Frequency-hiding Dependency-preserving Encryption for Outsourced Databases

ICDE'17

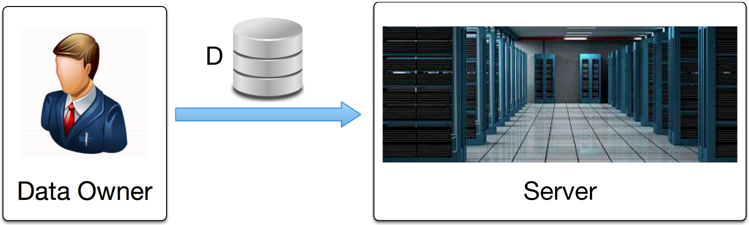
Boxiang Dong [1] Wendy Wang [2]

[1]Montclair State University
Montclair, NJ

[2]Stevens Institute of Technology
Hoboken, NJ

April 20, 2017

Data-Management-as-a-Service (DMaS)



- Data owner with limited computational resources
- Computationally powerful server (e.g. cloud)
- Outsourcing provides a cost-effective solution for data management.

Functional Dependency (FD)

Definition A FD $X \rightarrow Y$ states that for any records r_1 and r_2, $r_1[X] = r_2[X]$ demands that $r_1[Y] = r_2[Y]$.

Applications

- Data schema improvement via normalization
- Data inconsistency repair

Outsourcing Requirement

Privacy Concern

- Protect the sensitive information from untrusted server.
- Encrypt the dataset before outsourcing.

Utility Concern

- Support FD-based applications.
- The encryption scheme should preserve FDs.

Challenges

Directly applying deterministic encryption (e.g. RSA) is vulnerable against the *frequency-analysis attack (FA attack)* [N+15].

FA-Attack$(\mathcal{P}, \mathcal{E})$

1. compute $\pi \leftarrow vSort(Hist(\mathcal{P}))$
2. compute $\varphi \leftarrow vSort(Hist(\mathcal{E}))$
3. foreach $e \in \mathcal{E}$

   output p if $Rank_{\varphi}(e) = Rank_{\pi}(p)$

ID	A	B	C
r_1	a_1	b_1	c_1
r_2	a_1	b_1	c_2
r_3	a_1	b_1	c_4
r_4	a_1	b_1	c_3
r_5	a_2	b_2	c_3
r_6	a_2	b_2	c_4

(a) Base table D ($A \rightarrow B$ $A \not\rightarrow C$, $B \not\rightarrow C$)

ID	A	B	C
r_1	$\hat{a}_1$	$\hat{b}_1$	$\hat{c}_1$
r_2	$\hat{a}_1$	$\hat{b}_1$	$\hat{c}_2$
r_3	$\hat{a}_1$	$\hat{b}_1$	$\hat{c}_4$
r_4	$\hat{a}_1$	$\hat{b}_1$	$\hat{c}_3$
r_5	$\hat{a}_2$	$\hat{b}_2$	$\hat{c}_3$
r_6	$\hat{a}_2$	$\hat{b}_2$	$\hat{c}_4$

(b) $\hat{D}_1$: deterministic encryption

Challenges

Applying probabilistic encryption may *destroy* original FDs or introduce *false positive* FDs.

ID	A	B	C
r_1	$\hat{a}_1^1$	$\hat{b}_1^1$	$\hat{c}_1^1$
r_2	$\hat{a}_1^2$	$\hat{b}_1^2$	$\hat{c}_2^1$
r_3	$\hat{a}_1^3$	$\hat{b}_1^3$	$\hat{c}_4^2$
r_4	$\hat{a}_1^4$	$\hat{b}_1^4$	$\hat{c}_3^1$
r_5	$\hat{a}_2^1$	$\hat{b}_2^1$	$\hat{c}_3^2$
r_6	$\hat{a}_2^1$	$\hat{b}_2^2$	$\hat{c}_4^1$

(c) $\hat{D}_2$: probabilistic encryption on A, B, C individually
Original FD $A \rightarrow B$ destroyed

