Determining an Optimal Threshold on the Online Reserves of a Bitcoin Exchange

Samvit Jain, Edward Felten, Steven Goldfeder

Princeton University, Department of Computer Science

Abstract

Online and offline storage of digital currency present conflicting risks for a Bitcoin exchange. While bitcoins stored on online devices are continually vulnerable to malware and other network-based attacks, offline reserves are endangered *on access*, as transferring bitcoins requires the exposure of otherwise encrypted and secured private keys. In particular, fluctuations in customer demand for deposited bitcoin require exchanges to periodically refill online storage systems with bitcoins held offline. This raises the natural question of what upper limit on online reserves minimizes losses due to theft over time.

In this paper, we investigate this optimization problem, developing a model that predicts the optimal ceiling on online reserves, given average rates of deposits, withdrawals, and theft. We evaluate our theory with an event driven simulation of the setup, and find that our equation yields a numerical value for the threshold that differs by less than 2% from empirical results. We conclude by considering open questions regarding more complex storage architectures.

1 Introduction

On January 5, 2015, Bitstamp, the world's third largest Bitcoin exchange [1], abruptly suspended operations. The UK-based service had detected theft of 19,000 bitcoins, worth $5.1 million at the time of press release [2]. In response to terrified customers and media frenzy, Bitstamp's CEO issued the following public statement:

> This breach represents a small fraction of Bitstamp's total bitcoin reserves, the overwhelming majority of which are held in secure offline cold storage systems. We would like to reassure all Bitstamp customers that their balances. . . will not be affected and will be honored in full [2].

Though unperturbed by such incidents to date, Bitstamp's American counterpart – the San Francisco-based wallet and exchange service Coinbase – assures a clientele spanning 24 countries:

Sleep Well Knowing Your Bitcoin Are Safe

Up to 97% of bitcoin is stored totally offline, in geographically distributed safe deposit boxes and physical vaults [3].

The public fears these statements aim to placate are not, in fact, unfounded. Bitcoin theft is alarmingly prevalent, and impacts both businesses managing vast reserves and individuals holding small quantities of bitcoin on their personal computers. The mechanisms of theft are numerous. Unsuspecting smartphone users often fall victim to malicious Android applications advertised as Bitcoin wallets [4]. Bitcoins stored on devices connected to the Internet are frequently compromised by various forms of malware [5], which extract and transmit the private keys used to authorize Bitcoin transactions.[1] Patrons of well-known exchanges, including Coinbase, often report lower-than-expected account balances, having been victimized by hackers who acquired their login credentials [7]. And major services, such as Bitstamp, periodically lose significant holdings of bitcoin to security exploits in client-facing software; in some cases, the responsible parties include company insiders [8].

Both Bitstamp and Coinbase's public assertions also allude to a second, critical aspect of Bitcoin management, and the central focus of this study – the concept of offline and online storage. Whereas storing bitcoins on devices connected to the Internet (online, or "hot", storage) is traditionally discouraged, as it entails exposure to malware contracted through the web and other network-based attacks, offline ("cold") storage involves its own hazards, specifically, the danger of compromise *on access*. For a Bitcoin exchange or banking service that must consistently meet customer demand, this results in a logistic dilemma. Storing too many bitcoins in hot storage poses the obvious problem of increased losses due to recurrent, network-based theft. But storing fewer bitcoins online necessitates frequent access of cold storage to meet fluctuations in customer demand. This in turn defeats the functional purpose of cold storage, which is to exchange liquidity for increased security. In particular, frequent access increases the probability of *cold storage theft*. This second risk has been underemphasized in the current literature, to the point that cold storage is increasingly portrayed as the definitive solution to most problems in Bitcoin security. This tendency can be seen in research papers [9], community documentation on Bitcoin [10], and in public security claims by major companies [11], [12].

1.1 Contributions

In this paper, we challenge the assumption that the only benefit to storing bitcoins in hot storage is availability, by demonstrating that maintaining some optimal value of online reserves in fact minimizes losses

[1] Notably, an estimated 3.4 million instances of Bitcoin malware were detected in 2014 [5], 22% of all financial malware [6].

due to theft. Our quantitative analysis confirms the idea that storing *too few* bitcoins in hot storage results in an arrangement that exposes the bulk of an organization's reserves to a small but continual probability of theft, which – given the track record of Bitcoin exchanges – can be catastrophic in the long run.

The heightened significance we attach to cold storage theft is motivated by an empirical study of 40 major Bitcoin exchanges operational at some point before January 2013, which found that 18 had ultimately shutdown, at least 5 of which had failed to reimburse their customers [13]. In particular, while more popular exchanges were less likely to shutdown, the likelihood of a security *breach* was positively correlated with the transaction volume handled by the exchange [13]. Though the details of these thefts are generally unknown, several explicit cases of cold wallets being emptied have been documented [8].

Given this evidence, we adopt a different approach to Bitcoin theft. While previous work has focused on the cryptographic layer, and reducing the *incidence* of theft (see Section 4), we instead investigate the optimal utilization of existing security systems. Our setup consists of a Bitcoin exchange that must service deposits and withdrawal requests, while mitigating losses due to unavoidable, periodic theft of its hot and cold storage systems. Specifically, we stipulate that cold storage theft occurs with a fixed probability on access, while times to hot storage theft are exponentially distributed. We model deposits and withdrawals, on the other hand, as Poisson processes. We then investigate the behavior of our system over a long time interval $[0, T]$, tracking the net balance of the exchange through internal and external events.

Notably, we propose a series of models that quantify the performance of various subsystems of our setup; namely, 1) net income into the exchange, 2) hot storage with no offline backup, and 3) the full dual storage system. Our culminating result is a formula for the expected net value of our exchange after T hours. This function is then numerically optimized, yielding a value for an optimal ceiling on online reserves which differs by less than 2% from empirical results. We conclude by discussing more complex storage architectures and their potential advantages.

1.2 Motivation

Mitigating losses due to Bitcoin theft is an undertaking of crucial importance on several levels. Firstly, Bitcoin's success as an emerging currency and alternative payment system is critically dependent on public trust in its institutions. Public optimism about Bitcoin determines its current dollar valuation, motivates entrepreneurs to build the tools that make Bitcoin useful for the general person, incentivizes developers to contribute improvements to the Bitcoin protocol, and spurs investment into security and privacy research. But public opinion is also particularly *sensitive* to news of heists and shutdowns, and to stories of major exchanges going bankrupt. As a result, Bitcoin theft not only affects its immediate victims - businesses and their customers, but hurts the Bitcoin community at large and hampers greater adoption of the currency.

A key economic principle is also at play. Losses due to theft experienced by Bitcoin storage and exchange services are subsidized by customers, through increased exchange fees and (in the future) higher insurance premiums or lower interest rates. This in turn is a disincentive for customers to store (i.e. invest) their savings in Bitcoin services. One of the key factors driving Bitcoin's growth today is that it *reduces* frictions involved in traditional payment mechanisms, by cutting out intermediary parties and automating transactions. These benefits are nullified, however, if Bitcoin remains a high-risk investment.

2 Background

Two aspects of Bitcoin are of crucial importance to this study. The first is the concept of Bitcoin ownership, which is a cryptographically-enforced guarantee that is published in a global ledger. The second is hot and cold wallet storage, a software and security abstraction that underpins the everyday usage of Bitcoin.

2.1 Bitcoin Ownership

An entity gains ownership of bitcoins by being the recipient of a publicly broadcasted Bitcoin *transaction*, a record of which is consolidated and published in a global log (the blockchain) through a decentralized, distributed mechanism (Bitcoin mining). A transaction specifies both senders and recipients, referenced by their respective 160-bit *public addresses*. Each public address is associated with a public and private key pair; in fact, the public address is just an encoded hash H of the public key PK. To send bitcoins to Bob, Alice must *digitally sign* with her private (secret) key SK_{Alice} a transaction of some value to Bob's public address $H(PK_{Bob})$. Alice's digital signature affirms that bitcoins previously transferred to her (i.e. to $H(PK_{Alice})$) by some third entity, say Carol, now in fact belong to $H(PK_{Bob})$.

