Order-Revealing Encryption:
New Constructions, Applications and Lower Bounds

Kevin Lewi and David J. Wu

Stanford University

Searching on Encrypted Data

ars TECHNICA SIGN IN

RISK ASSESSMENT —

Yahoo says half a billion accounts breached by nation-sponsored hackers

One of the biggest compromises ever exposes names, e-mail addresses, and much more.

DAN GOODIN - 9/22/2016, 1:21 PM

Searching on Encrypted Data


SECTIONS HOME SEARCH The New York Times SUBSCRIBE LOG IN

BUSINESS DAY

Data Breach at Anthem May Forecast a Trend

By REED ABELSON and JULIE CRESWELL FEB. 6, 2015

Searching on Encrypted Data


EDITION: UNITED STATES

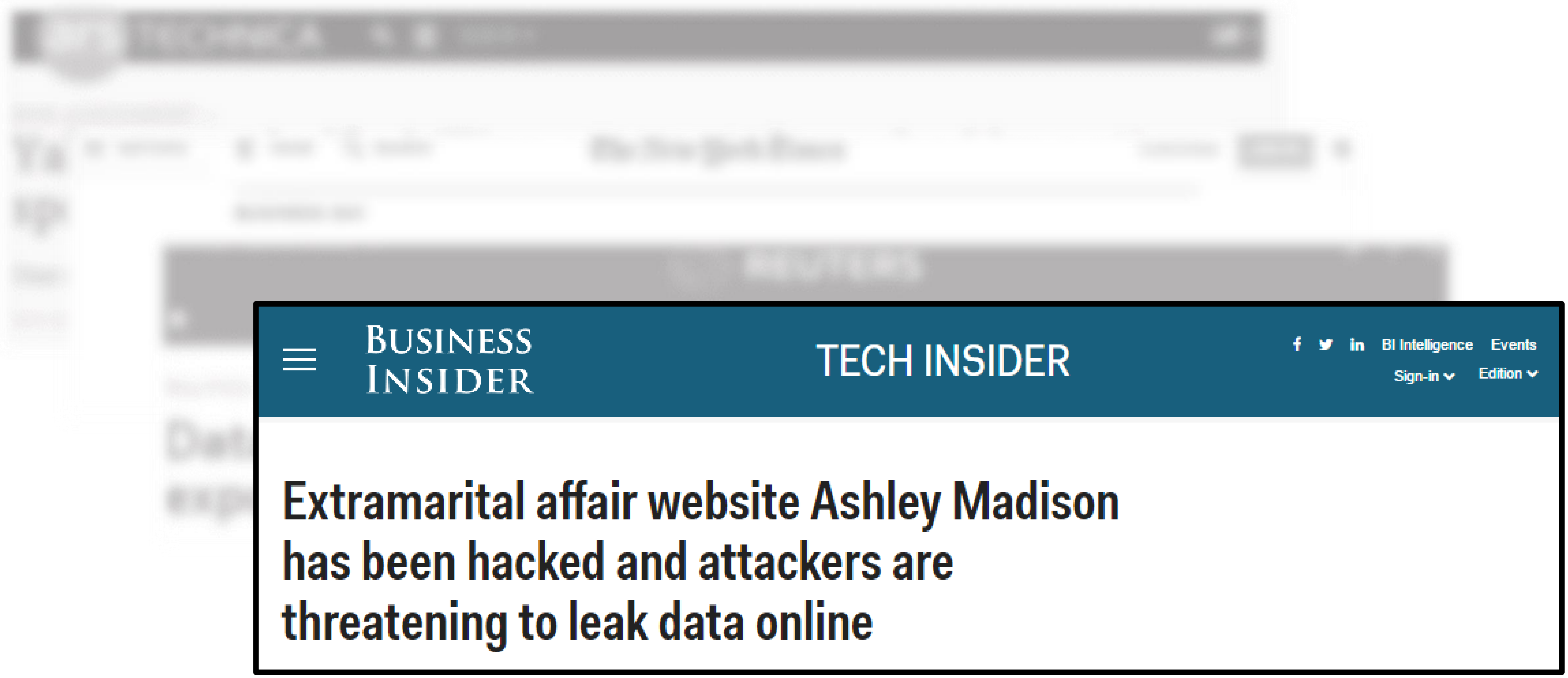
REUTERS

Business Markets World Politics Tech Commentary Breakingviews Money Life

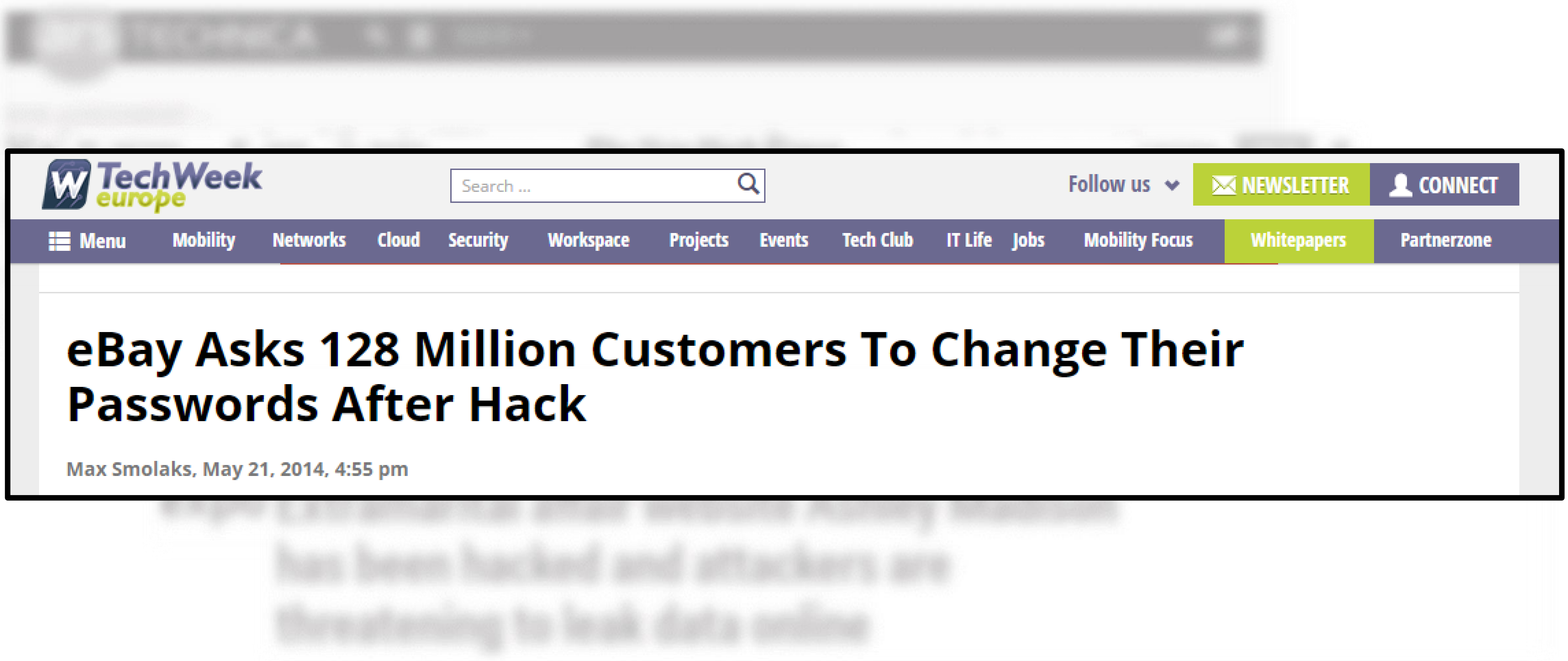
POLITICS | Mon Dec 28, 2015 | 4:52pm EST

