# Fuzzy Fingerprinting For Privacy-Aware Data-Loss Prevention


Danfeng (Daphne) Yao

Assistant Professor

*Department of Computer Science*

*Virginia Tech*

danfeng@cs.vt.edu

http://people.cs.vt.edu/~danfeng/


Joint work with Xiaokui Shu

Virginia Tech

July 2011, Huawei America Research Center

# Data loss incidents – intentional, accidental

Survey results reveal that 59% of ex-employees admit to stealing confidential company information [Symantec]

E.g., employees emailing sensitive content to personal Webmail accounts or

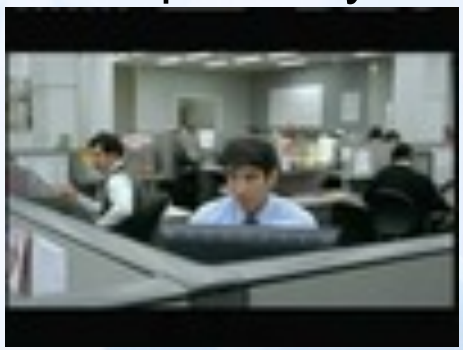
E.g., downloading it onto USB drives

*Accidental data leak*

E.g., email forwarding, web posting of sensitive data inadvertently

E.g., An Eli Lilly's lawyer sent documents to a NY Times reporter by mistake '08

REPLY-ALL by mistake http://www.youtube.com/watch?v=beF0LTvbdfw



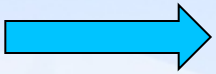
Virginia Tech

# Data Exfiltration – A Case Study

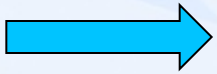
**Hydraq malware,** discovered on January 11, 2010

An Attack of Mythical Proportions. http://www.symantec.com/

Social engineering (targeted phishing email)



Drive-by download



Backdoor

Data exfiltration


- Trojan.Hydraq is a Trojan horse that opens a back door on the compromised computer (Windows OS)
- Tailored to target a small number of corporate users
  - sending a malicious document attached to an email or
  - sending a spoofed email message with a link to a malicious website

Infected machines will typically have the following components installed:

%System%\[RANDOM].dll: main file. Runs as a service and has back door capabilities

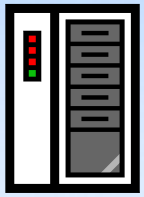
%System%\acelpvc.dll: Streams live desktop feed to the attacker

%System%\VedioDriver.dll: Helper dll for acelpvc.dll

# Multiple points where you may stop data leak

Data encryption on PC

Avoid social engineering attack



Patching

Data encryption on server

Internal servers

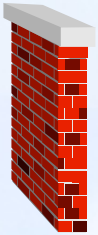
Employee

Patching

Work-place PC

Data loss prevention (DLP)

IDS/IPS



Firewall

Secure OS
e.g., memory protection
Secure applications
e.g., Email authentication
e.g., Browser sandbox

Server

Internet

An organization

Virginia Tech

# Data loss and prevention approaches

Network-based prevention – to inspect traffic content for unauthorized transmission of sensitive data

Host-based prevention – to monitor and control data transfer to physical devices

How to minimize the exposure of sensitive data during inspection?

Our solution: inspection based on special irreversible digests

Virginia Tech.

http://datalossdb.org/

http://www.mydlp.org/ being developed

# Data Loss Prevention in the Cloud

Problem: Data leaked through human errors, malware, insiders

e.g., Hydraq malware, Wikilea

Solution: DLP

( ), ...rs

...es (CISCO, Hu

Challenge: To preserve data privacy

Issues: providers' trustworthiness, cloud's security

data owner does not reveal sensitive data to providers

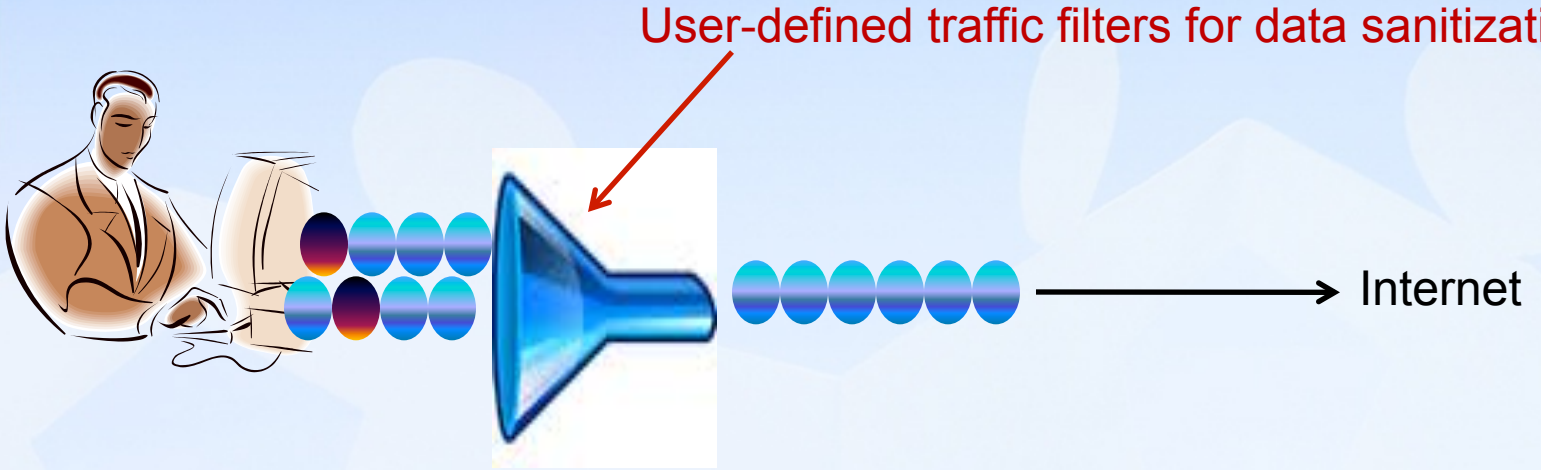
Our algorithm: Providers inspect traffic for patterns, without knowing what sensitive data is

Provisional patent filed on this technology by Virginia Tech (Mar 2011)