Note that an entity, such as an individual Alice or a banking service Bob, may choose to create and be associated with *multiple* public addresses. That entity is then responsible for protecting the corresponding private key for each address. Misplacing or destroying a private key results in an irrecoverable loss of any associated bitcoins, as it prevents those bitcoins from ever being transferred. Crucially, bitcoins can also be *stolen*. If a malicious entity Mallory learns of Alice's private key SK_{Alice}, she can create and sign a transaction transferring any associated bitcoins to one or more addresses owned by Mallory. As of now, there exists no legal or cryptographic measure in the Bitcoin protocol to reverse or even detect such transactions. Though it is surprisingly easy to link clusters of highly active public addresses to real world identities [14], determining the legitimacy of transactions (beyond specific kinds of fraud, such as *double spending*) is outside the scope, and antithetical to the motivations, of the Bitcoin system. This starkly contrasts fraudulent credit card activity, which, while a rampant problem in the United States and a major public burden, is relatively easy to challenge and reverse. In particular, while credit card users operate in a system critically reliant on the incentives of *reputation* – namely, that of credit card companies, credit card holders, and businesses and storeowners, Bitcoin owners construct transactions under a protocol that has exchanged institutional authority for pseudonymity and decentralization. The result – that Bitcoin theft is irreversible, and thus particularly damaging – is one of the major motivating ideas for this study.

2.2 Hot and Cold Wallet Storage

The second key concept underlying this study is that of hot and cold wallet storage. A Bitcoin *wallet* is a container for one or more private keys, often encrypted for confidentiality and stored in a secure location. Though a Bitcoin wallet does not *physically* contain any bitcoins, treating it as an account with a certain value is a useful abstraction that we will adopt in this paper. Bitcoin wallets come in many forms; common examples include an encrypted file on a hard disk locked in a safe, paper wallets with printed keys, an iPhone application secured through a passphrase, and a secret sharing scheme involving multiple, diversified points of high security in an organization.

A hot wallet is a collection of private keys stored on a device connected to the Internet. Hot wallets provide convenience and accessibility, but at a cost, as network connection entails a greater risk of compromise to external threats, such as botnets, spyware, or sophisticated attacks from an active, malicious adversary. For certain organizations and individuals, this may be a necessary price to pay. A high-frequency trader, for example, may require immediate access to her private keys to exploit transient fluctuations in currency value. A banking service, on the other hand, may be bound to its customers, who expect *availability* of deposited bitcoins. In general, hot wallets are secured through proper encryption practices, anti-malware software, limitations on Internet access, and specialization of the container device.

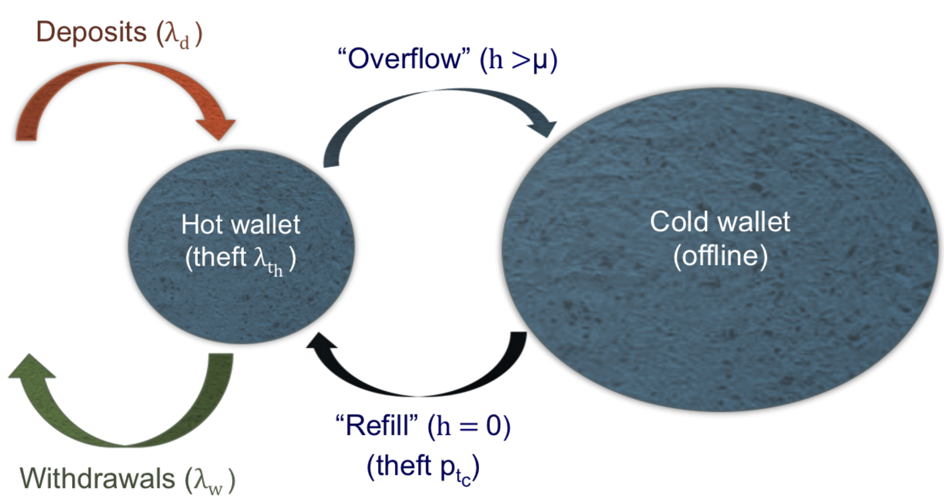
In contrast, a cold wallet consists of Bitcoin private keys stored on an offline device. Cold wallets often involve additional, physical barriers to access, and as such, are generally less vulnerable to outsiders, barring break-ins. In a company or organization handling Bitcoin reserves of high value, cold wallet access would likely be limited to cleared and trusted employees, with no one individual granted full privileges. Cold wallets may *need* to be accessed for a number of reasons, including for routine maintenance, to inspect and reinforce security systems, and, of particular importance to this study, to refill depleted hot wallets.

In particular, hot and cold wallets are vulnerable to theft in fundamentally distinct ways. A hot wallet on a computer perpetually connected to the Internet, a reasonable worst-case assumption, is *continually exposed*, even while the device is not in use; in contrast, a cold wallet is put at risk *on access*, as signing a transaction with a cold wallet's private key requires temporarily peeling back the layers of security encasing it. That the vulnerability of a cold wallet correlates with frequency of access is also suggested by the well-known security heuristic that the fidelity of a private key degrades with use.

3 Problem Formulation

Consider the servicing requirements of a Bitcoin exchange, which must accept or dispense bitcoin for fiat currency, or a banking service, which must allow customers to deposit and withdraw bitcoins at will from a common pool (the bank's fractional reserves). We can model deposits and withdrawals as *Poisson processes* with rate parameters λ_d and λ_w defined over a set time interval, such as hours. For example, $\lambda_d = 80$ corresponds to an average deposit rate of 80 bitcoins per hour.[2] We assume that our service accrues bitcoin on average ($\lambda_d > \lambda_w$), so that protecting accumulated customer reserves is a serious concern for the organization. (Note that in practice these parameters would be *empirically* determined, by extracting the relevant averages from transaction statistics.)

Figure 1: Problem Setup



In this study, we analyze the following simple two-wallet configuration (see Fig. 1). Our institution services deposits and withdrawals from a *hot wallet* that is continually connected to the Internet, and suffers theft, which empties the hot wallet, with Poisson rate parameter λ_{t_h}. For example, $\lambda_{t_h} = 0.002$ would

[2]We use 1 Bitcoin as the unit for the Poisson rate parameters, for conceptual clarity, but the choice is arbitrary. In theory, transactions of value as small as 1×10^{-8} BTC, or 1 satoshi, are possible, so our unit could just as well have been satoshis.

correspond to an expected time of 21 days between hot wallet thefts. The hot wallet is backed by a *cold wallet*, which contains the bulk of the organization's reserves and is accessed when necessary to refill the hot wallet. Transferring bitcoins to the hot wallet exposes the cold wallet to theft, as a stored private key must be invoked to sign the transaction. Since transfers are discrete events, cold wallet theft is assumed to occur with *probability* p_{t_c} on each access.

We consider a straightforward online algorithm in which whenever the hot wallet exceeds a threshold of μ bitcoins, a transaction $H \to C$ is made, "overflowing" the excess bitcoins into the cold wallet. Note that such a transfer does not expose the cold wallet to theft, as only the sender must provide a digital signature. However, since the hot wallet is ordinarily exposed to theft, we assume that such a transaction does not confer any additional risk. On the other hand, when the hot wallet is emptied, the cold wallet must immediately refill it to μ bitcoins through a $C \to H$ transaction.

This setup leads to an optimization problem that is the core focus of this paper. Given λ_d, λ_w, λ_{t_h}, and p_{t_c}, what value of μ maximizes the net balance in the organization's hot and cold wallets after some long time T? In particular, if μ is high, the organization will lose more to hot wallet thefts, as on average the hot wallet contains a number of bitcoins that varies (positively) with μ. But if μ is too low, then the cold wallet will need to be accessed more often, increasing the incidence of the much more damaging cold wallet thefts. Note that we assume that the interval $[0, T]$ is long enough for many hot wallet thefts and several cold wallet thefts to have occurred, so that the probability distribution of the net balance at T is a fair representation of the long-term performance of our algorithm.

4 Related Work

Previous work on mitigating losses due to Bitcoin theft has focused on designing protocols that make it more difficult for private keys to be divulged and misused. Though this study centers on optimizing the *arrangement* of existing architecture, as opposed to proposing new cryptography, we will discuss three developments in Bitcoin wallet security that form a crucial foundation for our work.

4.1 Multi-signature transactions

A *multi-signature transaction* is a transfer of Bitcoin involving an address "owned" by multiple parties; more specifically, an address associated with more than one private key. Multisig transactions are typically implemented with *m-of-n addresses*, a protocol in which signatures from m out of the n private keys associated with an address are required for a transaction to be enacted. The security benefits of such a scheme are readily apparent [15]. A 2-of-3 address, for instance, allows an individual Alice to keep private keys associated with a wallet over three separate devices [15]. Now, a malicious party cannot access Alice's bitcoins by simply hacking one of her machines. In the case that a single device *is* compromised, Alice can move her bitcoins to another safe address, by constructing a transaction with her 2 remaining keys.

