Defeating Vanish with Low-Cost Sybil Attacks Against Large DHTs

Scott Wolchok[1] Owen S. Hofmann[2]
Nadia Heninger[3] Edward W. Felten[3] J. Alex Halderman[1]
Christopher J. Rossbach[2] Brent Waters[2] Emmett Witchel[2]

[1] The University of Michigan [2] The University of Texas at Austin [3] Princeton University

Road Map

1. What is Vanish?
2. Attacking Vanish
3. Costs and performance
4. Countermeasures
5. What went wrong?

Why Self-Destructing Data?

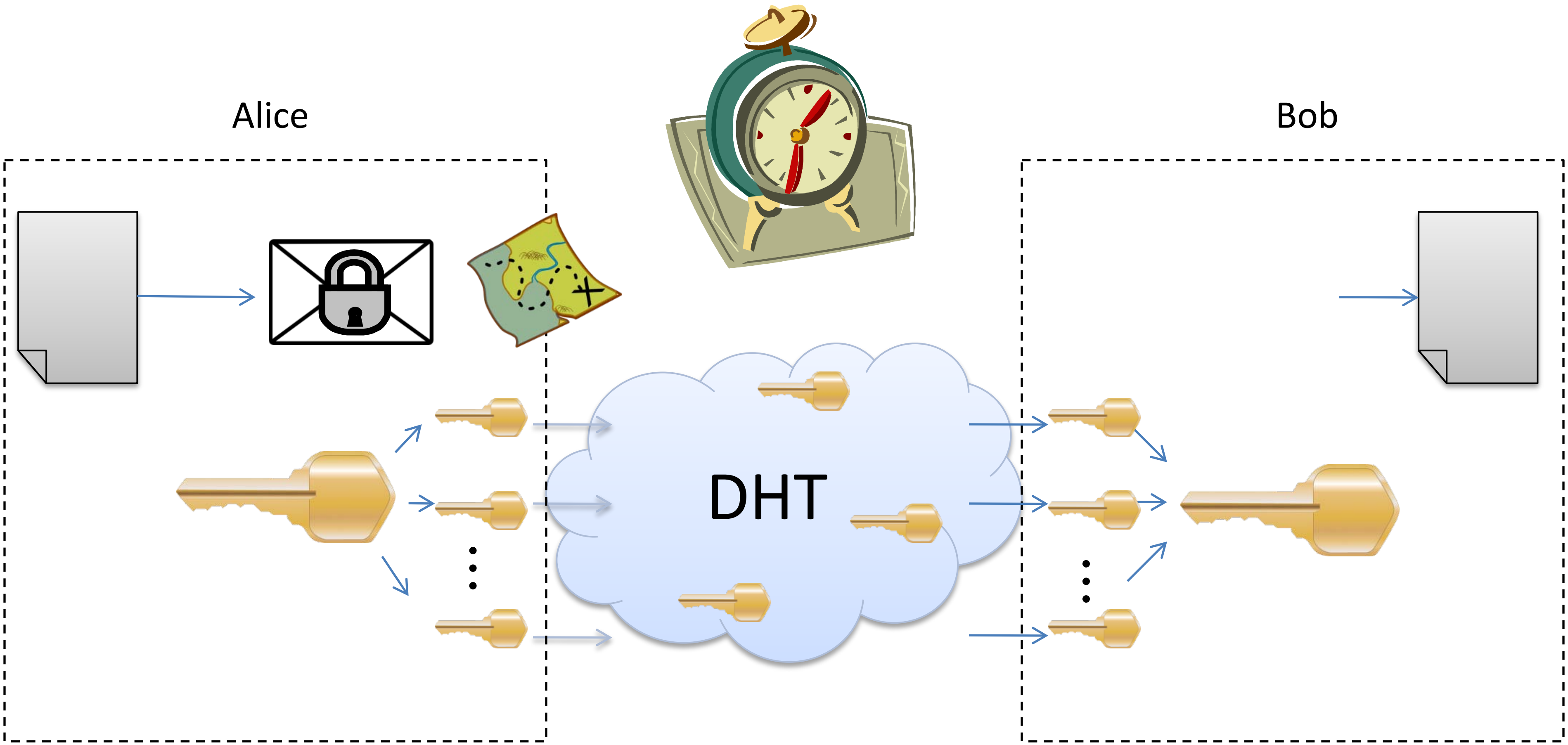
"Transient" messages tend to persist

