Presentation on “Vanish: Increasing Data Privacy with Self-Destructing Data”

Presented by William Ng

Needs for vanishing data

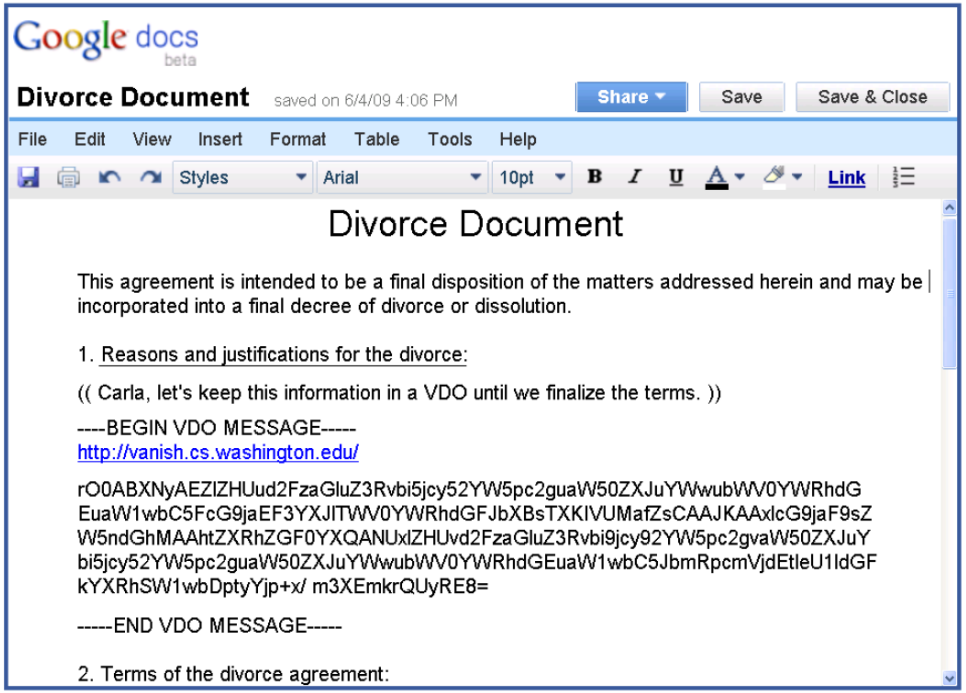
- Emails, facebook messages or any web contents that you created could come back and be used against you
  - Sensitive discussion on divorce
- Hard to control where the content is
- Hard to act on the content in remote location
- Encryption password could be forced to be given up

Solution

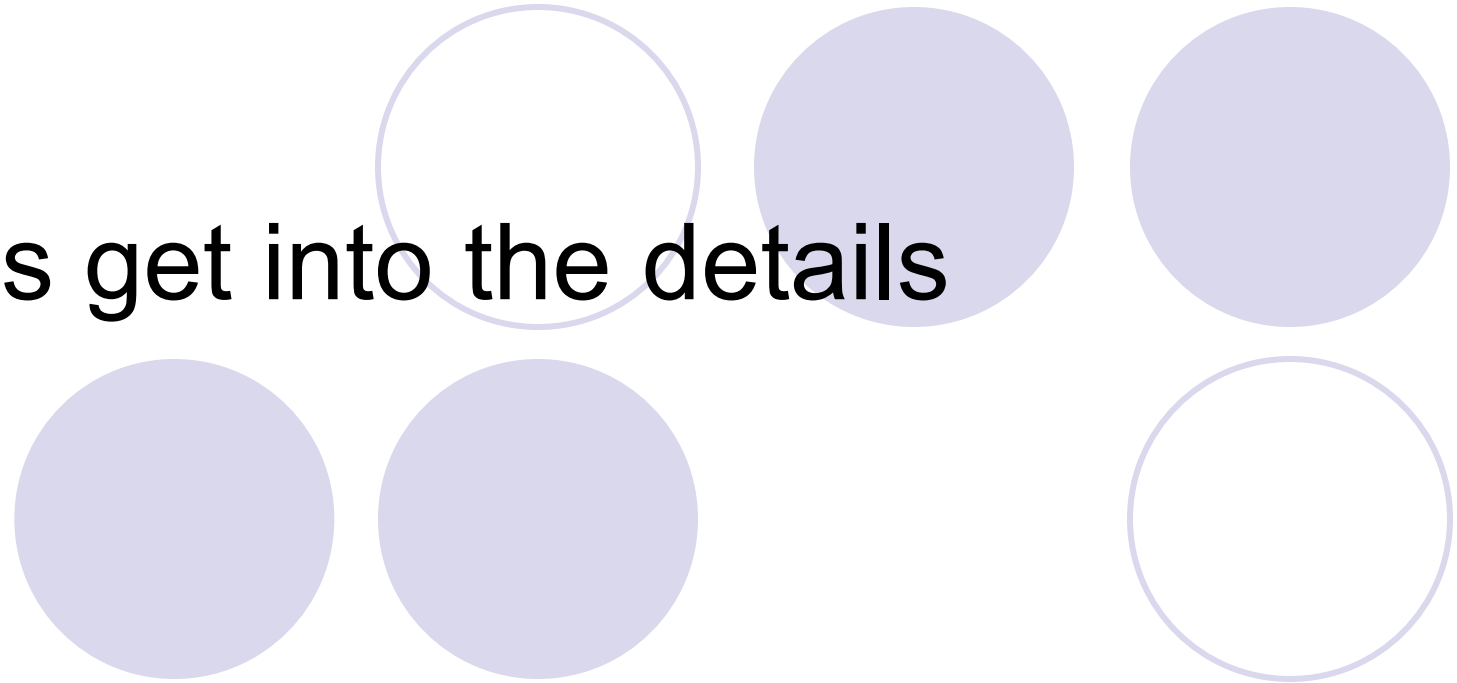
- Encrypt the content with a key and store the key in a high-churn globally-distributed DHT system
- Once it reaches the timeout value, the key would be erased from the DHT and forever lost. The content will not be readable without the key.