A Bitcoin exchange may also find it useful to maintain m-of-n addresses, in order to regulate access to a hot wallet. Specifically, by using multisig, the exchange can stipulate that only group action can invoke transactions, without running the risk of locking itself out of its own wallet (i.e. through the loss of a single private key). With this setup, theft requires a collusion of multiple insiders, which is significantly more difficult for a malicious party to arrange than the compromise of a single point of security.

4.2 Threshold signatures

The second development, the *threshold signature* scheme, is a natural progression of multisig transactions. The main innovation of threshold signatures is that they allow several parties to demonstrate joint control over a Bitcoin wallet, without using multiple private keys [16]. In particular, a single private key is split and shared among the wallet owners, so that some t out of n pieces are required to construct a valid signature [16]. Notably, the cryptography involved ensures that possession of even $t-1$ pieces does not provide any partial information (i.e. a speedup in a brute force attack) [17]. Unlike multisig transactions, threshold signatures constitute client-side technology, as they are not built-in to the Bitcoin protocol.

The threshold signatures scheme has two primary benefits. Firstly, it preserves the pseudonymity of the signing parties, as only a single collective address is published with each transaction [16]. Secondly, it avoids restrictions inherent to Bitcoin scripts, such as limits on the number of participants allowed in multisig transactions [16]. Several threshold signature schemes for the ECDSA signature algorithm used by Bitcoin have been proposed; see Mackenzie and Reiter [18], Gennaro et al., [19], and Gennaro et al. [20].

4.3 Deterministic wallets

Lastly, we consider *deterministic wallets.* This is a Bitcoin wallet architecture in which all private keys are derivable from a single seed, through a one-way hash function [21]. While this may initially appear to be a security *hazard*, note that holding all else constant, a wallet with multiple, unrelated private keys is no more secure than a wallet with a single "super key"; in either case, a compromised wallet entails the loss of all its contents. The primary benefits of deterministic wallets are that they 1) ensure that creating wallet backups are easy, 2) allow lost keys to be recovered, and 3) naturally support the creation of new key/address pairs, a capability that may be required from a privacy standpoint.

Of particular relevance to this study are *hierarchical deterministic* (HD) *wallets*, support for which was implemented with the BIP 32 (Bitcoin Improvement Proposal 32) standard [21]. In this architecture, private keys are derived in a tree structure, with every key except for the master key (which is derived directly from the seed) linked to parent and child keys [21]. This structure, and the one-way nature of key generation, allows tree "branches" to be delegated to different departments or security systems in an organization, without putting parallel branches at risk of compromise [21].

Perhaps the most powerful capability afforded by HD wallets is the ability to separate the generation of new private keys from the creation of their associated addresses [22]. To understand the utility of this particular cryptographic innovation, consider an organization that must consistently transfer bitcoins from a hot wallet to a cold wallet, as in our setup. The organization may want to periodically create and use new cold wallet addresses, but doing so (traditionally) requires connecting the cold wallet to the Internet, generating public/private key pairs, and transferring the new batch of addresses to the hot wallet [22]. This, besides being inconvenient, involves repeatedly exposing the cold wallet to theft. Using HD wallets, however, it is possible for the hot wallet to send bitcoins to a new address that has *not yet been invoked*. Specifically, upon initialization, the hot wallet receives a seed for address generation, while the cold wallet is entrusted with the seed for private key generation [22]. This setup allows the hot wallet to generate a series of addresses with a one-to-one correspondence with private keys "known" only to the cold wallet [22]. The hot wallet can then send bitcoins at will to the kth address with the guarantee that when the cold wallet is accessed, it will be able to redeem all transacted bitcoins.

In our subsequent discussion on the optimal ceiling on hot wallet reserves, we will assume that these constructs are implemented wherever appropriate, as our analysis is consistent with, and builds on, the security guarantees these protocols provide. Notably, we will diverge from previous work on Bitcoin wallets, which has focused on theft *prevention*, by assuming a different line of inquiry: given that hot and cold wallets thefts are occurring consistently, what high-level wallet structures can we propose to minimize net losses over time? In analyzing system design, rather than security fundamentals, we will operate at a layer of abstraction that, in our opinion, has been neglected in the current literature, but is critical to the long run viability of Bitcoin exchanges and banking services.

5 Approach

We seek to determine an optimal threshold μ so as to minimize losses due to hot and cold wallet theft over $[0, T]$. Since deposits, withdrawals, and hot wallet thefts are Poisson processes, we will in general be dealing with probability distributions. In this preliminary, motivating analysis, however, we consider expected value.

If we let $B(\mu)$ represent the expected balance in the wallets at time T, then by linearity of expectation it is reasonable to expect

$$B(\mu) = Ex[D - W] - c_1\mu^{\alpha} - \frac{c_2}{\mu^{\beta}} \qquad\qquad \alpha, \beta > 0 \tag{1}$$

This equation requires some unpacking. By $Ex[D-W]$ we denote the expected value of net arrivals into the wallets, where D and W are random variables representing the total value of deposits and withdrawals, respectively, over $[0, T]$. The second term, $c_1\mu^{\alpha}$, represents expected losses due to hot wallet theft, which we anticipate to be positively correlated with the threshold μ (while the *number* of hot wallet thefts is not dependent on μ, the expected loss associated with a single theft *is*). Finally, $\frac{c_2}{\mu^{\beta}}$ represents expected losses due to cold wallet theft. We expect this term to be *negatively* correlated with μ since a higher threshold implies less frequent hot wallet refills. Note that we have suppressed the implicit dependence of c_1 and c_2 on T for the sake of clarity.

If hot wallet and cold wallet thefts are indeed positively and negatively correlated with μ, respectively, as we expect, then it should be possible to optimize B with respect to μ

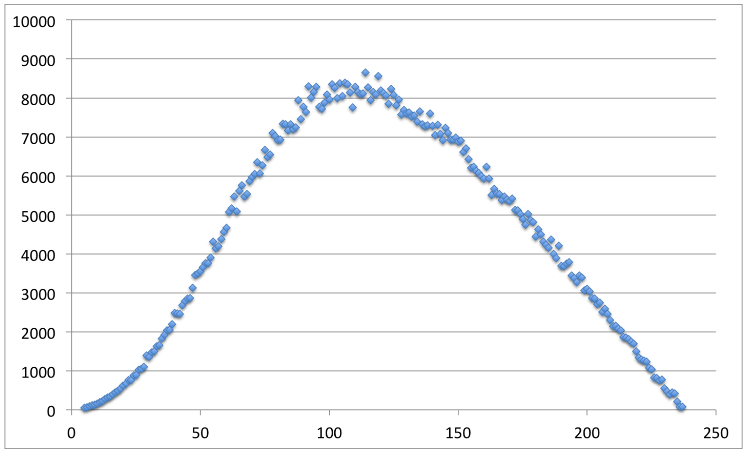
$$\frac{dB(\mu)}{d\mu} = -c_1\alpha\mu^{\alpha-1} + \frac{c_2\beta}{\mu^{\beta+1}} = 0 \tag{2}$$

$$\mu = \sqrt[\alpha+\beta]{\frac{c_2\beta}{c_1\alpha}} \tag{3}$$

That such an optimal threshold exists is also suggested by empirical results. To test our theoretical models, we developed an *event driven simulation* of the setup, which yielded experimental values for the net balance of the hot/cold wallet system after a fixed time T. In particular, we chose convenient sets of values for λ_d, λ_w, λ_{t_h}, and p_{t_c}, and drew pseudorandom numbers (i.e. java.util.Random) from the exponential distribution to generate waiting times to deposits, withdrawals, and hot wallet thefts. We set T to be 200 times the expected time to a hot wallet theft (and chose p_{t_c} so as to ensure that at least several cold wallet thefts would occur). We then tracked the balance of the hot and cold wallet over $[0, T]$, handling both external events (requests and thefts) and internal events (transfers) appropriately. This procedure was repeated for a range of values for μ, yielding the results seen in Fig. 2. Note that each data point (μ, B) represents an average over 1000 iterations of the simulation for $\lambda_d = 80$, $\lambda_w = 78$, $\lambda_{t_h} = 0.01$, and $p_{t_c} = 0.01$.

