The Economics of Hardware Security

Thesis proposal

Adam Hastings
Department of Computer Science
Columbia University
hastings@cs.columbia.edu

December 11, 2023

Abstract

Computer security has long been a pressing issue in society, yet most attacks today are due to weaknesses that are entirely avoidable. This is particularly true in hardware and architecture security, where many problems have been "solved" by researchers although these solutions are rarely built in real-world systems. An economic analysis of the problem reveals that the lack of security adoption is due to fact that security is a cost, and the four players in the game of security—namely, users, vendors, authorities, and attackers—rationally try to avoid this cost. In this thesis proposal, I present three of my research works that aim to improve hardware security by better understanding and managing the cost of hardware security. The first work measures one aspect of security's cost, namely the cost that security imposes on device performance. The second work provides a mechanism that can adjust the incentives for users and vendors to create a virtuous cycle of security improvement. The last work models another potential actuator of security improvement—cyberinsurance. These works will each contribute a thesis that lays the groundwork for interdisciplinary research at the intersection of computer architecture, security, usability, economics, and policy.

Contents

1 Introduction

Computer security has in recent years been a topic of great concern: Ransomware attacks cripple vital services like hospitals, foreign adversaries steal national secrets, individual privacy gets compromised, and society overall suffers billions of dollars annually [7]. Yet all this occurs despite decades of research on computer security and a multi-billion-dollar cybersecurity industry. So why does computer insecurity continue to exist?

In this proposal, I first present my research work that first aims to answer why certain types of insecurity persists, particularly insecurity in computer hardware and architecture. At the core of the problem is that in computer hardware and architecture, security nearly always comes at a cost, yet the question of *who* should have to pay this cost, and *where* it should be paid, remain unanswered. Seen through this lens, it becomes clear that the challenge of addressing hardware insecurity entails much more than the hardware and architecture communities can develop on their own and requires tools from other disciplines, including economics, policy, usability, and human computer interaction (HCI).

Having established the problem, my research then aims to address the problem through a variety of methods. My first work aims to understand and *quantify* one aspect of the cost of security, namely the cost that security imposes on end users in the form of the opportunity cost of computer performance. In this work, we design and conduct the first known experiments to quantify how much users value performance, in terms of US dollars. We use several methods to answer this question and find, for example, that the average user values a 10% performance loss at around two dollars per day. The significance of this work is that it provides a rigorous quantitative foundation by which researchers and policymakers can understand the cost of security.

My second work aims to address the costs of security head-on in a combined technical-regulatory mechanism called FAIRSHARE. This work explores how policymakers can take advantage of the resources provided by computer architects to make entirely novel approaches towards security regulation. The premise is simple: To achieve security, everyone needs to pay their fair share of the cost; FAIRSHARE is a solution that allows for an entirely new cost-based approach to security regulation, including the technical mechanisms needed to make this a viable path for regulators. Specifically, it promotes the idea that one form of the cost of security—performance loss—can be measured and accounted for just like other costs of security (e.g. capital expenditures or research funding). FAIRSHARE is a combined system of regulation and technical solutions that make it possible to mandate that not only organizations spend a sufficient amount of resources on security but that *devices themselves* spend a sufficient amount of *their* resources—like energy or CPU execution cycles—on security as well, essentially balancing the cost of security between vendor, user, and regulator. The work includes the technical mechanism needed to make this happen, namely the device-level accounting needed to ensure that systems themselves are putting forth a fair share of effort towards security. In doing so, I argue that FAIRSHARE adjusts the incentives for users and vendors to create a virtuous cycle of continued security improvements, thus fixing the incentive conditions that hinder security today.

Finally, in my third work, I examine a different kind of mechanism by which security can be improved: Cyberinsurance. In this work I develop a model of how cyberinsurance affects rational decision-making among insurees. Although still in progress, this work provides a foundation by which I will answer important questions about security investment and whether it is better to invest in security or simply buy insurance.

These three works will form core of my thesis. It is my intention and aim to convince my committee members that my works make a significant contribution towards advancing my thesis, which is this: *To make meaningful improvements to hardware security, computer architects need to engage in the interdisciplinary elements of security like usability, economics, and policy. These fields can be unified by framing security as a cost.* My work is the first work in the new interdisciplinary area.

The organization thesis proposal is as follows: In Section 2, I present the foundation for this work, establishing background where needed, and advance a framework for understanding (hardware) security as a cost. I present my first work, on measuring the value of performance (i.e. the cost of security), in Section 3. My second work, on designing a mechanism to kickstart architecture-level security adoption, is presented in Section 4. Section 5 presents my ongoing work on cyberinsurance. In Section 6, I lay out my plan for completing this work.

2 Background

My work is rooted in computer architecture and security but draws in concepts from other domains. This section provides a theoretical foundation for my work and brings in additional context when needed.

2.1 Security as a Cost

As stated earlier, a unifying theme in my work is identifying and dealing with the costs that security imposes. This section defines what I mean by "cost" and how these costs can be "paid" by developing a novel framework for interpreting security via four players in game of security, namely user, vendors, authorities, and attackers.

These four players each bear the cost of security in their own way. The interactions between these four groups is illustrated in Figure 1. Although a simplification of the full security landscape, it is a useful one, and this four-player abstraction of security will be used throughout my work. The definition of the "cost" of security is different for each player:

Users are those who use technology products (hence our definition of "user" is very broad, ranging from individual smartphone users to large organizations). Users can pay for security in many different ways: They can pay for security products like anti-virus software or network monitoring tools to detect or deter attackers from compromising their systems, or can pay agencies and consultants to help enhance their security posture. The cost of security is not just in terms of dollars though: When using security products, users "pay" in terms of the tradeoffs to usability, e.g. in terms of the time and amount of annoyance involved in two-factor authentication, or in the decreased device performance and responsiveness caused by system-level security features (like memory safety defenses). Users also pay for security in terms of taxes paid to government authorities (who may catch and punish attackers). Finally, if security fails, users pay by absorbing the costs of attack. This could entail paying ransom to ransomware groups, or paying the costs associated with having their systems breached (including any costs due to e.g. loss of privacy), or the costs of repairing or replacing compromised systems, including the cost of downtime (particularly for businesses and other organizations) during this process.

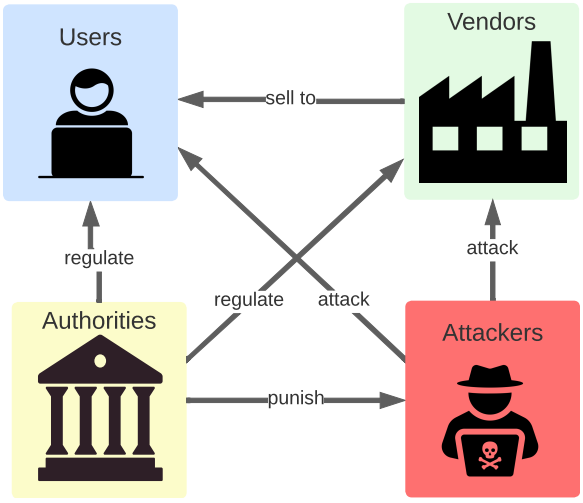


Figure 1: The four players identified in the Doctrine. All four players must bear some of the burden of security.

Vendors are those who create and sell technology products. This encompasses software vendors, hardware vendors, and service providers. Vendors too can "pay" for security in multiple ways: During products' design and development, there is the cost of ensuring that systems follow basic guidelines and best practices to avoid known security failures (e.g. as enumerated in the MITRE Common Weakness Enumeration lists). This cost can come in terms of the employee time (and wages) needed to hold e.g. code review meetings, or the time and effort needed to hire security consultants to review product designs. Importantly though, this cost (an opportunity cost, really) has further downstream costs: Each minute spent on security can delay time to market (a hugely important factor in technology) and each dollar spent on security is diverted away from other potentially revenue-generated activities (like advertising or the research and development on other features desired by users). Vendors can also pay for security via bug bounties to security researchers who find vulnerabilities. Vendors who do not sufficiently pay for security may pay the cost in terms of loss of reputation and loss of market share.

Authorities are those who have a degree of authority over the Vendors and the Users. This too is fairly wide in scope, and includes governments, regulatory agencies (including self-regulatory agencies, or SROs), and law enforcement, among others. Authorities can pass and enforce regulations in situations where market forces alone do not produce efficient security outcomes (more on this in the following sections). Authorities pay for the cost of security in terms of the money and effort needed to write regulations but also the money and effort needed to enforce regulations, including the ability to reward those who comply with regulations (or punish those who do not). Law enforcement agencies also pay for security in terms of the money and effort needed to catch and punish the attackers, as a form of deterrence.

Attackers are those who attack users for profit or espionage, and are the fourth player in the game of security capable of "paying" the costs of security. This group includes ransomware gangs, "script kiddies", and various nation states' intelligence agencies. At first this may seem semantically backwards, since attackers generally do not want security and certainly are not going to pay for it on behalf of their victims! Instead, we mean that in the landscape of security, attackers too pay a cost, in terms of the resource and efforts needed to breach their victims' systems. Whereas

it is generally desirable to minimize the costs for users, vendors, and authorities, in the case of attackers is it desirable to *maximize* the costs.

Hence security can be viewed as a game of costs, where security outcomes are the result of which parties pay for security, and where they pay for security, and by how much. This is a very useful lens for viewing security interactions in real world systems and processes. To illustrate, consider cryptography, a cornerstone in modern security. The cost of security is borne by the users, who do to encryption and decryption must endure losses to latency and bandwidth. Vendors also pay for some of this cost in terms of the costs required to develop secure implementations of established cryptographic algorithms (which can happen in either software or hardware) as well as the costs associated with participating in public key infrastructure (PKI) systems. Authorities pay a cost too: In an effort to establish an ecosystem of security, standards organizations like NIST sponsor competitions to select cryptographic algorithms and protocols that are widely agreed-upon by experts to be secure. Authorities pay for security by operating certification programs (like FIPS-140) and in certain cases may also require cryptography in certain sensitive domains like healthcare (which then requires the cost of ensuring regulatory compliance). Finally, the attackers pay in terms of the resources required to thwart security. This could be in terms of the resources required to find flaws in the implementation of cryptographic algorithms, or the computational cost of brute force attacks (for e.g. hash cracking), or simply even the cost of circumventing two-factor authentication.

In this example, even though the definition of security's "cost" can vary widely between players, there is still some fundamental quantity that imposed on the players in the efforts towards security, and although these varying costs are not necessarily fungible between one player and the next, it is clear that in many cases, the cost must be paid *somewhere*, and that a failure to pay a cost in one area will necessarily cause greater costs elsewhere (e.g. if the vendor does not pay the costs needed to build a secure system, the user will end up paying the cost in one way or another).