ID	A	B	C
r_1	$\hat{a}_1^1$	$\hat{b}_1^1$	$\hat{c}_1^1$
r_2	$\hat{a}_1^2$	$\hat{b}_1^2$	$\hat{c}_2^2$
r_3	$\hat{a}_1^3$	$\hat{b}_1^3$	$\hat{c}_4^3$
r_4	$\hat{a}_1^4$	$\hat{b}_1^4$	$\hat{c}_3^4$
r_5	$\hat{a}_2^5$	$\hat{b}_2^5$	$\hat{c}_3^5$
r_6	$\hat{a}_2^6$	$\hat{b}_2^6$	$\hat{c}_4^6$

(d) $\hat{D}_3$: probabilistic encryption on (A, B, C)
False positive FD $A \rightarrow C$ introduced

Challenges

The FD-preserving property introduces new inference attack [PR12].

$$(D_0, FD_0), (D_1, FD_1)$$

$$\downarrow$$

$$D_b \text{ s.t. } b \xleftarrow{\$} \{0, 1\}$$

$$\downarrow$$

$$\boxed{\textit{FD-preserving CPA-secure cipher}}$$

$$\downarrow$$

$$\hat{D}_b$$

$$\downarrow$$

$$b' = \begin{cases} 0 & \text{if } FD_0 \text{ holds on } \hat{D}_b \\ 1 & \text{otherwise} \end{cases}$$

Our Contributions

Security Definition

- $\alpha -$ *security* against *FA*-attack
- *Indistinguishability against FD-preserving chosen plaintext attack* (IND-FCPA)

Encryption Scheme

We design F^2, a frequency-hiding, FD-preserving encryption scheme based on probabilistic encryption.

Outline

Related Work

Privacy-preserving outsourced computing

- Data encoding [H+02a, H+02b]
- Data encryption [S+00, P+12]
- Property-preserving encryption [Ker15, B+11, G+06, B+09]

Inference attack

- FA attack [N+15]
- Query-recovery attack [I+12]

FD applications

- Data cleaning [T+11]
- Schema design [BFFR05, B+07]

Security Model

$$
\begin{array}{r}
\textbf{Experiment} \quad Exp_{\Pi}^{FA}() \\
p' \leftarrow A^{freq_{\mathcal{E}}(e), freq(\mathcal{P})} \\
Return\ 1\ if\ p' = Decrypt(k, e) \\
Return\ 0\ otherwise
\end{array}
$$

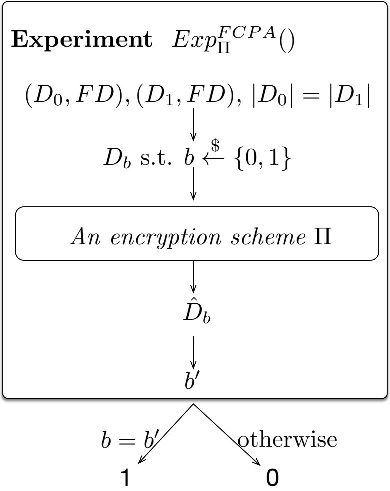
$Adv_{\Pi}^{FA}(A) = Prob(Exp_{\Pi}^{FA}(A) = 1)$ measures the success rate of FA attack.

Definition (α-security against FA Attack)

An encryption scheme Π is α-secure against FA if for every adversary A it holds that $Adv_{\Pi}^{FA}(A) \leq \alpha$, where $\alpha \in (0, 1]$ is user specified.

Security Model

The server may exploit the FDs to break the cipher.



Security Model

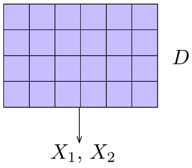
$Adv_{\Pi}^{FCPA}(A) = Prob(Exp_{\Pi}^{FCPA}(A) = 1) - 1/2$ measures the advantage of the *FCPA*-attack over a random guess.

Definition (Indistinguishability against FD-preserving Chosen Plaintext Attack (IND-FCPA))

An encryption scheme Π is IND-FCPA if for any polynomial-time adversary A, it holds that the advantage is negligible in λ, i.e., $Adv_{\Pi}^{FCPA}(A) = negl(\lambda)$, where λ is a pre-defined security parameter.