The graph (Fig. 2) clearly indicates that the net balance B peaks at a value of μ slightly over 110 (the absolute maximum occurs at $\mu = 114$), below and above which it falls to 0. These simulation results offer a

Figure 2: Net Balance $B(\mu)$ vs. Threshold μ.



preliminary confirmation of our hypothesis that losses due to cold wallet and hot wallet thefts are inversely related, and that an optimization problem in fact exists.

In this study, we develop the theory to precisely formulate this optimization problem, and seek an understanding of the probability mass function $P(B = k \mid \lambda_d, \lambda_w, \lambda_{t_h}, p_{t_c})$ describing the total balance at T (of which $B(\mu)$ is the expected value). In particular, we determine the explicit nature of the terms in Equation 1 describing deposits, withdrawals, and losses due to theft, as part of a complete model capable of predicting the optimal value of μ for any given values of λ_d, λ_w, λ_{t_h}, and p_{t_c}.

6 Results

6.1 Net Income

We begin by analyzing the Poisson processes that describe deposits and withdrawals. The Poisson distribution gives the probability that a random variable D denoting the number of deposits (or a random variable W denoting the number of withdrawals) takes on a particular value k. If λ_d and λ_w are the mean hourly deposit and withdrawal rates, then in any hour

$$P(D = k) = \frac{\lambda_d^k e^{-\lambda_d}}{k!} \tag{4}$$

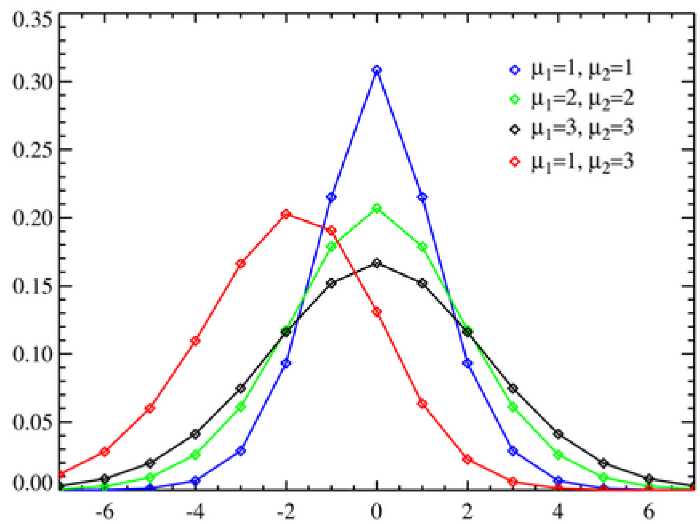
$$P(W = k) = \frac{\lambda_w^k e^{-\lambda_w}}{k!} \tag{5}$$

We now make an important claim: theft affects *accumulated wealth* in our wallets, so we care only about *net income* $I = D - W$. We will emphatically *not* need to reason about deposits and withdrawals as independent processes.

Unfortunately, net income $I = D - W$ is not a Poisson process, and its probability distribution cannot be modeled as such. A simple counterexample: $D - W$ can be less than 0, but Poisson processes describe only positive numbers of arrivals.

Instead, $D - W$ follows the *Skellam distribution*, a probability distribution that describes the difference of two Poisson random variables N_1 and N_2 with rate parameters λ_1 and λ_2 [23]. As suggested by intuition, the associated probability function attains its maximum (and expected) value at $\lambda_1 - \lambda_2$ (see Fig. 3).

Figure 3: Skellam Distribution [24]



Probability mass functions for various values of λ_1 and λ_2

In our case, the probability function describing net income $I = D - W$ is given by summing over all (d, w) pairs such that $d - w = k$. Since deposits and withdrawals are independent processes, the probability that $D = d$ and $W = w$, for any given pair (d, w), is just the product $P(D = d)P(D = w)$

$$P(D - W = k) = \begin{cases} \displaystyle\sum_{d=k}^{\infty} \frac{\lambda_d^d e^{-\lambda_d}}{d!} \frac{\lambda_w^{d-k} e^{-\lambda_w}}{(d-k)!} & k \geq 0 \\ \\ \displaystyle\sum_{w=-k}^{\infty} \frac{\lambda_d^{w+k} e^{-\lambda_d}}{(w+k)!} \frac{\lambda_w^{w} e^{-\lambda_w}}{w!} & k < 0 \end{cases} \tag{6}$$

$$= e^{-(\lambda_d + \lambda_w)} \sum_{d=\max[0,k]}^{\infty} \frac{\lambda_d^d}{d!} \frac{\lambda_w^{d-k}}{(d-k)!} \tag{7}$$

Though equation 7 is a valid formula for the net arrivals $D-W$, and suffices for our purposes, it is not the standard form of the Skellam probability density function. The standard representation has the advantage of being defined over all real numbers (as opposed to just the integers), so we will provide it as well. Given two Poisson random variables N_1 and N_2, the Skellam probability function is typically defined as [25]

$$P(N_1 - N_2 = k \mid \lambda_1, \lambda_2) = e^{-(\lambda_1+\lambda_2)} \left(\frac{\lambda_1}{\lambda_2}\right)^{k/2} I_{|k|}(2\sqrt{\lambda_1\lambda_2}) \tag{8}$$

where $I_\nu(z)$ is the modified Bessel function of the first kind [25]

$$I_\nu(z) = \left(\frac{z}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{\left(\frac{1}{4}z^2\right)^k}{k!\ \Gamma(v+k+1)} \tag{9}$$

Substituting in $I_{|k|}(2\sqrt{\lambda_1\lambda_2})$, the Skellam function describing net income $I = D - W$ in our problem is

$$P(D - W = k \mid \lambda_d, \lambda_w) = e^{-(\lambda_d+\lambda_w)} \left(\frac{\lambda_d}{\lambda_w}\right)^{k/2} \left(\sqrt{\lambda_d\lambda_w}\right)^{|k|} \sum_{j=0}^{\infty} \frac{(\lambda_d\lambda_w)^j}{j!\ \Gamma(|k|+j+1)} \tag{10}$$

It can be shown that for integer values of k this is in fact equivalent to 7.

We will now construct a series of models, incrementally introducing elements of the original problem to a preliminary setup consisting of only the hot wallet and excluding all thefts.

6.2 Model 1: Hot wallet only, unlimited capacity, no thefts

Let us suppose that the organization services all deposits and withdrawals from a hot wallet, but does not use a supporting cold wallet to secure excess bitcoins. Let us further assume (temporarily) that hot wallet theft is not a concern. A natural question arises: what is the probability mass function, $P(B(T) = k \mid \lambda_d, \lambda_w)$ of the net balance of the hot wallet after a long time T?

Since Poisson processes exhibit the properties of independent and stationary increments, the rate parameters λ_d and λ_w scale linearly with time. So we can replace λ_d and λ_w with $\lambda_d T$ and $\lambda_w T$ respectively in the probability mass function derived earlier

$$P(B(T) = k \mid \lambda_d, \lambda_w) = e^{-(\lambda_d+\lambda_w)T} \sum_{d=\max[0,k]}^{\infty} \frac{(\lambda_d T)^d}{d!} \frac{(\lambda_w T)^{d-k}}{(d-k)!} \tag{11}$$

In our subsequent analysis, we will use this function as a black box to describe net inflow into our hot wallet; specifically, we will use the term $PD_k(t)$ to represent the probability that a net arrival of k bitcoins occurs in time t, where PD represents the *Poisson difference*, or Skellam, function.

Note that it is possible for an empty hot wallet to encounter withdrawal requests, which, without a supporting cold wallet, it would not be able to fulfill. For now, however, we will allow the abstraction of a negative hot wallet balance.

6.3 Model 2: Hot wallet only, hot wallet theft with rate λ_{t_h}

We can ask the same question: given that hot wallet theft is occurring with rate λ_{t_h}, what is the probability function, $P(B(T) = k \mid \lambda_d, \lambda_w, \lambda_{t_h})$, of the net balance at T?

We begin by noting that the time between hot wallet thefts is given by the exponential distribution.

$$P(\text{next theft at time } t) = \lambda_{t_h} e^{-\lambda_{t_h} t} \tag{12}$$

Here we make the important observation that a hot wallet theft *resets* the state of our system, leaving the hot wallet with 0 bitcoins, as at the start. Then all bitcoins in the hot wallet at T are due to deposits and withdrawals since the *time of last theft*.