Implemented application

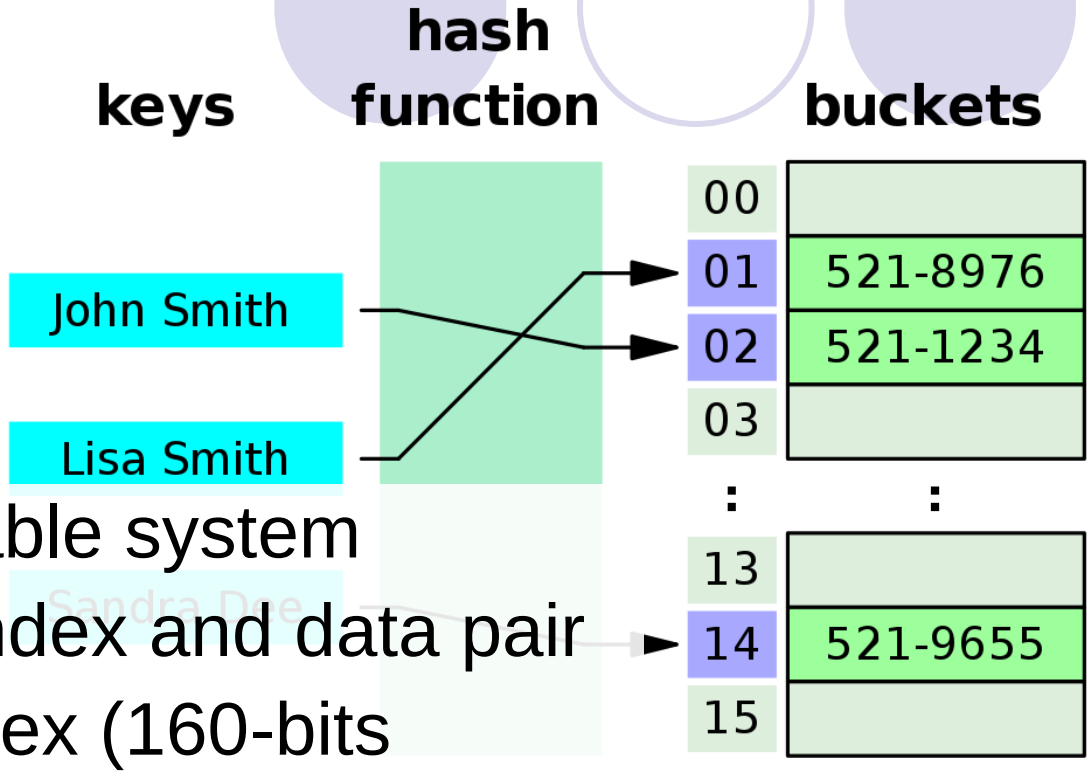
- FireVanish: Firefox plugin to be used on Gmail, facebook, or Google Docs, any Web page text input box
  - Simple, intuitive, wide-application
- Vanishing files



Let’s get into the details



DHT?



- Distributed Hash Table system
- Like an array with index and data pair
- Huge number of index (160-bits keyspace in Vuze = ~10^{48} indexes)
- Each node in the network are associated with an index (or node ID). It stored all the data with indexes closest to its ID.

DHT operation

- Using the hash function, a node found out an index that corresponds to the specific data
- It performs "***lookup***" to find out which node responsible for the index (could be different algorithm)
- It can then either ask that node to "***store***" the data or to "***get***" the data

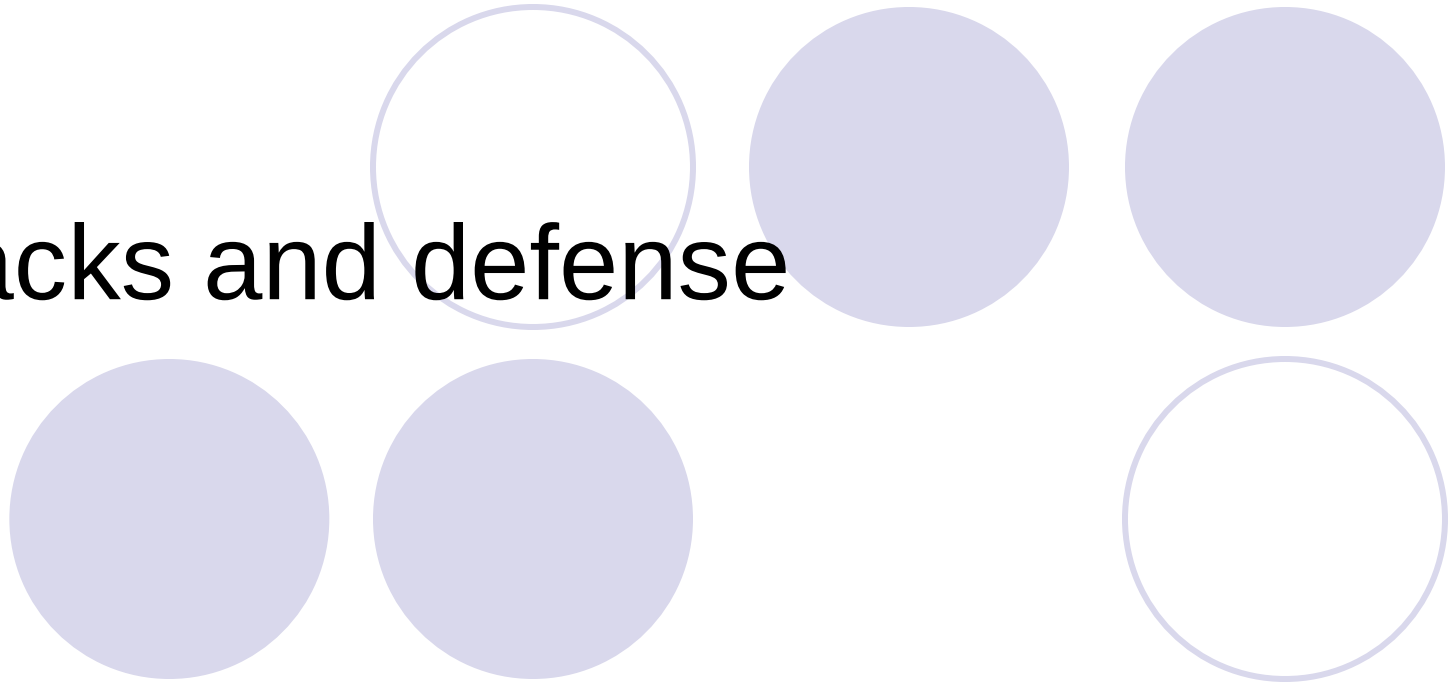
Vanish usage of DHT

- In normal use case of DHT, use hash function to find out the index of the desired data
- But in the case of Vanish, index and data are **not** related
- Vanish encrypts the content with key K, split it into N keys -> data to be stored
- Uses another random key L as seed of random generator to generate N indexes
- VDO consists of (L, C, N, threshold)

They pick Vuze DHT

- Vuze DHT
  - Open to be joined by any users
  - Millions plus nodes, geographical distributed through the world across different nations
  - High churn, user leaving and entering within the network (average duration around 2 hours – see appendix)
  - Fixed 8 hours timeout
- OpenDHT
  - Restricted membership
  - Variable time out up to 1 week

