Hindawi
Security and Communication Networks
Volume 2018, Article ID 8475818, 15 pages
https://doi.org/10.1155/2018/8475818

Research Article

Path Hopping: An MTD Strategy for Long-Term Quantum-Safe Communication

Reihaneh Safavi-Naini,[1] **Alireza Poostindouz**,[1] **and Viliam Lisy**[2]

[1] *University of Calgary, Calgary, AB, Canada*
[2] *Czech Technical University, Prague, Czech Republic*

Correspondence should be addressed to Reihaneh Safavi-Naini; rei@ucalgary.ca

Received 9 December 2017; Accepted 13 March 2018; Published 7 May 2018

Academic Editor: Hamed Okhravi

Moving target defense (MTD) strategies have been widely studied for securing computer systems. We consider using MTD strategies to provide long-term cryptographic security for message transmission against an eavesdropping adversary who has access to a quantum computer. In such a setting, today's widely used cryptographic systems including Diffie-Hellman key agreement protocol and RSA cryptosystem will be insecure and alternative solutions are needed. We will use a physical assumption, existence of multiple communication paths between the sender and the receiver, as the basis of security, and propose a cryptographic system that uses this assumption and an MTD strategy to guarantee efficient long-term information theoretic security even when only a single path is not eavesdropped. Following the approach of Maleki et al., we model the system using a Markov chain, derive its transition probabilities, propose two security measures, and prove results that show how to calculate these measures using transition probabilities. We define two types of attackers that we call risk-taking and risk-averse and compute our proposed measures for the two types of adversaries for a concrete MTD strategy. We will use numerical analysis to study tradeoffs between system parameters, discuss our results, and propose directions for future research.

1. Introduction

Cryptographic infrastructure of the Internet allows users from across the world to establish private and authenticated, confidential communication channels, and interact securely. Shor's discovery of a quantum algorithm that can efficiently solve integer factorization and discrete logarithm problems [1], the two mathematical problems that are the basis of the security of the most prominent public key crypto algorithms such as RSA public key encryption and Diffie-Hellman key agreement, effectively brings down the cryptographic infrastructure of the Internet. The NSA's recent call for quantum-safe cryptography and the prediction that significant progress in the development of quantum computers could be expected within fifteen years [2, 3] have created a flurry of activities in research community, standardization bodies [4, 5], and major industries [6–8]. The main approaches to quantum-safe cryptography use, (i) quantum cryptographic models and algorithms, (ii) cryptographic algorithms that rely on computational assumptions for which no efficient quantum algorithm is known [9], and (iii) cryptographic systems that do not use any computational assumptions. This last approach results in information theoretically secure systems and is followed in this paper.

A prominent and widely researched direction in information theoretically secure communication is *physical layer security* systems that base security on assumptions about the physical environment [10]. In these systems the advantage of the sender and the receiver over the adversary is captured through the properties of the physical layer of communication. For example, in Wyner's model [11] Alice is connected to Bob and Eve through two noisy channels, and the assumption is that Eve's reception is noisier than Bob's reception. The extra noise in Eve's channel is the resource and can be used for securing communication against Eve, without the need for a shared key. A unique property of this approach compared to computationally secure systems is providing *long-term security* which refers to the property that Eve's transcript of

communication cannot be used for offline attacks. This is because security is due to the lack of information and not adversary's limited computation.

In this paper we assume there are multiple communication *paths* (N paths) between the sender and the receiver. A path is an abstraction of a channel and can have different realizations in practice. For example, in wireless communication, a path can correspond to a frequency that is used for transmission and reception by the sender and the receiver; and multiple paths are specified by a set of frequencies that will be used by the two. Or in a sensor network, a path consists of a sequence of nodes in the network which are used to send messages from the sender to the receiver. A similar notion of path can be defined for communication over the Internet. If the adversary can eavesdrop all the paths, secure communication without additional assumptions (e.g., quantum mechanical, computational or other physical layer assumptions) is impossible. We assume that although the set of all paths (e.g., possible frequencies) is known to the attacker, they cannot eavesdrop all the paths at the same time.

To provide cryptographic security against an attacker whose goal is to learn the sent message, the sender can use an (N, N)-secret sharing (see Section 2) to construct N shares of the message and send each share along a path. This is an inefficient solution with communication rate of $1/N$ (i.e., for one bit of information, the sender must send N bits). If the number of paths that the adversary can simultaneously eavesdrop is bounded by $K' < N$, the sender and receiver can select a random set of K paths and use a (K, K)-secret sharing. Note that K is not necessarily equal to K', but to keep the introductory discussion simple, let $K = K'$; we will later discuss the relation between K and K' in Section 3. If the set of K paths is kept fixed, the attacker will discover them over time and will be able to learn the message. We propose "path hopping," where the sender and the receiver regularly change ("hop") one or more paths that have been chosen for communication. We also allow the attacker to change their selected paths. We model and analyze dynamic behaviour of the system and show that it results in efficient cryptographic security using an MTD strategy.

1.1. Our Work. Alice wants to send a stream of data to Bob. The adversary is computationally unlimited (no computational assumption is made). Alice and Bob are connected by a set of N communication paths, up to K of which can be eavesdropped by the adversary at the same time. The adversary probes a path to determine if it carries data, and if it does, it captures it (e.g., scanning the port on a server and if open, later break into the path). The adversary is *mobile* [12] and can move in the network in the sense that, in each time step, it can release a captured path and capture a new one. Hence, it can eavesdrop different sets of paths during different time periods. *We assume the attacker can eavesdrop up to K paths, but only eavesdrops those that carry data, and not the ones that are not in use between Alice and Bob* (one can consider a case where the attacker always eavesdrop on K paths. Although the analysis approach will be similar, the actual calculations will be different). Alice uses K (out of N) paths at each time and the attacker needs to know all the K paths (targets) to be able to determine the message.

Time is divided into fixed consecutive intervals, each referred to as a *time step*. In each time step, *defender* (which includes Alice and Bob), *attacker,* or none of them takes an action (move). Using the MTD framework of [13], the combination of the attacker's and the defender's actions in each time step can be modelled as a *Markov chain*. We define $G_{u,v}$ Markov chain where (exactly) u and v paths are hopped by the defender and the attacker, respectively, and derive transition probabilities of the chain. (One can also consider $G_{\leq u,\leq v}$ Markov chain where in each time step, the defender randomizes *up tou* paths and the attacker hops and probes *up tov* paths. We leave this for future work.) For our concrete analysis we focus on $G_{1,1}$ where the defender's and the attacker's actions involve a single path, only.

In each time step, the system can be in one of the $K + 1$ *states* labeled by $0, 1, \ldots, K$, where state 0 is the starting state of the system, state K is the *winning state* for the attacker, and in state i, attacker has captured i paths. We assume the defender leads in each time step and *moves* with a fixed probability λ. The attacker also moves in the same interval with a fixed probability μ that is upper bounded by $1 - \lambda$ (the model assumes that, in each time step, the attacker and the defender do not act simultaneously.) Note that μ can be chosen by the adversary, knowing the upper bound. We define a *risk-taking* and a *risk-averse* attacker, depending on their choice of μ. A risk-taking adversary chooses the highest available attack probability, that is, $\mu = 1 - \lambda$. A *risk-averse* attacker would like to stay undetected and so limits their action rate to a threshold that is determined by the intrusion detection system (IDS) of the defender. Thus, in the case of a risk-averse adversary, in each time step, there is a probability that no one moves.

We model the system as a Markov chain, and in Section 3, derive the transition probabilities of the Markov chains associated with the risk-taking (case C1) and the risk-averse (case C2) attackers.

Security Measures. We use two security measures to evaluate effectiveness of a path hopping strategy: (i) the expected number of times that the attacker reaches the winning state in T time steps, assuming that it is starting from state 0, and (ii) the expected number of time steps to enter the wining state for the first time, assuming that the attacker starts from state 0. The two measures are denoted by L_T and $E^{(1)}_{\text{win}}$, respectively.

These security measures capture security requirements of different scenarios. L_T is appropriate for data streams for which sporadic access to different parts of the stream may be tolerated. For example, small excerpts of a large file are not expected to leak much information about the file. Theorem 3 shows that L_T is upper bounded by the product $T \cdot \pi(K)$, where $\pi(K)$ is the Kth component of the stationary probability distribution of the Markov chain. This suggests that $\pi(K)$ *can be used to represent* L_T*, with higher values corresponding to less security.*

$E^{(1)}_{\text{win}}$ is appropriate for highly sensitive data streams that must stay strictly inaccessible to the adversary and the sender

wants to ensure that the expected number of time steps to the first compromise is sufficiently high (possibly higher than the length of the stream). Theorem 4 shows that $E^{(1)}_{\text{win}}$ can be calculated by solving a set of linear equations whose coefficients are derived from the transition probabilities of the Markov chain. We use $E^{(1)}_{\text{win}}$ *as a security measure with higher values corresponding to higher security*.

Numerical Results. Deriving closed form expressions for L_T and $E^{(1)}_{\text{win}}$ is a challenging task. For $G_{1,1}$, we use numerical calculations to study variations of security measures for different values of system parameters. Our results are given in Section 6. They show the following:

(1) For fixed N and K, security increases (i.e., $\pi(K)$ decrease, and $E^{(1)}_{\text{win}}$ increases) as λ, the defender's probability of action, increases (Figures 3 and 4 for C1 and Figures 6 and 7 for C2).