To determine the probability function describing the time of last theft, we use the fact that Poisson processes are *memoryless*. This is the surprising property of the exponential distribution that asserts that the waiting time t to the next arrival (theft) is not dependent on prior history, namely the time s we've already waited. Formally, if X is a random variable denoting the time to the next hot wallet theft [26]

$$\begin{aligned}
P(X > s + t \mid X > s) &= \frac{P\left(X > s + t \bigcap X > s\right)}{P(X > s)} \\
&= \frac{P(X > s + t)}{P(X > s)} \\
&= \frac{e^{-\lambda_{t_h}(s+t)}}{e^{-\lambda_{t_h} s}} \\
&= e^{-\lambda_{t_h} t}
\end{aligned} \tag{13}$$

Clearly the survival function (the probability that no theft occurs *before* time $s + t$) has no dependence on s.

We now claim that the probability that the last theft occurs at time $T - t$ is simply

$$P(\text{last theft at time } T - t) = \lambda_{t_h} e^{-\lambda_{t_h} t} \tag{14}$$

Proof. The last theft on $[0, T]$ is the *first* theft on $[T, 0]$. The waiting time t to the first theft on $[T, 0]$ has no dependence on "prior" thefts (i.e. the time s from the "previous" theft). Let X denote the time to the first theft on $[T, 0]$. Then

$$\begin{aligned}
P(X = s + t) &= \frac{d}{dt} P(X \leq s + t) \\
&= \frac{d}{dt}(1 - P(X > s + t)) \\
&= -\frac{d}{dt} e^{-\lambda_{t_h}(t+s)} \\
&= \lambda_{t_h} e^{-\lambda_{t_h}(t+s)}
\end{aligned} \tag{15}$$

It follows that

$$\begin{aligned}
P(X = s + t \mid X > s) &= \frac{P(X = s + t)}{P(X > s)} \\
&= \frac{\lambda_{t_h} e^{-\lambda_{t_h}(t+s)}}{e^{-\lambda_{t_h} s}} \\
&= \lambda_{t_h} e^{-\lambda_{t_h} t}
\end{aligned}$$

□

Now, the probability density function for the hot wallet balance at T is given by an integral over possible times of last theft $T - t$, plus a term for the (rare) case in which *no* hot wallet theft occurs in $[0, T]$. For each possible theft time, the balance at T is determined by the number of net arrivals between $T - t$ and T.

$$P(B(T) = k \mid \lambda_d, \lambda_w, \lambda_{t_h}) = \int_0^T \left(\lambda_{t_h} e^{-\lambda_{t_h} t}\right) PD_k(t)\ dt\ + e^{-\lambda_{t_h} T} PD_k(T) \tag{16}$$

Note that the integrand is the probability that two independent events take place: 1) the last hot wallet theft occurs at time $T - t$ and 2) k net arrivals occur in time t.

Equation 16 is a powerful result. The probability density function $P(B(T))$ provides a complete picture of the performance of a single hot wallet supporting deposits and withdrawals, and subject to recurring thefts. In the full theory that we now develop, we will borrow from this model the critical idea that thefts reset the state of our system.

6.4 Model 3: Hot and cold wallets

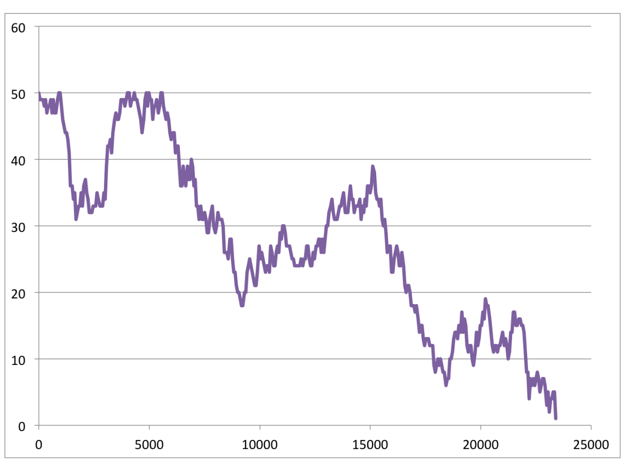
In this section, we consider the complex problem of a dual wallet system. We begin by reintroducing the following attributes of the original setup: 1) if the hot wallet reaches a threshold of μ bitcoins, we overflow the excess currency into the cold wallet, 2) if the hot wallet is emptied (by theft or by ordinary depletion), we move μ bitcoins from the cold to the hot wallet, and 3) refilling the hot wallet results in cold wallet theft with probability p_{t_c}. We seek a closed form expression for the final balance at T, so that we can optimize this quantity with respect to the threshold μ.

A first approach may be to analyze the balances of the hot and cold wallets separately, keeping track of the $C \to H$ and $H \to C$ transfers. These transfer times, however, are determined by continuous probabilistic processes (deposits, withdrawals, hot wallet theft) constrained by the "artifical" boundaries 0 and μ. As a result, the probability function describing the time of the kth transfer is *dependent* on the probability functions of the previous transfer times.

An alternative strategy is to disregard (most) interactions *between* the wallets, and instead consider the three global processes that determine the final balance: net arrivals (deposits minus withdrawals), losses due to hot wallet theft, and losses due to cold wallet theft. Note that the first two processes have no dependence on the *state of the hot wallet*, unlike transfer times, and are thus straightforward to model. We have thus isolated the complexity of our problem to the third process.

Quantifying cold wallet theft requires us to consider the frequency of $C \to H$ transfers, which occur whenever the hot wallet contains 0 bitcoins. If we model the hot wallet balance H as a *continuous time random walk*, then the expected time to reach $H = 0$ (empty) from the starting point $H = \mu$ (full) is precisely the expected time to a $C \to H$ transfer (see Fig. 4).

Figure 4: Hot Wallet Balance vs. Time (seconds)



Parameter values: $\mu = 50$; $\lambda_d = 79$, $\lambda_w = 78$, $\lambda_{t_h} = 0.001$

Formally, we wish to find X_μ, given that X_k represents the expected time to empty from $H = k$. We can write a recurrence relation for X_k by considering the state of the system after a small time interval t. One of four events can happen in t:

Event	State Transition	Probability
Deposit	$X_k \to X_{k+1}$	$\lambda_d t$
Withdrawal	$X_k \to X_{k-1}$	$\lambda_w t$
Hot Wallet Theft	$X_k \to X_0$	$\lambda_{t_h} t$
No Event	$X_k \to X_k$	$1 - (\lambda_d + \lambda_w + \lambda_{t_h})t$

Note that we are making the first-order approximation that only *one* event can occur in t. This is valid in the limit $t \to 0$. Then the following recurrence must hold

$$X_k = t + (\lambda_d t)X_{k+1} + (\lambda_w t)X_{k-1} + (\lambda_{t_h} t)X_0 + (1 - (\lambda_d + \lambda_w + \lambda_{t_h})t)X_k \tag{17}$$

subject to the boundary conditions

$$X_0 = 0 \tag{18}$$

$$X_\mu = t + (\lambda_w t)X_{\mu-1} + (\lambda_{t_h} t)X_0 + (1 - (\lambda_w + \lambda_{t_h})t)X_\mu \tag{19}$$

Combining like terms and dividing through by the time parameter t, we rewrite equation 17 as follows

$$((\lambda_d + \lambda_w + \lambda_{t_h})t)X_k = t + (\lambda_d t)X_{k+1} + (\lambda_w t)X_{k-1} + (\lambda_{t_h} t)X_0$$

$$0 = \lambda_d X_{k+1} - (\lambda_d + \lambda_w + \lambda_{t_h})X_k + \lambda_w X_{k-1} + 1 \tag{20}$$

Note that this is a non-homogeneous, second order recurrence relation. Its general solution is a linear combination of homogeneous and particular solutions. In this case, the particular solution is a constant

$$X_k = \frac{1}{\lambda_{t_h}} \tag{21}$$

while the homogeneous solutions are roots of the characteristic polynomial

$$\lambda_d x^2 - (\lambda_d + \lambda_w + \lambda_{t_h})x + \lambda_w = 0 \tag{22}$$

namely

$$x = \frac{(\lambda_d + \lambda_w + \lambda_{t_h}) \pm \sqrt{(\lambda_d + \lambda_w + \lambda_{t_h})^2 - 4\lambda_d\lambda_w}}{2\lambda_d} \tag{23}$$