Attacks and defense



Overall theory in defense

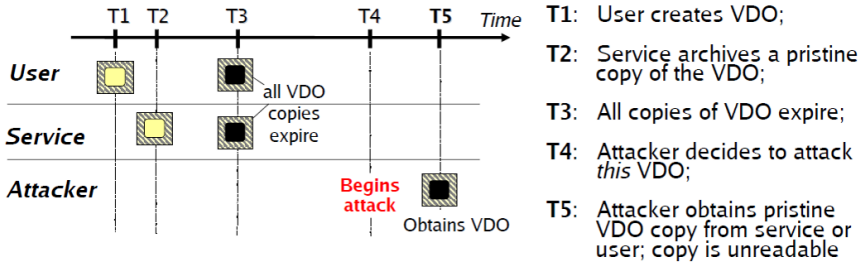


Figure 2: **Timeline for VDO usage and attack.**

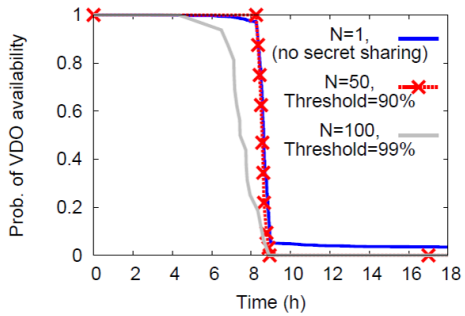
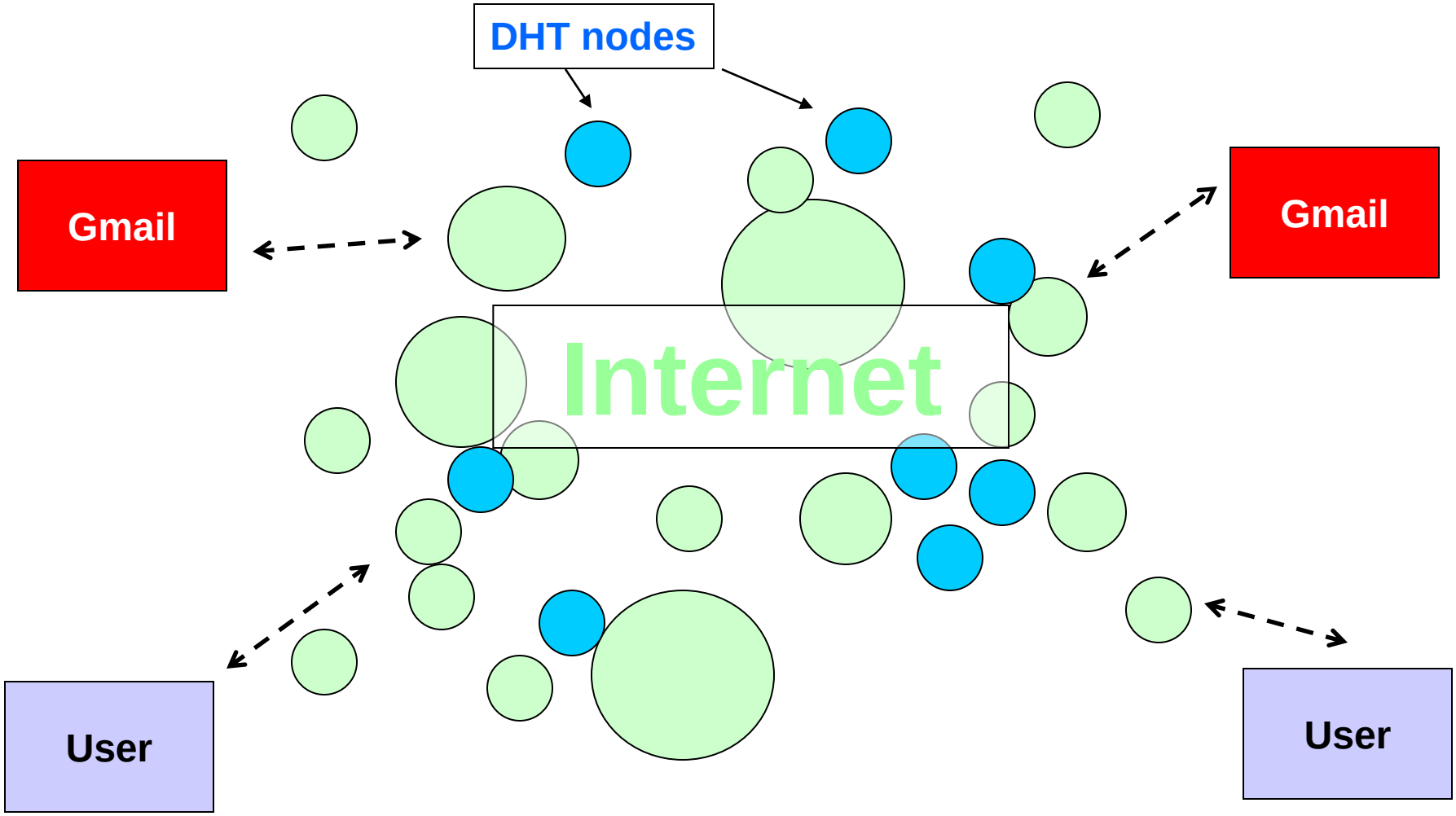


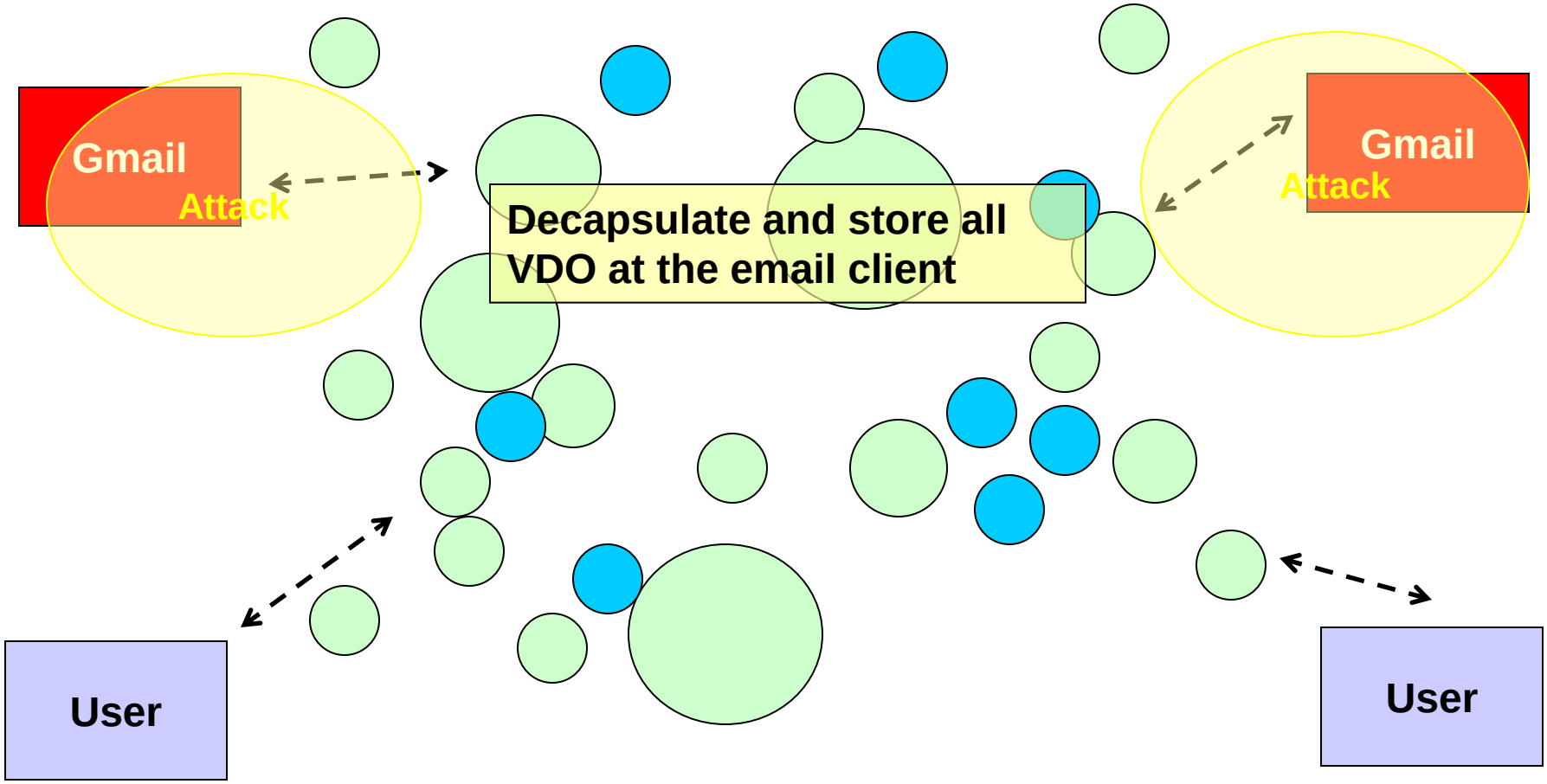
Figure 4: **VDO availability in the Vuze-based Vanish system.** The availability probability for single-key VDOs ($N = 1$) and for VDOs using secret sharing, averaged over 100 runs. Secret sharing is required to ensure pre-timeout availability and post-timeout destruction. Using $N = 50$ and a threshold of 90%

