# Inference Attacks on Property-Preserving Encrypted Databases

(Naveed, Kamara, and Wright 2015)

---

Andreas Pfefferle

21 October 2019

Current Topics in Information Security

# Question

Who of you already knows the "Elektronisches Patientendossier"?

# Elektronisches Patientendossier

- allows the exchange of information between hospitals, doctors and patients
- is intended to improve the quality of medical treatment
- **BUT:** highly sensitive data is stored on a (potentially) untrustworthy server

# Elektronisches Patientendossier

- allows the exchange of information between hospitals, doctors and patients
- is intended to improve the quality of medical treatment
- **BUT:** highly sensitive data is stored on a (potentially) untrustworthy server

# Elektronisches Patientendossier

Quick workaround: **End-to-End Encryption**

What if a hospital wants to find out how many male patients between 50 and 59 were hospitalized because of lung cancer since the beginning of this year?

# Elektronisches Patientendossier

Quick workaround: **End-to-End Encryption**

What if a hospital wants to find out how many male patients between 50 and 59 were hospitalized because of lung cancer since the beginning of this year?

# Elektronisches Patientendossier

Quick workaround: **End-to-End Encryption**

What if a hospital wants to find out how many male patients between 50 and 59 were hospitalized because of lung cancer since the beginning of this year?

```
SELECT COUNT(Patient.Id)
  FROM Patient
  INNER JOIN Hospitalization ON Patient.Id = Hospitalization.PatientId
  WHERE Hospitalization.Diagnosis = 'lung cancer'
  AND Hospitalization.Admission >= '2019-01-01'
  AND Patient.Sex = 'male'
  AND Patient.Age BETWEEN 50 AND 59;
```

# Potential Problems of this quick Workaround

- no search on (ordinarily) encrypted data
- downloading the whole encrypted database $\rightarrow$ overhead
- clients may not be powerful enough to handle large amounts of data
- clients must be trusted (not all clients should be authorized to see information about every patient)

# Potential Problems of this quick Workaround

- no search on (ordinarily) encrypted data
- downloading the whole encrypted database $\rightarrow$ overhead
- clients may not be powerful enough to handle large amounts of data
- clients must be trusted (not all clients should be authorized to see information about every patient)

# Potential Problems of this quick Workaround

- no search on (ordinarily) encrypted data
- downloading the whole encrypted database $\rightarrow$ overhead
- clients may not be powerful enough to handle large amounts of data
- clients must be trusted (not all clients should be authorized to see information about every patient)

# Potential Problems of this quick Workaround

- no search on (ordinarily) encrypted data
- downloading the whole encrypted database $\rightarrow$ overhead
- clients may not be powerful enough to handle large amounts of data
- clients must be trusted (not all clients should be authorized to see information about every patient)

# Potential Problems of this quick Workaround

- no search on (ordinarily) encrypted data
- downloading the whole encrypted database $\rightarrow$ overhead
- clients may not be powerful enough to handle large amounts of data
- clients must be trusted (not all clients should be authorized to see information about every patient)

**We need a fast and efficient search method on encrypted data that runs on a powerful yet untrustworthy server.**

# Property-Preserving Encryption: Deterministic Encryption

- DTE $= (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$: symmetric encryption scheme
- Enc not randomized $\rightarrow$ no IND-CPA security
- if $m_1 = m_2$ then $\mathsf{Enc}_K(m_1) = \mathsf{Enc}_K(m_2)$

# Property-Preserving Encryption: Order-Preserving Encryption

- $\mathsf{OPE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$: symmetric encryption scheme
- if $m_1 < m_2$ then $\mathsf{Enc}_K(m_1) < \mathsf{Enc}_K(m_2)$
- if $m_1 > m_2$ then $\mathsf{Enc}_K(m_1) > \mathsf{Enc}_K(m_2)$
- if $m_1 = m_2$ then $\mathsf{Enc}_K(m_1) = \mathsf{Enc}_K(m_2)$

# Inference Attacks on Encrypted Data

- all known practical encrypted search solutions inevitably leak information about the plaintext
- **Attack idea: combine leakage with publicly-available information and try to recover the encrypted data**

# Inference Attack: Simple Example

| PatientId | Mortality Risk |
|---|---|
| 210938 | major |
| 138103 | minor |
| 123084 | minor |
| 409283 | extreme |
| 139879 | moderate |
| 131933 | minor |
| ... | ... |

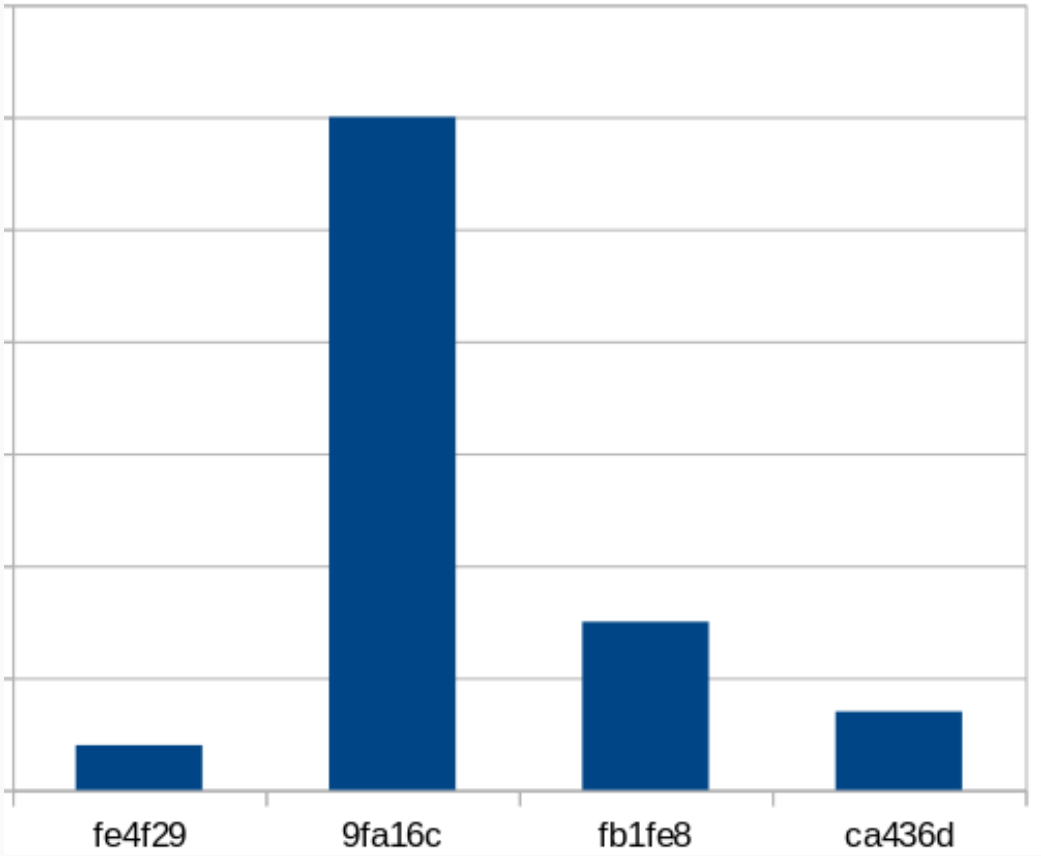
Plaintext Database

# Inference Attack: Simple Example

| Mortality Risk |
| --- |
| fe4f29 |
| 9fa16c |
| 9fa16c |
| ca436d |
| fb1fe8 |
| 9fa16c |
| ... |

