Dynamic Cyber-Physical System Security Planning using Markov Chains and Attack Graphs

Carmen Haseltine
University of Wisconsin-Madison
Electrical Engineering
Madison, Wisconsin, USA
haseltine@wisc.edu

Shaonan Wang
University of Wisconsin-Madison
Industrial and Systems Engineering
Madison, Wisconsin, USA
swang785@wisc.edu

Laura A. Albert
University of Wisconsin-Madison
Industrial and Systems Engineering
Madison, Wisconsin, USA
laura@engr.wisc.edu

ABSTRACT

Cyber-physical system security planning requires identifying vulnerabilities as well as selecting and deploying security mitigations to manage risk across multiple agents interacting with the system. Attack graphs provide a mathematical framework for characterizing cybersecurity risks and can be used to inform risk management decisions, such as the prioritization of security mitigations. However, attack graphs typically consider a static representation of risk, simple representations of security mitigations, and a single agent. We address these limitations by exploring how an attack graph analysis can be enhanced to assess temporal aspects of risk and mitigation, resource limitations, multiple agents, and interdependencies between security mitigations. We introduce a discrete-time Markov chain that considers vote-by-mail election processes with linkages to the attack graphs and mitigations. We illustrate the issues in a case study based on vote-by-mail election processes with the participation of multiple agents including election officials and voters. We contribute a new Markov chain model, data set, and analysis that allows for a descriptive and predictive assessment of risk in vote-by-mail election that provides insight into how to manage dynamic risks in resource-constrained infrastructure systems.

KEYWORDS

Cybersecurity; Security planning; Elections; Markov Chains

ACM Reference Format:
Carmen Haseltine, Shaonan Wang, and Laura A. Albert. 2022. Dynamic Cyber-Physical System Security Planning using Markov Chains and Attack Graphs. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Auckland, New Zealand, May 9–13, 2022*, IFAAMAS, 8 pages.

1 INTRODUCTION

Critical infrastructure comprises of multiple cyber-physical control and operation systems. As such, these processes must be protected from both malicious and non-malicious attacks and requires participation from multiple agents who use and manage the infrastructure systems. The protection of cyber-physical systems requires a rigorous technical analysis to identify vulnerabilities, evaluate security mitigations, and assess consequences. A key component of cyber-physical security is understanding all potential threats to a system to inform defensive planning decisions. This has traditionally been performed through the use of attack graph models specific to the infrastructure systems under consideration. Attack graphs describe threats to a system using formal methods, where the threats could be malicious or non-malicious [9]. They model all possible "paths" of an attack using Boolean logic gates (and, or), where each path begins with a specific attack node with the intent to disrupt a key part of the infrastructure system, and all branches show the multiple paths an attacker can use to carry out their goal [9]. Attack graphs organize vulnerabilities and visualize their dependencies, intended to provide an overall view of a system vulnerabilities.

Protecting a system requires knowledge of the system processes, the vulnerabilities that exist, and available security mitigations. Often these security mitigations are modeled as countermeasures that block attacks or reduce their likelihood of success. There are three primary aspects of attack graph modeling that are explored in this study. First, attack graphs reflect a static modeling of vulnerabilities and countermeasures in a system with respect to time. This is a crucial point, since the performance of many infrastructure systems often depend on the timing of attacks, countermeasures, and system processes. A countermeasure may only interdict an attack, for example, if it is implemented with adequate time, resulting in attacks that can have different consequences based on the timing of the attack and the associated mitigations. Likewise, an attack-mitigation event can cause delays that degrade overall system performance. Second, attack graphs do not directly consider the resources available to implement or deploy mitigations. For example, inadequate resources can introduce congestion that can result in additional delays. Third, attack graphs typically do not directly consider the interdependencies between mitigations, such as precedence requirements. These three issues are salient for dynamic security planning in resource-constrained infrastructure systems.

In this paper, we study the impact of these three assumptions in an attack graph modeling framework applied to a specific case study involving vote-by-mail (VBM) election processes. In particular, we study the impact of dynamic aspects of protection, resource limitations, and the interdependencies between security mitigations to highlight the importance of these issues in planning as well as their implications on system performance. To do so, we introduce a discrete-time Markov chain model that captures the overall VBM process as well as attack graphs and mitigations. Designing VBM processes that are resilient and require the participation of multiple agents including election officials and voters is crucial for fully supporting civic participation in elections. We contribute a new data set of VBM mitigations, a novel discrete-time Markov chain model, and an analysis of risk mitigation in VBM election processes against an array of vulnerabilities.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Auckland, New Zealand.

2 LITERATURE REVIEW

In recent years there has been increased interest in the protection of cyber-physical systems due to the nature of modern critical infrastructure. Enayaty-Ahanger et al. [4] provide a survey of optimization models for cyber-infrastructure security. Many of the studies into cyber-physical systems build from the core attack graph cybersecurity model to incorporate physical attacks. Cheh et al. [2] studies the use of "meta model" attack graph known as "ADversary VIew Security Evaluation (ADVISE)" to evaluate the cyber-physical security of a railway station. Zheng et al. [14] studies the selection of mitigations to manage cybersecurity risks in resource constrained systems using attack paths. The dependencies of the mitigations are modeled using multiple choice constraints directly and coverage models indirectly. Zheng and Albert [13] build on this research to study temporal issues in infrastructure protection. They introduce a bi-level network interdiction model to study how to maximally delay the total weighted attack times of multiple adversarial, however, they do not consider the timing of security mitigations. In fact, we found no studies that consider the timing of the attacks or the security mitigation dependencies. In this paper, we address this knowledge gap through the investigation of election systems, a particular critical infrastructure cyber-physical system.