- Available until expiration
- Automatically becomes unreadable, even without actions of the user
- No secure hardware required from both users
- No centralized system (unlike Hushmail) to be comprised by the government or hackers

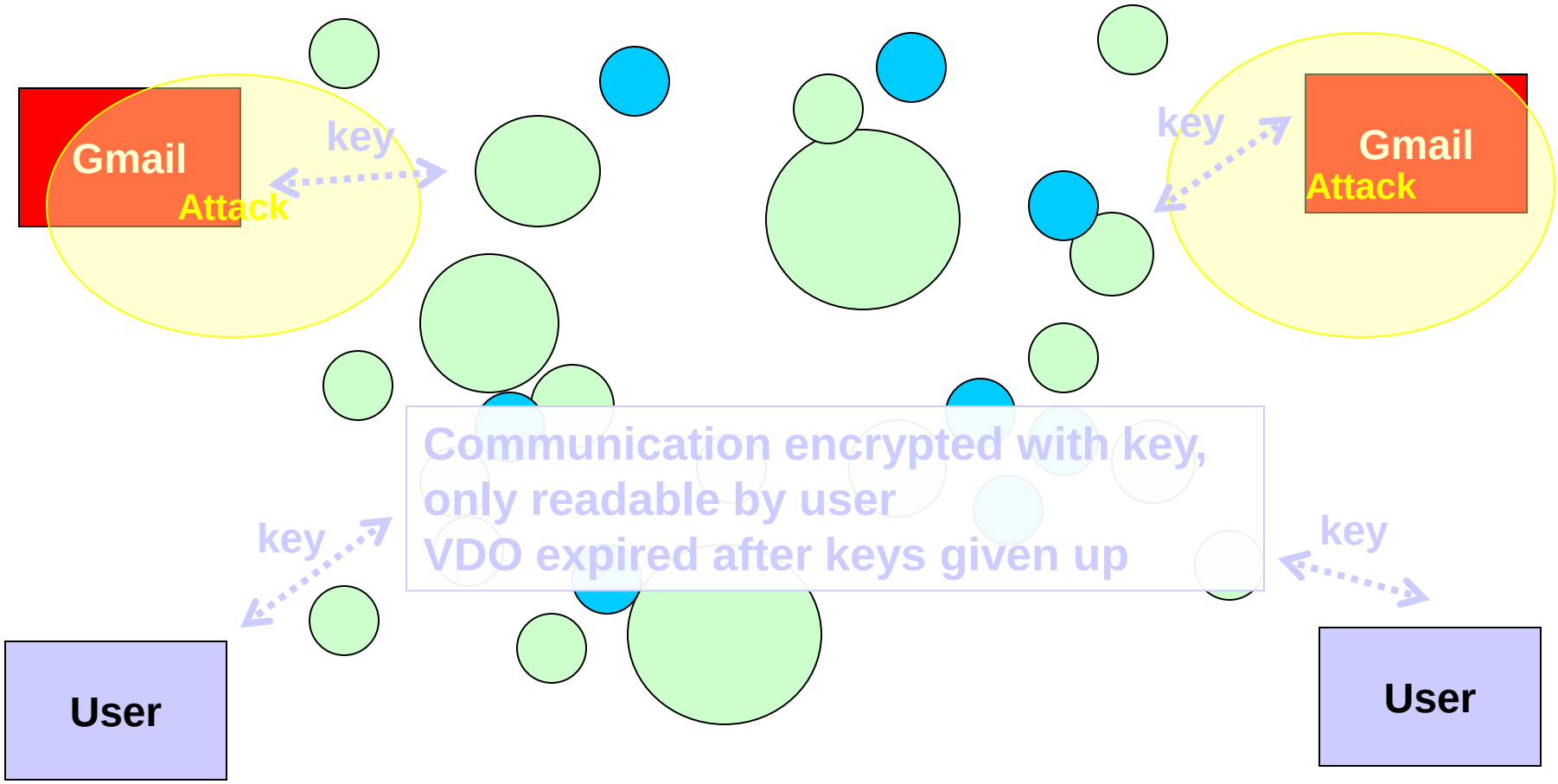