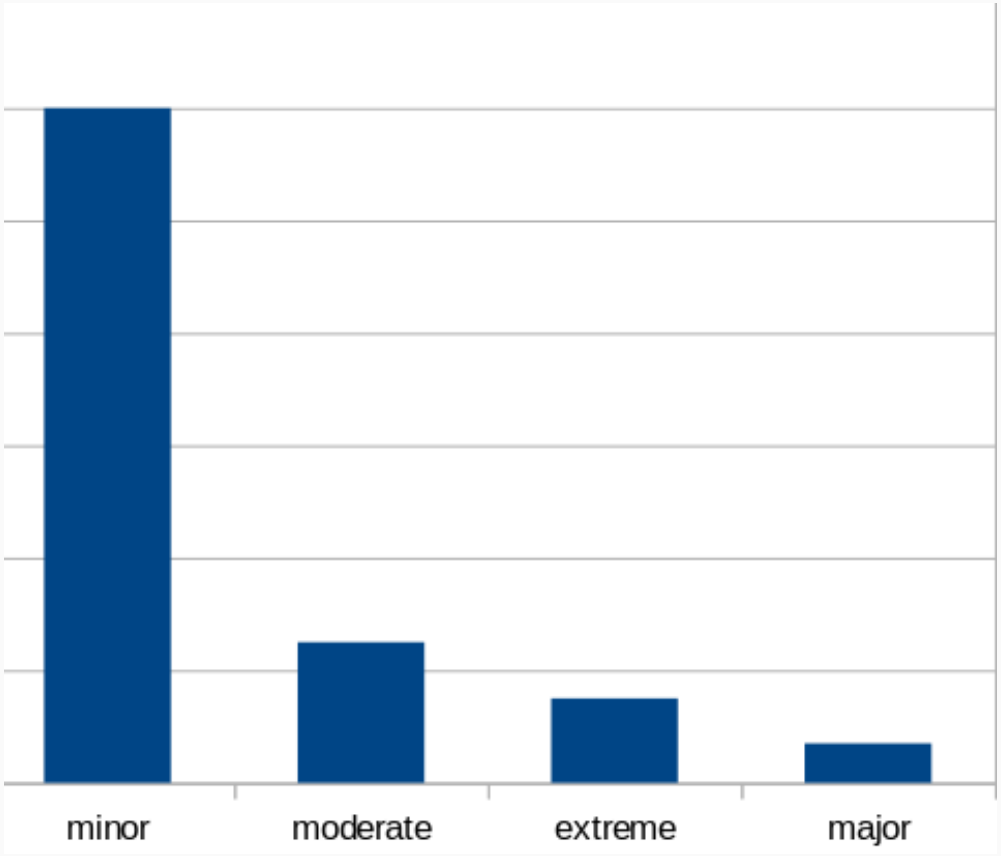
DTE Database

# Inference Attack: Simple Example



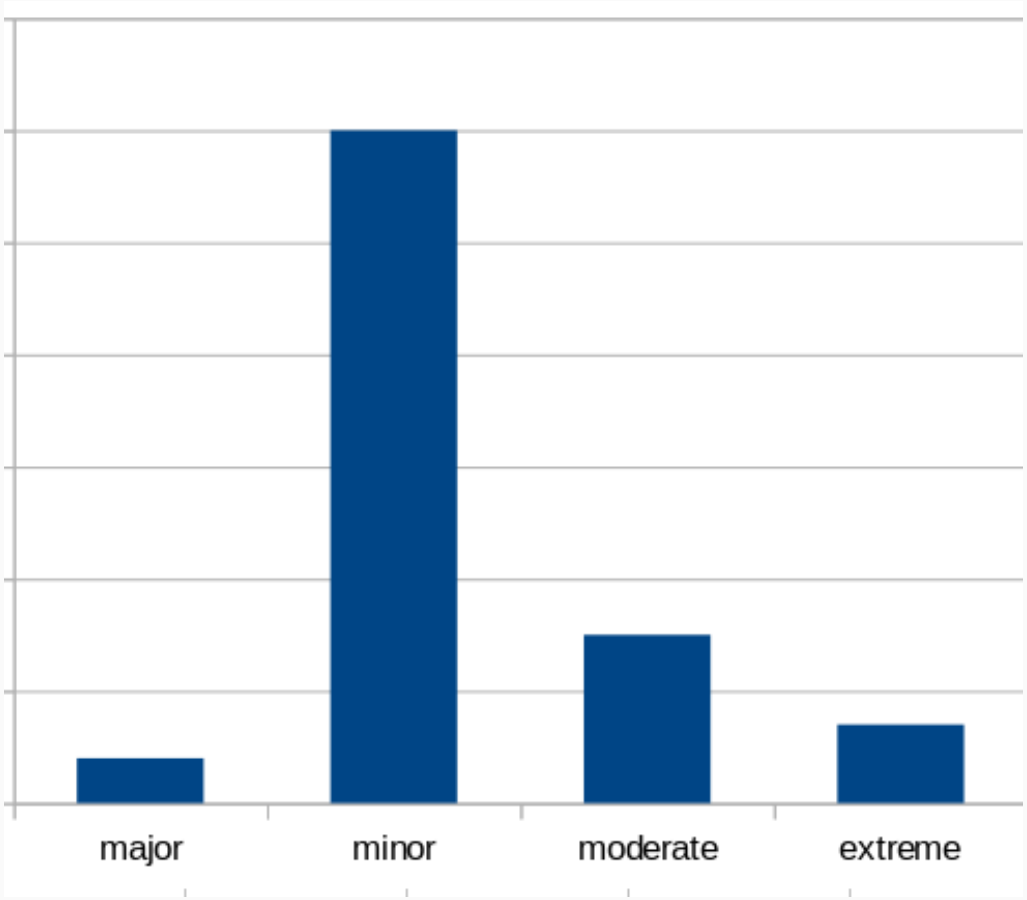

First step: histogram of ciphertexts

# Inference Attack: Simple Example




Second step: histogram of publicly-available dataset

# Inference Attack: Simple Example



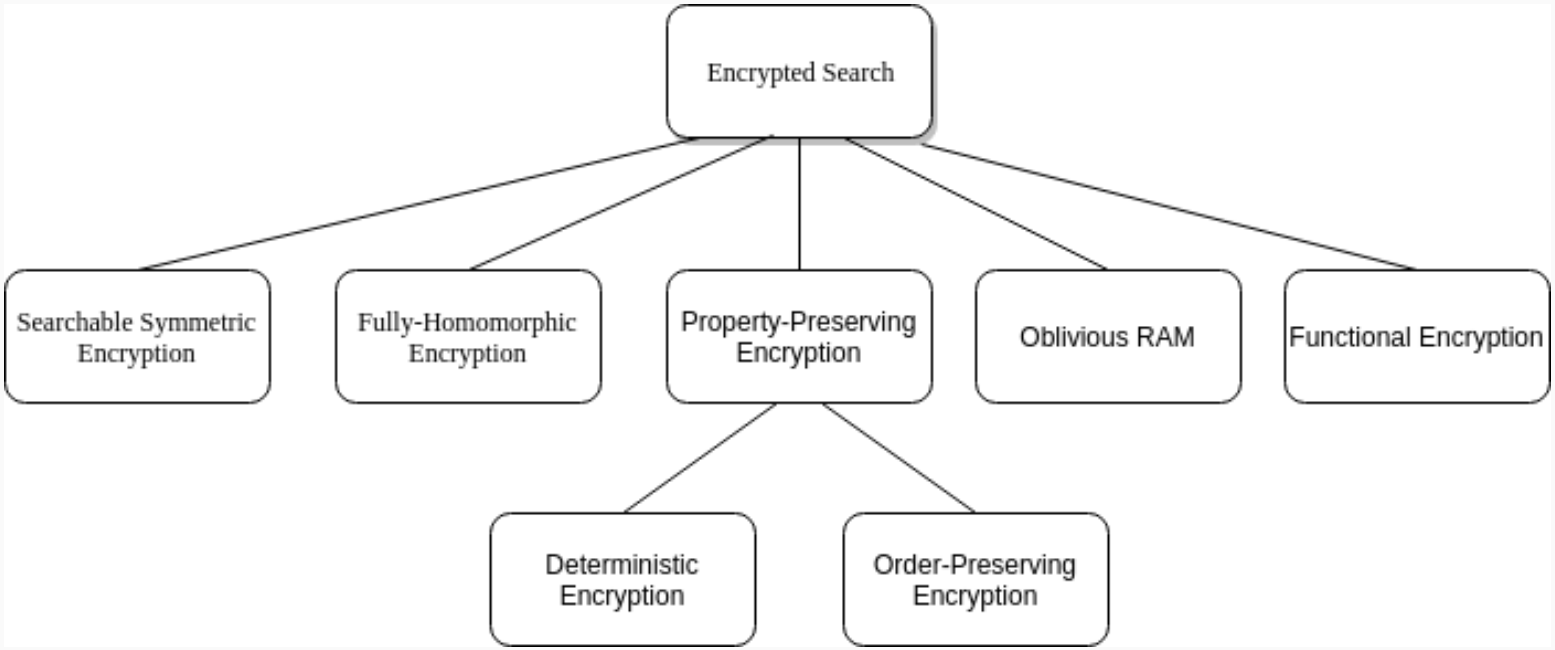

Third step: Assign (potentially) correct labels

# Overview

1. Encrypted Search Background
2. CryptDB: SQL on Encrypted Data
3. Inference Attacks on PPE Databases
4. Opinion and Thoughts

# Encrypted Search Background

# Encrypted Search




Overview of Encrypted Search Approaches

# Encrypted Search

- different trade-offs between security, query expressiveness, and efficiency
- the most secure approaches are currently too inefficient
- PPE (e.g., DTE and OPE) are state-of-the-art solutions