Virginia Tech

# Other DLP deployment scenarios and data exposure

- Personal firewall on PC



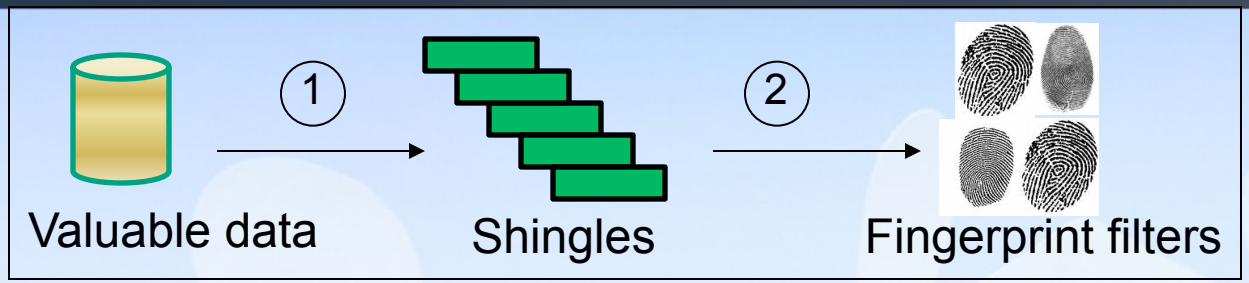

- Local area networks of organizations

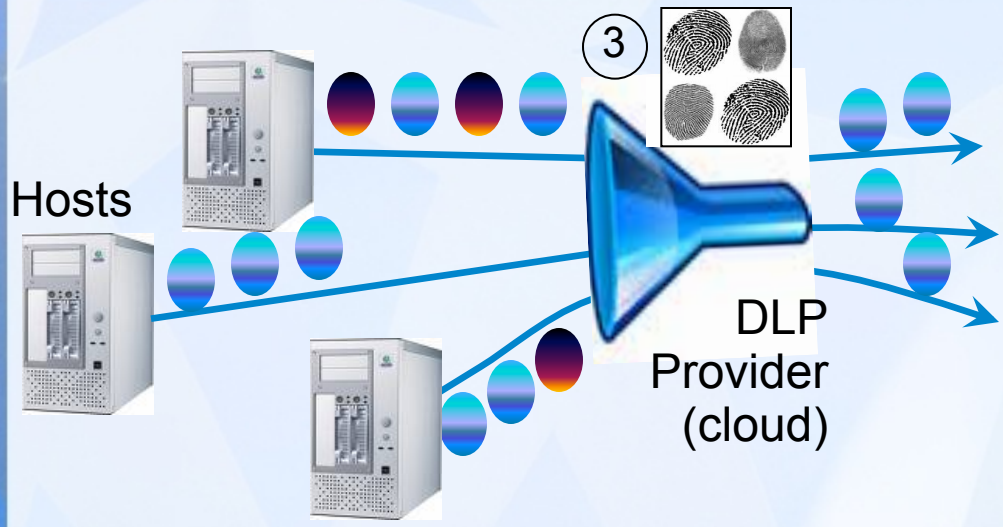
  To deploy DLP filter at gateway routers

  Data may be of any size or type

**Need to avoid exposing sensitive data at filters**

# Overview of Our Architecture




Types of players:

1. Data owner
2. User
3. DLP provider **(honest-but-curious)**

Sensitive data

Shingles are a sequence of fixed-size contiguous words (q-gram);

**Mozilla is aware of a critical vulnerability**

**Mozilla is**
**ozilla is a**
**zilla is aw**
**illa is awa**

Virginia Tech

## Our Security/Privacy Goal:

Data owner delegates DLP provider to detect data leak caused by malicious attackers (i.e., malware infecting hosts or insider),

without revealing sensitive data to provider.

Assume that the traffic is not encrypted;

Host-based detection needed for encrypted traffic.

# An example of fingerprints on shingles of two similar messages

## Sensitive data to be protected

Critical vulnerability in Firefox 3.5 and Firefox 3.6
10.26.10 - 02:30pm
Update (Oct 27, 2010 @ 20:12):
A fix for this vulnerability has been released for Firefox and Thunderbird users.
Firefox 3.6.12 and 3.5.15 security updates now available
Thunderbird 3.1.6 and 3.0.10 security updates now available
Issue:
Mozilla is aware of a critical vulnerability affecting Firefox 3.5 and Firefox 3.6 users. We have received reports from several security research firms that exploit code leveraging this vulnerability has been detected in the wild.
Impact to users:
Users who visited an infected site could have been affected by the malware through the vulnerability. The trojan was initially reported as live on the Nobel Peace Prize site, and that specific site is now being blocked by Firefox's built-in malware protection. However, the exploit code could still be live on other websites.

10 smallest fingerprints: (**4482868, 5207155, 5538456, 16590970, 18891336, 28959745, 29523072, 30605011, 46912339, 47163843**)
Total fingerprints set size: **756**
SHA-1:
**3c1e4ca6505e5d307cfe105104233e1b82b39b33**

## Captured payload in outbound traffic

Critical vulnerability in Firefox 3.5 and Firefox 3.6
10.26.10 - 02:30pm
Update (Oct 27, 2010 @ 20:12):<br />
A fix for this vulnerability has been released for Firefox and Thunderbird users. Firefox 3.6.12 and 3.5.15 security updates now available<br /> Thunderbird 3.1.6 and 3.0.10 security updates now available Issue:<br />
Mozilla is aware of a critical vulnerability affecting Firefox 3.5 and Firefox 3.6 users. We have received reports from several security research firms that exploit code leveraging this vulnerability has been detected in the wild.
Impact to users:<br />
Users who visited an infected site could have been affected by the malware through the vulnerability. The trojan was initially reported as live on the Nobel Peace Prize site, and that specific site is now being blocked by Firefox's built-in malware protection. However, the exploit code could still be live on other websites.

10 smallest fingerprints: (**4482868, 5538456, 16590970, 18891336, 28959745, 29523072, 30605011, 46912339, 47163843, 60018488**)
Total fingerprints set size: **806**
SHA-1:
**e86d8771e82c613706fab67adbee2e2b0e8e762e**

# Rabin's Fingerprint

$$A(t) = a_1 t^{m-1} + a_2 t^{m-2} + \cdots + a_m$$

$$f(A) = A(t) \bmod P(t)$$

A=($a_1$, $a_2$, …, $a_m$) is a binary string

P is a irreducible polynomial.

```
          1110
      --------
101 ) 110101
        101
        ---
         11101
         101
         ---
          1001
          101
          ---
           011
```

***An example***

110101 mod 101 = 11 is equivalent to:

$X^5 + X^4 + X^2 + 1 \bmod X^2 + 1 = X + 1$

**Advantages: oneway, fast**

In binary:

- 1 – 0 = 1
- 0 – 1 = -1 = 1
- So it is just XOR operation

Virginia Tech.

# A naïve data-loss detection protocol

1. *Data pre-processing* -- data owner computes digests; and reveals to DLP provider a subset of the digests
   - e.g., to select a smallest 20 fingerprints to release
2. *Traffic pre-processing* – DLP provider collects outbound network traffic of data owner; and computes digests of packets
3. *Inspection* – DLP provider alerts data owner if traffic digests match data digests

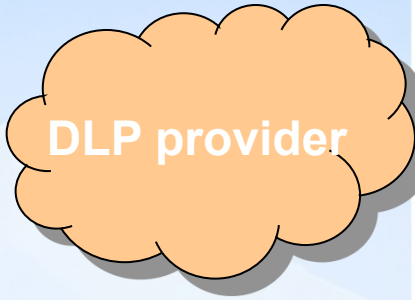
   e.g., based on pre-defined threshold

**Sensitivity test** $\frac{\text{Number of sensitive-data fingerprints per packet}}{\text{Total fingerprints per packet}}$

# The naïve detection leaks info to DLP provider if there is a match ☹

Company A has a secret recipe:
fish with garlic bake 20-min 450F

2. Fingerprints 375835 and 949609 →




1. Compute digest = f(data)

| 8-gram | fingerprint |
|---|---|
| **Fish wit** | **375835** |
| ish with | 907948 |
| sh with | 867025 |
| h with g | 098600 |
| with ga | 114534 |
| **with gar** | **949609** |
| … | … |