Stored copies enable **retroactive attacks**

Attacker subpoenas data months or years later

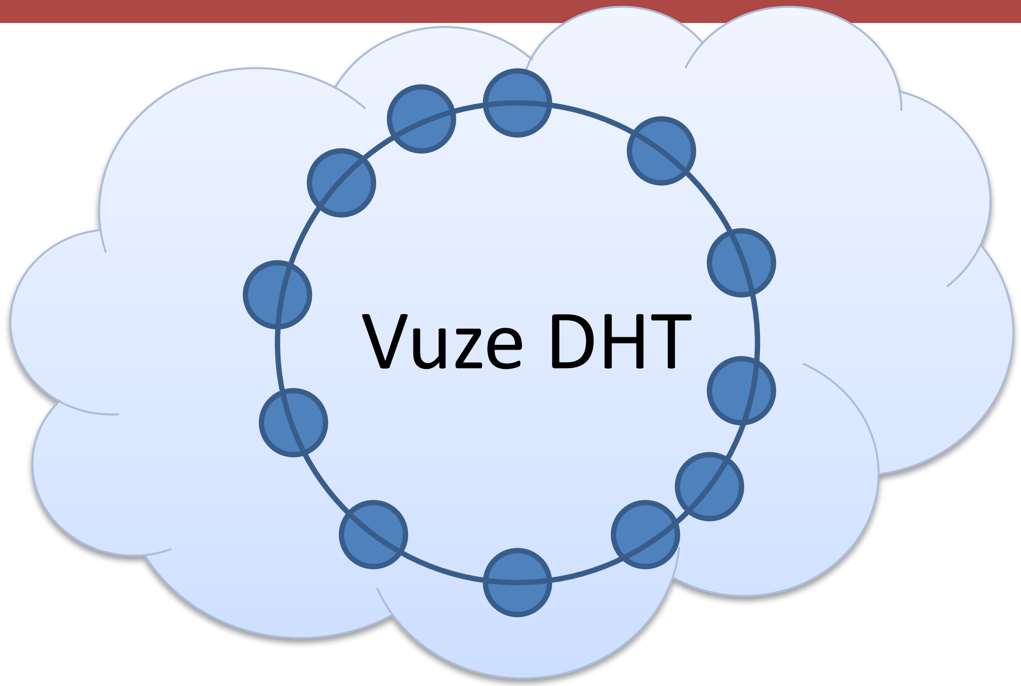
Vanish

Geambasu, Kohno, Levy, Levy — USENIX Security ’09



Mallory

Vanish and Vuze

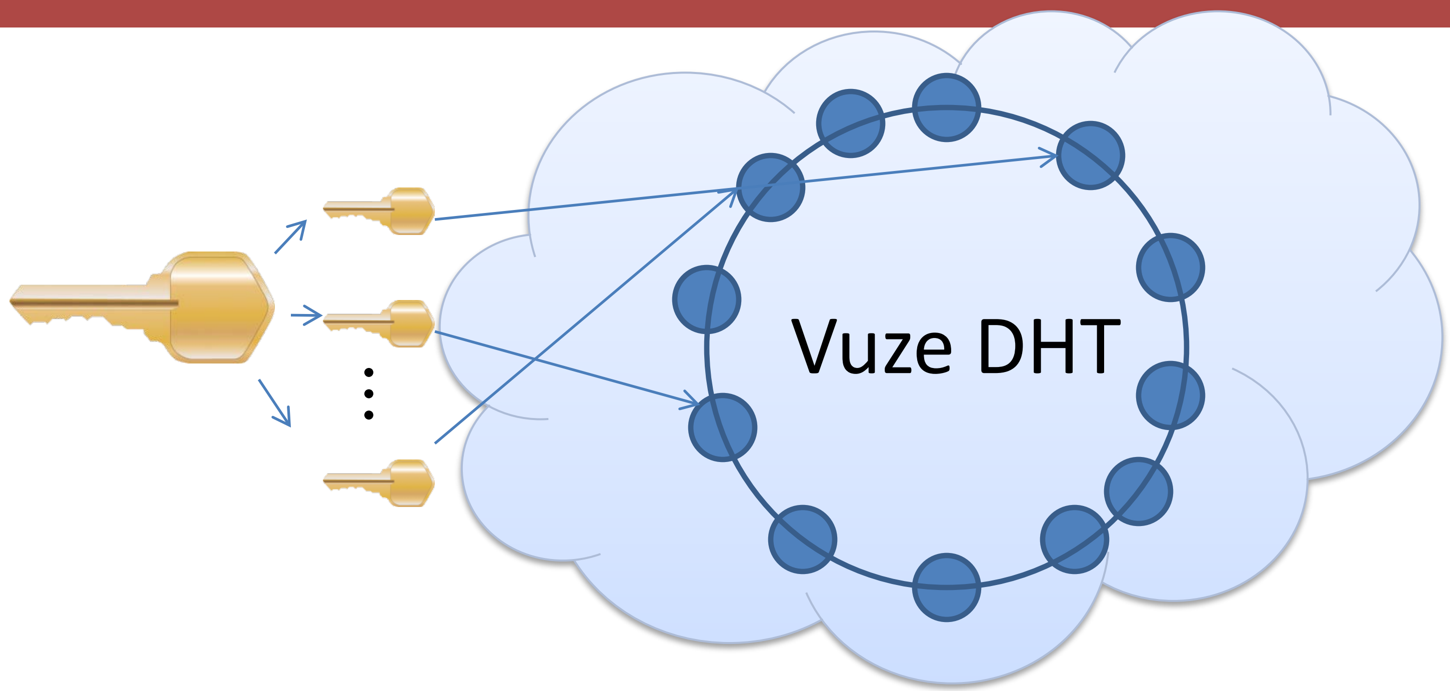


Vanish uses the Vuze DHT (Distributed Hash Table)

Over 1 million nodes, mostly BitTorrent

Nodes delete values after 8 hours

Vanish and Vuze



Shares placed at random locations in the DHT
Replicated to 20 “closest” nodes

Is Vanish Secure?