2.2 The Economic Factors Affecting Security Adoption

Much of insecurity, particularly insecurity in computer hardware and architecture, can be seen as a failure of users, vendors, and authorities to properly balance the costs of security between themselves. This occurs even though collective computer security is in the best interest of all three parties. We now borrow from the field of economics to explain.

Prior work finds that information security suffers from many cases of market failures, or situations where the markets fail to produce efficient outcomes [6, 8]. In one of my papers, we find that we can explain much of hardware insecurity also in terms of market failures, particularly using the above framing of security as a cost [25]. I outline a few examples below:

2.2.1 Moral Hazards

A moral hazard is a situation where the consequences of one's behavior are borne by someone else, which can incentivize irresponsible and reckless behavior [31]. In computer systems, this dynamic can be seen between vendor and user: If a vendor's negligence leads to security vulnerabilities in a product, the consequences of this negligence are paid by the user (who is now vulnerable to attack) and not the vendor. Since vendors may not suffer the consequences of their mistakes, they may

not be sufficiently motivated to fix them (or make the required investments to avoid them in the future).

2.2.2 Information Asymmetry

In a functioning market, the moral hazards would not be an issue since users would simply not purchase insecure products, which would incentivize vendors to make their products secure in the first place. However, this requires that users and vendors are operating under a shared understanding of the security and quality of goods in the market. In reality, this is not the case: Studies show that users are generally very uninformed about the basics of computer security and are not capable of judging good security from bad [43, 35, 16, 28, 19, 20, 32, 38, 3, 46, 36, 42, 24, 33]. This creates an information asymmetry between vendor and user, which is a well-known cause of market failure.

A famous example of this is the "market for lemons", which demonstrates that in a market where consumers cannot distinguish between high-quality and low-quality goods, they will not be willing to pay premiums for the high-quality goods, and the sellers of these goods will leave the market; the effect is that the presence of information asymmetry causes the quality of goods in the market to decline [5]. The same can happen in security: When consumers cannot tell evaluate the security of technology products, they will not be willing to pay extra for more secure products and hence the market exerts a downward force on overall product security.

2.2.3 Prisoners Dilemma

In hardware security, the presence of information asymmetry creates a variant of the prisoner's dilemma, a well-known result from game theory that demonstrates how certain arrangements of incentives can cause agents to make rational decisions that create undesirable outcomes. We find examples of this in hardware security.

Consider the problem of memory safety, which for decades has consistently been one of the leading vectors of exploit in systems []. Many memory safety issues can be solved via security defenses at the systems or architecture level. Yet these defenses generally come at a cost, typically to runtime performance or die area (which effectively translates back into decreased runtime performance). Hence if a vendor were to implement such a defense, their product would less performant compared to their competitors. Unlike security, performance can be quantified and easily communicated to users via various benchmarking methods; even without quantitative benchmark scores, decreased performance can cause delays to responsiveness which makes products "feel" slower, which is also noticeable to users. Vendors are then faced with a choice: If they choose to exchange performance for increased security, due to information asymmetry the user may not see the value in the tradeoff but will be able to see the negative effects, i.e. the decreased runtime performance. Consumers will not pay a premium for a feature they do not understand or cannot evaluate, and may choose a competitors' products which are insecure but faster. Hence vendors may collectively choose the "defect" strategy since they risk losing market share for taking on the burden of security.

2.2.4 Bystander Effects

Despite the above, vendors generally do care about the security of their products and will do what they can to improve their products' security. Programs like MITRE's CVE list also incentivize vendors to patch issues once they are known. But in hardware and architecture, problems often span multiple levels of the computing stack and many times it is not straightforward to determine who is responsible for fixing a given issue. One such example is Rowhammer, an attack on DRAM integrity [29]. Although the issue clearly arises from the instability of individual DRAM cells, the ability to defend against Rowhammer-style attacks can occur in numerous places, including in operating system, in the memory controller, or in the DRAM chips themselves. Given the general lack of vertical integration in the technology market, this means that the various system components that could implement Rowhammer defenses are each designed by different companies, and so each company may rationally try to "pass the buck" (i.e. the responsibility of fixing Rowhammer) onto another company somewhere else in the computing stack. Memory safety is another example: It can be addressed at virtually all levels of the computing stack, which is perhaps one of the reasons why memory safety issues have persisted for so long.

2.3 A Doctrine for Hardware Security

So far, I have focused on explaining the problem my thesis work aims to address, namely the economic forces that hinder hardware and architecture security. The remaining chapters are on proposing *solutions* for how to deal with the problem.

In one of my papers, we propose a doctrine for researchers and policymakers to follow when working in this interdisciplinary research area, which we call the Doctrine of Shared Burdens. In this paper, we first review previously proposed doctrines of security [], which are very briefly summarized as follows:

1. *Doctrine of Prevention:* Security should be achieved through the elimination of vulnerabilities. Using our above framing, this places the burden of security on the vendor. In practice this doctrine seems to be an unachievable (finding all bugs is computationally infeasible), and as mentioned above, vendors often do not have a sufficient economic incentive to find and eliminate all bugs.

2. *Doctrine of Risk Management:* Security should be achieved by acknowledging that attacks are inevitable, and efforts should focus on making investments that will offer the most protection for the least cost. This places the burden of security on the users. The problem with this doctrine is that it absolves vendors of responsibility and is also impractical to achieve, since there is a general lack of a quantitative data on which to build useful models and so this approach results in best-effort but ad hoc security investment.

3. *Doctrine of Deterrence:* Security should be achieved by punishing the attackers. This doctrine comes at a cost to the attacker (which is good) but, like the other doctrines, is largely unachievable in practice: Attackers could be based in unfriendly nations with no possibility for extradition, making punishment (and hence deterrence) improbable if impossible. Furthermore, a skilled attacker can use a variety of methods cover their tracks, making it even harder to assign blame and mete out punishments.

4. *Doctrine of Cybersecurity as a Public Good:* Security should be achieved by administered by the state in the interest of individuals' rights and public welfare. This doctrine tasks the authorities with the responsibility of security. The problems with this approach is that it does not hold attackers accountable and is hindered by authorities' ability to effectively manage and govern security, especially in hardware security, where the sheer complexity of systems makes it extremely difficult for non-vendors to determine how and where security should be administered.

The above doctrines each put the cost of security onto only the vendors, users, attackers, or authorities, respectively. Our addition to these framings is the Doctrine of Shared Burdens, which states that hardware security problems need to consider the effect of the costs of security on all four players (with an asymmetric burden placed on attackers). The remainder of this thesis proposal is work that I have done with the vision of implementing this doctrine.

3 Proposal Topic I: Measuring the Value of Performance

The first work in my thesis will be on my paper, *How Much is Performance Worth to Users?* This work is on understanding and in particular *quantifying* one of the ways hardware and architecture security is a cost. As explained in the background section, one of the causes for insecurity is the fact that security comes at a cost to performance, and product vendors have rational reasons for wanting to avoid exchaning performance for security. Therefore, in any effort to fairly distribute the burden of security between users, vendors, and authorities, it is of central importance to be able to at least define and measure what these burdens are. This work aims to capture how much users value performance, which allows us to quantify the opportunity cost of exchanging security for performance.

3.1 Background

Although motivated by a desire to quantify the security-performnace tradeoff, this work aims to answer a very general question—the value of performance—which, surprisingly, was previously unanswered. This means this work has applications and value that reaches far beyond security alone.

The benefit of this work is that it allows computer architects to quanify what previously could only be evaluated in qualitative terms. For decades, engineers have taken a quantitative approach to computer architecture and systems, which has been a key factor in achieving meaningful improvements year after year. Computer architects and designers have been tremendously successful at developing metrics to measure important quantities such as performance, power consumption, die area, and reliability, which have allowed systems designers to have a clear conversation about pros and cons of competing approaches. More recently, architects and designers are increasingly tasked with designing systems for user-facing requirements like responsiveness, security, and privacy, which can be broadly classified as a quality requirements [45, 4, 27, 23]. However, unlike traditional metrics, these user-facing requirements are often not easily measurable, making it difficult for architects to quantitatively determine which tradeoffs are worth making.

For example, should a phone designer add biometric authentication if it comes at the expense of storage space? How much device responsiveness should be exchanged for an always-on security

feature? These types of questions are typically unanswerable using traditional design metrics (like power or performance) because traditional design metrics are agnostic to how much various design features are worth to users. Yet deciding these tradeoffs is a necessary and consequential step of the design process. How can systems designers and computer architects more rigorously balance design requirements like power, performance, and die area against indeterminate requirements like user preferences?

In this work, we make the first known attempt to introduce user preferences as a quantifiable metric for design decision-making. Specifically, we aim to put a price to users' value of performance. In other words, this work finds the "exchange rate" between performance and user value, in terms of US$. By establishing this "exchange rate", we provide systems designers and architects with a quantitative metric by which they can balance tradeoffs between performance and other features which may "cost" performance (such as security or usability features, among others). To illustrate, suppose a systems architect must decide whether or not to include an image processing accelerator in a system that, if included, would come at the expense of cache sizes and decrease general system performance by 10%. Is this worth the cost? Via our methodology, an architect can put a monetary amount on the opportunity cost (in terms of user satisfaction) of such a feature, and can compare this to users' value of the feature itself (perhaps derived via market research studies). Our work makes this type of quantified decision-making possible.

3.2 Methods

We employ three methods to measure the value of performance. In all experiments, participants were recruited from Amazon's Mechanical Turk platform and were required to be at least 18 years old and working in the United States. All experimental protocols were reviewed and approved by our IRB.

3.2.1 Why Measure the Cost of Performance *Losses*?

In all our methods, we choose to measure security in terms of the users' willingness to accept *losses* to performance. This is not an obvious metric for valuating security, but is the most appropriate metric because it allows us to measure value in term of willingness to accept ("WTA"), a standard metric in economics for determining the minimum amount that a person would have to be paid to accept some unfavorable condition or outcome. Although perhaps not the most obvious way to study this problem, we find it to be the most appropriate value to measure given the constraints: First and foremost, to get an accurate measurement of user preferences, we felt it was necessary to study users in their own environment and on their own computing devices. This ruled out in-the-lab style experimentation[1]. Given this constraint, the only direction by which we can reliably adjust users' devices' performance is downward (after all, if there was an easy way to permanently increase device performance, many users would have done so already!).