User, email client, internet, DHT nodes



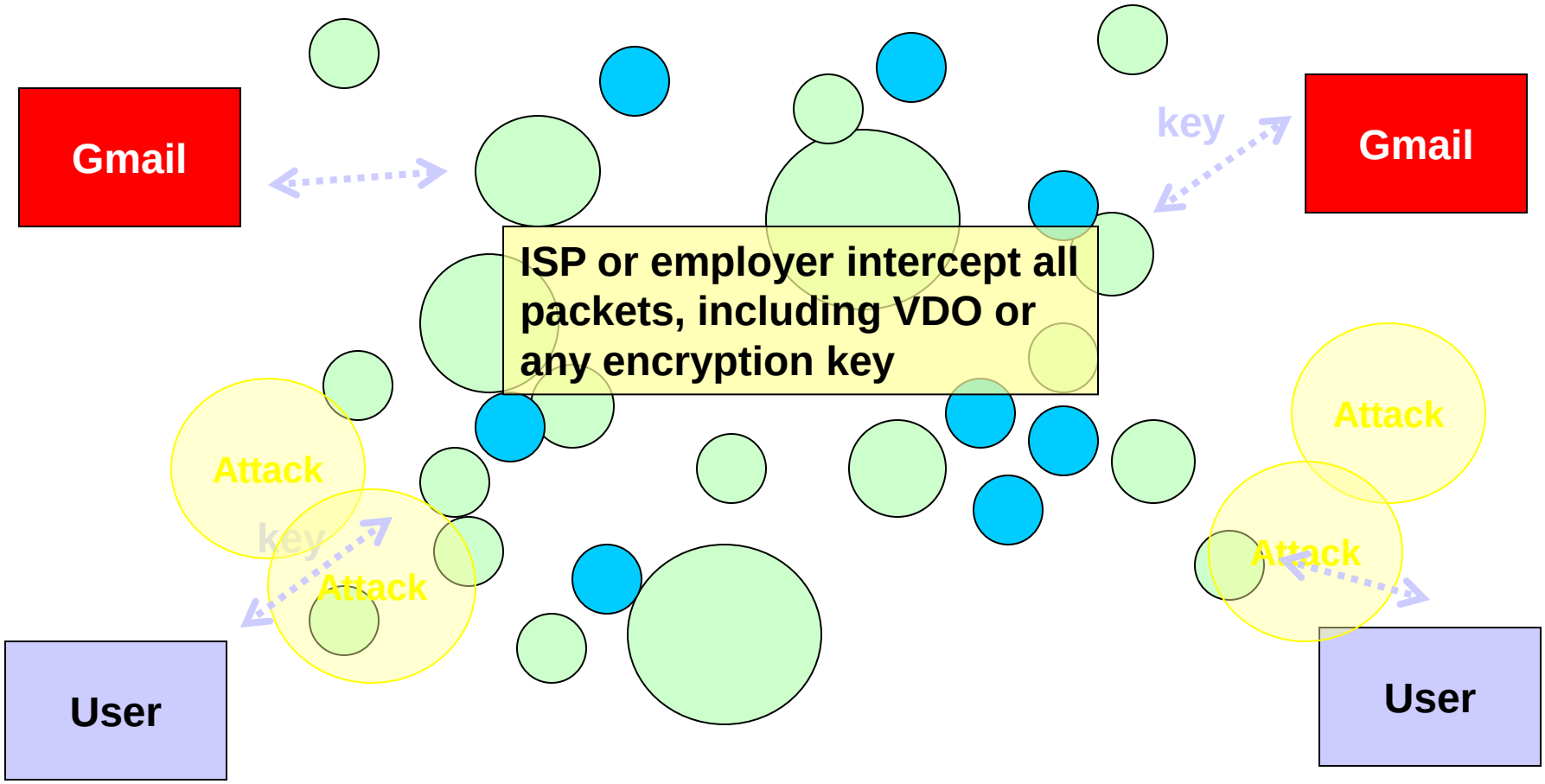
Attack strategy 1: Decapsulate VDO prior to Expiration



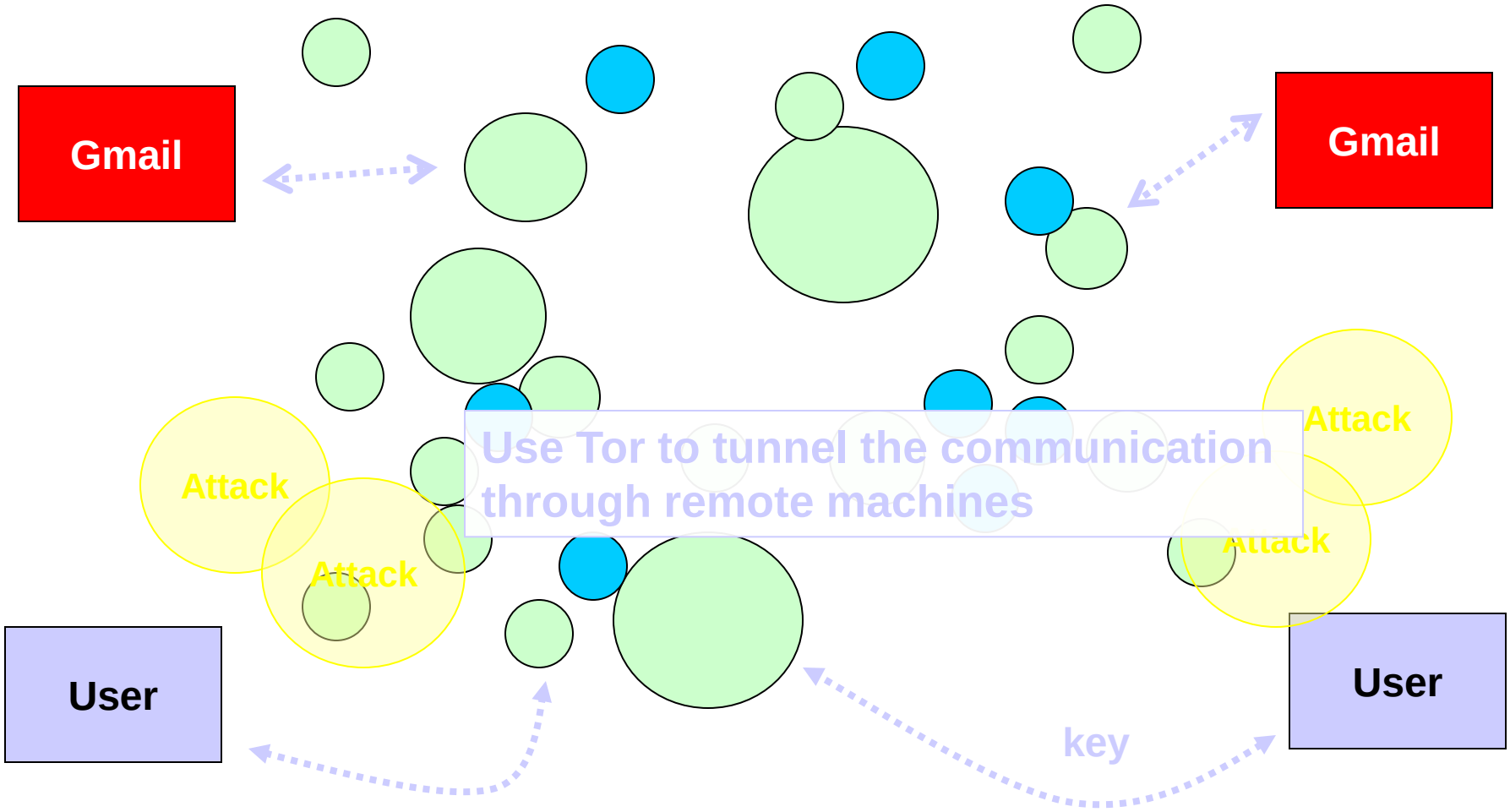
Defense: encrypt the VDO with another key encryption scheme like PGP or GPG