Vanish 0.1 prototype released at publication

Included user-friendly Firefox plugin

Focused wide attention on its practical security

Road Map

1. What is Vanish?
2. Attacking Vanish
3. Costs and performance
4. Countermeasures
5. What went wrong?

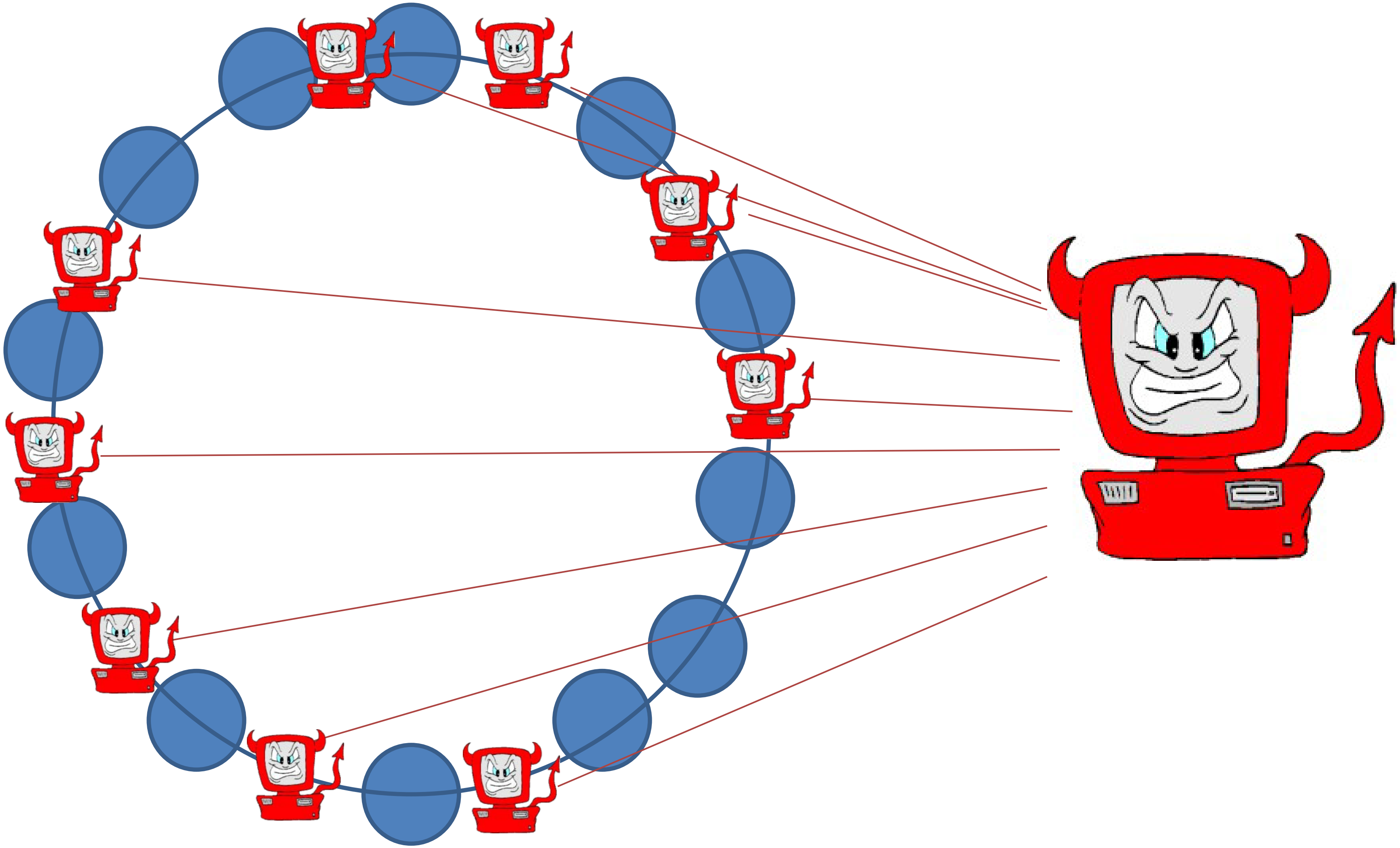
DHT Crawling Threat

Threat: attacker might continuously archive *all* data in the DHT

Later, query archive to decrypt messages

Don't need specific targets when recording

Crawling with a Sybil Attack

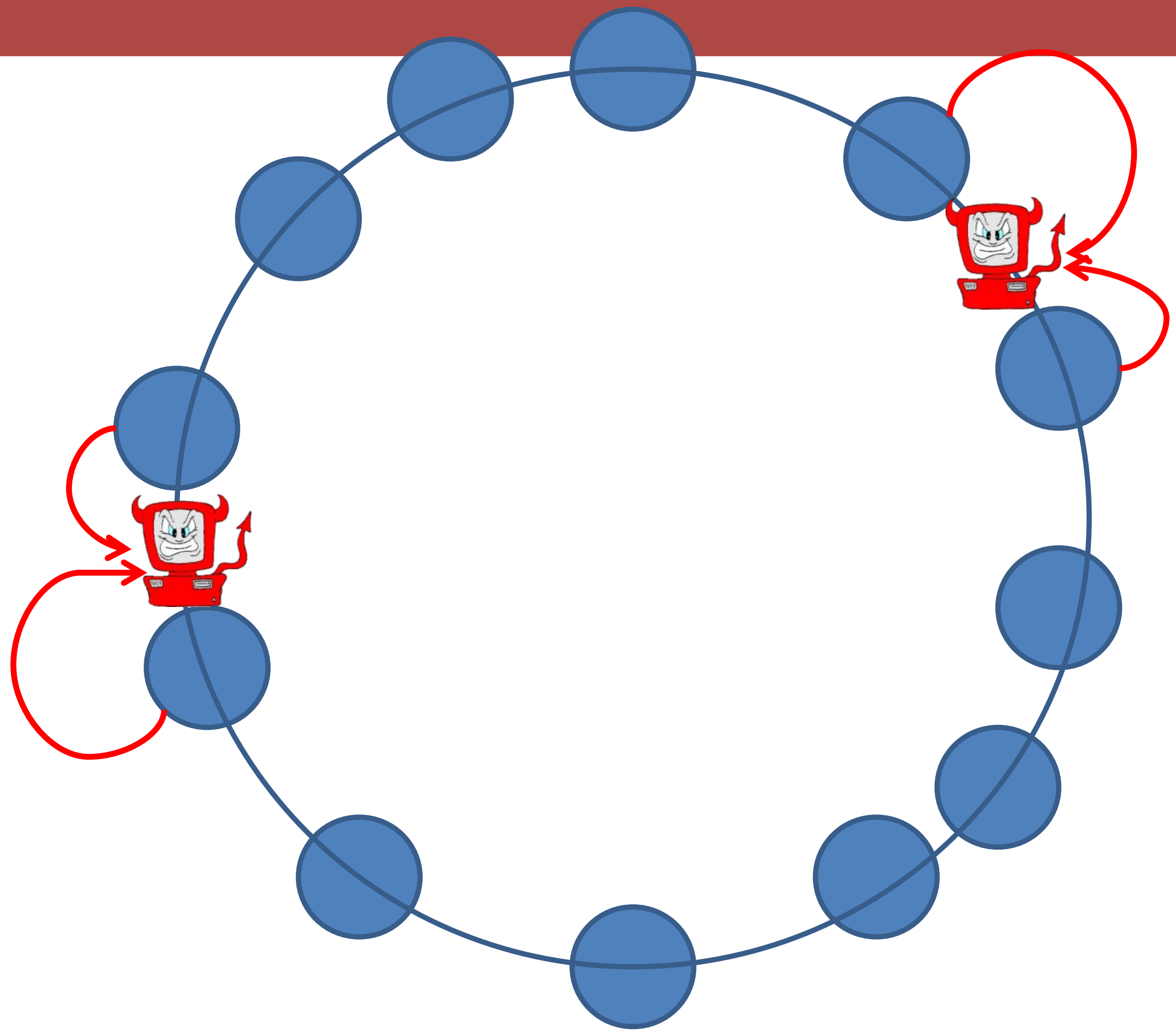


Making the Attack Practical

Insight: have **8 hours** to observe fragments

Vuze replicates to 20 nearest nodes

1. Every 30 minutes
2. On join!



“Hopping” Strategy

Sybils “hop” to new IDs every 3 minutes

160x resource amplification over 8 hours

Practical attack needs only ~2000 *concurrent* Sybils with hopping

Making the Attack Practical

Insight: Vuze client is a notorious resource hog

Only 50 instances fit in 2 GB of RAM!

Can we more efficiently support 2000 Sybils?

Optimized Sybil Client

C, lightweight, event-based implementation

Listen-only (no Vuze routing table!)

Thousands of Sybils in one process

Road Map

1. What is Vanish?
2. Attacking Vanish
3. Costs and performance
4. Countermeasures
5. What went wrong?

Attack Costs?

Vanish paper estimate (for 25% recovery at *k*=45, n=50):

- 87,000 Sybils
- $860,000/year

What does attacking Vanish *really* cost?