# Encrypted Search

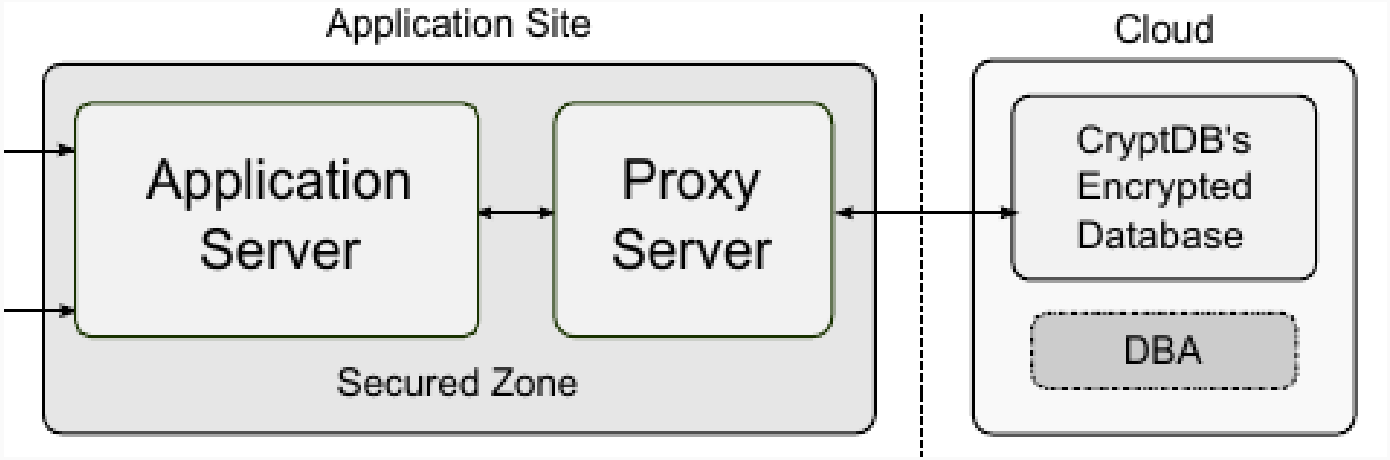
- different trade-offs between security, query expressiveness, and efficiency
- the most secure approaches are currently too inefficient
- PPE (e.g., DTE and OPE) are state-of-the-art solutions

# Encrypted Search

- different trade-offs between security, query expressiveness, and efficiency
- the most secure approaches are currently too inefficient
- PPE (e.g., DTE and OPE) are state-of-the-art solutions

