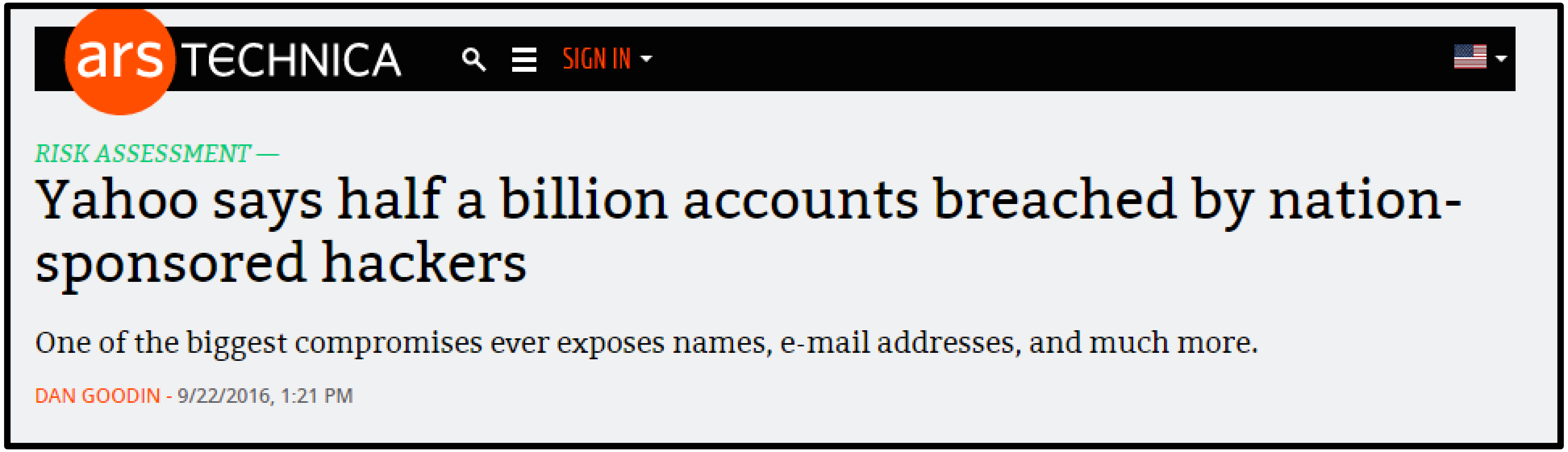
Order-Revealing Encryption:

New Constructions, Applications and Lower Bounds

Kevin Lewi and David J. Wu

Stanford University

Searching on Encrypted Data



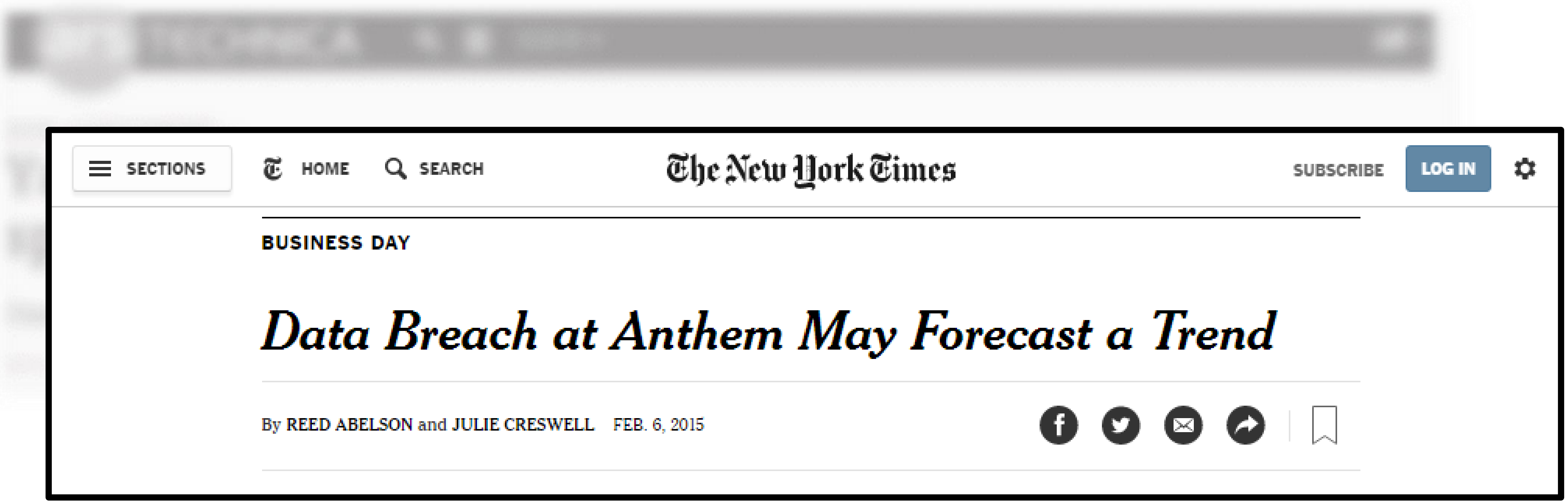
RISK ASSESSMENT —

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

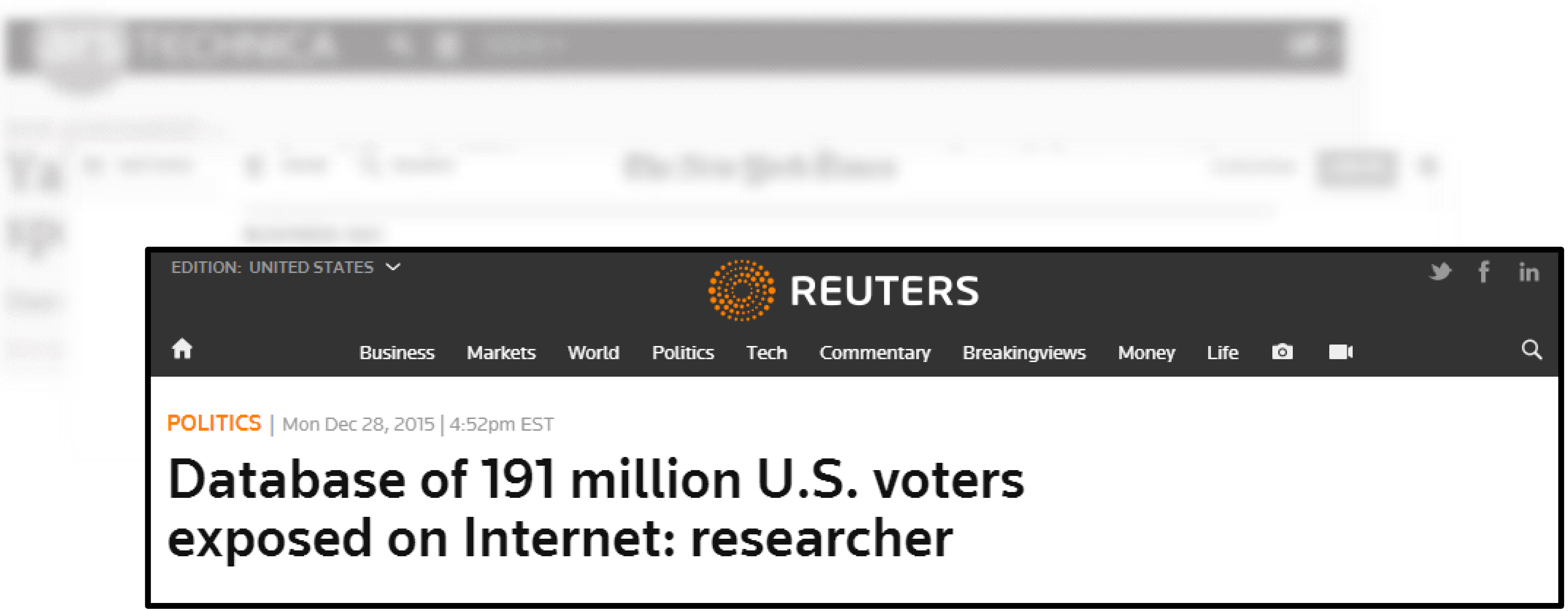


BUSINESS DAY

Data Breach at Anthem May Forecast a Trend

By REED ABELSON and JULIE CRESWELL FEB. 6, 2015

Searching on Encrypted Data



EDITION: UNITED STATES

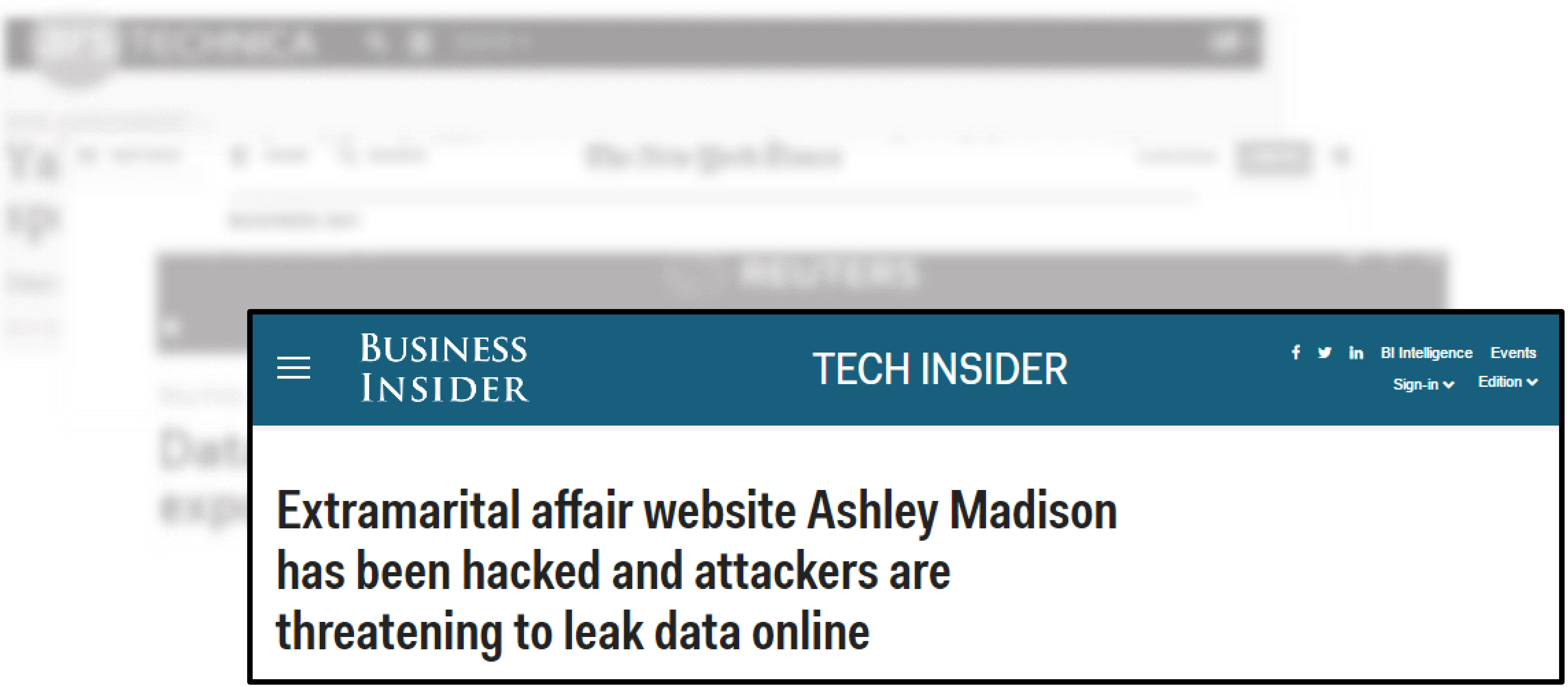
REUTERS

Business Markets World Politics Tech Commentary Breakingviews Money Life

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

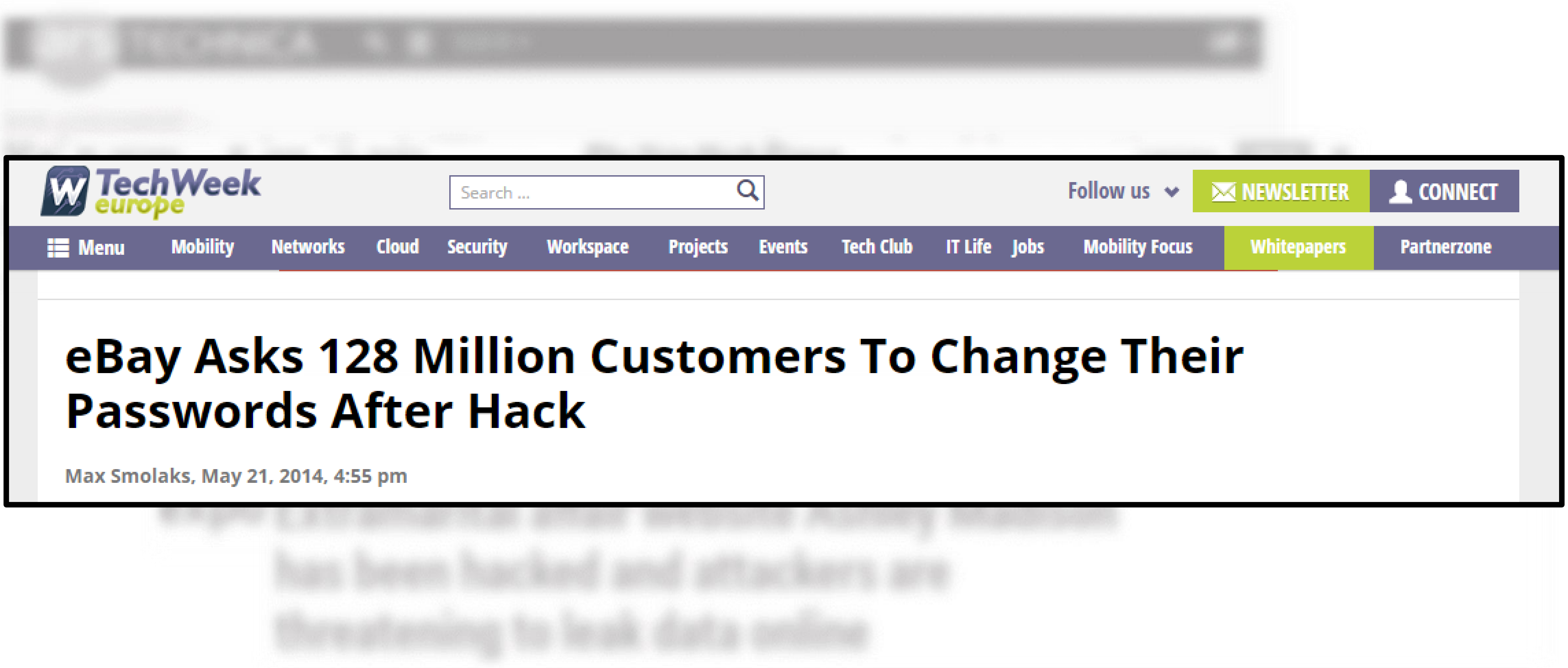
Database of 191 million U.S. voters exposed on Internet: researcher

