Property-revealing encryption

Tom Ristenpart
CS 6431
Client wants to store data up on Dropbox

- High availability, synch across devices
- Server includes much value-add functionality
  - Keyword search (find all files with “Tom” in text)
  - Deduplication (if two files same, store only one copy)
  - Thumbnail generation for images
Outsourced storage settings

Salesforce stores customer records for companies

- Name: Clarisse
- Age: 22
- Salary: 100,000
- Comments: Works in NYC office
- Gender: Female
- SSN: 555-31-4325

Much value-add server-side functionality
- Keyword search (find all records with name “Clarisse”)
- Range queries (all records with 20 <= Age <= 30)
- Sorted lists (return records ordered by salary)

What security threats would one worry about?
Symmetric encryption

key generation

Kg

R_k

K

Enc

C

M

C

Dec

M or error

C is a ciphertext

Correctness: \( \text{Dec}(K, \text{Enc}(K,M,R)) = M \) with probability 1 over randomness used

R signifies fresh per-message random bits.

Optional

R

M
IND-CPA security

Adversary gets to submit messages to oracle

\[ \text{Encrypt}(M_0, M_1) \]

\[ C \]

\[ < \]

\[ -\text{Enc} \]

\[ K, M_b \]

\[ \text{Ret} \ C \]

IND-CPA challenge game

Adversary outputs guess \( b' \) of \( b \)

Wins if \( b' = b \)

Adversary must query with \( |M_0| = |M_1| \)

Security goal: \( \text{Enc}(K, M) \) doesn’t even leak single bit about \( M \)

[Bellare, Desai, Jokipii, Rogaway 97]
Outsourced storage settings

Salesforce stores customer records for companies

What storage service provider functionalities broken?

- Field search  (Find all records with Name = Alice)
- Keyword search
- Range queries  (Find all people who make between 90k and 120k)
- Format problems  (Age must be integer between 0 and 130)
- ...
Instead: Property-Revealing Encryption (PRE)

Salesforce stores customer records for companies

Name: Alice
Age: 22
Salary: $100,000
Comments: Works in NYC office
Gender: Female
SSN: 555-31-4325

Encrypt data with special PRE schemes that leak just enough about plaintexts to perform some operations

Sometimes advertised as “computing over encrypted data” – metaphor is flawed
## Property-revealing encryption

<table>
<thead>
<tr>
<th>Problem</th>
<th>Crypto primitive</th>
<th>Description</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword search</td>
<td>Searchable symmetric encryption</td>
<td>Perform search over ciphertexts given encrypted search token</td>
<td>[Dawn, Song, Wagner 2000] [Curtmola et al. 2006] ...</td>
</tr>
<tr>
<td>Equality search</td>
<td>Deterministic encryption</td>
<td>(X = Y) implies (DET(X) = DET(Y))</td>
<td>[Rogaway Shrimpton 06]</td>
</tr>
<tr>
<td>Range queries</td>
<td>Order-preserving encryption</td>
<td>(X &gt; Y) implies (OPE(X) &gt; OPE(Y))</td>
<td>[Boldyreva et al. 2009]</td>
</tr>
<tr>
<td>Range queries</td>
<td>Order-revealing encryption</td>
<td>(X &gt; Y) implies (\text{Cmp}(\text{ORE}(X),\text{ORE}(Y)) = 1)</td>
<td>[Boldyreva et al. 11], [Boneh et al. 15]</td>
</tr>
<tr>
<td>Deduplication</td>
<td>Message-locked encryption</td>
<td>Different user’s encryptions of same plaintext give same ciphertext</td>
<td>[Douceur et al. 2002] [Bellare et al. 2013]</td>
</tr>
<tr>
<td>Format restrictions</td>
<td>Format-preserving encryption</td>
<td>Ciphertext has same format as plaintext</td>
<td>[Bellare et al. 2009]</td>
</tr>
</tbody>
</table>
Lots of industry activity

Cloud access security brokers (CASB)
PREs in databases

Encrypt by column

Name: C1 = DET(K1, “Alice” )
Age: C2 = OPE(K2, “22” )
Salary: C3 = OPE(K3, “100,000” )

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1’</td>
<td>C2’</td>
<td>C3’</td>
</tr>
<tr>
<td>C1’’</td>
<td>C2’’</td>
<td>C3’’</td>
</tr>
</tbody>
</table>
PREs in databases

Perform *equality search* by querying DET encryption of keyword

Property revelation: \( \text{DET}(K1,X) = \text{DET}(K1,Y) \iff X = Y \)

<table>
<thead>
<tr>
<th>DET(K1, “Alice”)</th>
<th>OPE(K2, “22”)</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DET(K1, “Bob”)</td>
<td>OPE(K2, “29”)</td>
<td>C3’</td>
</tr>
<tr>
<td>DET(K1, “Alice”)</td>
<td>OPE(K2, “31”)</td>
<td>C3’’</td>
</tr>
</tbody>
</table>
PREs in databases