Experiments

1. Insert key shares into the DHT
2. Run attack from 10 Amazon EC2 instances
3. Measure:

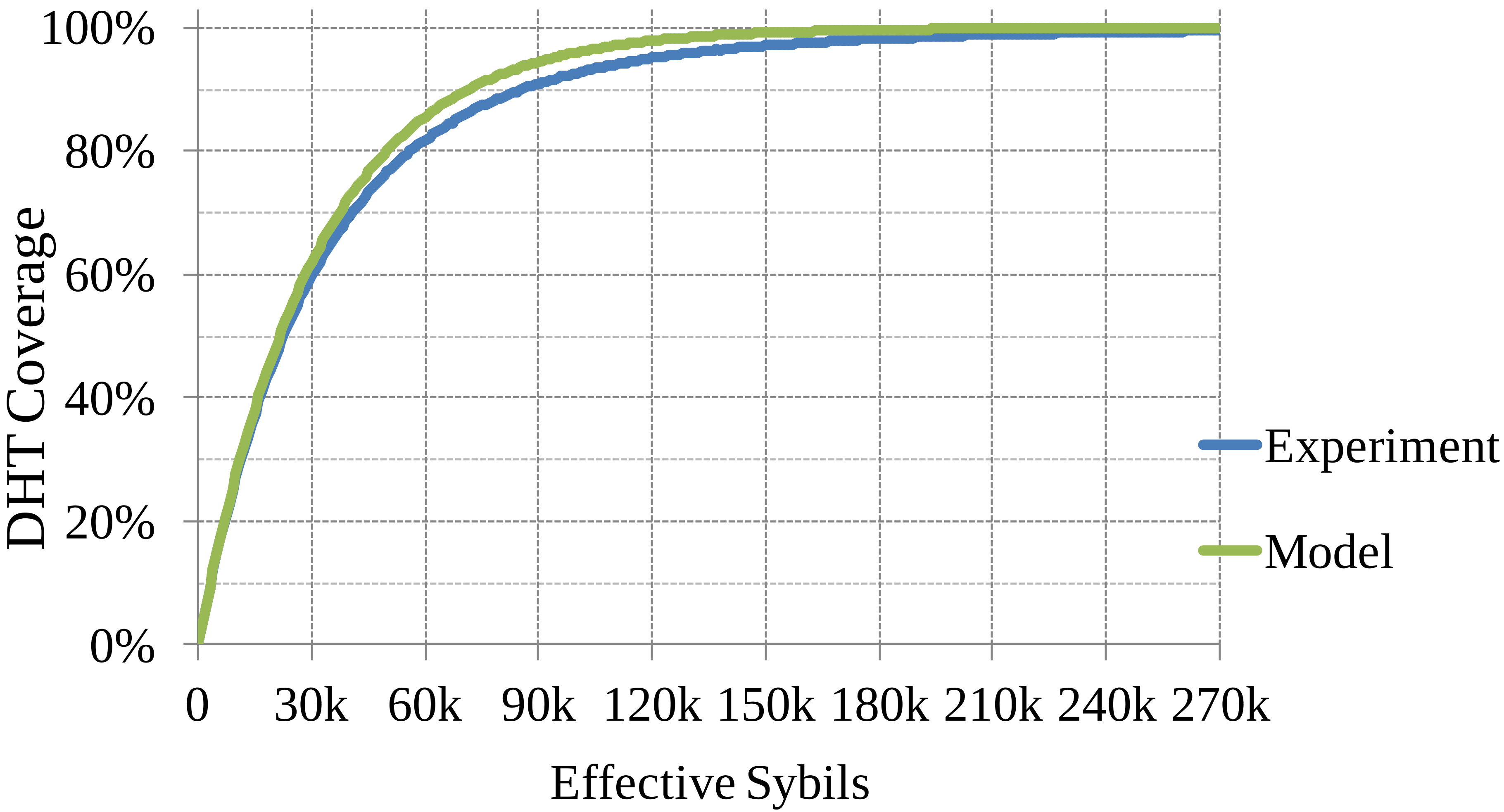
DHT coverage	=	% key shares recovered
Key coverage	=	% messages decrypted
Attack cost	=	EC2 charges (Sep. 2009)

Experimental Results

Cost for >99% Vanish key recovery?

