

# Buliding a Structurally-Encrypted Relationnal Database

Zheguang Zhao

Brown University  
zheguang.zhao@gmail.com

October 26, 2020

# Data Breach

*A data breach is a security violation in which sensitive, protected or confidential data is copied, transmitted, viewed, stolen or used by an individual unauthorized to do so. – Wikipedia.*

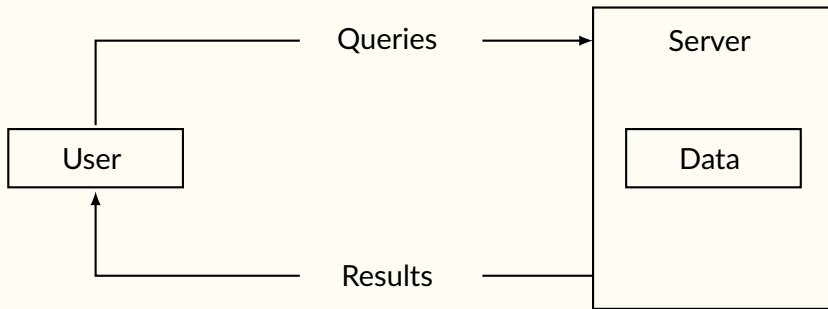
# Data Breach

- “If you have something that you don’t want anyone to know, maybe you shouldn’t be doing it in the first place.” – Eric Schmidt, then-CEO of Google, 2009.
- (In)famous data breach: Quora (2018, 100m), Equifax (2017, 145.5m), Yahoo (2017, every account).

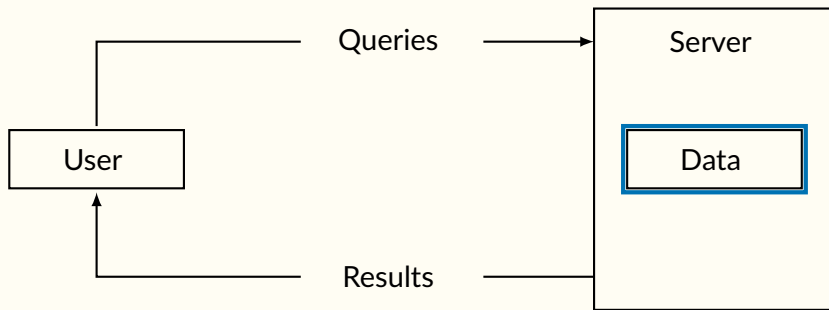
# Data Breach

- “If you have something that you don’t want anyone to know, maybe you shouldn’t be doing it in the first place.” – Eric Schmidt, then-CEO of Google, 2009.
- (In)famous data breach: Quora (2018, 100m), Equifax (2017, 145.5m), Yahoo (2017, every account).
- Just don’t have better tools

# Data Outsourcing to Untrusted Party

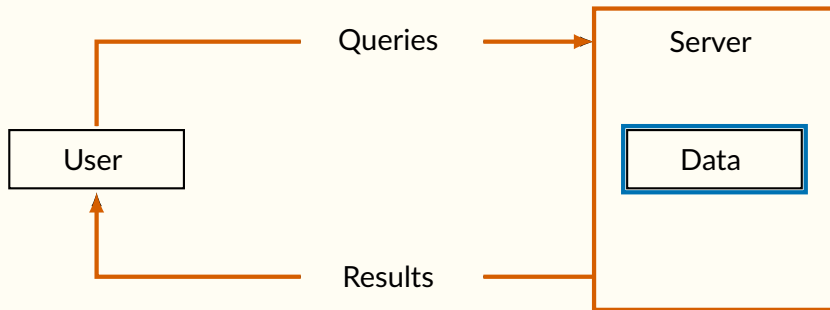


# Data Outsourcing to Untrusted Party



Snapshot attack

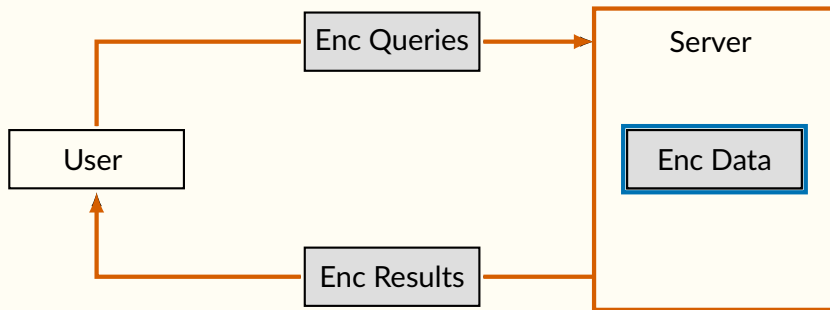
# Data Outsourcing to Untrusted Party



Persistent attack

Snapshot attack

# Data Outsourcing to Untrusted Party

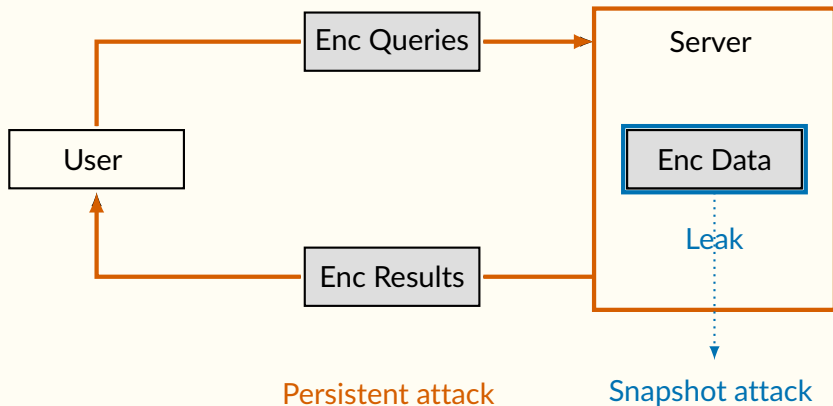


Persistent attack

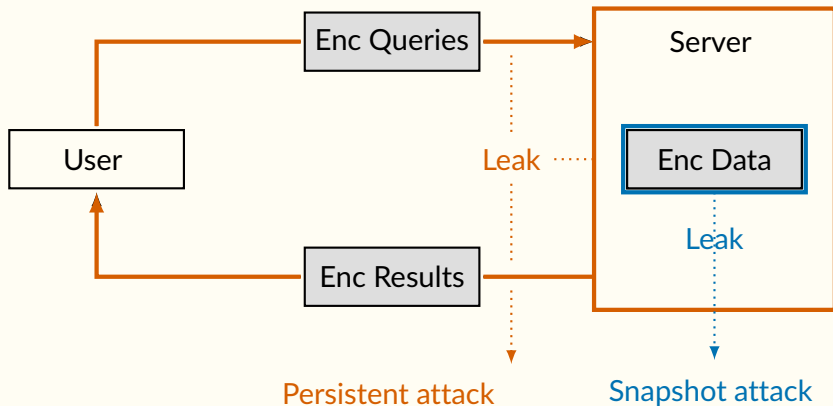
Snapshot attack



# Data Outsourcing to Untrusted Party



# Data Outsourcing to Untrusted Party



Customer		
Name	Pay	Nation
Alice	VISA	US
Bob	PayPal	US
Bob	PayPal	CAN

Supplier	
Name	Nation
Intel	US
IBM	US
RIM	CAN
Arca	MEX
Intel	MEX

# “Ideal” Encrypted Tables

- Setup leaks: table dimension

?	?	?
?	?	?
?	?	?

?	?
?	?
?	?
?	?
?	?

Legend:



Unleaked



Equality



Projected

# “Ideal” Encrypted Tables

- Encrypted Selection:  $\tilde{\sigma}$

?	?	?
?	?	?
?	?	?

?	?
?	?
?	?
?	?
?	?

Legend:



Unleaked



Equality



Projected

# “Ideal” Encrypted Tables

- Encrypted Projection:  $\tilde{\pi}$

?	?	?
?	?	?
?	?	?

?	?
?	?
?	?
?	?
?	?

Legend:



Unleaked



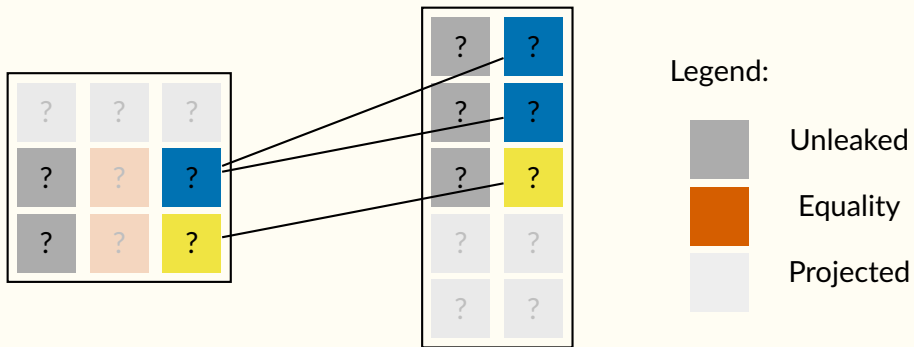
Equality



Projected

# "Ideal" Encrypted Tables

- Encrypted Join: ⋈



# “Ideal” Encrypted Tables

- Encrypted Join: ⚡

?	?	?	?
?	?	?	?
?	?	?	?

Legend:



Unleaked



Equality



Projected



# "Ideal" Encrypted Tables

- Query leaks: nothing outside of result; patterns within result

?	?	?
?	?	?
?	?	?

?	?
?	?
?	?
?	?
?	?

Legend:



Unleaked



Equality

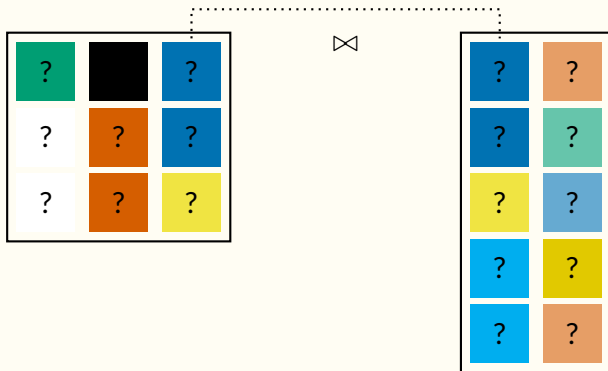


Projected

# Existing Approaches

## PPE-based schemes<sup>1</sup>

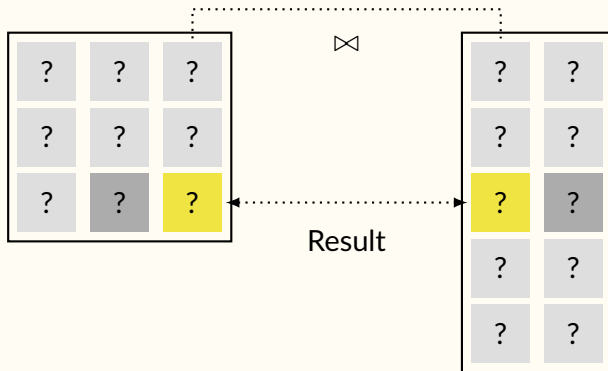
Use property-preserving encryption to encrypt every cell.



<sup>1</sup>Quantization [HILM02], CryptDB [PRZB11], Monomi [TKMZ13], Cipherbase [ABEKKRV13], SAP SEED, MS SQL Server Always Encrypted.

## PPE-based schemes<sup>1</sup>

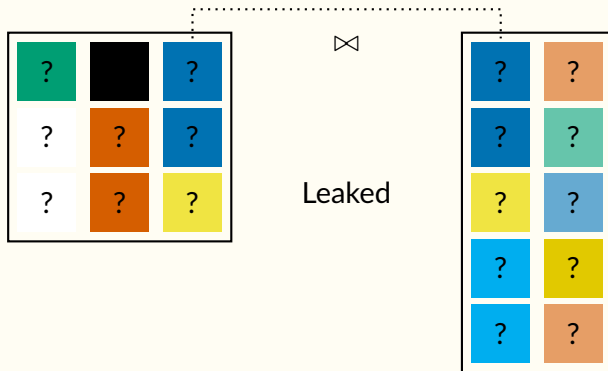
Use property-preserving encryption to encrypt every cell.



<sup>1</sup>Quantization [HILM02], CryptDB [PRZB11], Monomi [TKMZ13], Cipherbase [ABEKKRV13], SAP SEED, MS SQL Server Always Encrypted.

## PPE-based schemes<sup>1</sup>

Use property-preserving encryption to encrypt every cell.

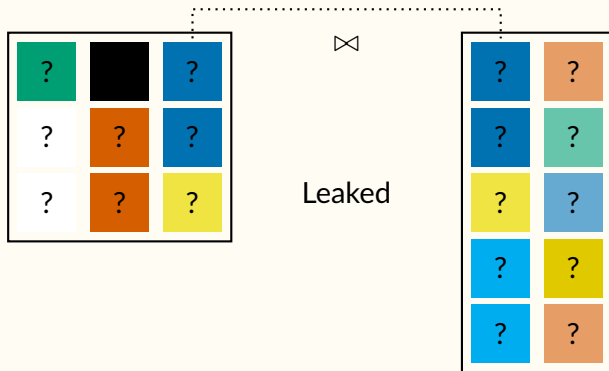


<sup>1</sup>Quantization [HILM02], CryptDB [PRZB11], Monomi [TKMZ13], Cipherbase [ABEKKRV13], SAP SEED, MS SQL Server Always Encrypted.

