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

Samvit Jain, Class of 2017 Advisor: Edward Felten

Motivation

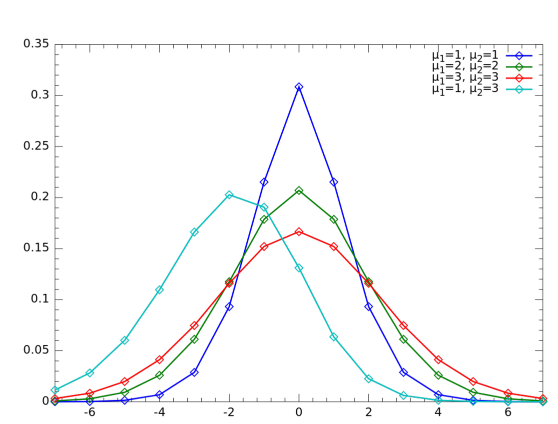
- Bitcoin theft is alarmingly pervasive
  - Impacts both major businesses (i.e. Bitcoin exchanges and wallet services) and individuals
  - 3.4 million instances of Bitcoin malware detected in 2014, 22% of all financial malware (Kaspersky Labs)
- What is being stolen?
  - *Private keys* – used to construct (authorize) transaction from A → B
  - Bitcoin ownership established through public key cryptography
- Mechanisms of theft
  - Malicious smartphone applications
  - Fraudulent Bitcoin management services
  - 150 strains of Bitcoin malware
    - Steal private keys stored on (Internet-connected) device
    - Steal login credentials to online wallet services
  - (Businesses) External attackers
    - Exploit vulnerabilities in client-facing software
    - Hack servers or databases
  - (Businesses) Insiders with access privileges

A Series of Models

- Methodology – probabilistic reasoning about net effect of memoryless processes (deposits, withdrawals, hot wallet theft) and discrete events (C → H transfers, cold wallet theft)
- To find net balance at T, developed theory (i.e. probability density functions) characterizing various subsystems of dual wallet structure

1) Net income *D – W* into the exchange
   - Skellam (Poisson Diff.) distribution
2) Hot wallet only, no theft
   - Q: Probability distribution at T?

$$P(D-W=k) = e^{-T(\lambda_d+\lambda_w)} \sum_{d=\max\{0,k\}}^{\infty} \frac{(\lambda_d T)^d}{d!}\frac{(\lambda_w T)^{d-k}}{(d-k)!}$$

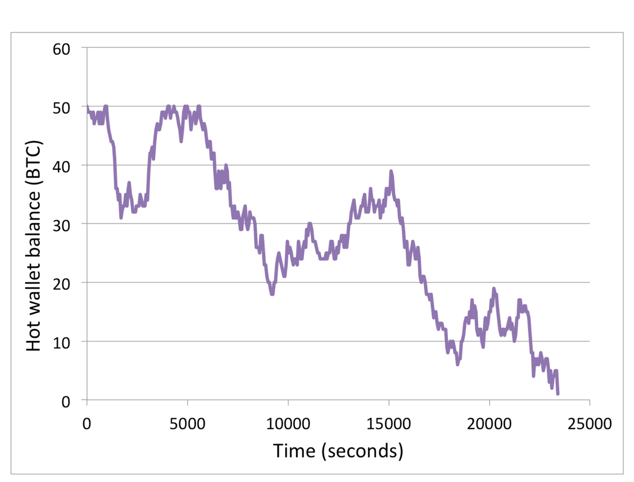


Skellam PMF for various values of λ_1, λ_2

3) Hot wallet only, theft λ_{th}
   - Thefts reset state of system, so only time of last theft matters
   - Poisson processes are *memoryless*

$$P(H_{bal}(T)=k \mid \lambda_d,\lambda_w,\lambda_{t_h}) = \int_0^T (\lambda_{t_h} e^{-\lambda_{t_h} t})\, PD_k(t)\, dt + e^{-\lambda_{t_h}T} PD_k$$