A stream of papers have studied how to protect voting systems against security vulnerabilities by considering the protections required. The US Election Assistance Commission published a report in 2009 detailing an attack tree analysis of all possible threats to election processes, including 42 risks to VBM processes [1]. More recently, the Cybersecurity and Infrastructure Security Agency published an assessment of mail-voting security prior to the 2020 General Election [3]. The majority of papers have focused on security of in-person voting. Simidchieva et al. [10] uses fault trees to analyze the in-person election process in reference to the overall goal of ballots being counted. Fault trees are similar to attack trees in that both focus on an overall attacker goal (attack tree) or hazard (fault tree) in a process. The fault tree analyses of an election process accounts for processing errors. It is for this reason the results could be used to identify potential areas for improvement. A separate attack tree risk assessment of voting systems focuses on malicious cyber-attacks to in-person voting machines [12]. Though these papers perform a risk analysis of specific concerns for equipment failures and cybersecurity aspects of the voting process, none systematically evaluates the impact of various risks to the entire cyber-physical voting system.

Few research efforts have identified risks associated with VBM processes. Scala et al. [7] develop a detailed, step-by-step model of the VBM process to identify the overall process components. However, this model does not directly account for attacks and mitigations possible in VBM processes. In our paper, we address this limitation by creating a Markov chain model that can account for (simplified) VBM process components as well as attack and mitigation scenarios.

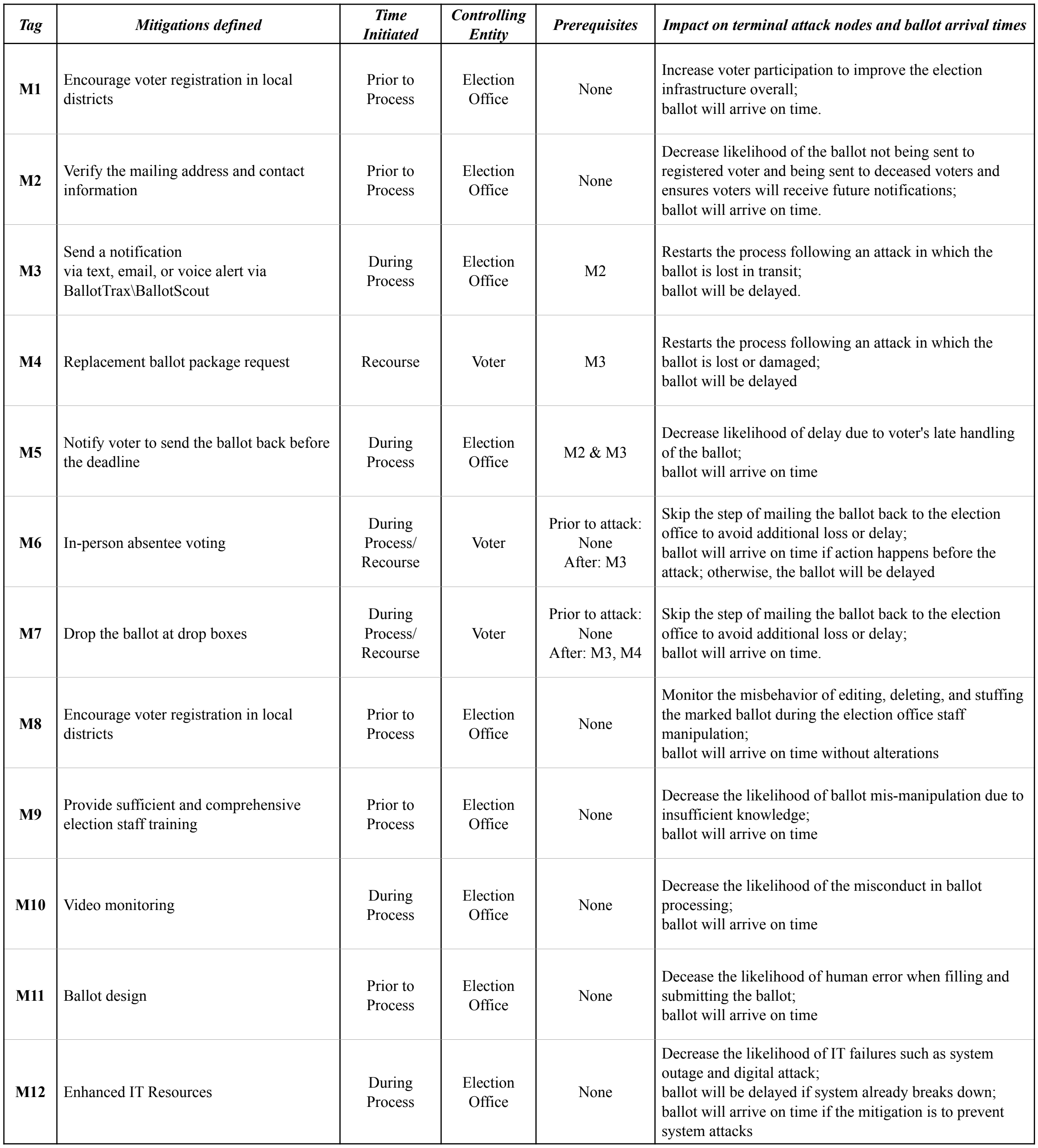
3 VOTE-BY-MAIL CASE STUDY

In this paper we explore aspects of cyber-physical infrastructure protection though a case study of the VBM process utilized in many precincts during the 2020 General Election in the United States (US). Elections are a part of our nation's critical infrastructure, and VBM is a cyber-physical system with time- and resource-constrained interactions between election offices and voters. Elections have predetermined deadlines, thus making elections favorable for studying temporal issues. The 2020 General Election occurred while many states operated under emergency orders related to the COVID-19 pandemic. Record numbers of voters chose to vote absentee using VBM to limit virus exposure. In the 2020 General Election, voting by mail accounted for 46% of all ballots cast in the US according to election evaluation metrics [5], more than any prior general election and motivating a number of studies and reports regarding VBM security. We acknowledge that there is not a single process for VBM, since elections in the US are organized at the state and local levels. However, VBM processes are largely the same in different locals, and therefore, the model we propose is high level and easily modified for different locations. Since each election is a one-time event with no "redo" option, election officials must plan for a range of potential conditions and scenarios prior to an election to allocate limited resources effectively. There exists no way to test election operations at scale prior to an election, which drives the need for a rigorous analytical approach to security planning.