(2) For fixed N, security can be maintained by increasing λ, even when $K = N - 1$ (Figure 5) and in all cases communication rate is $1/K$.

Figure 5 also shows that, for given values of N and λ, as K increases, security initially increases, then it reaches a plateau and then starts to decrease. This is because when K is small (relative to N), the target paths are hidden among many available paths and the success chance of correctly guessing a path would be small. However, when K is large (relative to N), the attacker's probability of correct guessing increases. Interestingly, this point of saturation increase as λ that represents variability of the system increases. This graph can be used to select the optimal value of K to provide maximum security, while achieving the highest communication rate.

(3) Using numerical analysis, one can estimate the *cost of being risk-averse* in terms of decrease in L_T or increase in $E^{(1)}_{\text{win}}$. In Section 7 we show that an adversary who chooses not to use all their attacking power (although they can act with probability $1 - \lambda$, they choose $\mu < 1 - \lambda$) will effectively reduce the expected number of times that they will occupy the winning state (proportional to $\pi(K)$) and will have higher $E^{(1)}_{\text{win}}$.

Attack Costs. Our model focusses on the defender's ability to provide security by making the *physical environment* dynamic and does not consider the associated costs. Attacker and defender's actions have payoffs. The attacker needs to spend resources to launch attacks and also bear consequences of being detected. The defender must spend resources to implement the randomization strategy. This introduces side effects such as packet loss and communication delays that are a function of the rate of randomization (captured with parameter λ). The attacker's reward of their action is related to getting closer to the winning state, and the defender's reward is preventing the attacker reaching the final state. In Section 7 we discuss these payoffs. We also use our numerical calculation results to quantify the cost of being risk-averse.

Randomness Requirements of the System. Our proposed system assumes that the sender and the receiver share the set of target paths that is used for communication in each time step.

In practice if one can assume that the receiver will receive on all paths all the times, then no shared randomness is required: the sender will hop the paths and the receiver will receive the content on the target paths used in each time step. If receiver has the same restriction as the sender on the number of target paths, that is, the receiver can only receive on K paths (e.g., cost or restriction on the receiving equipment), then the sender and the receiver need shared randomness to simultaneously hop the paths. This can be realized in two ways: (i) using a preshared random string or (ii) employing a secure pseudorandom generator to extend an initial shared random seed.

The adversary view of the system in state i, in addition to the eavesdropped shares that are sent over that i target paths, includes the labels of the i target paths. In case (i), the sequence of random numbers associated with the labels of target paths will not reveal any information about future values of the sequence of target paths and so future path labels will remain unpredictable. In the case of (ii) however, each observation (of a target path) will leak information about the seed of the PR generator and one needs to use a PR generator with appropriate security level (e.g., a quantum-safe PR generators using a secure block cipher). Note that the MTD system will retain its security because the recorded transcript of communication although may reveal the seed of the PR generator in an offline attack will not have enough information about communicated message and so the message transmission will have long-term security.

1.2. Related Work. Breaking information into shares, to provide confidentiality and reliability, has been used in many cryptographic systems such as secret sharing [14] and information dispersal [15], in information theoretic setting, as well as computational setting [16]. These algorithms have been used in distributed storage systems [17] and are the building blocks of Secure Message Transmission [18] and network coding [19] which use multiple paths between the sender and receiver for providing *security and reliability*.

Uncoordinated Frequency Hopping (UFH) [20] has similarity to our work. In UFH the sender and receiver send and receive on two independently chosen subsets of frequencies, and the eavesdropper uses a third subset of frequency for eavesdropping. Authors show that, assuming public key infrastructure, one can communicate securely and reliably in this setting. The work of [21] uses a similar abstract model to construct information theoretic protocols for secure communication, without requiring public key infrastructure. The communication rate in this latter construction, however, is very low. Our approach is *coordinated* path hopping where the sender and receiver share an initial secret key that can be established using the scheme of [21]. We leave the analysis of the secret key requirement of our system and in particular efficient ways of generating new keys at the required hopping rate for future work.

Using diversity and introducing dynamic properties has been widely used in security systems. System properties

that can be diversified and randomized include program instructions [22, 23], operating system distributions [24], and systems [25]. A comprehensive study of various methods is given in [26]. Using game theory for analyzing attackers' strategies in dynamic systems has been studied in [27, 28].

Organization. Section 2 recalls the MTD Markov chain framework. Section 3 presents our path hopping model. Security analysis and measures for our model are introduced in Section 4. Sections 5 and 6 present our simulation results for the $G_{1,1}$ game. Sections 7 and 8 cover utility discussions and our concluding remarks.

2. Preliminaries

We recall the basic MTD Markov chain framework that is used in our work and review construction and properties of (t, t) secret sharing schemes.

MTD Markov Framework [13]. The system is a defined by the interaction between a *defender* and an *attacker*. The defender and the attacker each have a set of possible actions, denoted by $\mathscr{D} = \{\epsilon, d_1, d_2, \ldots\}$ and $\mathscr{A} = \{\epsilon, a_1, a_2, \ldots\}$, respectively; in both ϵ shows no action. Time is divided into time steps. In each time step the system is in one of the N possible states. In each time step, the defender and the attacker get a turn to move and the state change probability is determined by their chosen actions and their results. A *strategy* of a player determines all actions taken by the player in all points of the game. Using *Markov model* allows the player's strategy to only depend on the state that the system is in and independent of the history of how the system has reached a state.

Definition 1. An $\mathscr{M}$-MTD game is defined by a $(K+1)\times(K+1)$ transition matrix M which describes a Markov chain of state transitions that reflects both defender and attacker moves. Initially the game starts in the state 0. At each next time step the game transitions from its current state i to a new state j with probability $M_{i,j}$.

The state K is the winning state from the adversary's view (defender losing the game). Initially the system is in state 0 (from both attacker and defender's view point). In each time step the defender takes an action according to matrix M^D with probability λ, the attacker takes an action according to M^A with probability μ, and with probability $1 - \lambda - \mu$, both remain without any action.

Definition 2. An (M^D, λ, M^A, μ)-MTD Game is defined by

(1) parameters λ and μ that satisfy $0 \leq \lambda + \mu < 1$; the parameters represent the rate of defender's and the attacker's play, respectively;

(2) $(K + 1) \times (K + 1)$ transition matrices M^D and M^A; for $i, j \in \{0, 1, \ldots, K\}$, $M^D_{i,j}$ (or $M^A_{i,j}$) represents the probability of transitioning from state i to state j when the defender (or the attacker) plays a move in state i.

Thus, in each time step a three-sided coin is tossed, and for each side, the corresponding action is realized, and we have the transition matrix

$$M = \lambda M^D + \mu M^A + (1 - \lambda - \mu) I_{K+1}, \tag{1}$$

where I_{K+1} is the $(K + 1) \times (K + 1)$ identity matrix.

A Markov chain M is *irreducible* if each state can be reached from any other state. A Markov chain is *aperiodic*, if all the states have period 1 where the period of state i is defined as $\gcd\{n > 0 : \Pr(X_n = i \mid X_0 = i)\}$, where X_n is the random variable describing the state of the game after n steps. The two properties together guarantee the existence of a limiting *stationary distribution*π, where $\pi = \pi M$.

Secret Sharing. A (t, t)-secret sharing is a cryptographic primitive [14] that divides a secret m into t shares, each given to a party, satisfying two properties: (i) *reconstructability* which means the share of all parties can perfectly reconstruct the original secret and (ii) *perfect secrecy* which means that if a single share is missing, the secret remains perfectly uncertain. A secret sharing scheme provides two algorithms for *share generation* and *secret reconstruction*. Let $\mathscr{M} = Z_L$, where Z_L is the set of integers modulo L, denote the set of secrets, and assume that all secrets are equally likely ($\Pr(Z = z_i) = 1/L$). The share generation algorithm takes a message $m \in \mathscr{M}$ as input and generates t shares $s_1, s_2, \ldots, s_t$ as follows. For s_i, $i = 1, \ldots, s_{t-1}$, randomly chooses an element r_i in Z_L. Then the shares of the secret m are $(r_1, r_2, \ldots, r_{t-1}, r_t = m - \sum_{i=1}^{t-1} r_i \pmod L)$. It is easy to see that t shares recover the secret (finding the sum modulo L), and even if $t-1$ shares are known, the secret remains completely uncertain.

3. The MTD Game of Path Hopping

We consider the setting described in Section 1.1: there is a message source that generates a stream of data that must be protected against an eavesdropper. There are N communication paths that connect the sender to the receiver. To protect message transmission against an eavesdropper who can simultaneously eavesdrop up to K' paths ($K' < N$), the sender does the following: (i) randomly chooses a subset of $K < N$ available paths; (ii) uses a (K, K) secret sharing to construct K shares for the message, and (iii) sends each share on one of the selected path. The chosen paths are also called *target paths*. The receiver knows the paths that are used by the sender in each time step. If the adversary eavesdrops only a subset of the target paths (and not all target paths), because of the perfect secrecy of the (K, K)-secret sharing scheme, the attacker will stay completely uncertain about the message.