Database of 191 million U.S. voters exposed on Internet: researcher

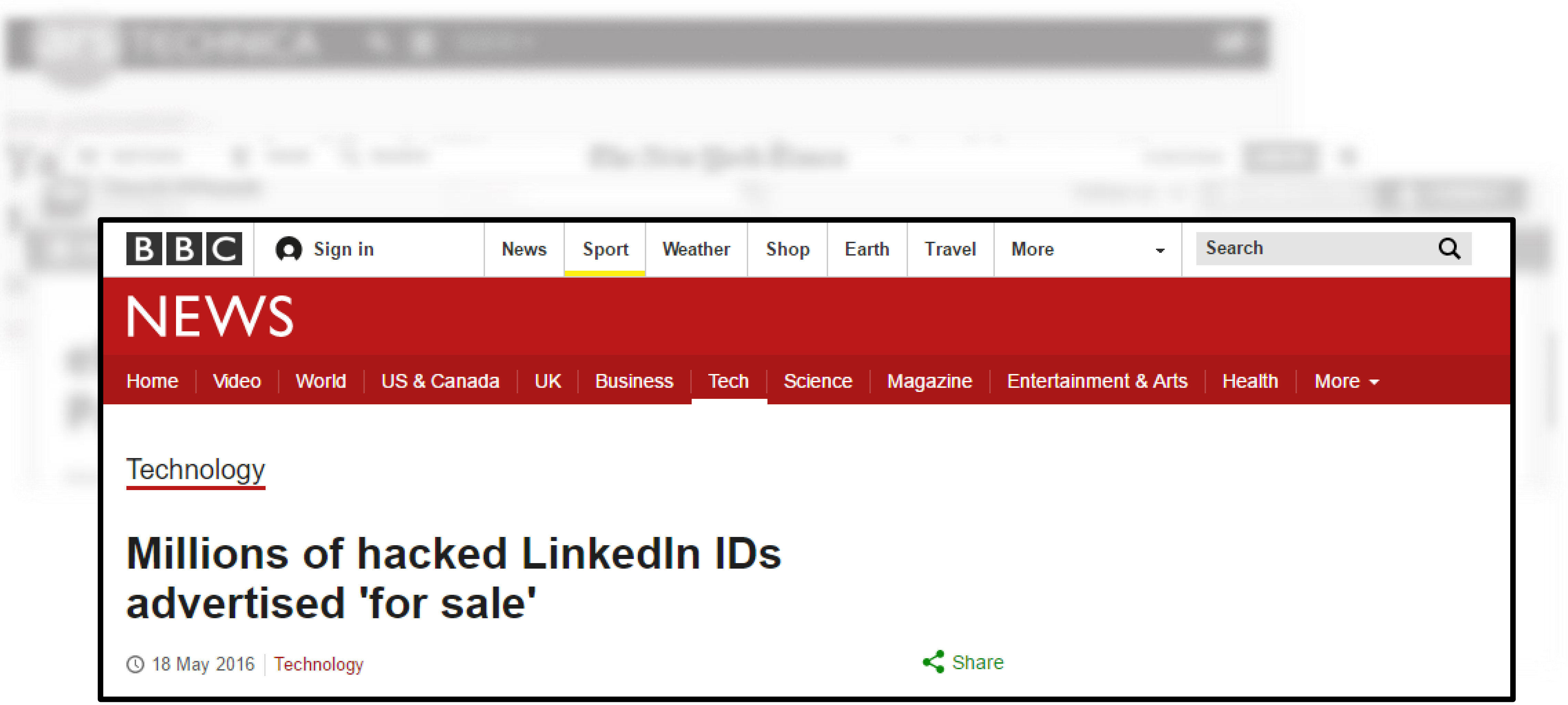
Searching on Encrypted Data



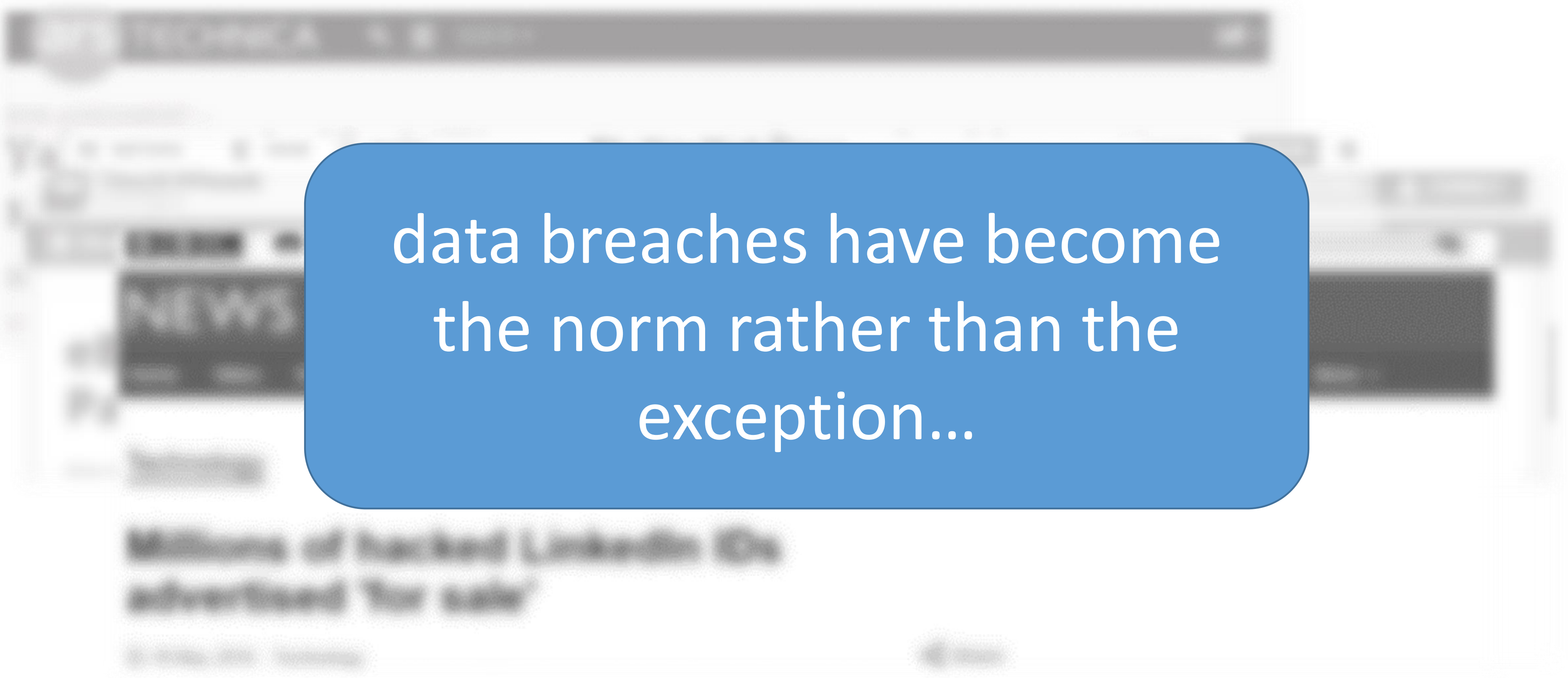
Searching on Encrypted Data



Searching on Encrypted Data

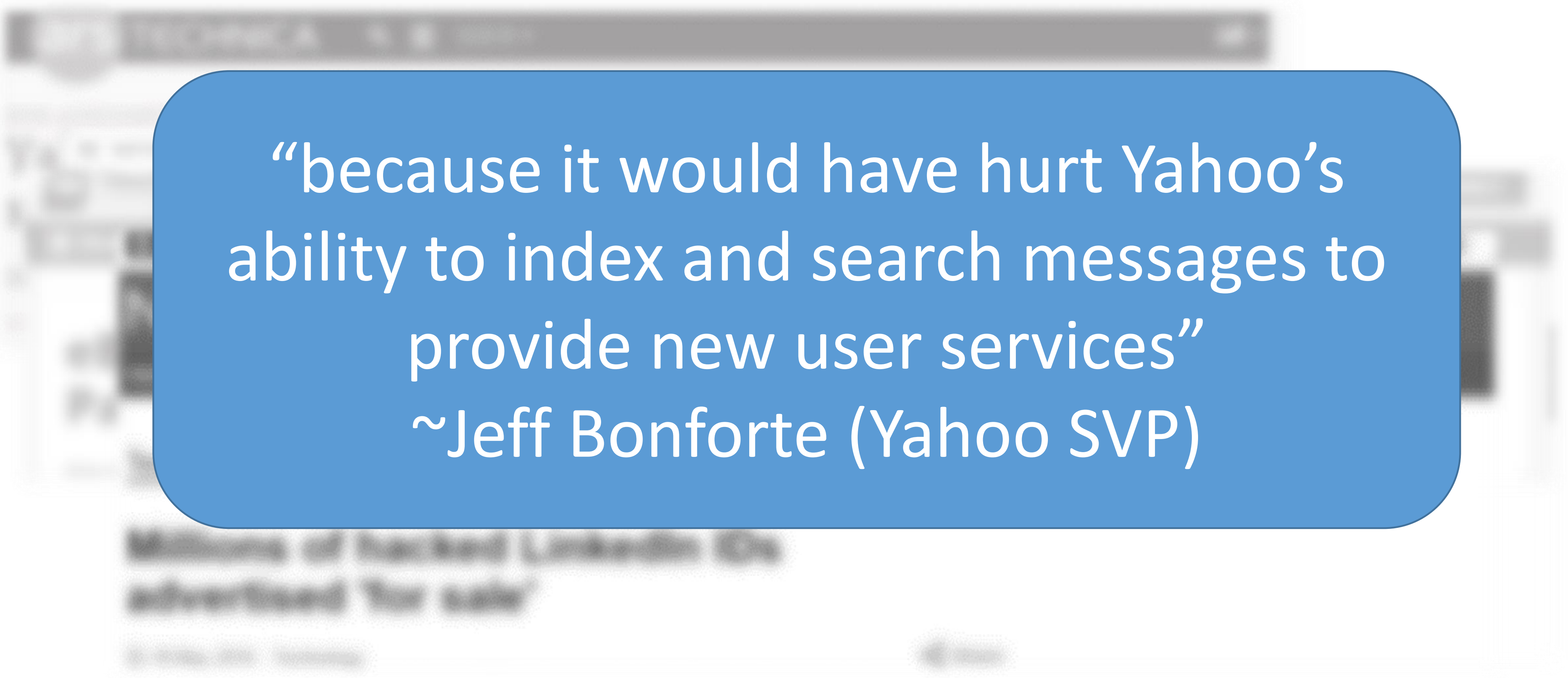


Searching on Encrypted Data



data breaches have become the norm rather than the exception...

Why Not Encrypt?



"because it would have hurt Yahoo's ability to index and search messages to provide new user services"
~Jeff Bonforte (Yahoo SVP)

Order-Revealing Encryption [BLRSZZ'15]

secret-key encryption
scheme

Which is greater: the value encrypted by ct_1 or the value encrypted by ct_2?

sk

$$\begin{aligned} \mathrm{ct}_1 &= \mathrm{Enc}(\mathrm{sk}, 123) \\ \mathrm{ct}_2 &= \mathrm{Enc}(\mathrm{sk}, 512) \\ \mathrm{ct}_3 &= \mathrm{Enc}(\mathrm{sk}, 273) \end{aligned}$$

(legacy-friendly) range queries on encrypted data

client

server

Order-Revealing Encryption [BLRSZZ'15]

given any two ciphertexts

$\mathrm{ct}_1 = \mathrm{Enc}(\mathrm{sk}, x)$

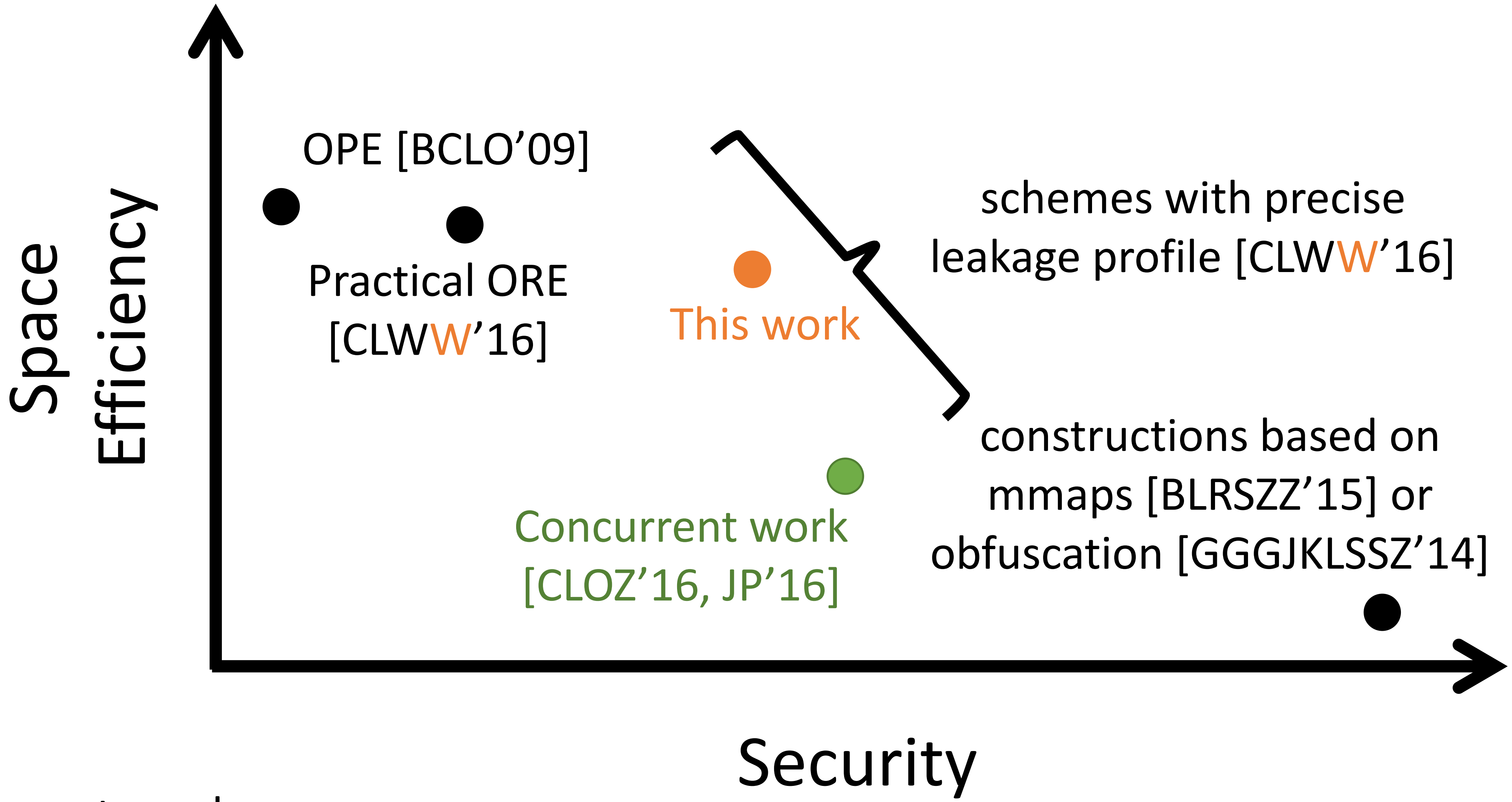
$\mathrm{ct}_2 = \mathrm{Enc}(\mathrm{sk}, y)$