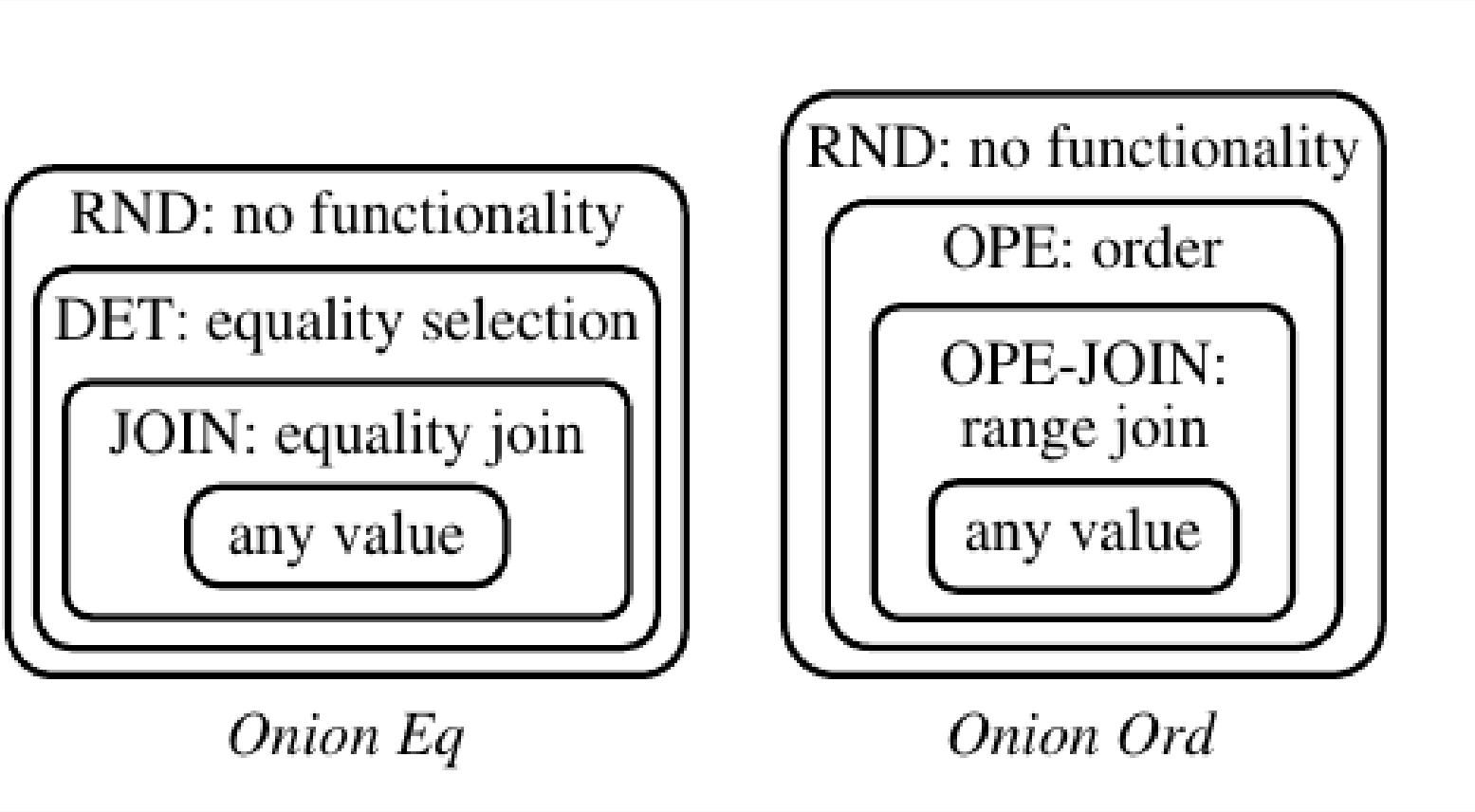
# CryptDB: SQL on Encrypted Data

# CryptDB: Overview



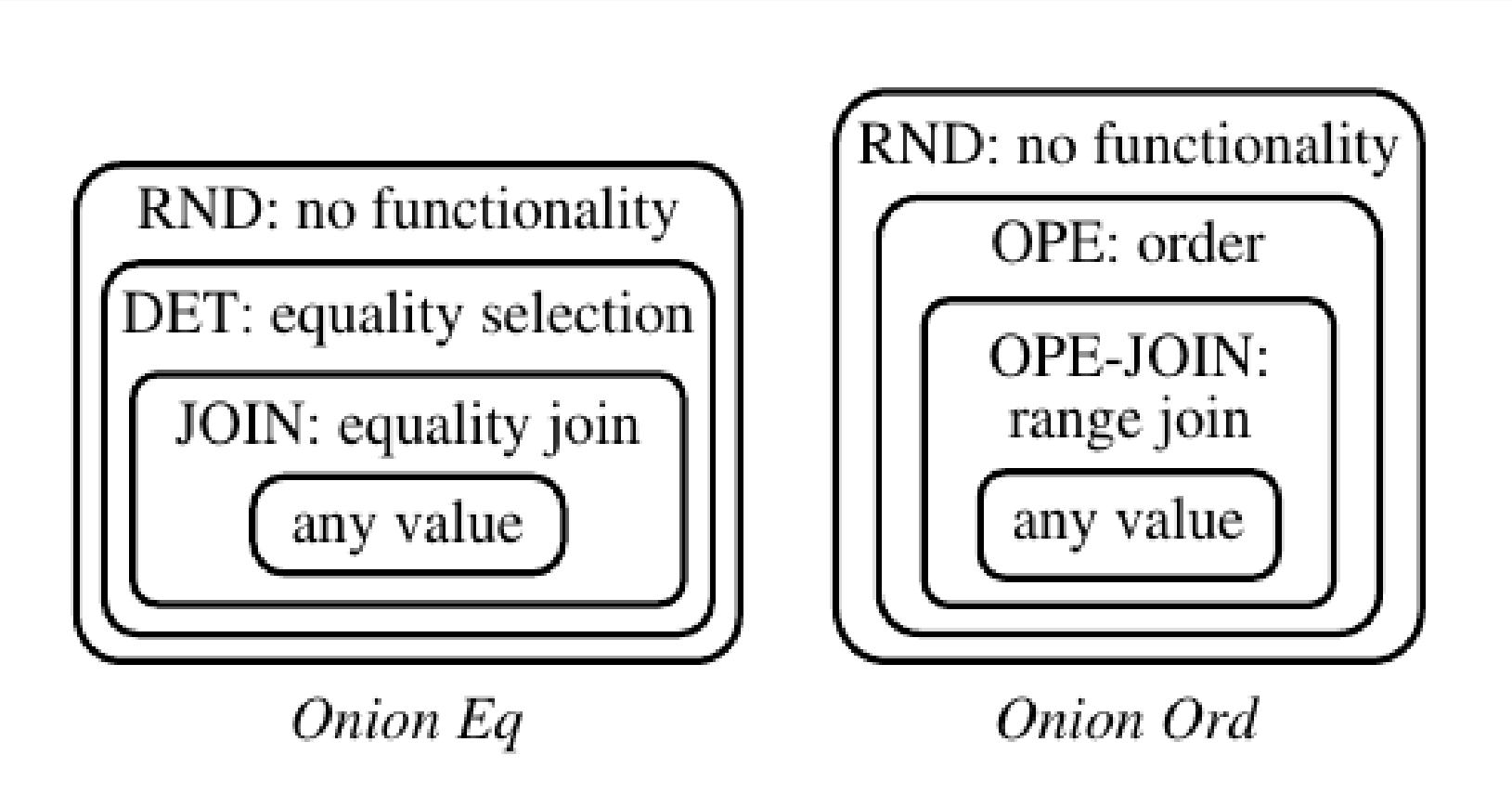

CryptDB Architecture

# CryptDB: Onions



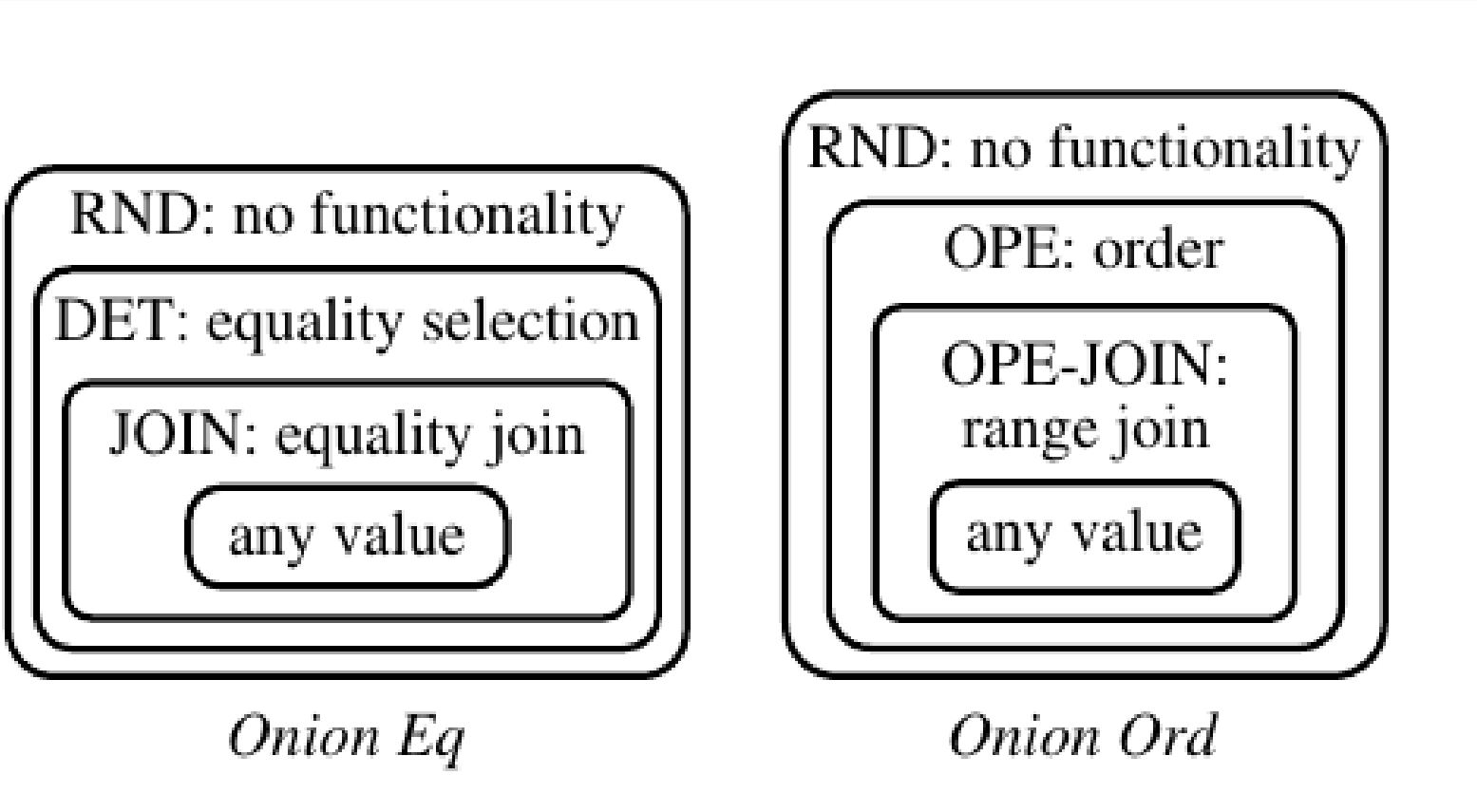

- Equality onion: $\mathrm{ct} = \mathrm{Enc}^{\mathrm{RND}}_{K_S}\left(\mathrm{Enc}^{\mathrm{DTE}}_{K_D}\left(\mathrm{Enc}^{\mathrm{EJOIN}}_{K_J}(s)\right)\right)$
- Order onion: $\mathrm{ct} = \mathrm{Enc}^{\mathrm{RND}}_{K_S}\left(\mathrm{Enc}^{\mathrm{OPE}}_{K_O}\left(\mathrm{Enc}^{\mathrm{RJOIN}}_{K_{OJ}}(s)\right)\right)$

# CryptDB: Onions




- Equality onion: $\mathrm{ct} = \mathsf{Enc}^{\mathsf{RND}}_{K_S}\left(\mathsf{Enc}^{\mathsf{DTE}}_{K_D}\left(\mathsf{Enc}^{\mathsf{EJOIN}}_{K_J}(s)\right)\right)$
- Order onion: $\mathrm{ct} = \mathsf{Enc}^{\mathsf{RND}}_{K_s}\left(\mathsf{Enc}^{\mathsf{OPE}}_{K_o}\left(\mathsf{Enc}^{\mathsf{RJOIN}}_{K_{OJ}}(s)\right)\right)$

# CryptDB: Onions




- Equality onion: $\mathrm{ct} = \mathsf{Enc}^{\mathsf{RND}}_{K_S}\left(\mathsf{Enc}^{\mathsf{DTE}}_{K_D}\left(\mathsf{Enc}^{\mathsf{EJOIN}}_{K_J}(s)\right)\right)$
- Order onion: $\mathrm{ct} = \mathsf{Enc}^{\mathsf{RND}}_{K_S}\left(\mathsf{Enc}^{\mathsf{OPE}}_{K_O}\left(\mathsf{Enc}^{\mathsf{RJOIN}}_{K_{OJ}}(s)\right)\right)$

# CryptDB: Queries

- encrypted cells are decrypted down a certain layer (*peeling*)
- proxy keeps track of peeling and rewrites queries
- equality: query $v$ is replaced with $\mathrm{ct} = \mathsf{DTE}.\mathsf{Enc}_K(v)$
- range/order: query $v$ is replaced with $\mathrm{ct} = \mathsf{OPE}.\mathsf{Enc}_K(v)$

# CryptDB: Queries

- encrypted cells are decrypted down a certain layer (*peeling*)
- proxy keeps track of peeling and rewrites queries
- equality: query $v$ is replaced with $\mathrm{ct} = \mathsf{DTE}.\mathsf{Enc}_K(v)$
- range/order: query $v$ is replaced with $\mathrm{ct} = \mathsf{OPE}.\mathsf{Enc}_K(v)$

# CryptDB: Queries

- encrypted cells are decrypted down a certain layer (*peeling*)
- proxy keeps track of peeling and rewrites queries
- equality: query $v$ is replaced with $\mathrm{ct} = \mathsf{DTE}.\mathsf{Enc}_K(v)$
- range/order: query $v$ is replaced with $\mathrm{ct} = \mathsf{OPE}.\mathsf{Enc}_K(v)$