## PPE-based schemes<sup>1</sup>

Use property-preserving encryption to encrypt every cell.

- Data-recovery attacks [NKW15,DDC16,GSBNR17].

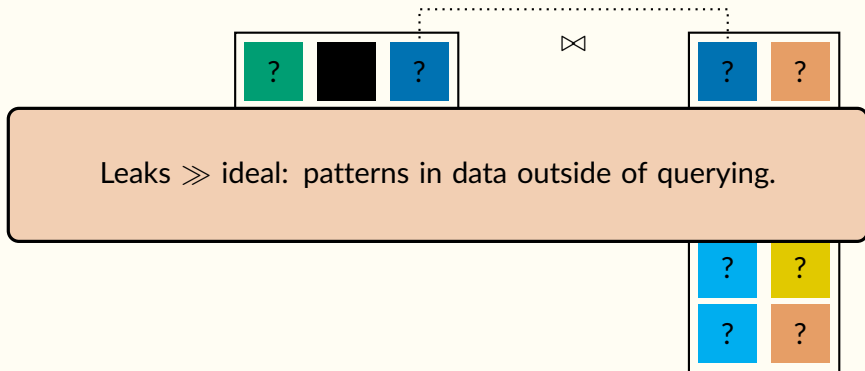


<sup>1</sup>Quantization [HILM02], CryptDB [PRZB11], Monomi [TKMZ13], Cipherbase [ABEKKRV13], SAP SEED, MS SQL Server Always Encrypted.

## PPE-based schemes<sup>1</sup>

Use property-preserving encryption to encrypt every cell.

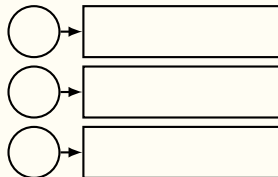
- Data-recovery attacks [NKW15,DDC16,GSBNR17].



<sup>1</sup>Quantization [HILM02], CryptDB [PRZB11], Monomi [TKMZ13], Cipherbase [ABEKKRV13], SAP SEED, MS SQL Server Always Encrypted.

## Encrypted Multimaps<sup>2</sup>

Use multimaps to represent tables and operators.



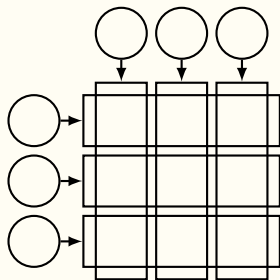
---

<sup>2</sup>SPX [KM18], OPX [KMZZ20,ZKMZ21], Pibas [CJJJKRS14].



## Encrypted Multimaps<sup>2</sup>

Use multimaps to represent tables and operators.

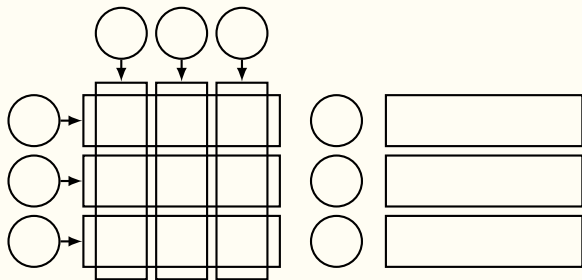


---

<sup>2</sup>SPX [KM18], OPX [KMZZ20,ZKMZ21], Pibas [CJJJKRS14].

## Encrypted Multimaps<sup>2</sup>

Use multimaps to represent tables and operators.

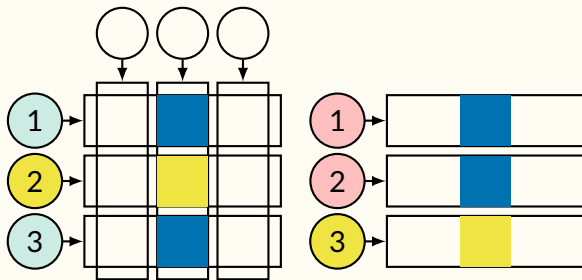


---

<sup>2</sup>SPX [KM18], OPX [KMZZ20,ZKMZ21], Pibas [CJJJKRS14].

## Encrypted Multimaps<sup>2</sup>

Use multimaps to represent tables and operators.

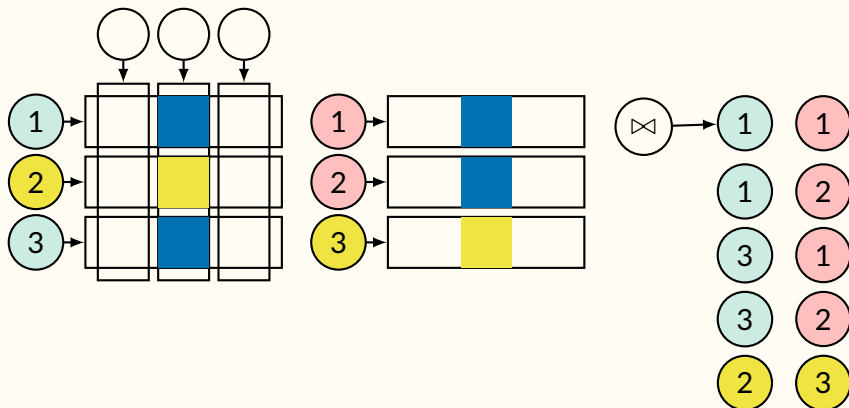


---

<sup>2</sup>SPX [KM18], OPX [KMZZ20,ZKMZ21], Pibas [CJJJKRS14].

## Encrypted Multimaps<sup>2</sup>

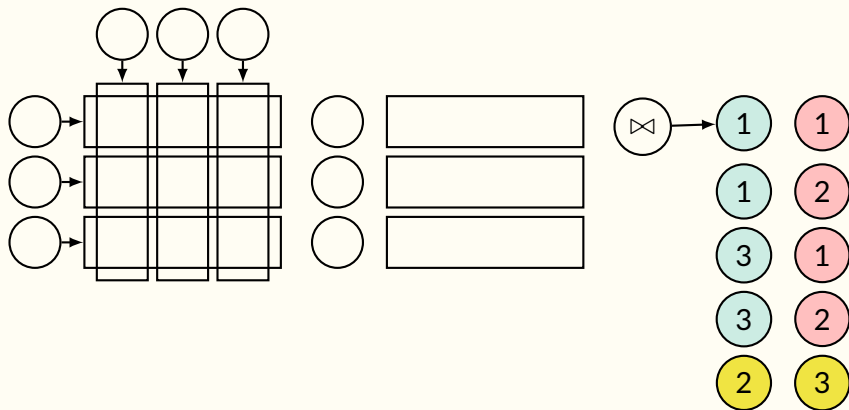
Use multimaps to represent tables and operators.



<sup>2</sup>SPX [KM18], OPX [KMZZ20,ZKMZ21], Pibas [CJJJKRS14].

## Encrypted Multimaps<sup>2</sup>

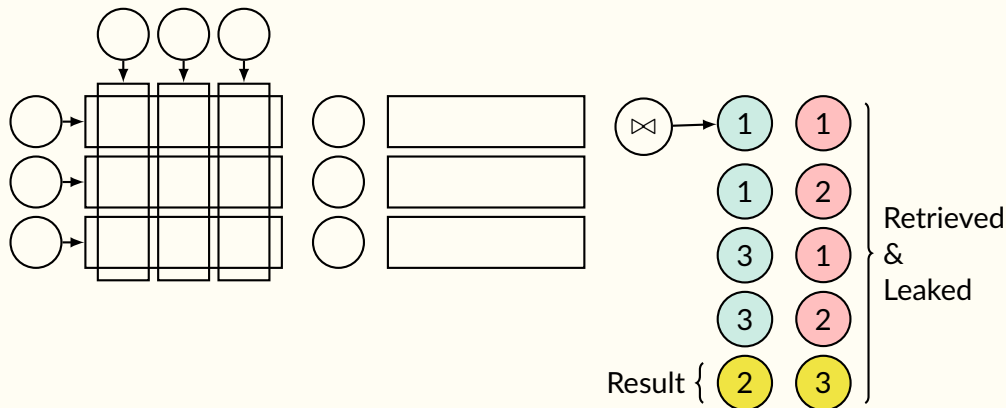
Use multimaps to represent tables and operators.



<sup>2</sup>SPX [KM18], OPX [KMZZ20,ZKMZ21], Pibas [CJJJKRS14].

## Encrypted Multimaps<sup>2</sup>

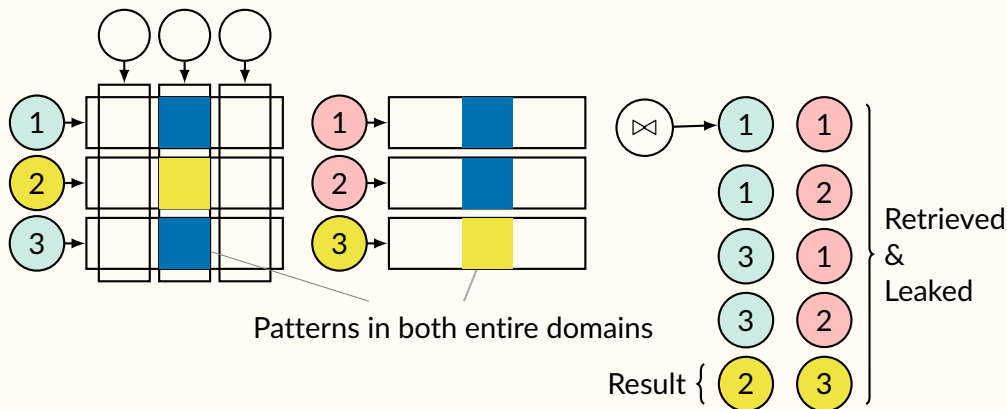
Use multimaps to represent tables and operators.



<sup>2</sup>SPX [KM18], OPX [KMZZ20,ZKMZ21], Pibas [CJJJKRS14].

## Encrypted Multimaps<sup>2</sup>

Use multimaps to represent tables and operators.



<sup>2</sup>SPX [KM18], OPX [KMZZ20,ZKMZ21], Pibas [CJJJKRS14].

## Encrypted Multimaps<sup>2</sup>

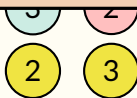
Use multimaps to represent tables and operators.

- Quadratic cost:
  - $\mathcal{O}(T^2)$  time/space for join.
  - Precompute all joins; “look up” join results.
- Leaks  $>$  ideal: full joint pattern for filtered join.

Retrieved  
&  
Leaked

Patterns in both entire domains

Result {



<sup>2</sup>SPX [KM18], OPX [KMZZ20,ZKMZ21], Pibas [CJJJKRS14].



# Encrypted Multimaps<sup>2</sup>

Use multimaps to represent tables and operators.

Quadratic cost:

- “Abstraction cost”
  - Data duplication.
  - Lack of access locality.
- No algebra: no query optimization; complicated scheme

trieved

aked

---

<sup>2</sup>SPX [KM18], OPX [KMZZ20,ZKMZ21], Pibas [CJJJKRS14].

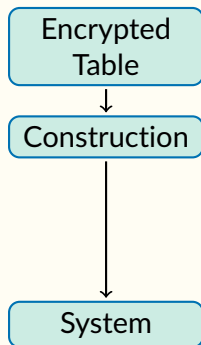
# Our Approach

# Our Approach

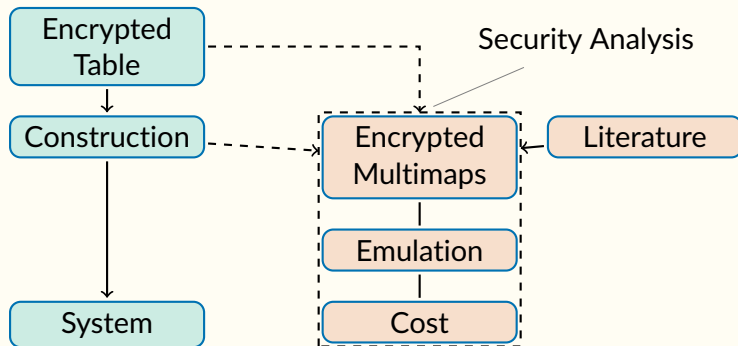
## Encrypted Table

- Linear cost:
  - $\mathcal{O}(T)$  time/space.
  - $\mathcal{O}(T)$  precomputation.
- Relational algebra: query optimization; composition.
- Reduce leakage for filtered joins.