[1]The justification here is that testing participants in a tightly-controlled laboratory environment would require experimentation on different devices (with different performance specs) than what the participant might be used to; hence it would be inappropriate to measure participants' value of performance against such a contrived setting. For example, questions of "how much would you have to be paid to make this device 10% slower" are meaningless unless participants already have a baseline understanding of how fast or responsive a device is in the first place. Hence we chose to study users only in their own environment and on their own personal devices.

Another reason for measuring users' willingness to accept performance losses is that it allows for systems and architectural features to be evaluated and compared by their opportunity cost in terms of both performance *and* user preference. To illustrate this point, consider a systems architect who wants to include hardware support for secure memory bounds checking. At the systems and architectural level, the opportunity cost of such a feature is the performance gain that could otherwise be achieved without the bounds checking, while at the end user level, the opportunity cost is the additional features that this extra performance could allow. By quantifying the relationship between performance and end user value, we provide systems designers and architects a means by which they can translate from the low-level domain of systems and devices to the high-level domain of user value.

Finally, by measuring the cost of performance losses, we also make it possible measure the in-the-field cost of security patches to hardware vulnerabilities like Meltdown and Spectre (see Section 3.4 for more details).

3.2.2 How Do We Throttle Participants' Devices?

Given the above constraints, we needed a method to remotely and reliably throttle the performance of study participants' personal devices. We found that the best way to do this was by throttling CPU frequency. On Windows devices, this is achieved via the `powercfg` command line utility by adjusting the value of `PROCTHROTTLEMAX`, which limits the systems' CPU frequency relative to its maximum (i.e. setting `PROCTHROTTLEMAX` to `50` on a 4.0 GHz device should lower CPU frequency to 2.0 GHz). We limited experimentation to participants using Windows 10 devices[2]. We run benchmarks tests (SPECint, SPECrate, and WebXPRT3 to confirm that throttling device CPU frequency is an adequate method of simulating lowered performance.)

3.2.3 Method #1: Longitudinal Study

The first methodology was the **INCENTIVE COMPATIBLE STUDY**. This study was designed to overcome the issues with survey-based approaches, namely the observation that any attempt to simply ask users outright how much they value device performance is likely to yield dubious results because performance is an ill-defined concept for typical users. Further, even if users could ascertain how much they value performance, it is difficult to accurately elicit this value from paid study participants who are motivated to answer questions as fast as possible (especially participants from crowdsourcing platforms like Mechanical Turk) and who may give little or no thought to their answers.

The INCENTIVE COMPATIBLE STUDY was designed to overcome both of these threats to validity. Briefly, the experimental protocol gives study participants an offer with real-world consequences: We give participants the option of slowing down their personal computer by a fixed percentage for some set amount of money for at least 7 days in a row. Participants who accept their offer are able to experience firsthand, over a long timespan, the consequences of a quantitative

[2]Our reasons for this were that 1) Windows 10 comprises a majority of desktop and laptop users, and 2) we could not find a reliable method for throttling CPU frequency on MacOS devices. On Linux devices, throttling CPU frequency is easy but the share of desktop and laptop Linux users remains small. We also considered experimenting on mobile phones, but were not able to throttle device performance without asking study participants to "jailbreak" their devices; we figured that potential participants would be unwilling to do so.

performance loss, allowing them to develop an opinion on how irksome it was to accept a performance loss of, say, 30%. Second, the protocol fixes the problem the careless crowdsource worker: Participants who accept their given offer are forced to deal with the consequences of their choices in terms of performance loss; this removes the incentive to give thoughtless responses and instead encourages participants to only accept the agreed-upon performance loss if exchange is worth the amount of money offered. In terms of study design, this mechanism makes our protocol "incentive compatible", which is the gold standard in user study designs.

Each participant was given a single take-it-or-leave-it offer price. We varied both the offer price (between \$0/day and \$10/day) and the slowdown percentage (either 10%, 20%, or 30%) across multiple participants, with each participant receiving a single unique combination of offer price and slowdown percentage. Via logistic regression, we then found the "threshold" dollar amount at which it becomes more likely than not that a participant will accept a given performance loss for the entire duration. In terms of economics, this threshold point is called *willingness to accept* (WTA), which is a standard metric for quantifying individuals' preferences and loosely is defined as the minimum an individual would have to be paid to accept some unfavorable condition (e.g. throttled device performance).

3.2.4 Method #2: Short-term, Task based Study

Our second methodology was the **TASK-BASED STUDY**. This methodology was created to complement the INCENTIVE COMPATIBLE STUDY, which, due to the desired incentive compatibility, required us to accept some undesirable tradeoffs. Namely, the INCENTIVE COMPATIBLE STUDY yields WTA in terms of a *day-by-day* exchange rate, but it is also useful and important to find a *permanent*, lifetime exchange rate, i.e. how much users would have to be paid in a single lump sum to accept a pre-determined permanent performance loss on their computer. This is both a more intuitive approach towards performance loss WTA as well as a closer match to how performance losses are actually experienced by users in the real world (e.g. due to microcode patches to architectural security vulnerabilities. But if we wanted to extend the INCENTIVE COMPATIBLE STUDY to find the lump sum WTA, it would potentially require us to pay participants every day for the entire duration of their devices' lifetimes! Hence the first study had to be completely re-designed, leading to the TASK-BASED STUDY.

The protocol is briefly summarized here: Participants are directed to complete a series of tasks on their personal device while experiencing randomly chosen, and alternating periods of throttled and un-throttled performance. The tasks were aimed at mimicking typical device usage and included subtasks such as using a web browser, playing video content, and using a word processor program. The order and count of each set of subtasks was carefully considered to give participants the best chance of adequately comparing throttled versus unthrottled device performance. Afterwards, participants were asked via an exponential search mechanism to estimate the minimum amount of money they would have to be paid to permanently accept the experienced performance loss on their device. As with the INCENTIVE COMPATIBLE STUDY, the protocol takes considerable measures to ensure as best as possible the scientific validity of the findings (see paper for details).

The TASK-BASED STUDY is not without compromise, though: We necessarily forgo incentive compatibility in the TASK-BASED STUDY to make the experiment conclude in a reasonable amount of time. Additionally, participants only interact with throttled performance in a limited

timeframe, and our effort to model "typical" usage via our selected subtasks may be subject to selection biases (similar to how SPEC benchmarks may induce selection biases). Nevertheless, it is a best-effort attempt to find a complementary measure of WTA that is unobtainable using the INCENTIVE COMPATIBLE STUDY.

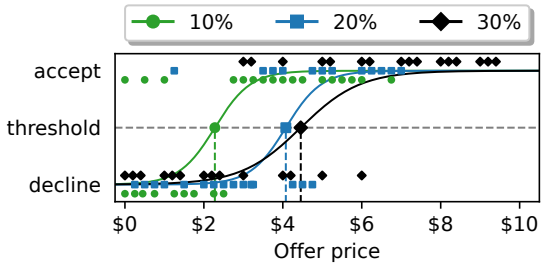
3.2.5 Method #3: Survey based study

The prior two studies required participants to place a great amount of trust in the researchers as it asks participants to download a program from the Internet and run it with Administrator privileges—a security risk many were unwilling to take.

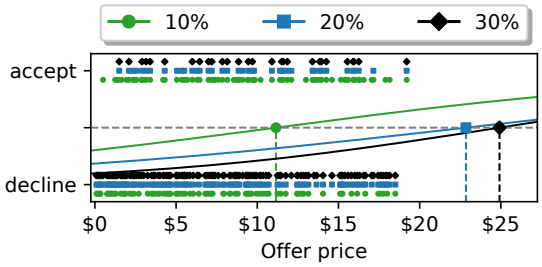
To achieve a larger sample size, we employed a third and final methodology, the **LARGE-SCALE ONLINE SURVEY**. While we understood the drawbacks of surveys, we wanted to verify our intuition with real experimental data on surveys. The survey asked participants the same questions as the the INCENTIVE COMPATIBLE STUDY and TASK-BASED STUDY but with no hands-on interaction with throttled devices and with no incentive compatibility. This yielded substantially different results: Study participants claimed they would accept performance losses of 10%, 20%, and 30% for $11.15, $22.85, and $24.92 per day or $499, $1,214, and $3,723 overall, respectively. Across the board, the survey results are far higher than the corresponding study results, suggesting that the hands-on experience in the study was essential towards properly calibrating participants to answer questions regarding their valuation of performance and performance losses.

3.3 Results

3.3.1 Method #1: INCENTIVE COMPATIBLE STUDY Results



(a) Incentive compatible study results. At offer prices of $2.27 per day, $4.07 per day, and $4.44 per day, it becomes more likely than not that a participant will accept a 10%, 20%, and 30% performance loss, respectively.



(b) Non-incentive compatible online survey results. At offer prices of $11.15 per day, $22.85 per day, and $24.92 per day, it becomes more likely than not that a participant will accept a 10%, 20%, and 30% performance loss, respectively.

Figure 2: Results from the INCENTIVE COMPATIBLE STUDY, which finds participants' willingness to accept per day performance losses in an incentive compatible manner. Each marker (i.e. green circle, blue square, and black diamond) represents a unique response to our offer to throttle device performance by $N \in \{10\%, 20\%, 30\%\}$ in exchange for $X per day.

The INCENTIVE COMPATIBLE STUDY was conducted for slowdowns of 10%, 20%, and 30%, for both the incentive compatible study (N=21, N=24, and N=22, respectively) and the online

survey (N=306). In both cases, we split participants into two categories: the "accept" group, who accept the throttled performance for the duration of their participation, and the "decline " group, who do not. The "decline" group includes participants who immediately reject the offer, as well and participants who may initially accept their offer only to later reject it before the participation period ends. The "accept" group values the amount performance lost *less* than the amount offered to them (since they accepted the offer), while the "decline" group values the performance loss *more* than the amount offered to them. Results are plotted in Figure 2.

To summarize the data and find the per day WTA, we use logistic regression to find the dollar amount at which it becomes more likely than not that a participant will accept their given offer. To do this, we first model the "decline" group as a 0 and the "accept" group as a 1. We then use the STAN modeling framework to fit a logistic curve to the data which models the probability of a participant accepting an offer given the offer price $\$x$. To summarize the data, we find the offer price x at which

$$p(\text{outcome} = \text{“accept”}|\text{offer} = \$x) \geq 0.5$$

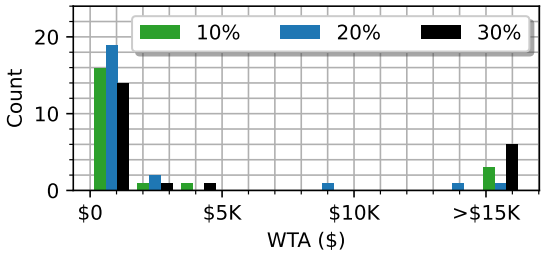
i.e. the point at which it is more likely than not that a participant will accept a given offer price, which we call the threshold value. From the incentive compatible study, we find the threshold values to be \$2.27 per day, \$4.07 per day, and \$4.44 per day for slowdowns of 10%, 20%, and 30%, respectively. At the 95% confidence level, the threshold values are between \$1.54 and \$3.02 for a 10% slowdown, between \$3.39 and \$4.74 for a 20% slowdown, and between \$3.40 and \$5.50 for a 30% slowdown. From the online survey, we find the threshold values to be \$11.15 per day, \$22.85 per day, and \$24.92 per day, respectively. At the 95% confidence level, the threshold values are between \$7.66 and \$15.38 for a 10% slowdown, between \$14.69 and \$41.40 for a 20% slowdown, and between \$17.31 and \$39.70 for a 30% slowdown. These curves are plotted in Figure 2.

