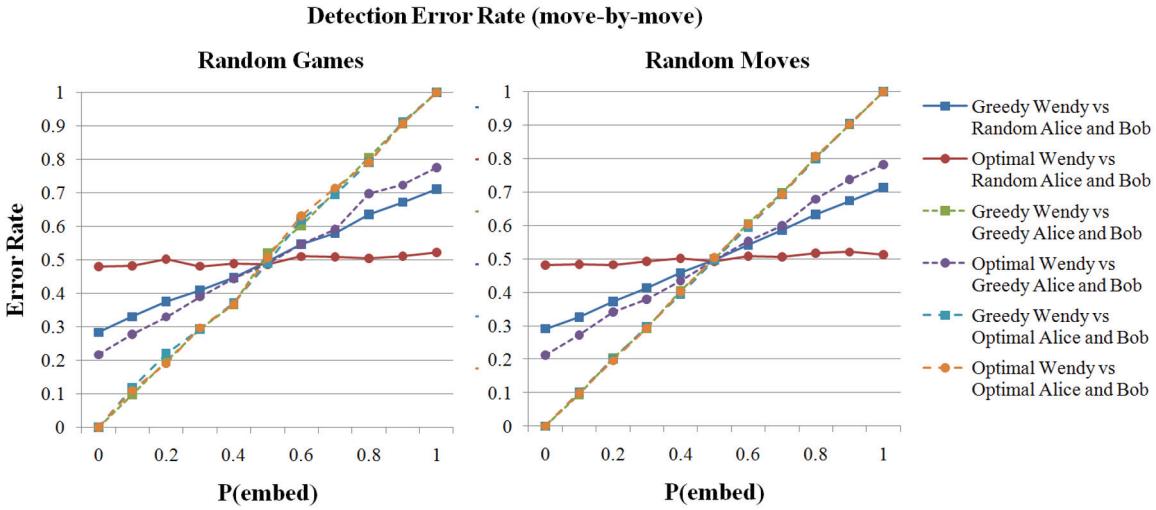

Figure 4: Detection Error Rate (move-by-move) experimental results

game. The large board size results in a much larger state-space for this game, which makes compuation of globally optimal choices expensive in terms of time and memory. Such a large initial state-space also means that seemingly random play early in the game is not out of the ordinary and a good-enough endgame strategy is often enough to secure a win against one's oppenent or opponents. The large board coupled with the nearly complete freedom of the first several turns means that the capacity of the covert channel is much larger than for Tic-Tac-Toe. For instance, an $n \times n$ grid for a game of Dots and Boxes contains $(n+1)^2$ dots and $2n(n-1)$ borders linking the dots. Thus, under the null strategy, Alice can send approximately $\frac{1}{N} log_2(2n(n-1)!)$ bits of information per game, as can each of the other $N-1$ players. Using a strategy which aims to win, or, equivalently, to not lose, will reduce this capacity but will allow the bits to be sent undetectably.

The second game to mention is Battleship. While still a 2-player turn-based pencil-and-paper game, Battleship is not actually a combinatorial games since the players do not enjoy full information while the game is being played. Before the first move is made, players hide a set of ships of varying length in a 10x10 grid with the hopes of having their well-placed fleet survive bombarment by the opposing player. The players take turns bombing a grid location belonging to their opponent and are told only if they hit or missed. The player who sinks all of their opponent's ships first wins. There are actually two independent covert channels in the game, one using ship placement and one using bombing locations. Using ship locations, the hidden message is encoded as a permutation of ship orientation and location. Once a player has won, she will know the location and orientation of all her opponent's ships, from which she can extract the hidden message. Using bombing targets to implement the covert channel is very much similar to the process used in Tic-Tac-Toe and Dots and Boxes: each choice encodes a bit sequence and the choices are driven by the bits of the secret information being transmitted. Since a player's ship placement and choice of targets are independent, the total capacity of the covert channel in Battleship is the sum of the capacties of using each individually. The capacity of the ship-placement channel is approximately 34 bits and that of the bombing target channel is 21 bits, thus the capacity of both used together is 55 bits per game.

6. CONCLUSION

A general framework for exploiting covert channels in combinatorial games was presented. The application of this framework to the game of Tic-Tac-Toe was demonstrated and it was shown that, even in the case of such a simple game, Wendy cannot accurately distinguish stego-games from clean games and nor can she, through application of the strict requirement of forcing perfect play, reduce the capacity of the channel to zero and thereby prevent covert communication. Experimental and analytical results on the capacity of the channel showed that playing optimally reduces, but does not eliminate the capacity of the channel. Experimental results on the steganographic security of the channel showed that Wendy's ability to accurately detect usage of the covert channel comes down to whether or not her model of play is better than Alice and Bob's. We also described how to exploit covert channels in multiplayer games and games without perfect information, such as Dots and Boxes and Battelship, which are more complex and have much larger covert channel capacity than Tic-Tac-Toe.

Multiplayer games are abundant on the internet and websites exist which provide the means for users to play these anonymously games with others. The difficulty that Wendy has in detecting and preventing covert communcation between Alice and Bob in isolation is compounded many times when Alice and Bob have access to anonymizing services which they alone can circumvent using covert authentication and communication through online games.

7. ACKNOWLEDGEMENTS

This research has been supported by NSF CNS-0716398 and NSF CCF-0939370. The authors would also like to thank the reviewers for providing comments to help strengthen the presentation and findings of the research.

8. REFERENCES

[1] K. Crowley and R. S. Siegler. Flexible strategy use in young children's tic-tac-toe. *Cognitive Science*, 17(4):531 – 561, 1993.

[2] A. Desoky and M. Younis. Chestega: chess steganography methodology. *Security and Communication Networks*, 2(6):555–566, 2009.

[3] A. Fraenkel. Combinatorial games: Selected bibliography with a succinct gourmet introduction. In R. J. Nowakowski, editor, *Games of No Chance*, pages 493 – 537. MSRI Publications, Cambridge University Press, 1996.

[4] G. T. Heineman, G. Pollice, , and S. Selkow. *Algorithms in a Nutshell*, chapter 7, pages 213 – 217. Oreilly Media, 2008.

[5] J. C. Hernandez-Castro, I. Blasco-Lopez, J. M. Estevez-Tapiador, and A. Ribagorda-Garnacho. Steganography in games: A general methodology and its application to the game of go. *Computers and Security*, 25(1):64 – 71, 2006.

[6] N. F. Johnson and S. C. Katzenbeisser. A survey of steganographic techniques. In S. Katzenbeisser and F. A. Petitcolas, editors, *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, Inc., 2000.

[7] D. Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner, 1996.

[8] B. Lange. Steganography using the chess PGN standard format. Technical report, SANS Institute, 2004.

[9] S. J. Murdoch and P. Zielinski. Covert channels for collusion in online computer games. In *Information Hiding 2004*, pages 355 – 369, 2004.

[10] R. L. Rivest. Chaffing and winnowing: Confidentiality without encryption, 1998. http://people.csail.mit.edu/rivest/Chaffing.txt.

[11] P. Sallee. Model-based steganography. In *Digital Watermarking*, volume 2939, pages 254–260. Springer Berlin / Heidelberg, 2004.

[12] G. J. Simmons. The prisoner's problem and the subliminal channel. In *Advances in Cryptology: Proceedings of CRYPTO '83*. Plenum Press, 1984.