4) Hot wallet *and* cold wallet
   - C → H transfers occur after hot wallet is emptied
   - Want to know expected time to empty hot wallet, X_μ
   - Key idea: hot wallet balance as *continuous time random walk*
   - Can write recurrence relation with boundary conditions

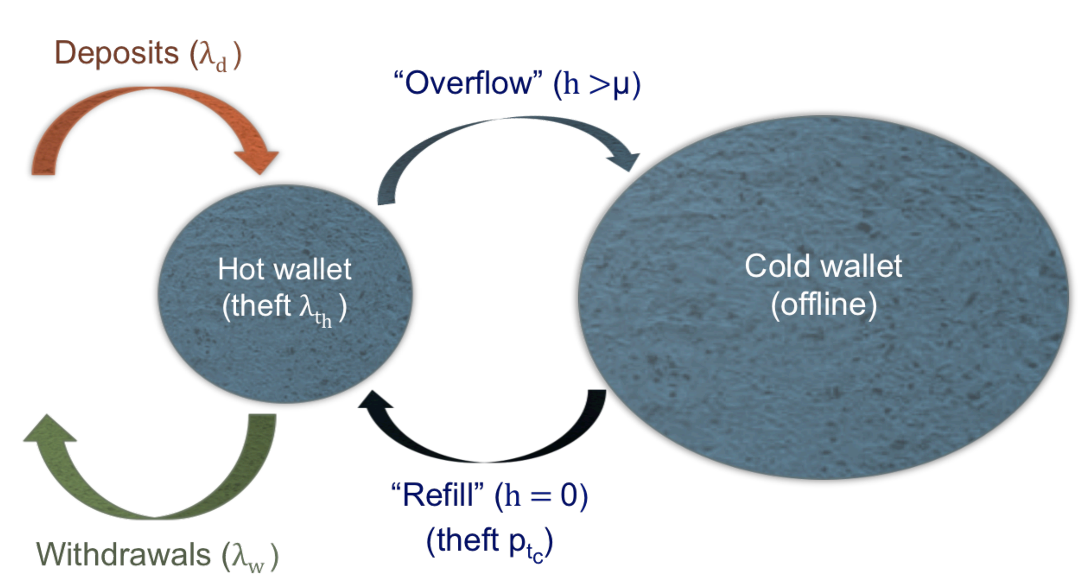


Hot Wallet Balance vs. Time for μ = 50

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

Problem Formulation

- Storage schemes
  - Online storage (hot wallet) – i.e., encrypted file on computer, iPhone app
    - Provides accessibility and convenience, but vulnerable to malware (botnets, spyware, ransomware) and other network-based attacks
  - Offline storage (cold wallet) – i.e., file on hard disk locked in safe, paper wallet
    - Less convenient, but more secure, so contains bulk of organization's reserves
    - Crucial nuance: must be connected to Internet to move bitcoins out



- Problem setup
  - Poisson processes
  - Cold wallet theft with fixed probability (p_{t_c})
- Online algorithm
  - Overflow if full (safe)
  - Refill if empty (risky)
- Goal
  - Maximize net balance of wallets over [0, T]
- The dilemma
  - Organization must service customer deposit and withdrawal requests
  - Storing too much in hot wallet – attrition due to recurrent, network-based theft
  - Storing too little in hot wallet – must access cold wallet to refill (risky)
  - *Central question* – what ceiling μ on hot wallet balance minimizes losses?

The Expected Balance

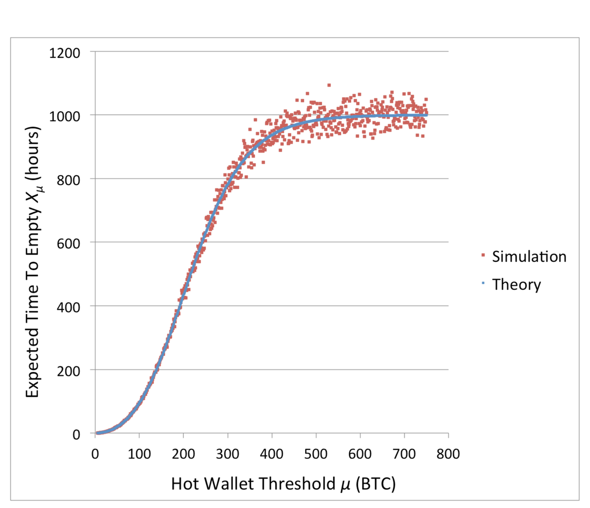
- Net balance at T is determined by events since the time of last cold wallet theft t_1
  - Net balance at T = net arrivals in $[t_1, T]$ - losses to hot wallet theft in $[t_1, T]$