From the data, several trends emerge. First, we find that as the offer price increases, so does the likelihood of a participant accepting their offer. This is an intuitive and expected result.

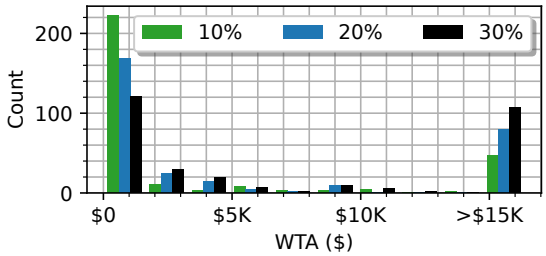
Second, in the experience-based study—where participants interact with throttled device performance for several days at a time—we find that collected WTAs are once again lower than the results obtained via the online survey. Like with the TASKS-BASED STUDY, a likely explanation is that the hands-on experience with a throttled device calibrates participants to accurately price the cost of performance losses, whereas survey participants are given no such calibration.

Third, we observe that the logistic curves are not as evenly spaced as perhaps expected: In both the study and survey data, the threshold values for the 20% performance loss are much closer to the threshold values for 30% performance losses than for 10% performance losses. While perhaps due to sampling issues, the explanation may also be that users view the harms of performance losses non-linearly. However, we also point out that this trend was not observed in Experiment #1. More experimentation is needed to definitively explain this observation.

Finally, we observe that the survey data is far noisier than the study data: In Figure 2a, the fitted logistic curves exhibit a fairly sharp rise from "decline" to "accept" and fairly tight confidence bounds (a couple dollars or less), whereas the survey data in Figure 2b exhibits a very slow rise and confidences bounds of tens of dollars or more. This is in spite of the survey having roughly ten times the amount of data! Once again, the likely explanation is that the lack of hands-on experience and incentive compatibility comes at a cost to precision in addition to accuracy.



(a) Simulation study results. The median WTA for a permanent device performance loss of 10%, 20%, and 30% is $381, $457, and $853, respectively.



(b) Online survey results. The median WTA for a permanent device performance loss of 10%, 20%, and 30% is $499, $1,214, and $3,723, respectively.

Figure 3: Results from the TASKS-BASED STUDY. We plot histograms of the data collected using the exponential yes/no elicitation mechanism for both the hands-on simulation study and the online survey. All distributions exhibit a sharp exponential decay followed by a bump at the ¿$15K mark. Although this may give the data the appearance of a bathtub curve, this is simply an artifact of limiting responses to be no more that $16,384 (for reasons discussed in Section 3.2.4).

3.3.2 Method #2: TASKS-BASED STUDY Results

The TASKS-BASED STUDY was conducted for slowdowns of 10%, 20%, and 30%, for both the incentive compatible study (N=26, N=29, and N=30, respectively) and the online survey (N=306). Results from both experiments are plotted as histograms in Figure 3.

In all cases, we find that WTA approximates a power law distribution. That is, the majority of responses are grouped towards the lefthand side of the histogram (indicating a WTA of hundreds of dollars or less) while remaining responses follow a long-tailed decaying distribution. At the very end of the tail is a "bump" at the $16,384 mark, which was the maximum response possible. To prevent this bump from influencing summary statistics, we choose to characterize the distributions by their median value.

From our simulation study, we find that median WTA for a performance loss of 10%, 20%, or 30% is $381, $457, and $853, respectively. From the online survey, we find that median WTA for a performance loss of 10%, 20%, and 30% is $499, $1,214, and $3,723, respectively.

We observe two trends in the data: First, we observe that, in both the study and the survey, the median WTA increases with the degree of slowdown. That is, participants would require to be paid more money in order to accept increasingly larger losses to device performance. This is an intuitive and expected result.

Second, we observe that the median WTAs collected from the study were across the board lower than the median WTAs collected from the online-only survey. Although the results are close for the 10% slowdown ($381 versus $499), the results become greatly disparate by the 30% slowdown ($853 versus $3,723). Why might this be? Our hypothesis is that in the hands-on study, where participants actually experienced throttled performance on their personal device, participants were better calibrated to put a price on how much the loss in performance is actually worth. Indeed, the very reason we chose to conduct the study in the first place was because we doubted that online survey participants would be able to accurately put a price to the cost of performance losses without experiencing performance losses firsthand. For example, is a 30% loss to perfor-

mance a mild annoyance, or a device death sentence? Only the study participants would be able to reasonably answer such questions. Thus a plausible explanation for the gap in the study and survey WTAs is that, after experiencing throttling firsthand, the study participants realized that the performance throttling was not as bad as they may have expected it to be.

3.4 Discussion and Applications

I now discuss some of the implications and applications of this work.

Pricing Patches and Rebates: In the many industries (such as automotive and food, among others) there are well-established precedents, norms, and regulations for issuing product recalls and rebates when products are found to be defective or unsafe. At present, very few such precedents, norms, or regulations exist in the domain of computer systems. One recent exception was the so-called "Batterygate" affair, where Apple was found to have throttled older iPhones (allegedly to incite users into buying new phones) [37] and, after a class action lawsuit, agreed to pay consumers roughly $25 per affected device [1]. Court documents show that this settlement was reached by supposedly analyzing the resale market; however, to our knowledge, the settlement did not consider or study the impact of performance losses on actual users. In fact, our results suggest that $25 worth far less than even minor performance losses[3]. By experimentally measuring user's value of performance, we open the door to more equitably determining such rebates.

Quantifying the cost of patches becomes especially important given the rise of hardware vulnerabilities like Spectre [30] and Meltdown [34], which when patched cause significant losses to performance. It is not unreasonable to consider such critical security flaws (especially Meltdown) to be product defects, and if the trends continue, consumers may demand to have performance losses recouped in the form of rebates. Our work provides a quantitative basis for determining the harm done to users when their devices lose performance.

Implications for Security: As previously mentioned in Section 2, one of the reasons why longstanding yet fixable security issues like memory safety, Spectre v-1, and Rowhammer [29] persist is at least partially because security—which users cannot quantify or evaluate—comes at the expense of performance—which users *can* quantify and and evaluate—and product vendors are reluctant to trade performance for security if it makes their products appear less valuable in the marketplace [25]. Our results suggest that perhaps there is more overhead for security than previously thought. We point to two observations: First, during the incentive compatible study, all but one of the participants who accepted their offer participated for the full duration of the experiment. This indicates that, after accepting the first offer, participants' experience with throttled performance was, in general, not worse than expected. Second, the WTAs as found in the hands-on simulation study and incentive compatible study were both lower than the counterpart WTAs as found by the online surveys. Combined, this suggests that participants' high resistance to performance losses is more psychological than based on actual needs. While our results are for end users only (and not for server-class devices where perhaps customer information asymmetry is lower and the need for performance greater), the takeaway is that user resistance to performance losses due to patches and security updates may be artificially holding back the deployment of security.

Balancing Security Tradeoffs: As previously mentioned, our experiments provide the first

[3]We point out that our experiments were conducted on desktop and laptop users rather than phone users, but we believe the argument here still stands.

known effort to find the "exchange rate" between performance and user satisfaction. This exchange rate can help systems designers and architects quantitatively balance competing demands for security and performance: Suppose that a product contains a security vulnerability that, due to ransomware, costs users an average of $1000 per year, and that the average device lifespan is two years. Now suppose that architects develop a patch for the vulnerability, but that the patch incurs a 30% performance overhead. Is this a worthwhile tradeoff? Our results suggest no: Users require at least $4.43 per day to accept a 30% performance loss, or $3241.20 across a two-year span, which is a higher than the expected $2000 losses due to ransomware over the same two-year span. Now, suppose that architects improve the patch to incur only a 10% performance overhead. According to our results, such a defense "costs" users an average of $2.27 per day, or $1657.10 over a two year span, which is lower than the expected loss to ransomware; therefore, this new defense provides more protection than it costs, in terms of user value. Thus by finding the exchange rate between performance and user satisfaction, we provide architects with a novel user-centered metric that goes beyond traditional metrics like power, performance, area, and reliability.

Are Surveys a Useful Method for Architecture Research? A secondary research goal of this work was to answer, *are surveys a worthwhile method for user-centered architecture research?* Our findings suggest not. In both experiments, the hands-on studies yielded much lower WTAs than the counterpart online surveys. This supports our belief that users are not sufficiently knowledgeable or experienced with device performance to be able to make decisions that accurately reflect their true preferences. Unfortunately, this means that user-focused architecture research requires experimentation rather than relying on simple and easy-to-deploy surveys. Our work provides a template for future researchers on how to design and build such experiments.

Cross-Validating the Experiments: Another observation we make is that the results of the two sets of experiments may appear to be somewhat incongruous: Users would accept at 10% performance loss for $381 but also $2.27 per day, putting the "break even" point between the two results at roughly half a year[4]. If participants responded rationally in both cases, we might expect this "break even" point to be closer to device lifetime, which is almost certainly longer than half a year. What might this be? We offer two possible explanations: First, the simulation study is not incentive compatible, allowing for the possibility that the study participants in the TASKS-BASED STUDY did not answer the WTA question with as much thought and attention as it may have deserved. Another likely explanation is simply that humans are not perfectly rational when reasoning about small amounts (i.e. per day WTAs) vs. long-term events (i.e. device lifetime performance losses).

4 Proposal Topic II: Incentivizing the Creation and Adoption of Architectural Mechanisms for Security

Whereas the first work in my thesis focuses on measuring security tradeoffs, the second work focuses on achieving desired security tradeoffs in real-world systems. This work is explicitly aimed at actuating a desired balance of security between users, vendors, and authorities.

In this work the role of the authority is played by a regulator, i.e. either a government or perhaps a SRO (self-regulatory organization), and will hence be referred to as the "Regulator".