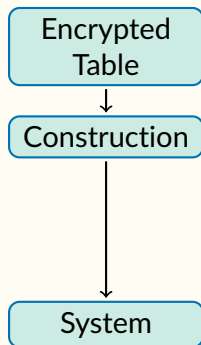
# Our Approach



# Our Approach



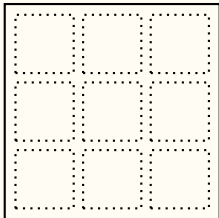
# Our Approach



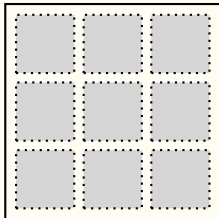
# Encrypted Table

# Encrypted Table

T

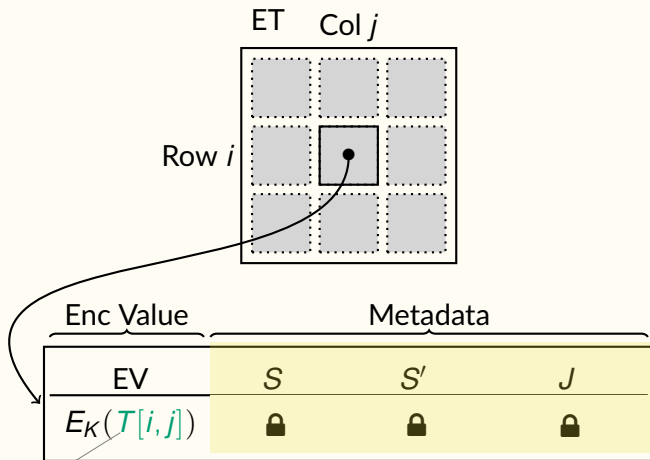


ET





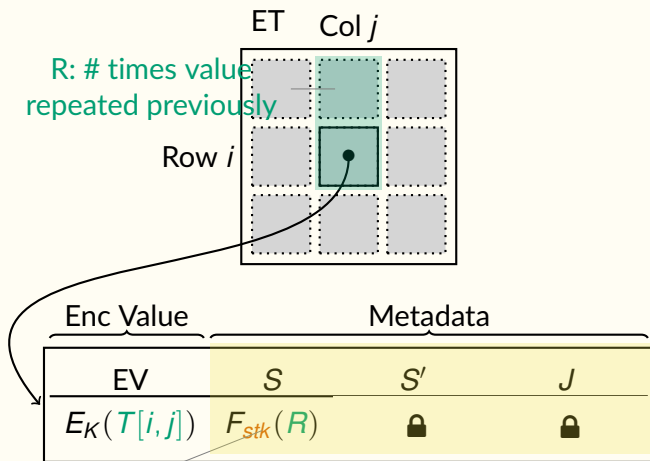
# Encrypted Table



Secret key  $K$

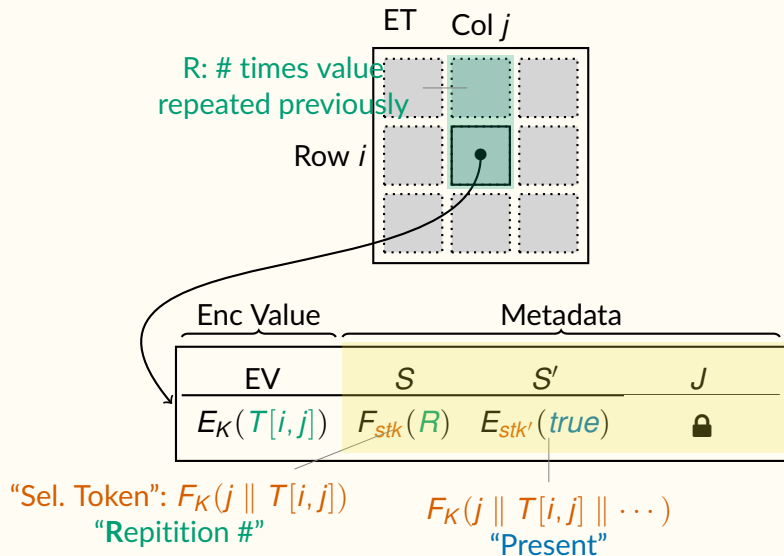
Plaintext cell

# Encrypted Table

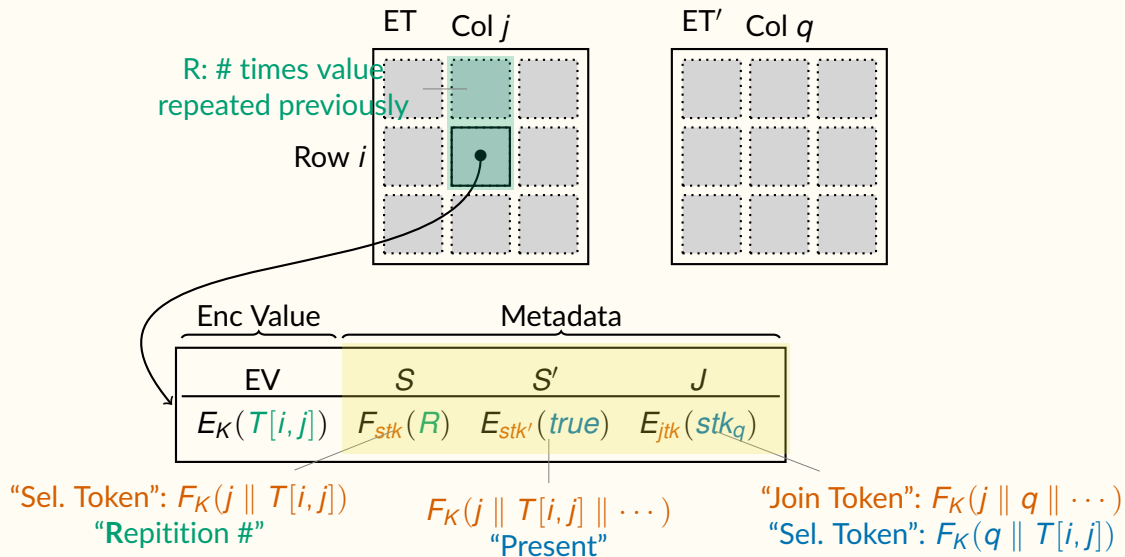


"Sel. Token":  $F_K(j \parallel T[i, j])$   
"Repetition #"

# Encrypted Table



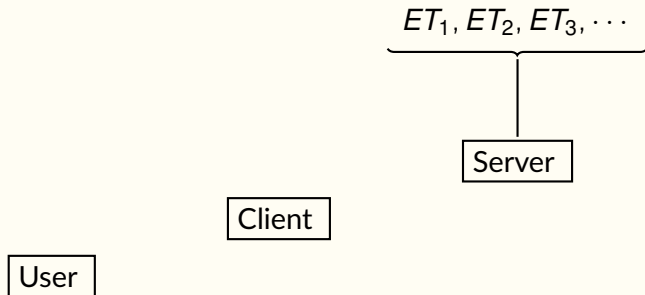
# Encrypted Table



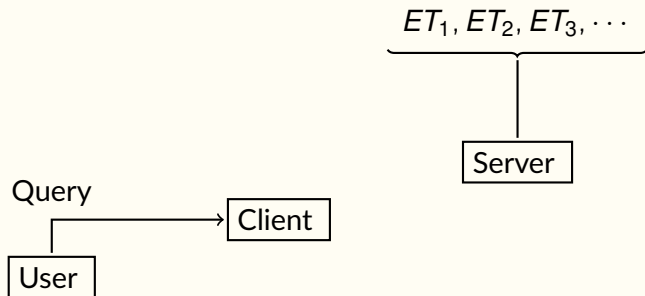
# Encrypted Table

$$ET_1, ET_2, ET_3, \dots$$

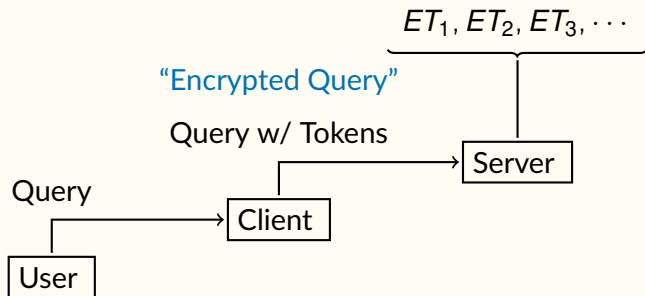
# Encrypted Table



# Encrypted Table

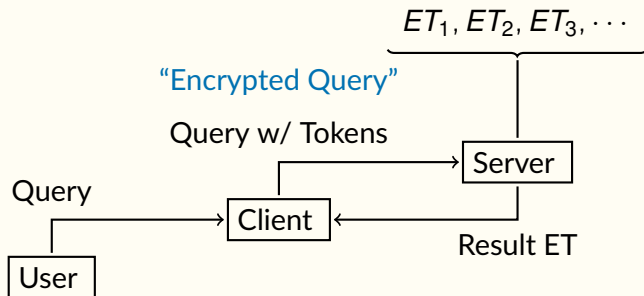


# Encrypted Table

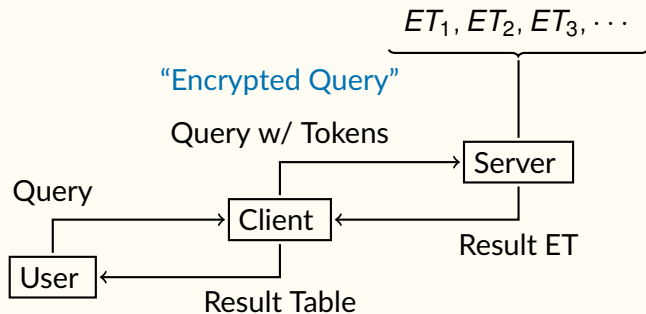




# Encrypted Table

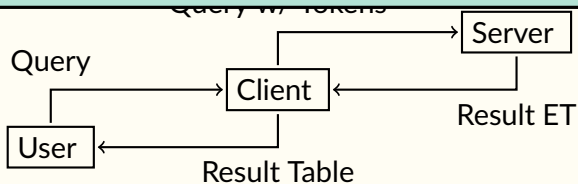


# Encrypted Table



# Encrypted Table

- Encrypted value + metadata per cell
- Linear cost



# Encrypted Selection

# Encrypted Selection

Name	Pay	Nation
Alice	VISA	US
Bob	PayPal	US
Bob	VISA	CAN

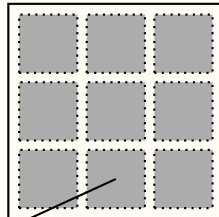
# Encrypted Selection

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

Subscripted rep #

# Encrypted Selection

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

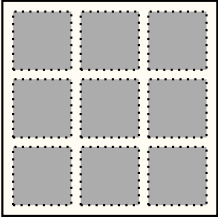
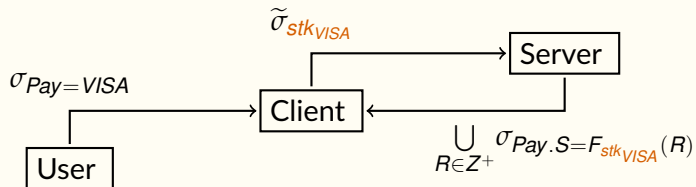


EV	$S$	$S'$	...
$E_K(VISA)$	$F_{stk}(2)$	$E_{stk}(true)$	...

$$F_K(Pay \parallel VISA)$$

# Encrypted Selection

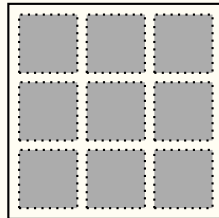
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



# Encrypted Selection

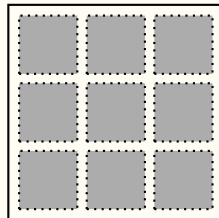
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



$\tilde{\sigma}_{stk_{VISA}}$

# Encrypted Selection

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



$$\tilde{\sigma}_{stk_{VISA}} \leftarrow F_K(\text{Pay} \parallel \boxed{\text{VISA}})$$

# Encrypted Selection

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

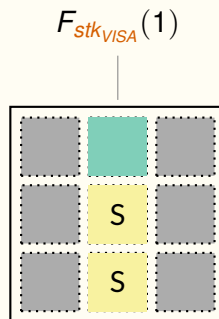
$F_{stk}(R)$

	S	
	S	
	S	

$$\tilde{\sigma}_{stk_{VISA}} \longleftarrow F_K(Pay \parallel \boxed{VISA})$$

# Encrypted Selection

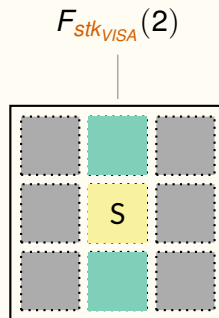
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



$$\tilde{\sigma}_{stk_{VISA}} \longleftarrow F_K(Pay \parallel \boxed{VISA})$$

# Encrypted Selection

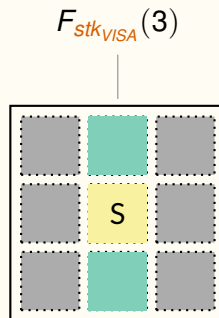
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



$$\tilde{\sigma}_{stk_{VISA}} \leftarrow F_K(Pay \parallel \boxed{VISA})$$

# Encrypted Selection

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

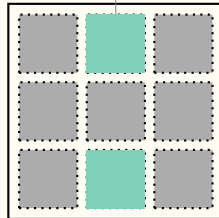


$$\tilde{\sigma}_{stk_{VISA}} \longleftarrow F_K(Pay \parallel \boxed{VISA})$$

# Encrypted Selection

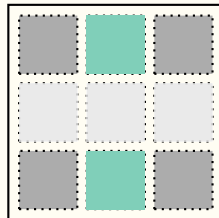
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

“Fixed point”



# Encrypted Selection

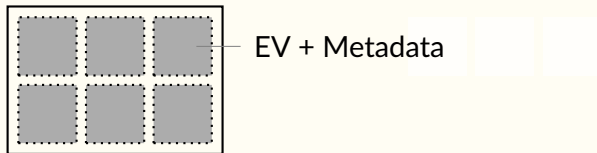
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>





# Encrypted Selection

Closure: Still an ET

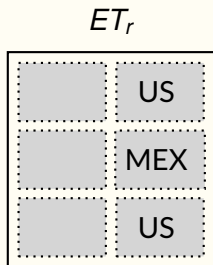
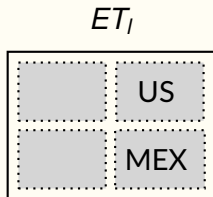


# Encrypted Selection

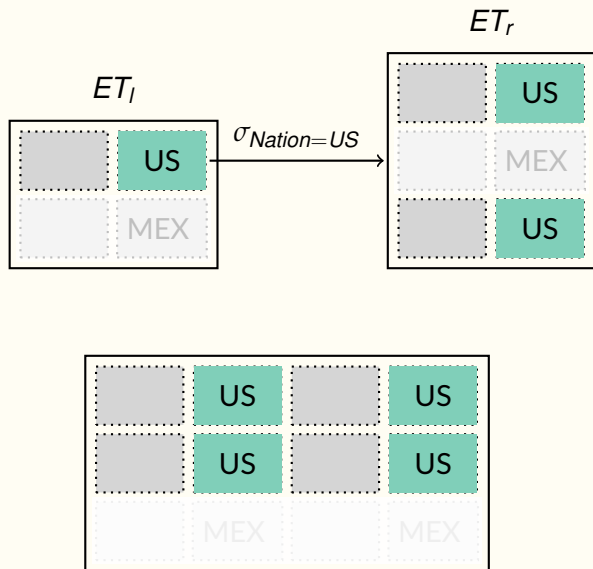
- Closure: good for algebra.
- Ideal leakage: pattern in selected cells.
- Linear cost.
- Extension to *Conjunction*.