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

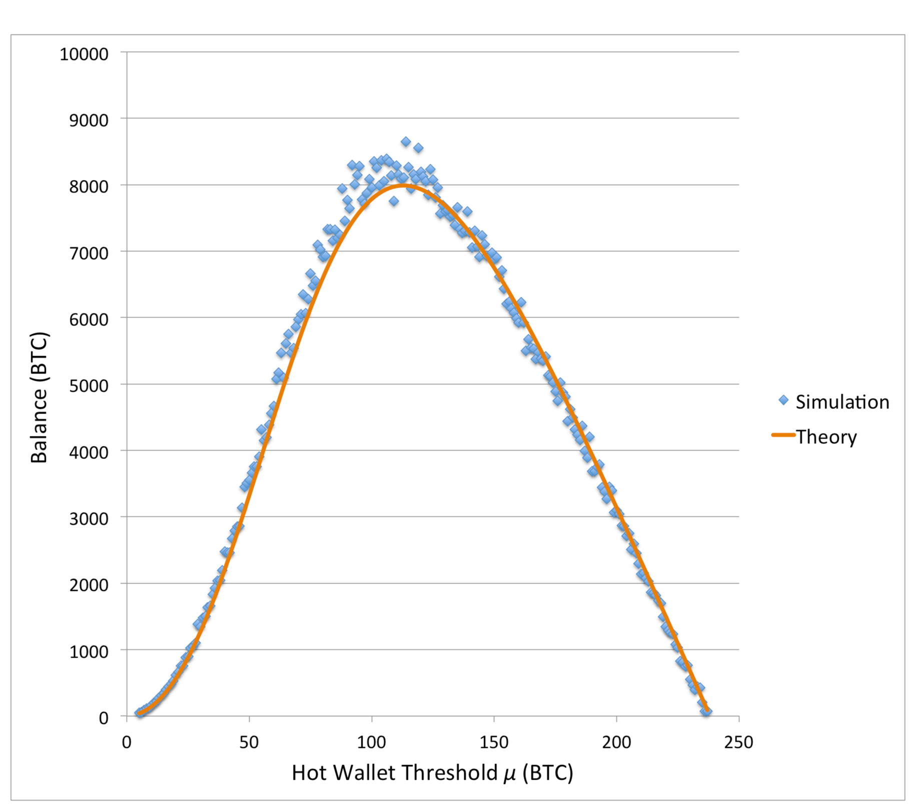
- Optimal value of μ

Theory (γ = 0.84)	Empirical (absolute max.)	Empirical (interpolation)
$\mu = 112.88$	$\mu = 114$	$\mu = 111.05$

- Evaluation of theoretical result
  - Within 1% of absolute empirical maximum
  - Within 2% of maxima of interpolated polynomial



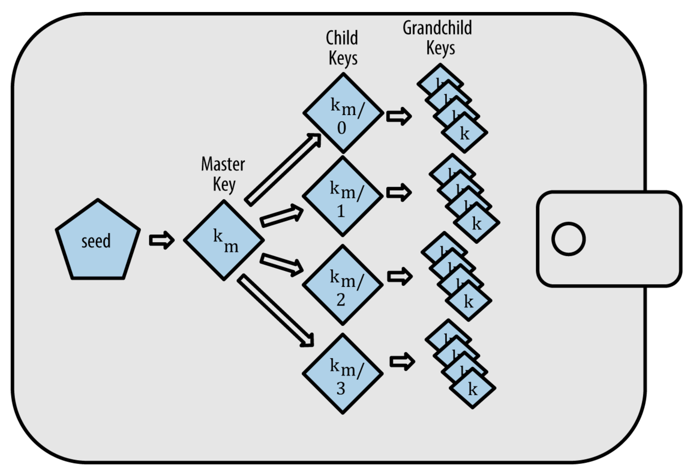
(Subsystem) Expected Time X_μ vs. Threshold μ