$x > y$

there is a <u>public</u> function for performing comparisons

OPE [BCLO'09]: comparison function is <u>numeric comparison</u> on ciphertexts

The Landscape of ORE



not drawn to scale

Inference Attacks [NKW'15, DDC'16, GSBNR'16]

ID	Name	Age	Diagnosis
wpjOos	2wzXW8	SqX9l9	KqLUXE
XdXdg8	y9GFpS	gwilE3	MJ23b7
P6vKhW	EgN0Jn	S0pRJe	aTaeJk
orJRe6	KQWy9U	tPWF3M	4FBEO0

encrypted database

+

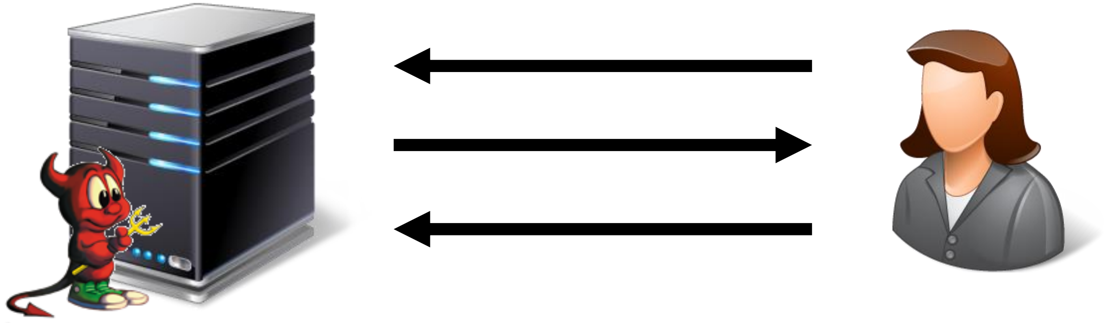
public information

frequency and statistical analysis

ID	Name	Age	Diagnosis
???	Alice	30-35	2
???	Bob	45-50	3
???	Charlie	40-45	2
???	???	40-45	4

plaintext recovery

Online vs. Offline Security



adversary sees encrypted database +
queries and can interact with the database

online attacks (e.g., active corruption)

offline attacks (e.g., passive snapshots)

adversary only sees contents
of encrypted database

typical database breach:
contents of database are stolen
and dumped onto the web

Inference Attacks [NKW’15, DDC’16, GSBNR’16]

ID	Name	Age	Diagnosis
wpjOos	2wzXW8	SqX9l9	KqLUXE
XdXdg8	y9GFpS	gwilE3	MJ23b7
P6vKhW	EgN0Jn	S0pRJe	aTaeJk
orJRe6	KQWy9U	tPWF3M	4FBEO0

encrypted database

\+

public information

frequency and statistical analysis

ID	Name	Age	Diagnosis
???	Alice	30-35	2
???	Bob	45-50	3
???	Charlie	40-45	2
???	???	40-45	4

plaintext recovery

PPE schemes <u>always</u> reveal certain properties (e.g., equality, order) on ciphertexts and thus, are vulnerable to offline inference attacks

Can we <u>fully</u> defend against offline inference attacks while remaining legacy-friendly?

This Work

Can we <u>fully</u> defend against offline inference attacks while remaining legacy-friendly?

Trivial solution: encrypt the entire database, and have client provide decryption key at query time

But no online security!

Desiderata: an ORE scheme that enables:

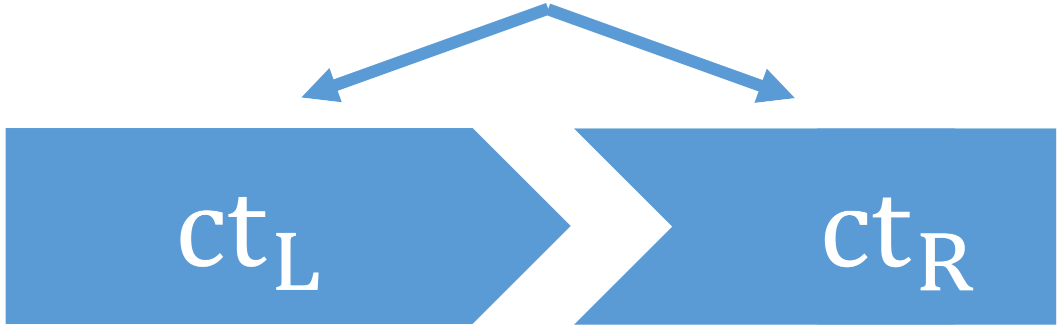
- perfect offline security
- limited leakage in the online setting

ORE with Additional Structure

Focus of this work: performing range queries on encrypted data

Key primitive: order-revealing encryption scheme where ciphertexts have a "decomposable" structure

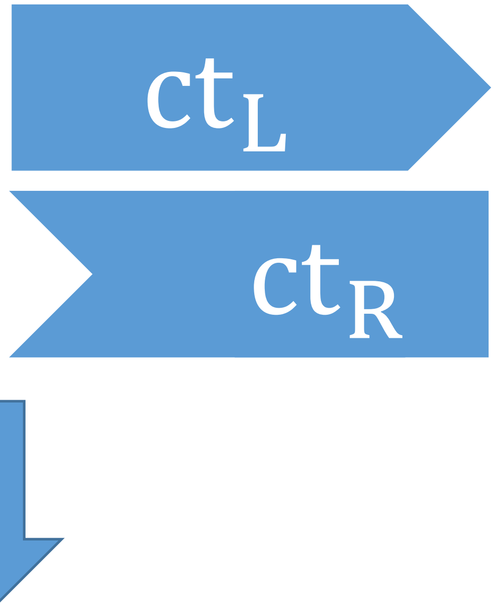
$\mathrm{Enc}(101)$



ciphertexts naturally split into two components

$\mathrm{Enc_L}(101)$

$\mathrm{Enc_R}(100)$



greater than

ORE with Additional Structure

$\mathrm{Enc}_L(101)$ ct_L

$\mathrm{Enc}_R(100)$ ct_R

comparison can be performed between left ciphertext and right ciphertext

right ciphertexts provide **semantic security**!

robustness against offline inference attacks!

Encrypted Range Queries

ID	Name	Age	Diagnosis
0	Alice	31	2
1	Bob	47	3
2	Charlie	41	2
3	Inigo	45	4

build encrypted index

store right ciphertexts in sorted order

Age	ID
$\mathrm{Enc_R}(31)$	$\mathrm{Enc}(0)$
$\mathrm{Enc_R}(41)$	$\mathrm{Enc}(2)$
$\mathrm{Enc_R}(45)$	$\mathrm{Enc}(3)$
$\mathrm{Enc_R}(47)$	$\mathrm{Enc}(1)$

record IDs encrypted under independent key

Name	ID
$\mathrm{Enc_R}(\mathrm{Alice})$	$\mathrm{Enc}(0)$
Enc	
Enc	
Enc	

Age	ID
$\mathrm{Enc_R}(31)$	$\mathrm{Enc}(0)$
Enc	
Enc	
Enc	

Diagnosis	ID
$\mathrm{Enc_R}(2)$	$\mathrm{Enc}(2)$
$\mathrm{Enc_R}(2)$	$\mathrm{Enc}(0)$
$\mathrm{Enc_R}(3)$	$\mathrm{Enc}(1)$
$\mathrm{Enc_R}(4)$	$\mathrm{Enc}(3)$

separate index for each searchable column, and using independent ORE keys

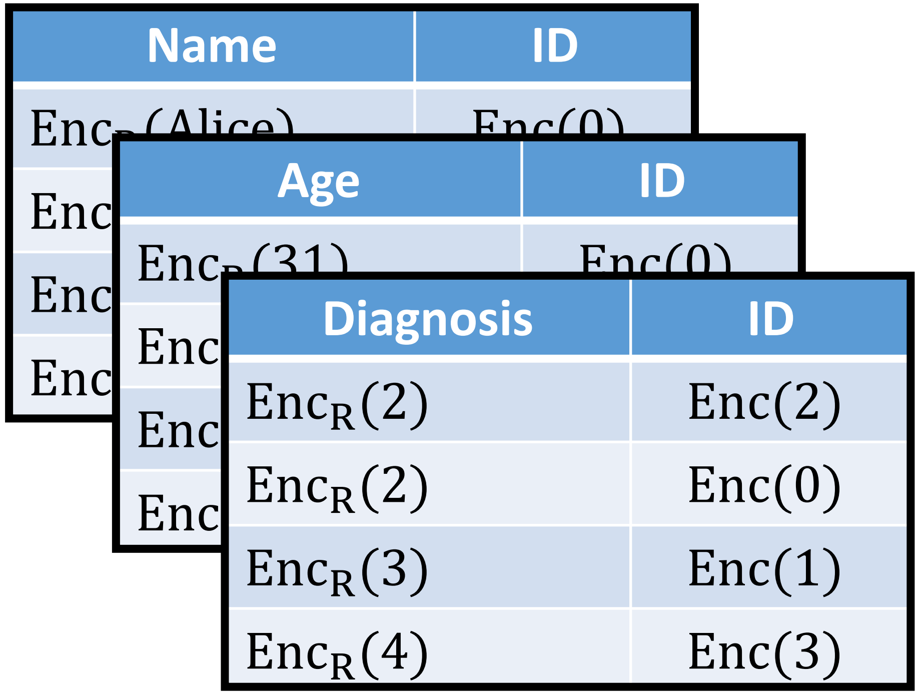
Encrypted Range Queries

Encrypted database:

ID	Name	Age	Diagnosis
0	Alice	31	2
1	Bob	47	3
2	Charlie	41	2
3	Inigo	45	4

columns (other than ID) are encrypted using a semantically-secure encryption scheme