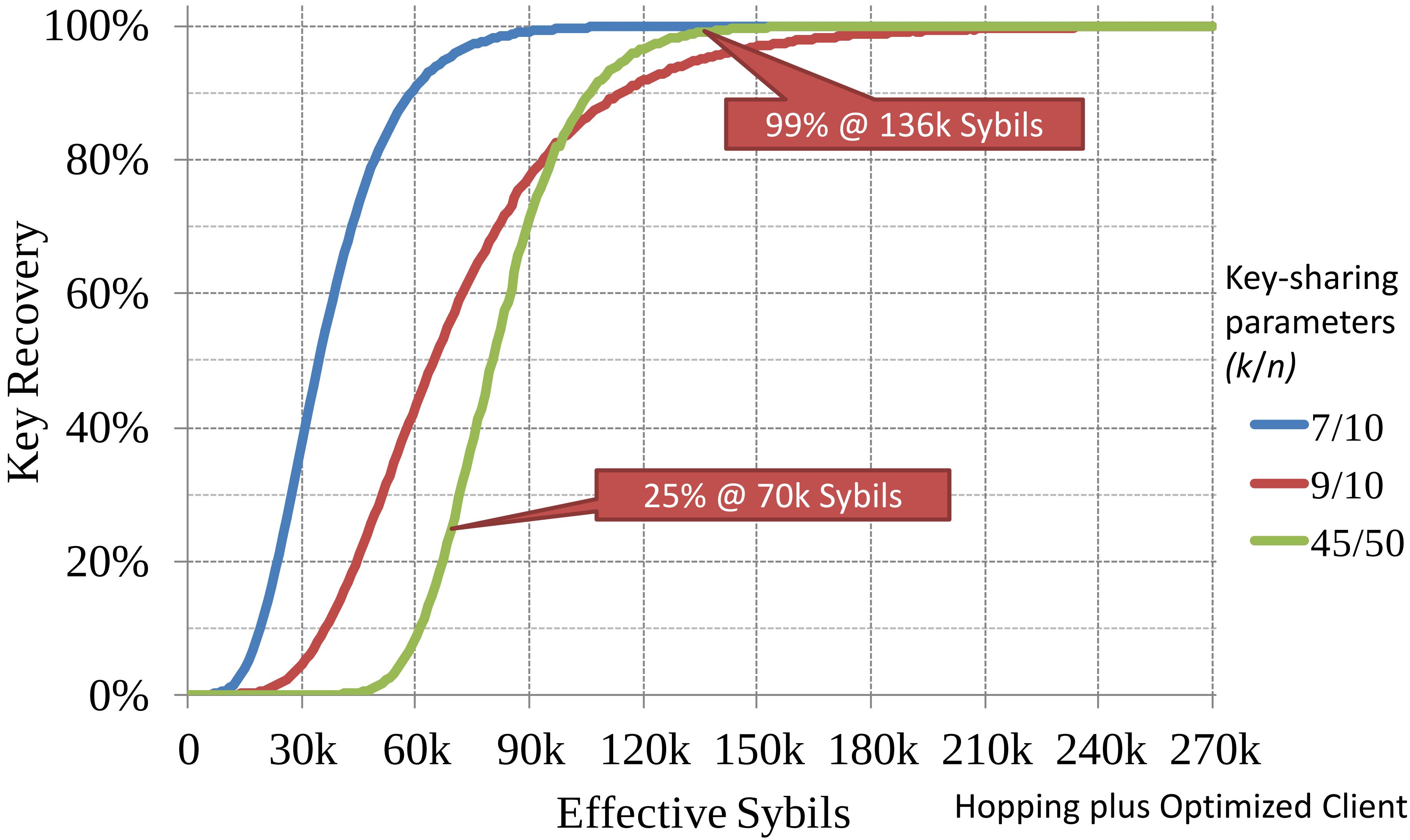
Attack	Concurrent Sybils	Key Shares Recovered	Annual Attack Cost*
Hopping	500	92%	**$23,500**
Hopping + Optimized Client	2000	99.5%	**$9,000**