[4]I.e. after half a year, the value of receiving a daily $2.27 payment exceeds a flat rate payment of $381.

4.1 Background

Computer security has recently become a priority for governments across the world, who have been making efforts to improve security through mandates and regulation [22, 2, 48, 40, 21, 15, 47, 9]. Underpinning most of these efforts is the philosophy of "Secure By Design", the idea that security should be a built-in design requirement at each step of products' design, implementation, and deployment. While there are agreed-upon usable definitions of what secure-by-design means for both software [11] and hardware [10], there is currently no such definition for computer architecture. This is a cause for concern, for two reasons: First, architecture's unique role in the computing stack allows architects to develop cross-cutting security solutions that involve both hardware and software (e.g., virtualization or memory safety), and without an architectural definition of Secure By Design these system-level approaches to security may not be properly considered. And second, if computer architects do not formulate a usable definition of Secure By Design themselves, then given the regulatory momentum and importance of architecture to full-system security, architects may be forced to adopt concepts from software- and hardware-specific definitions of Secure By Design, likely leading to suboptimal outcomes. This work aims to create a usable definition of Secure By Design for architecture, and in the process, create a combined technical-regulatory system that balances the cost of security between users, vendors, and authorities.

We call our proposal FAIRSHARE (Fair Architecture-Inclusive Regulation for Secure Hardware via Allocated REsources). The basic idea is simple: *Require architecture-level product vendors to allocate a fixed percentage of a product's resources towards security*. In other words, FAIRSHARE asks hardware manufacturers to "stop passing the buck" when it comes to security [18] and allocate a fair share of resources towards a security budget. Importantly, in FAIRSHARE this security budget applies to traditional budgeted resources like product development costs, which represent non-recurring costs, *but also includes system resources like CPU cycles or energy*, which account for recurring expenses. FAIRSHARE specifies the roles and responsibilities for the parties involved (authorities, vendors, and users) and provides a technical means by which device-level resources (specifically CPU cycles) allocated for security can be measured, hence making our FAIRSHARE proposal not just a high-level goal but a concrete and enforceable regulatory mechanism as well.

Our proposal naturally brings up the question: how does one determine what percentage of resources should be allocated towards security? Rather than rely on social processes to determine this number, we demonstrate a quantitative approach: We design a model of a security game played between groups of Attackers and Defenders, using real-world data where possible, and then run simulations over the games' parameter space to study how the resource set-aside for security affects game dynamics and outcomes. Our data show that, for our simulations, a security budget of 20–40% provides the most protection at the least cost.

Also, for any regulatory proposal to be useful it needs to be enforceable. For our proposal this means that we need a way to measure how many resources are being spent on security in the field. We demonstrate the feasibility of doing this measurement using neural network regression models and show that the runtime overhead of security defenses can be captured with precision and at low cost.

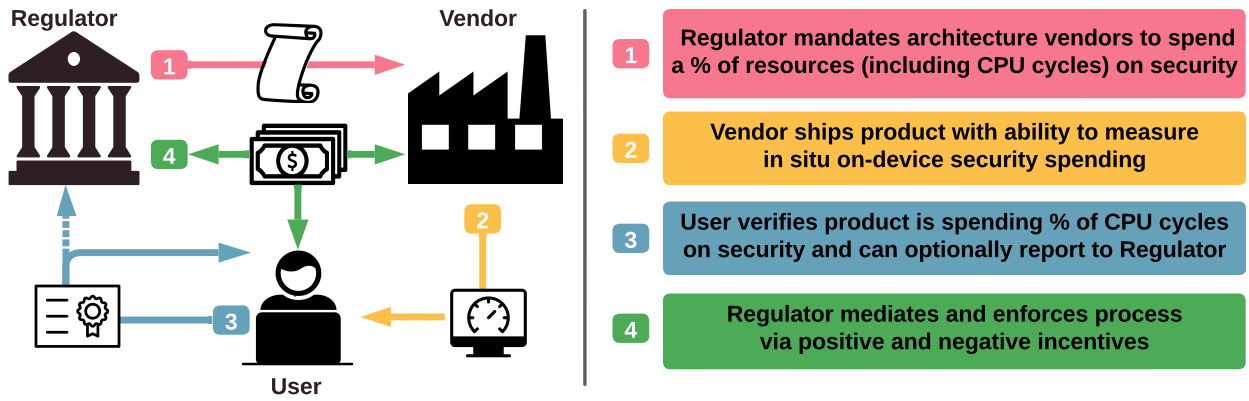


Figure 4: The FAIRSHARE mechanism outlined in four steps.

4.2 Our Solution: FAIRSHARE

FAIRSHARE is a four-step process of interactions and responsibilities between Regulator, Vendor, and User, shown in Figure 4 and described below.

Step 1: The regulator (or the government) requires that products for a certain sector (e.g., healthcare or critical infrastructure) dedicate at least a fixed percentage of resources towards security. At the vendor-level, this means that vendors must a portion of product development costs towards security-related expenses like security research & development, bug bounty programs, developer training, external security reviews (among others). To prevent the misappropriation of these funds earmarked for security, the Regulator requires Vendors to disclose how they spent this budget (more on this in Step 4).

A product's "resources" extend beyond just development costs though: The cost of security is paid not only during the product design phase but also in the field, where security comes at the expense of other system design requirements like performance or battery life. Hence the second device-level aspect of the mandate requires that *products themselves* must spend a fixed percentage of *their* resources on security as well. Since the notion of a product's "budget" is not clearly defined, we propose using either CPU cycles or energy consumption as a proxy (since both are in some sense the currency of compute); i.e. if applied to CPU cycles, the mandate might require that systems spend X% of their cycles on security-related functions (we demonstrate a working example of how this might be accomplished in Section 4.5).

Step 2: The Vendor chooses how to spend vendor-level and device-level security budgets as they see fit. To enforce the vendor-level aspect of the mandate, the Regulator can employ standard auditing practices to prevent Vendors from squandering this budget on frivolous or useless expenses. This topic is largely out of scope of this work (although we also argue in Section 4.2.1 that Vendors are incentivized to use the vendor-level budget wisely anyway).

For the device-level aspect of the mandate, the Vendor is free to select the defenses and security features that they see as providing the most security benefit given the required security budget. The Vendor also publishes the allocation of the product's security budget as part of the official product description for both the Regulator and User to see.

As part of this process, the Vendor must also provision their products to be able to measure on-device security budget expenditures, to confirm that the mandate is being met during real product

usage and that the combined defenses consume or "spend" all of the allocated security budget. A practical implementation of such an on-device, real-time mechanism is provided in Section 4.5.

Step 3: While the User uses the Vendor's product, the product measures the on-device system behavior to ensure that the system is allocating a sufficient amount of resources towards security. This is periodically recorded in audit logs generated by the device which the User can use to verify that the mandate is being met locally on their device.

Step 4: Finally, the Regulator mediates the FAIRSHARE system as a whole to ensure that the other parties are fulfilling their roles and responsibilities. First, the Regulator ensures that Vendors are allocating a sufficient percentage of product development costs towards security. Second, the Regulator ensures that Vendor's products are also allocating a sufficient percentage of system resources towards security. This can be achieved by testing products on the marketplace or by incentivizing Users to report their device-generated audit logs to the Regulator. To incentivize this process, the Regulator can reward Users for participation via positive incentives like tax credits. Various privacy-preserving mechanisms such as randomized responses and differential privacy may be employed to encourage audit reporting without exposing sensitive runtime information like usage patterns [50, 17]. Vendors found to be in violation of the mandate may be fined to encourage compliance.

4.2.1 Benefits of FAIRSHARE

The above process produces a virtuous cycle of architecture-based security improvements. We give five reasons below:

1) FAIRSHARE removes the obstacles for adopting cross-cutting architecture-based defenses. Recall that due to information asymmetry, there are incentives for vendors to favor performance over security. FAIRSHARE solves this problem by requiring all vendors to allocate an equal share of (performance) resources for security and preventing vendors from skimping on security. Under FAIRSHARE, architects are given the (performance) budget needed to defend systems against longstanding and critical weaknesses like memory safety without fear of losing out on marketplace competitiveness.

2) FAIRSHARE promotes a race-to-the-top competition between vendors to maximize the value of the security budget. A natural concern is that Vendors might waste the security budgets on useless or perhaps self-indulgent expenses (e.g. sending developers to security training seminars of dubious quality in expensive destination locations, or spending CPU cycles on useless computation in an effort to meet the mandate). However, we believe that FAIRSHARE will incentivize the opposite: Vendors have nothing to lose by making the most of the sunk cost and in fact can use their chosen allocation of the security budget as a selling point over competing Vendors' products. For example, a Vendor who implements defenses X, Y, and Z within the security budget gains a competitive advantage over a Vendor who implements only defenses X and Y in a comparable product. In other words, FAIRSHARE turns the market mechanism against itself to promote maximally efficient resource spending within the security budget. We point out that although information asymmetry is still at play and Users may never be able to thoroughly evaluate the merits of various defenses, FAIRSHARE at least makes security an advertised feature of a product's design, allowing domain experts and product reviewers to make reasonable judgments on behalf of non-experts. This also can be supplemented or integrated with ongoing work on device security labeling [19, 26].

3) FAIRSHARE puts security decision-making in the hands of the domain experts. An issue with many regulations is that those who write the rules (i.e. regulators and policymakers) are necessarily removed to some extent from the domains they are regulating. For architecture security regulation, the risk is that the regulators may not fully understand the burdens of their demands or are uninformed as what which parts of a systems are in most need of security investments. FAIRSHARE avoids this problem by letting the Vendors themselves decide where to spend the security budget[5]. This can also mitigate"regulatory capture" (which can occur if large incumbent technology companies lobby to craft regulations in their favor [13]), since each Vendor is free to spend the budget as they see fit.

4) FAIRSHARE makes security an explicit design feature. One of the root cause of systems and hardware insecurity is that the Vendors who make insecure products do not suffer the consequences of their insecurity (which instead is passed on to the User) and because Vendors at present lack liability for harms caused to Users as a result of product insecurity. This is complicated by the fact that vulnerabilities are generally not known until after product release, and it makes little sense to ex post facto hold Vendors liable after vulnerabilities are discovered, especially if the attack vectors themselves were previously unknown (such as the case with Spectre and Meltdown). FAIRSHARE addresses these misaligned incentives head-on by requiring Vendors to disclose how the security budget is spent (including at the device level), introducing a degree of responsibility and liability among Vendors. For example, if a product explicitly states that it is secured against buffer overflows and then later is found to be vulnerable to the very exploits it claims to defend against, then the product does not hold up to its advertised claims. Hence FAIRSHARE opens up an avenue of responsibility and liability previously unseen in systems and product design, further incentivizing Vendors to take security issues seriously.