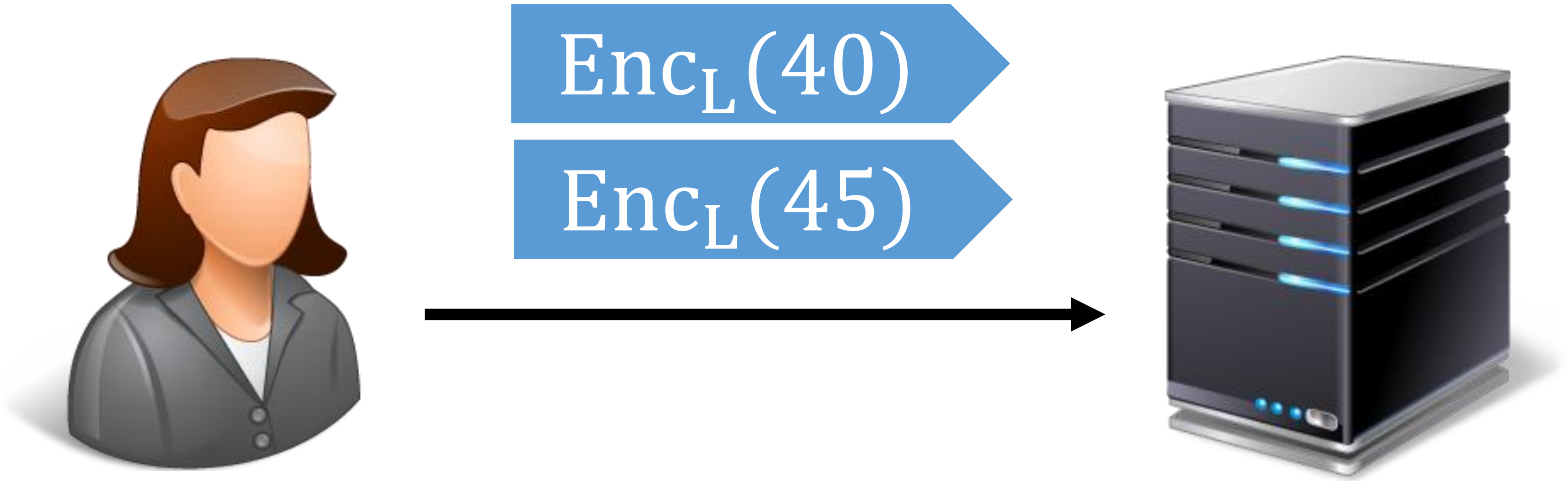
clients hold (secret) keys needed to decrypt and query database



encrypted search indices

Encrypted Range Queries

Query for all records where $40 \geq \text{age} \geq 45$:



Encrypted Range Queries

Query for all records where $40 \geq \text{age} \geq 45$:

$\text{Enc}_\text{L}(40)$

$\text{Enc}_\text{L}(45)$

Age	ID
$\text{Enc}_\text{R}(31)$	$\text{Enc}(0)$
$\text{Enc}_\text{R}(41)$	$\text{Enc}(2)$
$\text{Enc}_\text{R}(45)$	$\text{Enc}(3)$
$\text{Enc}_\text{R}(47)$	$\text{Enc}(1)$

Encrypted Range Queries

Query for all records where $40 \geq \text{age} \geq 45$:

$\text{Enc}_\text{L}(40)$

$\text{Enc}_\text{L}(45)$

Age	ID
$\text{Enc}_\text{R}(31)$	$\text{Enc}(0)$
$\text{Enc}_\text{R}(41)$	$\text{Enc}(2)$
$\text{Enc}_\text{R}(45)$	$\text{Enc}(3)$
$\text{Enc}_\text{R}(47)$	$\text{Enc}(1)$

use binary search to determine endpoints (comparison via ORE)

Encrypted Range Queries

Query for all records where $40 \geq \text{age} \geq 45$:

$\mathrm{Enc_L}(40)$

$\mathrm{Enc_L}(45)$

Age	ID
$\mathrm{Enc_R}(31)$	$\mathrm{Enc}(0)$
$\mathrm{Enc_R}(41)$	$\mathrm{Enc}(2)$
$\mathrm{Enc_R}(45)$	$\mathrm{Enc}(3)$
$\mathrm{Enc_R}(47)$	$\mathrm{Enc}(1)$

use binary search to determine endpoints (comparison via ORE)

Encrypted Range Queries

Query for all records where $40 \geq \text{age} \geq 45$:

$\text{Enc}_L(40)$

$\text{Enc}_L(45)$

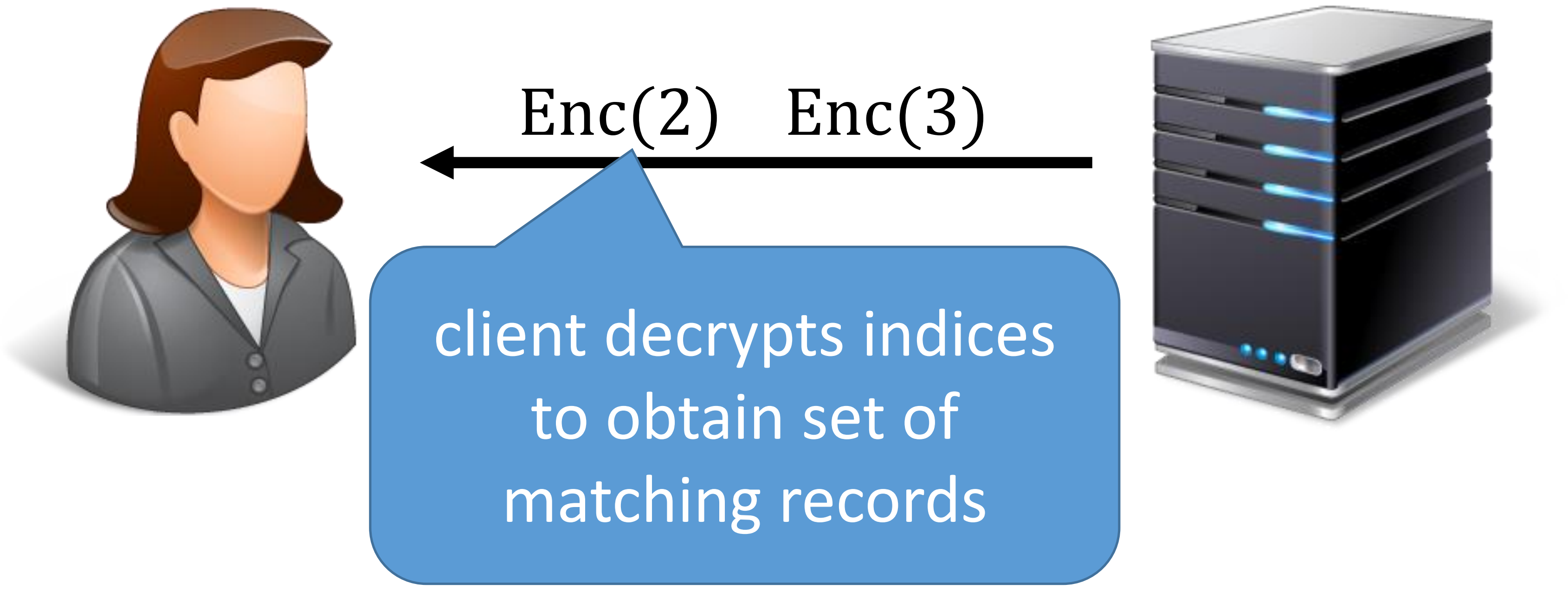
Age	ID
$\text{Enc}_R(31)$	$\text{Enc}(0)$
$\text{Enc}_R(41)$	$\text{Enc}(2)$
$\text{Enc}_R(45)$	$\text{Enc}(3)$
$\text{Enc}_R(47)$	$\text{Enc}(1)$

return encrypted indices that match query

use binary search to determine endpoints (comparison via ORE)

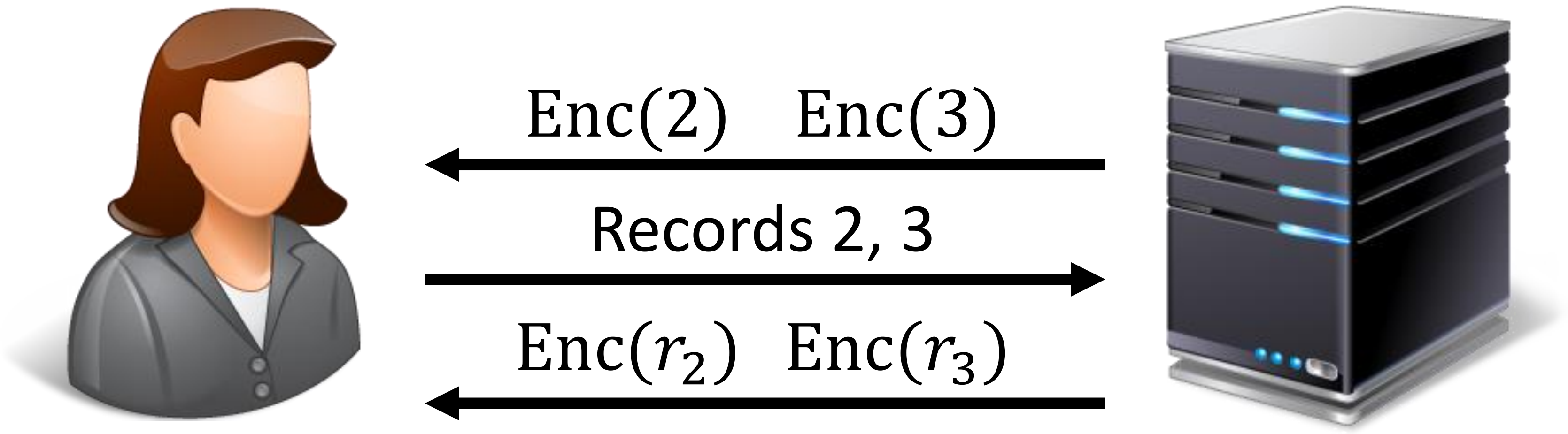
Encrypted Range Queries

Query for all records where $40 \geq \text{age} \geq 45$:



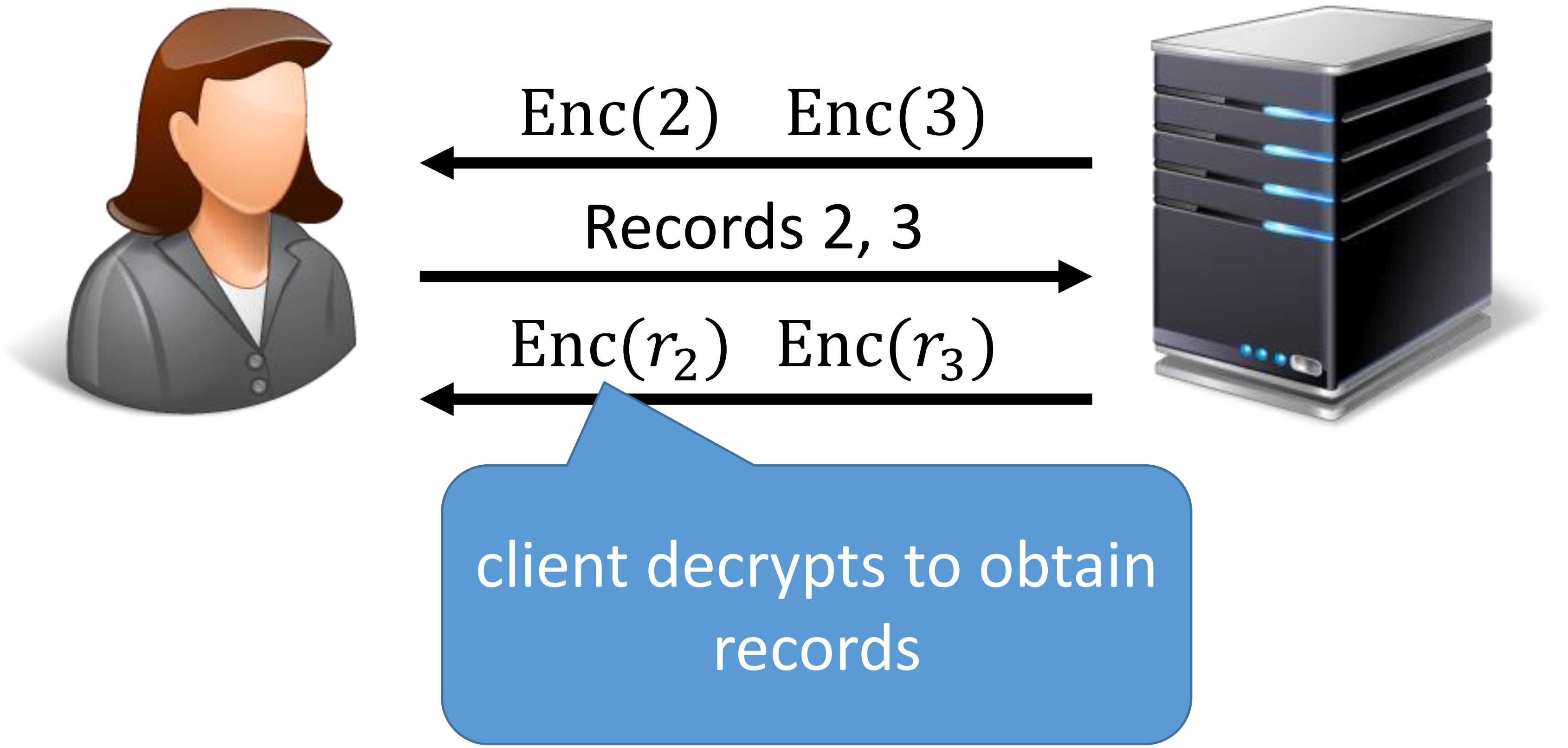
Encrypted Range Queries

Query for all records where $40 \geq \text{age} \geq 45$:



Encrypted Range Queries

Query for all records where $40 \geq \text{age} \geq 45$:



Encrypted Range Queries

Query for all records where $40 \geq \text{age} \geq 45$:

$\mathrm{Enc}(2) \quad \mathrm{Enc}(3)$

Records 2, 3

$\mathrm{Enc}(r_2) \quad \mathrm{Enc}($

some online leakage: access pattern + ORE leakage

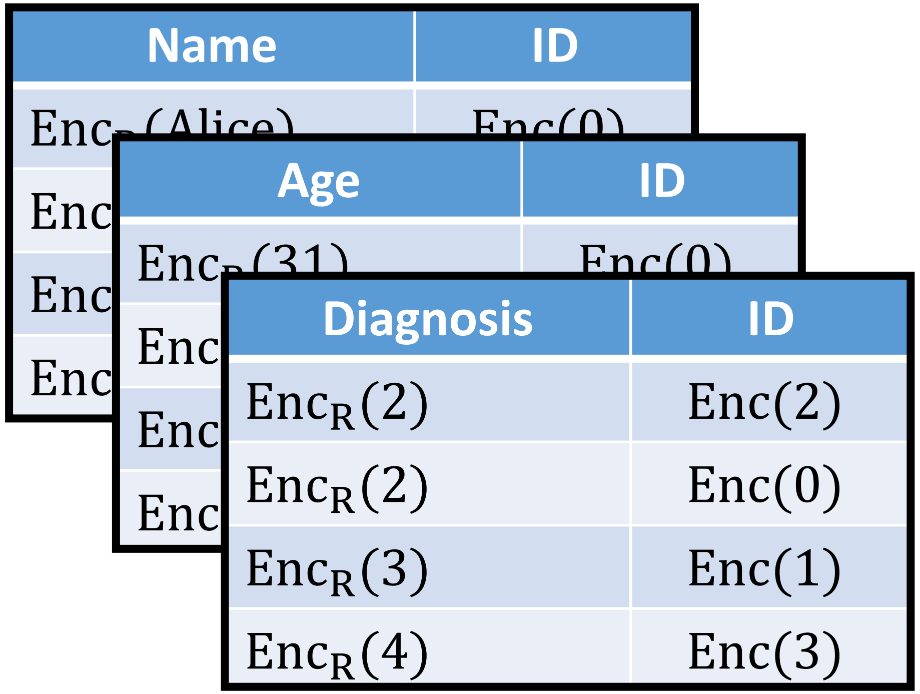
Encrypted Range Queries

Encrypted database:

ID	Name	Age	Diagnosis
0	Alice	31	2
1	Bob	47	3
2	Charlie	41	2
3	Inigo	45	4

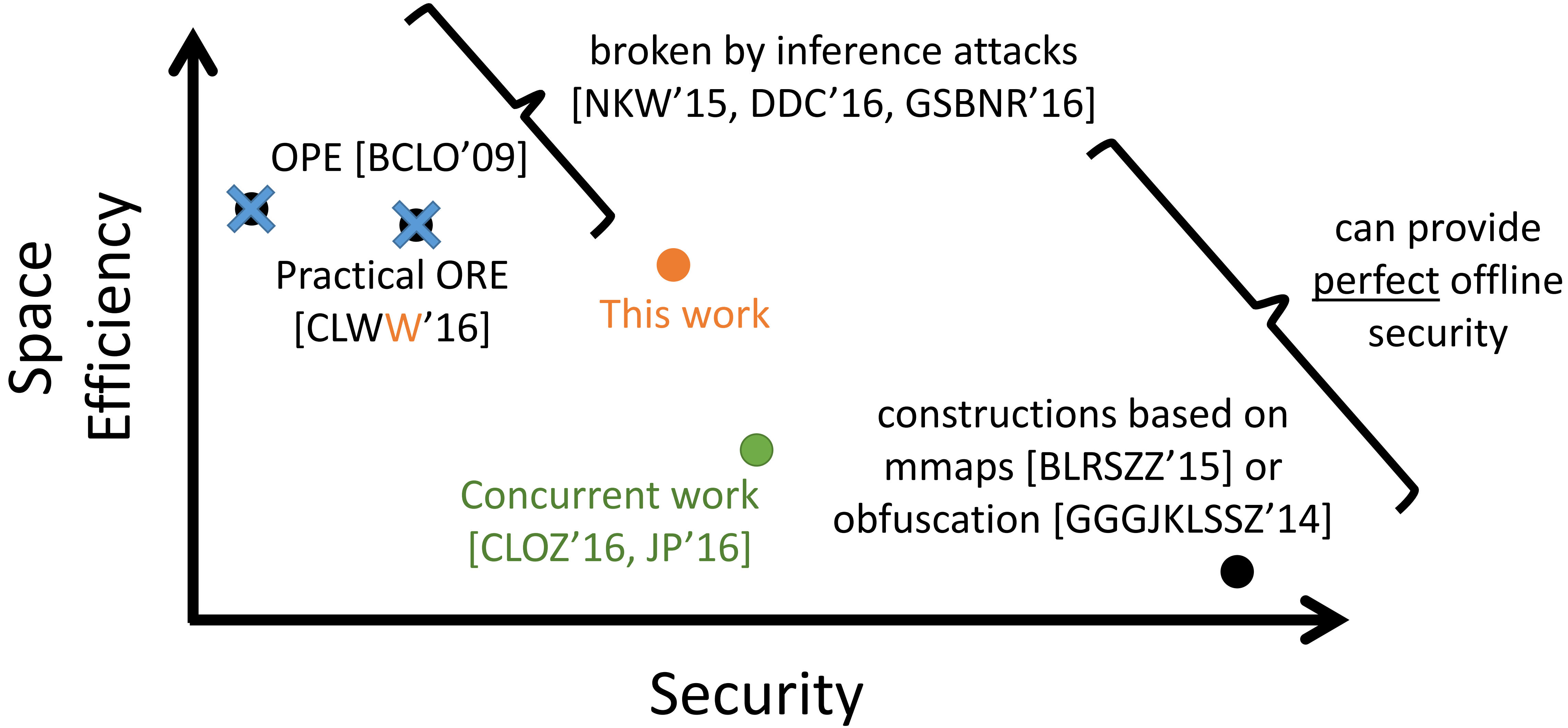
encrypted database is semantically secure!

<u>Perfect offline security</u>



encrypted search indices

The Landscape of ORE



Not drawn to scale

Our New ORE Scheme

“small-domain” ORE with best-possible security

+

domain extension technique inspired by CLWW’16

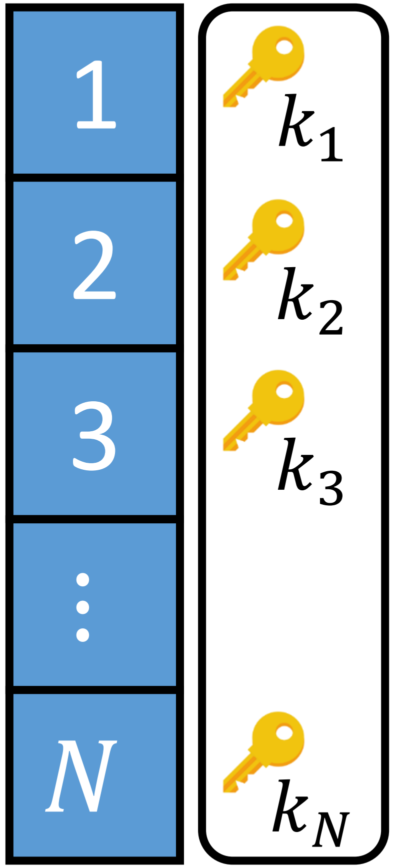
→

“large-domain” ORE with some leakage

Small-Domain ORE with Best-Possible Security

Suppose plaintext space is small: $\{1,2,\ldots,N\}$

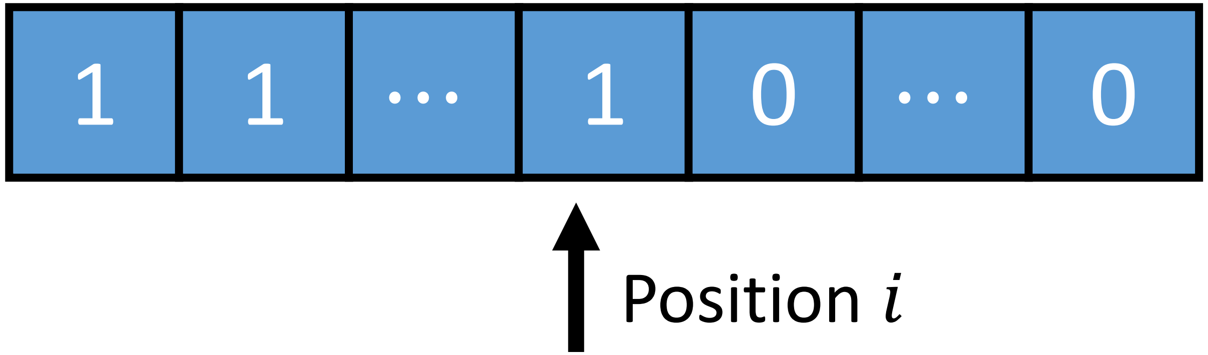
associate a key
with each value



$(k_1,\ldots,k_N)$ is the secret key
(can be derived from a PRF)