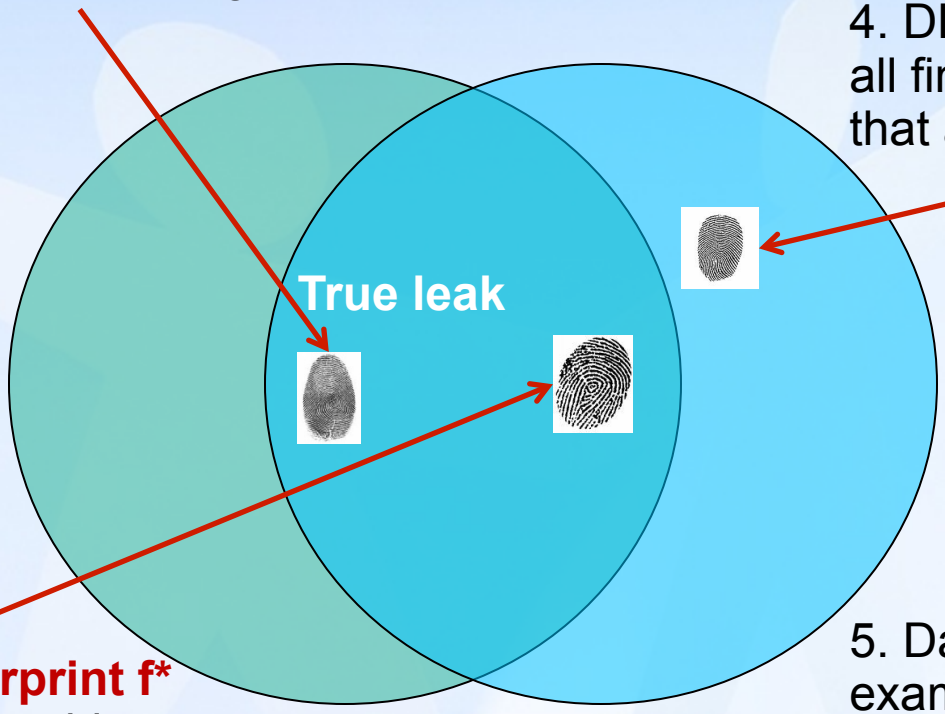
3. Monitor the traffic of A

4. Find a packet whose fingerprints contain 375835 and 949609

DLP has the content of the packet,
Thus learns the secret recipe ☹

# Our solution: fuzzy fingerprint – to hide sensitive fingerprint in a crowd

1. Original sensitive fingerprint f




4. DLP provider alerts all fingerprints of traffic that are close to f*

3. **Fuzzy fingerprint f*** given to DLP provider

5. Data owner examines alerts for **true leaks**

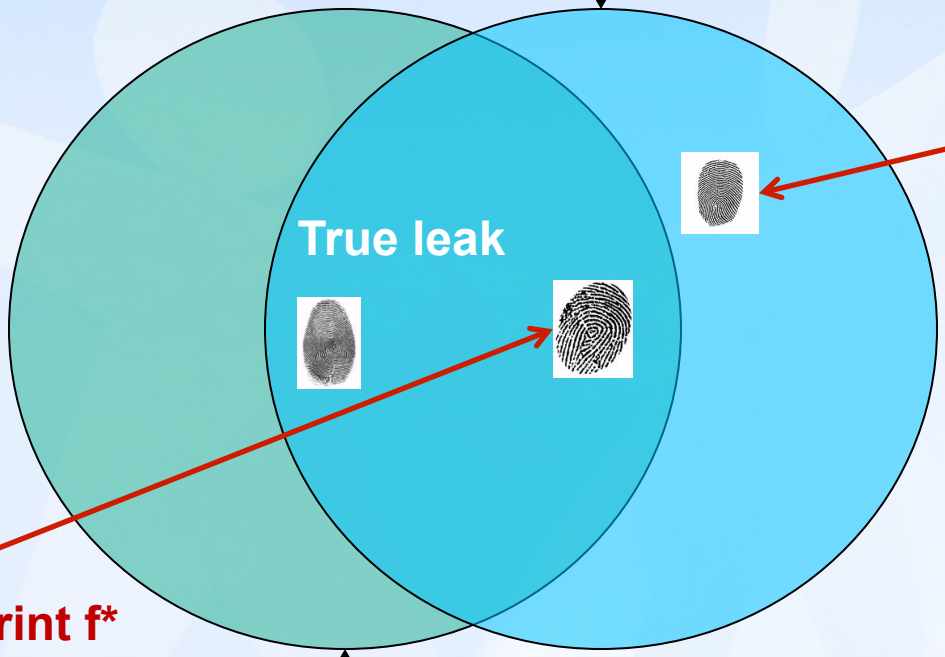
2. Perturb f by randomizing least significant bits

Similar to the k-anonymity in relational DB

Virginia Tech

# Hide fingerprint in a crowd

How big is the crowd?




Data owner: how to perturb the sensitive fingerprint?

# Fuzzy length and fuzzy set

**Fuzzy length**

Given a fingerprint f, fuzzy length d is the number of the least significant bits in f that may be perturbed by the data owner, and d is less than the degree of the polynomial used to generate the fingerprint.

**Fuzzy set**

Given a fuzzy length d, and a collection of fingerprints, the fuzzy set S(f,d) of a fingerprint f is the number of distinct fingerprints in the collection whose values differ from f by at most $2^d - 1$.

# Fuzzy fingerprint operations

Company A has a secret recipe:
fish with garlic bake 20-min 450F

2. Fuzzy fingerprints

DLP provider

1. Compute digest = f(data)

| 8-gram | fingerprint |
|---|---|
| **Fish wit** | **375835** |
| ish with | 907948 |
| sh with | 867025 |
| h with g | 098600 |
| with ga | 114534 |
| **with gar** | **949609** |
| … | … |

3. Monitor the traffic of A

4. Find a packet whose fingerprints contain 375835 and 949609

DLP has the content of the packet,
Thus learns the secret recipe ☹

Virginia Tech

# Fuzzy fingerprint operations

**Fuzzify:** Data owner flips an unbiased coin d times to generate the new least significant d bits in fuzzy fingerprint f*.

f* is given to DLP provider.

**Range-based detection:** a fuzzy fingerprint f of some sensitive data and a fingerprint $f_0$ from the traffic, and a fuzzy length d, the DLP provider outputs 1 (indicating possible data leak) if values of f and $f_0$ differ by at most $2^d - 1$, and 0 otherwise.