The general solution to equation 20 is then

$$X_k = \frac{1}{\lambda_{t_h}} + a_1(x_1)^k + a_2(x_2)^k \tag{24}$$

To find the constants a_1 and a_2 we impose the boundary conditions. We first rewrite condition 19 as

$$(\lambda_w + \lambda_{t_h})X_\mu = 1 + \lambda_w X_{\mu-1} \tag{25}$$

Then substituting our general solution into 18 and 19 yields

$$X_0 = \frac{1}{\lambda_{t_h}} + a_1 + a_2 = 0 \tag{26}$$

$$(\lambda_w + \lambda_{t_h})\left(\frac{1}{\lambda_{t_h}} + a_1(x_1)^\mu + a_2(x_2)^\mu\right) = 1 + \lambda_w\left(\frac{1}{\lambda_{t_h}} + a_1(x_1)^{\mu-1} + a_2(x_2)^{\mu-1}\right) \tag{27}$$

Solving this system for a_1 and a_2

$$a_1 = \frac{\frac{1}{\lambda_{t_h}}\left[\lambda_w(x_2^\mu - x_2^{\mu-1}) + \lambda_{t_h}x_2^\mu\right]}{\left[\lambda_w(x_1^\mu - x_1^{\mu-1}) + \lambda_{t_h}x_1^\mu\right] - \left[\lambda_w(x_2^\mu - x_2^{\mu-1}) + \lambda_{t_h}x_2^\mu\right]} \tag{28}$$

$$a_2 = \frac{-\frac{1}{\lambda_{t_h}}\left[\lambda_w(x_1^\mu - x_1^{\mu-1}) + \lambda_{t_h}x_1^\mu\right]}{\left[\lambda_w(x_1^\mu - x_1^{\mu-1}) + \lambda_{t_h}x_1^\mu\right] - \left[\lambda_w(x_2^\mu - x_2^{\mu-1}) + \lambda_{t_h}x_2^\mu\right]} \tag{29}$$

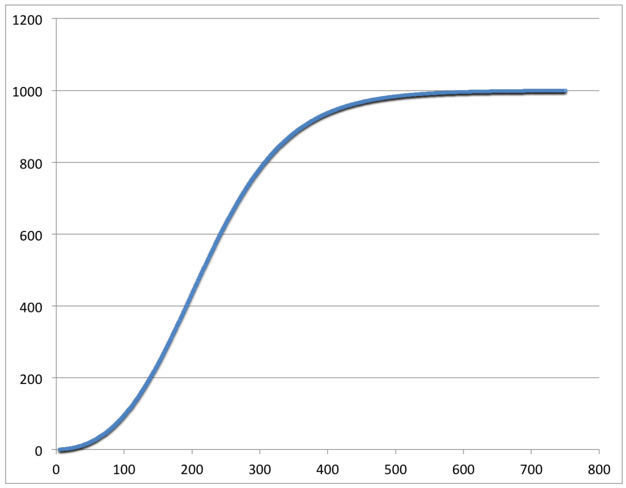
Substituting a_1 and a_2 into our general solution, letting $k = \mu$, and simplifying

$$X_\mu = \frac{1}{\lambda_{t_h}} + \frac{1}{\lambda_{t_h}}\left(\frac{\lambda_w(x_2 - x_1)(x_1x_2)^{\mu-1}}{\left[\lambda_w(x_1 - 1) + \lambda_{t_h}x_1\right]x_1^{\mu-1} - \left[\lambda_w(x_2 - 1) + \lambda_{t_h}x_2\right]x_2^{\mu-1}}\right) \tag{30}$$

This, finally, is the closed form expression for the expected time to an empty hot wallet (and thus to a

$C \to H$ transfer). Notably, X_μ plotted as a function of μ exhibits the properties of a *logistic equation.* (The resemblance is clearer if the numerator and denominator are divided by $(x_1 x_2)^{\mu-1}$.)

Figure 5: Expected Time To Empty X_μ vs. Threshold μ (Equation 30)



Parameter values: $\lambda_d = 80$, $\lambda_w = 78$, $\lambda_{t_h} = 0.001$

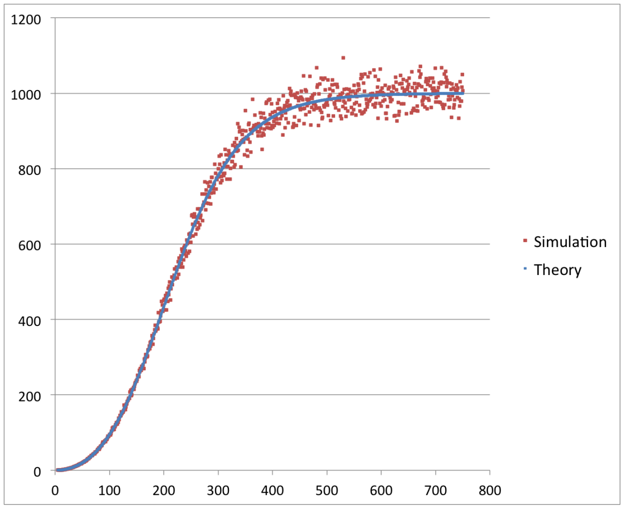
In particular, the function grows rapidly in a region $\mu_1 < \mu < \mu_2$, and then flattens out, approaching an asymptote of $X_\mu = \frac{1}{\lambda_{t_h}}$ (see Fig. 5). This behavior is precisely what intuition would suggest. Given our assumption that $\lambda_d > \lambda_w$, as μ becomes larger, it becomes unlikelier that net withdrawals, even over a period of atypical activity, can empty the hot wallet. Then the dominating factor driving H to 0 becomes hot wallet theft, which empties the hot wallet on expectation every $\frac{1}{\lambda_{t_h}}$ hours. This is the bound that X_μ tends to as μ approaches infinity.

Values for X_μ predicted by this model determine an accurate trend line for the empirical results yielded by an event driven simulation of the hot wallet balance (see Fig. 6). As in the previous simulation, waiting times to the next deposit, withdrawal, and hot wallet theft are computed by selecting pseudorandom numbers from the exponential distribution. Each simulation data point (μ, X_μ) corresponds to the mean time to empty over 1000 iterations of the continuous time random walk.

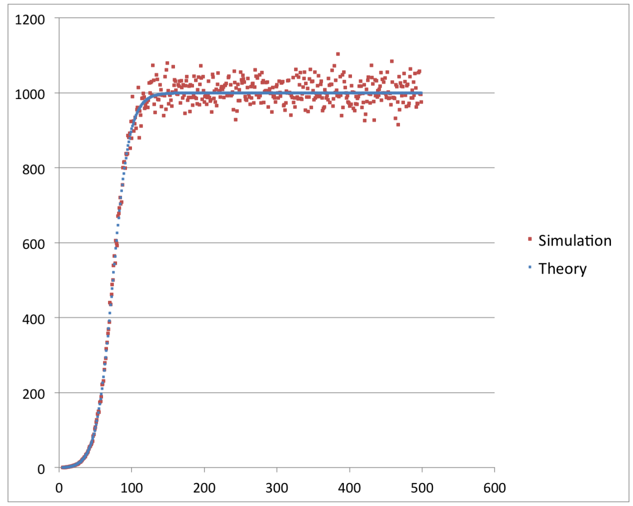
While the simulation results exhibit greater variance for larger values of μ, Fig. 6 clearly indicates that they are clustered around or on the theoretical values over the whole domain. Note also that for the larger value of $\lambda_d - \lambda_w$ (Fig. 6b), X_μ reaches its asymptote faster. This trend, too, is in line with intuition. When deposits far exceed withdrawals, it becomes unlikelier that the hot wallet can be emptied by aberrations in net arrivals. As a consequence, hot wallet theft becomes the dominating factor *earlier.*

Figure 6: Expected Time To Empty X_μ

Theory (Equation 30) vs. Event Driven Simulation



(a) $\lambda_d = 79$, $\lambda_w = 78$, $\lambda_{t_h} = 0.001$



(b) $\lambda_d = 85$, $\lambda_w = 78$, $\lambda_{t_h} = 0.001$

6.5 The Expected Balance

We are now in a position to present our culminating result - an expression for the expected net balance at time T. The key idea we invoke is that cold wallet thefts *reset our system*, in a manner analogous to hot wallet theft in Model 2. In particular, cold wallet thefts only occur after $C \to H$ transfers; $C \to H$ transfers, in turn, only occur if the hot wallet is empty. Thus, after a cold wallet theft, *both* wallets contain 0 bitcoins, which is precisely the state of the system at $T = 0$.