# Encrypted Join

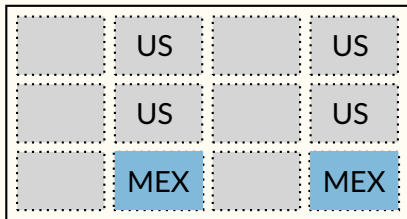
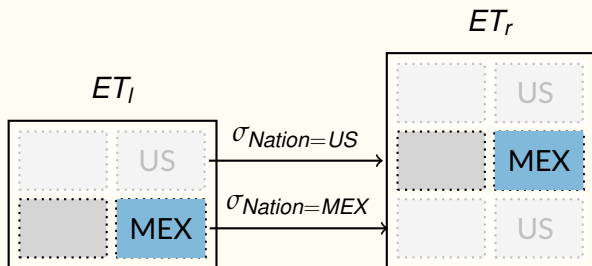
# Encrypted Join



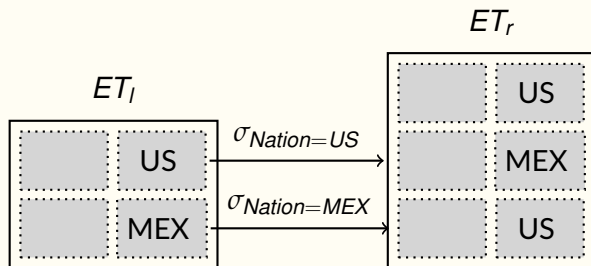
# Encrypted Join



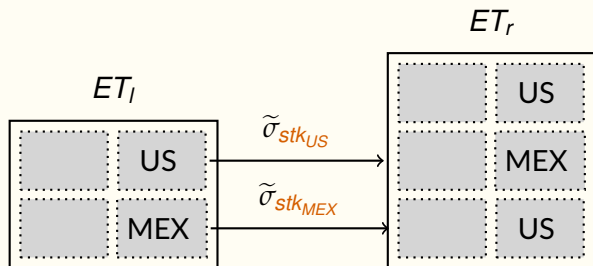
# Encrypted Join



# Encrypted Join

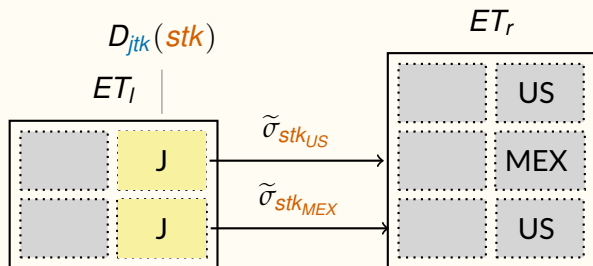


# Encrypted Join

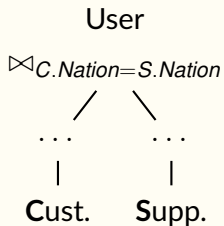
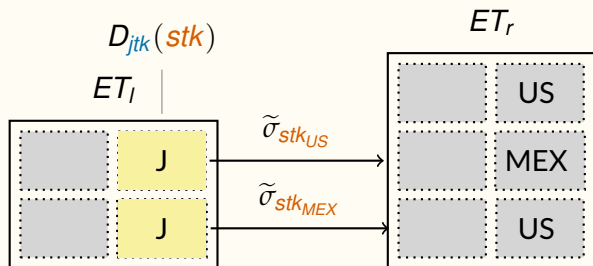




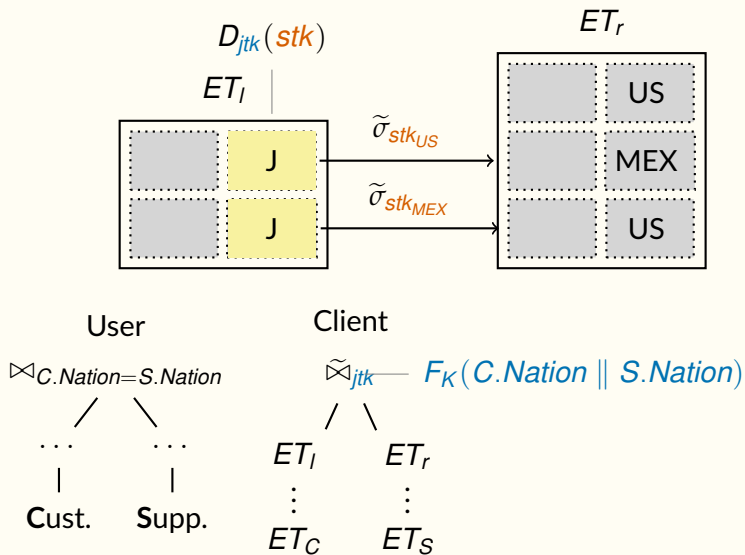
# Encrypted Join



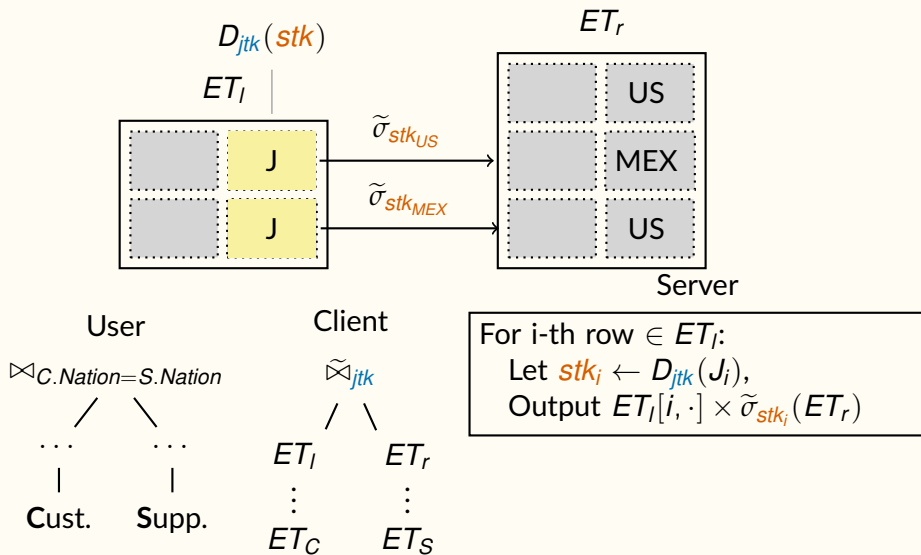
# Encrypted Join



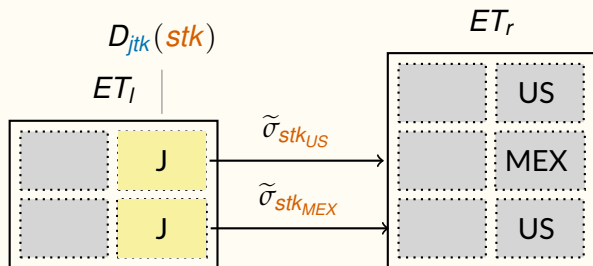
# Encrypted Join



# Encrypted Join



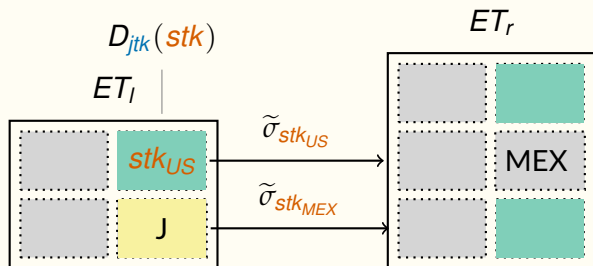
# Encrypted Join



Client

$$\begin{array}{c}
 \boxtimes_{jtk} \text{ --- } F_K(C.Nation \parallel S.Nation) \\
 / \quad \backslash \\
 ET_I \quad ET_r \\
 \vdots \quad \vdots \\
 ET_C \quad ET_S
 \end{array}$$

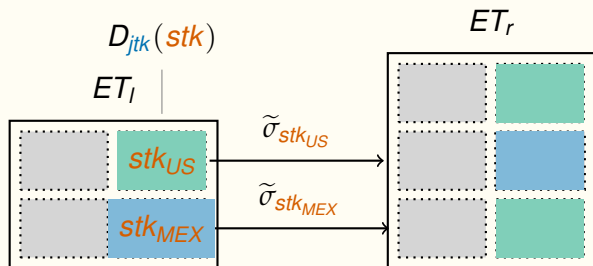
# Encrypted Join



Client

$$\begin{array}{c}
 \boxtimes_{jtk} \text{ --- } F_K(C.Nation \parallel S.Nation) \\
 / \quad \backslash \\
 ET_I \quad ET_r \\
 \vdots \quad \vdots \\
 ET_C \quad ET_S
 \end{array}$$

# Encrypted Join

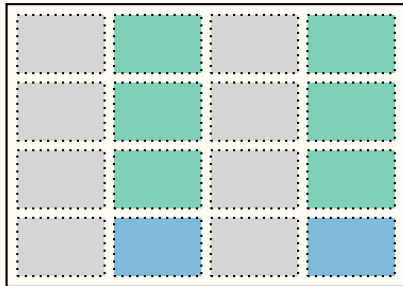


Client

$$\begin{array}{c}
 \boxtimes_{jtk} \text{ --- } F_K(C.Nation \parallel S.Nation) \\
 / \quad \backslash \\
 ET_I \quad ET_r \\
 \vdots \quad \vdots \\
 ET_C \quad ET_S
 \end{array}$$

## Encrypted Join

Closure: Still an ET



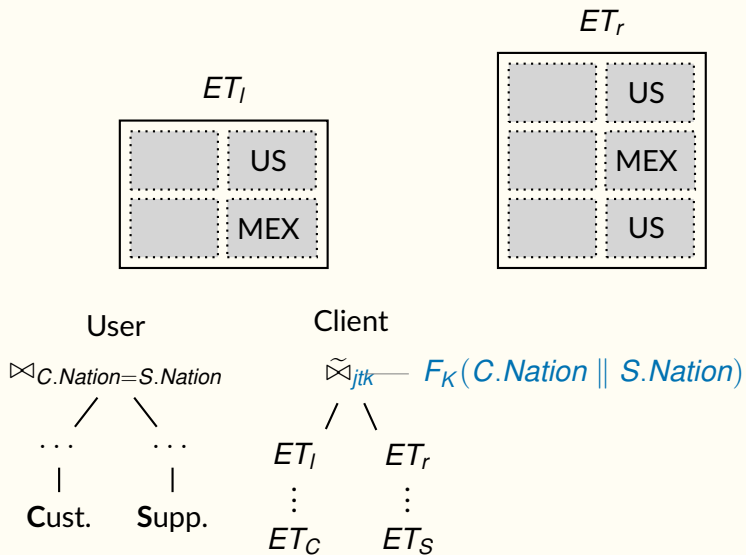


## Encrypted Join

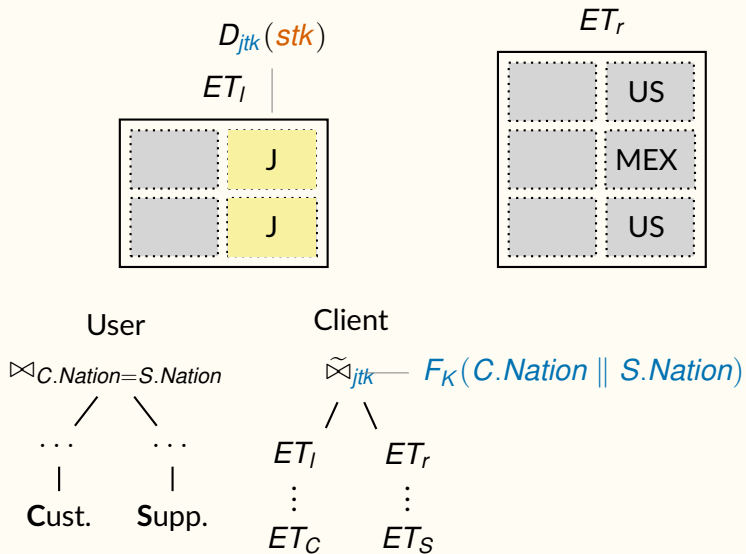
- Encrypted join as encrypted selections.
- Linear cost.