Searching on Encrypted Data



Extramarital affair website Ashley Madison has been hacked and attackers are threatening to leak data online

Searching on Encrypted Data

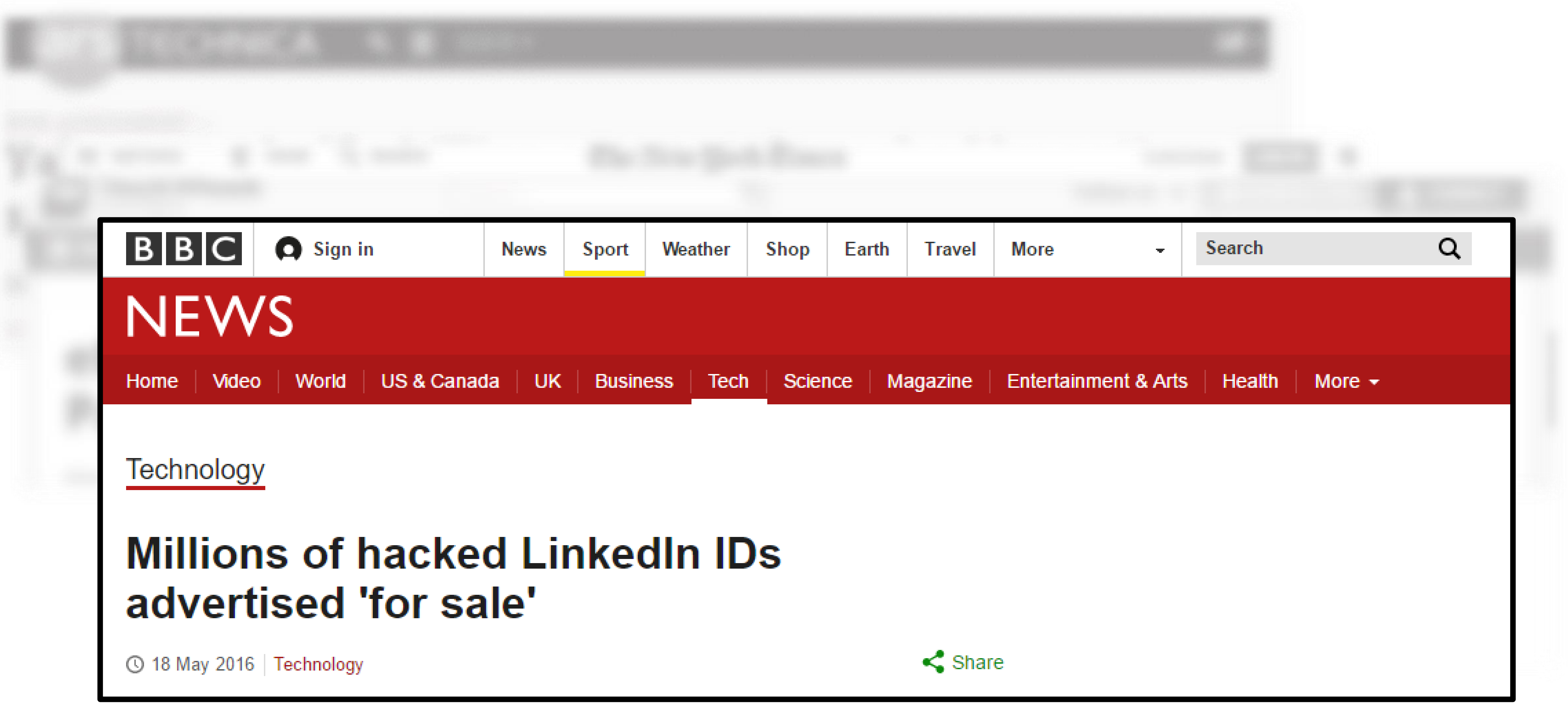


eBay Asks 128 Million Customers To Change Their Passwords After Hack

Max Smolaks, May 21, 2014, 4:55 pm

has been hacked and attackers are threatening to leak data online

Searching on Encrypted Data

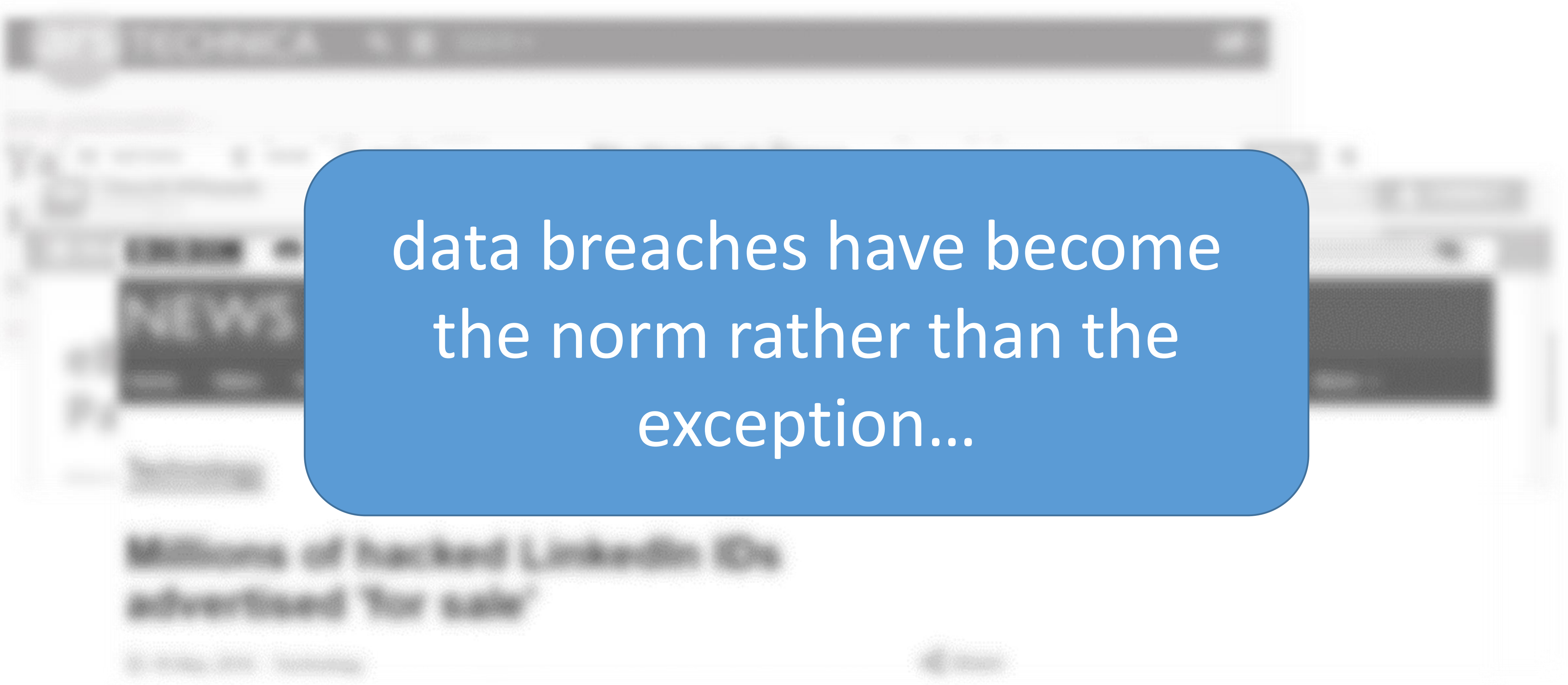


Technology

Millions of hacked LinkedIn IDs advertised 'for sale'

18 May 2016 | Technology

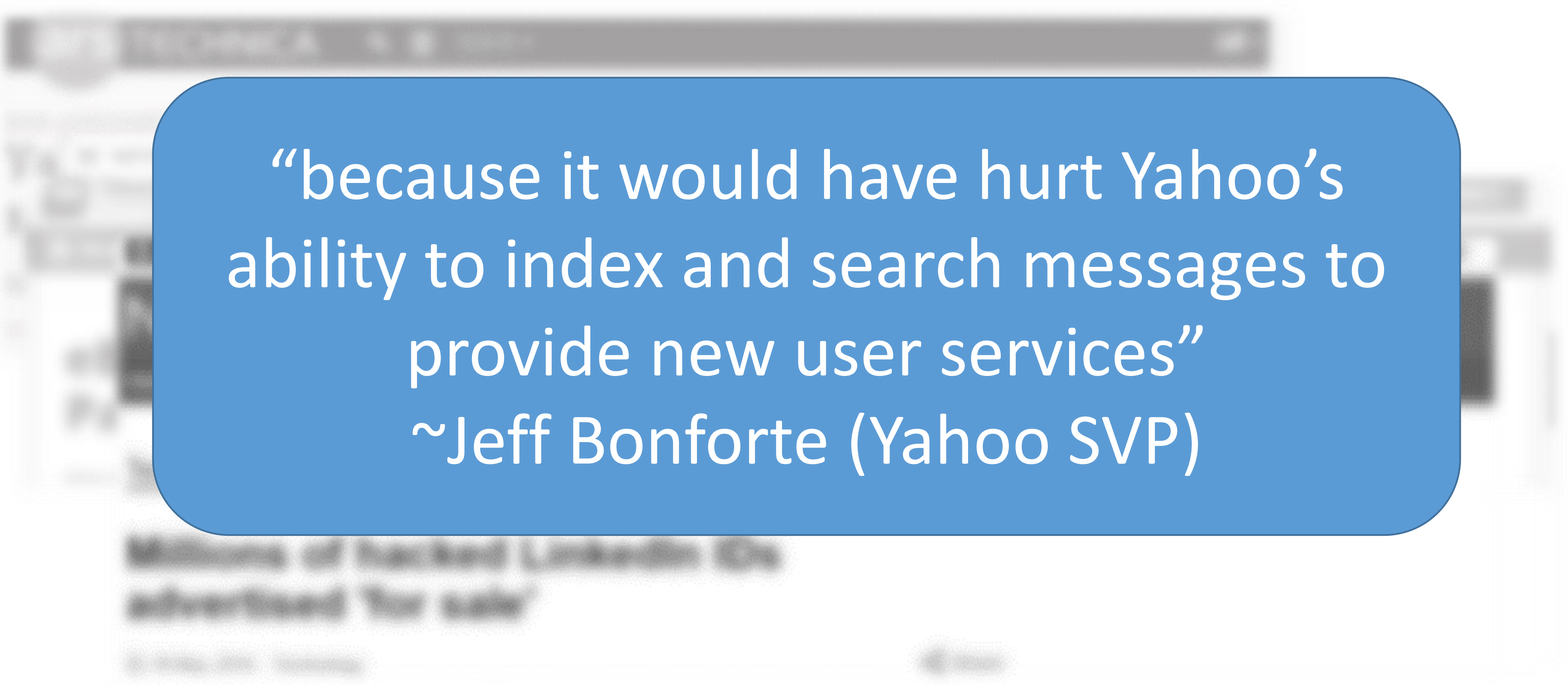
Searching on Encrypted Data



Millions of hacked LinkedIn IDs advertised 'for sale'

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

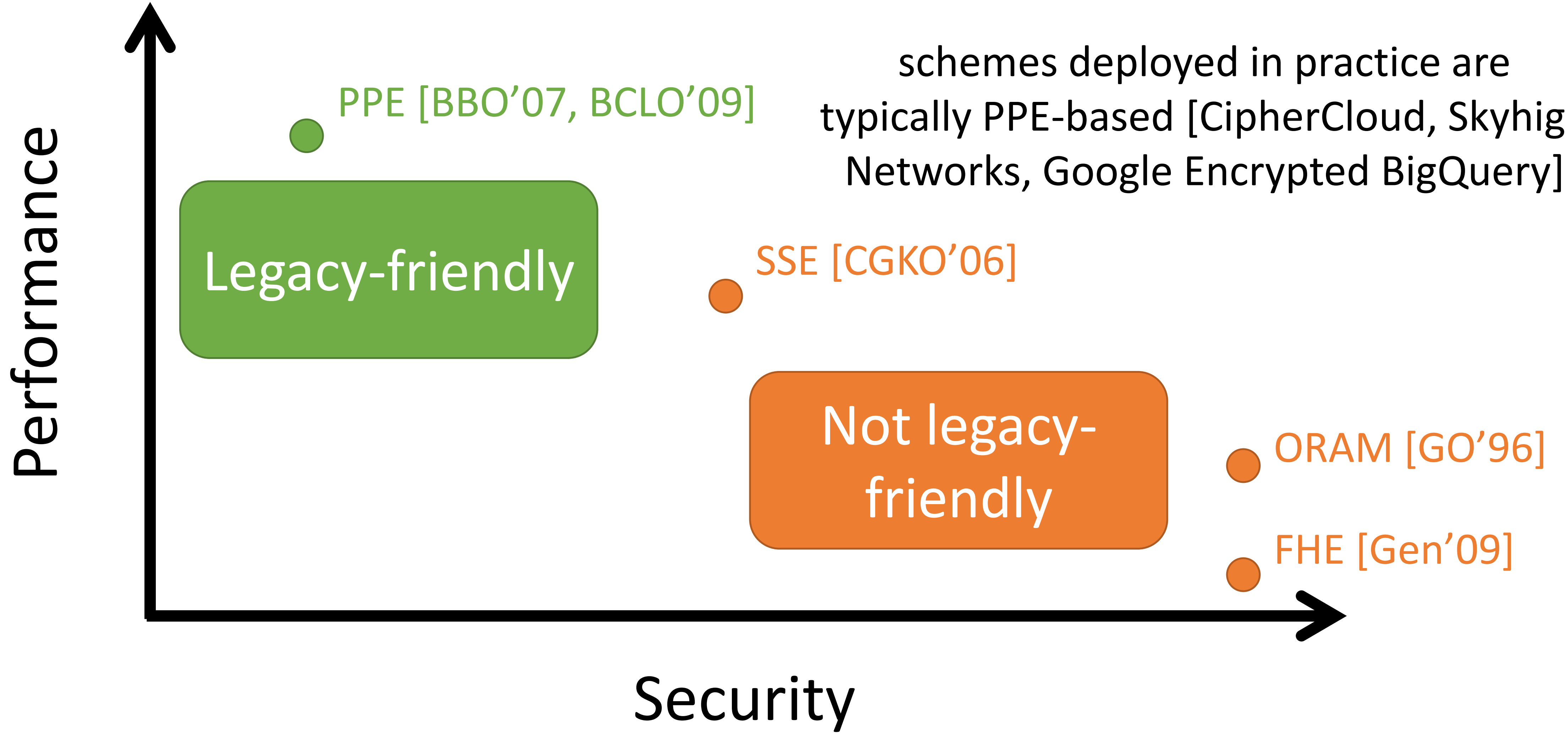
Why Not Encrypt?