It follows that all bitcoins in the wallets at T accumulated since the time of *last* cold wallet theft t'. The expected time to this last theft (looking "backward" from T) is just the product of the expected time to a $C \to H$ transfer, X_μ and the expected number of transfers before a theft occurs $\frac{1}{p_{t_c}}$, or $\frac{X_\mu}{p_{t_c}}$.

To determine the net balance after t', we must account for net income $D - W$ and expected losses to hot wallet theft in $[t', T]$. In particular, by noticing that the system resets with cold wallet theft, we have rendered *losses* due to cold wallet theft irrelevant to our calculation.

6.5.1 Net Income

Net income is given by the Skellam (Poisson difference) function $PD_k(t)$ from Section 6.1, which describes the probability of k net arrivals in time t. The expected value of $PD_k(t)$ is, as intuition suggests, simply $(\lambda_d - \lambda_w)t$. Then the expected net income in $[t', T]$ is $(\lambda_d - \lambda_w)\frac{X_\mu}{p_{t_c}}$.

6.5.2 Hot Wallet Theft Losses

The expected losses due to hot wallet theft is a product of (1) the expected loss due to a single hot wallet theft and (2) the expected *number* of hot wallet thefts in $[t', T]$.

Quantity (1) is the expected value of the hot wallet balance just before hot wallet theft occurs, a value we know little about. We can expect, however, that it will be some fraction γ of the hot wallet threshold μ. Then (1) is just $\gamma\mu$, where $0 < \gamma < 1$.

Quantity (2) is the expected number of arrivals in a Poisson process. This is simply the rate parameter λ_{t_h} scaled over the given time interval, or $\lambda_{t_h} \frac{X_\mu}{p_{t_c}}$.

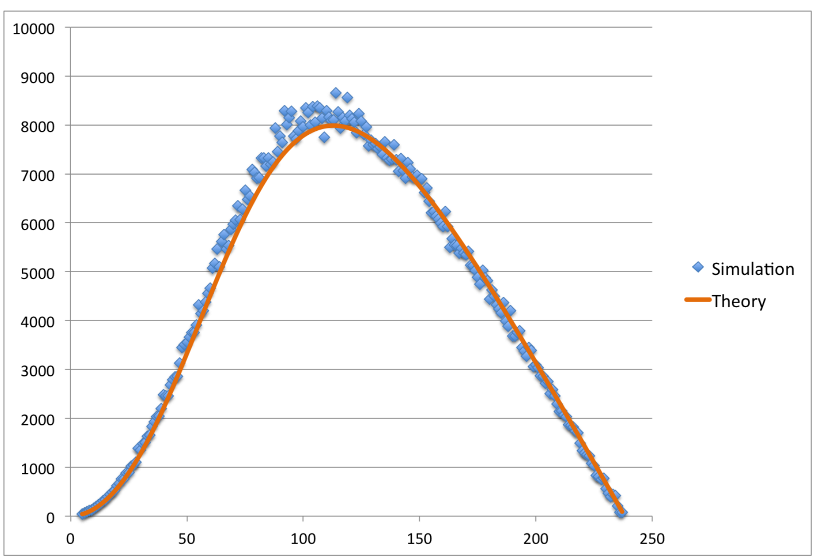
Combining these results, the expected next balance at T is

$$B = (\lambda_d - \lambda_w)\frac{X_\mu}{p_{t_c}} - (\gamma\mu)\left(\lambda_{t_h}\frac{X_\mu}{p_{t_c}}\right) \tag{31}$$

We can compare the values for $B(\mu)$ predicted by this formula with those produced by our computer model, first described in Section 5 to motivate this analysis, which simulates the behavior of the dual wallet system over a long time interval $[0, T]$. A strong match in the theoretical and empirical results is evident when $\gamma = 0.84$ in Equation 31 above (see Fig. 7).

Figure 7: Net Balance $B(\mu)$

Theory (Equation 31) vs. Event Driven Simulation



Parameter values: $\lambda_d = 80$, $\lambda_w = 78$, $\lambda_{t_h} = 0.01$, $p_{t_c} = 0.01$

Notably, equation 31 allows us to numerically determine the value of the optimal threshold for a given set of parameters. To illustrate this, we provide the precise data points yielded by our theoretical model for $109 \leq \mu \leq 115$ (see Table 1)

Threshold (μ)	Balance
109	7970.1143
110	7978.1031
111	7983.6853
112	7986.8970
113	7987.7753
114	7986.3581
115	7982.6842

Table 1: Balance vs. Threshold (Equation 31)

Note that the balance reaches a maximum of 7987.78 bitcoins at $\mu = 113$. This predicted optimal threshold differs by less than 1% from the empirical maximum of $\mu = 114$ and by less than 2% from the local maxima of a polynomial interpolation of the simulation data, $\mu = 111.05$.

6.6 Further Work

In an immediate follow up to this study, we plan to address two concerns regarding Equation 31. Firstly, we hope to establish that the time of last cold wallet theft and the magnitude of net arrivals after that time are indeed independent, an assumption that underlies the first term in Equation 31. This independence was apparent in the isolated hot wallet model, as hot wallet theft, deposits, and withdrawals are physically distinct Poisson processes, but remains to be proved for the dual wallet system.

Secondly, we hope to theoretically determine γ, the expected balance of the hot wallet over instances of hot wallet theft. In this study, γ was extrapolated from simulation results. Ideally, however, we would like Equation 31 to be a function of $\lambda_d, \lambda_w, \lambda_{t_h}$, and p_{t_c} alone. We are optimistic that further analysis of the random walk governing the hot wallet balance may yield a closed form expression for γ in terms of our fundamental parameters.

7 Applications and Extensions

7.1 Calibrated Threshold

A real world Bitcoin exchange or banking service may observe that customer deposit and withdrawal requests exhibit predictable trends. For example, the volume of Bitcoin transactions may peak at certain times of the day (e.g. after the opening of the New York Stock Exchange), shadow the price of the dollar, or demonstrate periodicity. In fact, there is strong evidence that Bitcoin transaction rates exhibit weekly and daily cycles, with troughs in transaction volume seen on Saturdays and Sundays, and daily peaks observed between 16 and 22 UTC, the time of day in which exchanges on the U.S. East Coast and Western Europe are active [27].

A second category of fluctuations to which an organization may be able to respond are those triggered by major events in the Bitcoin ecosystem. Deposits may plummet in the weeks following the shutdown of a major exchange, as seen after Mt. Gox, or skyrocket in the wake of a cyberattack targeting personal computers. An organization may also wish to respond to internal events, such as an increased incidence of hot wallet theft or heightened cold wallet security. In circumstances in which recent history can be used to make viable predictions, and in which customer behavior fluctuates significantly, a *calibrated threshold scheme* may prove particularly useful.

The scheme is a straightforward application of the main idea of this study. An organization maintains a *history block* that contains a record of 1) hourly deposits, 2) hourly withdrawals, 3) $C \to H$ transfers, 4) hot wallet thefts, and 5) cold wallet thefts for the past k hours. The history block is organized as 5 parallel, time-indexed arrays of length k, and is updated cyclically so that a new record overwrites one created k hours ago. This data is then used to recompute λ_d, λ_w, λ_{t_h}, and p_{t_c}, which are simply hourly rates, and thus update the capacity of the hot wallet each hour. Such a scheme would allow a company to maintain a threshold on online reserves that is optimal, given recent history. A hybrid approach which assigns greater weight to more recent $(\lambda_d, \lambda_w, \lambda_{t_h}, p_{t_c})$ tuples, but includes all of an organization's data, is also clearly feasible, and would yield results that reflect both macroscopic trends *and* recent history.

7.2 Multiple Wallet Systems

In this section, we consider the broader goal of an optimal online algorithm and storage scheme for servicing requests and holding Bitcoin reserves. In doing so, we are motivated by two main ideas. Firstly, we note that the approach presented in this study, which centers on probabilistic analysis of net outcomes, allows us to *compare* the performance of different servicing algorithms. This raises the natural question of what alternative systems are possible. Secondly, we seek to address a major shortcoming of the two wallet model: refilling the hot wallet endangers the bulk of our organization's reserves, even though it requires only a fraction of the bitcoins in the cold wallet.