5) FAIRSHARE fairly distributes the burden of security. One of the reasons why insecurity persists is that it is not clear who should have to pay to fix it [25]. FAIRSHARE presents a fair solution to this dilemma by making *all* players in the game of security pay for at least *part* of the burden: The Regulator pays for the cost of security by enacting and enforcing the mandate; the Vendor by implementing security in their products; the User by enduring the opportunity cost of resources dedicated to security instead of performance. Even attackers, who traditionally are not thought of as being one of the players who bear the burdens of security, are included in this model: By making across-the-board investments in security, FAIRSHARE forces attackers to work harder (i.e. spend more resources) to break systems; hence FAIRSHARE can be seen as a mechanism that places a distributed burden on regulators, vendors, and users, and an asymmetric burden on attackers.

4.2.2 Drawbacks and Other Considerations

As with any regulation, the cure must not be worse than the disease. We now aim to fairly present the drawbacks to FAIRSHARE (along with possible solutions) along with other considerations.

1) Different technology domains may not need the same fixed-percentage security budget. Indeed, an airgapped compute cluster does not have the same threat model as a public-facing

[5]We add the caveat that it is possible that some Vendors may have a low degree of in-house technical sophistication (especially if their products are comparatively low-tech) and perhaps are not particularly equipped to find and address their products' most pressing security issues. In this case, such Vendors should spend the product development budget on external security reviewers and consultants to help identify product weaknesses.

internet router, and it may not be appropriate to hold both to the same standard of security. A regulator may consider mandating different mandate levels for different domains and industries.

2) Compliance may raise the barrier to entry for startups and small businesses. To mitigate, a regulator might allocate funds towards grants or workshops to help startups and small businesses work towards compliance.

4.3 How Should Security Budgets be Set?

FAIRSHARE asks manufacturers to allocate a fraction of resources towards security. In this section, we propose and demonstrate a methodology for determining what this fraction might be. Our novel finding is that a higher security budget is not necessarily better, and that there is an optimum when security efficiency is considered.

We use an agent-based simulation with agents split between two groups, Attackers and Defenders. Agents are initialized with a fixed number of tokens that represent their wealth/resources. Attackers try to gain tokens by stealing from Defenders while Defenders try to prevent losses by making investments in security. The system is inefficient when the level of defense spending (i.e. `MANDATE`) does not minimize Defenders' overall losses.

In an effort to mirror reality, we make the following assumptions:

- Agents' initial tokens are drawn from a lognormal distribution (approximating the distribution of global wealth [12]).
- There are more Defenders than Attackers.
- At initialization, Defenders are wealthier than Attackers.
- Defenders exhibit a range of "security posture" (i.e. some are more vulnerable to attack than others).
- Attacks on wealthier defenders yield a higher payoff for the attacker but are also more expensive to accomplish.
- Attack success is probabilistic in nature but partially depends how much a Defender invests in security.
- A Defender can prevent attacks (or at least make attacks less likely) by making security investments.
- Attackers know Defenders' wealth, but Defenders do not know Attackers' wealth.
- Attackers only attack if expected earnings are > 0.

We formalize these heuristics into a set of six discretized game parameters in Table 1. Since we do not know which configuration of parameters best approximates the real world, we simulate all possible 1.1×10^6 parameter combinations.

To model the probabilistic nature of attacks, each defender `D` is initialized with a probability `D.p_atk_success`, which represents "security posture" or the probability that a given attack will be successful[6]. Defenders are also initialized with `D.attack_cost` $=$ `D.tokens` $\times$ `WAGER` to represent how much an attacker must spend to attempt an attack.

[6]This probability is drawn from a normal distribution with mean and standard deviation inferred from an industry report that finds, on average, 39% of ransomware attacks are successful with a standard deviation of 6.2% [44].

Parameter	Definition
MANDATE	% of defender assets that are spent on security
ATTACKERS	Number of Attackers (as a % of number of Defenders)
PAYOFF	% of a defender's assets that can be stolen in an attack
EFFICIENCY	% of MANDATE by which the cost to attack increases
INEQUALITY	Amount of Attackers' collective wealth (as a % of Defenders' wealth)
WAGER	% of a defender's assets that count toward D.atk_cost

Table 1: Simulations are initialized with six parameters, each of which can take a value between 0.1 and 1 at 0.1 intervals, except for MANDATE, which can also take the value 0.

Gameplay is iterated over a series of rounds. During each round, each attacker is paired to "fight" with a randomly selected defender by following Algorithm 1. Rounds are iterated until either all defenders lose all their tokens, or the game converges to a stable equilibrium[7].

Algorithm 1 Fight between an Attacker A and Defender D

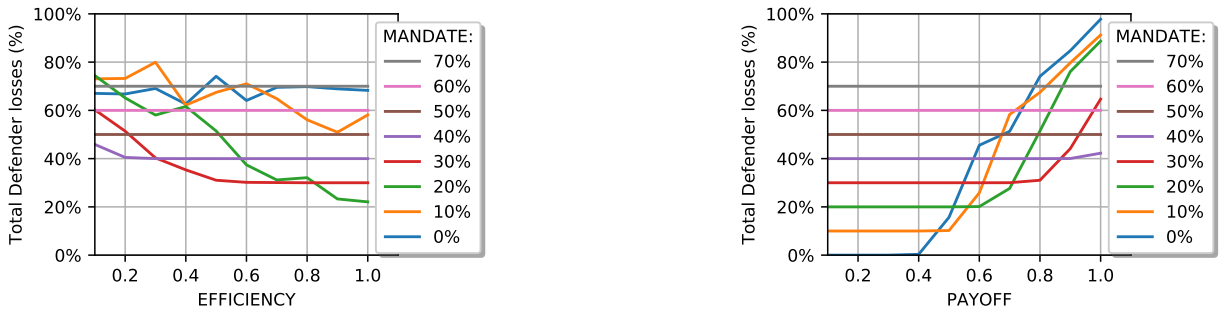
```
loot ← D.tokens × PAYOFF
expected_earnings ← loot × D.p_atk_success
if expected_earnings > D.atk_cost then
    if D.atk_cost < A.tokens then
        r ← random.uniform(0,1)
        if r < D.p_atk_success then  // A wins
            D.tokens ← D.tokens − loot
            earnings ← loot − D.atk_cost
            A.tokens ← A.tokens + earnings
        else // D wins
            A.tokens ← A.tokens − D.atk_cost
        end if
    end if
end if
```

4.4 Simulation Results and Findings

To narrow the large parameter space, we prune the games where Defenders do not suffer losses (since these parameter settings are not representative of the real world).

We calculate the expected value of each parameter (MANDATE=0.2, ATTACKERS=0.5, PAYOFF=0.8, EFFICIENCY=0.5, INEQUALITY=0.5, and WAGER=0.3) to give us a "baseline" parameter setting; the distance from 0.5 gives a relative measure of how strongly biased a parameter is towards favoring Attackers or Defenders.

[7]In our games, convergence is defined as when the sum total of all Defenders' tokens changes by less than ϵ for at least δ rounds of the game. We choose $\epsilon = 100$ (which is very small relative to Defenders' initial sum tokens) and $\delta = 50$ (which is a significant fraction of most games' number of rounds). No games failed to reach a stable equilibrium.



(a) Defender losses across a sweep of EFFICIENCY for various MANDATE levels. The optimal MANDATE is 20%, 30%, or 40%, depending on the value of EFFICIENCY.

(b) Defenders losses across a sweep of the PAYOFF for various MANDATE levels. The optimal MANDATE ranges from 0% to 40% depending on the value of PAYOFF.

Figure 5: Dynamics of the simulation are revealed by sweeping parameters across their full range of values while simultaneously holding all other parameters at their "baseline" value.

Using this baseline configuration, we observe how a given mandate affects total Defender losses (= mandate cost + losses to Attackers) across a range of parameter values. This approach reveals—for each value of each parameter—which level of MANDATE minimizes Defenders' collective total losses.

To illustrate, see Figure 5a, which sweeps across values of the EFFICIENCY parameter for various levels of MANDATE. If EFFICIENCY < 0.3 (meaning that security spending does little to raise security posture) then losses are minimized when MANDATE=40%. However, if $0.3 \leq$ EFFICIENCY ≤ 0.8, then losses are minimized when MANDATE=30%. When EFFICIENCY > 0.8, only a 20% security mandate is needed to minimize Defender losses. For all values of EFFICIENCY, MANDATE $< 20\%$ fails to sufficiently protect against losses while MANDATE $> 50\%$ is too costly to justify the additional protection.

We repeat for the PAYOFF parameter in Figure 5b. When PAYOFF < 0.5, Defenders suffer no losses at all (presumably because there is not sufficient incentive for Attackers to attack). As PAYOFF increases, losses are minimized under a 10%, 20%, 30%, and finally a 40% MANDATE (when PAYOFF ≥ 0.9). Above 40%, the cost of the mandate itself outweighs the protections it provides. While not shown here, we add that for the WAGER parameter, Defender losses are also minimized when $0.0 <$ MANDATE < 0.4. We omit plots for the ATTACKERS and INEQUALITY parameters, since we find that these parameters do not have an effect on Defenders' collective losses. Hence losses are collectively minimized across all parameters when $0.2 \geq$ MANDATE ≥ 0.4.

The benefit of this approach is that we can observe what security budget levels are most effective across the full range of parameter values without requiring us speculate on what these values might actually be in the real world. For example, although there is undoubtedly some relationship between security spending and resulting security posture, the real-world function between the two is largely unknown due to a lack of publicly available data. Our approach allows us to build models from reasonable heuristics and make observations about the dynamics of security in the absence of strong quantitative real-world measurements.