We assume that the attacker will not keep a path that is not carrying data. That is, because of the limitation on the number of paths that they can simultaneously eavesdrop, they prefer to release a path that is not used in the current time step and wait for the next time step to try again, noting that, due to their probabilistic strategy, there would be a chance to try the released path in the next time step again. To simplify our analysis, we first consider the case that $K = K'$. For the cases that $K > K'$ or $K < K'$, similar analysis can be used; we omit details because of space.

To protect against this adversary, in each time step, the sender and receiver will *hop* one or more of the target paths, noting that lacking access to even one of the target paths will leave the adversary completely uncertain. We will use the MTD game framework of Definition 2 and model the problem as a dynamic system (game) influenced by (between) two *players,* a *system defender (or simply defender)* that includes the *sender* and the *receiver* and an *attacker*. The attacker wins the MTD game (in each time step) if they find the K target paths.

3.1. $G_{u,v}$ Games. In each time step, the defender can randomize a subset of u target paths. Similarly, the adversary can simultaneously probe v paths.

We first describe the Markov chain associated with the game, then derive transition probabilities of $G_{u,v}$, and finally present a detailed analysis of $G_{1,1}$.

3.2. Markov Chain. The set of the defender's and the attacker's actions is $\mathscr{D} = \{\epsilon, d_1\}$ and $\mathscr{A} = \{\epsilon, a_1\}$, respectively, where d_1 and a_1 are defender and attacker actions and ϵ is no action. Let $\mathcal{S} \subset [N]$ denote the set of current target paths and $\mathcal{S}_{|A}$ denote the subset of target paths known to the adversary.

Defender's Move. The defender cannot determine with certainty if a path is being eavesdropped. We thus consider a defender who, in all time steps, plays a *memoryless strategy*. That is the defender plays (issues the move a_1) with probability λ, irrespective of any learnt information about the attacker's state, or own history of actions. When the defender plays in state i, they will choose a subset S_D^i of the current target paths $\mathcal{S}$ and replace the paths in S_D^i with a randomly selected subset of $(N - K)$ nontarget paths.

The chosen paths in S_D^i may belong to S_A^i (attacker's known path in state i) or be outside it.

Attacker's Move. The attacker is adaptive. In state i, the set $\mathcal{S}_{|A}$ of target paths that is known to the adversary is of size i. The adversary randomly selects a subset S_A^i of size v of $([N] \setminus S_A^i)$ possible target paths and keeps the message carrying paths and releases the rest. For the adversary, all paths that are not in their set of i known target paths have the same probability of being a target path.

We assume that, in state i, as soon as the defender reallocates a target path that is in S_A^i, the attacker can detect the change (the path is not one of the K target paths). However this will not affect the adversary's action at this state simply because they know that those paths are not possible target paths.

No Move. Defender and attacker are probabilistic and no moves can be issued by either of them.

In a time step, if the attacker does not issue an action, they will bear the risk of potentially losing one of their known target paths during the next time step. This is because the defender will play a memoryless strategy and will move with probability λ. This extra risk would translate into a higher probability of not being able to reach the winning position of the game.

To reduce the probability of losing a target path while waiting, the attacker should act when possible and use the available $1 - \lambda$ action rate. We refer to this attacker as a *risk-taking* attacker as they focus on maximizing their winning chance. More frequent attacks however have the risk of triggering alarm in the defender's intrusion detection system (IDS), tightening security, and reducing access to the system. Let τ be a threshold that is used by the defender's IDS to raise the threat level of the system. To avoid reduction in accessing the system, the attacker may prefer to keep their attack rate below τ. We refer to this attacker as a *risk-averse* attacker.

The defender plays memoryless with probability λ, and so in each time step the attacker moves with probability

$$\mu = \min\{\tau, 1 - \lambda\}. \tag{2}$$

There will be no move by any of the players in a time step, with probability $1 - \lambda - \mu$. Thus the system transition matrix will be

$$M = \lambda M^D + \mu M^A + (1 - \lambda - \mu) I_{k+1}. \tag{3}$$

Equation (2) shows that, depending on the value of τ (the attack detection threshold of the defender), we have two cases:

C1: $\tau > 1 - \lambda$. In this case from (2), we have $\mu = 1 - \lambda$ and

$$M = \lambda M^D + (1 - \lambda) M^A. \tag{4}$$

C2: $\tau < 1 - \lambda$. In this case from (2), we have $\mu = \tau$ and

$$M = \lambda M^D + \tau M^A + (1 - \lambda - \tau) I_{k+1}. \tag{5}$$

We refer to C1 and C2 as risk-taking and risk-averse attacker, respectively.

3.3. Transition Probabilities of $G_{u,v}$. In state i, the attacker knows i target paths in $\mathcal{S}_{|A}$. A state transition that starts from state i is in general because of the combination of the defender's and the attacker's actions in the following time step. A defender's action reallocates target paths and (since the attacker only holds target paths) can result in state i to change to j where $j \leq i$. An attacker's action, however, could result in more target paths being captured and so change the state to $j \geq i$. The state will not change that is stays at i, because of the defender *or* the attacker's action *or* no moves at all. In the following, we obtain transition probabilities (starting from state i) for (i) the defender's move and (ii) the attacker's move and combine them to obtain the transition probabilities of the chain. For the case of "no move" (which happens with probability $1 - \lambda - \mu$) the state of the game will not change.

Defender's Move in State i. Defender chooses a set S_D^i of u paths from the set $\mathcal{S}$ of current paths and replaces them with a set $S_D'^i$ of u paths chosen from the $N - K$ candidate target paths $[N] \setminus \mathcal{S}$.

Let $S^i_{D,1} = S^i_D \cap \mathcal{S}_{|A}$ be the intersection of S^i_D and the adversary's set of captured paths, and let $|S^i_{D,1}| = x$. We note that $0 \le x \le \min\{i, u\}$. Thus the state of the game after the defender's action will be $i - x$ (because x target paths have been removed from $\mathcal{S}_{|A}$) and we have

$$p_D(i - x \mid i) = \frac{\binom{i}{x}\binom{K-i}{u-x}}{\binom{K}{u}}, \quad 0 \le x \le \min\{i, u\}. \tag{6}$$

Note that, for $x = 0$, the state of the game will not change.

Attacker's Move in State i. Attacker holds the target paths in $\mathcal{S}_{|A}$ and knows the state of the game. The attacker will choose a set S^i_A of v paths from $[N] \setminus \mathcal{S}_{|A}$, the set of $N - i$ available candidate target paths.

Let $S^i_{A,1} = S^i_A \cap (\mathcal{S} \setminus \mathcal{S}_{|A})$ be the intersection of S^i_A and defender's set of target paths $\mathcal{S}$ that are not captured yet, and let $|S^i_{A,1}| = y$. We have $0 \le y \le \min\{K - i, v\}$. With the new y captured target paths, the sate of the game will become $i + y$ and we have

$$p_A(i + y \mid i) = \frac{\binom{K-i}{y}\binom{N-K}{v-y}}{\binom{N-i}{v}}, \quad 0 \le y \le \min\{K - i, v\}. \tag{7}$$

For $y = 0$, the state of the game will not change.

Transition Probability from State i to j. Transition probabilities from state i to j will be calculated using (6) and (7) and $\max\{0, i-u\} \le j \le \max\{i+v, K\}$. We note that transitions with $j < i$ occur only due to the defender's move, and transitions with $j > i$ occur due to the attacker's move. No transition will be due to the defender, the attacker, or no move, with probabilities λ, μ and $1 - \lambda - \mu$, respectively. Thus we have the following transition probabilities:

$$p(j \mid i) = \begin{cases} \lambda p_D(j \mid i) & \text{if } i - \min\{i, u\} \le j < i; \\ \lambda p_D(i \mid i) + \mu p_A(i \mid i) + (1 - \lambda - \mu) & \text{if } j = i; \\ \mu p_A(j \mid i) & \text{if } i < j \le i + \min\{K - i, v\}; \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Here $p_D(j \mid i)$ and $p_A(j \mid i)$ are defined in (6) and (7), respectively.

The above probabilities show that in each time step a state can be changed to up to $u + v$ other states or stay the same.

3.4. System Parameters. The Markov chain that models the system is determined by the parameters N, K, λ, and μ. In the following we will define security measures for the system and prove Theorems that relate these measures to the system parameters.

4. Security Analysis

We use two security measures related to the success criteria of the attack.

4.1. Expected Number of Compromises. Consider the system over a period of T time steps, starting from the state 0. Within these T time steps, the expected number of times that the system will be in the compromised state, that is, the attacker is able to learn the message, is an important security measure. (Note that one can use coding strategies [15] to spread information over longer sequences, and so estimating the expected number of compromises provides the required parameter for encoding.)

Theorem 3. *For an MTD game of path hopping with transition matrix M and stationary distribution $\pi = (\pi(0), \pi(1), \ldots, \pi(K))$, where K is the winning state, L_T, which is the expected number of times the adversary wins in the first T time steps, is less than or equal to $T \cdot \pi(K)$, assuming that the game starts with the $\mathbf{0} = (1, 0, \ldots, 0)$ distribution.*

Proof. The game starts at $\mathbf{0} = (1, 0, \ldots, 0)$. Let L_T denote the expected number of times the attacker wins in the first T time steps.