Millions of hacked LinkedIn IDs advertised 'for sale'

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

Searching on Encrypted Data

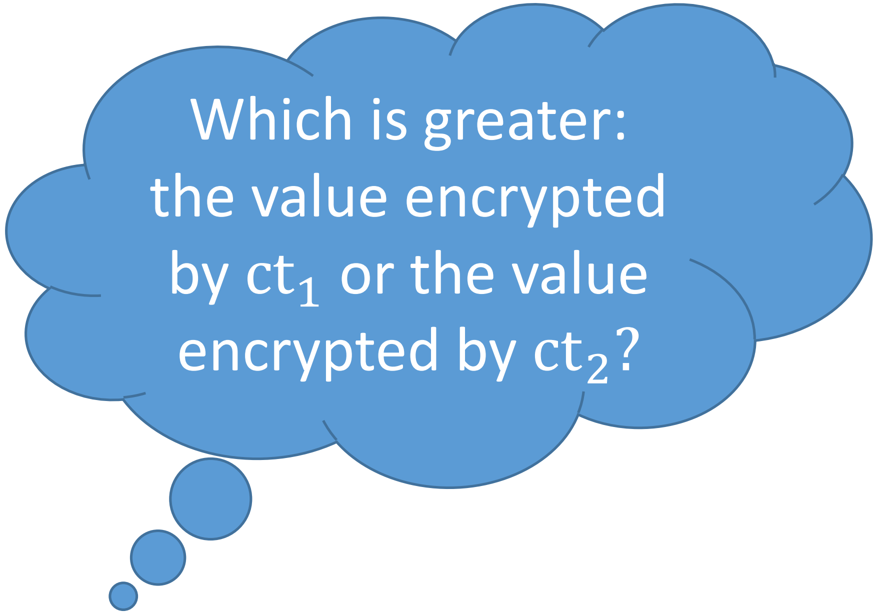


schemes deployed in practice are typically PPE-based [CipherCloud, Skyhigh Networks, Google Encrypted BigQuery]

not drawn to scale

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?

client

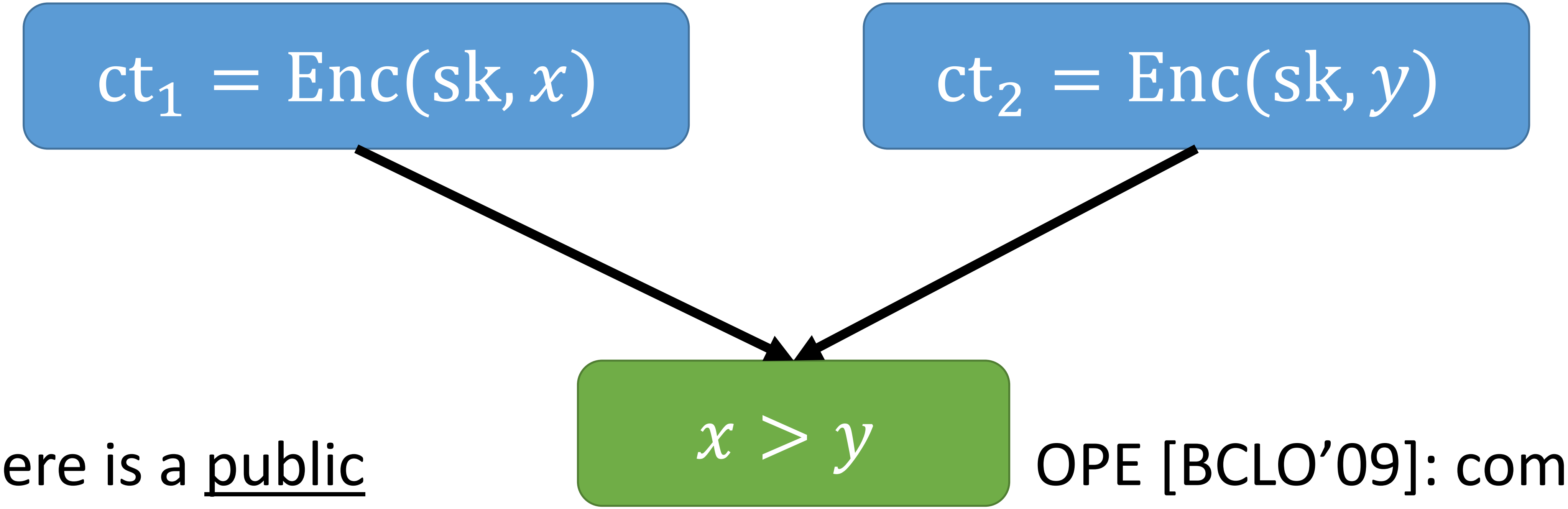
$$\mathrm{ct}_1 = \mathrm{Enc}(\mathrm{sk}, 123)$$
$$\mathrm{ct}_2 = \mathrm{Enc}(\mathrm{sk}, 512)$$
$$\mathrm{ct}_3 = \mathrm{Enc}(\mathrm{sk}, 273)$$

server

range queries on encrypted data

Order-Revealing Encryption [BLRSZZ'15]

given any two ciphertexts



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

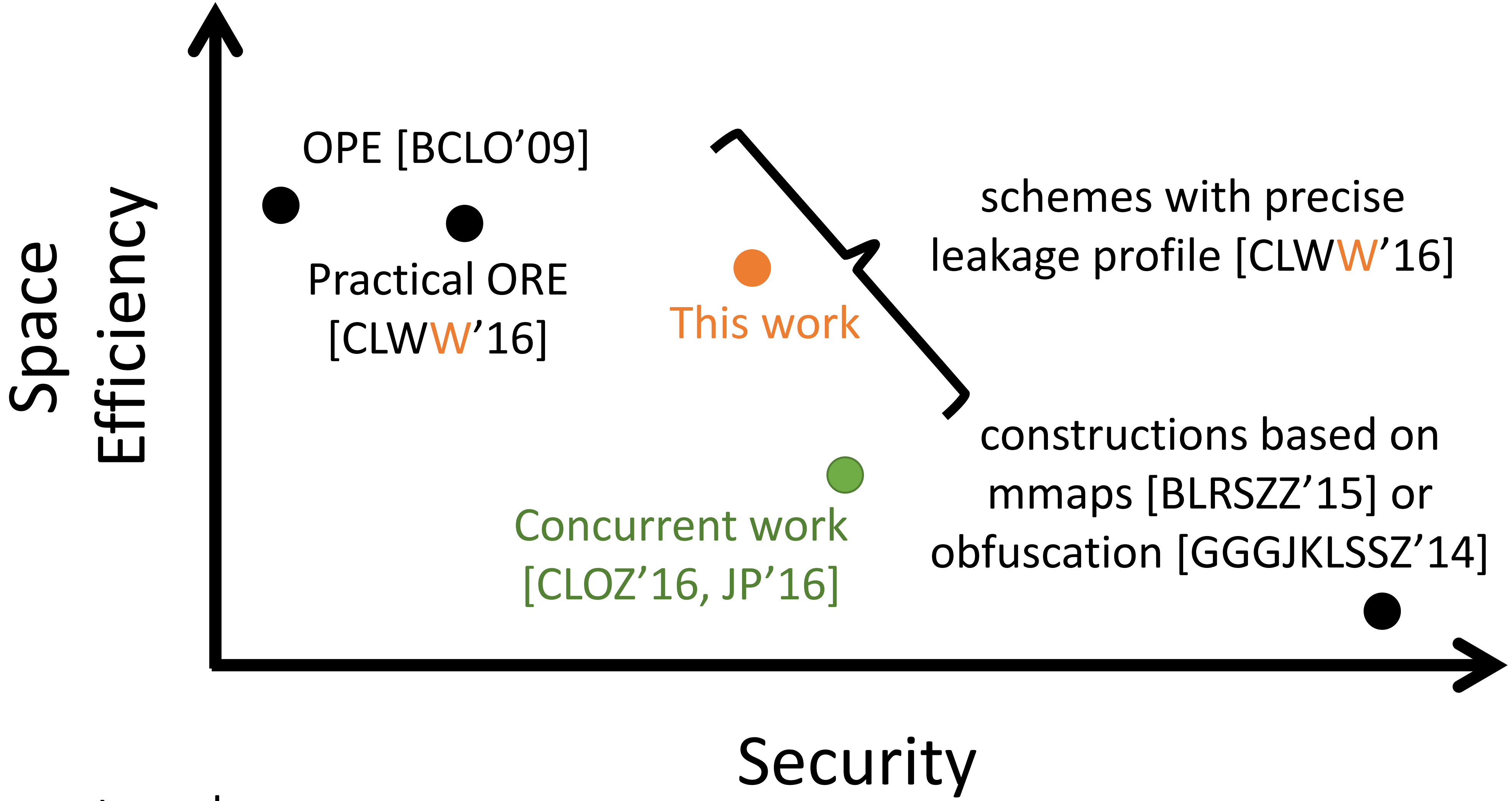
$\mathrm{ct}_2 = \mathrm{Enc}(\mathrm{sk}, y)$

$x > y$

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

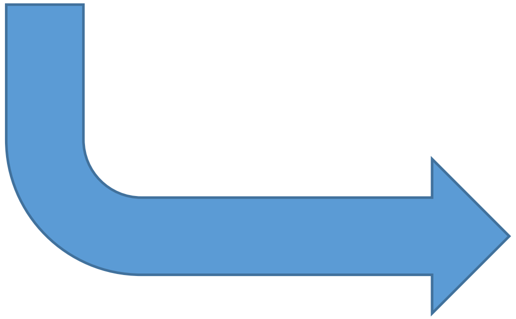
Inference Attacks [NKW’15, DDC’16, GSBNR’16]

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

encrypted database

+

public information

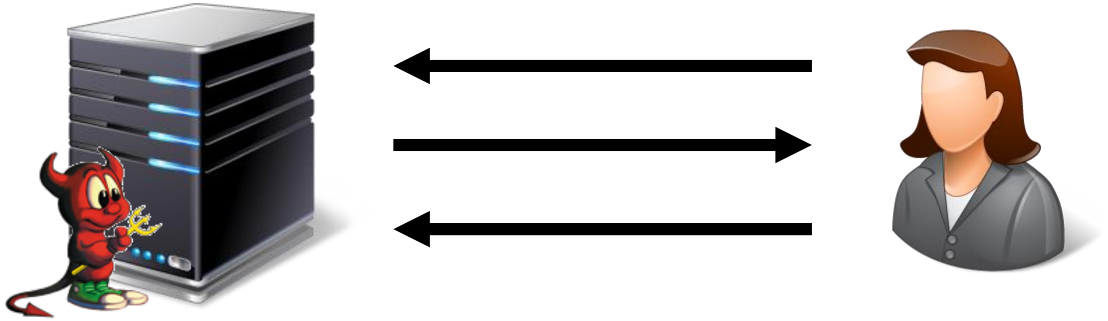


frequency and statistical analysis

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

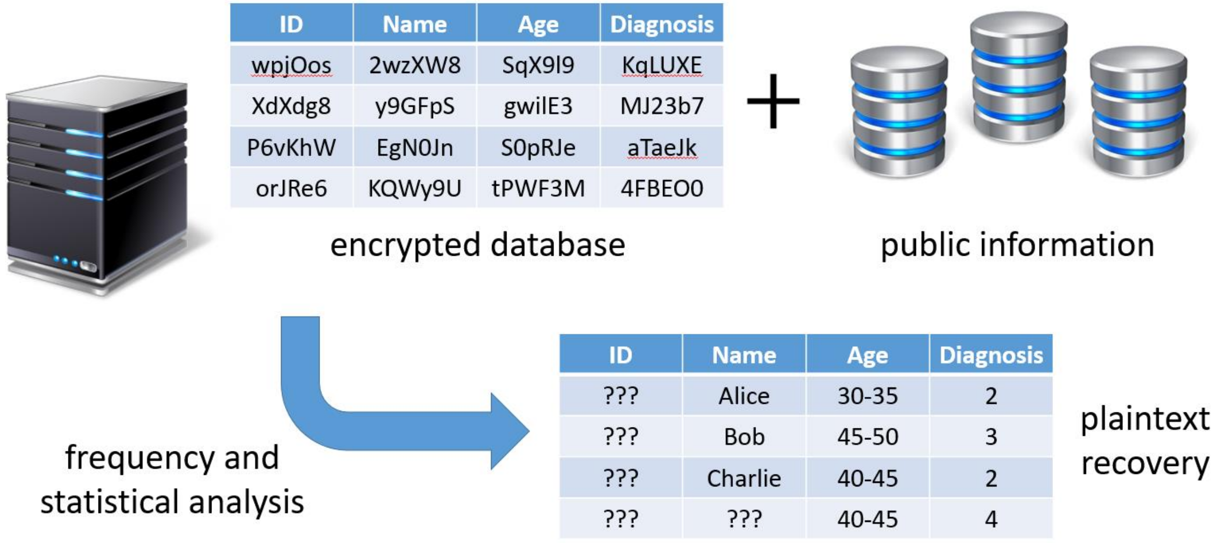
online attacks (e.g., active corruption)

offline attacks (e.g., passive snapshots)

adversary only sees contents of encrypted database

typical database breach: database contents 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