F^2 Encryption Scheme - Overview

F^2, a frequency-hiding FD-preserving encryption scheme, consists of four steps.



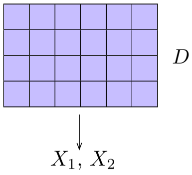
Step 1. Identifying Maximal Attribute Sets

F^2 Encryption Scheme - Overview

F^2, a frequency-hiding FD-preserving encryption scheme, consists of four steps.

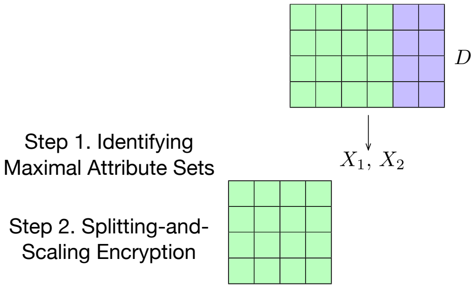
Step 1. Identifying Maximal Attribute Sets

Step 2. Splitting-and-Scaling Encryption



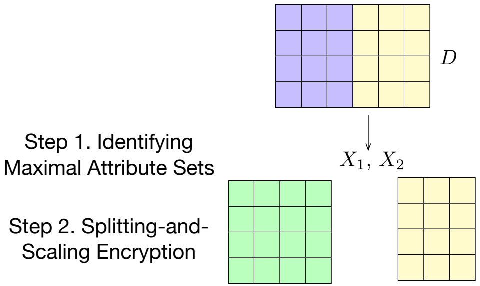
F^2 Encryption Scheme - Overview

F^2, a frequency-hiding FD-preserving encryption scheme, consists of four steps.



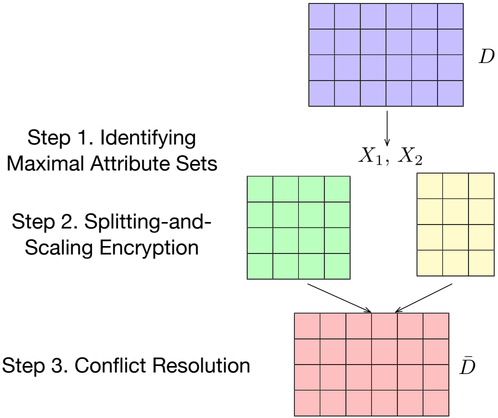
F^2 Encryption Scheme - Overview

F^2, a frequency-hiding FD-preserving encryption scheme, consists of four steps.



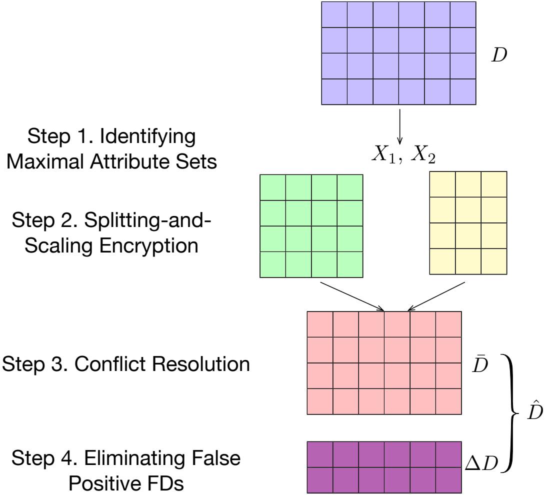
F^2 Encryption Scheme - Overview

F^2, a frequency-hiding FD-preserving encryption scheme, consists of four steps.



F^2 Encryption Scheme - Overview

F^2, a frequency-hiding FD-preserving encryption scheme, consists of four steps.



Step 1 - Identifying Maximal Attribute Sets

Theorem

Given a dataset D and a FD $X \rightarrow Y$, if we apply *probabilistic encryption scheme* on attribute set $\mathcal{A}$ and get $\hat{D}$, then $\hat{D}$ preserves $X \rightarrow Y$ if $(X \cup Y) \subseteq \mathcal{A}$.