# Inference Attacks on PPE Databases

# Attacking DTE Columns: Frequency Analysis

1. compute $\psi \leftarrow$ vSort ( Hist ($\mathbf{c}$))
2. compute $\boldsymbol{\pi} \leftarrow$ vSort ( Hist ($\mathbf{z}$))
3. output: $\alpha : \mathbb{C}_k \rightarrow \mathbb{M}_k$ such that

$$\alpha(c) = \begin{cases} \pi\left[\mathsf{Rank}_{\psi}(c)\right] & \text{if } c \in \mathbf{c} \\ \perp & \text{if } c \notin \mathbf{c} \end{cases}$$

# Attacking DTE Columns: Frequency Analysis

1. compute $\psi \leftarrow$ vSort ( Hist ($\mathbf{c}$))
2. compute $\boldsymbol{\pi} \leftarrow$ vSort ( Hist ($\mathbf{z}$))
3. output: $\alpha : \mathbb{C}_k \rightarrow \mathbb{M}_k$ such that

$$\alpha(c) = \begin{cases} \pi\left[\mathsf{Rank}_{\psi}(c)\right] & \text{if } c \in \mathbf{c} \\ \perp & \text{if } c \notin \mathbf{c} \end{cases}$$

# Attacking DTE Columns: Frequency Analysis

1. compute $\psi \leftarrow$ vSort ( Hist ($\mathbf{c}$))
2. compute $\boldsymbol{\pi} \leftarrow$ vSort ( Hist ($\mathbf{z}$))
3. output: $\alpha : \mathbb{C}_k \rightarrow \mathbb{M}_k$ such that

$$\alpha(c) = \begin{cases} \pi\left[\mathsf{Rank}_\psi(c)\right] & \text{if } c \in \mathbf{c} \\ \perp & \text{if } c \notin \mathbf{c} \end{cases}$$

**This what we have done earlier!**

# Attacking DTE Columns: $l_p$-Optimization

1. compute $\psi \leftarrow \mathsf{vSort}\ (\ \mathsf{Hist}\ (\mathbf{c}))$
2. compute $\boldsymbol{\pi} \leftarrow \mathsf{vSort}\ (\ \mathsf{Hist}\ (\mathbf{z}))$
3. output: $\arg\min_{X \in \mathbb{P}_n} \|\psi - X \cdot \boldsymbol{\pi}\|_p$

# Attacking DTE Columns: $l_p$-Optimization

1. compute $\psi \leftarrow$ vSort ( Hist ($\mathbf{c}$))
2. compute $\boldsymbol{\pi} \leftarrow$ vSort ( Hist ($\mathbf{z}$))
3. output: $\arg\min_{X \in \mathbb{P}_n} \|\psi - X \cdot \boldsymbol{\pi}\|_p$

The frequency attack ignores the amplitude of the frequencies and only takes into account their rank

# Attacking DTE Columns: Summary

- both attack variants are able to recover a significant amount of cells in a real-world attack scenaria, e.g. 100% of mortality risk
- even significant fractions of larger message spaces such as age or length of stay could be recovered
- frequency analysis and $l_2$- and $l_3$-optimization performed equally well in experiments, $l_1$-optimization performed worse
- advantage of $l_p$-optimization: also produces cost information

# Attacking DTE Columns: Summary

- both attack variants are able to recover a significant amount of cells in a real-world attack scenaria, e.g. 100% of mortality risk
- even significant fractions of larger message spaces such as age or length of stay could be recovered
- frequency analysis and $l_2$- and $l_3$-optimization performed equally well in experiments, $l_1$-optimization performed worse
- advantage of $l_p$-optimization: also produces cost information

# Attacking DTE Columns: Summary

- both attack variants are able to recover a significant amount of cells in a real-world attack scenaria, e.g. 100% of mortality risk
- even significant fractions of larger message spaces such as age or length of stay could be recovered
- frequency analysis and $l_2$- and $l_3$-optimization performed equally well in experiments, $l_1$-optimization performed worse
- advantage of $l_p$-optimization: also produces cost information

# Attacking DTE Columns: Summary

- both attack variants are able to recover a significant amount of cells in a real-world attack scenaria, e.g. 100% of mortality risk
- even significant fractions of larger message spaces such as age or length of stay could be recovered
- frequency analysis and $l_2$- and $l_3$-optimization performed equally well in experiments, $l_1$-optimization performed worse
- advantage of $l_p$-optimization: also produces cost information

# Attacking OPE Columns: Sorting Attack for Dense Columns

1. compute $\psi \leftarrow$ vSort (Unique (c))
2. compute $\boldsymbol{\pi} \leftarrow$ vSort $(\mathbb{M}_k)$
3. output $\alpha : \mathbb{C}_k \rightarrow \mathbb{M}_k$ such that

$$\alpha(c) = \begin{cases} \pi\left[\mathsf{Rank}_{\psi}(c)\right] & \text{if } c \in \mathbf{c} \\ \perp & \text{if } c \notin \mathbf{c} \end{cases}$$

# Attacking OPE Columns: Sorting Attack for Dense Columns

1. compute $\psi \leftarrow$ vSort (Unique (c))
2. compute $\boldsymbol{\pi} \leftarrow$ vSort $(\mathbb{M}_k)$
3. output $\alpha : \mathbb{C}_k \rightarrow \mathbb{M}_k$ such that

$$\alpha(c) = \begin{cases} \pi\left[\mathsf{Rank}_{\psi}(c)\right] & \text{if } c \in \mathbf{c} \\ \perp & \text{if } c \notin \mathbf{c} \end{cases}$$

# Attacking OPE Columns: Cumulative Attack

1. compute $\psi \leftarrow$ vSort ( Hist ($\mathbf{c}$)) and $\varphi \leftarrow$ CDF($\mathbf{c}$)
2. compute $\boldsymbol{\pi} \leftarrow$ vSort ( Hist ($\mathbf{z}$)) and $\mu \leftarrow$ CDF($\mathbf{z}$)
3. output: $\arg\min_{X \in \mathbb{P}} \sum_{i=1}^{|\mathbb{M}_k|} (|\psi_i - X_i \cdot \pi| + |\varphi_i - X_i \cdot \boldsymbol{\mu}|)$

# Attacking OPE Columns: Cumulative Attack

1. compute $\psi \leftarrow$ vSort ( Hist ($\mathbf{c}$)) and $\varphi \leftarrow$ CDF($\mathbf{c}$)
2. compute $\boldsymbol{\pi} \leftarrow$ vSort ( Hist ($\mathbf{z}$)) and $\mu \leftarrow$ CDF($\mathbf{z}$)
3. output: $\arg\min_{X \in \mathbb{P}} \sum_{i=1}^{|\mathbb{M}_k|} (|\psi_i - X_i \cdot \pi| + |\varphi_i - X_i \cdot \boldsymbol{\mu}|)$

# Attacking OPE Columns: Cumulative Attack

1. compute $\psi \leftarrow$ vSort ( Hist ($\mathbf{c}$)) and $\varphi \leftarrow$ CDF($\mathbf{c}$)
2. compute $\boldsymbol{\pi} \leftarrow$ vSort ( Hist ($\mathbf{z}$)) and $\mu \leftarrow$ CDF($\mathbf{z}$)
3. output: $\arg\min_{X \in \mathbb{P}} \sum_{i=1}^{|\mathbb{M}_k|} (|\psi_i - X_i \cdot \pi| + |\varphi_i - X_i \cdot \boldsymbol{\mu}|)$

# Attacking OPE Columns: Summary

- if column is dense, then the sorting attack can recover all cells, otherwise it fails
- cumulative attack performs extremely well even for low-density attributes

# Attacking OPE Columns: Summary

- if column is dense, then the sorting attack can recover all cells, otherwise it fails
- cumulative attack performs extremely well even for low-density attributes