We first assume the attacker's starting position is chosen according to the stationary distribution $\pi = (\pi(0) \cdots \pi(K))$. Our goal is to find the expected number of times the attacker wins in T steps, starting with this distribution.

Let X_j, $j = 1, \ldots, T$ be an indicator variable that takes the value 1 if the attacker wins in the time step j and zero otherwise. Note that, starting from stationary distribution π, the distribution of next step position of the attacker is $\pi M = \pi$, and so each X_j has identical distribution $\Pr(X_j = 1) = \pi(K)$.

The random variable $S = \sum_{j=1}^{T} X_j$ is the number of times that the attacker wins in T time steps. Noting the linearity of the expectation function $\mathbb{E}(\cdot)$, that is, $\mathbb{E}(X+Y) = \mathbb{E}(X)+\mathbb{E}(Y)$, we have

$$\mathbb{E}(S) = T \cdot \pi(K). \tag{9}$$

The adversary will have zero chance of winning in the first K/v time steps if they start with the 0 distribution, and so

$$L_T \le T \cdot \pi(K). \tag{10}$$

The last step of the argument assumes that, starting from the initial distribution $(1, 0, \ldots, 0)$, $\Pr(X_j = 1)$ is monotonically

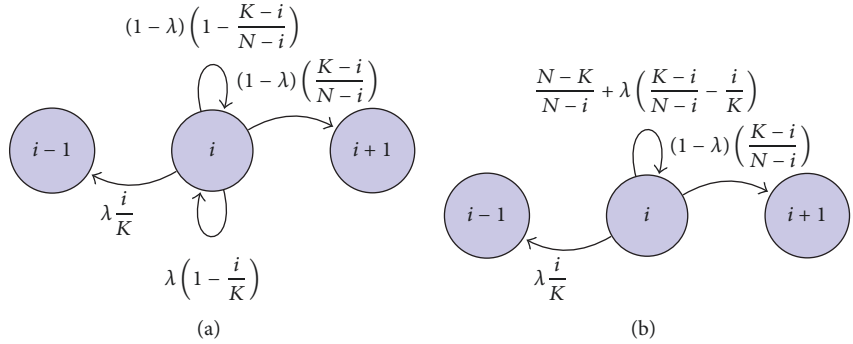

FIGURE 1: State transition probabilities for case C1. (a) shows the defender's moves, λM^D, with arrows bellow and attacker's moves, $(1-\lambda)M^A$, with arrows above. (b) shows the combined transition matrix $M = \lambda M^D + (1-\lambda)M^A$.

increasing (in fact a weaker assumption would suffice for this last step of the argument, which is $\Pr(X_j = 1) \leq \pi(K)$ for all j) in each step of the chain until it reaches $\pi(K)$. □

In our numerical computation we use $\pi(K)$ to represent this security measure.

4.2. Expected Number of Steps to the First Time Win. Our second security metric is the *expected number of steps to first time compromise*. This is an important measure for defender to estimate unbreakability of the system and for the attacker to estimate the work (in terms of the number of time steps that could be translated into attacker's cost) needed to break the system. This measure can be calculated by solving a set of linear equations.

Theorem 4. *Consider an MTD game with transition matrix M. Let v_j denote the expected number of times to reach the state K (the winning state) for the first time, if the game M has started with state j. We have*

$$\mathbf{v} = \mathbf{r} + \widetilde{M}\mathbf{v}, \tag{11}$$

where $\mathbf{r}(j) = 1$ for all $j \in \{0, \ldots, K-1\}$ and $\widetilde{M}$ is the same matrix as M with the last (Kth) row removed.

Proof. We consider an attacker that starts in state 0. Let $E^{(1)}_{\text{win}}$ denote the expected number of time steps to compromise the system. Let v_j denote the expected number of time steps to reach state K (the winning state) for the first time, if the game M starts at state j. In both MTD games, starting from state 0, the next state will be state 0 or 1, and so we can write

$$v_0 = 1 + M_{00}v_0 + M_{01}v_1. \tag{12}$$

That is, the expected number of times to reach state K from state 0 is one (time step) more than the weighted average of the expected number of times to reach state K from state 0 if the next move was to 0, and the expected number of times to reach state K from state 1, if the next move was to state 1. We can write similar equations for all states except state K, for which $v_K = 0$. Let $\mathbf{v}$ and $\mathbf{r}$ be column vectors such that $\mathbf{v}(j) = v_j$ and $\mathbf{r}(j) = 1$ for all $j \in \{0, \ldots, K-1\}$. The set of equations can be written as

$$\mathbf{v} = \mathbf{r} + \widetilde{M}\mathbf{v}, \tag{13}$$

where $\widetilde{M}$ is M with the Kth row removed. This linear equation can be solved for $\mathbf{v}$ for any given M and the first element of $\mathbf{v}$ is our desired $E^{(1)}_{\text{win}}$ security metric. In Section 6 we will present graphs of $E^{(1)}_{\text{win}}$ for various game settings. □

5. The $G_{1,1}$ Game

To better understand the relationship between parameter values (N, K, λ, μ), in the following we will focus on $G_{1,1}$ where $|S^i_A| = |S^i_D| = 1$.

That is, in state i, the defender moves by randomly choosing one of the K paths in $\mathcal{S}$ and swapping it with a randomly chosen path from the $N-K$ paths in $[N] \setminus \mathcal{S}$. If the random choice from $\mathcal{S}$ is one of the i paths in S^i_A that is known to the adversary, the adversary loses one of their i captured paths, and the state will move to state $i-1$, and this happens with probability i/K. Otherwise, if the defender's selected path is not in S^i_A the system stays in the same state i and this has probability $1 - i/K$.

The attacker will randomly choose one of the $[N] \setminus \mathcal{S}_{|A}$ possible paths. The new path will be a new target path with probability $(K-i)/(N-i)$ and so the system will move to the state $i+1$ with this probability. On the other hand with probability $1-(K-i)/(N-i)$, the selected path will not be a target path and with this probability the system will remain in state i.

5.1. Case C1: Risk-Taking Adversary. State transition probabilities are given by (4). Figure 1(a) shows state transition probabilities due to the attacker's and the defender's actions. The transition probabilities on the upper part of the figure are due to the attacker's action. Figure 1(b) shows the combined transition probabilities.

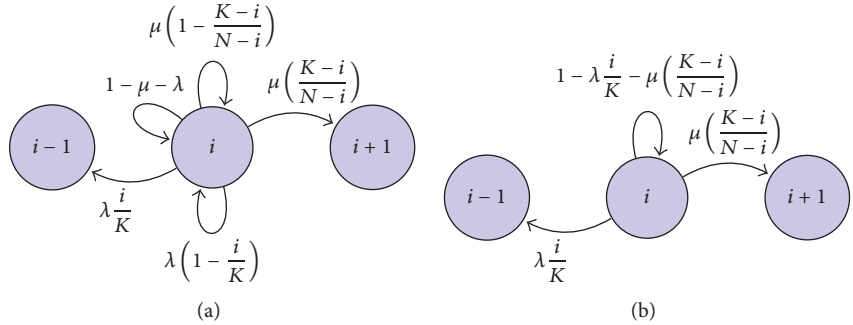

FIGURE 2: State transition probabilities for case C2. (a) shows the defender's moves, λM^D, with arrows bellow and attacker's moves, μM^A, with arrows above and the no move $1-\mu-\lambda$. (b) shows the combined transition matrix $M = \lambda M^D + \mu M^A + (1-\lambda-\mu)I$.

This is a Markov chain and the state transition matrix $M = [M_{i,j}] = [p(j \mid i)]$ can be obtained from Figure 1(b) as

$$p(i-1 \mid i) = \lambda\frac{i}{K}, \tag{14}$$

$$p(i \mid i) = \frac{N-K}{N-i} + \lambda\left(\frac{K-i}{N-i} - \frac{i}{K}\right), \tag{15}$$

$$p(i+1 \mid i) = (1-\lambda)\left(\frac{K-i}{N-i}\right). \tag{16}$$

It is easy to see that the Markov chain is irreducible and aperiodic. Using this matrix we can find stationary probability distribution of the system denoted by $\pi = (\pi(0)\cdots\pi(K))$.

5.2. Risk-Averse Adversary: Case C2. The state transition matrix is given by (5). At each time step, (i) defender moves (randomize) with probability λ, (ii) attacker moves with probability τ, and (iii) no move happens with probability $1-\lambda-\tau$.

The adversary knows the state and moves with probability τ. The state transition matrix M can be obtained from Figure 2(b), given by

$$\begin{aligned} p(i-1 \mid i) &= \lambda\frac{i}{K}, \\ p(i \mid i) &= 1-\lambda\frac{i}{K}-\mu\left(\frac{N-K}{N-i}\right), \\ p(i+1 \mid i) &= \mu\left(\frac{K-i}{N-i}\right), \end{aligned} \tag{17}$$

where $M = [M_{i,j}] = [p(j \mid i)]$. Using this matrix we can find stationary probabilities $\pi = (\pi(0)\cdots\pi(K))$. Again, the Markov process is irreducible and aperiodic and a limiting stationary distribution always exists.

