Exploiting Leakage in Searchable Encryption and Machine Learning

Covering joint work with:
David Cash, Paul Grubbs, Jason Perry  (Searchable encryption)
Matthew Fredrikson, Eric Lantz, Simon Lin, David Page, Somesh Jha  (ML)
Plaintext keyword search

The attached contract is ready for signature. Please print 2 documents and have Atmos ...

Keyword stemming

Upload documents

Search: “contract”

Email client

Keyword | Documents
--- | ---
contract | 1, 7
signatur | 8, 9, 1, 15, 200

Email storage provider
**Appended-PRF Searchable Encryption**

Email client

Encrypt plaintext & keyed hash of keywords

Upload encrypted documents

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_K(\text{contract})$</td>
<td>1, 7</td>
</tr>
<tr>
<td>$H_K(\text{signature})$</td>
<td>8, 9, 1, 15, 200</td>
</tr>
</tbody>
</table>

Email storage provider
Appended-PRF Searchable Encryption

Encrypt plaintext & keyed hash of keywords

Email client

Upload encrypted documents

Search: “7813fed”

Legacy compatible:
Works with existing plaintext storage interfaces

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>7813fed</td>
<td>1, 7</td>
</tr>
<tr>
<td>456abc3</td>
<td>8, 9, 1, 15, 200</td>
</tr>
</tbody>
</table>

Email storage provider
Two more schemes to consider

(2) Unordered appended-PRFs

Randomize order of PRF values

<table>
<thead>
<tr>
<th>The attached contract is ready for signature. Please print 2 documents and have Atmos ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_k$(contract)</td>
</tr>
<tr>
<td>$H_k$(ready)</td>
</tr>
<tr>
<td>$H_k$(attach)</td>
</tr>
</tbody>
</table>

(3) Encrypted index

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_k$(contract)</td>
<td></td>
</tr>
<tr>
<td>$H_k$(signatur)</td>
<td></td>
</tr>
</tbody>
</table>

Encrypt each document list under keyword-specific key
Qualitative comparison of schemes

Appended-PRF scheme used in industry

Unordered appended-PRF used in research literature

Mimesis Aegis [Lau et al. 2014]
ShadowCrypt [He et al. 2014]

Encrypted index in literature & starting to appear in industry

[Cash et al. 2014]
Qualitative comparison of schemes

Appended-PRF scheme used in industry

Unordered appended-PRF used in research literature

Encrypted index in literature & starting to appear in industry

Ease of deployment

Provable security claims
Leakage-abuse attacks

All searchable encryption leaks information about plaintexts and queries. Appended-PRF case:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_K$(contract)</td>
<td>1, 7</td>
</tr>
<tr>
<td>$H_K$(signatur)</td>
<td>8, 9, 1, 15, 200</td>
</tr>
</tbody>
</table>

Upload encrypted documents

Search: “$H_K$(contract)”
Leakage-abuse attacks

All searchable encryption leaks information about plaintexts and queries. Appended-PRF case:

“Keyword 7813fed came second in Document 1”
(Keyword location)

“Keyword 7813fed searched often”
(Search frequency)

“Document 1 and 7 both contain 7813fed”
(Co-occurrence relationships)

Upload encrypted documents

Search: “7813fed”

Unordered appended-PRF: order of keywords not leaked

Encrypted index: order of keywords not leaked & leakage only after queries made

[Islam, Kuzu, Kantarcioglu – 2013]
[Cash, Grubbs, Perry, R. – 2015]
We don’t know answers to basic security questions:

- Does leakage damage confidentiality?
- How much more security does one achieve via more complex schemes?
- What adversarial capabilities are likely to arise in practice?
## Leakage-abuse attack taxonomy

<table>
<thead>
<tr>
<th>Attacker goal</th>
<th>Query recovery</th>
<th>Plaintext recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker capabilities</td>
<td>Passive</td>
<td>Observe queries and stored ciphertexts</td>
</tr>
<tr>
<td></td>
<td>Active</td>
<td>Force insertion of documents and/or queries</td>
</tr>
<tr>
<td>Document knowledge</td>
<td>Full</td>
<td>Know all plaintexts exactly</td>
</tr>
<tr>
<td></td>
<td>Partial</td>
<td>Know some plaintexts</td>
</tr>
<tr>
<td></td>
<td>Distributional</td>
<td>Know similar plaintexts</td>
</tr>
</tbody>
</table>

IKK 2013 against encrypted index:  
Query recovery  Passive  Full

Simulations with Enron email corpus:  80% of queries recoverable
We’ll come back to this
Partial plaintext recovery against appended-PRF