Small-Domain ORE with Best-Possible Security

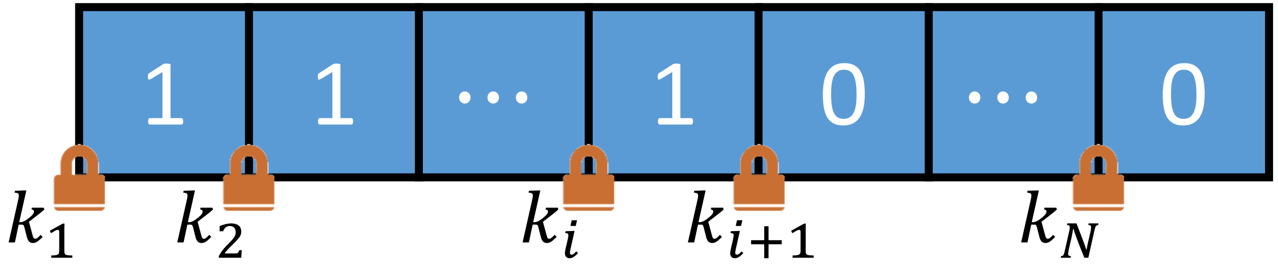
Encrypting a value i



Invariant: all positions $\leq i$ have value 1 while all positions $> i$ have value 0

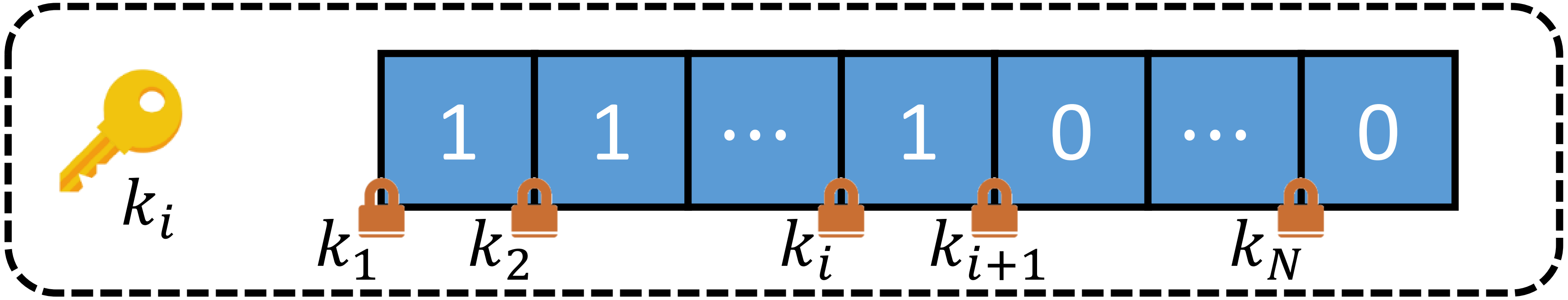
Small-Domain ORE with Best-Possible Security

Encrypting a value i



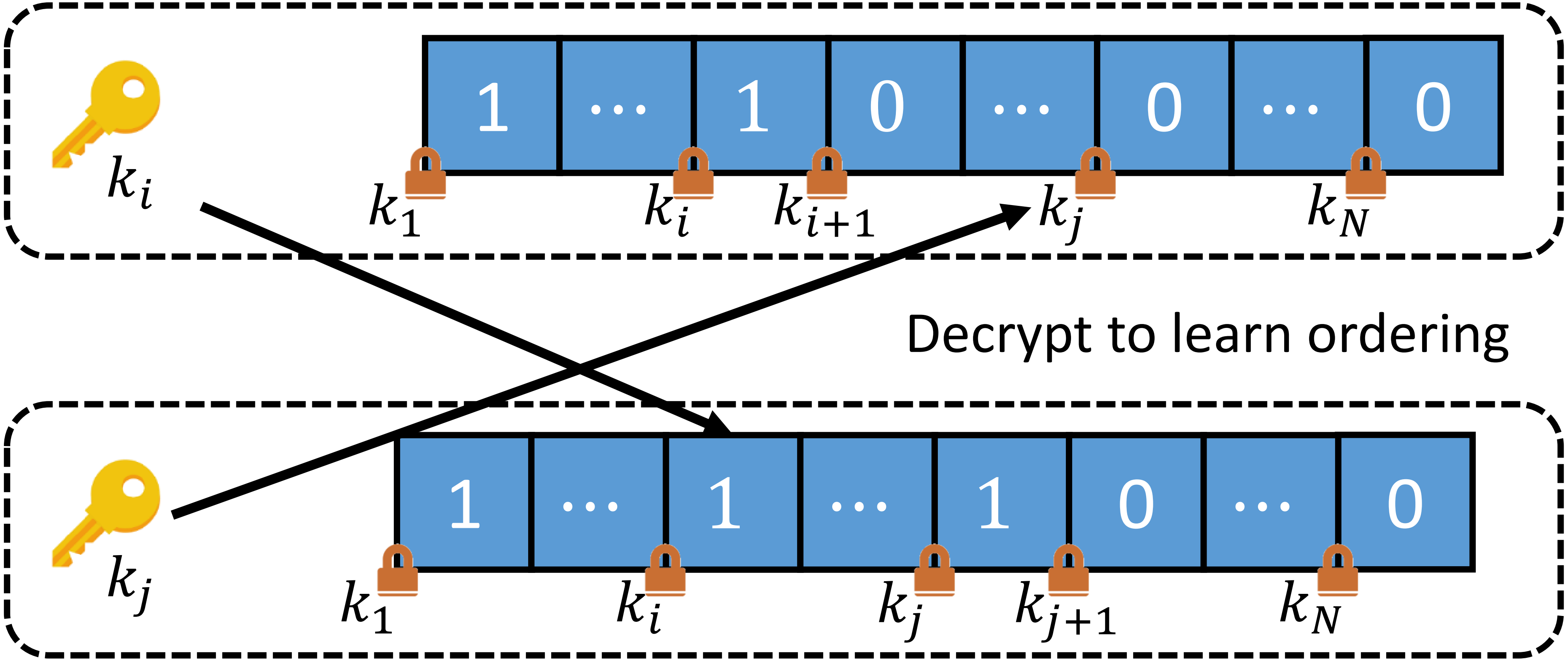
encrypt each slot with key for that slot

To allow comparisons, also give out key for slot i



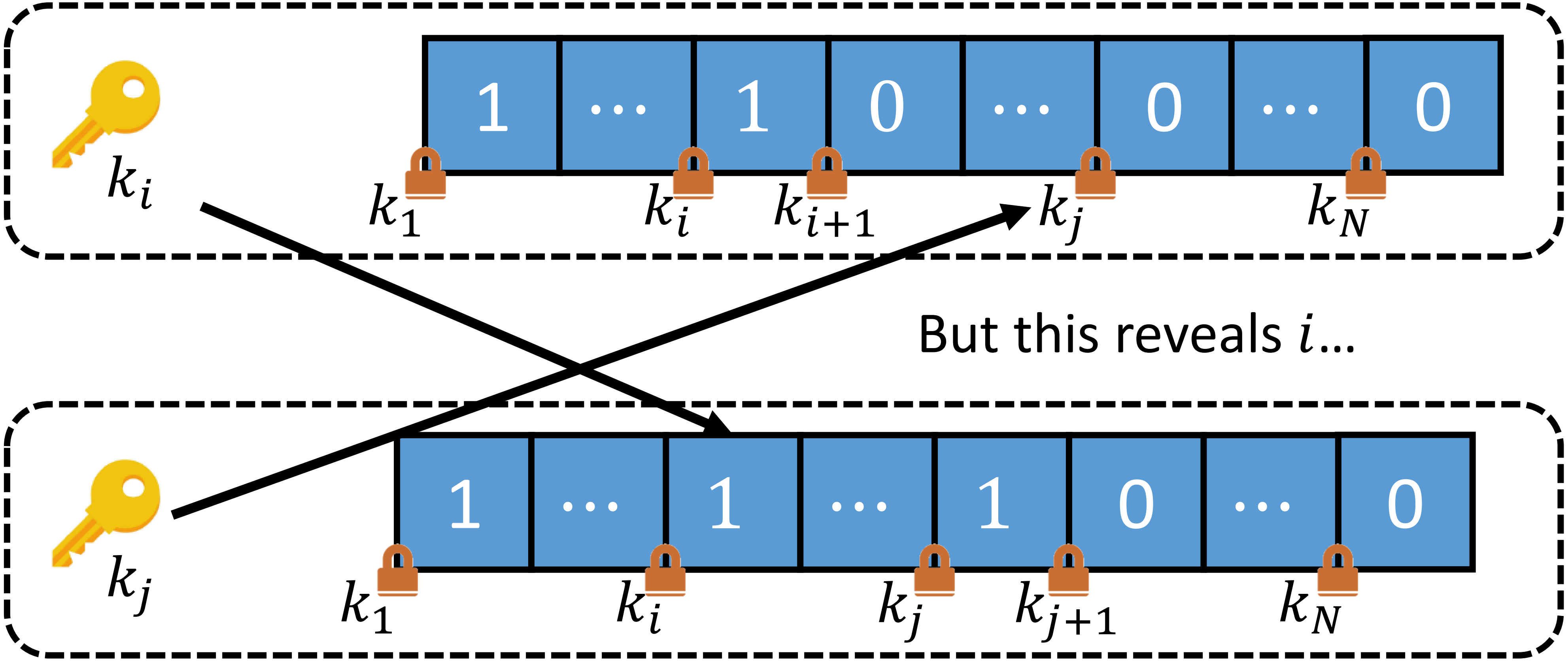
Small-Domain ORE with Best-Possible Security

Given two ciphertexts



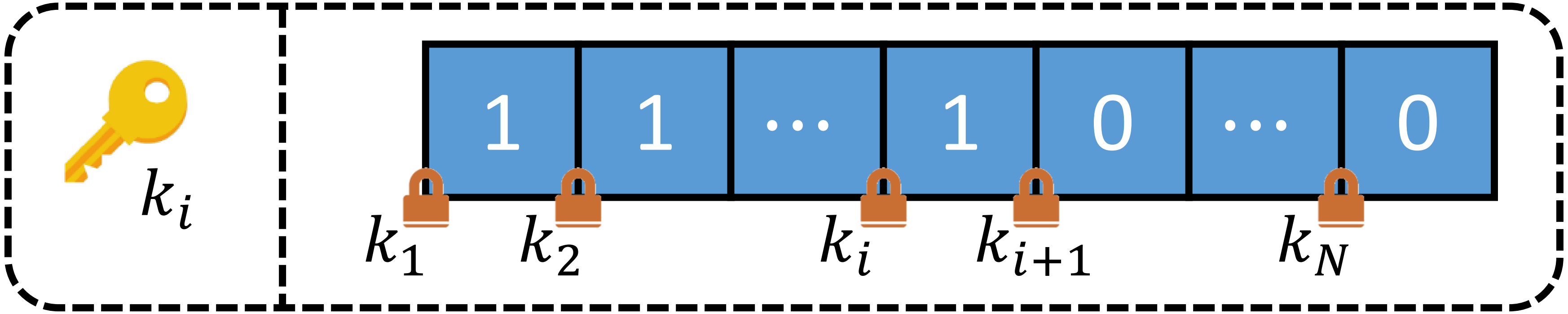
Small-Domain ORE with Best-Possible Security