The US Election Assistance Commission report [1] provides initial insight into vulnerabilities of VBM processes. These attacks are separated into several main branches: insider attacks, masquerade, process attacks, errors in the voting system process. More recently, Scala et al. [8] revised the VBM process attack graph to incorporate 30 additional attack paths created by the newer procedures adopted for absentee voting (drop boxes, in-person absentee voting, ballot validation, etc.). In this study we use the 51 insider terminal attack paths from Scala et al. [8]. In Section 4, we list and discuss insider attacks, a subset of all VBM attack paths, for brevity. The performance measure considered is the number of non-altered marked ballots successfully counted by the voting deadline. For a ballot to be counted, it must arrive on time, be accepted by election officials, and not be altered. To evaluate this performance measure, we consider the effect each attack has on a ballot, captured by the terminal leaf node of the revised attack graph [8]. An attack could result in a ballot either being lost, delayed, or altered.

4 ANALYSIS OF MITIGATIONS AND ATTACK PATHS

Next, we introduce the mitigations and their linkages to the attack paths, each of which captures a vulnerability. We created a mitigation dataset that builds upon those from the Cybersecurity Infrastructure Security Agency report [9], which outlines the risks for the 2020 General Election and proposed countermeasures. From these countermeasures we defined the twelve mitigations summarized in Figure 1. Note that this set represents all possible mitigations, and a subset of these mitigations may be in place in any single voting precinct. Each mitigation is described according to several attributes and its precedence relationships with other mitigations. The "time-initiated" column captures temporal aspects of when the mitigation would need to be implemented. Mitigations put into place prior to the voting process are not time-sensitive. Mitigations put into place during the voting process or as recourse to new information require implementation in real-time and consume the



Tag	*Mitigations defined*	*Time Initiated*	*Controlling Entity*	*Prerequisites*	*Impact on terminal attack nodes and ballot arrival times*
M1	Encourage voter registration in local districts	Prior to Process	Election Office	None	Increase voter participation to improve the election infrastructure overall; ballot will arrive on time.
M2	Verify the mailing address and contact information	Prior to Process	Election Office	None	Decrease likelihood of the ballot not being sent to registered voter and being sent to deceased voters and ensures voters will receive future notifications; ballot will arrive on time.
M3	Send a notification via text, email, or voice alert via BallotTrax\BallotScout	During Process	Election Office	M2	Restarts the process following an attack in which the ballot is lost in transit; ballot will be delayed.
M4	Replacement ballot package request	Recourse	Voter	M3	Restarts the process following an attack in which the ballot is lost or damaged; ballot will be delayed
M5	Notify voter to send the ballot back before the deadline	During Process	Election Office	M2 & M3	Decrease likelihood of delay due to voter's late handling of the ballot; ballot will arrive on time
M6	In-person absentee voting	During Process/ Recourse	Voter	Prior to attack: None After: M3	Skip the step of mailing the ballot back to the election office to avoid additional loss or delay; ballot will arrive on time if action happens before the attack; otherwise, the ballot will be delayed
M7	Drop the ballot at drop boxes	During Process/ Recourse	Voter	Prior to attack: None After: M3, M4	Skip the step of mailing the ballot back to the election office to avoid additional loss or delay; ballot will arrive on time.
M8	Encourage voter registration in local districts	Prior to Process	Election Office	None	Monitor the misbehavior of editing, deleting, and stuffing the marked ballot during the election office staff manipulation; ballot will arrive on time without alterations
M9	Provide sufficient and comprehensive election staff training	Prior to Process	Election Office	None	Decrease the likelihood of ballot mis-manipulation due to insufficient knowledge; ballot will arrive on time
M10	Video monitoring	During Process	Election Office	None	Decrease the likelihood of the misconduct in ballot processing; ballot will arrive on time
M11	Ballot design	Prior to Process	Election Office	None	Decease the likelihood of human error when filling and submitting the ballot; ballot will arrive on time
M12	Enhanced IT Resources	During Process	Election Office	None	Decrease the likelihood of IT failures such as system outage and digital attack; ballot will be delayed if system already breaks down; ballot will arrive on time if the mitigation is to prevent system attacks

Figure 1: Mitigations List

limited voting resources in the days prior to an election. Mitigations that occur during the process require resources for every VBM ballot whereas recourse requires resources only for the ballots that trigger its use. The controlling entity refers to who initiates the mitigation, either the election office or the voter. We note that two mitigations—in-person absentee voting (M6) and placing a ballot in a drop box (M7)—are initiated by the voter and also require the election office putting necessary processes in place ahead of time. The prerequisites column refers to the mitigations that must be in place before the mitigation can be used, which reflects the dependencies between the mitigations. The impact on terminal attack nodes and ballot arrival times capture election performance measures and possible delays.