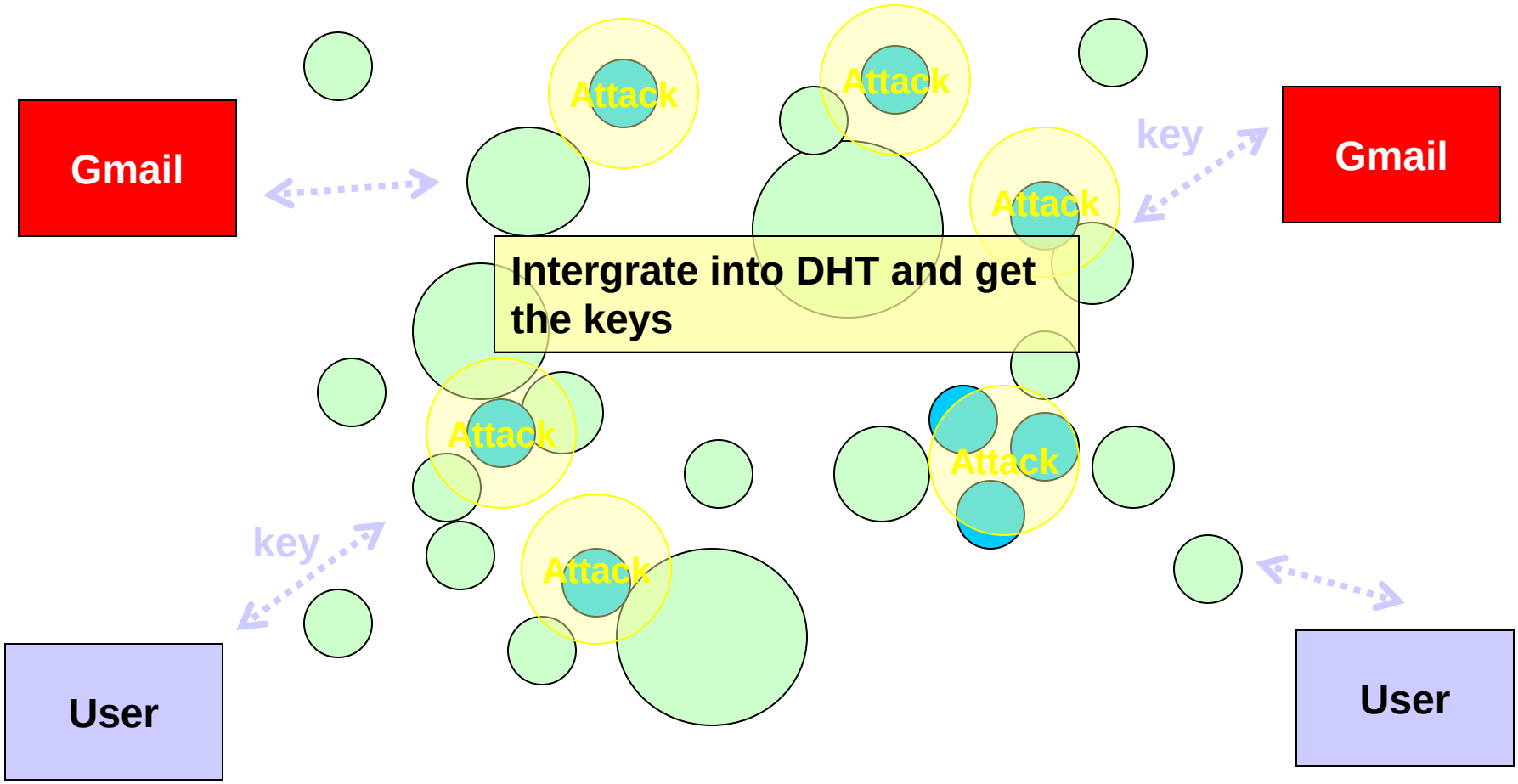
Attack strategy 2: Sniff User's internet Connection



Defense: use Tor

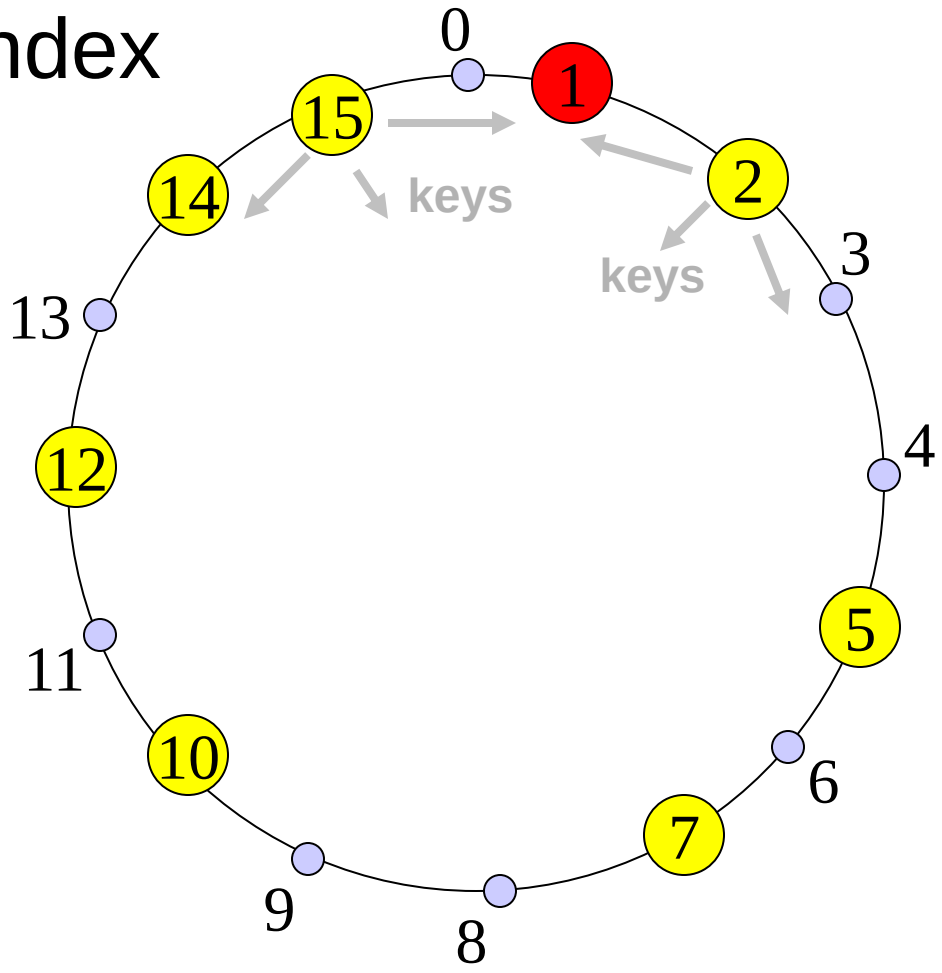


Attack DHT!