# Opinion and Thoughts

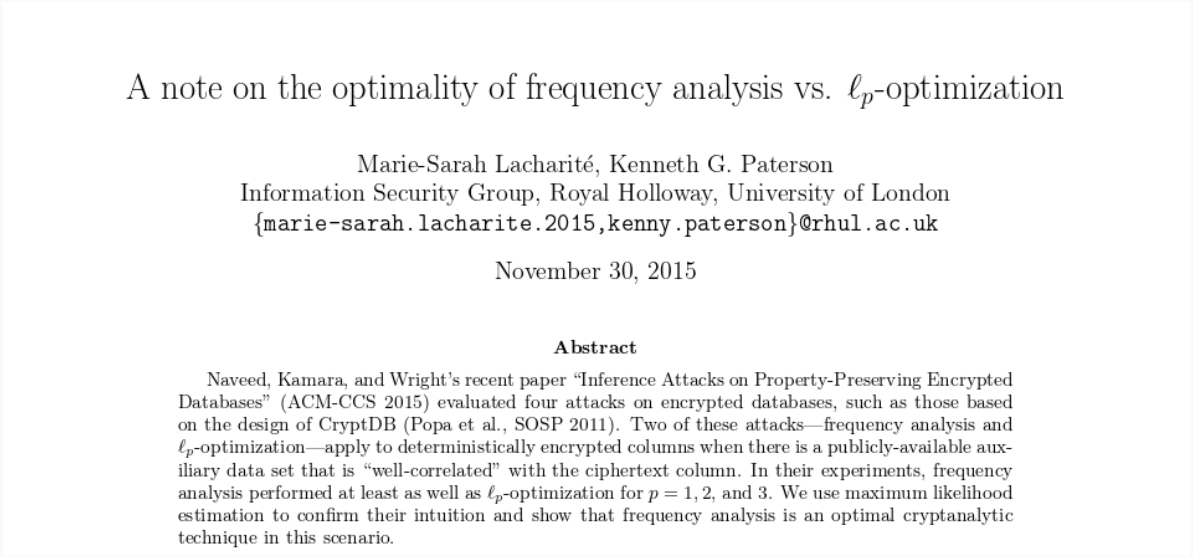
# Opinion and Thoughts: Unclear/Informal Wording

## 9.2 Attacks on OPE-Encrypted Columns

The sorting attack succeeds only if a column has density 1, meaning that all possible values of an attribute are present in both the target and the auxiliary data. If this condition

The sorting attack actually does not need any auxiliary data

# Opinion and Thoughts: Unclear/Informal Wording



A note on the optimality of frequency analysis vs. $\ell_p$-optimization

Marie-Sarah Lacharité, Kenneth G. Paterson
Information Security Group, Royal Holloway, University of London
{marie-sarah.lacharite.2015,kenny.paterson}@rhul.ac.uk

November 30, 2015

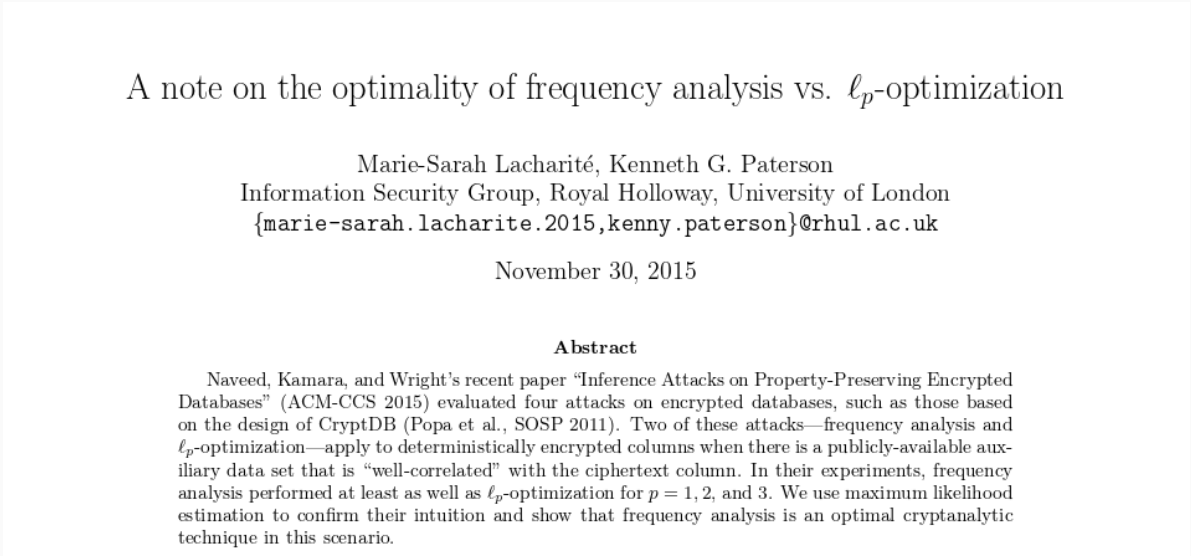
**Abstract**

Naveed, Kamara, and Wright's recent paper "Inference Attacks on Property-Preserving Encrypted Databases" (ACM-CCS 2015) evaluated four attacks on encrypted databases, such as those based on the design of CryptDB (Popa et al., SOSP 2011). Two of these attacks—frequency analysis and $\ell_p$-optimization—apply to deterministically encrypted columns when there is a publicly-available auxiliary data set that is "well-correlated" with the ciphertext column. In their experiments, frequency analysis performed at least as well as $\ell_p$-optimization for $p = 1, 2$, and 3. We use maximum likelihood estimation to confirm their intuition and show that frequency analysis is an optimal cryptanalytic technique in this scenario.

"We use the language of statistics to state explicitly **what we believe** is Naveed, Kamara, and Wright's assumption"

(Lacharité and Paterson 2015)

# Opinion and Thoughts: The $l_p$-Optimization



A note on the optimality of frequency analysis vs. $\ell_p$-optimization

Marie-Sarah Lacharité, Kenneth G. Paterson
Information Security Group, Royal Holloway, University of London
{marie-sarah.lacharite.2015,kenny.paterson}@rhul.ac.uk

November 30, 2015

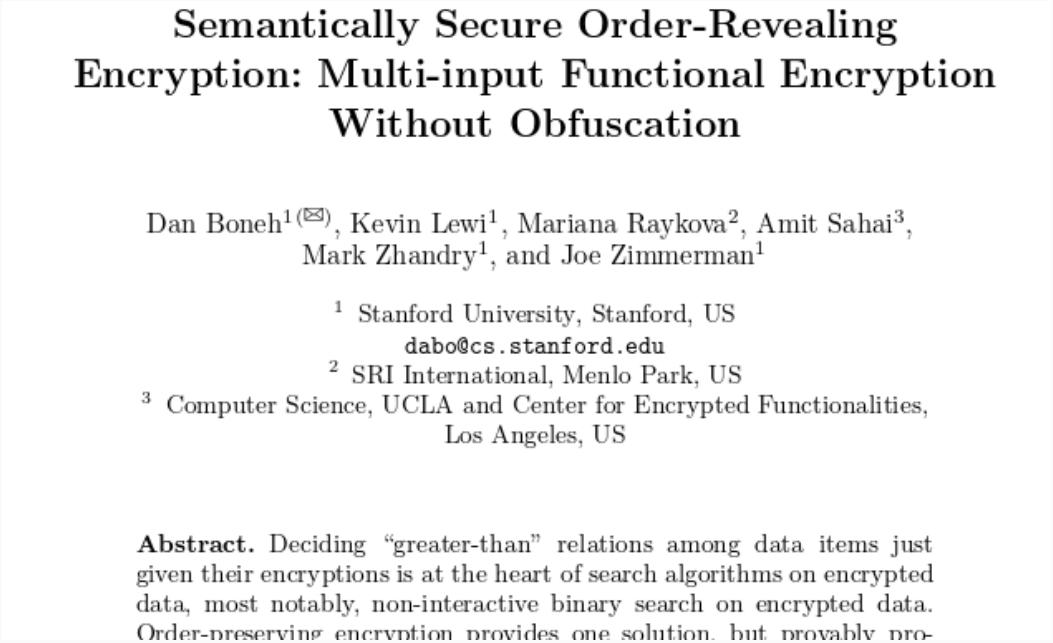
**Abstract**

Naveed, Kamara, and Wright's recent paper "Inference Attacks on Property-Preserving Encrypted Databases" (ACM-CCS 2015) evaluated four attacks on encrypted databases, such as those based on the design of CryptDB (Popa et al., SOSP 2011). Two of these attacks—frequency analysis and $\ell_p$-optimization—apply to deterministically encrypted columns when there is a publicly-available auxiliary data set that is "well-correlated" with the ciphertext column. In their experiments, frequency analysis performed at least as well as $\ell_p$-optimization for $p = 1, 2$, and 3. We use maximum likelihood estimation to confirm their intuition and show that frequency analysis is an optimal cryptanalytic technique in this scenario.

"Therefore, the most likely permutation $\pi$ is the one that assigns the most frequent plaintext in the auxiliary data to the most frequent ciphertext in the encrypted column, and so on. This permutation is simply the frequency analysis attack mentioned in the first section."