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>7813fed</td>
<td>1, 7</td>
</tr>
<tr>
<td>456abc3</td>
<td>8, 9, 1, 15, 200</td>
</tr>
</tbody>
</table>

Adversarial storage provider

Known email

<table>
<thead>
<tr>
<th>contract</th>
<th>file</th>
<th>today</th>
</tr>
</thead>
<tbody>
<tr>
<td>7813fed</td>
<td>18fda83</td>
<td>64a3b4 ...</td>
</tr>
</tbody>
</table>

Unknown email

<table>
<thead>
<tr>
<th>contract</th>
<th>file</th>
<th>today</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab34df</td>
<td>7813fed</td>
<td>873f63 ...</td>
</tr>
</tbody>
</table>

[Cash, Grubbs, Perry, R. – 2015]
Partial plaintext recovery against appended-PRF

Simulations with Enron email corpus
- 30,109 emails from employee sent_mail folders
- Adversary knows 20 random emails (0.06%)
- Simply match keywords in known emails to unknown

<table>
<thead>
<tr>
<th>Unknown email plaintext</th>
<th>Recovered information</th>
</tr>
</thead>
<tbody>
<tr>
<td>The attached contract is ready for signature. Please print 2 documents and have Atmos execute both and return same to my attention. I will return an original for their records after ENA has signed. Or if you prefer, please provide me with the name / phone # / address of your customer and I will Fed X the Agreement.</td>
<td>attach contract signatur pleas print 2 document have execut both same will origin ena sign prefer provid name agreement</td>
</tr>
</tbody>
</table>

[Perry, R. – 2015] [Cash, Grubbs, Perry, R. – 2015]
Randomizing hash order

Leaving hashes in document order makes attack easy

Simple change: randomize order of hashes to leak less information
(sort by hash value)

<table>
<thead>
<tr>
<th>Known email</th>
<th>7813fed</th>
<th>18fda83</th>
<th>64a3b4 ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown email</td>
<td>ab34df</td>
<td>7813fed</td>
<td>873f63 ...</td>
</tr>
</tbody>
</table>

contract file today

Known email

Unknown email

contract
Randomizing hash order

Leaving hashes in document order makes attack easy

Simple change: randomize order of hashes to leak less information
(sort by hash value)

<table>
<thead>
<tr>
<th>Contract</th>
<th>File</th>
<th>Today</th>
</tr>
</thead>
<tbody>
<tr>
<td>???</td>
<td>???</td>
<td>???</td>
</tr>
</tbody>
</table>

| Known email | 18fda83 | 64a3b4 | 7813fed | ...
|-------------|--------|--------|---------|
| Unknown email | ab34df | 7813fed | 873f63 | ...

Order issue left implicit in prior work
Mimesis Aegis: randomizes order due to Bloom filter
ShadowCrypt: implementation randomizes order, paper does not discuss
Chosen-email attacks

Email client

\[
\begin{array}{c}
89123fdbf32a665befg8819890fbacda \\
4320182321a1343187fabaedf3140fba \\
H_K(\text{signatur}) \quad H_K(\text{contract})
\end{array}
\]

Insert new email

Send victim an email

To: victim@victim.com
From: sally@sally.net
Contract signature

Adversarial storage provider

Plaintext recovery
Active Distributional

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_K(\text{contract})$</td>
<td>1, 7</td>
</tr>
<tr>
<td>$H_K(\text{signatur})$</td>
<td>8, 9, 1, 15, 200</td>
</tr>
</tbody>
</table>
Chosen-email attacks

Email client

- 89123fdbf32a665befg8819890fbacda
- 4320182321a1343187fabaedf3140fba
- 456abc3 7813fed

Insert new email

Send victim an email
To: victim@victim.com
From: sally@sally.net

Contract signature

Plaintext recovery
Active
Distributional

Keyword | Documents
--- | ---
7813fed | 1, 7
456abc3 | 8, 9, 1, 15, 200

Adversarial storage provider

Disambiguate 2 keywords by their expected frequency
Disambiguation performance

Related: split Enron into training and testing sets, train frequency on training
Unrelated: train on distinct email corpus (Apache corpus)
Case studies of three attacks

1. Simple attack against *appended-PRF*

2. Chosen-email attack against *unordered appended-PRF*

3. Query recovery against *encrypted index schemes*
IKK query recovery attack

Adversary knows **full plaintext corpus**
Goal is to uncover search query keywords used by client