Perform *equality search* by querying DET encryption of keyword

Property revelation: $\text{DET}(K1,X) = \text{DET}(K1,Y) \iff X = Y$

Perform *range search* by querying OPE encryption of end points

Property revelation: $\text{OPE}(K2,X) < \text{OPE}(K2,Y) \iff X < Y$

What is revealed to adversarial server?
CryptDB

At this point nothing is leaked to Database:

- Full symmetric encryption security given by Enc
- Repeats not visible, ordering not visible
CryptDB

[Ada Popa et al. 2011]

SQL Database

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>C2</td>
<td>C3</td>
</tr>
<tr>
<td>C1'</td>
<td>C2'</td>
<td>C3'</td>
</tr>
<tr>
<td>C1''</td>
<td>C2''</td>
<td>C3''</td>
</tr>
</tbody>
</table>

Perform equality search by:
- Send outer encryption key for searched-on-column
- Server decrypts that column
- Use DET to find rows corresponding to Alice as before

At this point equality patterns of first column leaked
Perform range search by:

- Send outer encryption key for searched-on-column
- Server decrypts that column
- Use OPE range query mechanism from before
Perform range search by:
- Send outer encryption key for searched-on-column
- Server decrypts that column
- Use order preservation of OPEnc to find rows in range

At this point equality patterns of first and second column leaked
Ordering information on second column leaked
Standard PRE use and CryptDB

- CryptDB: lots more engineering details
  - Other types of queries, other kinds of onions
  - How to decompose SQL queries into appropriate levels of encrypted data
  - Showed that some columns need not be decrypted in some workloads
  - Does not work with legacy storage systems (e.g., SalesForce)
- “Standard” PRE use (no onions) easier to deploy. This is what is used in practice currently

How damaging is the leakage?
What is known about security?

Cryptographic theory offers proofs of security for OPE, DET. Preservation means can’t achieve IND-CPA definitions

**Solution:**
Weaken security notions until one can show achievability
IND-CPA security for DET

Adversary gets to submit messages to oracle

Adversary outputs guess b’ of b
Wins if b’ = b

Security goal: $\text{Enc}(K,M)$ doesn’t leak anything about plaintexts
IND-CPA security for DET

Challenge game

Encrypt(M0,M1)  
C <- DET(K,Mb) 
Ret C

Adversary gets to submit messages to oracle

M0,M1  
C'  
M0',M1'  
C'  
...  

b'  

Adversary outputs guess b' of b
Wins if b' = b

Adversary must query with |M0| = |M1|
and Mb,Mb', ... all distinct for b = 0,1

Security goal: Enc(K,M) doesn’t leak anything but plaintext equality
IND-OCPA security for OPE

[Boldyreva, Chenette, Lewi, O’Neill 09]

IND-OCPA challenge game

Adversary gets to submit messages to oracle

Adversary outputs guess b’ of b
Wins if b’ = b

b is a uniformly sampled bit and K is uniformly sampled key
Both hidden from adversary

Adversary must query with |M0| = |M1|
and M₀, M₀’, ... all distinct for b = 0, 1
and M₀ < M₀’ iff M₁ < M₁’ for all pairs of queries

Security goal: Enc(K,M) doesn’t leak anything but plaintext equality and order relationships
Achieving IND-OCPA hard

Scheme must have mutable ciphertexts (even w/ stateful clients & interactivity), otherwise require exponentially large ciphertexts

[Popa et al. 2011]

This degrades the deployability of IND-OCPA schemes. They aren’t currently used

**Instead**: weaker security goals such as indistinguishability from a random order-preserving function

[BCLO 09] give clever scheme provably secure under this definition

[BCO 11] show that ciphertexts leak at least *approximately half the plaintext*
Leakage-abuse attacks (LAAs)

Question: what can attackers learn from leakage of PRE schemes?

Searchable encryption
- [Islam, Kuzu, and Kantarcioglu 2012]
- [Cash, Grubbs, Perry, Ristenpart 2015]
- [Zhang, Katz, Papamanthou 2016]
- [Wright, Kamara 2016]

DET, OPE, ORE
- [Naveed, Kamara, Wright 2015]
- [Durak, DuBuisson, Cash 2016]
- [Grubbs et al. 2016]
Ciphertext-only inference attacks:
• get access to database
• Auxiliary information about plaintext distribution

Sorting attack against “dense” OPE columns

Already reveals everything for some column types (e.g., binary attributes)
# Frequency analysis

<table>
<thead>
<tr>
<th>DET(K1, “Alice” )</th>
<th>DET(K1, “Bob” )</th>
</tr>
</thead>
<tbody>
<tr>
<td>DET(K1, “Alice” )</td>
<td>DET(K1, “Alice” )</td>
</tr>
<tr>
<td>DET(K1, “Alice” )</td>
<td>DET(K1, “Bob” )</td>
</tr>
</tbody>
</table>

Auxiliary data is plaintext dataset distributed similarly to target. Histogram:

- Alice 15
- Bob 3

Match most frequent ciphertext to most frequent auxiliary value
Match next most frequent to next most frequent, etc.