# Reducing Leakage

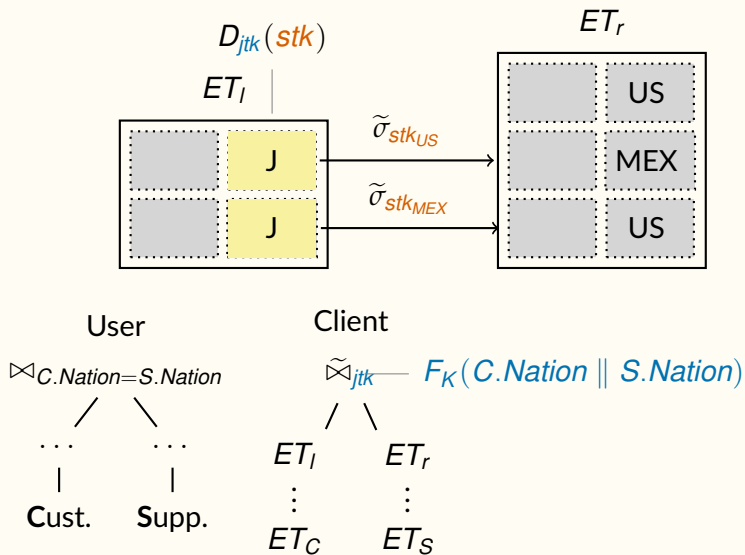
# Reducing Leakage



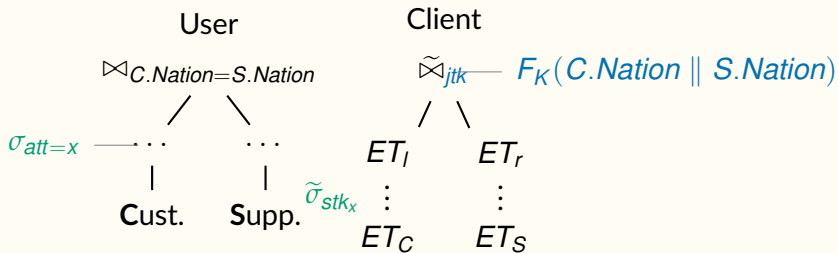
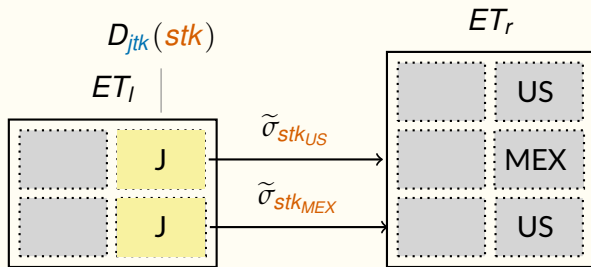
# Reducing Leakage



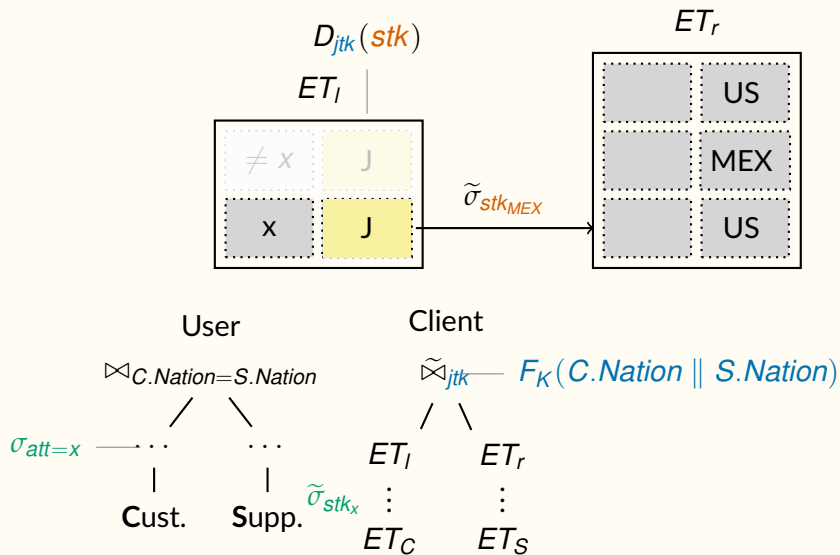
# Reducing Leakage



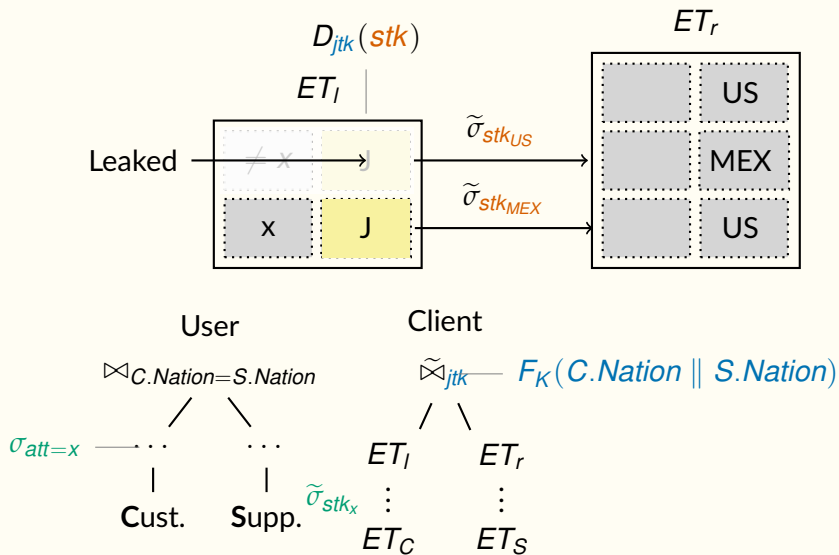
# Reducing Leakage



# Reducing Leakage

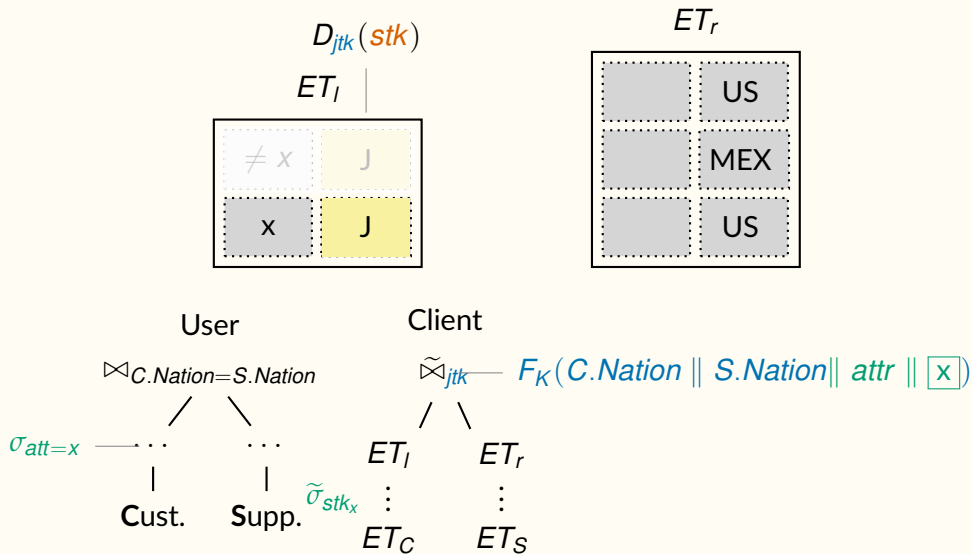


# Reducing Leakage

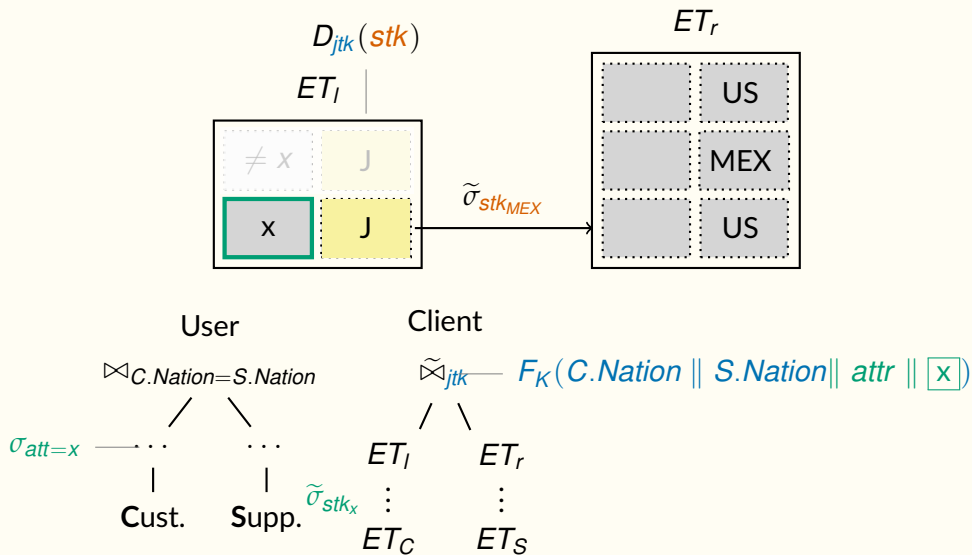




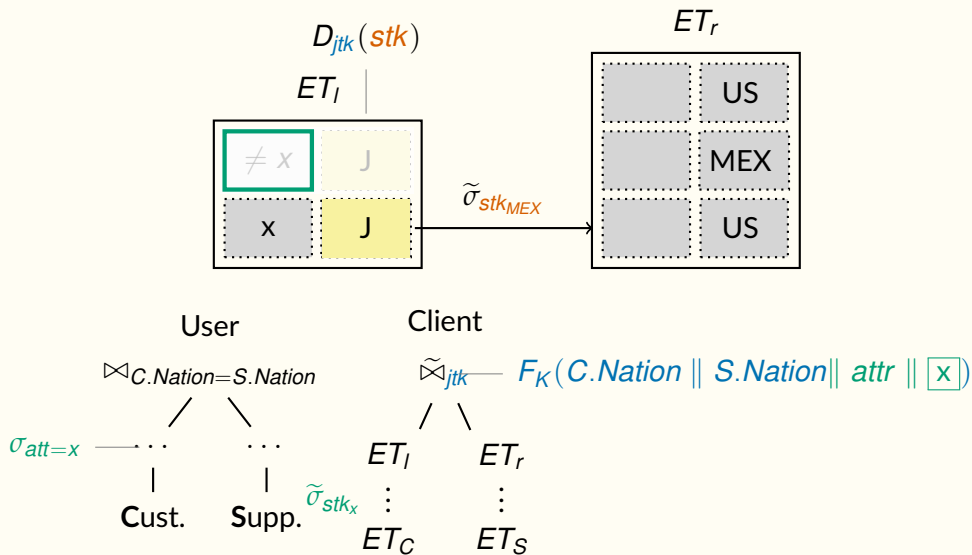
# Reducing Leakage



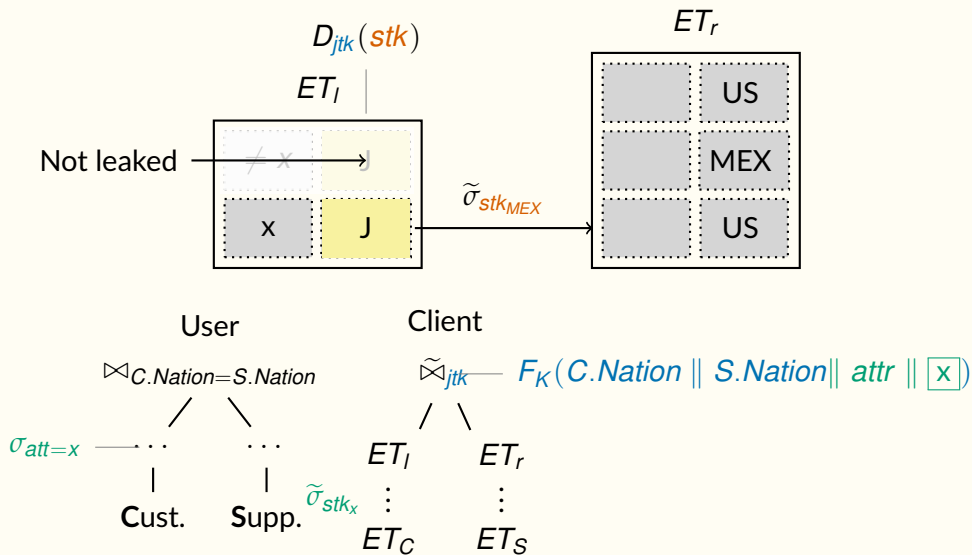
# Reducing Leakage



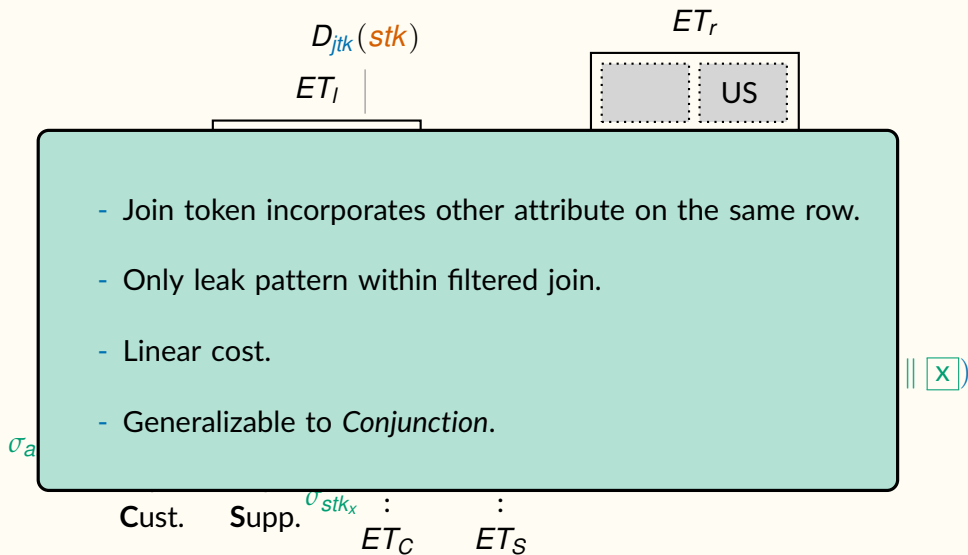
# Reducing Leakage



# Reducing Leakage



# Reducing Leakage



# Query Optimization

# Query Composition

- Encrypted {Selections, Joins, Projections} compose for Conjunctive Queries
  - Anti-Join, Semi-Join, Subqueries, Disjunctions.
- Algebra: ordering  $\implies$  query optimization.
  - Selection/Projection pushdown ( $19.8\times$  speedup on TPC-H)
  - Join/Selection reordering ( $12.6\times$  speedup o TPC-H)