For all the candidate data-leak instances detected during the range-based detection, the DLP provider outputs the set of $(x_1, f_1), \ldots, (x_i, f_i), \ldots)$ pairs to the data owner,

**Defuzzify:** Data owner searches alerts to see if the sensitive data's fingerprint exists.

DLP provider cannot distinguish true leaks and false alarms

Virginia Tech

# Generalization – bit mask

Sensitive fingerprint f 01000101111011010111100010

Fuzzy fingerprint f* 01000101111011100010111011

Perturb least significant bits

Data owner may randomize arbitrary bit positions

Sensitive fingerprint f 01000101111011010111100010

Bit mask _+++_+++_+__+_+_+++__++_++

Bit may change

No change

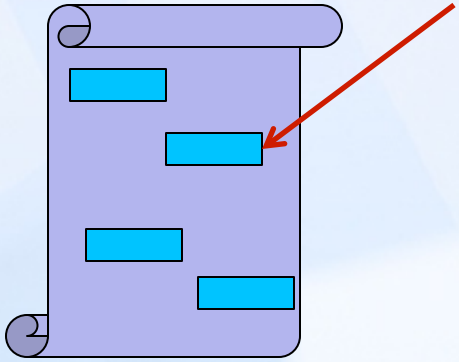
Fuzzy fingerprint f* 11000101010011010110100110

DLP provider applies bit mask to traffic; and
reports fingerprint that matches non-changing bits;

# Requirements of the digest algorithm

- *Onewayness:* Given a digest, it is computational hard to obtain the corresponding pre-image.
- *Noise tolerance:* Similar inputs yield similar digests.
  - Insertion, deletion, modification
- *Subset independence:* The partial digests are uniformly distributed across the dataset -- any part of the original data is equally likely to be selected.

Digests selected for detection need to be unbiased



**Rabin fingerprint has these properties**

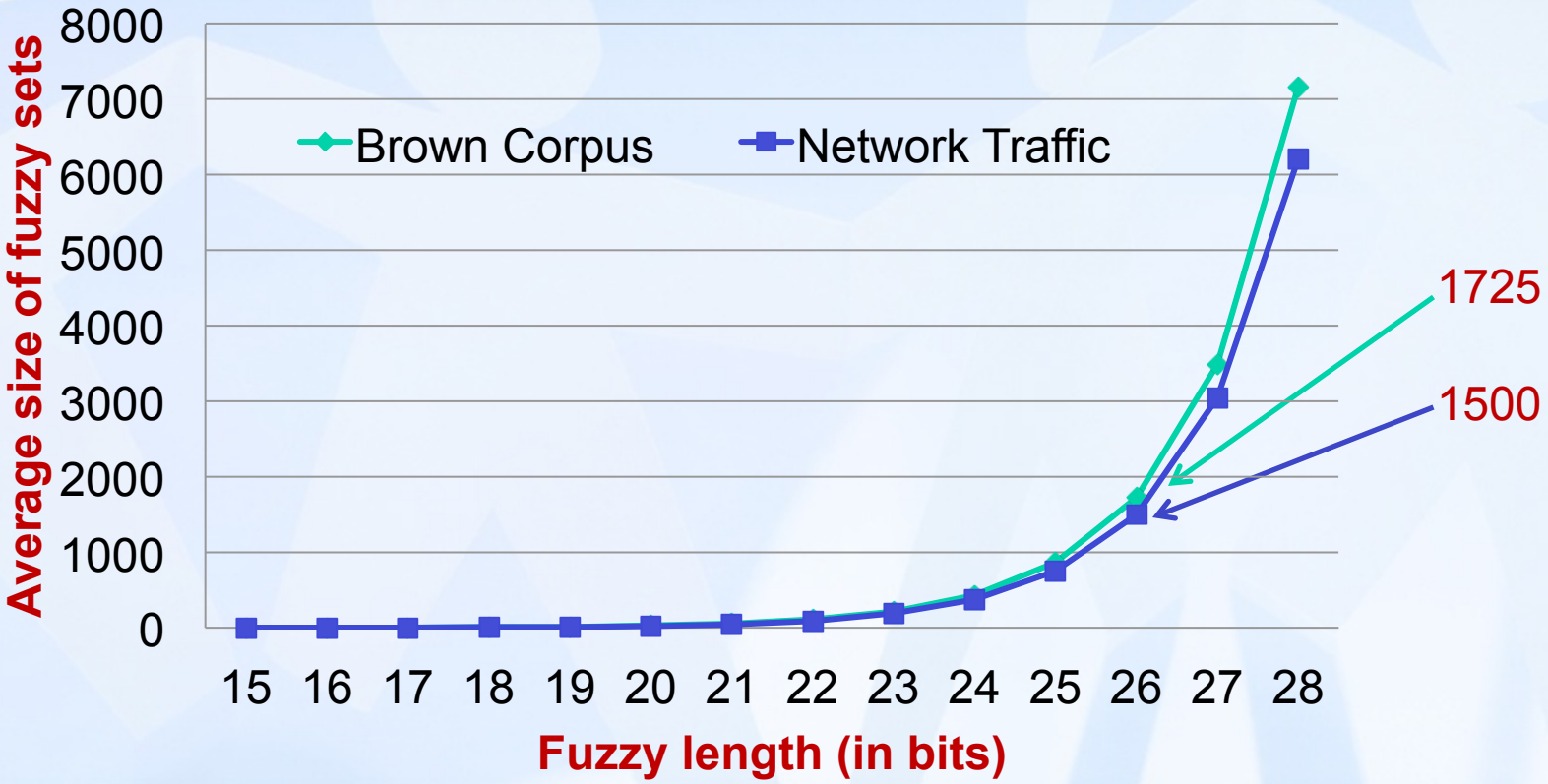
# Privacy Protection For Data Owner

**What does a semi-honest DLP provider need to do to uncover sensitive data?**

- The polynomial modulus computation
  - Adversary needs to reverse the computation to obtain the input polynomial.
- Fingerprint selection
  - Only a subset of smallest fingerprints from the sensitive data are used in the detection.
- Fuzzy fingerprint
  - Hard to distinguish sensitive fingerprint from its neighbors (assuming uniform distribution)

# Fuzzy set size

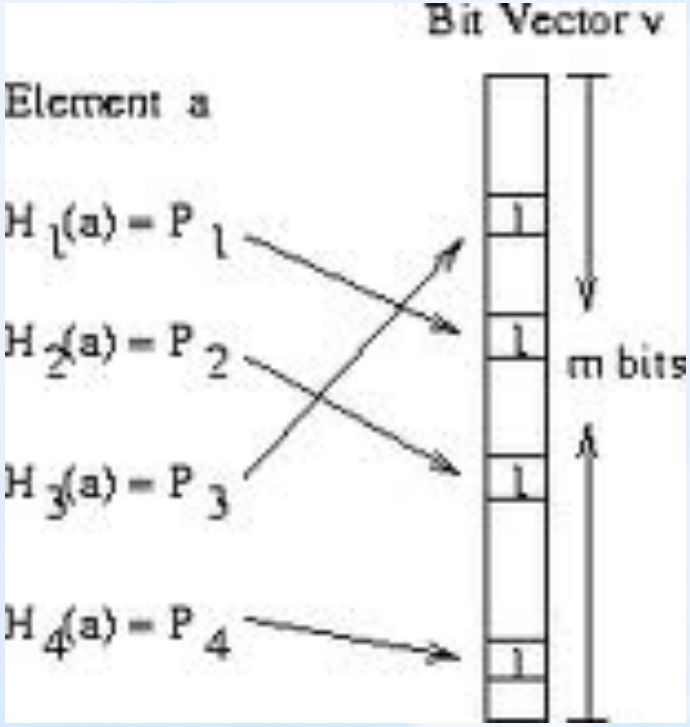
Average sizes of fuzzy sets per fingerprint in Brown Corp and network traffic using 32-bit polynomial modulus