Given two ciphertexts



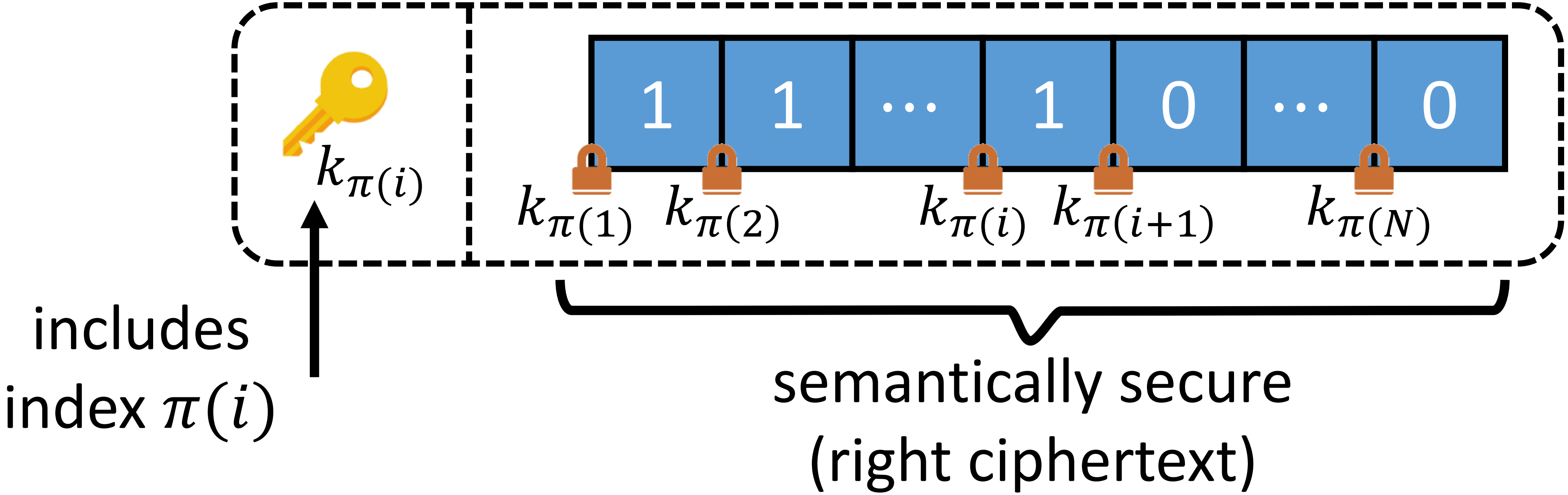
Small-Domain ORE with Best-Possible Security

Solution: apply random permutation π (part of the secret key) to the slots



Small-Domain ORE with Best-Possible Security

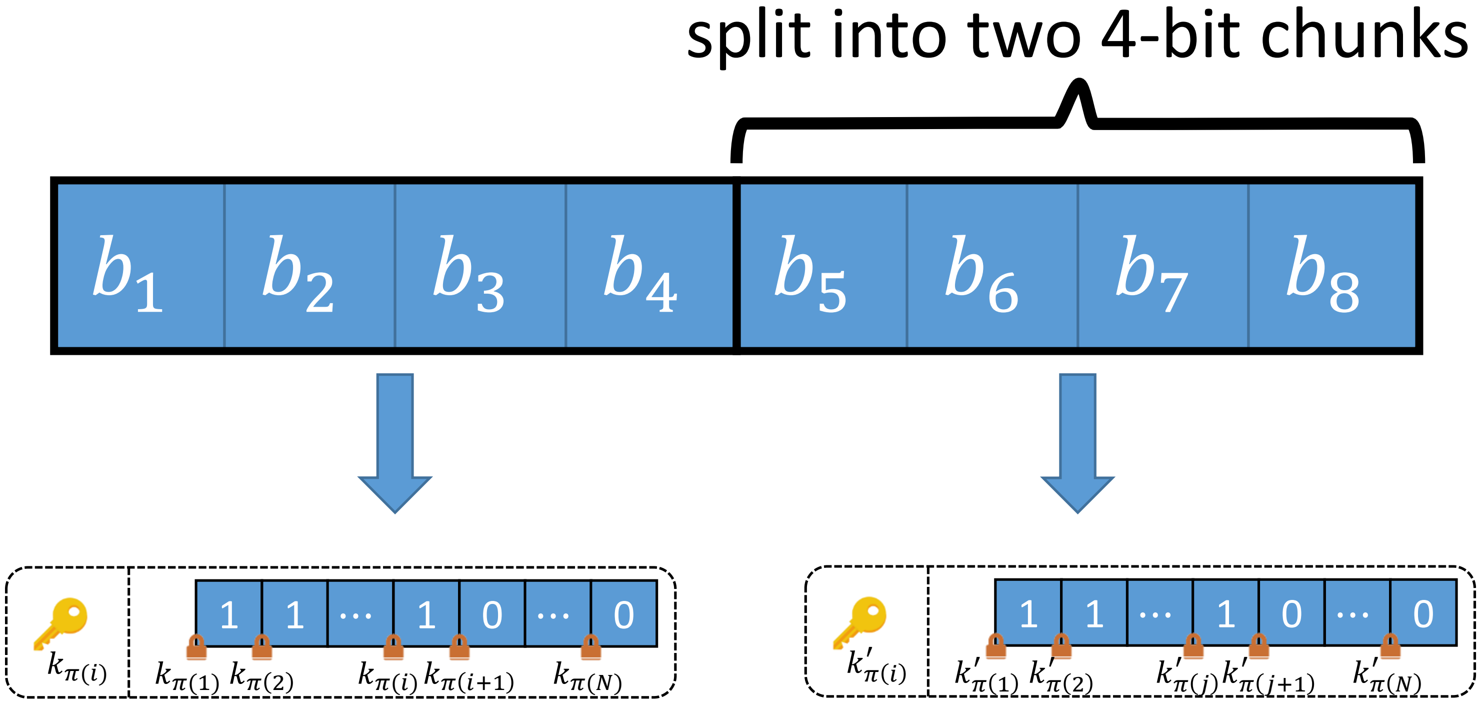
Solution: apply random permutation π (part of the secret key) to the slots



Achieves best-possible security, but ciphertexts are big

Domain Extension for ORE

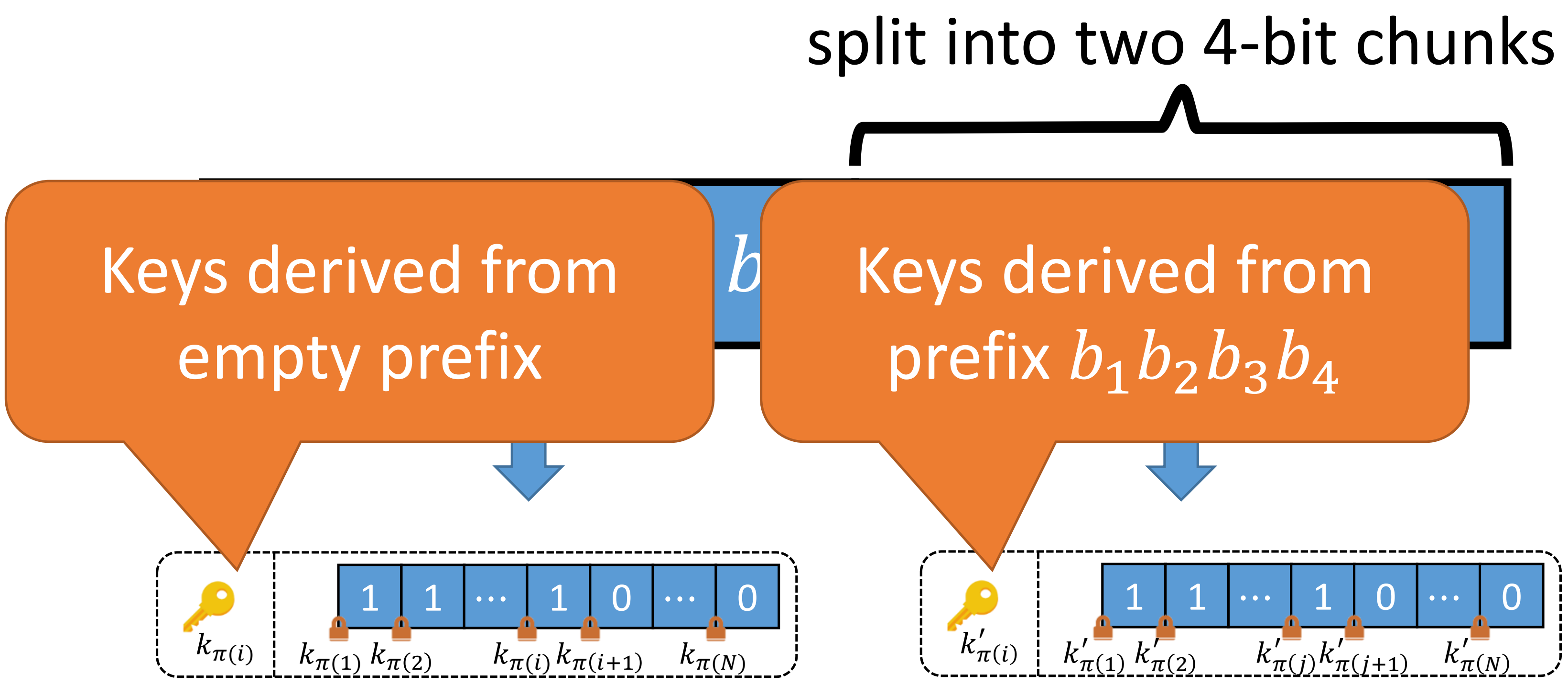
Key idea: decompose message into smaller blocks and apply small-domain ORE to each block



encrypt each chunk using an ORE instance with a secret key derived from the *prefix*