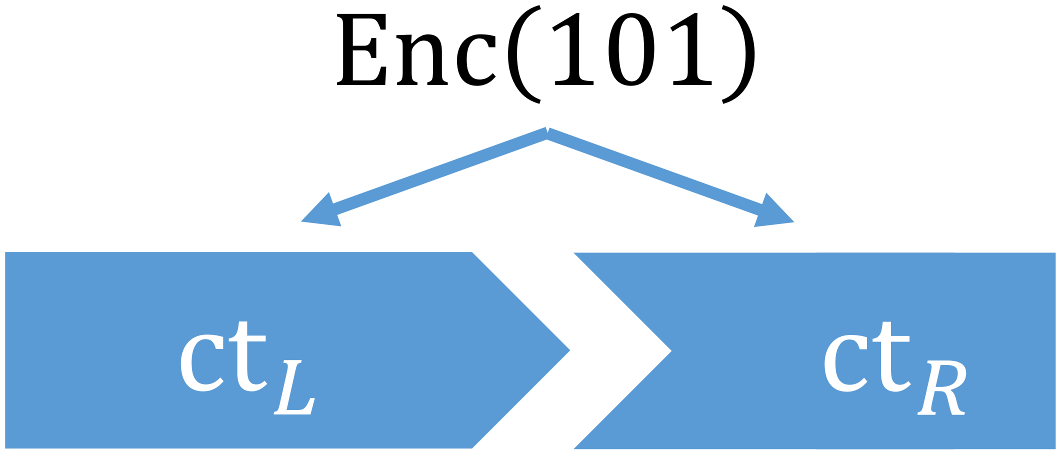
PPE schemes always reveal certain properties (e.g., equality, order) on ciphertexts and thus, are vulnerable to offline inference attacks

Can we obtain robustness against offline inference attacks while remaining legacy-friendly?

ORE with Additional Structure

Focus of this work: performing range queries on encrypted data

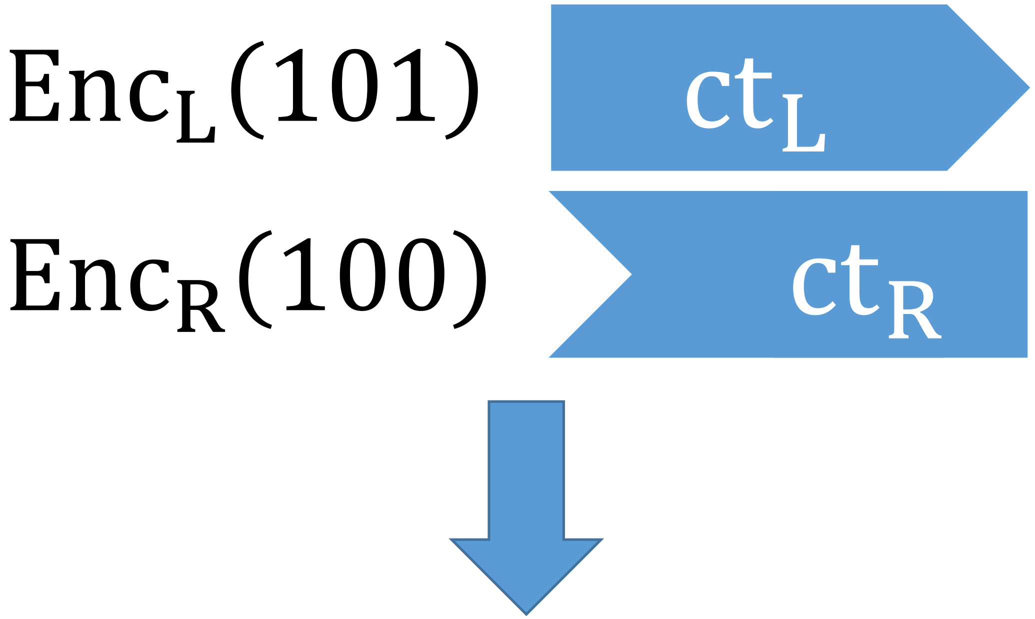
Key primitive: order-revealing encryption scheme where ciphertexts have a “decomposable” structure



$\mathrm{Enc}(101)$

ct_L ct_R

ciphertexts naturally split into two components



$\mathrm{Enc}_\mathrm{L}(101)$ ct_L

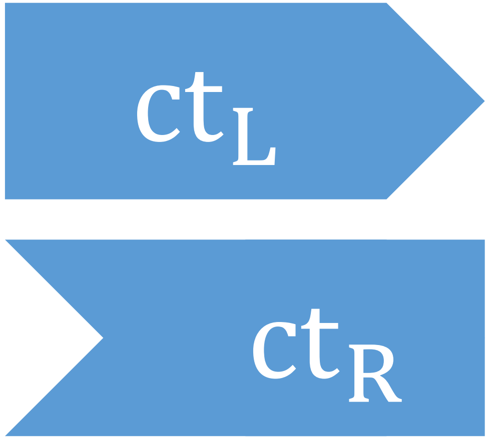
$\mathrm{Enc}_\mathrm{R}(100)$ ct_R

greater than

ORE with Additional Structure

$\mathrm{Enc}_{\mathrm{L}}(101)$

$\mathrm{Enc}_{\mathrm{R}}(100)$



comparison can be performed between left ciphertext and right ciphertext

right ciphertexts provide **semantic security**!

robustness against offline inference attacks!

Encrypted Range Queries

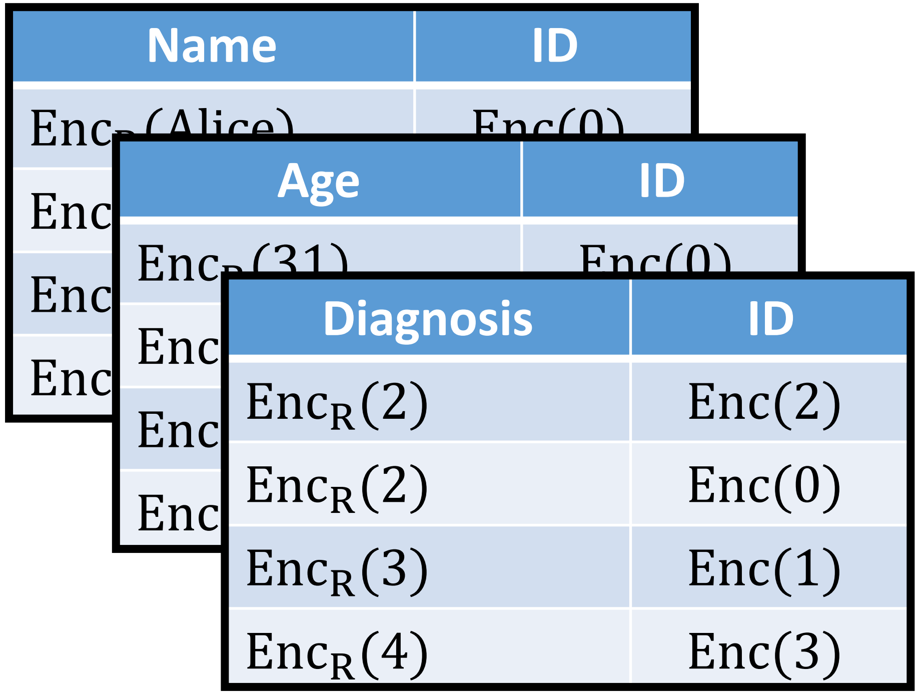
ID	Name	Age	Diagnosis
0	Alice	31	2
1	Bob	47	3
2	Charlie	41	2
3	Inigo	45	4

build encrypted index

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

store right ciphertexts in sorted order

record IDs encrypted under independent key



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

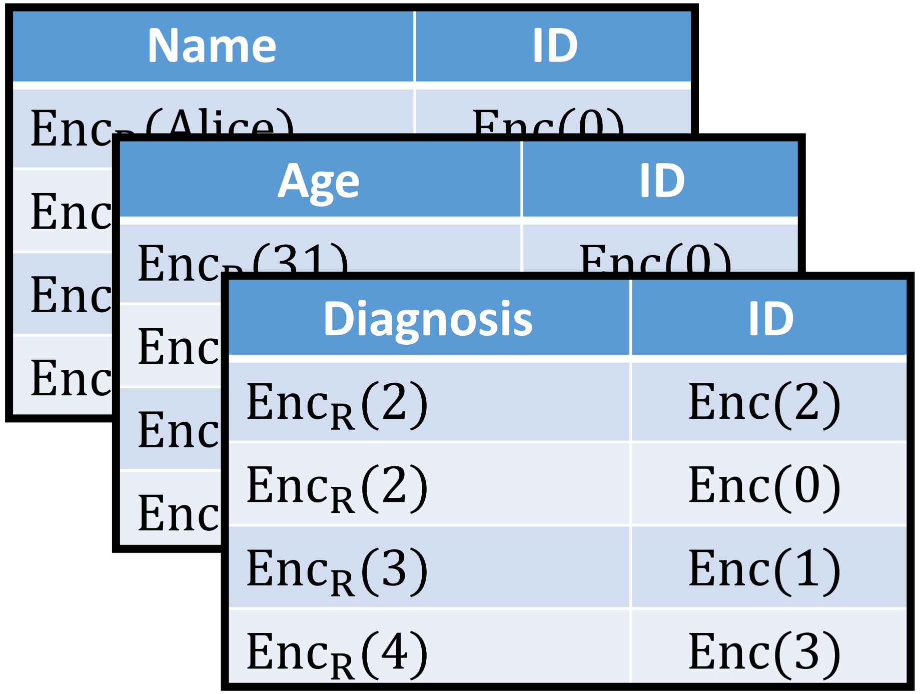
Encrypted Range Queries

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

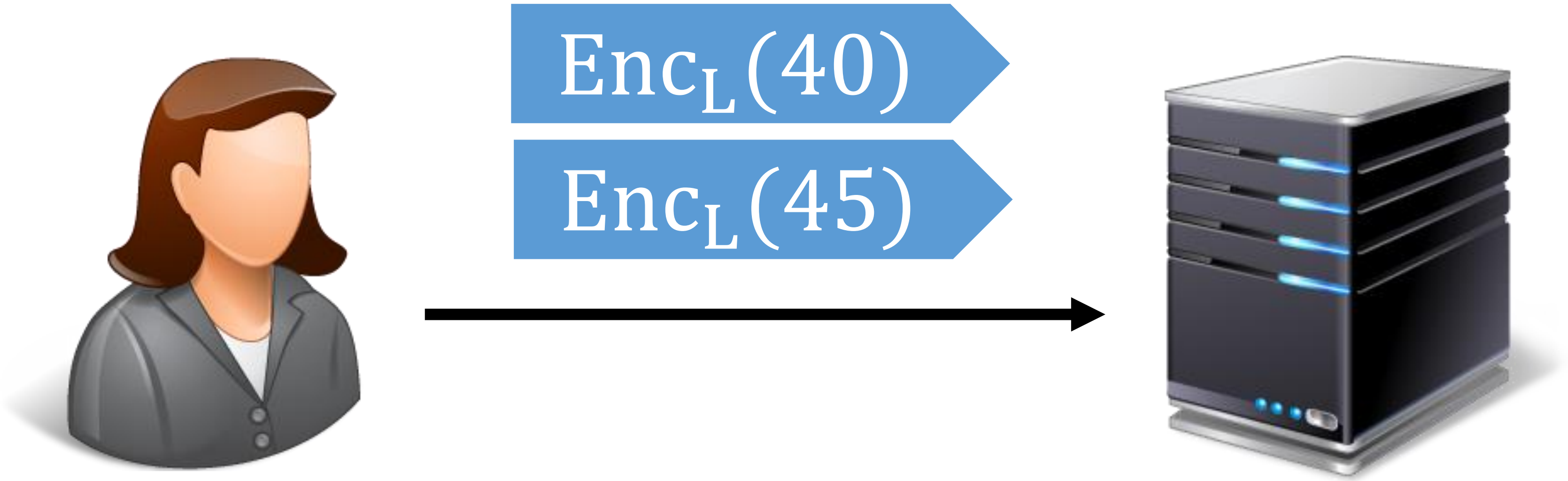

Name
ID
$\text{Enc}_\text{R}(\text{Alice})$
$\text{Enc}(0)$
Enc
Enc
Enc
Age
ID
$\text{Enc}_\text{R}(31)$
$\text{Enc}(0)$
Enc
Enc
Enc

Diagnosis	ID
$\text{Enc}_\text{R}(2)$	$\text{Enc}(2)$
$\text{Enc}_\text{R}(2)$	$\text{Enc}(0)$
$\text{Enc}_\text{R}(3)$	$\text{Enc}(1)$
$\text{Enc}_\text{R}(4)$	$\text{Enc}(3)$

encrypted search indices

Encrypted Range Queries

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



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

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

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 $40 \geq \text{age} \geq 45$:

$\text{Enc}_L(40)$

$\text{Enc}_L(45)$

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

use binary search to determine endpoints (comparison via ORE)

Encrypted Range Queries

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

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

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

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 $40 \geq \text{age} \geq 45$:

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

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

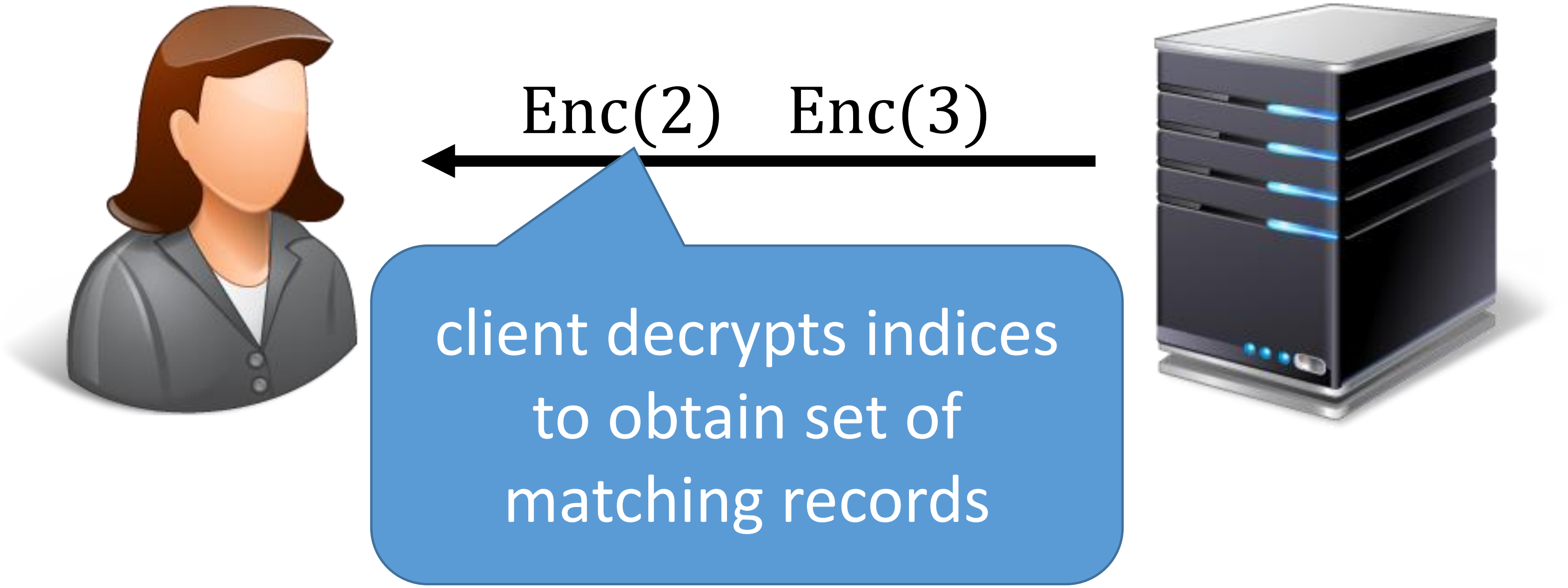
Age	ID
$\text{Enc}_\text{R}(31)$	$\text{Enc}(0)$
$\text{Enc}_\text{R}(41)$	$\text{Enc}(2)$
$\text{Enc}_\text{R}(45)$	$\text{Enc}(3)$
$\text{Enc}_\text{R}(47)$	$\text{Enc}(1)$

return encrypted indices that match query

use binary search to determine endpoints (comparison via ORE)

Encrypted Range Queries

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

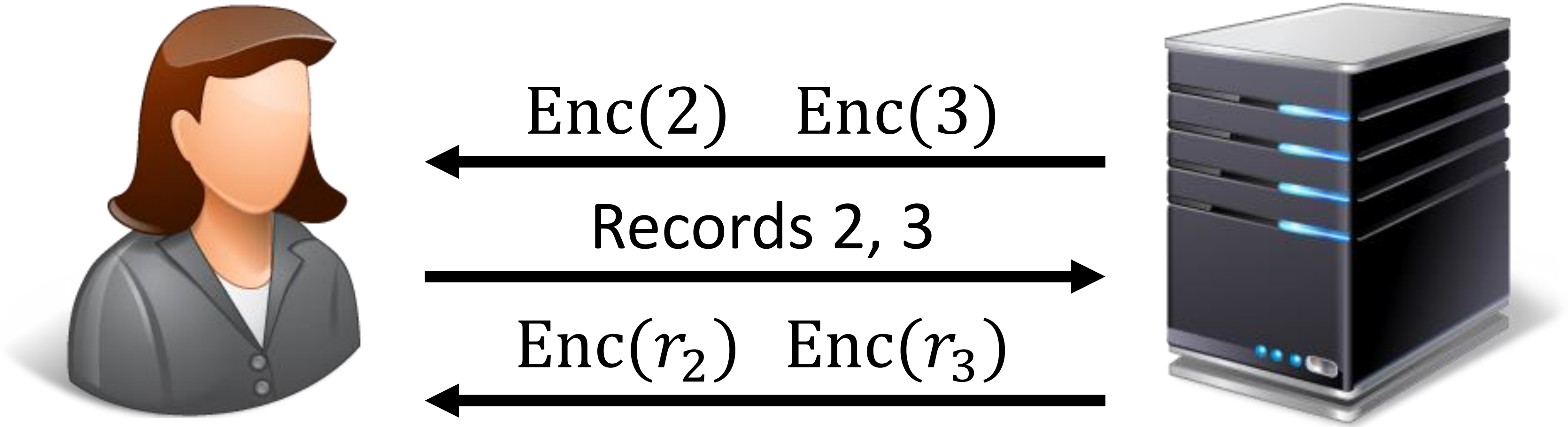


$\text{Enc}(2)$ $\text{Enc}(3)$

client decrypts indices to obtain set of matching records

Encrypted Range Queries

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



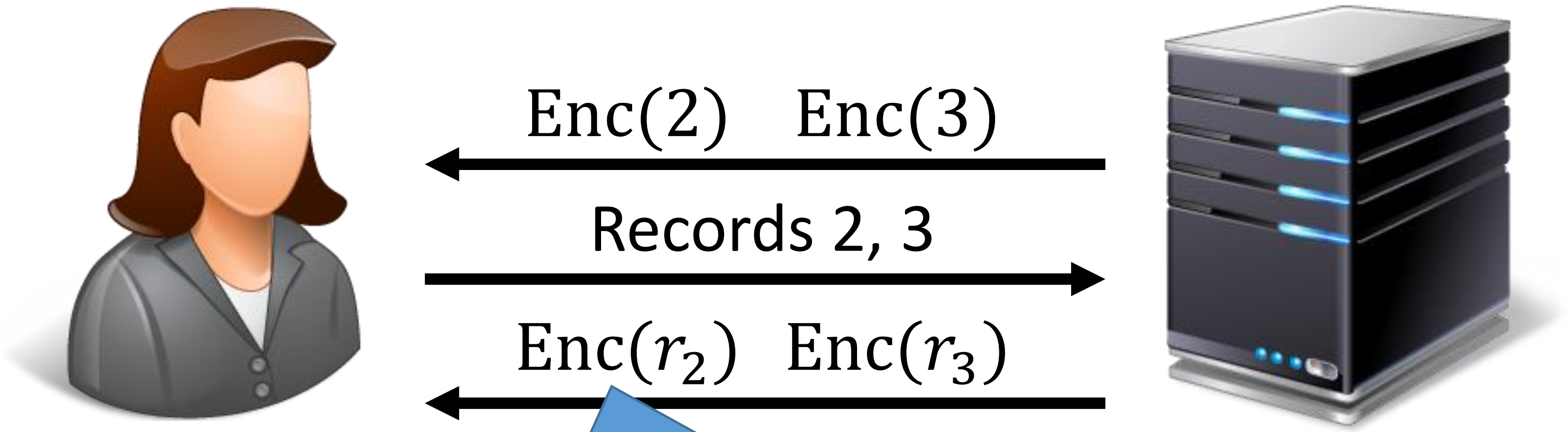
$\mathrm{Enc}(2) \quad \mathrm{Enc}(3)$

Records 2, 3

$\mathrm{Enc}(r_2) \quad \mathrm{Enc}(r_3)$

Encrypted Range Queries

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



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

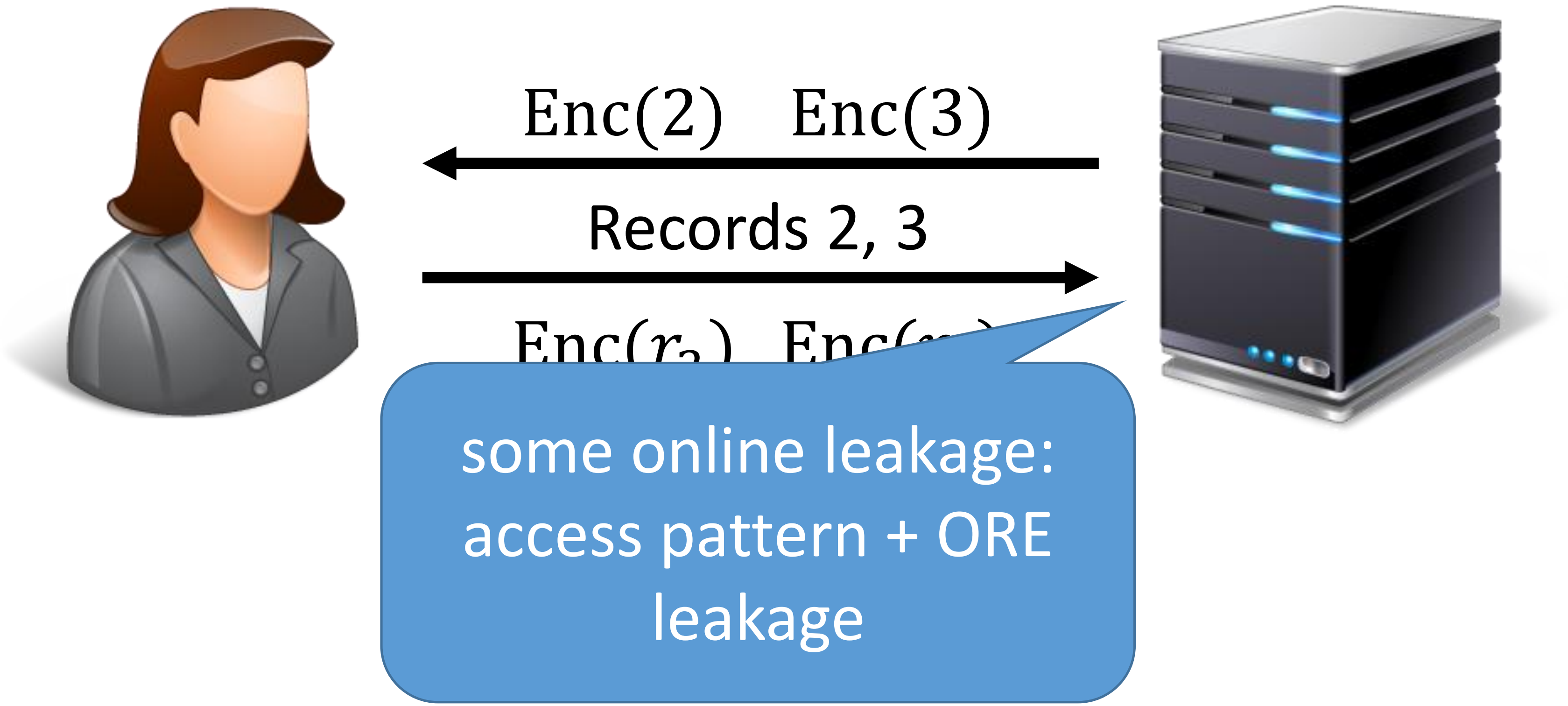
Records 2, 3

$\mathrm{Enc}(r_2)$ $\mathrm{Enc}(r_3)$

client decrypts to obtain records

Encrypted Range Queries

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



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

Records 2, 3

$\mathrm{Enc}(r_2) \quad \mathrm{Enc}(r_3)$

some online leakage: access pattern + ORE leakage

Note: trivial solution of just encrypting the index leaks everything in the online setting

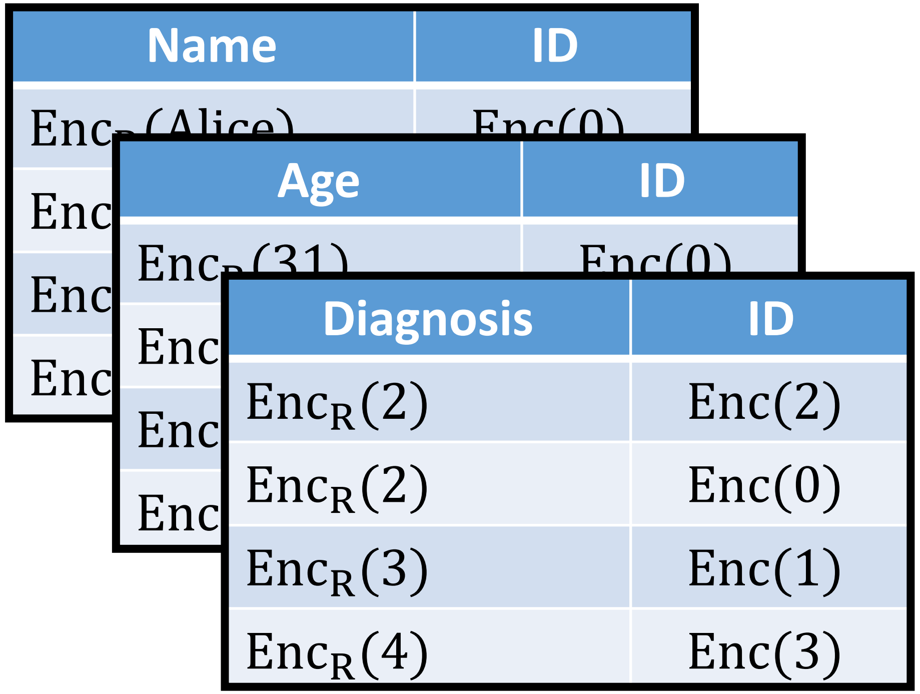
Encrypted Range Queries

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!

