Vanish:
Increasing Data Privacy with Self-Destructing Data

Roxana Geambasu
Yoshi Kohno
Amit Levy
Hank Levy

University of Washington
Outline

Part 1: Introducing Self-Destructing Data
Part 2: Vanish Architecture and Implementation
Part 3: Evaluation and Applications
Motivating Problem: Data Lives Forever

How can Ann delete her sensitive email?

- She doesn’t know where all the copies are
- Services may retain data for long after user tries to delete

---

Are "deleted" photos really gone from Facebook? Not always

When you delete embarrassing photos from sites like MySpace and Facebook, they don’t disappear immediately.
Archived Copies Can Resurface Years Later

Some time later…

Subpoena, hacking, …

Retroactive attack on archived data
The Retroactive Attack

Upload data
Copies archived
User tries to delete

months or years

Retroactive attack begins
Why Not Use Encryption (e.g., PGP)?

This is sensitive stuff. This is sensitive stuff. This is sensitive stuff. This is sensitive stuff. This is sensitive stuff. This is sensitive stuff.
Why Not Use Encryption (e.g., PGP)?

ISP

Hotmail

W

Gmail

Judge orders defendant to decrypt PGP-protected laptop

A federal judge has ordered a criminal defendant to decrypt his hard drive by typing in a password.

UK police can now demand encryption keys

People in the UK who encrypt their data are now obliged by law to give up the encryption keys to law enforcement officials if requested under the Regulation of Investigatory Powers Act 2000 (RIPA).
Why Not Use a Centralized Service?

Centralized Service

“Trust us: we’ll help you delete your data on time.”

Backdoor agreement
Why Not Use a Centralized Service?

Hushmail, a longtime provider of encrypted web-based email, markets itself by saying that "not even a Hushmail employee with access to our servers can read your encrypted e-mail, since each message is uniquely encoded before it leaves your computer."

But it turns out that statement seems not to apply to individuals targeted by government agencies that are able to convince a Canadian court to serve a court order on the company.

Centralized Service

“Trust us: we’ll help you delete your data on time.”
The Problem: Two Huge Challenges for Privacy

1. Data lives forever
   - On the web: emails, Facebook photos, Google Docs, blogs, ...
   - In the home: disks are cheap, so no need to ever delete data
   - In your pocket: phones and USB sticks have GBs of storage

2. Retroactive disclosure of both data and user keys has become commonplace
   - Hackers
   - Misconfigurations
   - Legal actions
   - Border seizing
   - Theft
   - Carelessness

*The Washington Post*

Palin’s Yahoo! Account Hacked

A group of computer hackers said yesterday they accessed a Yahoo! e-mail account of Alaska Gov. Sarah Palin, the Republican vice presidential nominee, publishing some of her private communications [...]*
The Problem: Two Huge Challenges for Privacy

1. **Data lives forever**
   - On the web: emails, Facebook photos, Google Docs, blogs, …
   - In the home: disks are cheap, so no need to ever delete data
   - In your pocket: phones and USB sticks have GBs of storage

2. **Retroactive disclosure of both data and user keys has become commonplace**
   - Hackers
   - Misconfigurations
   - Legal actions
   - Border seizing
   - Theft
   - Carelessness

---

*The New York Times*

F.B.I. Gained Unauthorized Access to E-Mail

WASHINGTON — A technical glitch gave the F.B.I. access to the e-mail messages from an entire computer network — perhaps hundreds of accounts or more — instead of simply the lone e-mail […]
The Problem: Two Huge Challenges for Privacy

1. **Data lives forever**
   - On the web: emails, Facebook photos, Google Docs, blogs, …
   - In the home: disks are cheap, so no need to ever delete data
   - In your pocket: phones and USB sticks have GBs of storage

2. **Retroactive disclosure of both data and user keys has become commonplace**
   - Hackers
   - Misconfigurations
   - Legal actions
   - Border seizing
   - Theft
   - Carelessness

---

The New York Times

Email Being Used More In Divorce Cases

By Mike Sachoff - Mon, 02/11/2008 - 13:05

The majority of U.S. divorce attorneys (88%) say they have seen an increase in the number of cases using electronic data as evidence during the past five years, according to a survey of American Academy of Matrimonial Lawyers (AAML).
The Problem: Two Huge Challenges for Privacy

1. Data lives forever
   - On the web: emails, Facebook photos, Google Docs, blogs, …
   - In the home: disks are cheap, so no need to ever delete data
   - In your pocket: phones and USB sticks have GBs of storage

2. Retroactive disclosure of both data and user keys has become commonplace
   - Hackers
   - Misconfigurations
   - Legal actions
   - Border seizing
   - Theft
   - Carelessness

---

*Seizing Laptops and Cameras Without Cause*

A controversial customs practice creates a legal backlash

By Alex Kingsbury

Posted June 24, 2008
Question:
Can we empower users with control of data lifetime?