Attack Tree Description	Effect if successful	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
O 1 Insider attack													
O 1.1 Edit marked ballots													
O 1.1.1 Edit at local elections office													
A 1.1.1.1. Edit during duplication													
T 1.1.1.1.1 (X1) Form collaboration of poll workers	Altered								O		O		
T 1.1.1.1.2 (X2) Gain exclusive access to ballots	Altered								O		O		
T 1.1.1.1.3 (X3) Mark under/over votes or changes votes	Altered								O		O		
T 1.1.1.2 (X4) Edit during counting	Altered								O		O		
T 1.1.1.3 (X5) Edit during other handling	Altered								O		O		
O 1.1.2 Edit in transit													
A 1.1.2.1 Edit in post office													
T 1.1.2.1.1 (X73) Form collaboration with mail worker and acquire access	Altered								O		O		
T 1.1.2.1.2 (X74) Break into post office	Altered								O		O		
T 1.1.2.1.3 (X6) Edit in post office	Altered								O		O		
A 1.1.3.1 Gain exclusive access to intermediate mailroom													
T.1.1.3.1.1 (X75) Form collaboration with mail worker and acquire access	Altered								O		O		
T.1.1.3.1.2 (X76) Break into intermediate mailroom	Altered								O		O		
T 1.1.3.1.3 (X7) Edit in intermediate mailroom	Altered								O		O		
O 1.2 Discard marked ballot													
O 1.2.1 Challenge committed ballot													
O 1.2.1.1 Errant challenge													
T 1.2.1.1.1 (X8) Judge misinterprets rule	Lost			X	X		X			O			
T 1.2.1.1.2 (X9) Errant failed signature	Lost			X	X		X			O			
O 1.2.1.2 Malicious challenge													
T 1.2.1.2.1 (X10) Challenge signature	Lost			X	X		X						
T 1.2.1.2.2 (X11) Challenge postmark	Lost			X	X		X						
T 1.2.1.2.3 (X12) Challenge intent	Lost			X	X		X						
O 1.2.2 Marked ballot lost in the mail													
T 1.2.2.1 (X13) Malicious loss	Lost			X	X		O	O					
T 1.2.2.2 (X14) Accidental loss	Lost			X	X		O	O					
O 1.2.3 Discard marked ballots at local elections office													
A 1.2.3.1 Delete during duplication													
T 1.2.3.1.1 (X15) Form collaboration of poll workers	Lost								O		O		
T 1.2.3.1.2 (X16) Gain exclusive access to ballots	Lost								O		O		
T 1.2.3.1.3 (X17) Overcome controls	Lost								O		O		
T 1.2.3.2 (X18) Remove during counting	Lost								O		O		
T 1.2.3.3 (X19) Mark registration system to reflect duplicate	Lost								O		O		
T 1.2.3.4 (X20) Remove during other handling	Lost								O	O	O		
O 1.3 Miscount duplicated ballots													
A 1.3.1 Count original and duplicate													
T 1.3.1.1 (X21) File duplicate with duplicated ballot	Altered								O	O	O		
T 1.3.1.2 (X22) Defeat ballot accounting	Altered								O	O	O		
T 1.3.2 (X23) Omit original and duplicate	Lost									O			
O 1.4 Marked ballot stuffing													
T 1.4.1 (X24) Insert ballots during envelope separation	Altered								O		O		
T 1.4.2 (X25) Insert ballots during counting	Altered								O		O		
T 1.4.3 (X26) Insert ballots during recount	Altered								O		O		
T 1.4.4 (X27) Insert ballots during audit	Altered								O		O		
O 1.5 Manipulate or discard votable ballot													
O 1.5.1 Delete at local elections office													
T 1.5.1.1 (X28) Fail to stuff envelope	Lost			X	X				O	O			
T 1.5.1.2 (X29) Send wrong or pre marked ballot	Altered			X	X					O			
T 1.5.1.3 (X30) Mis-address envelope (to voter)	Lost			X	X		O						
T 1.5.1.5 (X31) Destroy prepared envelope	Lost								O		O		
T 1.5.1.6 (X32) Destroy batch of prepared envelopes	Lost								O		O		
T 1.5.1.4 (X77) Manipulate return envelope	Lost								O		O		
O 1.5.2 Delay delivery past deadline													
T 1.5.2.1 (X33) Election process delay	Late					O				O			
T 1.5.2.2 (X34) Handling delay	Late					O				O			
T 1.5.2.3 (X35) Delay in the mail	Late					O	O	O					
O 1.5.3 Delete at destination													
T 1.5.3.1 (X36) Lost in destination mailroom	Lost			X	X		O	O					
T 1.5.3.2 (X37) Mailbox attack	Lost			X	X		O	O					
O 1.6 Suppress voter turnout													
T 1.6.1 (X78) Misallocate polling or drop box locations	Lost									O			
T 1.6.2 (X79) Provide regional mail-in voting misinformation	Late									O			
T 1.6.3 (X80) Hinder regional postal services	Late						O	O			O		
T 1.6.3 (X80) Suppress regional postal services	Late						O	O			O		
T 1.6.4 (X81) System outage	Lost												X
T 1.6.5 (X82) Name deliberately misspelled on ballot	Altered										O		
O 1.7 Digital Attack													
T 1.7.1 (X83) Paper ballot scanner hacked	Lost									O			O
T 1.7.2 (X84) Vote denied	Lost			X	X								O
T 1.7.3 (X85) Vote altered	Altered												O

X = Ballot Delayed, O = Ballot On-time

Figure 2: Attack paths for insider attacks, their impact on performance, and their linkages to mitigations

Figure 2 introduces the set of 51 terminal attack nodes associated with insider attack paths and describes the linkages between attacks and mitigations. It reports the effect that each successful attack would have on a ballot, either lost, late, or altered. The mitigations that counteract each attack path either result in the ballot arriving on-time to be counted (denoted, "O") or introduces a delay in the process to counteract the attack (denoted, "X"). We describe attack path X13, which represents a marked ballot being discarded and lost in transit by a malicious attack. This attack can be mitigated through four potential mitigations (M3, M4, M6, and M7). Mitigations M3 and M4 are effective following the X13 attack while the other two mitigations (M6, M7) can be used by the voter as a recourse option. Referencing Figure 1 we see that mitigation M4 depends on M3, and therefore, both M3 and M4 are needed to counter the X13 attack. Likewise, M6 requires M3 being in place, and M7 requires both M3 and M4 being in place.