Perfect offline security



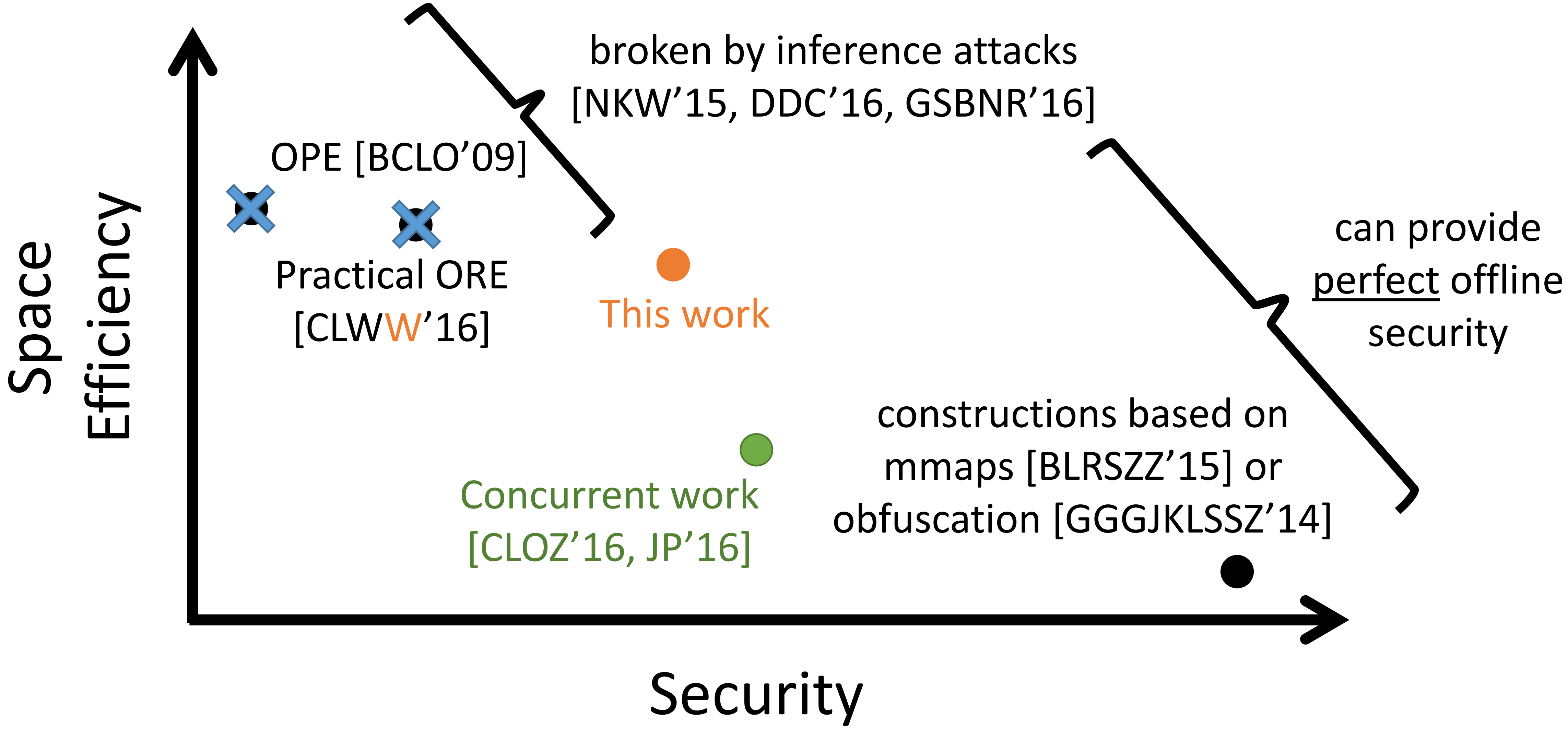
Name	ID
Enc_R(Alice)	Enc(0)
Enc	
Enc	
Enc	

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

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

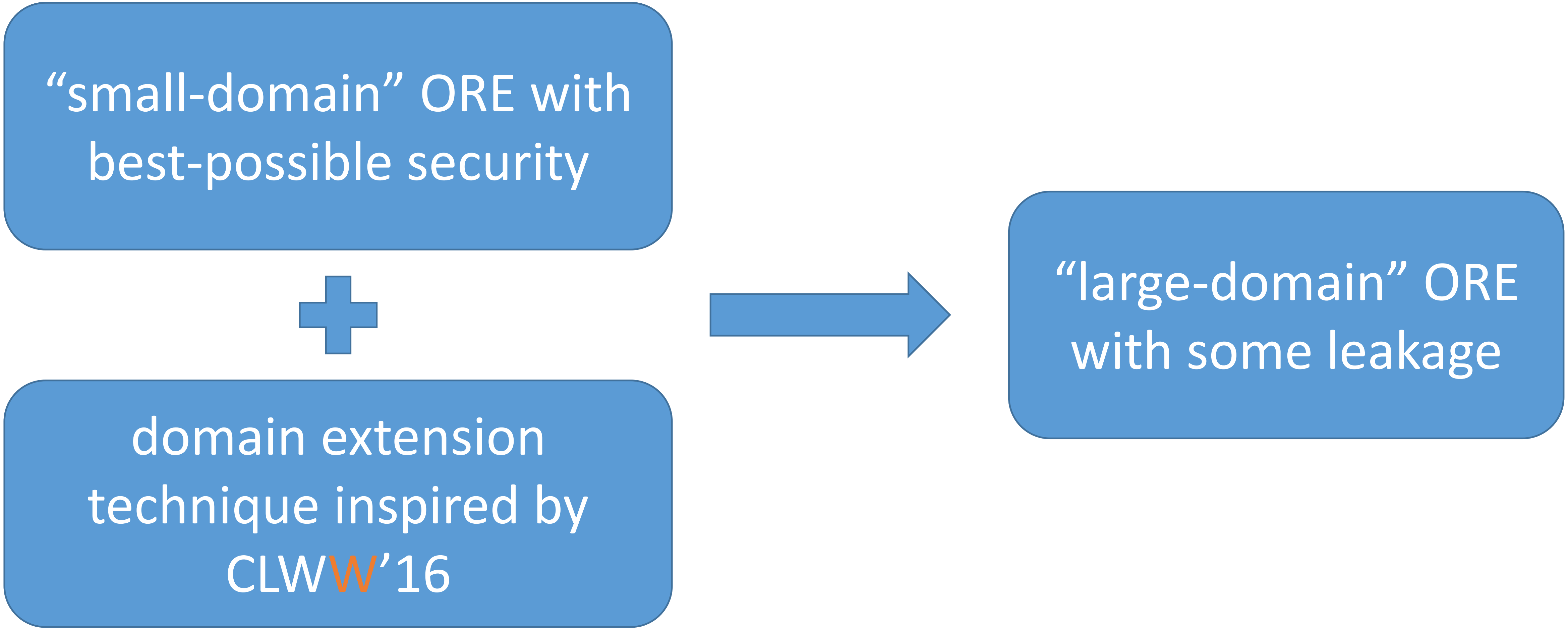
encrypted search indices

The Landscape of ORE



Not drawn to scale

Our New ORE Scheme



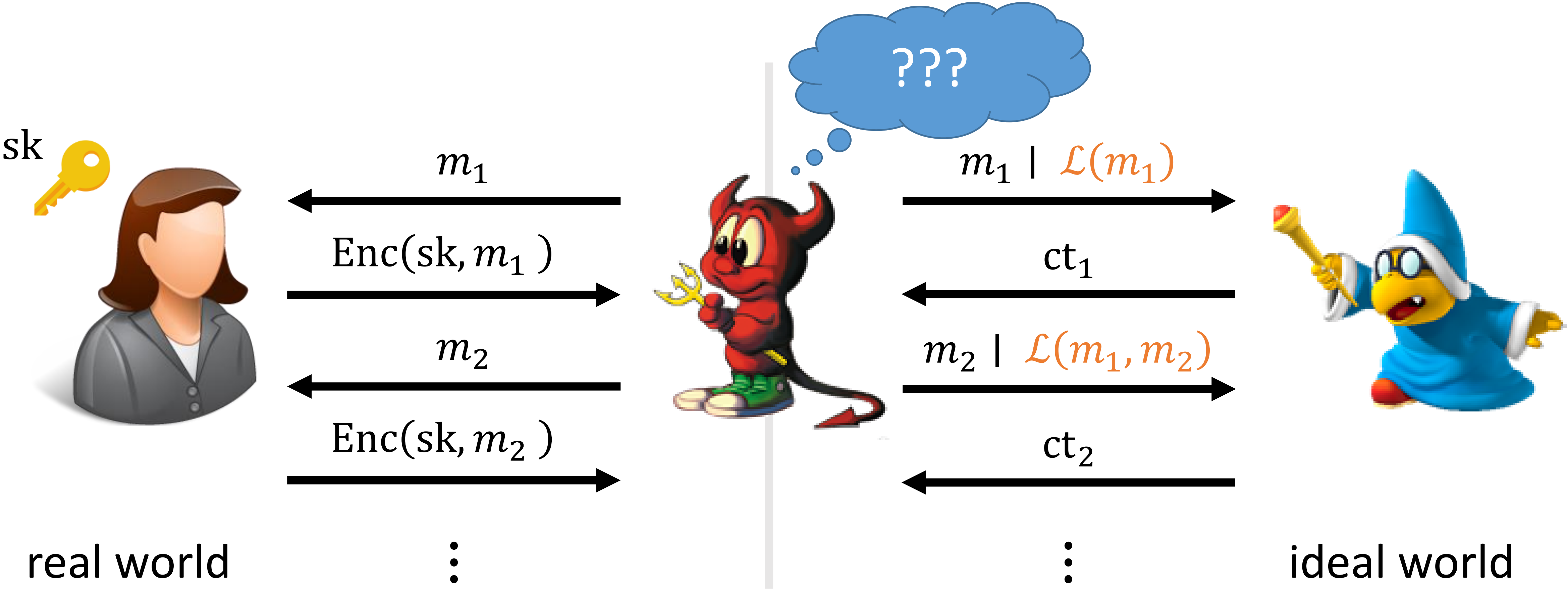
“small-domain” ORE with best-possible security

domain extension technique inspired by CLWW’16

“large-domain” ORE with some leakage

ORE with Leakage [CLWW'16]

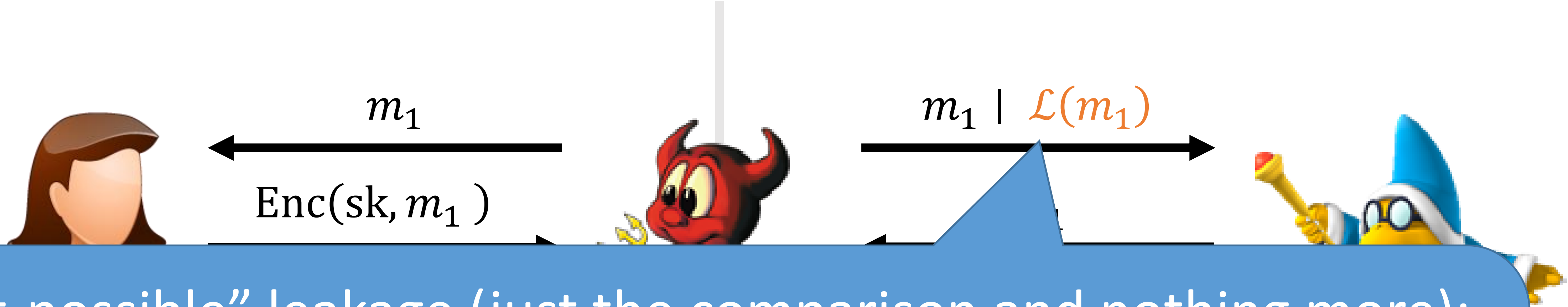
Model information leakage explicitly by a leakage function $\mathcal{L}$



???
sk
m_1
$\mathrm{Enc}(\mathrm{sk}, m_1)$
m_2
$\mathrm{Enc}(\mathrm{sk}, m_2)$
$m_1 \mid \mathcal{L}(m_1)$
ct_1
$m_2 \mid \mathcal{L}(m_1, m_2)$
ct_2
$\vdots$
$\vdots$
real world
ideal world

ORE with Leakage [CLWW'16]

Model information leakage explicitly by a leakage function $\mathcal{L}$



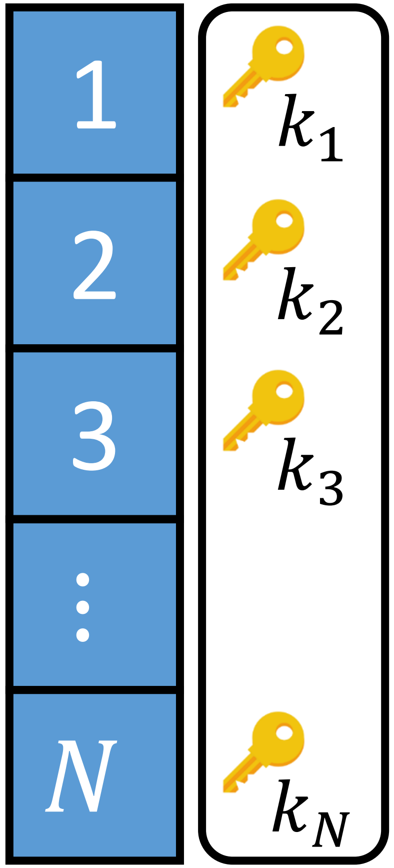
"Best-possible" leakage (just the comparison and nothing more):

$$\mathcal{L}(m_1, \ldots, m_q) = \{ (i, j, \mathbf{1}\{m_i < m_j\}) \mid 1 \leq i < j \leq q \}$$