# Fixed-Point Operator

- Recursion: common in Encrypted Selection and Encrypted Join

$$\bigcup_{R \in Z^+} \sigma_{S=F_{stk}(R)} ET$$



# Fixed-Point Operator

- Recursion: common in Encrypted Selection and Encrypted Join

$$\bigcup_{R \in Z^+} \sigma_{S=F_{stk}(R)} ET$$

- Fixed-point Operator: extension beyond relational algebra
  - Not in all database systems: SparkSQL
  - Postgres 8+/MySQL 8+/SQL Server 2005+: Recursive Common Table Expression
  - Oracle 11g Release 2: Recursive Subquery Factoring / CONNECT BY

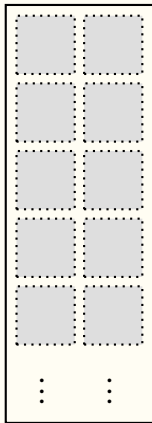
# Fixed-Point Operator

- Recursion: common in Encrypted Selection and Encrypted Join

$$\bigcup_{R \in Z^+} \sigma_{S=F_{stk}(R)} ET$$

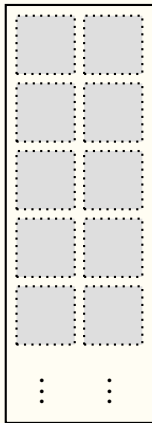
- Fixed-point Operator: extension beyond relational algebra
  - Not in all database systems: SparkSQL
  - Postgres 8+/MySQL 8+/SQL Server 2005+: Recursive Common Table Expression
  - Oracle 11g Release 2: Recursive Subquery Factoring / CONNECT BY
- Semantics & Optimization

# Fixed-Point Operator



# Fixed-Point Operator

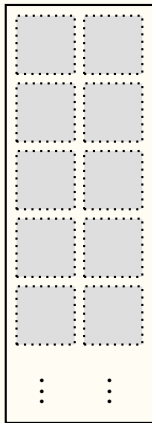
$\tilde{\sigma}_{stk_x}$



# Fixed-Point Operator

$$\sigma_S = F_{stk}(R)$$

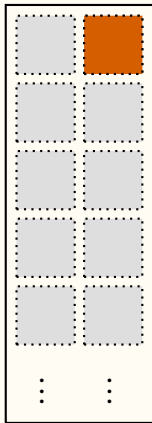
$\tilde{\sigma}_{stk_x}$



# Fixed-Point Operator

$$\sigma_{S=F_{stk}(R)}$$

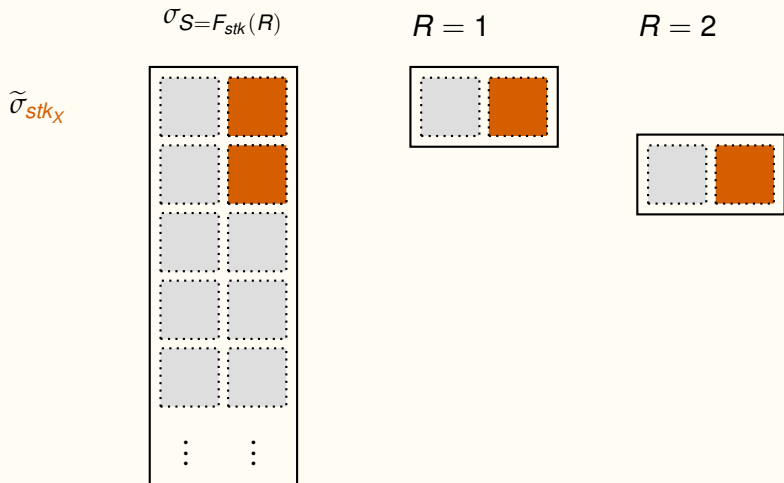
$\tilde{\sigma}_{stk_x}$



$$R = 1$$



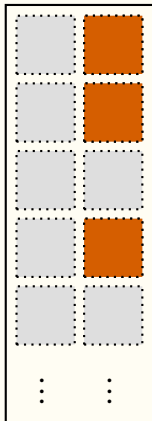
# Fixed-Point Operator



# Fixed-Point Operator

$\tilde{\sigma}_{stk_x}$

$$\sigma_{S=F_{stk}(R)}$$



$$R = 1$$



$$R = 2$$

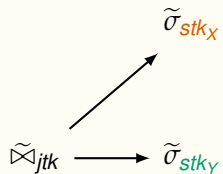


$$R = 3$$

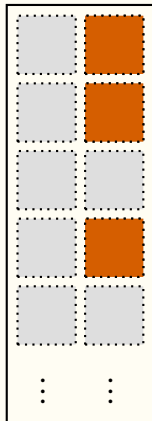




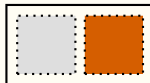
# Fixed-Point Operator



$$\sigma_{S=F_{stk}(R)}$$



$$R = 1$$



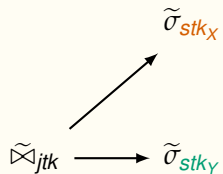
$$R = 2$$



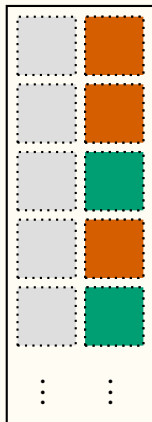
$$R = 3$$



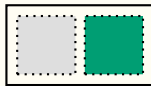
# Fixed-Point Operator



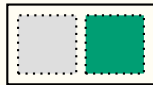
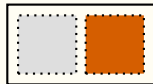
$$\sigma_{S=F_{stk}(R)}$$



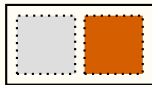
$$R = 1$$



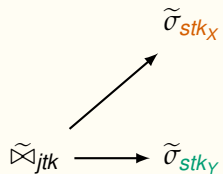
$$R = 2$$



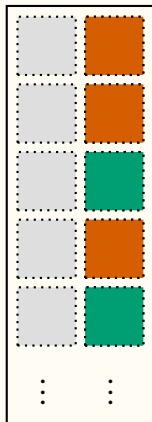
$$R = 3$$



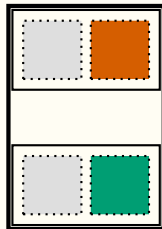
# Fixed-Point Operator



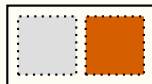
$$\sigma_{S=F_{stk}(R)}$$



$$R = 1$$



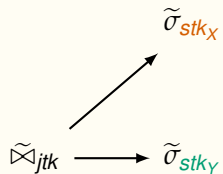
$$R = 2$$



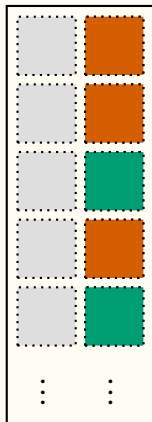
$$R = 3$$



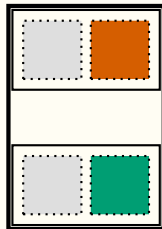
# Fixed-Point Operator



$$\sigma_{S=F_{stk}(R)}$$



$$R = 1$$



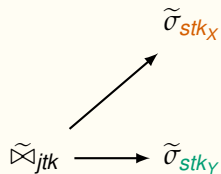
$$R = 2$$



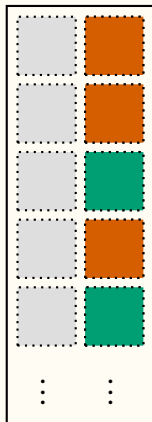
$$R = 3$$



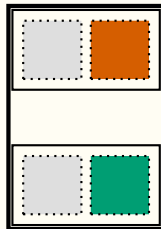
# Fixed-Point Operator



$$\sigma_{S=F_{stk}(R)}$$



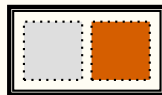
$$R = 1$$



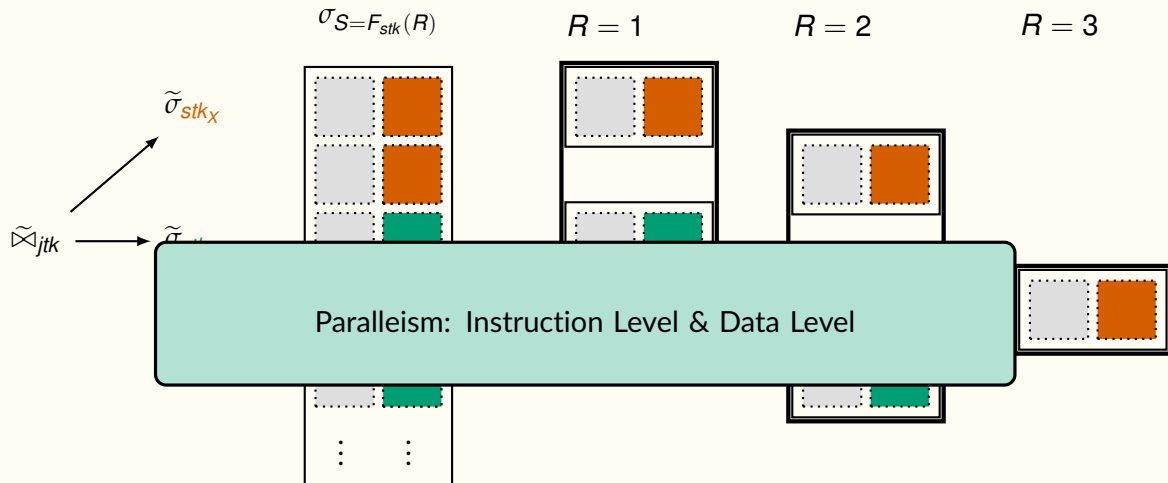
$$R = 2$$



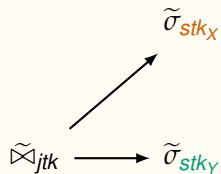
$$R = 3$$



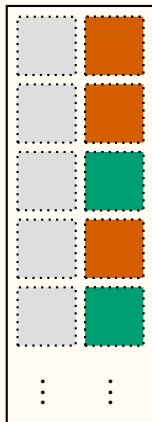
# Fixed-Point Operator



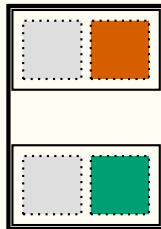
# Fixed-Point Operator



$$\sigma_{S=F_{stk}(R)}$$



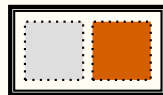
$$R = 1$$



$$R = 2$$



$$R = 3$$



# Fixed-Point Operator

$$\sigma_{S=F_{stk}(R)}$$

$$R = 1$$

$$R = 2$$

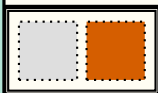
$$R = 3$$



$\bowtie_{jtk}$

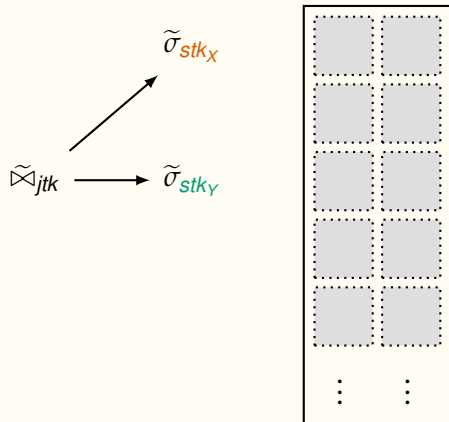
Minimize recursion depth:

- For increasing parallelism
- Depth=1: foreign key (FK) to primary key (PK) join
- Depth large: PK to FK join
- Heuristic: join from large to small table (Join Direction Rule) ( $12.6\times$  speedup on TPC-H)

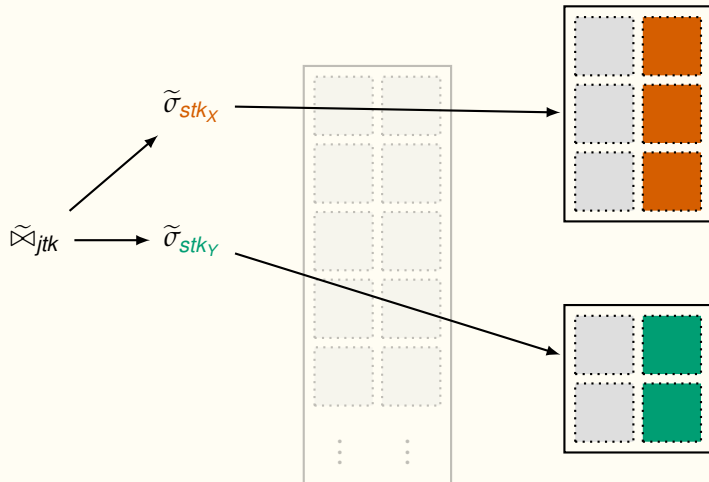




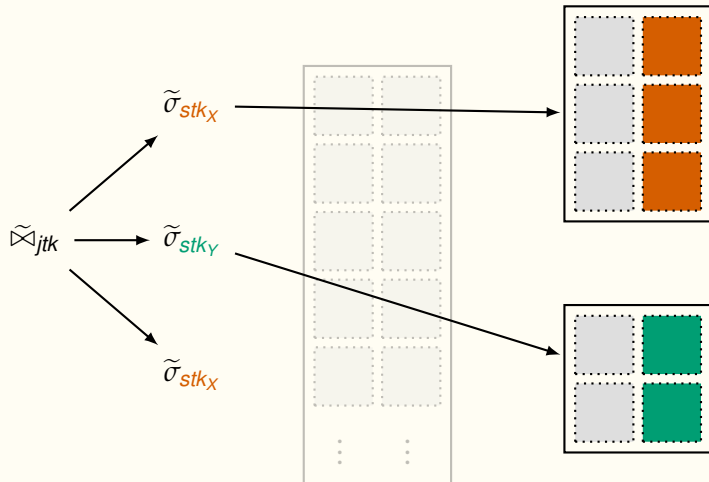
# Fixed-Point Operator



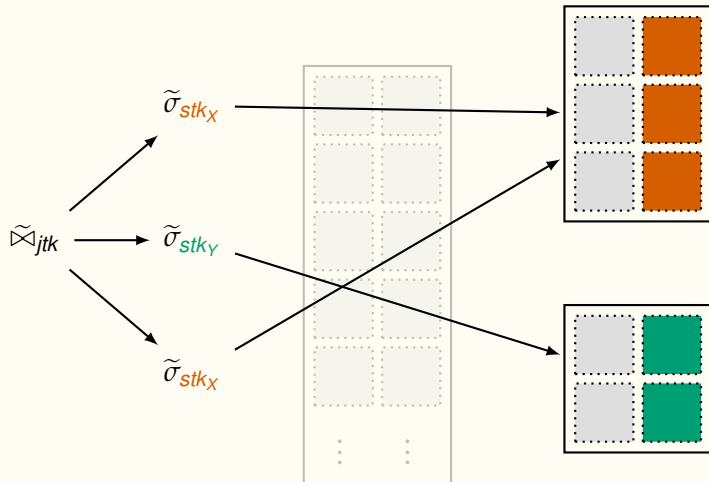
# Fixed-Point Operator



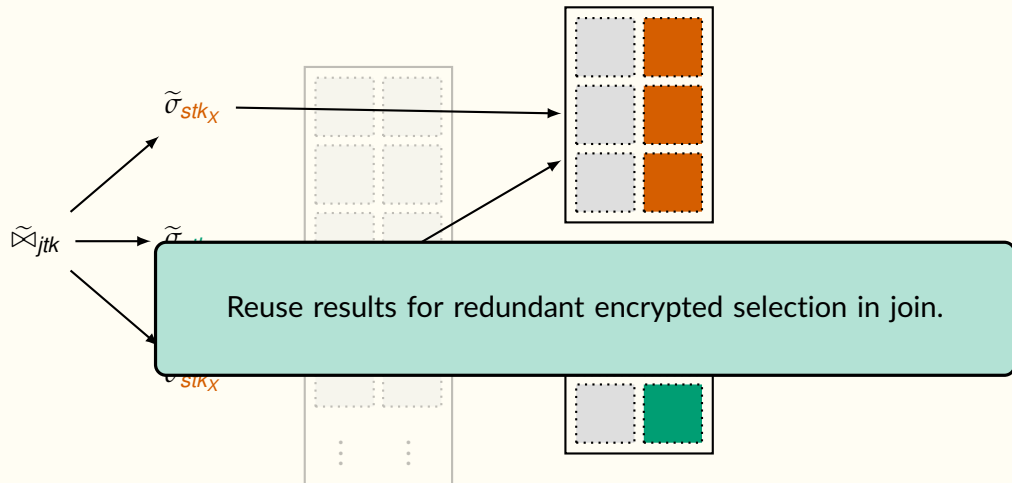
# Fixed-Point Operator



# Fixed-Point Operator

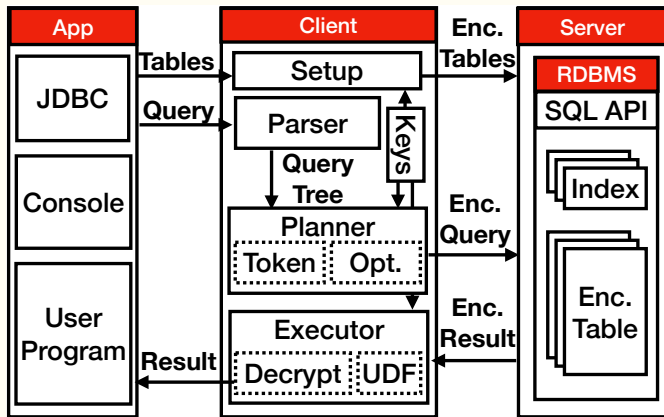


# Fixed-Point Operator



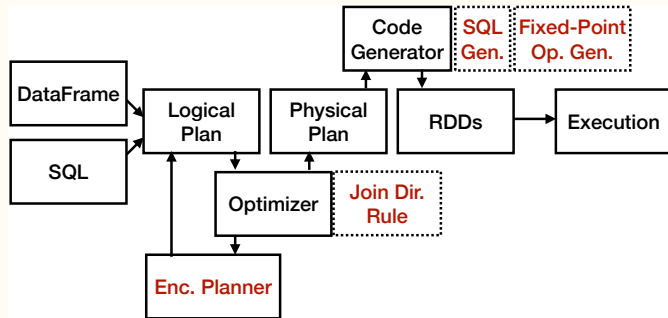
# System

# Legacy-Compliant Architecture



- Any SQL database backend
- Encryption transparent to Applications
- Leverage DB optimizations

# SparkSQL-based Implementation



- Query Tree partition into Unenc. & Enc.
- Rewrite with Enc. Operators
- Fixed-Point SQL generation
- Custom optimization rule



# Benchmark

# Overview

- PPE-based Schemes: CryptDB [PRZB11], Monomi [TKMZ13]

# Overview

- PPE-based Schemes: CryptDB [PRZB11], Monomi [TKMZ13]
- STE-based Schemes, EMM-based: SPX [KM18], OPX [KMZZ20,ZKMZ21];

# Overview

- PPE-based Schemes: CryptDB [PRZB11], Monomi [TKMZ13]
- STE-based Schemes, EMM-based: SPX [KM18], OPX [KMZZ20,ZKMZ21];
- Encrypted Table (ET) Scheme

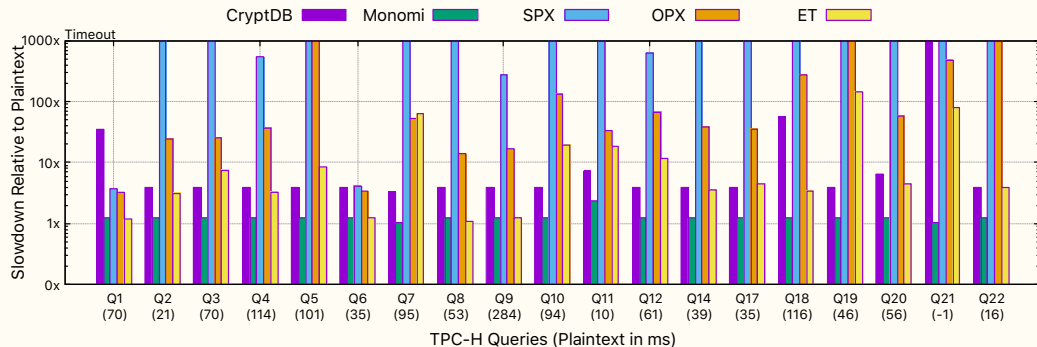
# Overview

- PPE-based Schemes: CryptDB [PRZB11], Monomi [TKMZ13]
- STE-based Schemes, EMM-based: SPX [KM18], OPX [KMZZ20,ZKMZ21];
- Encrypted Table (ET) Scheme
- TPC-H scale factor 10 ( 17GB);

# Overview

- PPE-based Schemes: CryptDB [PRZB11], Monomi [TKMZ13]
- STE-based Schemes, EMM-based: SPX [KM18], OPX [KMZZ20,ZKMZ21];
- Encrypted Table (ET) Scheme
- TPC-H scale factor 10 ( 17GB);
- 32GB RAM, 8 CPUs, 5.2TB Storage for SPX and OPX; 1.2TB Storage for CryptDB, Monomi and ET;

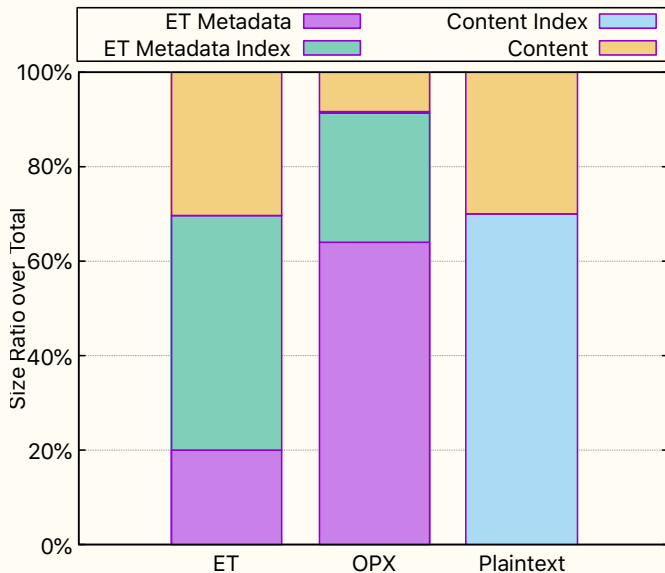
# Query Time



- ET achieves comparable query overhead ( $4\times$ ) than PPE-based CryptDB but for stronger security.
- ET is two orders magnitude better than STE-based precursors.

# Storage

System	Size
Plaintext	17.1GB
CryptDB	4.21×
Monomi	1.72×
SPX	252.22×
OPX	13.17×
ET	3.63×





# Summary

- STE-based Encrypted Table
  - Linear cost
  - Preserves relational algebra
  - Reduced leakage for conjunction
- Legacy compliant system based on SparkSQL and interace with any SQL DB.
  - Fixed-point operator optimization
- Efficiency comparable to PPE-based CryptDB but with stronger security ( $4\times$  storage and query overhead)

# Appendix

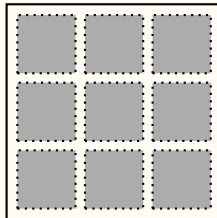
## Table vs. Multimap

Data Structure	Table	Multimap
Model	Relational (SQL)	Key-Value (NoSQL)
Language	Relational Algebra	Retrieval by Key
Optimality	$\mathcal{O}(T)$	$\mathcal{O}(Q)$
Basis for EDB	PKFK	SPX,OPX

# Encrypted Conjunction

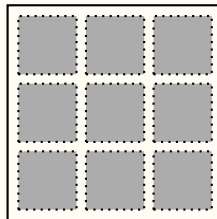
# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

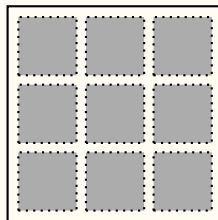


Conjunction

$$\sigma_{Play = VISA \wedge Name = Bob}$$

# Encrypted Conjunction

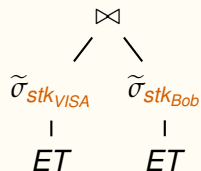
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



Conjunction

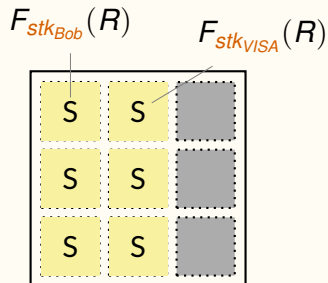
$$\sigma_{Play = VISA \wedge Name = Bob}$$