5 MARKOV CHAIN MODEL

Next, we introduce a discrete-time Markov chain (DTMC) model to capture the voting process as well as multiple facets of a cyber-physical system. The DTMC model captures the overall process, attack and mitigation scenarios, and final ballot system outputs over time. The literature suggests that both the timing of malicious and non-malicious attacks as well as the timing of aspects of the voting process is crucial for understanding performance and how to make improvements. Markov chains are a predictive tool that captures threats and mitigations and can quantify their impact on performance measures. Additionally, Markov chains can help us understand how the timing of attacks and their associated mitigations affect performance measures. Additionally, Markov chains help identify the appropriate timing of the mitigations, thereby providing guidance into policy decisions in support of equity.

In the Markov chain analysis, we first introduce a series of states that capture the status of a voter's ballot over time. The state captures the position of the ballot within the process at the end of a day. We also include states that capture the final status of a ballot that reflect the performance measures. The goal is to capture not only attacks and defenses, but also to capture the process itself. We initially model the VBM process with no attacks or defenses active. First, we define the key points of the VBM process in the perspective of monitoring a ballot through the system. Note that the VBM process may slightly differ by state and municipality, so it is important to capture overall process nodes. Scala specifies forty-seven process nodes for the "Mail-Based Voting Process Map" [7]. These nodes span three physical locations: Post Office, Election Office, and Voter. We propose a simplified base model with seven nodes capturing the VBM process and six final ballot states to better align our analysis with the research questions of concern in this paper. Figure 3 illustrates the base DTMC model defined at time $t = 0$ and describes the final states. Arcs represent the possibility of non-zero transition probabilities in the DTMC model between consecutive days.

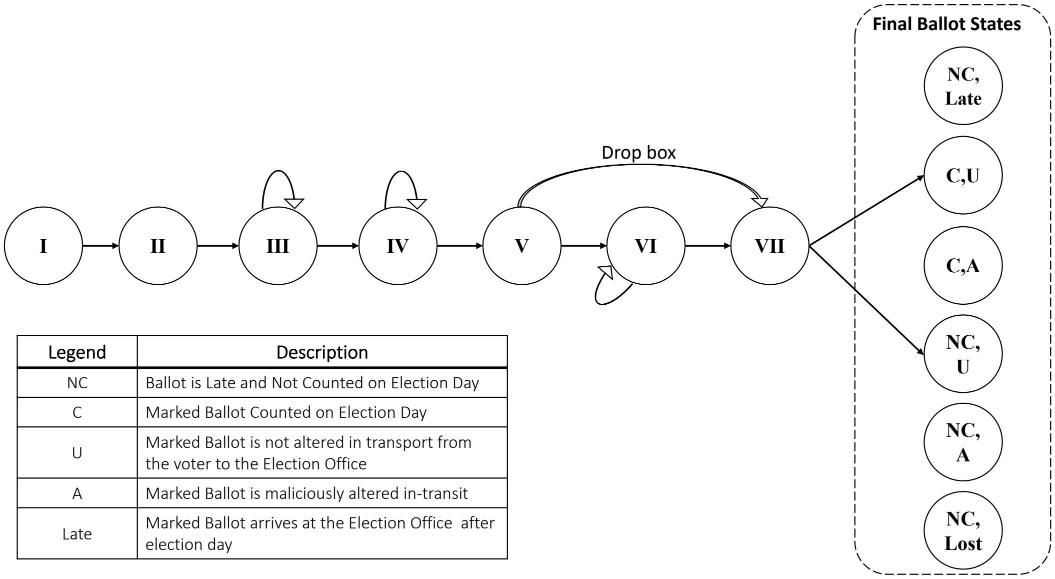
There are seven process states in the base DTMC model. The initial state I where the voter initiates the ballot request. In state II the ballot requests are fulfilled and unmarked ballots are sent to the voter via USPS. State III represents the USPS handling of unmarked ballots. Note that this state can transition to itself to account for processing times in USPS facilities that are longer than one day. In state IV the unmarked ballot reaches the voter, however, it is up to the voter when the marked ballot is filled out and returned. State V is where the voter selects the method of return. This is via drop box or a transition to state VI via USPS. The final process state VII occurs when the marked ballot, reaches the election office and is held until processing on election day. Note that in the base DTMC model, all ballots are unaltered.

In terms of their final status, ballots can be lost and not recovered, altered maliciously, or arrive too late to be counted. Therefore, there are six Markov states corresponding to the final ballot status: Counted, Unaltered; Counted, Altered; Not Counted, Late; Not Counted, Lost; Not Counted, Unaltered; and Not Counted, Altered. Vote-by-Mail process performance is evaluated by the number of ballots that reach the "Counted, Unaltered" state. On election day the DTMC arcs are reconfigured such that all ballots not in $\{V, VII\}$ transition to the state of "Not Counted, Late" (this is not illustrated in Figure 3 for simplicity). Ballots that reach final process state VII transition to either "Counted, Unaltered" or "Not Counted, Unaltered" in the base DTMC model. A ballot may not be counted if it is not signed by the voter or a valid witness (the latter is required in many US states).

The base DTMC model only considers ballots without attacks. Next, we describe how to enhance the base model to include attack and mitigation scenarios. We describe the overall process for evaluating the impact of attacks on performance. Due to the large number of attacks in Figure 2, we illustrate how to create an enhanced DTMC model using attack X13. In this attack, a marked ballot is discarded and lost in transit by a malicious attack, which can result in a ballot being lost or arriving late. This type of attack cannot lead to an altered ballot, so this end state is omitted from our analysis considering this one attack. Attack X13 can be mitigated through four potential mitigations—M3, M4, M6, and M7—and these mitigations can reduce the impact of the attack, require voter action, and alter the ballots' path through the system.