7.3 "Retirement Fund" Wallets

Our first proposal involves tracking excess bitcoins (deposits into a hot wallet holding μ bitcoins) into one of *two* cold wallets. In particular, a large fraction k of the overflow is transferred into a "savings account," a cold wallet that holds the majority of the organization's reserves; the remainder is deposited in a "checking account," a cold wallet responsible for refilling the hot wallet when needed. Note that it is possible that the checking account may itself need to be refilled; in this case, the savings account must reimburse it, and we once again are faced with our old problem. The system is still an improvement over the two wallet model, however, as it *reduces* the frequency with which large holdings of Bitcoin are accessed.

A quantitative analysis of this algorithm must determine three parameters: the hot wallet threshold, μ_h, the checking account threshold, μ_c, and the fractions k and $1-k$ of excess bitcoins that are tracked into the savings and checking account, respectively.

7.4 Pyramid Model

The *pyramid model* is a natural progression of the "retirement fund" idea just proposed. The structure involves a single hot wallet W_1 and a *series* of cold wallets W_2 through W_n. Each wallet W_k overflows into W_{k+1}, when W_k exceeds a threshold μ_k, and is responsible for replenishing wallet W_{k-1}, when W_{k-1} is emptied. We claim that wallets closer to the hot wallet (i.e. near the "top" of the pyramid) are accessed at least as frequently as those farther removed from the hot wallet (at the "bottom"). (In practice, wallets at the top should be accessed *much* more frequently.)

Proof. Suppose that the hot wallet W_1 must be replenished r times in a time period $[0, T]$. Then W_2 is accessed r times, and will need to be replenished $r' \leq r$ times, as it loses bitcoins on only those r occasions. The claim follows by induction on k. Note that if we further condition that $\mu_{k+1} > \mu_k$ (a reasonable assumption), then a strict inequality $r' < r$ holds. $\square$

It follows that each successive wallet W_{k+1} should hold more bitcoins than W_k. The optimal value μ_k for each wallet remains an open question for a follow-up study. Note that if wallet W_k holds on average $c2^k$ bitcoins, half of the organization's bitcoins are, on expectation, in the bottom cold wallet, and almost 90% in the bottom three. Thus the pyramid model may indeed successfully address our motivating concern, by divorcing the servicing of requests (i.e. refilling the hot wallet) from the maintenance of reserves.

A disclaimer: in proposing both the retirement fund and pyramid models, we assume that it is possible to diversify access control to some extent. In particular, if an organization can only support k independent security systems (i.e. trusted individuals, secure safes, etc.), then it confers no additional benefit to maintain more than k wallets, as the compromise of one system endagers all wallets entrusted to it. This is a very necessary physical constraint to impose; without it, the most secure model would potentially involve sending each deposit to a distinct Bitcoin wallet.

8 Conclusion

In this paper, we proposed an equation for the expected balance of a hot and cold wallet system over a period of indeterminate length, given empirically determined Poisson parameters describing deposits, withdrawals, and hot wallet theft. This equation yielded an optimal value for the hot wallet threshold that fell within 2% of simulation results, thus resolving the motivating question of this study.

For particular subsystems, such as the single hot wallet, we were able to provide a complete characterization, namely a probability distribution on the net balance. For other systems, including the continuous time random walk and the final dual wallet structure, our theoretical models effectively yielded the trend lines around which our empirical results were centered.

We ended with a discussion of multiple wallet systems, in particular a “pyramid wallet” model in which an organization employs several *layers* of offline storage. We are optimistic that our analysis of the dual wallet system may apply to each pair of wallets in this structure, yielding results for the optimal threshold at each pyramid level. This remains an open question for a subsequent study.

With this paper, we hope to open discussion on an aspect of Bitcoin security that has received little coverage until now; specifically, the design of higher-level Bitcoin wallet systems. Our work addresses a fundamental question regarding online and offline storage of digital currency, and has the potential to influence the design of real world systems built to service, and safeguard the savings of, Bitcoin users.

9 Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1148900.

References

[1] Bitcoin Charts, http://www.bitcoincharts.com/markets/, 2015.

[2] S. Higgins, "Bitstamp Claims $5 Million Lost in Hot Wallet Hack," http://www.coindesk.com/bitstamp-claims-roughly-19000-btc-lost-hot-wallet-hack/, 2015.

[3] Coinbase, "BitCoin Vault - Coinbase," https://www.coinbase.com/vault, 2015.

[4] Kaspersky Labs, "Financial cyber threats in 2013. Part 2: malware," http://securelist.com/analysis/kaspersky-security-bulletin/59414/financial-cyber-threats-in-2013-part-2-malware/, 2013.

[5] Kaspersky Labs, "Kaspersky Lab Report: Financial cyberthreats in 2014," https://securelist.com/files/2015/02/KSN_Financial_Threats_Report_2014_eng.pdf, 2014.

[6] N. Hajdarbegovic, "Report: Bitcoin Targeted in 22% of Financial Malware Attacks," http://www.coindesk.com/report-bitcoin-targeted-22-financial-malware-attacks/, 2014.

[7] R. Brandom, "A string of thefts hit Bitcoin's most reputable wallet service," The Verge, http://www.theverge.com/2014/2/7/5386222/a-string-of-thefts-hit-coinbase-bitcoins-most-reputable-wallet-service, 2014.

[8] Bitcoin Forum, "List of Bitcoin Heists," https://bitcointalk.org/index.php?topic=576337, 2014.

[9] S. Eskandari, D. Barrera, E. Stobert, and J. Clark, "A First Look at the Usability of Bitcoin Key Management," http://www.internetsociety.org/sites/default/files/05_3_3.pdf, 2015.

[10] Bitcoin Wiki, "Cold Storage," https://en.bitcoin.it/wiki/Cold_storage, 2016.

[11] Coinbase, "Secure Bitcoin Storage - Coinbase," https://www.coinbase.com/security, 2016.

[12] Bitfinex, "Bitfinex - Our Security Practices," https://www.bitfinex.com/pages/security, 2016.

[13] T. Moore and N. Christin, "Beware the Middleman: Empirical Analysis of Bitcoin-Exchange Risk," in *Proceedings of Financial Cryptography, 17th International Conference, FC 2013*, Springer, 2013, pp. 25-33.

[14] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 conference on Internet measurement conference*, ACM, 2013, pp. 127–140.

[15] M. Rosenfeld, "What are multi-signature transactions?" http://bitcoin.stackexchange.com/questions/3718/what-are-multi-signature-transactions, 2012.

[16] S. Goldfeder, "New research: Better wallet security for Bitcoin," Freedom To Tinker, https://freedom-to-tinker.com/blog/stevenag/new-research-better-wallet-security-for-bitcoin/, 2014.

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

[18] P. MacKenzie and M. K. Reiter, "Two-party generation of dsa signatures," in *Advances in Cryptology—CRYPTO 2001*, Springer, 2001, pp. 137–154.

[19] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold dss signatures," in *Advances in Cryptology—EUROCRYPT'96*, Springer, 1996, pp. 354–371.

[20] R. Gennaro, S. Goldfeder, and A. Narayanan, "Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security," in *Applied Cryptography and Network Security*, Springer, 2016.

[21] A. M. Antonopoulos, "Wallets," in *Mastering Bitcoin*, O'Reilly Ebooks, http://chimera.labs.oreilly.com/books/1234000001802/ch04.html#_wallets, 2015.

[22] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, "Chapter 4: How To Store and Use Bitcoins," in *Bitcoin and Cryptocurrency Technologies*, Princeton University, 2015.

[23] J. G. Skellam, "The Frequency Distribution of the Difference Between Two Poisson Variates Belonging to Different Populations," in *Journal of the Royal Statistical Society*, Vol. 109, No. 3 (1946), pp. 296.

[24] "Skellam Distribution," https://upload.wikimedia.org/wikipedia/commons/thumb/b/b2/SkellamDistribution.png/440px-SkellamDistribution.png, 2013.

[25] D. Karlis and I. Ntzoufras, "Bayesian modelling of football outcomes: using the Skellam's distribution for the goal difference," in *IMA J Management Math*, Vol. 20, No. 2 (2009), pp. 133-145.

[26] D. Myers, "CS 547 Lecture 9: Conditional Probabilities and the Memoryless Property," University of Wisconsin-Madison, 2012.

[27] O. Ofcorti, "Daily and weekly bitcoin transaction cycles," http://organofcorti.blogspot.com/2014/11/daily-and-weekly-bitcoin-transaction.html, 2014.