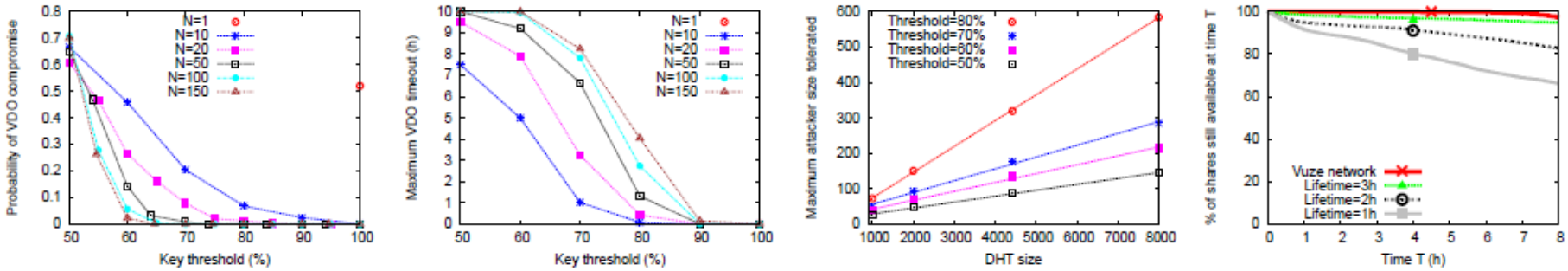

Gmail
Attack
Attack
key
Gmail
Attack
Intergrate into DHT and get the keys
Attack
Attack
key
Attack
User
User

Attack DHT: "*store*" sniffing

- Join the network and get as much keys and index pairs as possible
- Periodic push from neighbors



How to tweak parameters to defend against store sniffing



(a) Parameters and security. (b) Parameters and availability. (c) Tolerated attacker sizes. (d) Churn effect on availability.

Figure 7: **Analysis of the store sniffing attack.** Fig. (a): the attacker's success probability with increasing N and key threshold for a 1000-node DHT with 50 malicious nodes. Larger N and high thresholds ($\geq 65\%$) provide good security. Fig. (b): maximum VDO timeout supported for a .99 availability level. Large N with smaller key thresholds ($\leq 70\%$) provide useful VDO timeouts. Fig. (c): maximum number of attacker nodes that a DHT can tolerate, while none of the 1,000 VDOs we pushed were compromised. Fig. (a), (b), and (c) assume 2-hour churn. Fig. (d): the single-share availability decreases over time for different churn models in our private network and for the real Vuze network.

- Attackers needs to collect the keys in 8 hours!
- Have to do it 24x7 and store all of them

Cost to attackers using store sniffing

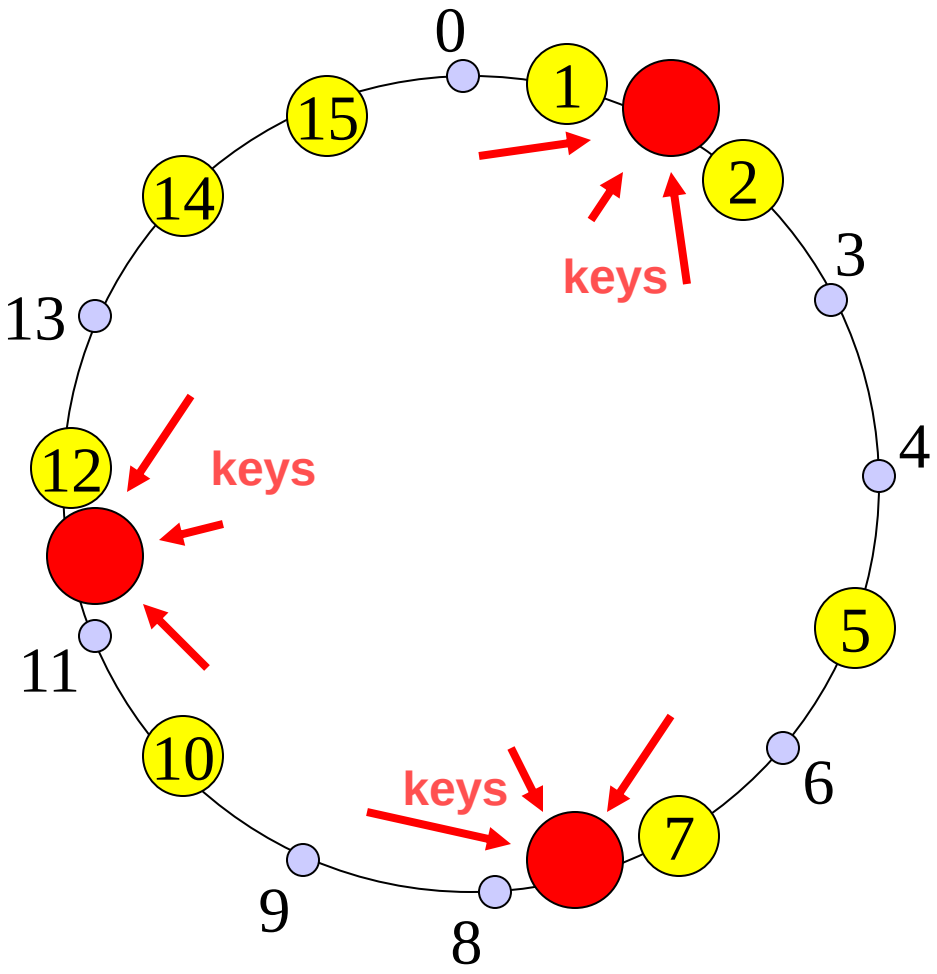
- Using 3 hour churn model, N=50, 90% threshold, in order to comprise 25% of VDO on Vuze, it is estimated to need
  - 87,000 nodes
  - = $860K per year

Attack DHT: “*Lookup*” sniffing

- Attackers don’t know what is valid key in the 160-bits keyspace
- Use the “lookup” request that comes to them
- Defense: change the local vuze node, so it obfuscates the key

Attack DHT: Sybil attack

- Assume different identities in a short period of time
- Each time it joins the network, it gets keys from neighboring node
- Attack used in Unvanish!