DHT Coverage vs. Attack Size



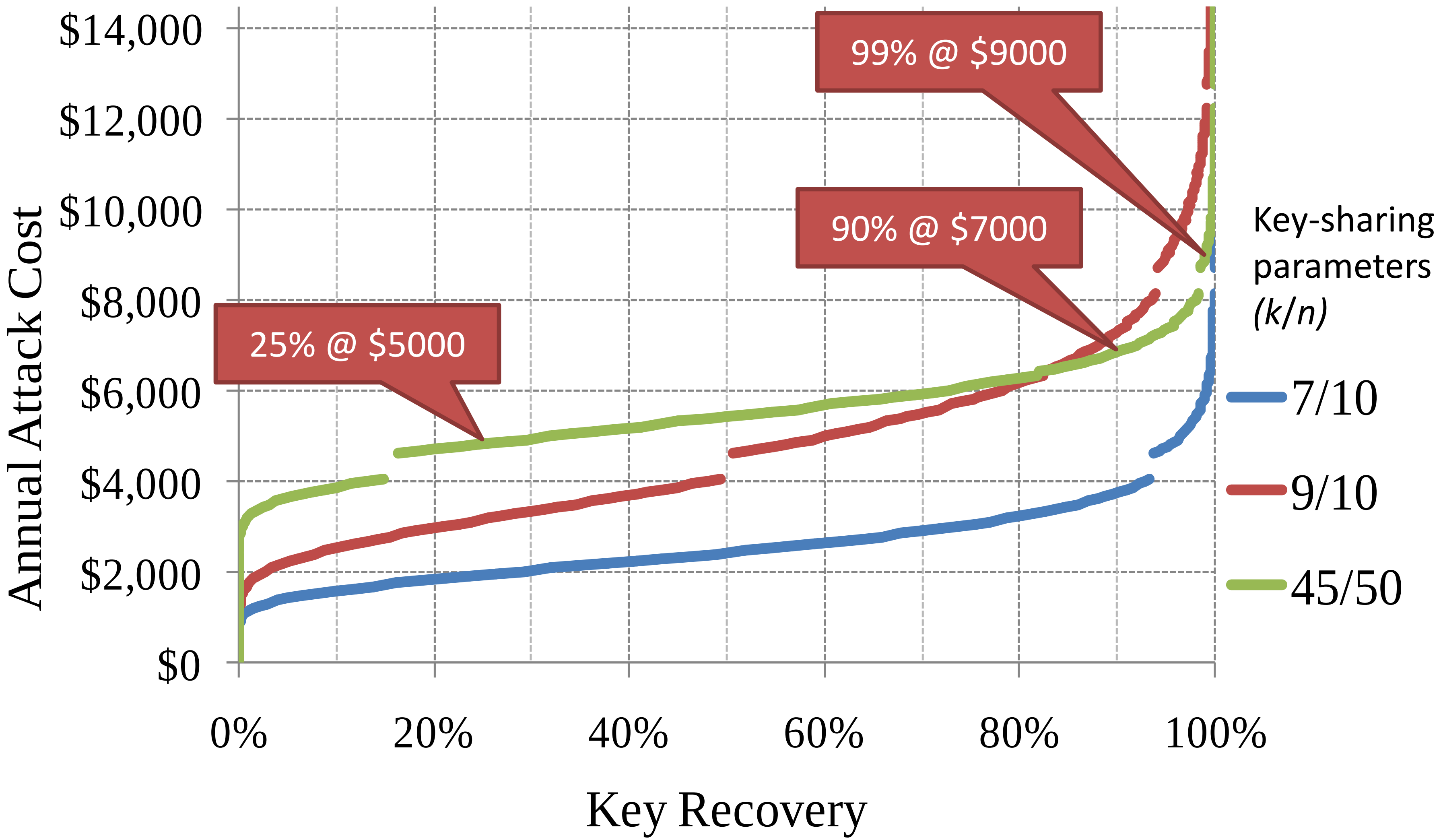
Hopping plus Optimized Client

Key Recovery vs. Attack Size



Hopping plus Optimized Client

Annual Cost vs. Key Recovery



Hopping plus Optimized Client

Storage

$1400/yr for *all* observed data

$80/yr for potential key shares

Road Map

1. What is Vanish?
2. Attacking Vanish
3. Costs and performance
4. Countermeasures
5. What went wrong?

Increase Key Recovery Threshold?

Required coverage increases in *n* and *k/n*

Why not raise them? (99/100?)

- **Reliability**: some shares lost due to churn
- **Performance**: pushing shares is slow!

Limit Replication?

Attack exploits aggressive replication

Less replication might make the attack harder, but how much?

More in a few slides…

Sybil Defenses from the Literature?

Client puzzles

Limit ports/IP, IPs/subnet, etc.