(Lacharité and Paterson 2015)

# Opinion and Thoughts: Attacking Advanced PPE-Schemes



**Semantically Secure Order-Revealing Encryption: Multi-input Functional Encryption Without Obfuscation**

Dan Boneh[1](✉), Kevin Lewi[1], Mariana Raykova[2], Amit Sahai[3], Mark Zhandry[1], and Joe Zimmerman[1]

[1] Stanford University, Stanford, US
dabo@cs.stanford.edu
[2] SRI International, Menlo Park, US
[3] Computer Science, UCLA and Center for Encrypted Functionalities, Los Angeles, US

**Abstract.** Deciding "greater-than" relations among data items just given their encryptions is at the heart of search algorithms on encrypted data, most notably, non-interactive binary search on encrypted data. Order-preserving encryption provides one solution, but provably pro-

"A secret-key encryption scheme is order-revealing if there is a **public procedure** that takes two encrypted plaintexts as input and reports their lexicographic ordering."

(Boneh et al. 2015)

# Opinion and Thoughts: Attacking Advanced PPE-Schemes

**Randomly Partitioned Encryption for Cloud Databases**

Tahmineh Sanamrad[1], Lucas Braun[1], Donald Kossmann[1],
and Ramarathnam Venkatesan[2]

[1] Systems Group, Computer Science Departement, ETH Zurich, Switzerland
{sanamrat,braunl,donaldk}@inf.ethz.ch
[2] Microsoft Research, Redmond CA, USA
venkie@microsoft.com

**Abstract.** With the current advances in Cloud Computing, outsourcing data has never been so tempting. Along with outsourcing a database comes the privacy versus performance discussion. Order-Preserving Encryption (OPE) is one of the most attractive techniques for database encryption since it allows to execute range and rank queries efficiently without decrypting the data. On the other hand, people are reluctant to use OPE-based techniques in practice because of their

"The main idea of RPE is to **randomly partition the domain and apply an order preserving encryption scheme to each partition**. This makes RPE a partially order-preserving encryption as each partition is ordered, but the total order is hidden."

(Sanamrad et al. 2014)

# Opinion and Thoughts: Everything Already Known?

**Randomly Partitioned Encryption for Cloud Databases**

Tahmineh Sanamrad[1], Lucas Braun[1], Donald Kossmann[1],
and Ramarathnam Venkatesan[2]

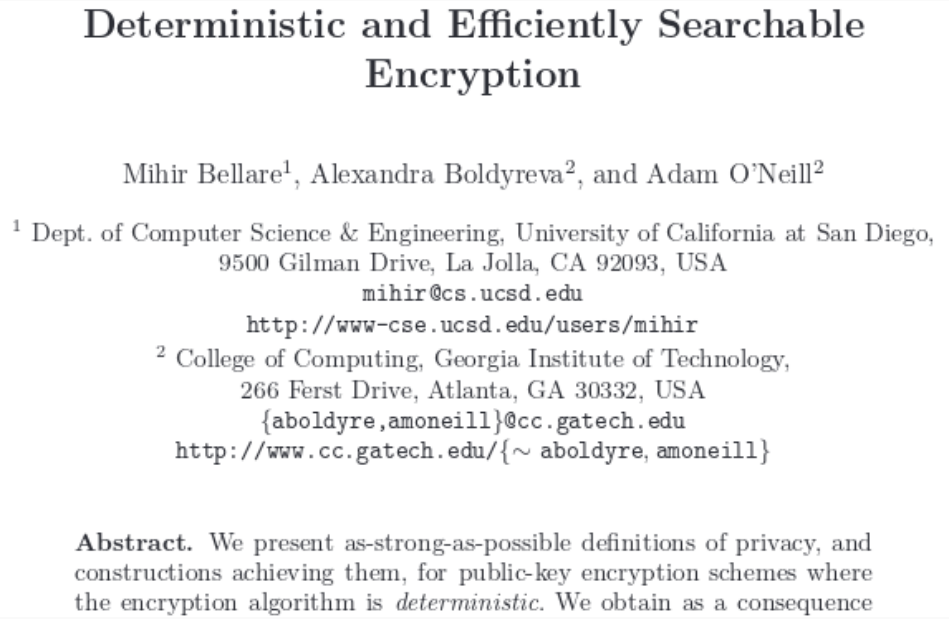
[1] Systems Group, Computer Science Departement, ETH Zurich, Switzerland
{sanamrat,braunl,donaldk}@inf.ethz.ch
[2] Microsoft Research, Redmond CA, USA
venkie@microsoft.com

**Abstract.** With the current advances in Cloud Computing, outsourcing data has never been so tempting. Along with outsourcing a database comes the privacy versus performance discussion. Order-Preserving Encryption (OPE) is one of the most attractive techniques for database encryption since it allows to execute range and rank queries efficiently without decrypting the data. On the other hand, people are reluctant to use OPE-based techniques in practice because of their

"However, the order relationship between plaintext and ciphertext remains intact after encryption, making it an **easy target for a Domain Attack**. Moreover, being deterministic, makes OPE particularly **vulnerable against Frequency Attacks.**"

(Sanamrad et al. 2014)

# Opinion and Thoughts: Everything Already Known?



Deterministic and Efficiently Searchable Encryption

Mihir Bellare[1], Alexandra Boldyreva[2], and Adam O'Neill[2]

[1] Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA
mihir@cs.ucsd.edu
http://www-cse.ucsd.edu/users/mihir

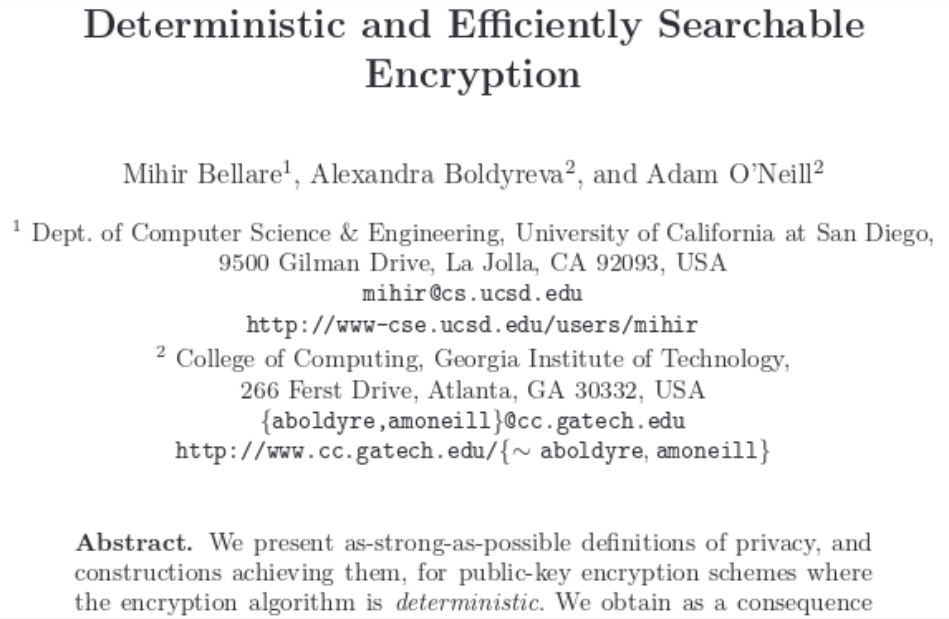
[2] College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332, USA
{aboldyre,amoneill}@cc.gatech.edu
http://www.cc.gatech.edu/{∼ aboldyre, amoneill}

**Abstract.** We present as-strong-as-possible definitions of privacy, and constructions achieving them, for public-key encryption schemes where the encryption algorithm is *deterministic*. We obtain as a consequence

"First, no privacy is possible if the plaintext is known to come from a **small space.**"

(Bellare, Boldyreva, and O'Neill 2007)

# Opinion and Thoughts: Everything Already Known?



Deterministic and Efficiently Searchable Encryption

Mihir Bellare[1], Alexandra Boldyreva[2], and Adam O'Neill[2]

[1] Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA
mihir@cs.ucsd.edu
http://www-cse.ucsd.edu/users/mihir