# Implementation and experiments

Implemented all components of our framework in Python including packet collection, shingling, Rabin fingerprinting

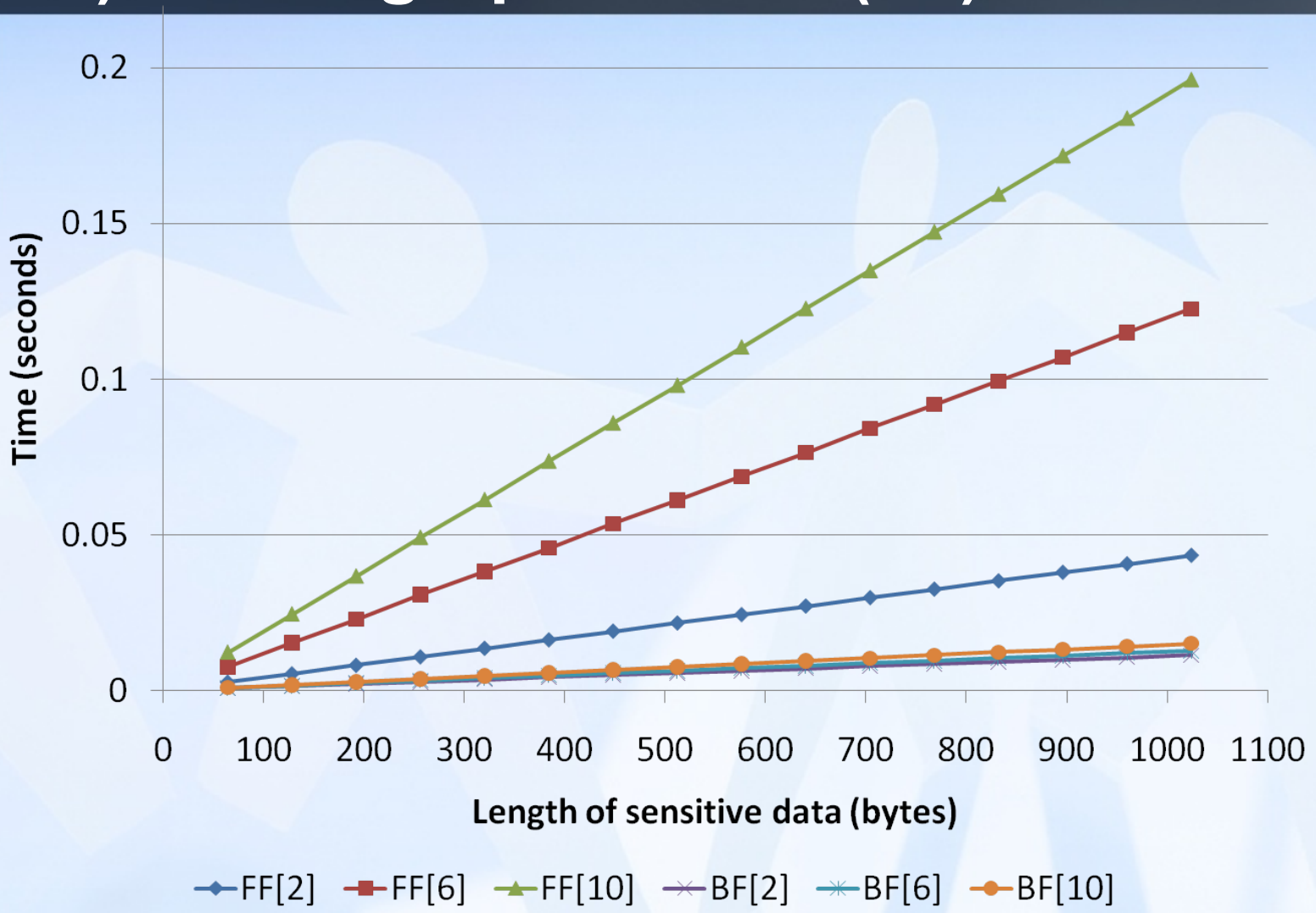
Fingerprint filter = Bloom filter + Rabin fingerprint




**Bloom filter for membership test**
**Space saving**
Pybloom library

**Experimental condition:**
8-byte shingle
32-bit polynomial
1024-byte packet payload

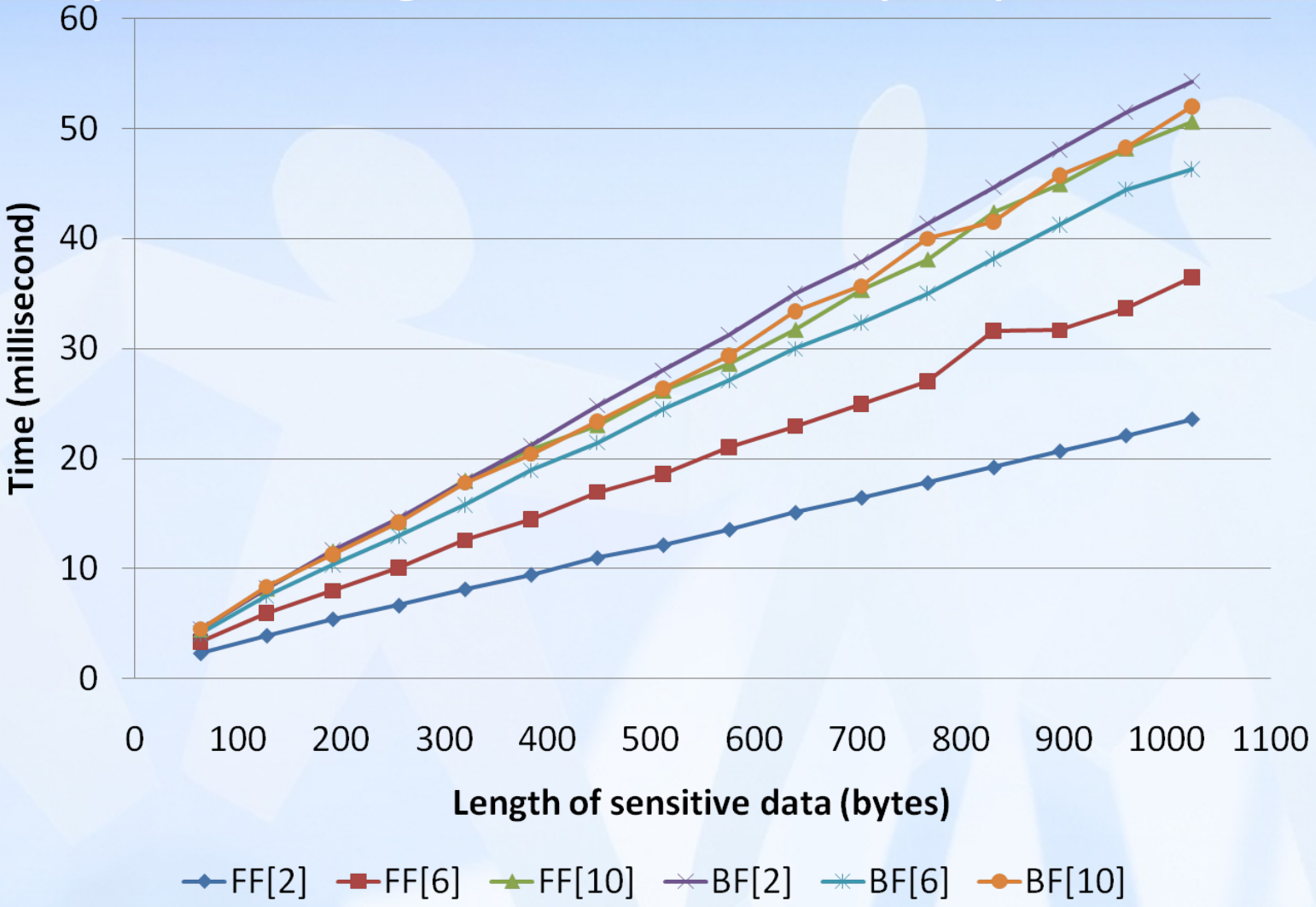
# Overhead for preparing the Bloom filter (BF) and fingerprint filter (FF)