With attack X13 considered, Figure 4 illustrates the DTMC model at time $t = 0$ where the ballots now have the potential to be lost and not recovered. Additionally, the dependence of M4 and M6 recourse actions on the M3 notification is captured as ballots must first go through state M3 prior to the voter having recourse options available. The state associated with notification, M3, has a transition to itself due to the fact that it is possible for the voter to not immediately choose a recourse action. It is important to note that on election day any ballots in states $M3, M4$ will be sent to a "Not Counted, Lost" state.

Next, we briefly summarize the transition probabilities. Due to space constraints, we provide a summary of the transition probabilities, which can be defined from public data [11] and from audit records and performance reports post election [6] as well as risk assessments [8]. While many transition probabilities are static across time, some transition probabilities may reflect the day of the week (e.g., the USPS does not deliver mail on Sundays). Let $t = 0$ capture the earliest point in time when election officials process requests for absentee ballots. The election is held at time T. There are transition



Legend	Description
NC	Ballot is Late and Not Counted on Election Day
C	Marked Ballot Counted on Election Day
U	Marked Ballot is not altered in transport from the voter to the Election Office
A	Marked Ballot is maliciously altered in-transit
Late	Marked Ballot arrives at the Election Office after election day

Figure 3: VBM baseline Markov chain model at $t = 0$ with no attacks

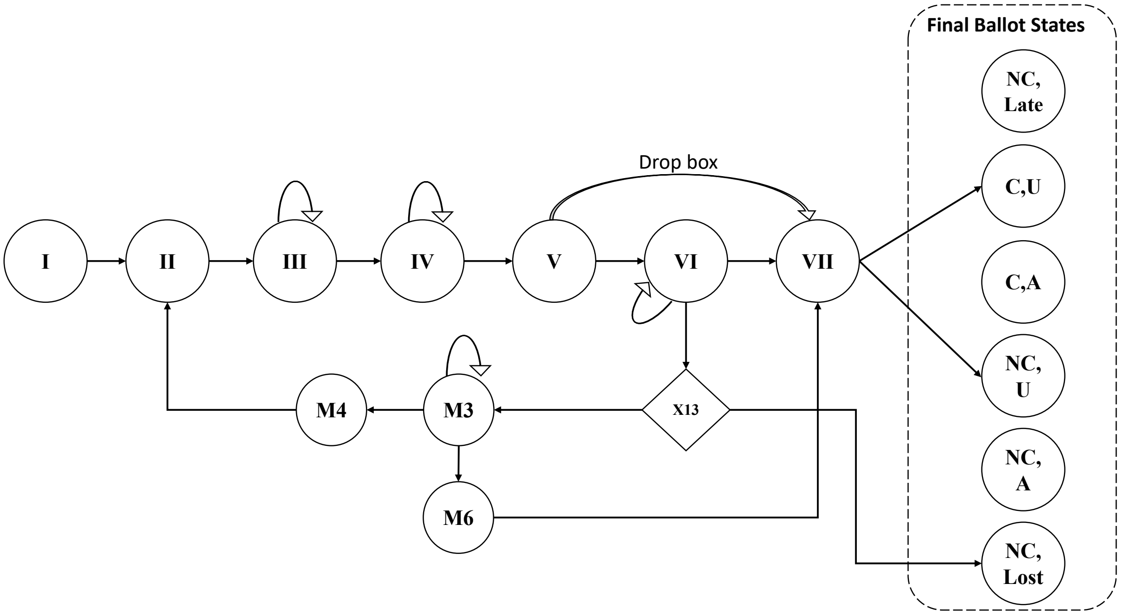


Figure 4: VBM Markov chain at $t = 0$ with attack X13 and mitigations M3, M4, M6, and M7 active

probabilities that move a ballot to its final post-election status in the last time period, $t = T + 1$, to evaluate the performance measures.

Let P_t capture the transition probability matrix immediately after simulation time $t = 0, 1, ..., T$. Voters can request ballots at different times. Let β_t capture the number of ballots requested at time $t = 0, ..., T$, which reflects the distribution of times when voters request absentee ballots. For ballots requested at time t, let α_t be the vector of the probability mass function for ballot starting states, where all ballots typically start in state I, i.e., $\alpha_t(I) = 1$. Then, for a ballot requested at time t we can compute the vector of state probabilities at the end of the time horizon ω_t at stage $T + 1$ after all ballots have been evaluated and transitioned to one of the ballot status states.

$$\omega_t = \alpha_t \prod_{t'=t}^{T} P_{t'},\ 0 \le t \le T. \qquad (1)$$

Then, we can compute the overall distribution of final ballot statuses as $\sum_{t=0}^{T} \beta_t \omega_t$.

6 ANALYSIS

We can now illustrate how the timing of both the attack and mitigations are instrumental in evaluating the impact on VBM system performance. Consider a single malicious attack of a ballot lost in the mail ($X13$). The DTMC model in Figure 4 shows the model changes associated with incorporating $X13$ and related mitigations M3, M4, M6, and M7. The mitigations are reflected as Markov states except for M7. This is due to the fact that drop box usage requires less than one day to reach the next state of the process, therefore, M7 is reflected by the arc from state V to state VII.

With the DTMC model in Figure 4, for brevity we evaluate the ballot paths of three voters using the VBM process labeled A, B, and C. All three voters receive an absentee ballot (state IV) fifteen days prior to the election ($t = 0$) and correctly fill out their ballot. The three voters are differentiated by when they mail back their ballot: fourteen days, seven days, and three days prior to the election, respectively. In the DTMC model this is reflected as the time in which the ballots enter state V after the voter fills out the ballot. The ballot then moves to state VI when the ballot is placed in the mail. Voters are typically instructed to submit their absentee ballot three days prior to the election deadline to ensure it arrives in time to be counted, and therefore, all voters returned their ballot "on-time." All mitigations listed in Figure 1 are available, however, only the mitigations associated with attack $X13$ are shown in Figure 4.