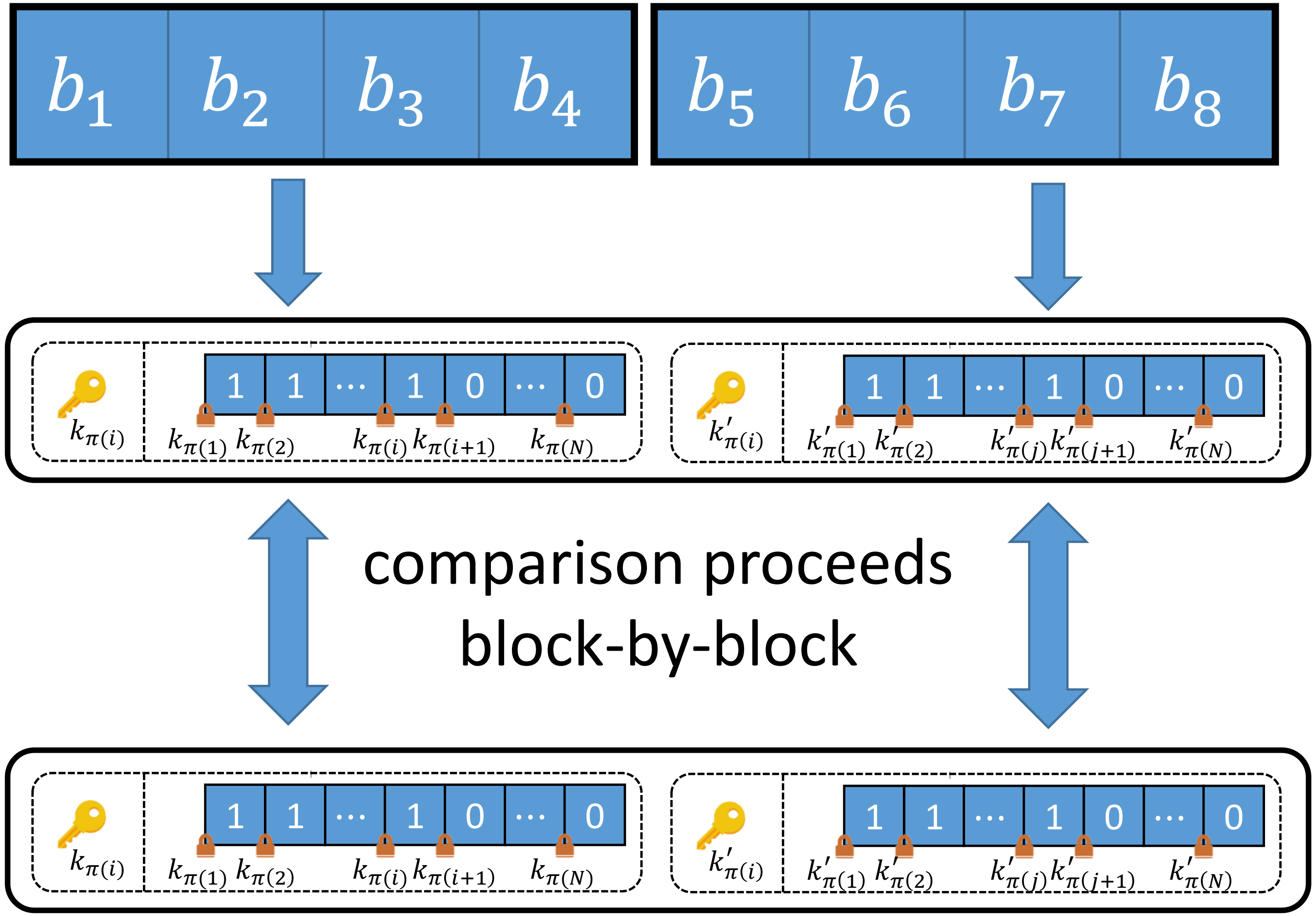
Domain Extension for ORE

Key idea: decompose message into smaller blocks and apply small-domain ORE to each block



encrypt each chunk using an ORE instance with a secret key derived from the *prefix*

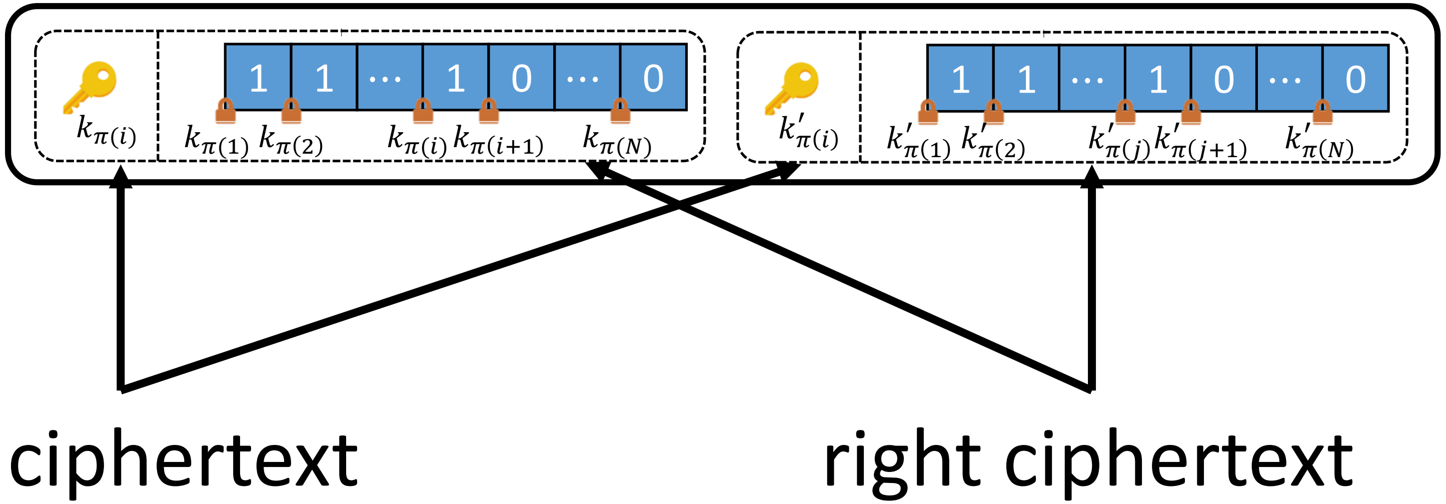
Domain Extension for ORE



Overall leakage: first **block** that differs

Domain Extension for ORE

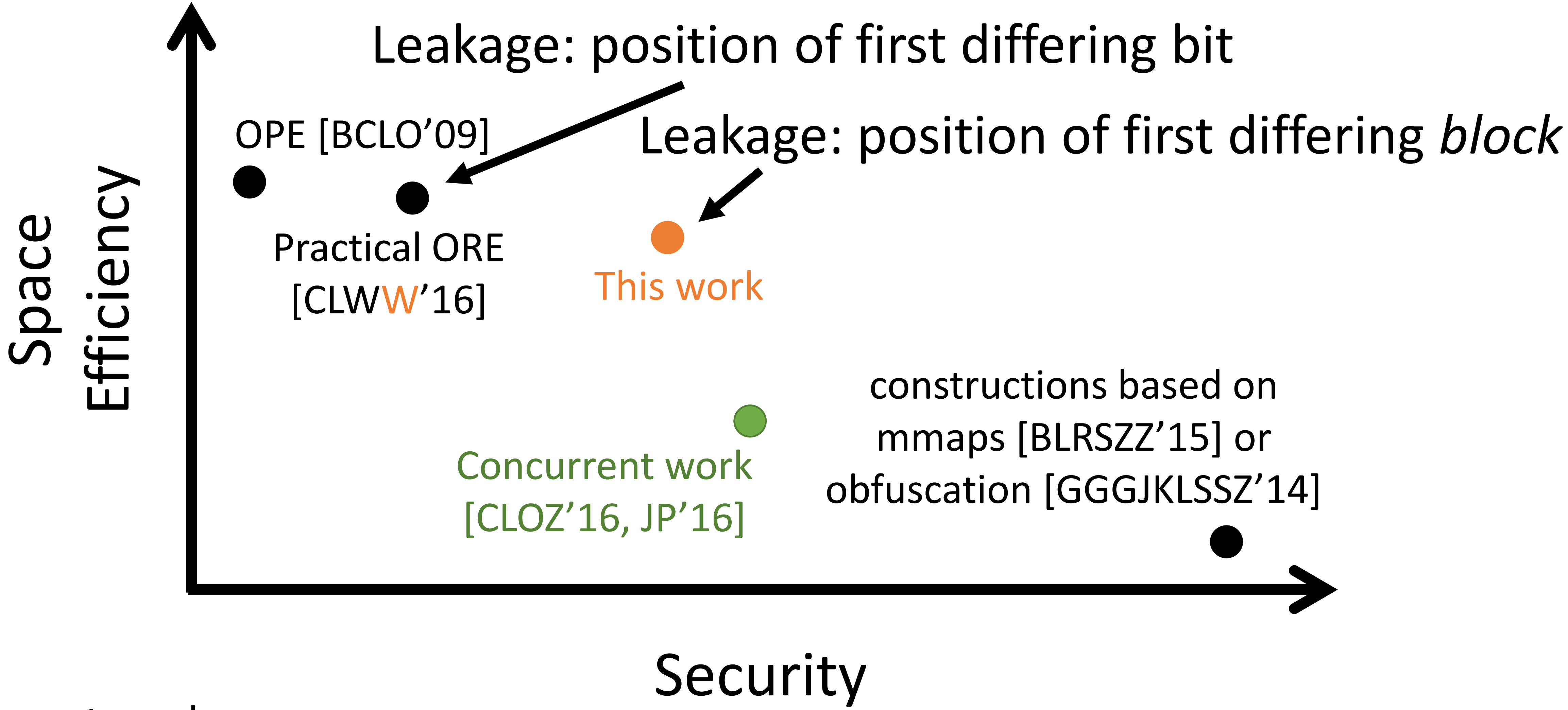
Same decomposition into left and right ciphertexts:



Right ciphertexts provide semantic security!

Note: optimizations are possible if we apply this technique in a non-black-box way to the small-domain ORE. See paper for details.

The Landscape of ORE



not drawn to scale

Performance Evaluation

Scheme	Encrypt (μs)	Compare (μs)	|ct| (bytes)
OPE [BCLO'09]	3601.82	0.36	8
Practical ORE [CLWW'16]	2.06	0.48	8
This work (4-bit blocks)	16.50	0.31	192
This work (8-bit blocks)	54.87	0.63	224
This work (12-bit blocks)	721.37	2.61	1612

Benchmarks taken for C implementation of different schemes (with AES-NI). Measurements for encrypting 32-bit integers.

Performance Evaluation

Scheme	Encrypt (μs)	Compare (μs)	|ct| (bytes)
OPE [BCLO'09]	3601.82	0.36	8
Practical ORE [CLWW'16]	2.06	0.48	8
This work (4-bit blocks)	16.50	0.31	192
This work (8-bit blocks)	54.87	0.63	224
This work (12-bit blocks)	721.37	2.61	1612

Encrypting byte-size blocks is 65x faster than OPE, but ciphertexts are 30x longer. Security is substantially better.

Performance Evaluation

Scheme	Encrypt (μs)	Compare (μs)	|ct| (bytes)
OPE [BCLO'09]	3601.82	0.36	8
Practical ORE [CLWW'16]	2.06	0.48	8
This work (4-bit blocks)	16.50	0.31	192
This work (8-bit blocks)	54[illegible]	[illegible]	[illegible]224
This work (12-bit blocks)	721[illegible]	[illegible]	[illegible]612

Can be substantial, but usually ORE would only be used for short fields.

Encrypting byte-size blocks [illegible]x faster than OPE, but ciphertexts are 30x longer. Security is substantially better.

Conclusions

ID	Name	Age	Diagnosis
wpjOos	2wzXW8	SqX9I9	KqLUXE
XdXdg8	y9GFpS	gwilE3	MJ23b7
P6vKhW	EgN0Jn	S0pRJe	aTaeJk
orJRe6	KQWy9U	tPWF3M	4FBEO0

encrypted database

+

public information

frequency and statistical analysis

ID	Name	Age	Diagnosis
???	Alice	30-35	2
???	Bob	45-50	3
???	Charlie	40-45	2
???	???	40-45	4

plaintext recovery

- Inference attacks render most conventional PPE-based constructions insecure
- However, ORE is still a useful building block for encrypted databases

- Introduced new paradigm for constructing ORE that enables range queries in a way that is mostly <u>legacy-compatible</u> and provides <u>offline semantic security</u>
- New ORE construction that is concretely efficient with strong security
- In paper: new impossibility results for security achievable using OPE

Paper: https://eprint.iacr.org/2016/612
Website: https://crypto.stanford.edu/ore/
Code: https://github.com/kevinlewi/fastore