Step 1 - Identifying Maximal Attribute Sets

Definition (Maximum Attribute Set (*MAS*))

Given a dataset D, an attribute set $\mathcal{A}$ is a *MAS* if:
(1) there exists at least an instance of $\mathcal{A}$ whose number of occurrences is larger than 1; and
(2) no superset of $\mathcal{A}$ satisfies this requirement.

Step 1 - Identifying Maximal Attribute Sets

Lemma

Given a dataset D and a FD $X \rightarrow Y$, there must exist at least a *MAS* M such that $(X \cup Y) \subseteq M$.

Step 1 - Identifying Maximal Attribute Sets

- To preserve *FD*s, we need to find the *MAS*s from the dataset.
- We adapt an efficient solution named *Ducc* [H+13].
- The complexity is much lower than FD discovery.

ID	A	B	C
r_1	a_2	b_1	c_1
r_2	a_1	b_1	c_1
r_3	a_1	b_1	c_2
r_4	a_3	b_1	c_2
r_5	a_4	b_2	c_2
r_6	a_5	b_2	c_3

$$FD : A \to B$$

Step 1 - Identifying Maximal Attribute Sets

- To preserve *FD*s, we need to find the *MAS*s from the dataset.
- We adapt an efficient solution named *Ducc* [H+13].
- The complexity is much lower than FD discovery.

ID	A	B	C
r_1	a_2	b_1	c_1
r_2	a_1	b_1	c_1
r_3	a_1	b_1	c_2
r_4	a_3	b_1	c_2
r_5	a_4	b_2	c_2
r_6	a_5	b_2	c_3

$$FD : A \rightarrow B$$
$$MAS = \{AB, BC\}$$

Step 1 - Identifying Maximal Attribute Sets

- To preserve *FD*s, we need to find the *MAS*s from the dataset.
- We adapt an efficient solution named *Ducc* [H+13].
- The complexity is much lower than FD discovery.

ID	A	B	C
r_1	a_2	b_1	c_1
r_2	a_1	b_1	c_1
r_3	a_1	b_1	c_2
r_4	a_3	b_1	c_2
r_5	a_4	b_2	c_2
r_6	a_5	b_2	c_3

$$FD : A \rightarrow B$$
$$MAS = \{AB, BC\}$$

Step 2 - Splitting-and-Scaling Encryption

for all *MAS* **do**
    Construct *equivalence classes (ECs)*
end for

ID	B	C	
r_1	b_1	c_1	C_1
r_2	b_1	c_1	
r_3	b_1	c_2	C_2
r_4	b_1	c_2	
r_5	b_2	c_2	C_3
r_6	b_2	c_3	C_4

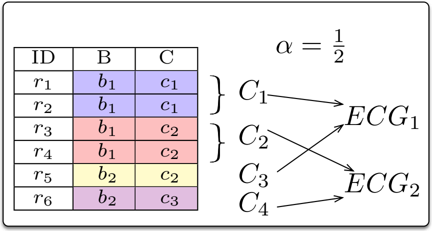
Step 2 - Splitting-and-Scaling Encryption

for all *MAS* **do**

Construct *equivalence classes (ECs)*

Organize *EC*s into collision-free groups of size at least $\frac{1}{\alpha}$

end for

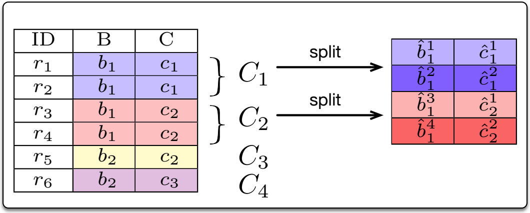


Step 2 - Splitting-and-Scaling Encryption

for all *MAS* **do**
  Construct *equivalence classes (ECs)*
  Organize *EC*s into collision-free groups of size at least $\frac{1}{\alpha}$
  Apply splitting and scaling to reach the same frequency
end for

Splitting Split a *EC* into ω copies with the same frequency.

Scaling Duplicate a *EC* to reach frequency homogenization.