Each voter's process can be viewed as a single realization of the DTMC model, indicating the states the ballot travels through at different times when starting in state IV at time $t = 0$. The election occurs at $t = 15$, when ballots are scanned and recorded. Figure 5 captures the voting process for each voter over time, including the attack and mitigations. The example studied in Figure 5 illustrates that the same attack occurring at different times for each voter leads to different mitigations being available, different ballot outcomes, and different paths voters use to cast a ballot. Voter A returns their marked ballot fourteen days prior to election day, the ballot is lost in transit, and four days after the ballot is lost Voter A is notified their ballot was not received. Voter A has two potential recourse options, either request a replacement ballot ($M4$) or cast an alternative in-person absentee ballot prior to election day ($M6$). Voter A chooses to request a replacement ballot. Voter A's replacement ballot is received, marked, and returned via drop box ($M7$) in time to be counted. In this scenario, the delay in the ballot's arrival caused by the attack-mitigation event path occurred early enough in the overall VBM process to allow Voter A's vote to be recovered. Voter B returns their marked ballot seven days prior to election day. The ballot is lost in transit, and Voter B is notified their ballot was not received four days prior to the election. Due to time constraints, only mitigation $M6$ is available for Voter B—filling out a provisional ballot on election day—illustrating that mitigations are not always available at all times. Finally, Voter C returns their marked ballot three days prior to the election. The ballot is lost in transit. However, Voter C is not notified and therefore has no recourse. Voter C's ballot is lost and is not counted. The voting process differs for all three voters. The final row of Figure 5 shows that actual VBM system performance differs across the voters—only Voters A and B have their (unaltered) ballots counted—despite the attack path being the same for all voters, highlighting the importance of the dynamic nature of both the attacks and mitigations on system performance.

7 DISCUSSION

Protecting cyber-physical systems against dynamic threats in resource-constrained systems requires analytical tools that capture the temporal nature of threats and security mitigations as well as the interdependence between mitigations. We examine these issues through an exploratory analysis of VBM processes that addresses temporal aspects of risk and mitigation, resource limitations, and interdependencies between security mitigations. We embed the attack graph paths in a Markov chain that captures the VBM ballot process as well as the linkages to the attacks and mitigations to show how these linkages affect the overall ballot process and voting outcomes. Through a case study based on VBM election processes we provide insight into how to manage dynamic risks in resource-constrained infrastructure systems. In particular, the analysis sheds light on how the timing of attacks and mitigations impact system performance, how mitigations can affect change the overall process, and how mitigations are interdependent and will lack availability at times. This paper demonstrates that attack graphs alone cannot model the dynamic nature of cyber-physical process attacks and mitigations and motivates the need to model attack graphs in the context of process models that consider overall system performance.

There are several limitations with the DTMC model introduced in this paper that motivate future research. First, the base DTMC model (see Figure 3) does not reflect realistic transition times observed in voting systems. This motivates the need to consider semi-Markov processes or Markov states that are time-indexed to reflect actual transition times. Additionally, historical data should be used to tune the model. Second, the transition probability matrices $P_t,\ t = 0, 1, ..., T$ in this paper do not directly consider the resources available to carry out certain election tasks or requests. For example, if an election office receives an excessive number of absentee ballot requests, this results in congestion of the system and ballots may not be immediately sent to voters. Work is in progress to address these limitations.