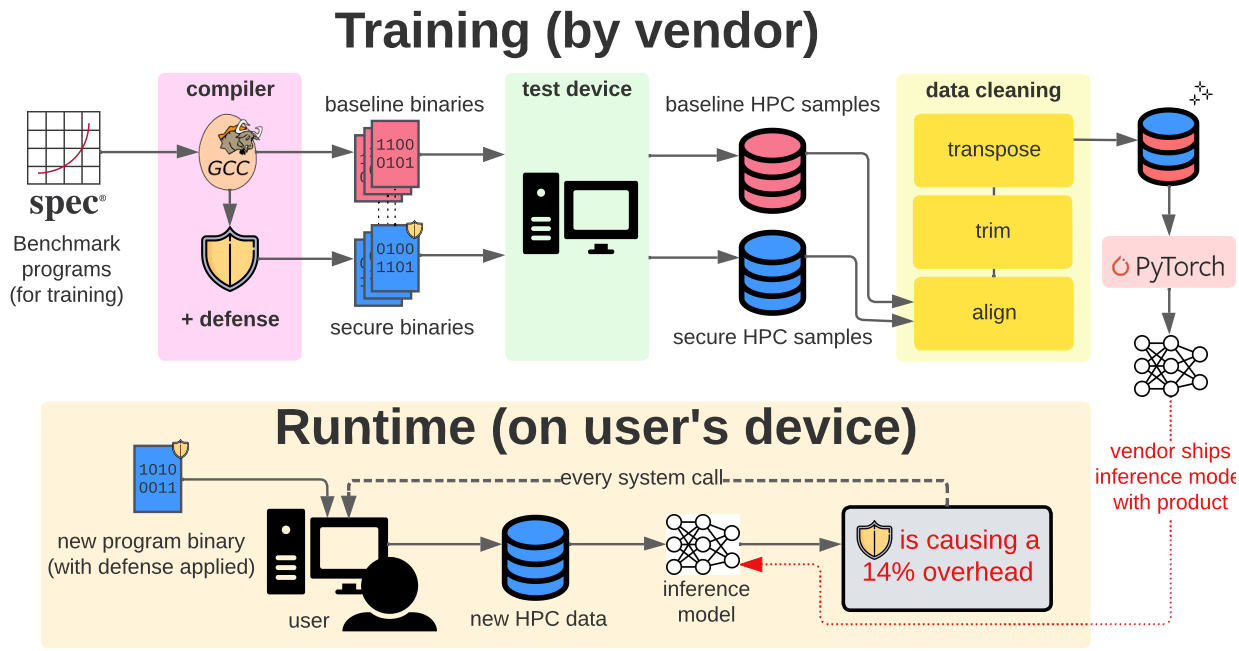


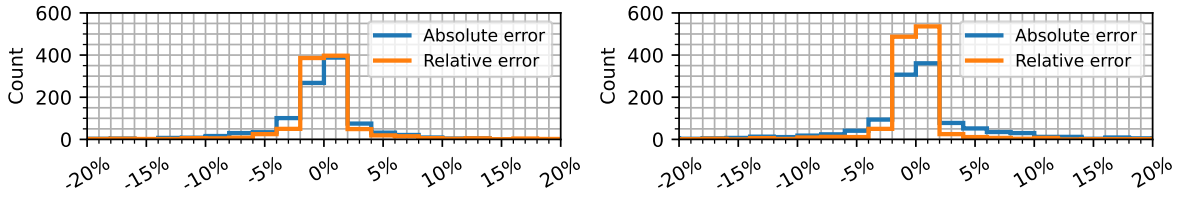
Figure 6: An implementation of the FAIRSHARE protocol defined in Figure 4.

4.5 Implementing FAIRSHARE

FAIRSHARE requires a mechanism that can measure resources allocated for security. This requires measurements to be taken **on-device** and **continuously** (since pre-computed overheads are only valid for the test devices and benchmarks used and are highly sensitive to changes in system configuration and usage behavior).

To achieve this, we train a predictive neural network model that leverages system data collected via specially chosen hardware performance counters ("HPCs") that track key indicators of runtime performance. First, we create training data by running the SPEC2017 benchmark suite with and without defenses applied and collect HPC data at each system call boundary (using the tools DynamoRIO and easyperf). Next, we perform sequence alignment on execution traces of the "baseline" and "secured" variants (using collected syscall numbers) to find periods of equivalent execution and determine the defenses' performance overhead *at each system call*.

Using this as our training data, we train a model using PyTorch that inputs 20 HPC events and predicts current security overhead as a scalar output[8]. We use a 90/5/5 split for the training, testing, and validation sets, respectively. To avoid overfitting, we choose the model that minimizes the validation set's loss. We also reduce model size by using half-precision (16-bit) weights to highlight the feasibility of building the model in hardware and achieve a final model size of 12KB.



(a) Binary hardening overhead estimation errors

(b) Memory safety technique estimation errors

Figure 7: Our DNN-based regression model estimates the real-time overhead of security for two contemporary defenses with high precision and accuracy.

4.5.1 Evaluation

We use the above process for two defenses: compiler-based binary hardening flags[9], and No-FAT [55]. We evaluate the models by comparing predicted overhead to observed overhead using both absolute and relative error. Results are plotted in Figure 7. For both defenses, we find that our models estimate performance overheads with very high accuracy and precision.

5 Proposal Topic III: Modeling the Effect of Cyberinsurance

The last work to be included in my thesis is my work on cyberinsurance. This portion of my thesis is a work in progress.

5.1 Background

Cyberinsurance is a tool organizations use to mitigate cyber risks. Like other forms of insurance, an insuree (or policyholder) pays a fee (the "premium") to an insurer who agrees to cover (or indemnify) losses in case of damages (e.g. loss of business due or to denial of service attacks, or to cover settlements with affected third parties) [14, 39]. Insurers aggregate risk by collecting premiums from multiple policyholders into a shared pool of funds (ibid). Policyholders who suffer losses may submit a claim to their insurer, who, after verifying the claim, disburses funds to cover the affected policyholder's losses (ibid). As part of the underwriting process, an insurer will use interviews, questionnaires, and other forms of data collection to estimate a customers' risk to cyber incidents and may use this information to price premiums accordingly. Policies typically define what types of losses are covered and by how much ("coverage limits") and specify actions or conditions under which coverage does not apply ("exclusions").

For years cyberinsurance has been presumed to one day be an efficient mechanism for improving security: The theory is that insurers, wanting to minimize payouts to insurees, will only sell policies to customers who implement certain security controls (or offer discounted premiums to those who do); combined with their risk forecasting expertise and access to privileged actuarial data, insurers may be able to quantitatively determine which security investments are worth

[8] We use four layers with Leaky ReLU activation functions. We used the Adam optimizer with a learning rate of 10^{-3}, an epsilon of 10^{-4}, and Mean Squared Error (MSE) as our loss function.

[9] Specifically, `-fPIE -pie -D_FORTIFY_SOURCE=2 -Wl, -z now, -z relro, -fstack-protector-all -fsanitize=safe-stack`

making and be and drive the market accordingly[10]. However, this process is impeded in practice: The costs of security investment may not be worth the cost of otherwise higher premiums [54]; brokerage practices incentivize non-stringent application procedures [51]; legal processes against information sharing prevent the collection of useful aggregated actuarial data [53]. Nevertheless, with the rise of ransomware attacks the market for cyberinsurance has grown, but this has been accompanied by reduced coverage, reduced limits, and spikes in premiums [49, 52].

5.2 Methods

To explore the effects of cyberinsurance, I use modeling and simulation techniques. This work builds upon the modeling in Section 4 but makes many key modifications. It adds a new group of players—Insurers—who sell insurance policies to Defenders.

In this model, each defender is imbued with a new parameter—`posture`, where $0 \leq \texttt{posture} \leq 1$—which represents the defender's level of security.

I also add an additional constraint that insurers profits are capped at a certain percentage V, meaning that that must pay out at least $1 - V$ in claims to insurees (this is akin to insurers' capped profits as specified in the Affordable Care Act). Insurers will therefore charge premiums to be the most they can while satisfying this constraint, i.e.

$$P = \frac{\mathbb{E}[\text{gain}_I]}{1 - V} \tag{1}$$

where V is the overhead that the insurer gets to keep and spend on administrative purposes (and is set to, say, $V = 20\%$). Then the amount the insurer expects to gain from a given policy is therefore

$$\mathbb{E}[\text{gain}_I] = (p_L)(R + P - \mu_{\text{P}}(\texttt{d.Assets})) + (1 - p_L)(P) \tag{2}$$

$$= P + p_L(R - \mu_{\text{P}}(\texttt{d.Assets})) \tag{3}$$

where P is the premium, R is the retention (also known as the deduction), p_L is the probability that a defender will be looted during a given round of gameplay, μ_{P} is the mean of the distribution of the payoff parameter scaling factor, and $(\texttt{d.Assets})$ is the defender `d`'s current assets. The retention R can be expressed in terms of the premium. Regression analysis of Romanosky 2019 finds that this is a linear function with $R = rP = 25P$ [39]. Plugging (1) into (3) and solving yields the following:

$$P = \frac{p_L \mu_{\text{P}}(\texttt{d.Assets})}{rp_L + V} = \frac{p_L \mu_{\text{P}}(\texttt{d.Assets})}{25p_L + V} \tag{4}$$

where p_L is defined as the probability an attack is attempted p_A scaled by the defender `d`'s risk, i.e. $p_L = p_A(1 - \texttt{d.Posture})$.

The insurer, using research and threat intelligence, can compute p_A, the precise probability a defender `d` will be attacked.

[10]This mechanism is standard practice in non-cyber insurance policies. For example, commercial fire insurance is typically contingent on policyholders installing fire safety control mechanisms like sprinklers [41].

$$p_A = \max\left(1, \frac{|\text{attackers}|}{|\text{defenders}|}\right) \cdot (p(\texttt{d.assets} > \texttt{d.costToAttack})) \tag{5}$$

where $\texttt{d.costToAttack} = (\texttt{d.Assets}) \times \texttt{d.Posture}$. The defenders don't have this information so they do a crude approximation,

$$\hat{p_A} = \frac{\text{num of attacks previous round}}{|\text{defenders}|}, \qquad \hat{p_L} = \hat{p_A}(1 - \texttt{d.Posture}) \tag{6}$$

With this information, the defender can now choose a strategy to minimize expected losses. Available strategies are invest in security, buy insurance or do nothing.

The loss the defender d expects to lose if they purchase insurance therefore is given by

$$\mathbb{E}[\text{loss,with insurance}] = P + \hat{p_L}R \tag{7}$$

The alternative is for the defender to forgo insurance and try to increase security posture themselves. An investment into security yields the expectation

$$\mathbb{E}[\texttt{d.posture}] = \min\left(1, \quad \mu_{\text{E}}\left[\texttt{d.posture}\left(1 + \frac{x}{\texttt{d.assets}}\right)\right]\right) \tag{8}$$

The expected loss is

$$\mathbb{E}[\text{loss (without insurance), x}] = \hat{p_L}((\texttt{d.Assets}) - x)\mu_{\text{P}} + x \tag{9}$$

which (thanks to Mathematica) achieves the following minimum:

$$\min(\mathbb{E}[\text{loss (without insurance), x}]) = -\frac{(\texttt{d.Assets})(1 - 2(p_A\mu_{\text{P}} + p_A(1 - 2(\texttt{d.Posture})\mu_{\text{E}})^2)\mu_{\text{P}}^2}{4(\texttt{d.Posture})p_A\mu_{\text{E}}\mu_{\text{P}}} \tag{10}$$

$$\mathbb{E}[\text{loss (without insurance)}] = -\frac{(\texttt{d.Assets})(1 - 2(p_A\mu_{\text{P}} + p_A(1 - 2(\texttt{d.Posture})\mu_{\text{E}})^2)\mu_{\text{P}}^2}{4(\texttt{d.Posture})p_A\mu_{\text{E}}\mu_{\text{P}}} \tag{11}$$

The defender then chooses the strategy that minimizes their expected losses. Gameplay otherwise continues as specified in Section 4.