Answer:
Self-destructing data
Question:
Can we empower users with control of data lifetime?

Answer:
Self-destructing data
1. Until timeout, users can read original message
1. Until timeout, users can read original message
2. After timeout, all copies become permanently unreadable
   2.1. even for attackers who obtain an archived copy & user keys
   2.2. without requiring explicit delete action by user/services
   2.3. without having to trust any centralized services
Self-Destructing Data Model

Goals of Self-Destructing Data

1. Until timeout, users can read original message
2. After timeout, all copies become permanently unreadable
   2.1. even for attackers who obtain an archived copy & user keys
   2.2. without requiring explicit delete action by user/services
   2.3. without having to trust any centralized services
Outline

Part 1: Introducing Self-Destructing Data
Part 2: Vanish Architecture and Implementation
Part 3: Evaluation and Applications
Vanish: Self-Destructing Data System

- Traditional solutions are not sufficient for self-destructing data goals:
  - PGP
  - Centralized data management services
  - Forward-secure encryption
  - …

- Let’s try something completely new!

Idea:
Leverage P2P systems
P2P 101: Intro to Peer-To-Peer Systems

- A system composed of individually-owned computers that make a portion of their resources available directly to their peers without intermediary managed hosts or servers. [~wikipedia]

Important P2P properties (for Vanish):
- **Huge scale** – millions of nodes
- **Geographic distribution** – hundreds of countries
- **Decentralization** – individually-owned, no single point of trust
- **Constant evolution** – nodes constantly join and leave
Distributed Hashtables (DHTs)

- Hashtable data structure implemented on a P2P network
  - Get and put (index, value) pairs
  - Each node stores part of the index space

- DHTs are part of many file sharing systems:
  - Vuze, Mainline, KAD
  - Vuze has ~1.5M simultaneous nodes in ~190 countries

- Vanish leverages DHTs to provide self-destructing data
  - One of few applications of DHTs outside of file sharing
How Vanish Works: Data Encapsulation

Encapsulate \((\text{data}, \text{timeout})\)

\[ C = E_K(\text{data}) \]

World-Wide DHT

Random indexes

Secret Sharing \((M \text{ of } N)\)

Ann

Vanish
How Vanish Works: Data Encapsulation

Encapsulate \((\text{data}, \text{timeout})\)

\[
C = E_K(\text{data})
\]

\[
\text{Secret Sharing (M of N)}
\]

\[
\text{Random indexes}
\]

\[
\text{World-Wide DHT}
\]

\[
\begin{align*}
&k_1 \\
&k_2 \\
&k_3 \\
&\vdots \\
&k_N
\end{align*}
\]
How Vanish Works: Data Encapsulation

Vanish

Encapsulate (data, timeout)

Secret Sharing (M of N)

C = E_K(data)

Random indexes

VDO = {C, L}

World-Wide DHT
How Vanish Works: Data Decapsulation

Ann

Vanish

C = E_K(data)

Secret Sharing
(M of N)

Random indexes

World-Wide DHT

L

Decapsulate
(VDO = {C, L})

Carla

Encapsulate
(data, timeout)

Vanish Data Object
VDO = {C, L}

Random indexes

L

VDO = {C, L}

k_1

k_2

k_3

... 

k_N
How Vanish Works: Data Decapsulation

Encapsulate \((data, \text{timeout})\)

Ann

Vanish Data Object \(VDO = \{C, L\}\)

World-Wide DHT

Decapsulate \((VDO = \{C, L\})\)

Carla

Secret Sharing (M of N)

\(C = E_K(data)\)

Random indexes

data = \(D_K(C)\)

Secret Sharing (M of N)

Random indexes

Vanish Data Object \(VDO = \{C, L\}\)
How Vanish Works: Data Decapsulation

Vanish

Encapsulate \((\text{data}, \text{timeout})\)

Ann

\(\text{VDO} = \{C, L\}\)

Vanish Data Object

Carla

Decapsulate \((\text{VDO} = \{C, L\})\)

\(\text{data}\)

Secret Sharing \((\text{M of N})\)

\(C = E_k(\text{data})\)

World-Wide DHT

Random indexes

Random indexes

Secret Sharing \((\text{M of N})\)

\(\text{data} = D_k(C)\)
How Vanish Works: Data Timeout

- The DHT **loses key pieces** over time
  - Natural churn: nodes crash or leave the DHT
  - Built-in timeout: DHT nodes purge data periodically

- **Key loss** makes all data copies permanently unreadable
Outline

Part 1: Introducing Self-Destructing Data
Part 2: Vanish Architecture and Implementation
Part 3: Evaluation and Applications
Evaluation