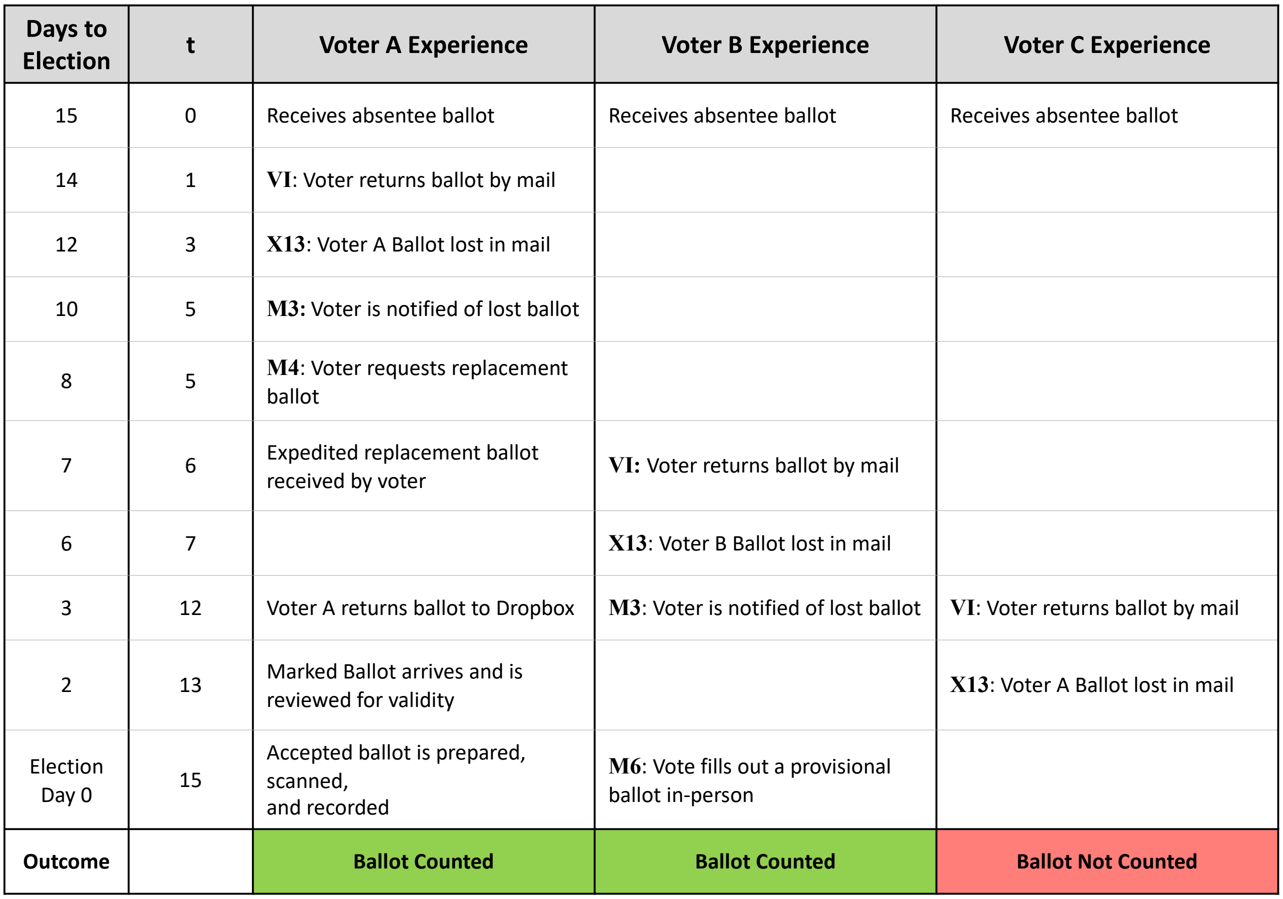
ACKNOWLEDGMENTS

This work was in part funded by the National Science Foundation Awards 1912166 and 2000986. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the National Science Foundation.

REFERENCES

[1] EAC Advisory Board and Standards Board. 2009. *Elections Operations Assessment: Threat Trees and Matrices and Threat Instance Risk Analyzer (TIRA).* Technical Report. University of South Alabama.

[2] C. Cheh, K. Keefe, B. Feddersen, B. Chen, W. G. Temple, and W. H. Sanders. 2017. Developing Models for Physical Attacks in Cyber-Physical Systems. *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy (CPS '17)*, 49–55.



Days to Election	t	Voter A Experience	Voter B Experience	Voter C Experience
15	0	Receives absentee ballot	Receives absentee ballot	Receives absentee ballot
14	1	**VI**: Voter returns ballot by mail		
12	3	**X13**: Voter A Ballot lost in mail		
10	5	**M3:** Voter is notified of lost ballot		
8	5	**M4**: Voter requests replacement ballot		
7	6	Expedited replacement ballot received by voter	**VI:** Voter returns ballot by mail	
6	7		**X13**: Voter B Ballot lost in mail	
3	12	Voter A returns ballot to Dropbox	**M3**: Voter is notified of lost ballot	**VI**: Voter returns ballot by mail
2	13	Marked Ballot arrives and is reviewed for validity		**X13**: Voter A Ballot lost in mail
Election Day 0	15	Accepted ballot is prepared, scanned, and recorded	**M6**: Vote fills out a provisional ballot in-person	
Outcome		🟩 **Ballot Counted**	🟩 **Ballot Counted**	🟥 **Ballot Not Counted**

Figure 5: Analysis of an attack scenario for three voters

[3] Cybersecurity and Infrastructure Security Agency. 2020. Mail-in Voting in 2020 Infrastructure Risk Assessment and Infographic.

[4] F. Enayaty-Ahangar, L.A. Albert, and E. DuBois. 2020. A survey of optimization models and methods for cyberinfrastructure security. *IISE Transactions* 53, 2 (2020), 182–198.

[5] MIT Election Data + Science Lab. 2021. Voting by mail and absentee voting. https://electionlab.mit.edu/research/voting-mail-and-absentee-voting

[6] 'Office of Inspector General'. 2021. *United States Postal Service Performance of Election and Political Mail During the November 2020 General Election.* Audit Report.

[7] N.M. Scala, I. Bloomquist, Y. Mezgebe, and B. Jilcha. 2021. A Process Map and Risk Assessment for Mail-based Voting. In *Proceedings of the 2021 Institute of Industrial and System Engineers (IISE) Annual Conference*, A. Ghate, K. Krishnaiyer, K. Paynabar, eds. (Ed.). Towson, MD.

[8] N.M. Scala, P. Goethals, J. Dehlinger, Y. Mezgebe, B. Jilcha, and I. Bloomquist. 2022. Evaluating mail-based security for electoral processes using attack trees. *Risk Analysis* (01 2022). https://doi.org/10.1111/risa.13876

[9] B. Schneier. 1999. Attack trees: Modeling security threats. *Dr. Dobb's Journal* (December 1999).

[10] B. I Simidchieva, S. J Engle, M. Clifford, A. C. Jones, S. Peisert, and M. Bishop. 2010. Modeling and Analyzing Faults to Improve Election Process Robustness. In *In Proceedings of the 2010 USENIX/ACCURATE Electronic Voting Technology Workshop.* https://escholarship.org/uc/item/0cg3b5vb#article_main

[11] Wisconsin Elections Commission. 2020. Absentee Ballot Report. https://elections.wi.gov/

[12] Pardue, J. Yasinisac, A. 2010. A Process for Assessing Voting System Risk Using Threat Trees. *Journal of Information Systems Applied Research* (2010).

[13] K. Zheng and L.A. Albert. 2019. A budgeted maximum multiple coverage model for cybersecurity planning and management. *Naval Research Logistics* 66, 5 (2019), 441–429.

[14] K. Zheng, L.A. Albert, J.R. Luedtke, and E. Towle. 2019. A budgeted maximum multiple coverage model for cybersecurity planning and management. *IISE Transactions* 51, 12 (2019), 1303–1317.