5.3. Bounds on $\pi(K)$. We prove an upper bound on $\pi(K)$ using the following lemma.

Lemma 5. *Let $M = [M_{i,j}] = [p(j \mid i)]$ be the transition matrix of a Markov chain such that, for any row i, all components except $M_{i,i-1}, M_{i,i}, M_{i,i+1}$ are zero.*

Then for any i one has

$$\pi(i)\, M_{i,i+1} = \pi(i+1)\, M_{i+1,i}. \tag{18}$$

Proof. We prove this by induction. For the base case $i = 0$, we know that

$$\pi(0) = \pi(0)\, M_{0,0} + \pi(1)\, M_{1,0}. \tag{19}$$

Also we know that elements of a row in M sum to 1, that is, $\sum_j M_{i,j} = 1$. Thus, $\pi(0) = \pi(0)M_{0,0} + \pi(0)M_{0,1}$. Therefore,

$$\pi(0)\, M_{0,1} = \pi(1)\, M_{1,0}. \tag{20}$$

Now we assume that the equality holds for the case $i = k$; that is,

$$\pi(k)\, M_{k,k+1} = \pi(k+1)\, M_{k+1,k}. \tag{21}$$

As $\pi = \pi M$, for the case of $i = k+1$ we have

$$\begin{aligned} \pi(k+1) = {} & \pi(k)\, M_{k,k+1} + \pi(k+1)\, M_{k+1,k+1} \\ & + \pi(k+2)\, M_{k+2,k+1}. \end{aligned} \tag{22}$$

Assuming the case $i = k$ holds we have

$$\begin{aligned} \pi(k+1) = {} & \pi(k+1)\, M_{k+1,k} + \pi(k+1)\, M_{k+1,k+1} \\ & + \pi(k+2)\, M_{k+2,k+1}. \end{aligned} \tag{23}$$

Recall that $\sum_j M_{i,j} = 1$; thus

$$M_{k+1,k} + M_{k+1,k+1} + M_{k+1,k+2} = 1. \tag{24}$$

Combining the last two equalities will yield

$$\begin{aligned} \pi(k+1) = {} & \pi(k+1)\left(1 - M_{k+1,k+2}\right) \\ & + \pi(k+2)\, M_{k+2,k+1}, \end{aligned} \tag{25}$$

or equivalently

$$\pi(k+1)\, M_{k+1,k+2} = \pi(k+2)\, M_{k+2,k+1}. \tag{26}$$

□

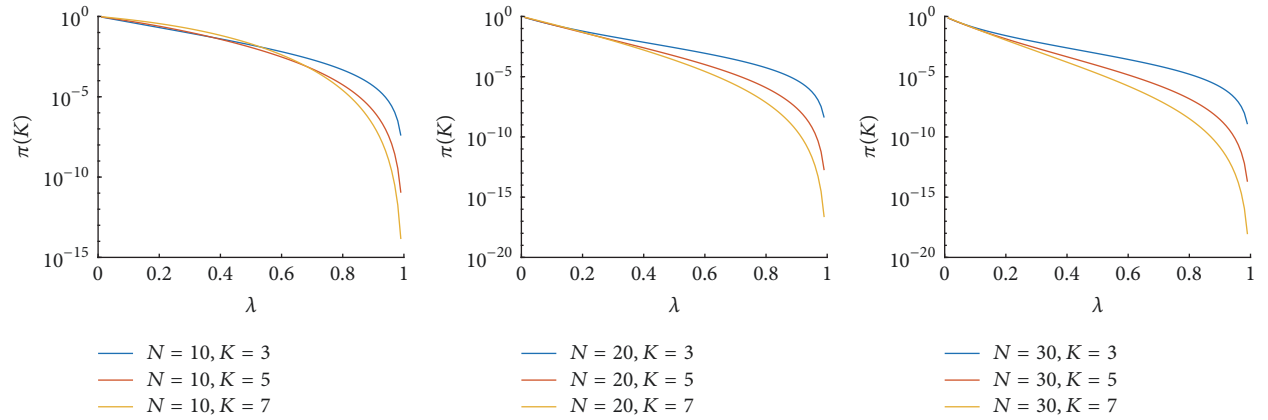

FIGURE 3: Numerical computation results for $\pi(K)$, the winning probability in the stationary state as a function of λ for different values of N and K for the case C1. For any fixed N and K, by increasing λ security of the system increases; that is, $\pi(K)$ decreases.

Theorem 6. *Let M be the $(K+1)\times(K+1)$ transition matrix of the Markov chain of the $G_{1,1}$ game defined in (17). Also let $\pi = (\pi(0)\cdots\pi(K))$ be the stationary probability distribution of M. Then the following inequality holds:*

$$\pi(K) \le \left(\frac{\mu K}{\lambda(N-K+1)}\right)^K. \tag{27}$$

Proof. The Markov chains of $G_{1,1}$ games satisfy Lemma 5. Therefore, using (17) we have

$$\begin{aligned}
\pi(K) &= \pi(0)\prod_{i=0}^{K-1}\frac{\pi(i+1)}{\pi(i)} = \pi(0)\prod_{i=0}^{K-1}\frac{M_{i,i+1}}{M_{i+1,i}} \\
&= \pi(0)\prod_{i=0}^{K-1}\frac{\mu((K-i)/(N-i))}{\lambda((i+1)/K)} \\
&= \pi(0)\times\left(\frac{\mu K}{\lambda}\right)^K\times\prod_{i=0}^{K-1}\frac{K-i}{(N-i)(i+1)} \\
&\le \left(\frac{\mu K}{\lambda(N-K+1)}\right)^K.
\end{aligned} \tag{28}$$

The second equality is due to Lemma 5, and the third equality is due to the transition probabilities of $G_{1,1}$. Note that for the case of risk-taking adversary $\mu = 1-\lambda$. □

6. Numerical Results for $G_{1,1}$

To study the effect of different system parameters on the system security, we calculated security measures of the two MTD games for different values of system parameters and graphed the results. We used the results of Section 4 to calculate $\pi(K)$ and $E^{(1)}_{\text{win}}$ for different choices of N, K, λ, and μ. We used MATLAB for our calculations. For each set of system parameters N, K, λ, or μ, using (6)–(14), we first obtained the transition matrix M. For C1 λ is varied from 1% to 99% in steps of 1%. For C2 λ is varied from 1% to $1-\mu$ in steps of 1%.

To calculate $E^{(1)}_{\text{win}}$, we used the results of Theorem 4 and employed the linear equation solver `(linsolve)` of MATLAB to solve the set of equations $\mathbf{v} = \mathbf{r} + \widetilde{M}\mathbf{v}$ for each parameter set. For large values of K, M becomes near-singular and an exact solution cannot be found (in fact, for large values of K and large values of λ a MATLAB warning occurred due to the equation being near-singular and thus only an approximation of the answer was calculated. We have excluded those cases from our analysis and the graphs are only included with exact results here). This explains the choice of $K < 10$ in our graphs. The choice of N determines the computation time but otherwise is not restricted. We chose values of N and K such that the cases of $K < N/2$ and $K > N/2$ both are shown in the graphs.

To calculate $\pi(K)$ we used MATLAB eigenvector analysis function `([V,D]=eig(M'))` to find for the stationary distribution π where $\pi M = \pi$. The stationary distribution is obtained by normalizing the eigenvector that corresponds to the eigenvalue + 1. We could use N up to 1000 and K up to 200 in this analysis. For the choice μ we ensured that $\mu < 0.5$, $\mu = 0.5$, and $\mu > 0.5$ are represented. The results of the above calculations are graphed.

6.1. Case C1. The results of numerical computations for different settings are depicted in Figures 3, 4, and 5.

Stationary Probability. For fixed N and K, as λ increases $\pi(K)$ decreases. For fixed N and λ, when K increases, $\pi(K)$ decreases. Both imply better security as expected from more dynamic systems (Figure 3).

Figure 5(b) shows that, for fixed N and λ, increase in K results in the reduction of $\pi(K)$ and increase in $E^{(1)}_{\text{win}}$ and so better security. However, this gain in security diminishes after K reaches a threshold. In this last case almost all paths are target paths and so attacker's chance of correctly guessing is high. The thresholding behaviour suggests using higher K (and so higher system cost) will not have a substantial effect on security.