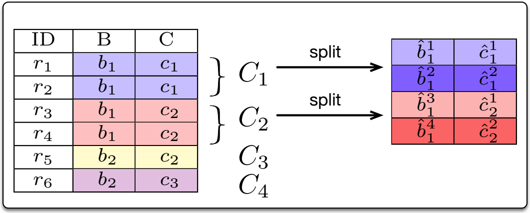


Step 2 - Splitting-and-Scaling Encryption

---

for all *MAS* **do**
  Construct *equivalence classes (ECs)*
  Organize *EC*s into collision-free groups
  Apply splitting and scaling to reach the same frequency
end for

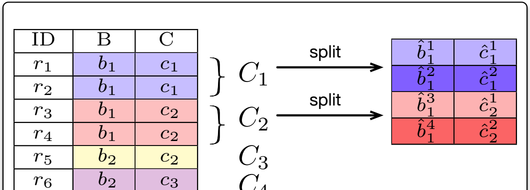
We design an algorithm to decide the splitting and scaling strategy to minimize the amount of duplications.



Step 2 - Splitting-and-Scaling Encryption

for all *MAS* **do**
  Construct *equivalence classes (ECs)*
  Organize *EC*s into collision-free groups
  Apply splitting and scaling to reach the same frequency
  Encrypt each *EC*
end for

For each unique plaintext value p, it is encrypted as $e =< r, F_k(r) \oplus p >$, where r is a random value, and F_k is a pseudorandom function.



Step 2 - Splitting-and-Scaling Encryption

for all *MAS* **do**
  Construct *equivalence classes (ECs)*
  Organize *EC*s into collision-free groups
  Apply splitting and scaling to reach the same frequency
  Encrypt each *EC*
end for

ID	B	C
r_1	b_1	c_1
r_2	b_1	c_1
r_3	b_1	c_2
r_4	b_1	c_2
r_5	b_2	c_2
r_6	b_2	c_3

$\longrightarrow$

ID	B	C
r_1	$\hat{b}_1^1$	$\hat{c}_1^1$
r_2	$\hat{b}_1^2$	$\hat{c}_1^2$
r_3	$\hat{b}_1^3$	$\hat{c}_2^1$
r_4	$\hat{b}_1^4$	$\hat{c}_2^2$
r_5	$\hat{b}_2^1$	$\hat{c}_2^3$
r_6	$\hat{b}_2^2$	$\hat{c}_3^1$

Step 3 - Conflict Resolution

- In Step 2, we apply encryption to each *MAS* independently.

ID	A	B
r_1	$\hat{a}_2^1$	$\hat{b}_1^1$
r_2	$\hat{a}_1^1$	$\hat{b}_1^2$
r_3	$\hat{a}_1^1$	$\hat{b}_1^2$
r_4	$\hat{a}_3^1$	$\hat{b}_1^4$
r_5	$\hat{a}_4^1$	$\hat{b}_2^1$
r_6	$\hat{a}_5^1$	$\hat{b}_2^2$

$Enc(D[AB])$

ID	B	C
r_1	$\hat{b}_1^1$	$\hat{c}_1^1$
r_2	$\hat{b}_1^2$	$\hat{c}_1^2$
r_3	$\hat{b}_1^3$	$\hat{c}_2^1$
r_4	$\hat{b}_1^4$	$\hat{c}_2^2$
r_5	$\hat{b}_2^1$	$\hat{c}_2^3$
r_6	$\hat{b}_2^2$	$\hat{c}_3^1$

$Enc(D[BC])$

Step 3 - Conflict Resolution

- In Step 2, we apply encryption to each *MAS* independently.
- However, there may exist **conflicts** between different *MAS*s.