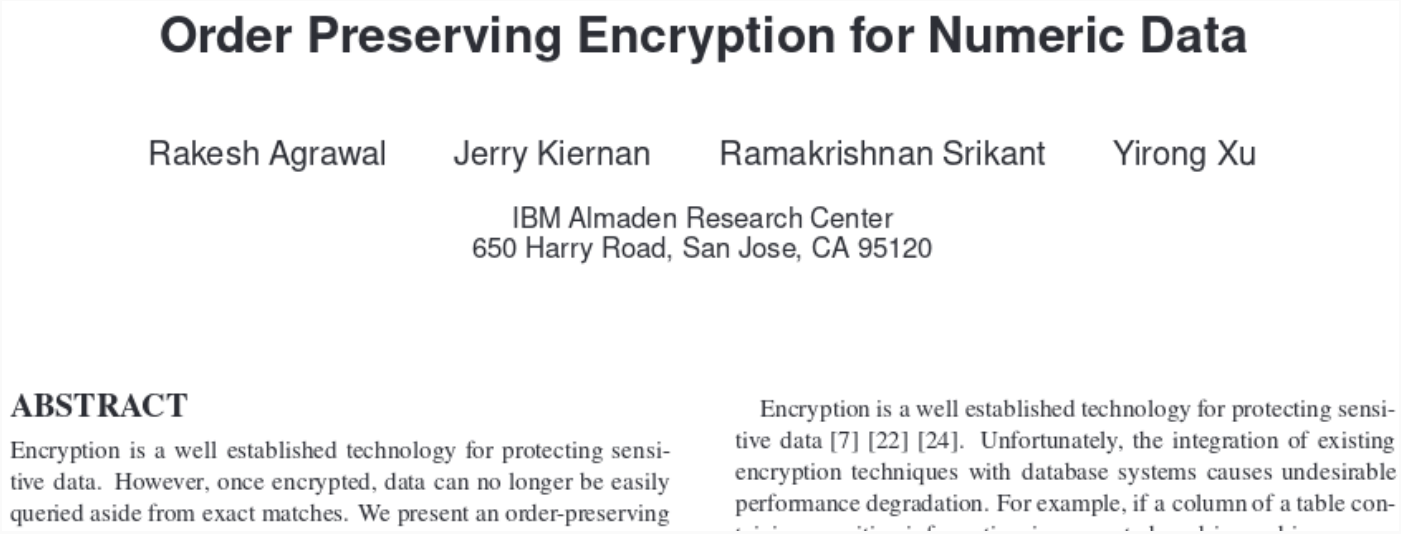
[2] College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332, USA
{aboldyre,amoneill}@cc.gatech.edu
http://www.cc.gatech.edu/{∼ aboldyre, amoneill}

**Abstract.** We present as-strong-as-possible definitions of privacy, and constructions achieving them, for public-key encryption schemes where the encryption algorithm is *deterministic*. We obtain as a consequence

"Our schemes only provide privacy for plaintexts that have high min-entropy. (This is **inherent in being deterministic or efficiently searchable**, not a weakness of our particular constructs.)"

(Bellare, Boldyreva, and O'Neill 2007)

# Opinion and Thoughts: Everything Already Known?



**Order Preserving Encryption for Numeric Data**

Rakesh Agrawal Jerry Kiernan Ramakrishnan Srikant Yirong Xu

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

**ABSTRACT**

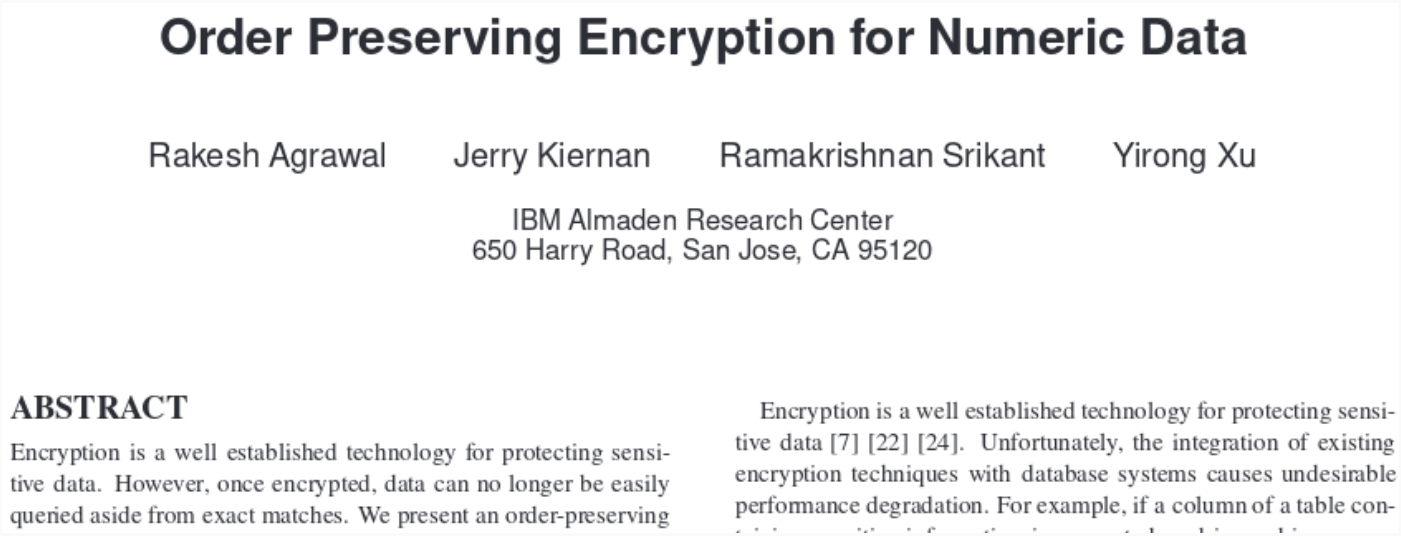
Encryption is a well established technology for protecting sensitive data. However, once encrypted, data can no longer be easily queried aside from exact matches. We present an order-preserving

Encryption is a well established technology for protecting sensitive data [7] [22] [24]. Unfortunately, the integration of existing encryption techniques with database systems causes undesirable performance degradation. For example, if a column of a table con-

"The proposed scheme has been designed to be deployed in application environments in which the intruder can get access to the encrypted database, but **does not have prior domain information** such as the distribution of values"

(Agrawal et al. 2004)

# Opinion and Thoughts: Everything Already Known?



**Order Preserving Encryption for Numeric Data**

Rakesh Agrawal Jerry Kiernan Ramakrishnan Srikant Yirong Xu

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

**ABSTRACT**

Encryption is a well established technology for protecting sensitive data. However, once encrypted, data can no longer be easily queried aside from exact matches. We present an order-preserving

Encryption is a well established technology for protecting sensitive data [7] [22] [24]. Unfortunately, the integration of existing encryption techniques with database systems causes undesirable performance degradation. For example, if a column of a table con-

"Similarly, any order-preserving encryption is **not secure** against tight estimation exposure if the adversary can **guess the domain and knows the distribution** of values in that domain."

(Agrawal et al. 2004)

# Opinion and Thoughts: Realistic Scenario?


"One of the creators of CryptDB, Raluca Ada Popa, said she did not believe the findings proved CryptDB weak as the flawed pieces of the software were **not designed to handle sensitive information**. She said OPE encryption should be **used for 'high-entropy values'** where the order does not reveal much and that CryptDB was still a worthy way to protect information."

(Brewster 2015)

# Opinion and Thoughts: Realistic Scenario?

Guidelines for Using the CryptDB System Securely

Raluca Ada Popa
UC Berkeley

Nickolai Zeldovich
MIT CSAIL

Hari Balakrishnan
MIT CSAIL

## 1 Introduction

This report has two goals. First, we review guidelines for using the CryptDB system [PRZB11, Pop14] securely by the administrators of database applications. These guidelines were already described in [PRZB11] and elaborated on in [Pop14], but in light of some recent work [NKW15] that applied these guidelines incorrectly, a short document devoted to summarizing these guidelines may be useful.

"Hence, the conclusions drawn in that paper regarding CryptDB's guarantees for medical applications are incorrect: had the **guidelines been followed, none of the claimed attacks would have been possible.**"

(Popa, Zeldovich, and Balakrishnan 2015)

# Opinion and Thoughts: Realistic Scenario?


https://www.forbes.com/sites/thomasbrewster/2015/09/03/microsoft-dumb-attacks-cracks-next-gen-cryptography/

Forbes Billionaires Innovation Leadership Money Business Small Business

18,407 views | Sep 3, 2015, 10:20am

**How A 1200-Year-Old Hacking Technique Can Already Crack Tomorrow's Encrypted Vaults**

Thomas Brewster Forbes Staff

Cybersecurity

*Associate editor at Forbes, covering cybercrime, privacy, security and surveillance.*

"Professor Ross Anderson, a crypto luminary from the University of Cambridge Computer Laboratory, believes the research proves what he always thought: such schemes aren't worth using at all. '**Hopefully nobody will be dumb enough to rely on such schemes to protect real data**. The bad news is that dozens of cryptography researchers have spent years of their lives building this stuff.'"

(Brewster 2015)

# Elektronisches Patientendossier