- Experiments to understand and improve:
  1. data availability before timeout
  2. data unavailability after timeout
  3. performance
  4. security

- Highest-level results:
  - Secret sharing parameters (N and M) affect availability, timeout, performance, and security
  - Tradeoffs are necessary

In the paper

Discussed next
Threat Model

- **Goal:** protect against *retroactive attacks* on old copies
  - Attackers don’t know their target until after timeout
  - Attackers may do non-targeted “pre-computations” at any time

- **Communicating parties trust each other**
  - E.g., Ann trusts Carla not to keep a plain-text copy
## Attack Analysis

<table>
<thead>
<tr>
<th>Retroactive Attack</th>
<th>Defense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obtain data by legal means (e.g., subpoenas)</td>
<td>P2P properties: <em>constant evolution, geographic distribution, decentralization</em></td>
</tr>
<tr>
<td>Gmail decapsulates all VDO emails</td>
<td>Compose with traditional encryption (e.g., PGP)</td>
</tr>
<tr>
<td>ISP sniffs traffic</td>
<td>Anonymity systems (e.g., Tor)</td>
</tr>
<tr>
<td>DHT eclipse, routing attack</td>
<td>Defenses in DHT literature (e.g., constraints on routing table)</td>
</tr>
<tr>
<td>DHT Sybil attack</td>
<td>Defenses in DHT literature; Vuze offers some basic protection</td>
</tr>
<tr>
<td>Intercept DHT “get” requests &amp; save results</td>
<td>Vanish obfuscates key share lookups</td>
</tr>
<tr>
<td>Capture key pieces from the DHT (pre-computation)</td>
<td>P2P property: <em>huge scale</em></td>
</tr>
<tr>
<td>More (see paper)</td>
<td></td>
</tr>
</tbody>
</table>
## Attack Analysis

<table>
<thead>
<tr>
<th>Retroactive Attack</th>
<th>Defense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obtain data by legal means (e.g., subpoenas)</td>
<td>P2P properties: constant evolution, geographic distribution, decentralization</td>
</tr>
<tr>
<td>Gmail decapsulates all VDO emails</td>
<td>Compose with traditional encryption (e.g., PGP)</td>
</tr>
<tr>
<td>ISP sniffs traffic</td>
<td>Anonymity systems (e.g., Tor)</td>
</tr>
<tr>
<td>DHT eclipse, routing attack</td>
<td>Defenses in DHT literature (e.g., constraints on routing table)</td>
</tr>
<tr>
<td>DHT Sybil attack</td>
<td>Defenses in DHT literature; Vuze offers some basic protection</td>
</tr>
<tr>
<td>Intercept DHT “get” requests &amp; save results</td>
<td>Vanish obfuscates key share lookups</td>
</tr>
<tr>
<td>Capture key pieces from the DHT (pre-computation)</td>
<td>P2P property: huge scale</td>
</tr>
<tr>
<td>More (see paper)</td>
<td></td>
</tr>
</tbody>
</table>
Retroactive Attacks

<table>
<thead>
<tr>
<th>Attack</th>
<th>Defense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capture any key pieces from the DHT (pre-computation)</td>
<td>P2P property: huge scale</td>
</tr>
</tbody>
</table>

- Given the huge DHT scale, how many nodes does the attacker need to be effective?
- Current estimate:
  - Attacker must join with ~8% of DHT size, for 25% capture
  - There may be other attacks (and defenses)
Vanish Applications

- Self-destructing data & Vanish support many applications

Example applications:
- **Firefox plugin**
  - Included in our release of Vanish
- Thunderbird plugin
  - Developed by the community two weeks after release 😊
- Self-destructing files
- Self-destructing trash-bin
- …
Firefox Plugin For Vanishing Web Data

- Encapsulate text in any text area in self-destructing VDOs
Firefox Plugin For Vanishing Web Data

- Encapsulate text in **any text area** in self-destructing VDOs
Firefox Plugin For Vanishing Web Data

- Encapsulate text in any text area in self-destructing VDOs
Firefox Plugin For Vanishing Web Data

- Encapsulate text in **any text area** in self-destructing VDOs

**Effect:**
Vanish empowers users with seamless control over the lifetime of their Web data.
Conclusions

- Two formidable challenges to privacy:
  - Data lives forever
  - Disclosures of data and keys have become commonplace

- Self-destructing data empowers users with lifetime control

- Vanish:
  - Combines global-scale DHTs with secret sharing to provide self-destructing data
  - Firefox plugin allows users to set timeouts on text data anywhere on the web

- Vanish ≠ Vuze-based Vanish
  - Customized DHTs, hybrid approach, other P2P systems
  - Further extensions for security in the paper

http://vanish.cs.washington.edu/