BF is slightly faster to prepare than FF

Virginia Tech.

# Overhead of detection with Bloom filter (BF) and fingerprint filter (FF)




FF is slightly faster than BF for detection (fingerprinting is faster than hashing)

Virginia Tech

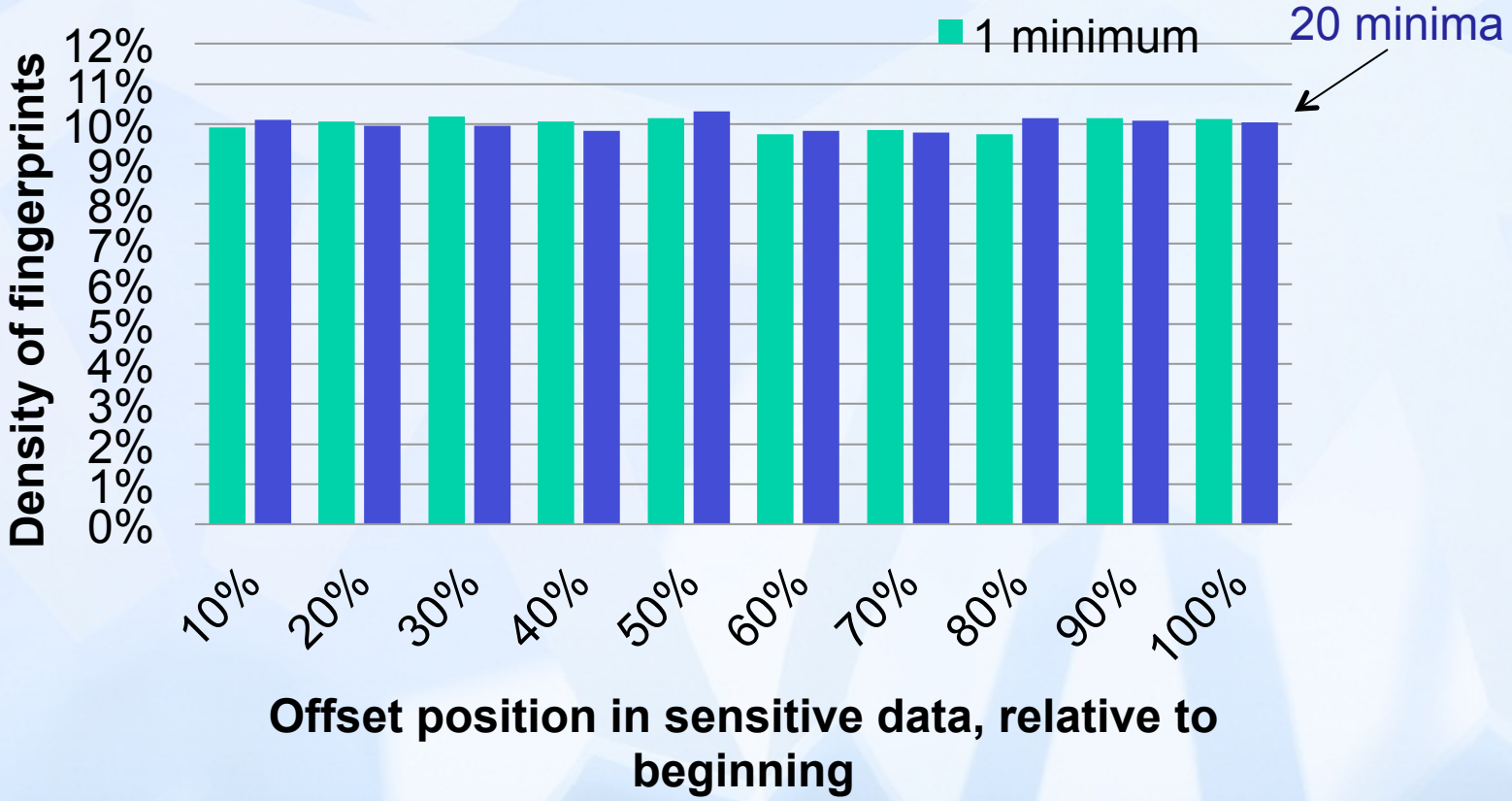
# Verifying the Subset Independent Property