Expected Net Balance vs. Hot Wallet Threshold μ

Prior Work – Better Wallet Security

- Multi-signature transactions
  - *n* private keys for one public address; need *m-of-n* to construct transaction
  - Ensures that bitcoins aren't lost (stolen) if single machine is compromised
- Threshold cryptography
  - One private key split between *n* key holders (secret sharing protocol)
  - Benefits: privacy of signatory parties, bypasses limits of Bitcoin Script
- Deterministic wallets
  - Multiple keys derived from single seed via one-way hash fn.
  - Allows easy backups, recovery
  - Hierarchical deterministic wallets (BIP0032)
- All three – important advancements in wallet security, privacy, usability



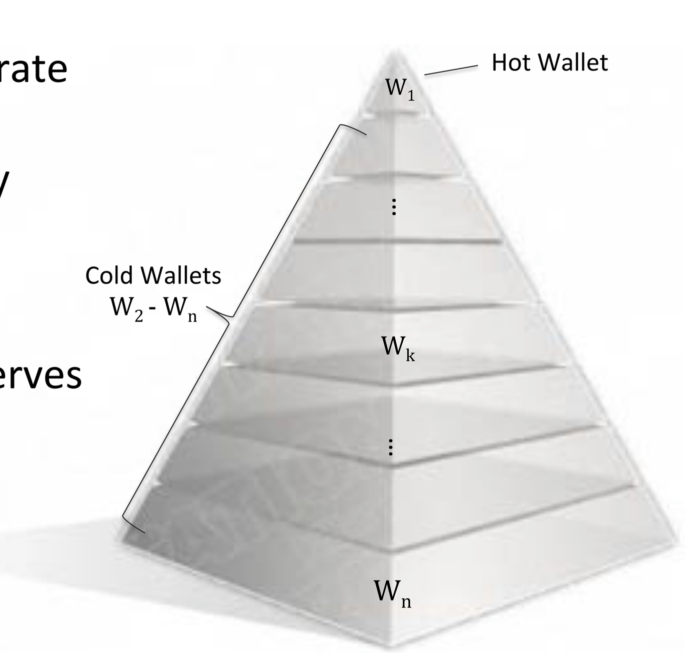
Hierarchical deterministic (HD) wallet

Event Driven Simulations

- Wrote test modules ExpectedTimeToEmpty and ExpectedBalance to evaluate equations for X_μ and Ex[B] respectively
- Chose sets of values for parameters λ_d, λ_w, λ_{th}, p_{t_c} to test predictions made by theoretical models against empirical values
  - Chose time frame [0, T] to be long enough to allow many (~200) hot wallet thefts and several (3-60) cold wallet thefts
- Main body of simulation
  - ExpectedTimeToEmpty: while (hotWalletBalance > 0) { … }
  - ExpectedBalance: while (time < T) { … }
  - Drew pseudorandom numbers from exponential distribution to generate waiting times to deposits, withdrawals, and hot wallet theft
  - Tracked hot/cold wallet balance over [0, T]
- Each data point (μ, X_μ) and (μ, B) corresponds to average over 1000 iterations of simulation

Further Work – More Complex Architectures

- Calibrated threshold
  - If deposit/withdrawal rates demonstrate predictable trends or periodicity…
  - Set threshold based on recent history (arrivals and thefts in last *k* hours)
- Multiple wallet systems
  - Goal: refills should not endanger reserves
  - "Retirement fund" wallets
    - Two cold wallets: 1) checking account and 2) savings account
  - Pyramid model
    - Multiple layers of cold storage
    - Bottom: more BTC, less frequent access



Pyramid wallet model