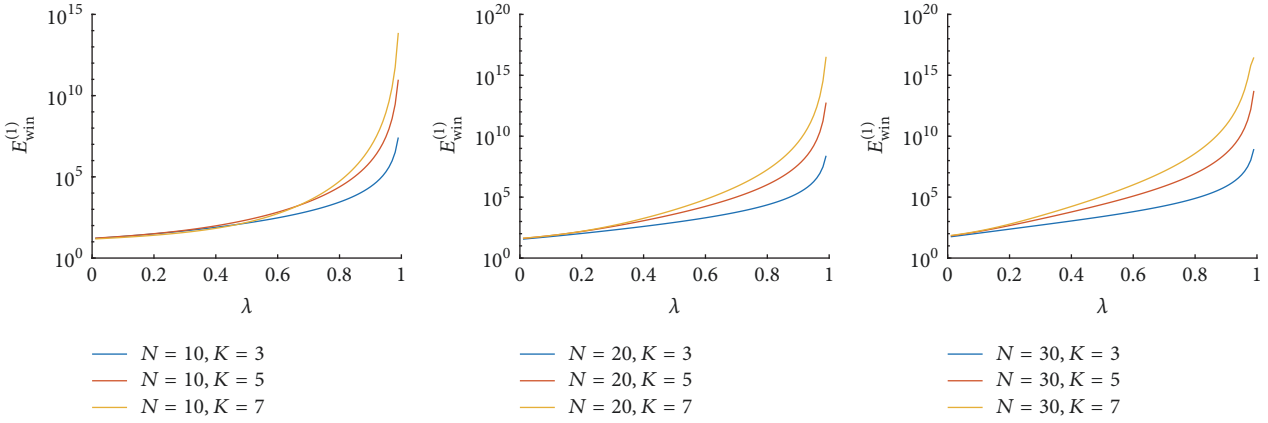

FIGURE 4: Variation of $E^{(1)}_{\text{win}}$ as a function of λ for different values of N and K for C1, using analytic calculations. Similar to $\pi(K)$, for any fixed N and K, increasing λ improves security of the system; that is, $E^{(1)}_{\text{win}}$ increases.

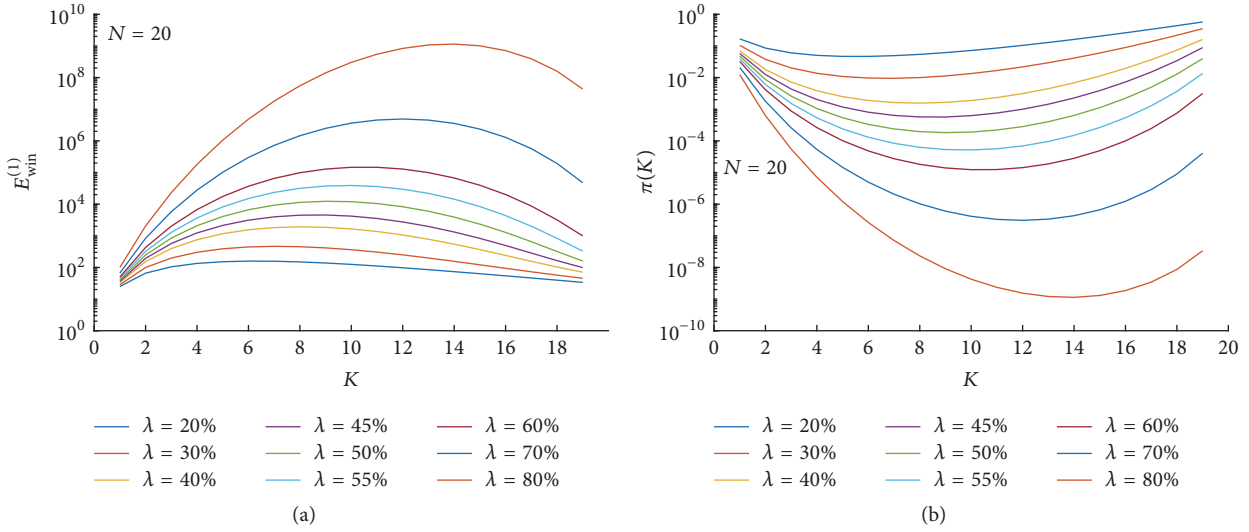

FIGURE 5: Results for security measures $\pi(K)$ and $E^{(1)}_{\text{win}}$ as functions of K for $N = 20$ and different values of λ. Note that for large values of λ the system can remain secure even if $K = N - 1$ as $E^{(1)}_{\text{win}}$ remains very large (a) or $\pi(K)$ remains very close to zero (b).

The figures also show that, for fixed K as λ increases, $\pi(K)$ decreases. For example, for $N = 30$, $K = 3, 5$, and 7, for all $\lambda \geq 0.5$, we have $\pi(K) \leq 10^{-3}$ (see Figure 3).

Expected Number of Steps to the First Win $E^{(1)}_{\text{win}}$. Figure 4 shows that, for given parameters N and K, increasing λ increases the security of the game. Moreover, we observe that $E^{(1)}_{\text{win}}$ behaves linearly in the log graphs of Figure 4 as λ increases, and it increases faster for higher λ's. Therefore, with increasing λ, the risk-taking attacker needs much longer time to compromise the system.

We also observe that, for given N and λ, as K increases, after a certain threshold value, $\pi(K)$ increases. For example for $N = 20$, $K = N - 1 = 19$, and $\lambda = 0.2$, this threshold is $\pi(K) = 0.57$ (see Figure 5(b)). For $\lambda > 0.5$ however, even for $K = N - 1$, $\pi(K)$ remains small (very close to zero). The same behaviour exists for $E^{(1)}_{\text{win}}$: for given N and λ, the expected number of steps to the first win of the adversary decreases as K increases. However, decrease in security is negligible if λ is sufficiently large (see Figure 5(a)).

6.2. Case C2. The stationary distribution for given N, K, λ, and μ is given in Figure 6. We consider three different values of $\mu = 0.3, 0.5$, an 0.7.

We also show our results for $E^{(1)}_{\text{win}}$ in Figure 7. The figures demonstrate that the defender achieves better security by choosing higher λ's.

Figure 6 shows that, for fixed values of N, K, and μ, the security of the system increases with the increase in λ, represented by the reduction in $\pi(K)$. We also observe that, similar to the case C1, for fixed K, λ, and μ, we gain more security with increasing N. Moreover, as can be expected, smaller values of μ will let defender to reach higher security by increasing λ.

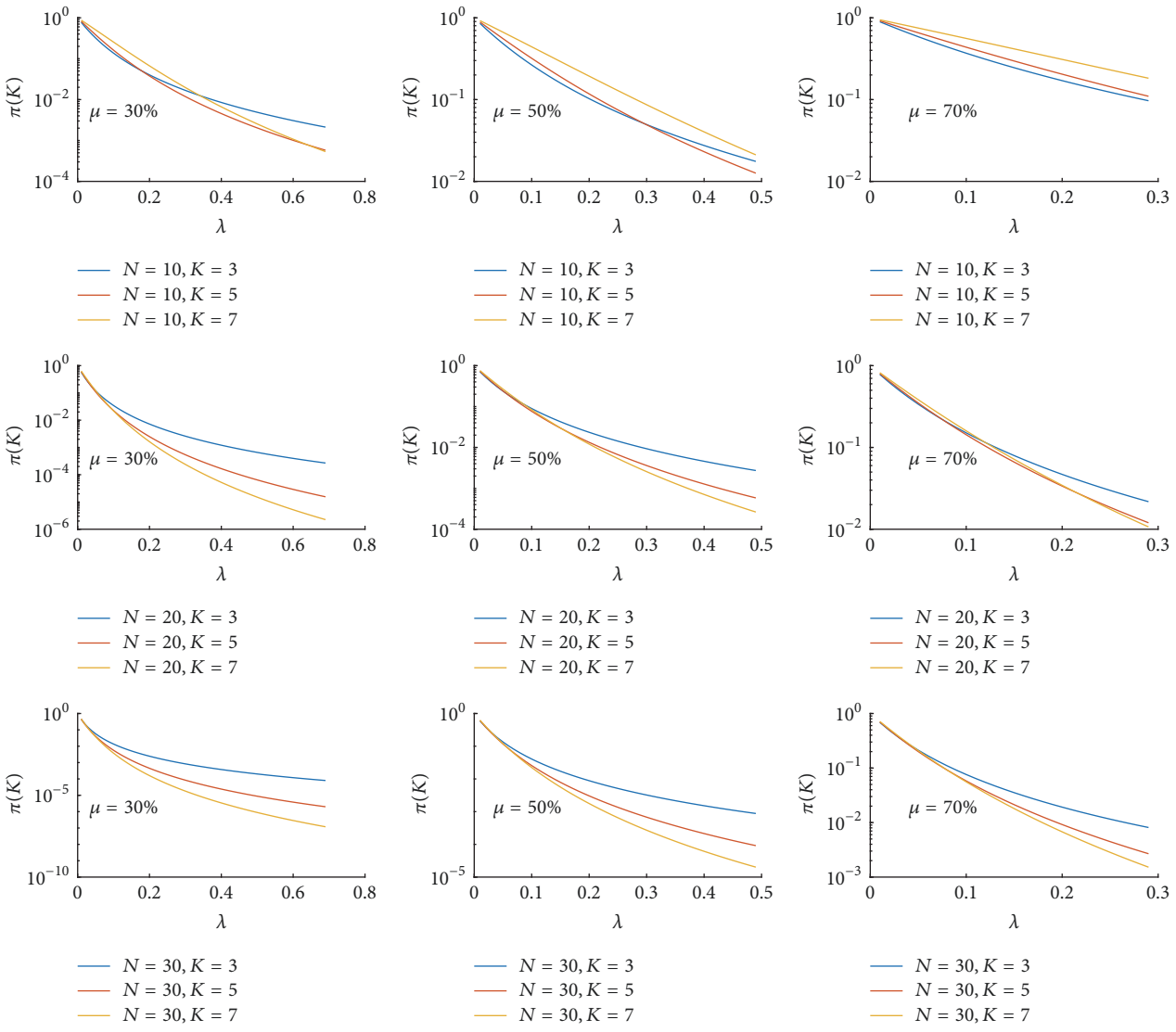

FIGURE 6: Numerical computation results for $\pi(K)$, the winning probability in the stationary state as a function of λ for different values of N and K for the case C2. For any fixed N, K, and μ, by increasing λ security of the system increases; that is, $\pi(K)$ decreases.