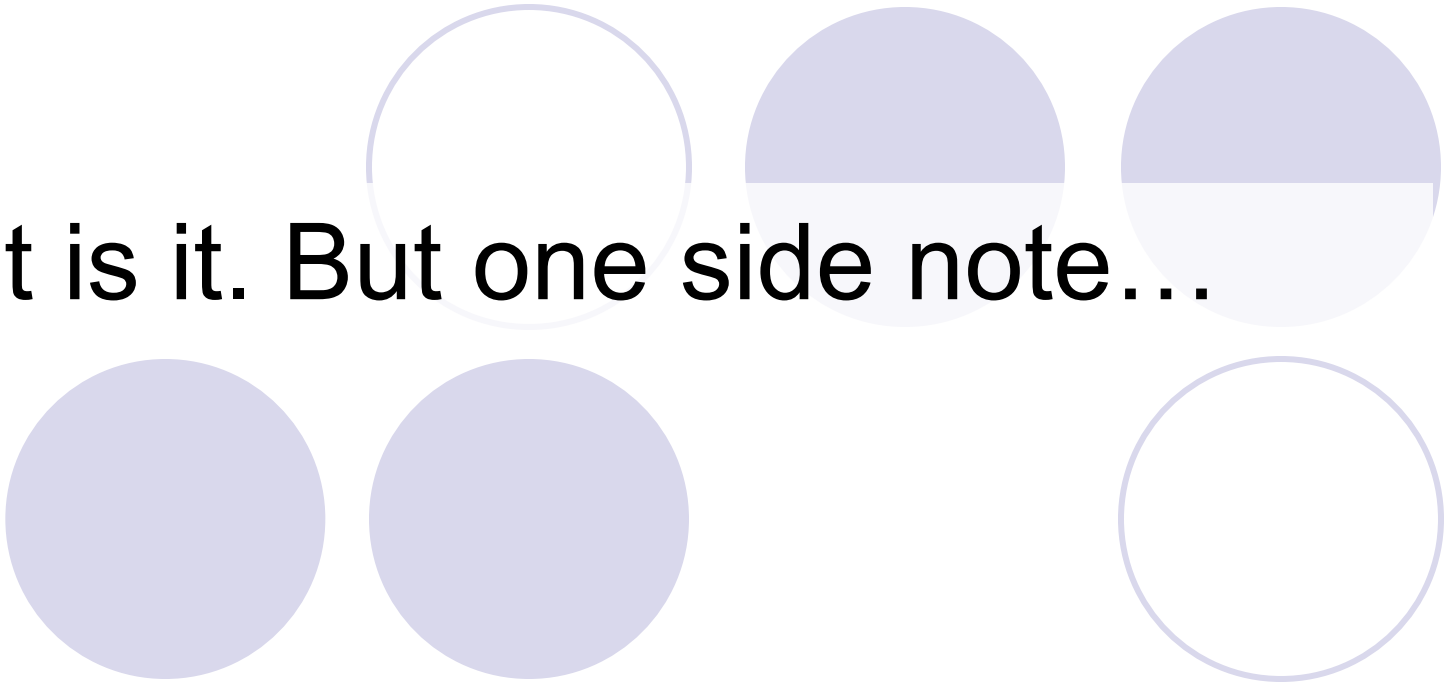
Data from the paper “Unvanish”

- One machine can emulate 500 Vuze identities at the same time
- Selectively store the data which looks like Vanish encryption key, guessing it from the size
- Able to launch the attack using only 10 Amazon EC2 instances, only less than $5000 a year
- Decrypt 100% VDO instances in the default security setting, which N=10, threshold = 70%; decrypt 79% VDO, which N=50, threshold = 90%

Comments on Sybil attacks on our paper “Vanish”

- Vuze DHT has planned an upgrade to guard against Sybil attack
  - However, from “Unvanish”, the Vuze DHT has not yet implement this upgrade. Vuze DHT has a different goal in mind than Vanish.
- Use openDHT or multiple DHT or design its own DHT system

That is it. But one side note…



Could data sanitization be a problem?

- Could the browser or the OS cached the decrypted copy?
- And we need "*secure methods for overwriting data on disk [31], encrypting virtual memory [50], and leveraging OS support for secure deallocation*" ?????

Appendix / backup slides

2007 data on number of nodes responding to probe over 48 hours

- **From "Profiling a Million User DHT"**

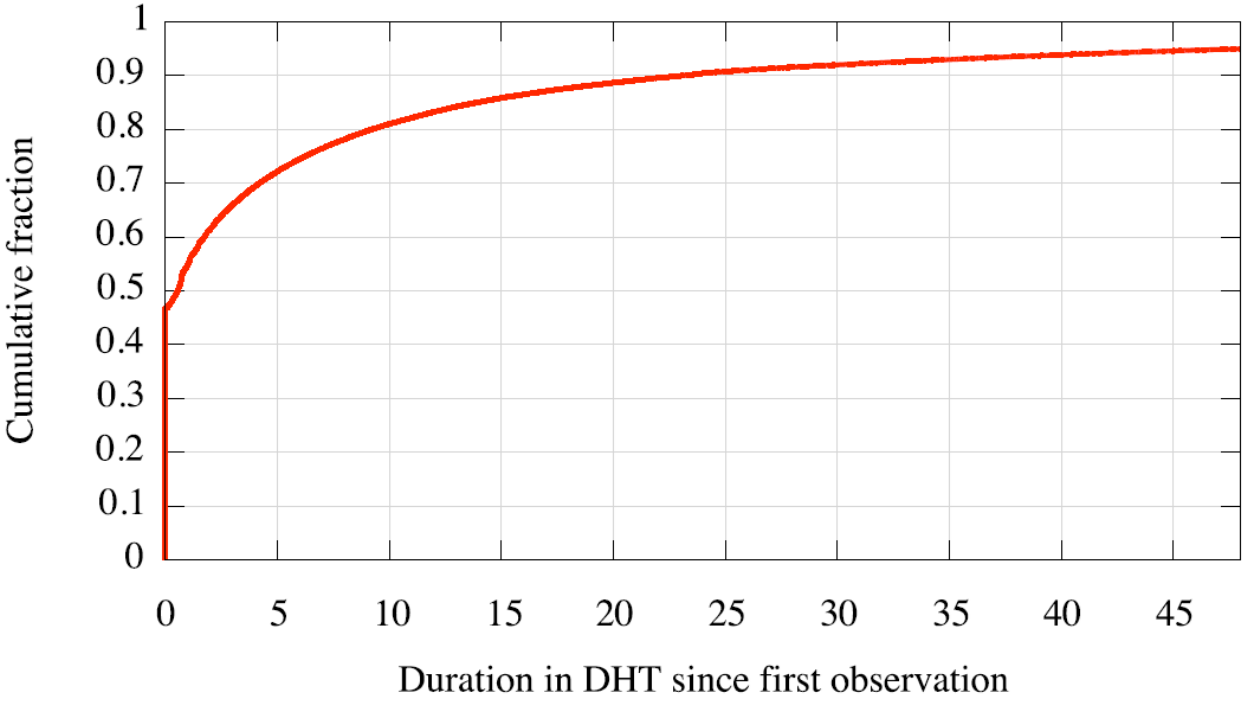


Figure 1: Freshness of DHT routing table entries and persistence for 48 hours after first observation.