Email client
Uniformly selects keywords to search

IKK detail expensive attack using simulated annealing to solve NP-complete problem sufficient to reveal queries

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_K(\text{contract})$</td>
<td>1, 7</td>
</tr>
<tr>
<td>$H_K(\text{signatur})$</td>
<td>8, 9, 1, 15, 200</td>
</tr>
</tbody>
</table>
We give way simpler attack

Adversary knows *full plaintext corpus*

Goal is to uncover search query keywords used by client

### Email client

Uniformly selects keywords to search

- **Search:** “$H_K(\text{contract})$”
  - [Document 1], [Document 2]

- **Search:** “$H_K(\text{signatur})$”
  - [Document 3], [Document 4], [Document 5], [Document 6]

Attacker sees number of documents returned

Many keywords appear in a unique number of documents

Disambiguate with co-occurrence relationships

### Query recovery

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_K(\text{contract})$</td>
<td>1, 7</td>
</tr>
<tr>
<td>$H_K(\text{signatur})$</td>
<td>8, 9, 1, 15, 200</td>
</tr>
</tbody>
</table>

### Adversarial storage provider

Passive

Full
IKK vs “count” attack

Subset of Enron emails (known to attacker)
Most popular x keywords considered
10% of keywords uniformly sampled and queried
Summary of leakage-abuse attacks

*Provable security must be (at least) paired with empirical security analyses*

Lots of open questions:
- Leakage of richer queries
- Role of updates
- Effect of re-encryption
- Viability of active attacks in practice

And challenges:
- Better data sets for simulations
- Query traces
- Countermeasures
Part 2: Machine learning model inversion
Machine learning (ML) systems

(1) Gather some labeled data

(2) Train ML model $f$ from data

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_n) = y$$

(3) Use $f$ in some application or publish it for others to use
Increasing use of ML

Medical applications

Cloud computing

Facial recognition

facebook

Sky Biometry

Cloud-based Face Detection and Recognition API
Privacy concerns in machine learning?

Release of sensitive data?
Even de-identified data dangerous
  [Sweeney ‘00]
  [Naranayan & Shmatikov ‘08] ...

k-anonymity  [Sweeney ‘02]

Differential privacy
  [Dwork, McSherry, Nissim, Smith ‘06]
  ...

Overarching lesson: Don’t release sensitive data sets without due care
Privacy concerns in machine learning?

Release of sensitive data?
Even de-identified data dangerous
  [Sweeney ‘00]
  [Naranayan & Shmatikov ‘08] ...

k-anonymity   [Sweeney ‘02]

Differential privacy
  [Dwork, McSherry, Nissim, Smith ‘06]
  ...

What about risks related to adversarial access to (just) model f?

[Ateniese et al. 2013]: Determine one bit of info about DB given ability to download f
New privacy concerns in ML

Model inversion attacks:

(1) Linear regression for personalized medicine
    \textit{Predict genotypes of patients}

(2) Decision trees trained from lifestyle surveys
    \textit{Predict marital infidelity of training set members}

(3) Neural networks for facial recognition
    \textit{Recover recognizable images of training set members}

Preliminary investigation of countermeasures

\textit{Differential privacy}
\textit{Sensitive-feature-aware CART decision trees}
\textit{Rounded confidence values}

[Fredrikson, Lantz, Lin, Jha, Page, R. – Security `14]
[Fredrikson, Jha, R. – CCS `15]
Privacy in pharmacogenetics

Case study in context of *personalized medicine*

IWPC study:
- Linear regression based classifier
- Trained on demographics, health history, and genetic markers
- Predicts initial dose of warfarin
- [IWPC] researchers showed evidence that this outperformed clinical practice

Data set is publicly available (in de-identified form), but similar data sets must be private
Warfarin model inversion attack

[f(x_1, \ldots, x_n) = y]

Demographic information
Health history
Genotype

Suggested initial dose of warfarin

Info on x_1, \ldots, x_{n-1}
Stable dose y' (y' \neq y)
Model f

Model inversion algorithm
Target person’s genotype

Linear regression model f

Fredrikson, Lantz, Lin, Jha, Page, R. – Security ‘14]
Warfarin model inversion attack

[Fredrikson, Lantz, Lin, Jha, Page, R. – Security `14]

\( x_n \) takes on values in set \{v_1, \ldots, v_s\}

(1) Compute feasible set of input vectors:

\[
\begin{align*}
  z_1 &= (x_1, \ldots, x_{n-1}, v_1) \\
  z_2 &= (x_1, \ldots, x_{n-1}, v_2) \\
  \vdots \\
  z_s &= (x_1, \ldots, x_{n-1}, v_s)
\end{align*}
\]