Figure 7 shows our analytical calculation results of $E^{(1)}_{\text{win}}$. Again for fixed values of N, K, and μ, the security of the system increases with the increase in λ, and this is shown by the increase in $E^{(1)}_{\text{win}}$. It can also be seen that for fixed values of K, λ, and μ, more security can be gained with the increase in N. The figures also indicate that smaller values of μ allows the defender to increase security of the system by increasing λ.

As it was discussed in Section 7, we can calculate the cost of being a risk-averse adversary in terms of $E^{(1)}_{\text{win}}$. Figure 8 shows the behaviour of this cost (penalty) as a function of $\lambda < 1 - \mu$.

7. Utilities

There are costs and gains associated with the defender's and the attacker's actions. Below we outline important aspects of utilities of the two players and note that our basic modelling and numerical analysis could provide insight into better quantifying these utilities. More concrete analysis and estimation of utilities require considering specific realization of path hoping systems and more detailed modelling and numerical computations.

Defender's cost in state i is denoted by $c_i^{D,\epsilon}$ and c_i^{D,d_1} for no action and action d_1, respectively.

(i) $c_i^{D,\epsilon}$ is the cost of the defender if they do not move. In this case, the defender does not need to bear any resource cost; however, the chance of the attacker winning in the next time step would increase because the attacker may capture additional target paths in the next time step and the state will change to $j > i$. The defender, however, does not know the exact state of the system, i, and so their cost would be an expected value, where expectation will be taken over the stationary probabilities of the system.

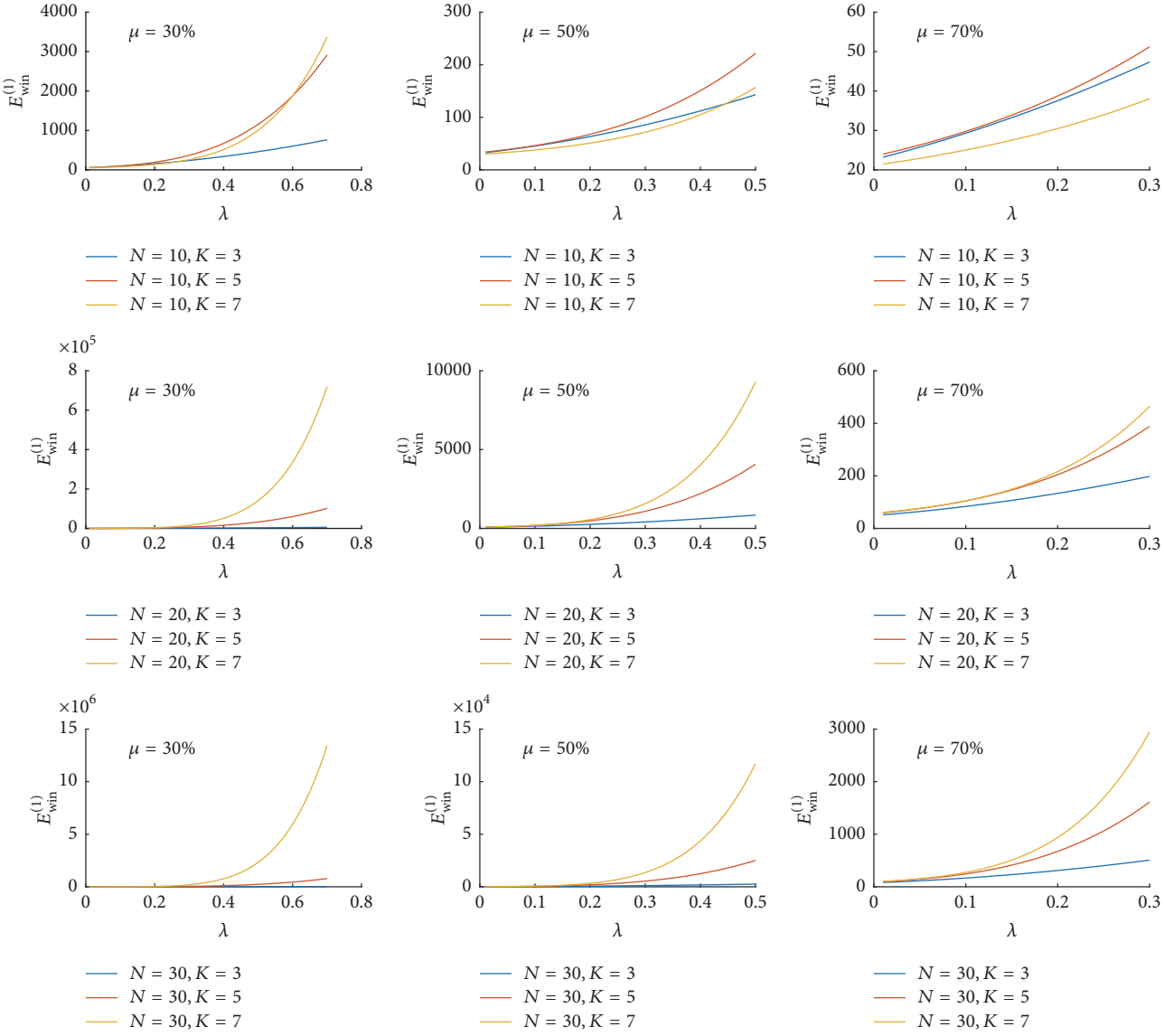

FIGURE 7: Variation of $E^{(1)}_{win}$ as a function of λ for different values of N and K for C2, using analytic calculations. Similar to $\pi(K)$, for any fixed N, K, and μ, increasing λ improves security of the system; that is, $E^{(1)}_{win}$ increases.

(ii) c_i^{D,d_1} is the cost of randomization. This will be a fixed cost (that could depend on the number of of paths that will be randomized in one time slot) associated with redirecting traffic, possibly packet losses, and delays to reestablish the paths.

Attacker's costs in state i is denoted by $c_i^{A,\epsilon}$ and c_i^{A,a_1} for no action and action a_1.

(iii) $c_i^{A,\epsilon}$ is the cost of the attacker not taking action. They do not need to spend resources, but their success chance would reduce because the defender's action in the next time step may result in one of the target paths in S_A^i to be removed from the target path set. As discussed below, this cost can be estimated in terms of increase in $E^{(1)}_{win}$.

(iv) c_i^{A,a_1} is the cost of launching the attack in state i and indicates the resources that the attacker must spend to realize the attack. This cost would be fixed as long as the attack rate is below τ and will increase if the attack rate μ is above τ.

One can also consider gains associated with actions of the attacker and the defender. Utilities of the players will be a function of the costs and the gains.

Estimating $c_i^{A,\epsilon}$. A risk-averse adversary will use an attack rate below $1 - \lambda$ and so, with probability $1 - \lambda - \mu$, there is no action from the attacker. Intuitively, no action means that the attacker will have a reduced success chance in breaking security. This can be quantified by the larger expected number of time steps to win for the first time. For example, consider $N = 30$, $K = 7$, and $\lambda = 0.6$. A risk-averse adversary who moves with probability $\mu = 0.3 < 1 - 0.6 = 0.4$ will have $E^{(1)}_{win}(\mu = 0.3, \lambda = 0.6) = 6 \times 10^6$; however, for the same values