Technique is thousands of years old; applied most often to substitution ciphers
But: DET is just a substitution cipher over large alphabet
Graph viewpoint: DET

Bipartate graph problem: find matching of ciphertexts, names that minimizes total weight

\[ \text{Weight}(C, \text{Name}) = |\text{Freq}(C) - \text{Freq}(\text{Name})| \]

Equivalent to frequency analysis
Bipartate graph problem: find non-crossing matching of ciphertexts, names that minimizes total weight

$$\text{Weight}(C, \text{Name}) = |\text{Freq}(C) - \text{Freq}(\text{Name})|$$

This takes into account ordering constraints

[Grubbs et al. 2016]
Graph viewpoint: OPE

**Bipartate graph problem**: find *non-crossing* matching of ciphertexts, names that minimizes total weight

$$\text{Weight}(C, \text{Name}) = |\text{Freq}(C) - \text{Freq}(\text{Name})|$$

This takes into account ordering constraints

[Grubbs et al. 2016]
Non-crossing vs. NKW attack on first and last names

First names: red  Last names: blue  [Grubbs et al. 2016]
Taking account of additional leakage

Recall that most OPE schemes reveal more information than just order, frequency
BCLO scheme (the one used most widely in practice) leaks $\sim$half of plaintext bits

BCO ’11 gave way to compute value $m_c$ which is midpoint of range of size $\sqrt{N}$
that has 99% chance of including plaintext

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Ciphertext</th>
<th>$m_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>michael</td>
<td>cyrzjipnouushzh</td>
<td>michaelkypfbkfr</td>
</tr>
<tr>
<td>david</td>
<td>aenpse cevvpkmr</td>
<td>david jwbvhec</td>
</tr>
<tr>
<td>robert</td>
<td>emlqrnycvblqquad</td>
<td>robert lwyecorr</td>
</tr>
<tr>
<td>john</td>
<td>cccnczzzzpruvjhd</td>
<td>johmzzzsfbunn</td>
</tr>
<tr>
<td>james</td>
<td>bzkxrq gzortby</td>
<td>james zyovtq</td>
</tr>
<tr>
<td>daniel</td>
<td>aelsfopcabjdvjc</td>
<td>daniel jgaginu</td>
</tr>
<tr>
<td>richard</td>
<td>ekrzjmjhjxykbba</td>
<td>richardkmfnwwx</td>
</tr>
<tr>
<td>jose</td>
<td>ccqrlzzuzzozokin</td>
<td>josdzzzzvfruqq</td>
</tr>
<tr>
<td>mark</td>
<td>cwmfzzzzjxhklh</td>
<td>marjzzzzzqyduv</td>
</tr>
<tr>
<td>christopher</td>
<td>zokwwbrbibyouou</td>
<td>christotnqfolw</td>
</tr>
</tbody>
</table>

[Grubbs et al. 2016]
Can prune graph problem using this additional leakage

Bipartate graph problem: find non-crossing matching of ciphertexts, names that minimizes total weight

\[
\text{Weight}(\text{C, Name}) = |\text{Freq}(C) - \text{Freq}(\text{Name})|
\]

This takes into account ordering constraints

[Grubbs et al. 2016]
Can prune graph problem using this additional leakage

**Bipartate graph problem**: find *non-crossing* matching of ciphertexts, names that minimizes total weight

\[
\text{Weight}(C, \text{Name}) = |\text{Freq}(C) - \text{Freq}(\text{Name})|
\]

This takes into account ordering constraints

[Grubbs et al. 2016]
LAAs Fallout

Lots of open questions on how to improve already damaging attacks

Unclear if, and if so where, PREs can still be used securely

In practice PREs are still the only option in many important contexts otherwise: must download to process data

<table>
<thead>
<tr>
<th>Scheme(s)</th>
<th>First names</th>
<th>Last names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kerschbaum [27]</td>
<td>26%</td>
<td>6%</td>
</tr>
<tr>
<td>Popa et al. [36], Kerschbaum [28]</td>
<td>84%</td>
<td>38%</td>
</tr>
<tr>
<td>BCLO [12, 13]</td>
<td>99%</td>
<td>97%</td>
</tr>
<tr>
<td>CLWW [18]</td>
<td>98%</td>
<td>75%</td>
</tr>
<tr>
<td>BCLO + CLWW [18]</td>
<td>85%</td>
<td>44%</td>
</tr>
<tr>
<td>Baseline Guessing</td>
<td>4%</td>
<td>1%</td>
</tr>
</tbody>
</table>