$$\Pr\left(\min\{\pi(X)\} = \pi(X)\right) = \frac{1}{|X|}$$

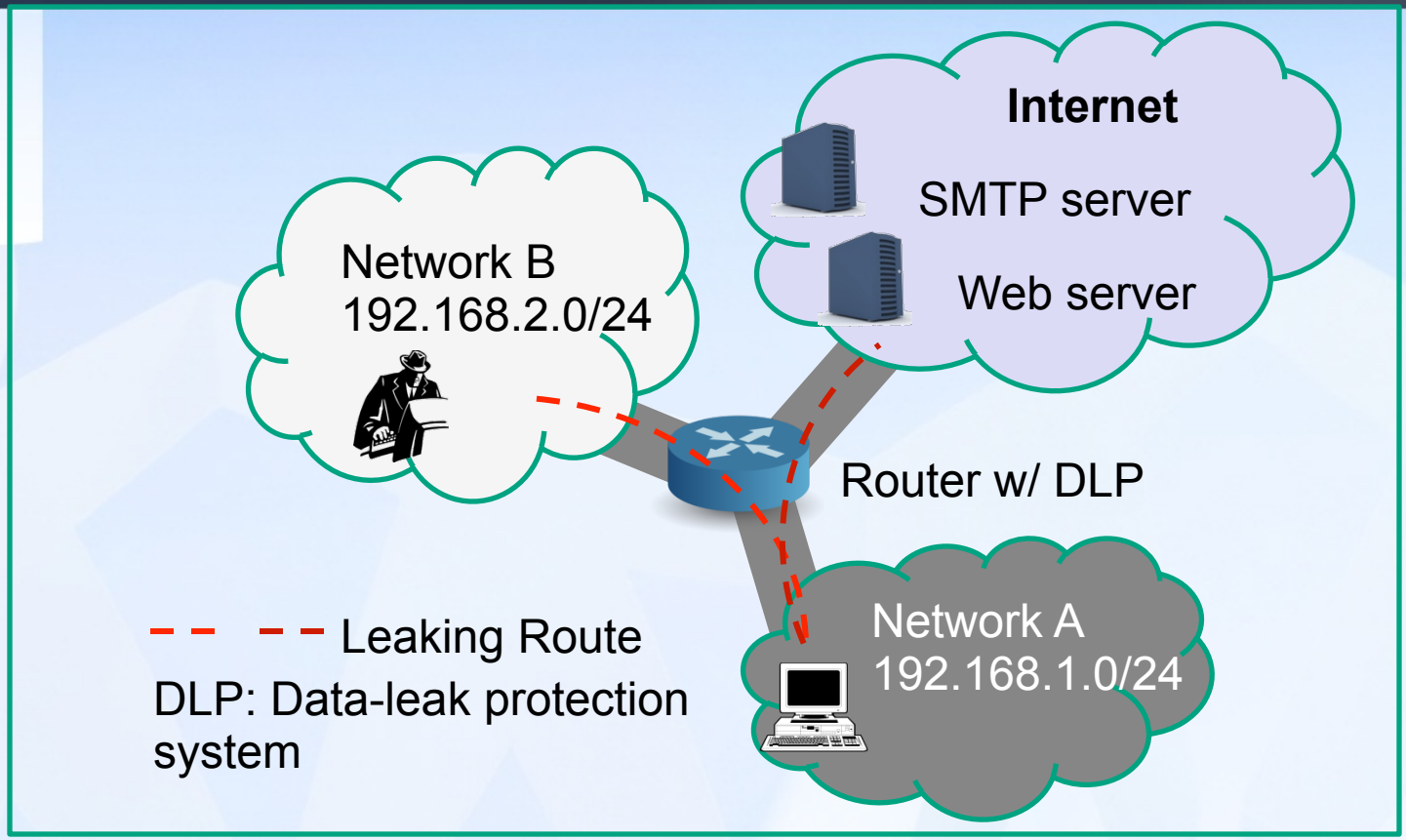
Linear transformations

$$\pi(x) = ax + b \bmod p$$

Brown corp of English



# Setup of the malware test




**We detect packets whose sensitivity values are above a threshold**

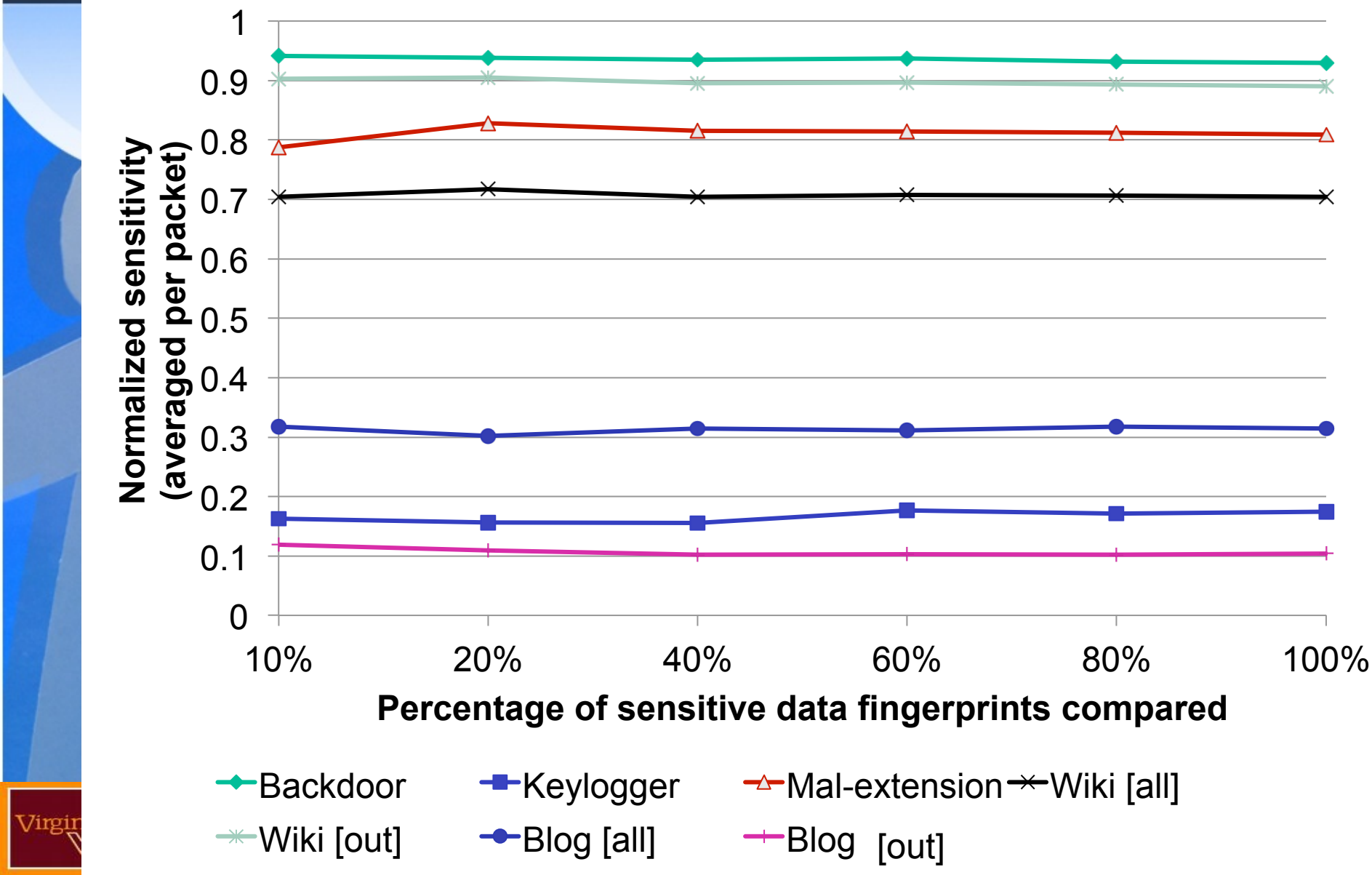
**Sensitivity test:** $\dfrac{\text{Number of sensitive-data fingerprints per packet}}{\text{Total fingerprints per packet}}$

# Malware experiments

| Leaking Methods | Protocol | Traffic | # of sensitive pkt found | Maximum sensitivity | Average sensitivity in sensitive pkts |
|---|---|---|---|---|---|
| **Backdoor** | TCP | Out | 19 | 0.97 | 0.93 |
| **Keylogger** | SMTP | Out | 3 | 0.23 | 0.18 |
| **Malicious Browser Extension** | SMTP | Out | 20 | 0.97 | 0.81 |
| **Wiki System (MediaWiki)** | HTTP | All | 41 | 0.97 | 0.70 |
| | | Out | 20 | 0.97 | 0.89 |
| **Blog System (WorldPress)** | HTTP | All | 37 | 0.95 | 0.31 |
| | | Out | 22 | 0.25 | 0.10 |