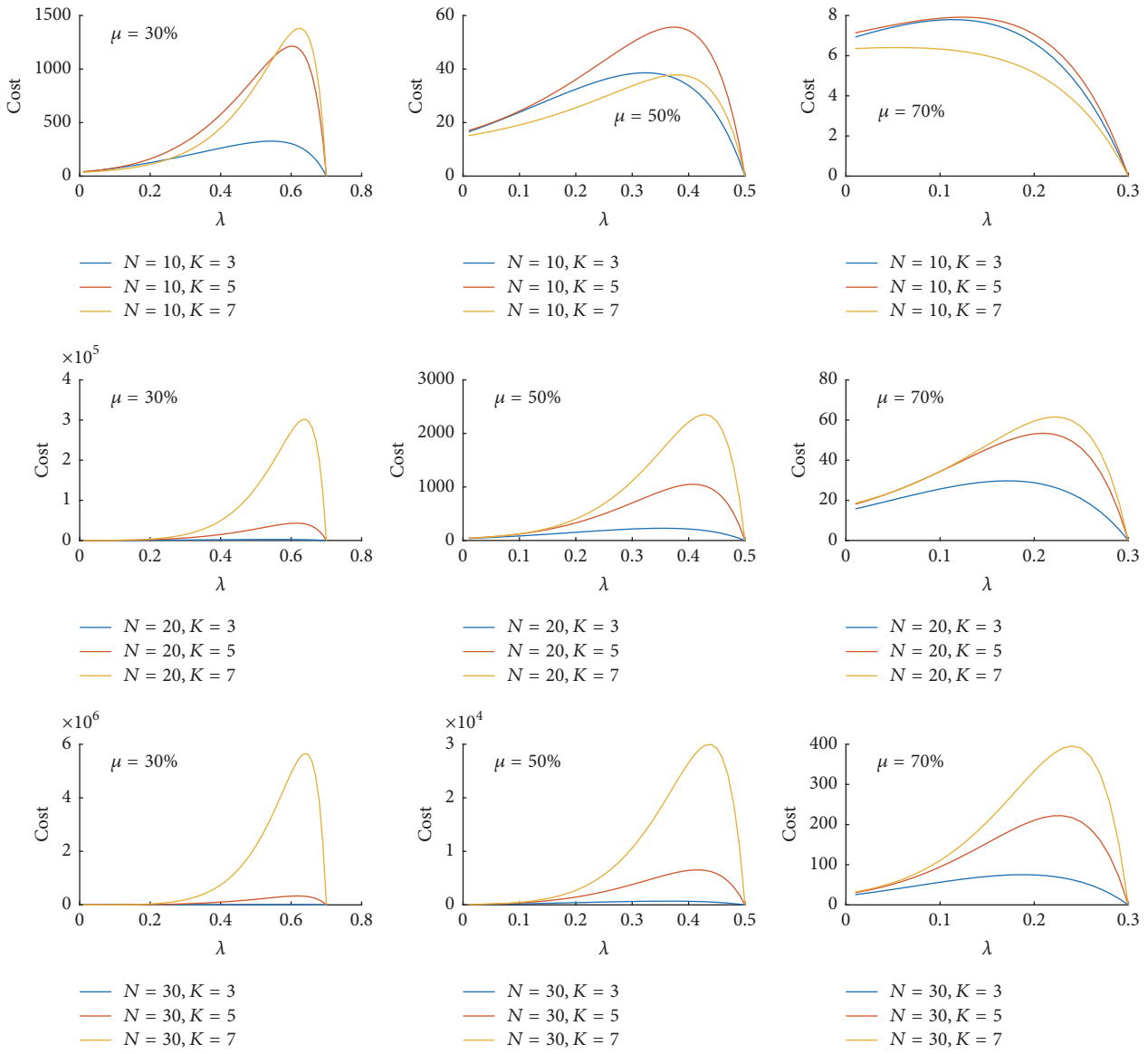

FIGURE 8: The cost of being risk-averse in terms of $E^{(1)}_{\text{win}}$ as a function of λ for different values of N, K and μ.

of N and K, a risk-taking attacker with $\mu = 1 - \lambda = 0.4$ will have $E^{(1)}_{\text{win}}(1-\lambda,\ \lambda = 0.6) \approx 1.0 \times 10^6$. This shows that a risk-averse adversary is paying the penalty of being risk-averse and (most likely) must wait for its first win much longer than a risk-taking adversary. By defining the cost of being risk-averse as $RiskAverseCost = E^{(1)}_{\text{win}}(\mu,\lambda) - E^{(1)}_{\text{win}}(1-\lambda,\lambda)$ for a risk-averse adversary with probability of moving μ, we can graph the behaviour of this cost (penalty) as a function of $\lambda < 1 - \mu$.

Figure 8 shows that smaller μ (being more risk-averse) will have higher costs, and as μ increases, the cost decreases. However, as defender increases λ, the available attack rate of the attacker $(1 - \lambda)$ decreases and after a certain threshold value of λ, the cost of being risk-averse decreases and becomes 0 when $\lambda = 1 - \mu$.

8. Concluding Remarks

We introduced path hopping as an approach to providing efficient long-term cryptographic security for communication against an adversary with access to a quantum computer. We considered a general class of dynamic strategies that can be modelled as a Markov chain that models the attacker's and the defender's interaction and gave detailed analysis of $G_{1,1}$. Our work opens new directions for future work including considering more complex set of actions for the players. For example, allow defender and attacker to choose λ and μ in each state, and/or use different values of u and v at different state. Including the actual costs of the defender and the attacker in the modelling and analysis will unravel the limits of randomization strategies in practice.

Also, an important question is to efficiently provide sufficient shared randomness for the sender and the receiver, to be used in the path hopping algorithm. For our analysis, we assumed that sufficient number of random bits has been shared by the sender and receiver through another protocol, before transmission starts. Sharing true random bits without computational assumptions in practice requires sharing a random pad which would restrict application of the system in practice. Efficient sharing of randomness can be achieved by using a pseudorandom number generator and only sharing an initial random seed. Security of the resulting system, however, needs to be analyzed to ensure that this will not affect postquantum security of the system. This will be an interesting problem for future research. There are also similar games that can be modelled using this approach. For example, one can consider the path hopping game associated with the model in [21].

Disclosure

A preliminary version of this paper was presented at the 2017 Workshop on Moving Target Defense (MTD '17), Dallas, Texas, USA, October 30, 2017.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was partially supported by Natural Sciences and Engineering Research Council of Canada and by TELUS Communications.

References

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[2] R. Brandom, "Microsoft lab predicts a working quantum computer within 10 years," 2015, http://www.theverge.com/2015/10/15/9539033/working-quantum-computer-prediction-ten-years-microsoft.

[3] D. Evans, Top 25 Technology Predictions—Cisco Systems, 2009, https://www.cisco.com/c/dam/en_us/about/ac79/docs/Top_25_Predictions_121409rev.pdf.

[4] S. Dahmen-Lhuissier, Quantum-Safe Cryptography, 2016, http://www.etsi.org/technologies-clusters/technologies/quantum-safe-cryptography.

[5] L. Chen, S. Jordan, Y. Liu et al., NISTIR 8105 Draft—Report on Post Quantum Cryptography, 2016, http://csrc.nist.gov/publications/drafts/nistir-8105/nistir_8105_draft.pdf.

[6] M. Braithwaite, Experimenting with Post-Quantum Cryptography, July 2016, https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html.

[7] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila, "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem," in *Proceedings of the 36th IEEE Symposium on Security and Privacy (SP '15)*, pp. 553–570, IEEE, San Jose, Calif, USA, May 2015.

[8] E. Brickell, "Intel strategy for post quantum crypto," in *Proceedings of the 7th International Conference on Post-Quantum Cryptography*, (Invited Talk), Fukuoka, Japan, 2016, https://pqcrypto2016.jp/data/Brickell-Post_Quantum_Strategy-PQC_2016_final.pdf.

[9] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, article 34, 2009.

[10] X. Zhou, L. Song, and Y. Zhang, *Physical Layer Security in Wireless Communications*, CRC Press, 2013.

[11] A. D. Wyner, "The wire-tap channel," *Bell Labs Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[12] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: how to cope with perpetual leakage," in *Advances in Cryptology—CRYPT0 '95*, pp. 339–352, Springer, Berlin, Germany, 1995.

[13] H. Maleki, S. Valizadeh, W. Koch, A. Bestavros, and M. Van Dijk, "Markov modeling of moving target defense games," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, pp. 81–92, ACM, 2016.

[14] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[15] M. O. Rabin, "The information dispersal algorithm and its applications," in *Sequences*, pp. 406–419, Springer, 1990.

[16] H. Krawczyk, "Secret sharing made short," in *Proceedings of the Annual International Cryptology Conference*, pp. 136–146, Springer, 1993.

[17] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[18] D. Dolev, C. Dwork, O. Waarts, and M. Yung, "Perfectly secure message transmission," *Journal of the ACM*, vol. 40, no. 1, pp. 17–47, 1993.

[19] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, 2006.

[20] M. Strasser, C. Pöpper, S. Capkun, and M. Cagalj, "Jamming-resistant key establishment using uncoordinated frequency hopping," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '08)*, pp. 64–78, Oakland, Calif, USA, May 2008.

[21] H. Ahmadi and R. Safavi-Naini, "Multipath private communication: an information theoretic approach," https://arxiv.org/abs/1401.3659.

[22] K. Scott and J. Davidson, "Strata: a software dynamic translation infrastructure," Tech. Rep., University of Virginia, Charlottesville, Va, USA, 2001.

[23] N. Nethercote and J. Seward, "Valgrind: a framework for heavy-weight dynamic binary instrumentation," in *Proceedings of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '07)*, pp. 89–100, ACM, San Diego, Calif, USA, June 2007.

[24] H. Okhravi, A. Comella, E. Robinson, and J. Haines, "Creating a cyber moving target for critical infrastructure applications using platform diversity," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 1, pp. 30–39, 2012.

[25] A. Saidane, V. Nicomette, and Y. Deswarte, "The design of a generic intrusion-tolerant architecture for web servers," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 1, pp. 45–58, 2009.

[26] H. Okhravi, T. Hobson, D. Bigelow, and W. Streilein, "Finding focus in the blur of moving-target techniques," *IEEE Security & Privacy*, vol. 12, no. 2, pp. 16–26, 2014.

[27] K. M. Carter, J. F. Riordan, and H. Okhravi, "A game theoretic approach to strategy determination for dynamic platform defenses," in *Proceedings of the 1st ACM Workshop on Moving Target Defense (MTD '14)*, pp. 21–30, ACM, Scottsdale, Ariz, USA, November 2014.

[28] R. Colbaugh and K. Glass, "Predictability-oriented defense against adaptive adversaries," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '12)*, pp. 2721–2727, October 2012.