ID	A	B
r_1	$\hat{a}_2^1$	$\hat{b}_1^1$
r_2	$\hat{a}_1^1$	$\hat{b}_1^2$
r_3	$\hat{a}_1^1$	$\hat{b}_1^2$
r_4	$\hat{a}_3^1$	$\hat{b}_1^4$
r_5	$\hat{a}_4^1$	$\hat{b}_2^1$
r_6	$\hat{a}_5^1$	$\hat{b}_2^2$

$Enc(D[AB])$

ID	B	C
r_1	$\hat{b}_1^1$	$\hat{c}_1^1$
r_2	$\hat{b}_1^2$	$\hat{c}_1^2$
r_3	$\hat{b}_1^3$	$\hat{c}_2^1$
r_4	$\hat{b}_1^4$	$\hat{c}_2^2$
r_5	$\hat{b}_2^1$	$\hat{c}_2^3$
r_6	$\hat{b}_2^2$	$\hat{c}_3^1$

$Enc(D[BC])$

ID	A	B	C
r_1	$\hat{a}_2^1$	$\hat{b}_1^1$	$\hat{c}_1^1$
r_2	$\hat{a}_1^1$	$\hat{b}_1^2$	$\hat{c}_1^1$
r_3	$\hat{a}_1^1$	$\hat{b}_1^2$ / $\hat{b}_1^3$	$\hat{c}_2^1$
r_4	$\hat{a}_3^1$	$\hat{b}_1^4$	$\hat{c}_2^2$
r_5	$\hat{a}_4^1$	$\hat{b}_2^1$	$\hat{c}_2^3$
r_6	$\hat{a}_5^1$	$\hat{b}_2^2$	$\hat{c}_3^1$

Step 3 - Conflict Resolution

- In Step 2, we apply encryption to each *MAS* independently.
- However, there may exist conflicts between different *MAS*s.
- We design an efficient algorithm to resolve the conflicts.

ID	A	B	C
r_1	$\hat{a}_2^1$	$\hat{b}_1^1$	$\hat{c}_1^1$
r_2	$\hat{a}_1^1$	$\hat{b}_1^2$	$\hat{c}_1^1$
r_3	$\hat{a}_1^1$	$\hat{b}_1^2$ / $\hat{b}_1^3$	$\hat{c}_2^1$
r_4	$\hat{a}_3^1$	$\hat{b}_1^4$	$\hat{c}_2^2$
r_5	$\hat{a}_4^1$	$\hat{b}_2^1$	$\hat{c}_2^3$
r_6	$\hat{a}_5^1$	$\hat{b}_2^2$	$\hat{c}_3^1$

$\downarrow$

ID	A	B	C
r_1	$\hat{a}_2^1$	$\hat{b}_1^1$	$\hat{c}_1^1$
r_2	$\hat{a}_1^1$	$\hat{b}_1^2$	$\hat{c}_1^1$
r_3	$\hat{a}_1^1$	$\hat{b}_1^2$	$\hat{c}_2^4$
r_4	$\hat{a}_3^1$	$\hat{b}_1^4$	$\hat{c}_2^2$
r_5	$\hat{a}_4^1$	$\hat{b}_2^1$	$\hat{c}_2^3$
r_6	$\hat{a}_5^1$	$\hat{b}_2^2$	$\hat{c}_3^1$
r_7	$\hat{a}_1^2$	$\hat{b}_1^3$	$\hat{c}_2^1$

Step 4 - Eliminating False Positive FDs

- Step 1 - 3 may introduce *false positive* FDs.

ID	A	B	C
r_1	a_2	b_1	c_1
r_2	a_1	b_1	c_1
r_3	a_1	b_1	c_2
r_4	a_3	b_1	c_2
r_5	a_4	b_2	c_2
r_6	a_5	b_2	c_3

$FD : A \rightarrow B$

$\longrightarrow$

ID	A	B	C
r_1	$\hat{a}_2^1$	$\hat{b}_1^1$	$\hat{c}_1^1$
r_2	$\hat{a}_1^1$	$\hat{b}_1^2$	$\hat{c}_1^1$
r_3	$\hat{a}_1^1$	$\hat{b}_1^2$	$\hat{c}_2^4$
r_4	$\hat{a}_3^1$	$\hat{b}_1^4$	$\hat{c}_2^2$
r_5	$\hat{a}_4^1$	$\hat{b}_2^1$	$\hat{c}_2^3$
r_6	$\hat{a}_5^1$	$\hat{b}_2^2$	$\hat{c}_3^1$
r_7	$\hat{a}_1^2$	$\hat{b}_1^3$	$\hat{c}_2^1$