Small-Domain ORE with Best-Possible Security

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

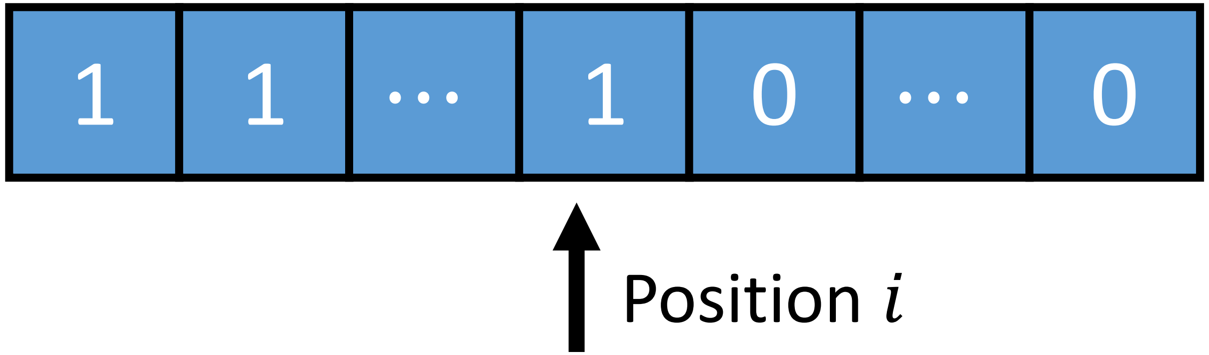
associate a key with each value



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

Small-Domain ORE with Best-Possible Security

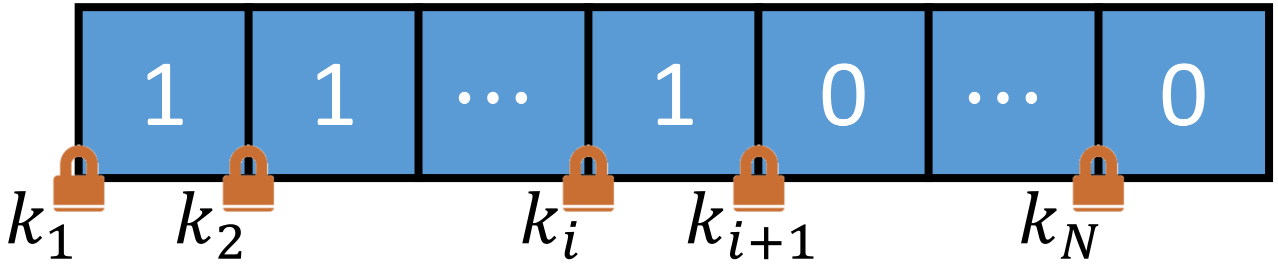
Encrypting a value i



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

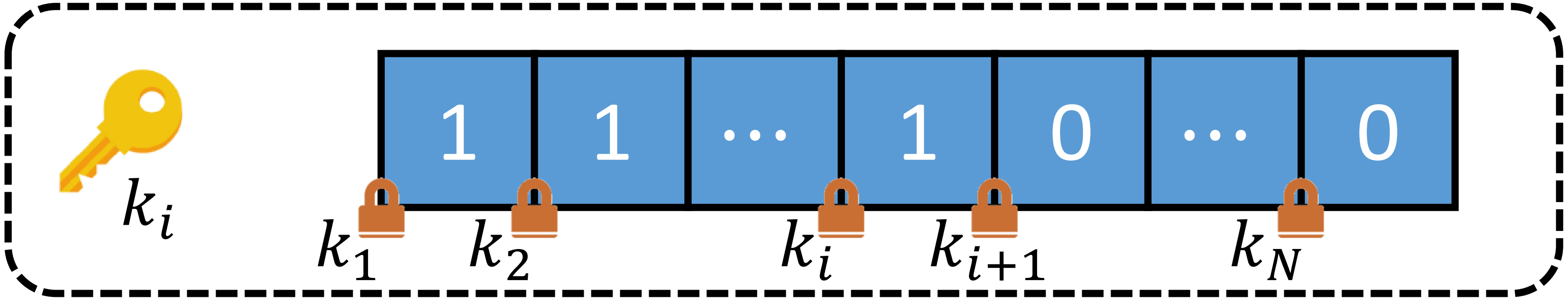
Small-Domain ORE with Best-Possible Security

Encrypting a value i



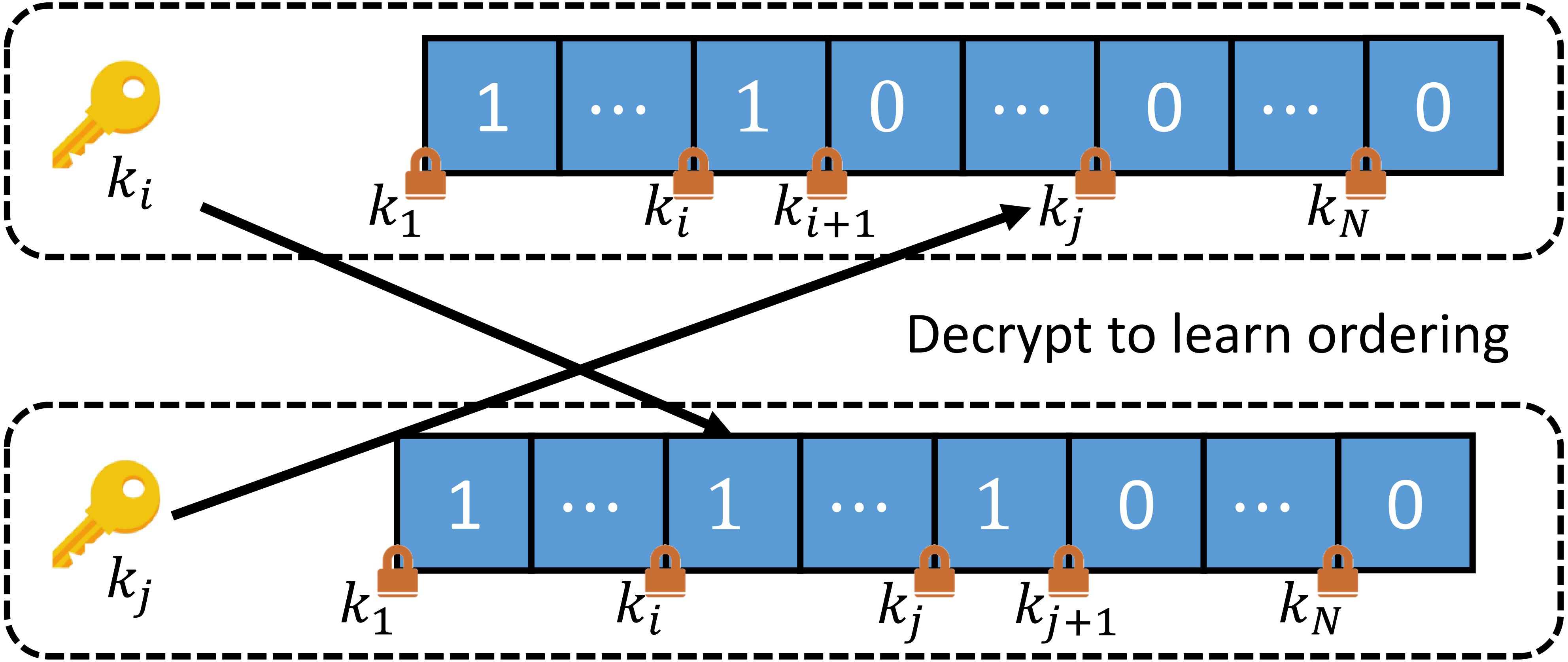
encrypt each slot with key for that slot

To allow comparisons, also give out key for slot i



Small-Domain ORE with Best-Possible Security

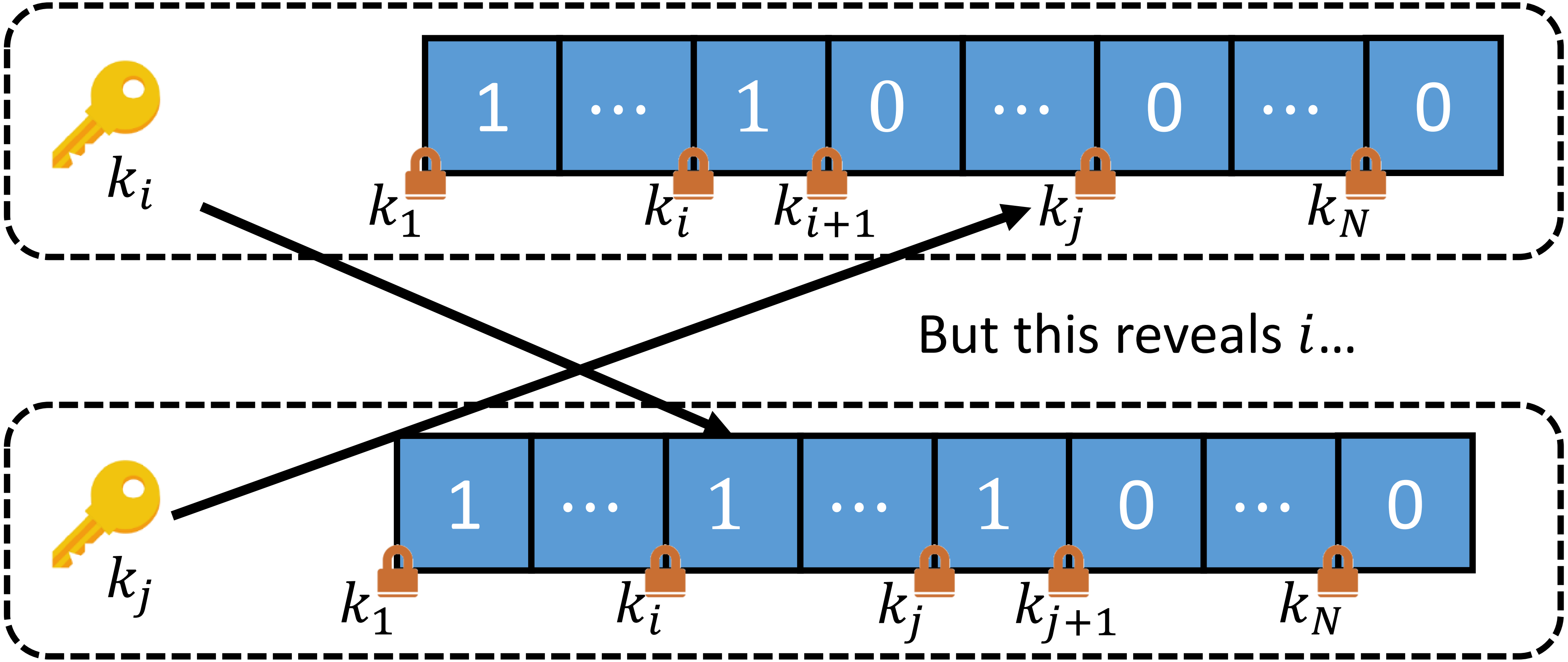
Given two ciphertexts



Decrypt to learn ordering

Small-Domain ORE with Best-Possible Security

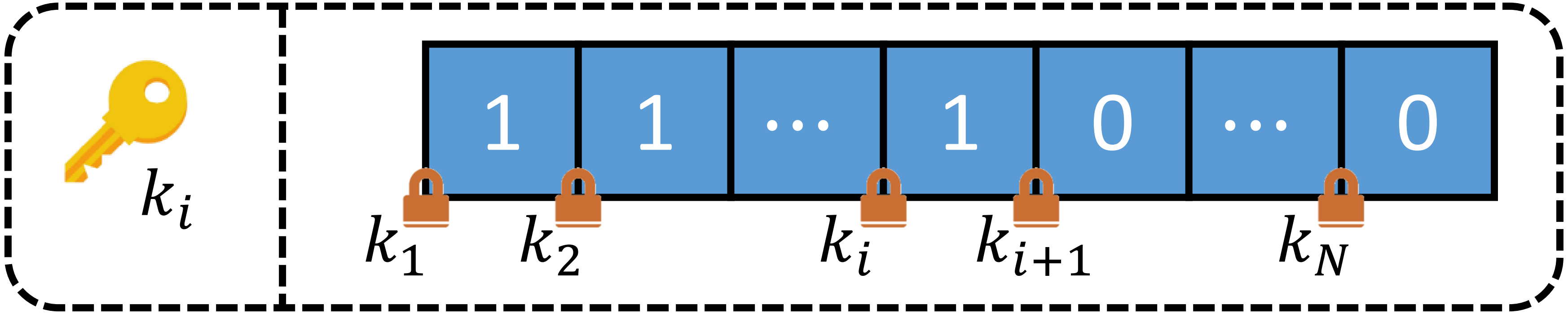
Given two ciphertexts



But this reveals i…

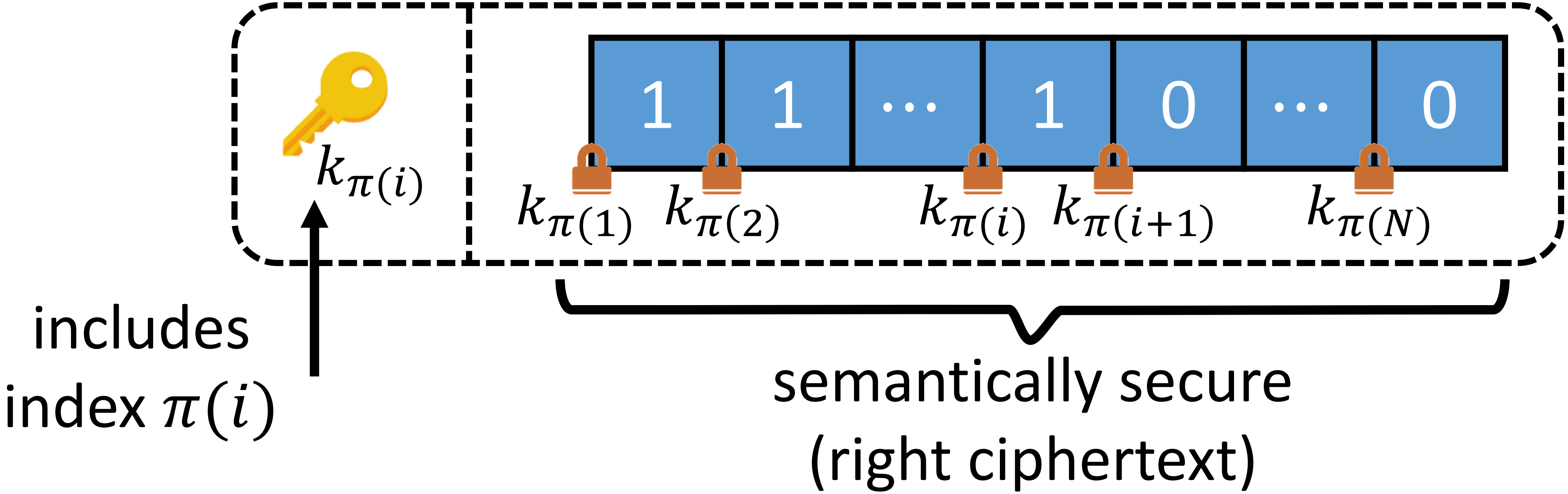
Small-Domain ORE with Best-Possible Security