5.3 Results

The outcome of this game cannot be determined analytically and must be simulated. I have built a monte carlo simulator of this model, but since this is still a work in progress, I do not yet have full results. However, I have been able to validate key expected behavior. In particular, I have been able to validate that the model creates sufficiently rich gameplay where defenders rationally choose between purchasing security investment, purchasing cyberinsurance, and doing nothing at all, depending on which option is expected to minimize overall losses.

I am currently working on running the model to test two hypotheses:

1. *Do investments in security or insurance produce more efficient outcomes?*

2. *Does adding more insurer players create a "race to the bottom" effect?* My suspicion is that, due to the uncertainty inherent when insurers attempt to estimate insurees' security posture, the presence of more insurers will increase the likelihood that an insurer overestimates an insuree's posture and offer a policy that will ultimately lose the insurer money. Therefore, does adding more insurers to the game worsen security outcomes?

3. *What happens when I increase the variance of the security posture estimation function?*

4. *Is there an optimum point of security investment?*

6 Research plan

The majority of the work needed to complete my thesis has already been done. Table 2 shows my plan for completion of the research.

Timeline	Work	Progress
	Publish *A New Doctrine for Hardware Security* (ASHES'20 workshop paper)	completed
	Publish *How Much is Performance Worth to Users?* (Computing Frontiers '23)	completed
	Publish *Architecture Security Regulation* (Computer Architecture Letters)	completed
	Submit *Incentivizing the Creation and Adoption of Arch. Mechanisms for Sec.* (ISCA'24)	completed
Feb. 2024	Complete cyberinsurance work and submit to Usenix Security	ongoing
Spring 2024	Thesis writting	planned
June 2024	Thesis defense	planned

Table 2: Plan for completion of my research

Given the above timelines, I plan to defend my thesis in six months' time, in June 2024.

References

[1] In re: Apple inc. device performance litigation.

[2] National cybersecurity strategy, Mar 2023.

[3] Ruba Abu-Salma, M. Angela Sasse, Joseph Bonneau, Anastasia Danilova, Alena Naiakshina, and Matthew Smith. Obstacles to the adoption of secure communication tools. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 137–153, 2017.

[4] Akamai. The state of online retail performance, 2017.

[5] George A Akerlof. The market for lemons: Quality uncertainty and the market mechanism. In *Uncertainty in economics*, pages 235–251. Elsevier, 1978.

[6] Ross Anderson. Why information security is hard-an economic perspective. In *Seventeenth Annual Computer Security Applications Conference*, pages 358–365. IEEE, 2001.

[7] Ross Anderson, Chris Barton, Rainer Böhme, Richard Clayton, Carlos Ganán, Tom Grasso, Michael Levi, Tyler Moore, and Marie Vasek. Measuring the changing cost of cybercrime. 2019.

[8] Ross Anderson and Tyler Moore. The economics of information security. *Science*, 314(5799):610–613, 2006.

[9] Cybersecurity Certification Centre. Cybersecurity certification guide. Technical report, Cyber Security Agency of Singapore, 2021.

[10] The MITRE Corp. Cwe view: Hardware design.

[11] The MITRE Corp. Cwe view: Software development.

[12] Credit Suisse Research Institute. Global wealth report 2021. Technical report, June 2021.

[13] Ernesto Dal Bó. Regulatory capture: A review. *Oxford review of economic policy*, 22(2):203–225, 2006.

[14] Savino Dambra, Leyla Bilge, and Davide Balzarotti. Sok: Cyber insurance–technical challenges and a system security roadmap. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1367–1383. IEEE, 2020.

[15] Media Department for Digital, Culture and United Kingdom Sport. New smart devices cyber security laws one step closer. *Press release*, 2022.

[16] Amitava Dutta and Rahul Roy. Dynamics of organizational information security. *System Dynamics Review: The Journal of the System Dynamics Society*, 24(3):349–375, 2008.

[17] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[18] Jen Easterly and Eric Goldstein. Stop passing the buck on cybersecurity, Feb 2023.

[19] Pardis Emami-Naeini, Yuvraj Agarwal, Lorrie Faith Cranor, and Hanan Hibshi. Ask the experts: What should be on an iot privacy and security label? In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 447–464. IEEE, 2020.

[20] Pardis Emami-Naeini, Henry Dixon, Yuvraj Agarwal, and Lorrie Faith Cranor. Exploring how privacy and security factor into iot device purchase behavior. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[21] European Union European Commision. Proposal for a regulation of the european parliament and of the council on horizontal cybersecurity requirements for products with digital elements and amending regulation (eu) 2019/1020. Technical report, European Commission, 2022.

[22] Executive Office of the President. Executive Order 14028. Technical report, May 2021.

[23] Dennis F Galletta, Raymond Henry, Scott McCoy, and Peter Polak. Web site delays: How tolerant are users? *Journal of the Association for Information Systems*, 5(1):1, 2004.

[24] Julie Haney, Yasemin Acar, and Susanne Furman. "it's the company, the government, you and i": User perceptions of responsibility for smart home privacy and security. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 411–428. USENIX Association, August 2021.

[25] Adam Hastings and Simha Sethumadhavan. Wac: A new doctrine for hardware security. In *Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security*, pages 127–136, 2020.

[26] The White House. Fact sheet: Biden-harris administration delivers on strengthening america's cybersecurity. *Statements and Releases*, 2022.

[27] John A Hoxmeier and Chris DiCesare. System response time and user satisfaction: An experimental study of browser-based applications. 2000.

[28] Patrick Gage Kelley, Joanna Bresee, Lorrie Faith Cranor, and Robert W Reeder. A" nutrition label" for privacy. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–12, 2009.

[29] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. Flipping bits in memory without accessing them: An experimental study of dram disturbance errors. *ACM SIGARCH Computer Architecture News*, 42(3):361–372, 2014.

[30] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, et al. Spectre attacks: Exploiting speculative execution. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1–19. IEEE, 2019.

[31] Yehuda Kotowitz. Moral hazard. In *Allocation, information and markets*, pages 207–213. Springer, 1989.

[32] Katharina Krombholz, Karoline Busse, Katharina Pfeffer, Matthew Smith, and Emanuel von Zezschwitz. "if https were secure, i wouldn't need 2fa" - end user and administrator mental models of https. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 246–263, 2019.

[33] Katharina Krombholz, Wilfried Mayer, Martin Schmiedecker, and Edgar Weippl. "i have no idea what i'm doing" - on the usability of deploying HTTPS. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1339–1356, Vancouver, BC, August 2017. USENIX Association.

[34] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. Meltdown. *arXiv preprint arXiv:1801.01207*, 2018.

[35] Tyler Moore. The economics of cybersecurity: Principles and policy options. *International Journal of Critical Infrastructure Protection*, 3(3-4):103–117, 2010.

[36] Katharina Pfeffer, Alexandra Mai, Adrian Dabrowski, Matthias Gusenbauer, Philipp Schindler, Edgar Weippl, Michael Franz, and Katharina Krombholz. On the usability of authenticity checks for hardware security tokens. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 37–54. USENIX Association, August 2021.

[37] John Poole. iPhone Performance and Battery Age, 2017.

[38] Joshua Reynolds, Trevor Smith, Ken Reese, Luke Dickinson, Scott Ruoti, and Kent Seamons. A tale of two studies: The best and worst of yubikey usability. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 872–888, 2018.

[39] Sasha Romanosky, Lillian Ablon, Andreas Kuehn, and Therese Jones. Content analysis of cyber insurance policies: How do carriers price cyber risk? *Journal of Cybersecurity*, 5(1):tyz002, 2019.

[40] White House Briefing Room. Biden-harris administration announces cybersecurity labeling program for smart devices to protect american consumers, July 2023.

[41] Bruce Schneier. Insurance and the computer industry. *Communications of the ACM*, 44(3):114–114, 2001.

[42] Bingyu Shen, Lili Wei, Chengcheng Xiang, Yudong Wu, Mingyao Shen, Yuanyuan Zhou, and Xinxin Jin. Can systems explain permissions better? understanding users' misperceptions under smartphone runtime permission model. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 751–768. USENIX Association, August 2021.

[43] Margaret W Smith. Information asymmetry meets data security: The lemons market for smartphone apps. *Pol'y Persp.*, 26:85, 2019.

[44] Sophos. The state of ransomware 2021. Accessed: 2021-11-2.

[45] Jim Scovell Sudha Ganesh and Clem Wong. Measuring what matters: Project athena innovation programs real-world testing, 2020.

[46] Matthew Tischer, Zakir Durumeric, Sam Foster, Sunny Duan, Alec Mori, Elie Bursztein, and Michael Bailey. Users really do plug in usb drives they find. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 306–319, 2016.

[47] TRAFICOM. Statement of compliance for the cybersecurity label. Technical report, National Cyber Security Centre, Finland, 2019.

[48] United States 116th Congress (2019-2020). H.r.1668 - iot cybersecurity improvement act of 2020, 2020.

[49] United States Government Accountability Office. CYBER INSURANCE: Insurers and Policyholders Face Challenges in an Evolving Market, May 2021.

[50] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[51] Daniel Woods and Andrew Simpson. Policy measures and cyber insurance: a framework. *Journal of Cyber Policy*, 2(2):209–226, 2017.

[52] Daniel W Woods. A turning point for cyber insurance. *Communications of the ACM*, 66(3):41–44, 2023.

[53] Daniel W Woods, Rainer Böhme, Josephine Wolff, and Daniel Schwarcz. Lessons lost: Incident response in the age of cyber insurance and breach attorneys. In *Proceedings of the 32nd USENIX Security Symposium, Anaheim, California*, 2023.

[54] Daniel W Woods and Tyler Moore. Does insurance have a future in governing cybersecurity? *IEEE Security & Privacy*, 18(1):21–27, 2019.

[55] Mohamed Tarek Ibn Ziad, Miguel A Arroyo, Evgeny Manzhosov, Ryan Piersma, and Simha Sethumadhavan. No-fat: Architectural support for low overhead memory safety checks. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pages 916–929. IEEE, 2021.