Virginia Tech

# Detection rates vs. size of partial fingerprint sets used



# Noises in traffic and their impact on detection

Original data:

A computer, called a [[router]], is provided with an interface to each network. It forwards [[packet (information technology)|packets]] back and forth between them.<ref>RFC 1812

Localized noise ☺ -- shingles are tolerant to local noises

A computer, called a [[router]], is provided with an interface to each network. It forwards [[packet (information technology)|packets]] back and forth between them.<ref>RFC 1812

Pervasive noise ☹

A+computer%2C+called+a+%5B%5Brouter%5D%5D%2C+is+provided+with+an+interface+to+each+network.+It+forwards+%5B%5Bpacket+%28information+technology%29%7Cpackets%5D%5D+back+and+forth+between+them.%26lt%3Bref%26gt%3BRFC+1812

# Summary on fuzzy fingerprint for data loss protection

- Detection rates do not decrease much with fewer fingerprints ☺
  - Even when 7 fingerprints used
  - Better privacy for data owner, revealing less info to provider
- Noise tolerance if local data features are preserved
  - E.g., Wiki
  - Pervasive noise destroys patterns, e.g., Blog
    - Shorter shingles increase false positives
- Set intersection based tests are very fast
  - Faster than Bloom filter and fingerprint filter
- Experimentally validate min-wise independence
  - Allowing the use of partial fingerprints for detection

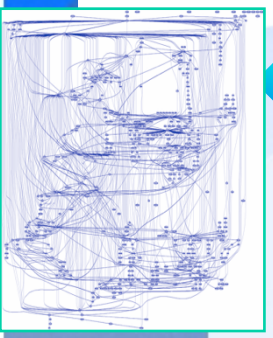
Our work provides the first privacy-aware data loss protection solution

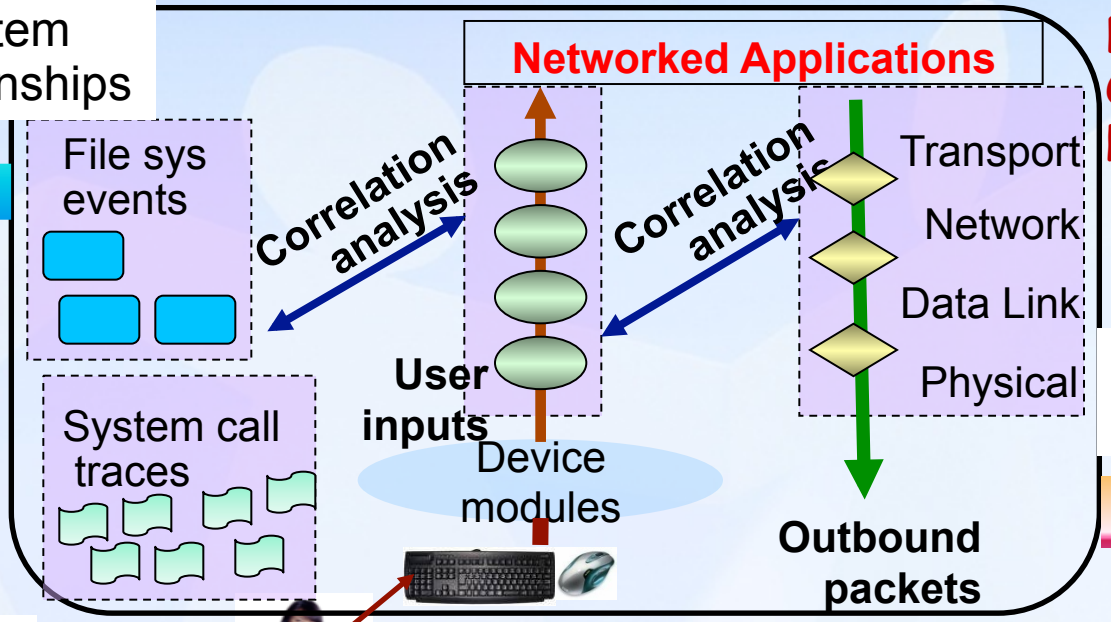
# To ensure system integrity and provide forensic analysis

## With our human-centric approach – An overview

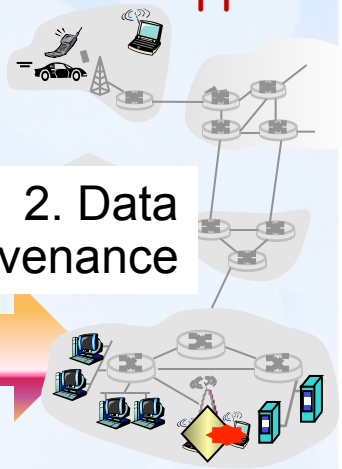
Our Techniques:
Crypto & algorithm
Data analysis
OS engineering
Hardware support

1. Mining system causal relationships



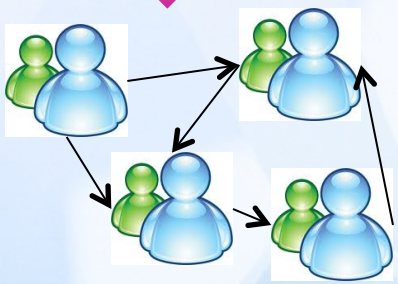



2. Data provenance

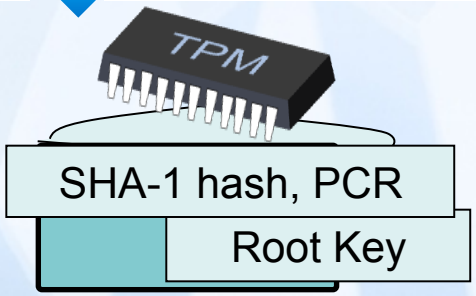


3. Visual analytics & forensics

5. Collaborative security



4. Hardware root-of-trust, device security




Virginia Tech

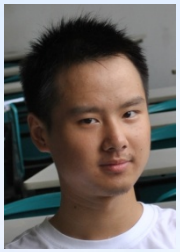
# Personnel and Select Publications from Yao Group

| | | | |
|---|---|---|---|
| 1. | Stefan, Wu, Yao, & Xu | *ACNS '10* | *System integrity* |
| 2. | Butler, Xu, & Yao | ACNS '11 | DNS bot |
| 3. | Xiong, et al | ICICS *'09* | *Malware detection* |
| 4. | Thompson & Yao | *ASIACCS '09* | *Graph data privacy* |
| 5. | Zarandioon, Yao & Ganapathy | *ACSAC '08* | *Crypto in browser* |
| 6. | Stefan & Yao **(Best Paper Award)** | *CollaborateCom '10* | *Keystroke security* |

VT Ph.D.s


Kui Xu


Huijun Xiong


Johnny Shu


Tony Zhang


Hussain Almohri


Karim Elish

PhD
Rutgers


Saman Zarandioon

Funding Sources:

- NSF CAREER, ARO, DHS, VT ICTAS

Thank you very much!

danfeng@cs.vt.edu

Virginia Tech