(2) Compute \( y_j = f(z_j) \) for each \( j \)

(3) Output \( v_j \) that maximizes

\[
\sum_{j=1}^{s} \left( \pi(y, y_j) \cdot \prod_{i=1}^{n} p(z_j[i]) \right)
\]

Weight by error

Independent priors

Realizes MAP estimator (optimal subject to info available)
Model inversion results for IWPC model

Baseline is guessing without access to model (36% accuracy)

Linear regression model directly trained from dataset

Only 5% lower

Everything but genotype

Basic demographics about person

Model aids attacker in prediction almost as much as training directly on data set
New privacy concerns in ML

Model inversion attacks:

1. Linear regression for personalized medicine
   \textit{Predict genotypes of patients}

2. Decision trees trained from lifestyle surveys
   \textit{Predict marital infidelity of training set members}

3. Neural networks for facial recognition
   \textit{Recover recognizable images of training set members}

Preliminary investigation of countermeasures

\textit{Differential privacy}
\textit{Sensitive-feature-aware CART}
\textit{Rounded confidence values}
ML-as-a-service APIs

Free or pay-per-prediction

Black-box (only make predictions) or white-box (download model)
Sensitive decision tree models

538 steak survey
GSS marital happiness study (see paper)

Survey of 332 people to determine if “risky” lifestyle choices correlates with steak preferences

\[ f(x_1, ..., x_n) = y \]

Household income
Whether person gambles
Whether cheated on significant other
...

Prediction of how person likes steak prepared:
- rare
- medium-rare
- medium
- medium-well
- well-done

De-identified training dataset available, we use to simulate attacks
Given:
- $x_1, \ldots, x_{n-1}$
- Actual steak preference $y'$
- Marginal priors, queries to $f$
- Confusion matrix $C$ for $f$

Predict:
- Infidelity status $x_n$

Confusion matrix $C_{y',y} = \#$ training instances w/ steak type $y'$ predicted as $y$

**Simple black-box MAP estimator (like the warfarin one):**

$$\arg\max_{x_n} \frac{C_{y', f(x_1, \ldots, x_n)}}{\sum_{l \in Y} C_{y', l}} \cdot \Pr [ x_n ]$$
Black-box warfarin-like attack for 538 survey

Given:
- $x_1, \ldots, x_{n-1}$
- Actual steak preference $y'$
- Marginal priors, queries to $f$
- Confusion matrix $C$ for $f$

Predict:
- Infidelity status $x_n$

Model inversion algorithm

$C_{y', y} =$ # training instances w/ steak type $y'$ predicted as $y$

Performance:

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline guessing</td>
<td>82.9%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>MI attack</td>
<td>85.8%</td>
<td>85.7%</td>
<td>21.1%</td>
</tr>
</tbody>
</table>
BigML reveals confidence values

For each path:

Confidence = \frac{\# \text{ correct matching}}{\# \text{ total matching}}

# rare instances matching,
# medium-rare matching,
...

5 instances 11.6% of data
New MI attack using granular confidence data

**Given:**

\[ x_1, \ldots, x_{n-1} \]
Actual steak preference \( y' \)
Marginal priors, queries to \( f \)
Confusion matrix \( C \) for \( f \)

**Path counts**

\[ C_{y',y} = \# \text{ training instances w/ steak type } y' \text{ predicted as } y \]

**Predict:**

Infidelity status \( x_n \)

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline guessing</td>
<td>82.9%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>MI attack</td>
<td>85.8%</td>
<td>85.7%</td>
<td>21.1%</td>
</tr>
<tr>
<td>MI attack w/ confidences</td>
<td>86.4%</td>
<td>100%</td>
<td>21.1%</td>
</tr>
</tbody>
</table>
New privacy concerns in ML

Model inversion attacks:

1. Linear regression for personalized medicine
   *Predict genotypes of patients*

2. Decision trees trained from lifestyle surveys
   *Predict marital infidelity of training set members*

3. Neural networks for facial recognition
   *Recover recognizable images of training set members*

Preliminary investigation of countermeasures

* Differential privacy
* Sensitive-feature-aware CART
* Rounded confidence values
Model inversion for facial recognition

\[ x_1, \ldots, x_n \rightarrow y \]

\[ \text{DB of data} \]

\[ \text{Training} \]

\[ \text{ML model } f \]
Model inversion for facial recognition

Softmax
Multi-layer perceptron (MLP)
Stacked de-noising auto-encoder (DAE)

Can attacker use f to recover images of training member’s faces?