$FD : A \rightarrow B$

$B \rightarrow A$

Step 4 - Eliminating False Positive FDs

- Step 1 - 3 may introduce *false positive (FP)* FDs.
- We search for the FP FDs by following the attribute set lattice.
- To break a FP FD $X \rightarrow Y$, we insert two artificial tuples
  - $r_1[X] = r_2[X]$
  - $r_1[Y] \neq r_2[Y]$

ID	A	B	C
r_1	a_2	b_1	c_1
r_2	a_1	b_1	c_1
r_3	a_1	b_1	c_2
r_4	a_3	b_1	c_2
r_5	a_4	b_2	c_2
r_6	a_5	b_2	c_3

$FD : A \rightarrow B$

ID	A	B	C
r_1	$\hat{a}_2^1$	$\hat{b}_1^1$	$\hat{c}_1^1$
r_2	$\hat{a}_1^1$	$\hat{b}_1^2$	$\hat{c}_1^1$
r_3	$\hat{a}_1^1$	$\hat{b}_1^2$	$\hat{c}_2^4$
r_4	$\hat{a}_3^1$	$\hat{b}_1^4$	$\hat{c}_2^2$
r_5	$\hat{a}_4^1$	$\hat{b}_2^1$	$\hat{c}_2^3$
r_6	$\hat{a}_5^1$	$\hat{b}_2^2$	$\hat{c}_3^1$
r_7	$\hat{a}_1^2$	$\hat{b}_1^3$	$\hat{c}_2^1$

r_8	$\hat{a}_3$	$\hat{b}_3$	$\hat{c}_4$
r_9	$\hat{a}_4$	$\hat{b}_3$	$\hat{c}_5$

$\hat{D}$

FD-preserving Property

Theorem (FD-preserving Property)

Given any dataset D, let $\hat{D}$ be the encrypted dataset using Step 1 - 4, it must be true that the FDs on D and $\hat{D}$ are exactly the same.

Security Analysis - FD

Theorem (α-Security against FA Attack)

F^2 provides α-security against the FA attack, i.e., $Adv_{F^2}^{FA}(A) \leq \alpha$.

Theorem (Security against FCPA Attack)

The advantage of FCPA attack against F^2 is $Adv_{F^2}^{FCPA}(A) = \frac{1}{g}$, where g is the minimum number of equivalence classes in a MAS that have the same value on X, Y, and $X \rightarrow Y$ is a valid FD.

In practice, $Adv_{F^2}^{FCPA}(A)$ is very small. ($g = 5,000,000$ for a dataset with 15 million tuples).

Experiments

Testbed 2.5GHz CPU, 60GB RAM, Linux

Datasets

- *Customer* dataset from TPC-C benchmark
  - 906K tuples
  - 21 attributes
- *Orders* dataset from TPC-H benchmark
  - 1.5 million tuples
  - 9 attributes

Baseline

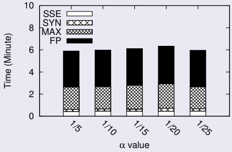
Deterministic *AES*

Probabilistic *Paillier*

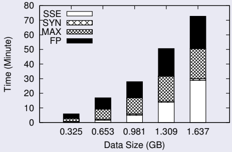
Property-preserving *FHOP* [Ker15]
(frequency-hiding order-preserving)

Time Performance

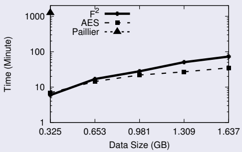
Time Performance (*Orders* Dataset)



(a) Various α values



(b) Various data sizes



(c) Comparison with baselines

- Time performance keeps stable with various α values.
- Time performance is subquadratic to the data size.
- F^2 is as efficient as *AES*, a deterministic encryption scheme.

Security Against FA Attack

Security against FA Attack

Approach	Attack Accuracy
$F^2(\alpha = 0.02)$	**0.01417**
$F^2(\alpha = 0.05)$	**0.03192**
$F^2(\alpha = 0.1)$	**0.0719**
$F^2(\alpha = 0.25)$	**0.1056**
FHOP	0.1214
Paillier	0.1002
AES	0.3395

- Attack accuracy is the fraction of ciphertext that are successfully recovered.
- F^2 provides strong security even for a weak security guarantee ($\alpha = 0.25$).