Social networking

Detecting Attackers

Find and target IPs with too many clients

Use node enumerator, *Peruze*

Can detect attack IPs hours after the attack

Detected the Vanish demo

Road Map

1. What is Vanish?
2. Attacking Vanish
3. Costs and performance
4. Countermeasures
5. What went wrong?

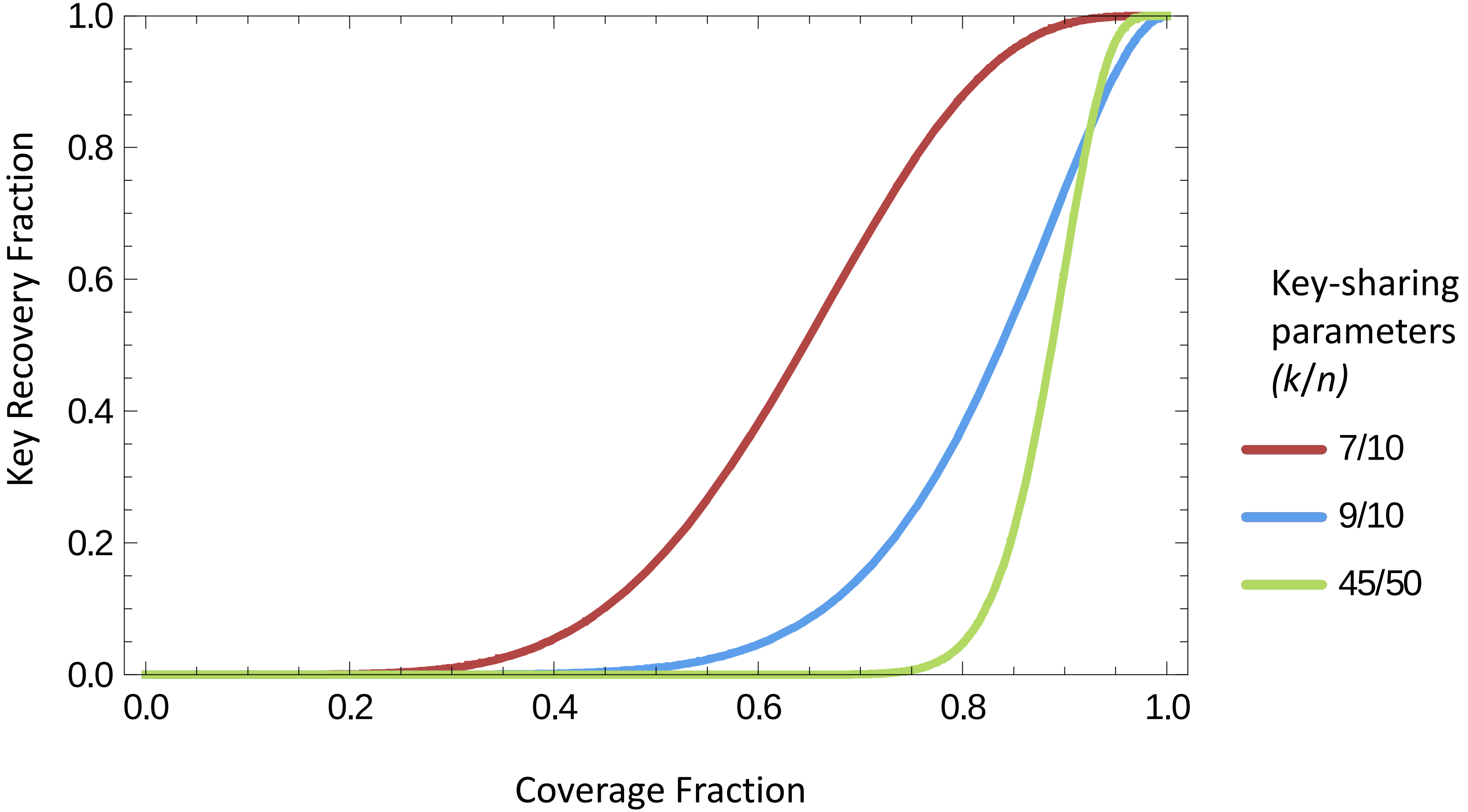
Cost Estimation Issues

Vanish paper extrapolated from 8000-node DHT

Assumed Sybils must run continuously

Assumed attacker uses inefficient Vuze client

Cost Not Linear in Recovery



Response to Our Work

Second report and prototype by Vanish team[1]

New defenses

- Use both Vuze DHT and OpenDHT
- Disable replicate-on-join in Vuze
- Use less aggressive "threshold replication"

Will these defenses stop real attackers?

[1] Geambasu, Falkner, Gardner, Kohno, Krishnamurthy, Levy. "Experiences building security applications on DHTs". Technical report, UW-CSE-09-09-01.

Conclusion

Showed attacks that defeat Vanish 0.1 in practice for $9000/year

Vanish team has proposed new defenses

Future work: are new defenses effective?

Our take: building Vanish with DHTs seems risky.

Defeating Vanish with Low-Cost Sybil Attacks Against Large DHTs

Scott Wolchok[1] Owen S. Hofmann[2]
Nadia Heninger[3] Edward W. Felten[3] J. Alex Halderman[1]
Christopher J. Rossbach[2] Brent Waters[2] Emmett Witchel[2]

[1] The University of Michigan [2] The University of Texas at Austin [3] Princeton University

http://z.cs.utexas.edu/users/osa/unvanish/

Vanish Attack Model

Need to recover *k* of *n* fragments

p = Pr{recover key fragment}

Pr{recover VDO} = Pr{recover *k* or more fragments}

Binomial distribution

Pr{recover VDO} = $\sum_{i=k}^{n} \binom{n}{k} p^{i} (1-p)^{n-i}$

Coverage Model

m Sybils see c of N objects

Balls-in-bins problem

Expected fraction = $1 - e^{-cm/N} = 1 - e^{-sm}$

$s = c/N$ is the (overlapping) fraction of the network observed by each Sybil

Prior Work

- Enumerating DHT nodes
  - Cruiser [Stutzbach 2006a,b]
  - Blizzard [Steiner 2007a]
- Measuring DHT traffic
  - Mistral [Steiner 2007b]
  - Montra [Memon 2009]

Hopping plus Optimized Client

Concurrent Sybils	Hours	# VDO Fragments	Fragments Found
2000	8	1650	1640 (99.4%)
2000	7.5	1700	1692 (99.5%)
500	8	1650	1561 (91.8%)