Approach 1



# Encrypted Conjunction

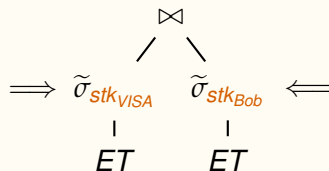
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



Approach 1

Conjunction

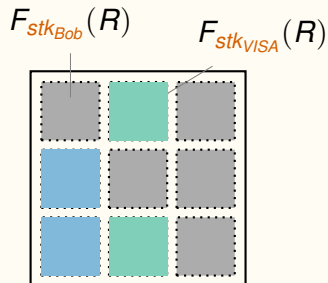
$$\sigma_{Play=VISA \wedge Name=Bob}$$





# Encrypted Conjunction

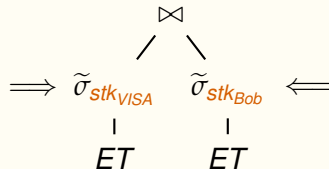
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



Approach 1

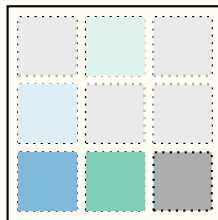
Conjunction

$$\sigma_{Play=VISA \wedge Name=Bob}$$



# Encrypted Conjunction

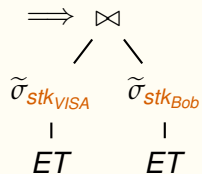
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



Conjunction

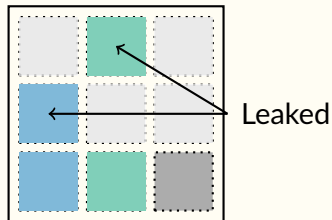
$$\sigma_{Play = VISA \wedge Name = Bob}$$

Approach 1



# Encrypted Conjunction

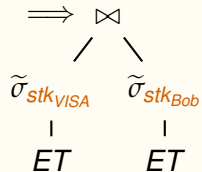
Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



Conjunction

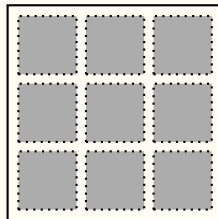
$$\sigma_{Play=VISA \wedge Name=Bob}$$

Approach 1



# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



Approach 2

$$\begin{array}{c} \tilde{\sigma}_{stk'_{Bob}} \\ | \\ \tilde{\sigma}_{stk_{VISA}} \\ | \\ ET \end{array}$$

# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

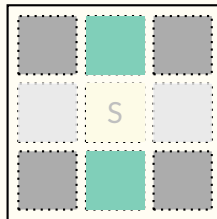
	S	
	S	
	S	

Approach 2

$$\begin{array}{c} \tilde{\sigma}_{stk'_{Bob}} \\ | \\ \Rightarrow \tilde{\sigma}_{stk_{VISA}} \\ | \\ ET \end{array}$$

# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

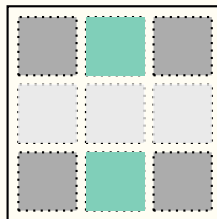


Approach 2

$$\begin{array}{c} \tilde{\sigma}_{stk'_{Bob}} \\ | \\ \Rightarrow \tilde{\sigma}_{stk_{VISA}} \\ | \\ ET \end{array}$$

# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



Approach 2

$$\begin{aligned}
 &\Rightarrow \tilde{\sigma}_{stk'_{Bob}} \text{ --- } F_K(\text{Name} \parallel \boxed{\text{Bob}} \parallel \text{Pay} \parallel \boxed{\text{VISA}}) \\
 &\quad | \\
 &\quad \tilde{\sigma}_{stk_{VISA}} \\
 &\quad | \\
 &\quad ET
 \end{aligned}$$

# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

$E_{stk}(true)$

$S'$		
$S'$		

Approach 2

$$\begin{aligned}
 &\Rightarrow \tilde{\sigma}_{stk'_{Bob}} \text{ --- } F_K(\text{Name} \parallel \boxed{\text{Bob}} \parallel \text{Pay} \parallel \boxed{\text{VISA}}) \\
 &\quad | \\
 &\quad \tilde{\sigma}_{stk_{VISA}} \\
 &\quad | \\
 &\quad ET
 \end{aligned}$$



# Encrypted Conjunction

$$D_{\text{stk}'_{\text{Bob}}}(S') = \text{true?}$$

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

S'		
S'		

Approach 2

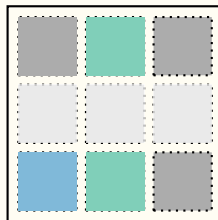
$$\begin{aligned}
 &\Rightarrow \tilde{\sigma}_{\text{stk}'_{\text{Bob}}} \text{ --- } F_K(\text{Name} \parallel \boxed{\text{Bob}} \parallel \text{Pay} \parallel \boxed{\text{VISA}}) \\
 &\quad | \\
 &\quad \tilde{\sigma}_{\text{stk}_{\text{VISA}}} \\
 &\quad | \\
 &\quad ET
 \end{aligned}$$



# Encrypted Conjunction

$$D_{\text{stk}'_{\text{Bob}}}(S') = \text{true?}$$

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



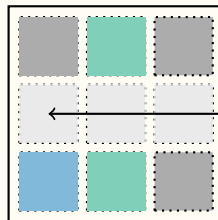
Approach 2

$$\begin{aligned}
 &\Rightarrow \tilde{\sigma}_{\text{stk}'_{\text{Bob}}} \text{ --- } F_K(\text{Name} \parallel \boxed{\text{Bob}} \parallel \text{Pay} \parallel \boxed{\text{VISA}}) \\
 &\quad \quad \quad | \\
 &\quad \quad \quad \tilde{\sigma}_{\text{stk}_{\text{VISA}}} \\
 &\quad \quad \quad | \\
 &\quad \quad \quad ET
 \end{aligned}$$

# Encrypted Conjunction

$$D_{stk'_{Bob}}(S') = true?$$

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



Not leaked

Approach 2

$$\begin{aligned} &\Rightarrow \tilde{\sigma}_{stk'_{Bob}} \text{ --- } F_K(\text{Name} \parallel \boxed{\text{Bob}} \parallel \text{Pay} \parallel \boxed{\text{VISA}}) \\ &\quad \quad \quad | \\ &\quad \quad \quad \tilde{\sigma}_{stk_{VISA}} \\ &\quad \quad \quad | \\ &\quad \quad \quad ET \end{aligned}$$

# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

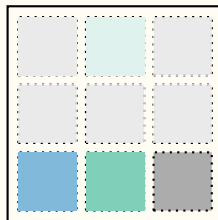
Only check on smaller ET


Approach 2

$$\begin{aligned}
 &\Rightarrow \tilde{\sigma}_{\text{stk}'_{\text{Bob}}} \text{ --- } F_K(\text{Name} \parallel \boxed{\text{Bob}} \parallel \text{Pay} \parallel \boxed{\text{VISA}}) \\
 &\quad | \\
 &\quad \tilde{\sigma}_{\text{stk}_{\text{VISA}}} \\
 &\quad | \\
 &\quad ET
 \end{aligned}$$

# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>

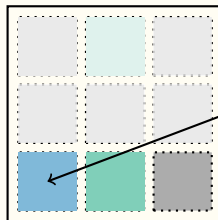


Approach 2

$$\begin{aligned}
 &\Rightarrow \tilde{\sigma}_{\text{stk}'_{\text{Bob}}} \text{ --- } F_K(\text{Name} \parallel \boxed{\text{Bob}} \parallel \text{Pay} \parallel \boxed{\text{VISA}}) \\
 &\quad | \\
 &\quad \tilde{\sigma}_{\text{stk}_{\text{VISA}}} \\
 &\quad | \\
 &\quad ET
 \end{aligned}$$

# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



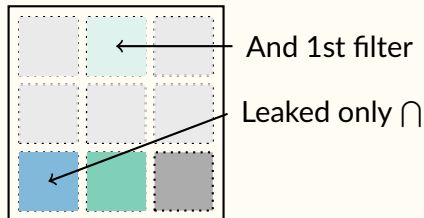
Leaked only  $\cap$

Approach 2

$$\begin{aligned}
 &\Rightarrow \tilde{\sigma}_{stk'_{Bob}} \text{ --- } F_K(\text{Name} \parallel \boxed{\text{Bob}} \parallel \text{Pay} \parallel \boxed{\text{VISA}}) \\
 &\quad | \\
 &\quad \tilde{\sigma}_{stk_{VISA}} \\
 &\quad | \\
 &\quad ET
 \end{aligned}$$

# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>
Bob <sub>1</sub>	PayPal <sub>1</sub>	US <sub>2</sub>
Bob <sub>2</sub>	VISA <sub>2</sub>	CAN <sub>1</sub>



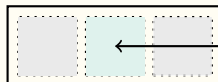
Approach 2

$$\begin{aligned}
 &\Rightarrow \tilde{\sigma}_{\text{stk}'_{\text{Bob}}} \text{ --- } F_K(\text{Name} \parallel \boxed{\text{Bob}} \parallel \text{Pay} \parallel \boxed{\text{VISA}}) \\
 &\quad \quad \quad | \\
 &\quad \quad \quad \tilde{\sigma}_{\text{stk}_{\text{VISA}}} \\
 &\quad \quad \quad | \\
 &\quad \quad \quad ET
 \end{aligned}$$



# Encrypted Conjunction

Name	Pay	Nation
Alice <sub>1</sub>	VISA <sub>1</sub>	US <sub>1</sub>



And 1st filter

ed only  $\cap$

- Composition  $\Rightarrow$  leak less
- Ordering  $\Rightarrow$  smaller intermediate ETs  $\Rightarrow$  faster

$$\Rightarrow \tilde{\sigma}_{stk'_{Bob}} \text{ --- } F_K(\text{Name} \parallel \boxed{\text{Bob}} \parallel \text{Pay} \parallel \boxed{\text{VISA}})$$

|

$$\tilde{\sigma}_{stk_{VISA}}$$

|

ET

# Security

# “Ideal” Encrypted Tables

- Setup leaks: table dimension

?	?	?
?	?	?
?	?	?

?	?
?	?
?	?
?	?
?	?

Legend:



Unleaked



Equality



Projected

# “Ideal” Encrypted Tables

- Encrypted Selection:  $\tilde{\sigma}$

?	?	?
?	?	?
?	?	?

?	?
?	?
?	?
?	?
?	?

Legend:



Unleaked



Equality



Projected

# “Ideal” Encrypted Tables

- Encrypted Projection:  $\tilde{\pi}$

?	?	?
?	?	?
?	?	?

?	?
?	?
?	?
?	?
?	?

Legend:



Unleaked



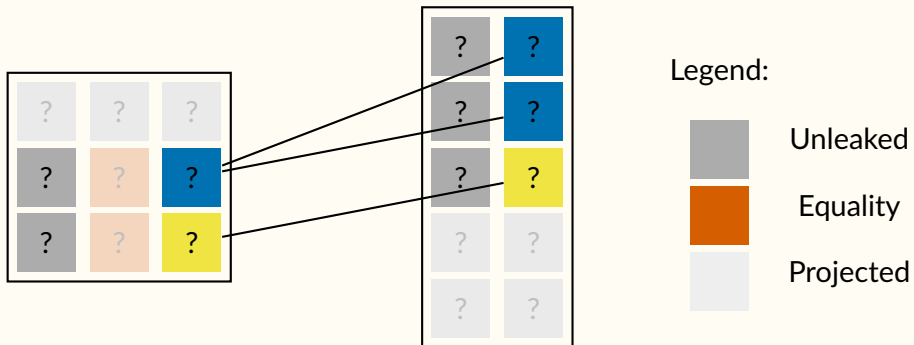
Equality



Projected

# "Ideal" Encrypted Tables

- Encrypted Join: ⋈



# "Ideal" Encrypted Tables

- Encrypted Join: ⋈

?	?	?	?
?	?	?	?
?	?	?	?

Legend:



Unleaked



Equality



Projected

# "Ideal" Encrypted Tables

- Query leaks: nothing outside of result; patterns within result

?	?	?
?	?	?
?	?	?

?	?
?	?
?	?
?	?
?	?

Legend:



Unleaked



Equality

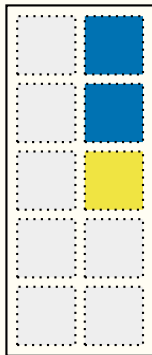
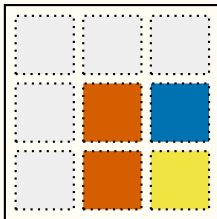


Projected



# Simulation

Show: leaks patterns in query result and nothing else.

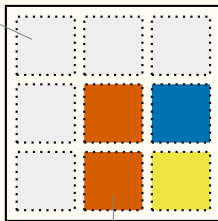


## Simulation

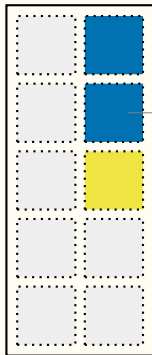
Show: leaks patterns in query result and nothing else.

- Non-adaptive security: swap non-queried cells with random noise

Random noise



Metadata S as defined



Metadata J as defined

# Simulation

Show: leaks patterns in query result and nothing else.

- Adaptive security in ROM:  $F_K(x) \doteq H(K \parallel x)$ ,  $E_K(m) \doteq (r, H(K \parallel r) \oplus m)$  for random  $K, r$ .