Pixel data
Prediction

ML model f

x₁, ..., xₙ
y

DB of data
Training
Taking advantage of confidence values

\[ f(x_1, \ldots, x_n) = [y_{Bob}, \ldots, y_{Jake}] \]

Unknown pixel data

Vector of class confidences each in [0,1]
Output label of highest confidence class

AT&T faces dataset:
\[ n = 92 \times 112 = 10,304 \]
\[ |x_i| = 8 \text{ bits (grayscale intensity value)} \]

\[ 8^{10,304} \text{ possible images} \]

Naïve brute-force search won’t work
Taking advantage of confidence values

\[ f(x_1, \ldots, x_n) = [y_{Bob}, \ldots, y_{Jake}] \]

Unknown pixel data

Vector of class confidences each in \([0,1]\)
Output label of highest confidence class

**Insight:**
Confidences allows efficient gradient descent-based search

Find \(x_1, \ldots, x_n\) with highest confidence for ‘Bob’

**Gradient descent:**
- White-box we calculate symbolically
- Black-box need to do numerical estimation

<table>
<thead>
<tr>
<th>Model (trained on AT&amp;T faces)</th>
<th>Local white-box time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax</td>
<td>1</td>
</tr>
<tr>
<td>Multi-layer perceptron</td>
<td>1,298</td>
</tr>
<tr>
<td>Denoising autoencoder</td>
<td>692</td>
</tr>
</tbody>
</table>
Example outputs of MI attack for different models

Target  Softmax  MLP  DAE

Inversion for three neural-network classifiers:
Softmax, Multi-layer perceptron, De-noising auto-encoder
Trained on AT&T faces dataset (40 individuals, 400 images)
The image on the left is a face that was altered by computer processing. It may or may not correspond to one of the faces displayed to the right of it.

If you believe that it does correspond to one of the other faces, please select the corresponding image. If you do not believe that it corresponds to one of the other faces, select “Not Present”.

---

**Recognizability?**

Amazon Mechanical Turk to evaluate image reconstruction recognizability

---

Re-identification accuracy up to 95% for skilled workers
New privacy concerns in ML

Model inversion attacks:

1. Linear regression for personalized medicine
   *Predict genotypes of patients*

2. Decision trees trained from lifestyle surveys
   *Predict marital infidelity of training set members*

3. Neural networks for facial recognition
   *Recover recognizable images of training set members*

Preliminary investigation of countermeasures

*Differential privacy*

*Sensitive-feature-aware CART*

*Rounded confidence values*
Differential privacy [Dwork, McSherry, Nissim, Smith ‘06]

Given model $f$ adversary can’t learn whether any single individual contributed to training data set

Inversion success: Can’t vary by $> e^\varepsilon$ for dataset with or w/o individual

Guarantees nothing about absolute success
End-to-end analysis of DP in warfarin case

Differentially private version of model hides whether individual contributed to training data set with efficacy a function of privacy budget $\epsilon$ [Zhang et al.] functional mechanism for private linear regression

We performed end-to-end case study:

- Evaluate model inversion disclosure risk for DP models
- Use simulated clinical trials to evaluate utility of DP models
Other simple countermeasures?

Attacks that rely on confidence data: degrade it

Our MI attack against softmax with rounded confidences:

![Image of face reconstructions with different rounding levels]

<table>
<thead>
<tr>
<th>Rounding Level</th>
<th>Image Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>no rounding</td>
<td>Original image</td>
</tr>
<tr>
<td>$r = 0.001$</td>
<td>Image with slight distortion</td>
</tr>
<tr>
<td>$r = 0.005$</td>
<td>Image with moderate distortion</td>
</tr>
<tr>
<td>$r = 0.01$</td>
<td>Image with significant distortion</td>
</tr>
<tr>
<td>$r = 0.05$</td>
<td>Image with extreme distortion</td>
</tr>
</tbody>
</table>

Rounding confidence values to nearest $r$:

Sensitive-feature-aware CART decision tree training
(see paper)
Model inversion and ML privacy

Adversarial access to models has subtle implications

Open questions: better attacks, handling more sophisticated ML models, principled countermeasures
Exploiting Leakage in Searchable Encryption and Machine Learning

Tom Ristenpart

Covering joint work with:
David Cash, Paul Grubbs, Jason Perry (Searchable encryption)
Matthew Fredrikson, Eric Lantz, Simon Lin, David Page, Somesh Jha (ML)