Conclusion

We design an efficient frequency-hiding FD-preserving encryption scheme, F^2, that:

- Preserves the FDs without requiring the awareness of them.
- Guarantees α-security against FA attack.
- Provides strong security against the FCPA attack.

In the future, we aim at supporting efficient data update.

References I

[B+07] Philip Bohannon et al.
Conditional functional dependencies for data cleaning.
In *IEEE International Conference on Data Engineering*, pages 746–755, 2007.

[B+09] Mihir Bellare et al.
Format-preserving encryption.
In *International Workshop on Selected Areas in Cryptography*, pages 295–312, 2009.

[B+11] Alexandra Boldyreva et al.
Order-preserving encryption revisited: Improved security analysis and alternative solutions.
In *Annual Cryptology Conference*, pages 578–595, 2011.

[BFFR05] Philip Bohannon, Wenfei Fan, Michael Flaster, and Rajeev Rastogi.
A cost-based model and effective heuristic for repairing constraints by value modification.
In *Proceedings of the International Conference on Management of Data*, pages 143–154, 2005.

[G+06] Vipul Goyal et al.
Attribute-based encryption for fine-grained access control of encrypted data.
In *Conference on Computer and Communications Security*, pages 89–98, 2006.

[H+02a] Hakan Hacigumus et al.
Executing sql over encrypted data in the database-service-provider model.
In *ACM International Conference on Management of Data*, pages 216–227, 2002.

[H+02b] Hakan Hacigumus et al.
Providing database as a service.
In *IEEE International Conference on Data Engineering*, pages 29–38, 2002.

References II

[H+13] Arvid Heise et al.
Scalable discovery of unique column combinations.
Proceedings of Very Large Database Endowment, pages 301–312, 2013.

[I+12] Mohammad Saiful Islam et al.
Access pattern disclosure on searchable encryption: Ramification, attack and mitigation.
In *Network and Distributed System Security Symposium*, pages 12–23, 2012.

[Ker15] Florian Kerschbaum.
Frequency-hiding order-preserving encryption.
In *ACM Conference on Computer and Communications Security*, pages 656–667, 2015.

[N+15] Muhammad Naveed et al.
Inference attacks on property-preserving encrypted databases.
In *ACM Conference on Computer and Communications Security*, pages 644–655, 2015.

[P+12] Raluca Ada Popa et al.
Cryptdb: Processing queries on an encrypted database.
Communications of the ACM, pages 103–111, 2012.

[PR12] Omkant Pandey and Yannis Rouselakis.
Property preserving symmetric encryption.
In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 375–391, 2012.

[S+00] Dawn Xiaoding Song et al.
Practical techniques for searches on encrypted data.
In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000.

References III

[T^+11] Nilothpal Talukder et al.
Detecting inconsistencies in private data with secure function evaluation.
Technical report, Purdue University, 2011.

Q & A

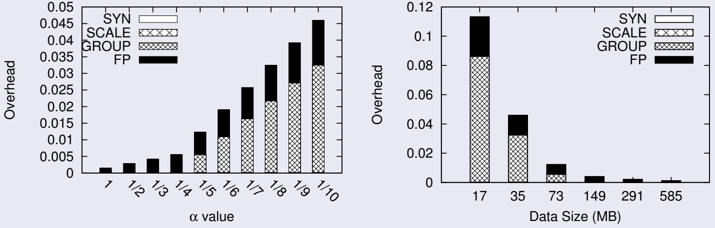
Thank you!

Questions?

dongb@montclair.edu
Hui.Wang@stevens.edu

Storage Overhead

Storage Overhead (*Orders* Dataset)



(a) Various α values

(b) Various data sizes

- *overhead* $= \frac{|\hat{D}|-|D|}{|D|}$ measures the fraction of artificial tuples inserted.
- Strong security requirement (small α value) demands more overhead.
- The overhead is small, especially for large datasets.