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

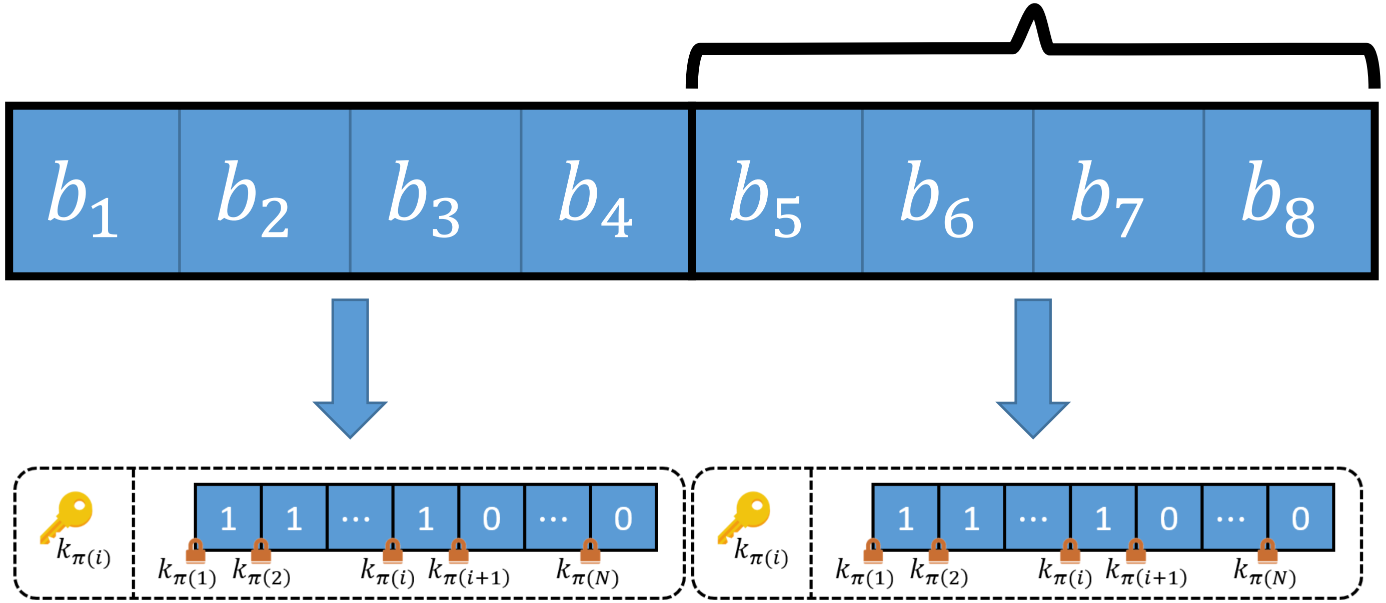


Achieves best-possible security, but ciphertexts are big

Domain Extension for ORE

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

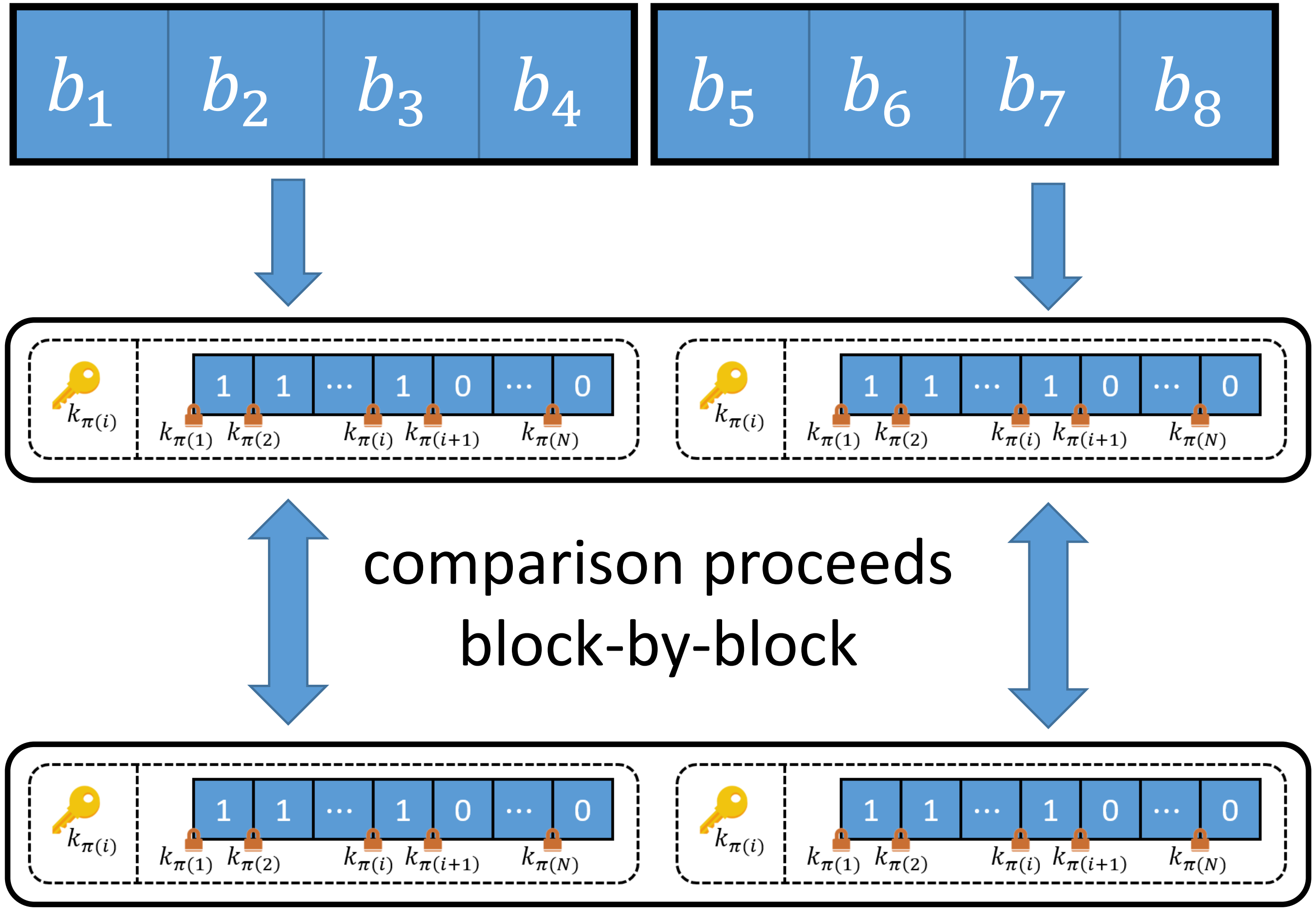
split into two 4-bit chunks



b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8
$k_{\pi(i)}$
1 1 ⋯ 1 0 ⋯ 0
$k_{\pi(1)}$ $k_{\pi(2)}$ $k_{\pi(i)}$ $k_{\pi(i+1)}$ $k_{\pi(N)}$
$k_{\pi(i)}$
1 1 ⋯ 1 0 ⋯ 0
$k_{\pi(1)}$ $k_{\pi(2)}$ $k_{\pi(i)}$ $k_{\pi(i+1)}$ $k_{\pi(N)}$

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

Domain Extension for ORE



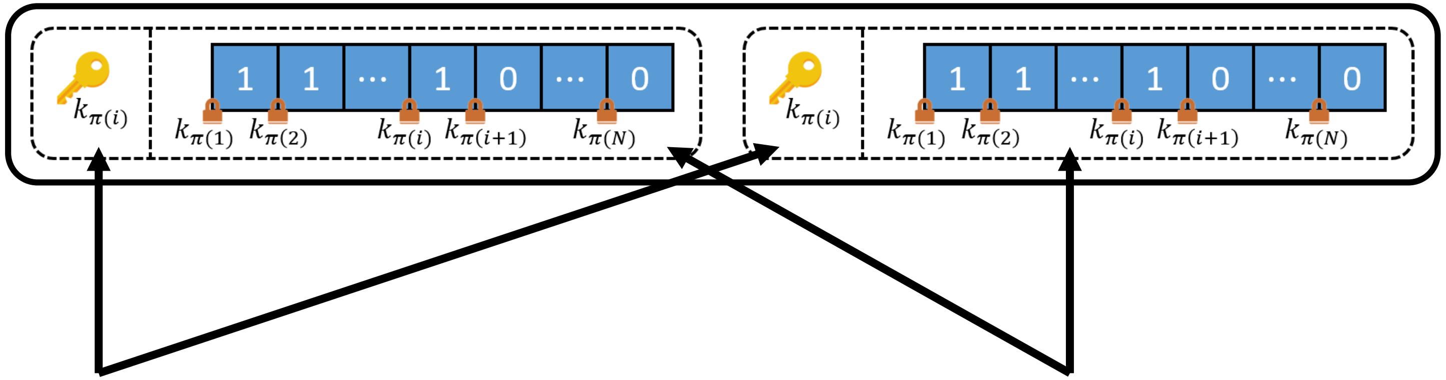
b_1
b_2
b_3
b_4
b_5
b_6
b_7
b_8
$k_{\pi(i)}$
1 1 ⋯ 1 0 ⋯ 0
$k_{\pi(1)}$ $k_{\pi(2)}$ $k_{\pi(i)}$ $k_{\pi(i+1)}$ $k_{\pi(N)}$

comparison proceeds block-by-block

Overall leakage: first **block** that differs

Domain Extension for ORE

Same decomposition into left and right ciphertexts:



$k_{\pi(i)}$
1 1 ⋯ 1 0 ⋯ 0
$k_{\pi(1)}$ $k_{\pi(2)}$ $k_{\pi(i)}$ $k_{\pi(i+1)}$ $k_{\pi(N)}$
$k_{\pi(i)}$
1 1 ⋯ 1 0 ⋯ 0
$k_{\pi(1)}$ $k_{\pi(2)}$ $k_{\pi(i)}$ $k_{\pi(i+1)}$ $k_{\pi(N)}$

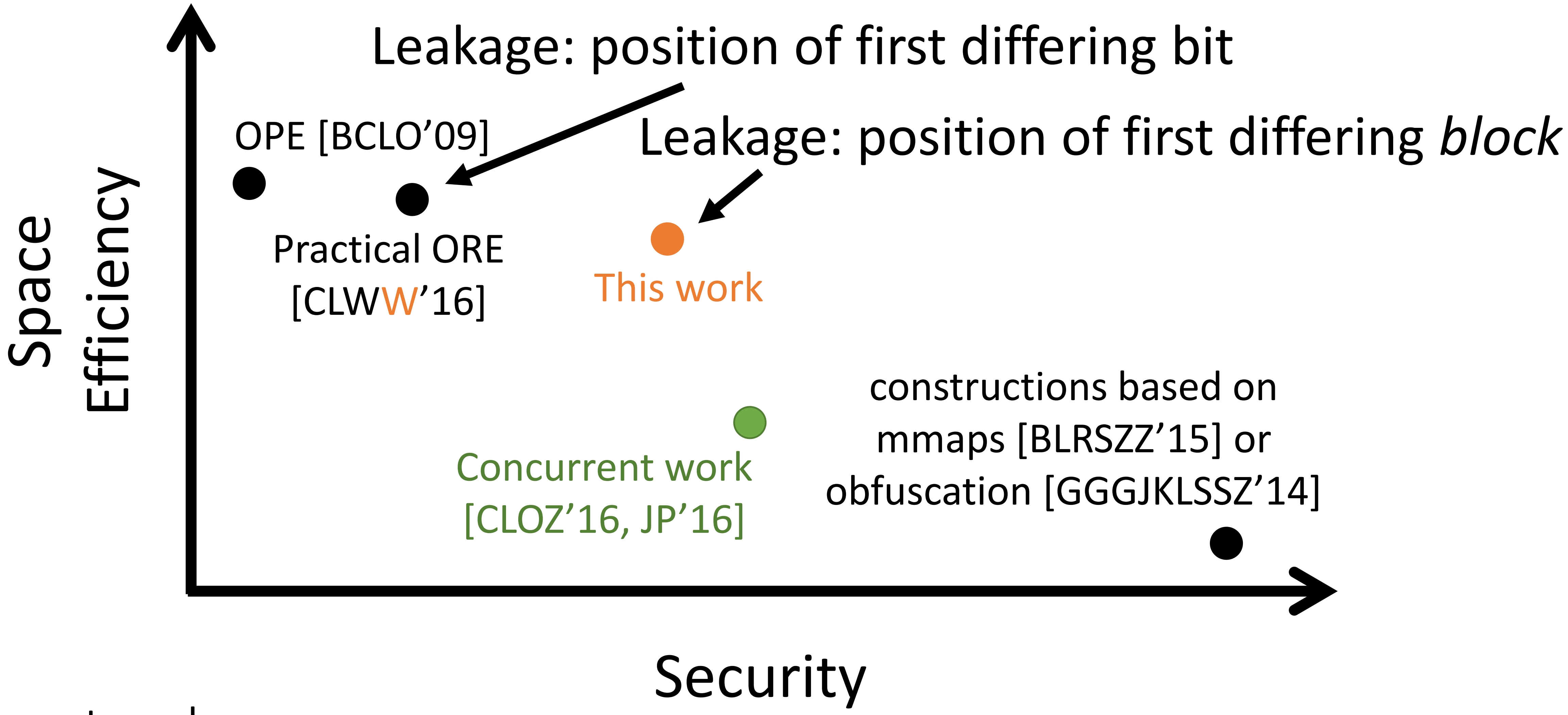
left ciphertext

right ciphertext

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



Leakage: position of first differing bit

Leakage: position of first differing *block*

OPE [BCLO’09]

Practical ORE [CLWW’16]

This work

Concurrent work [CLOZ’16, JP’16]

constructions based on mmaps [BLRSZZ’15] or obfuscation [GGGJKLSSZ’14]

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.

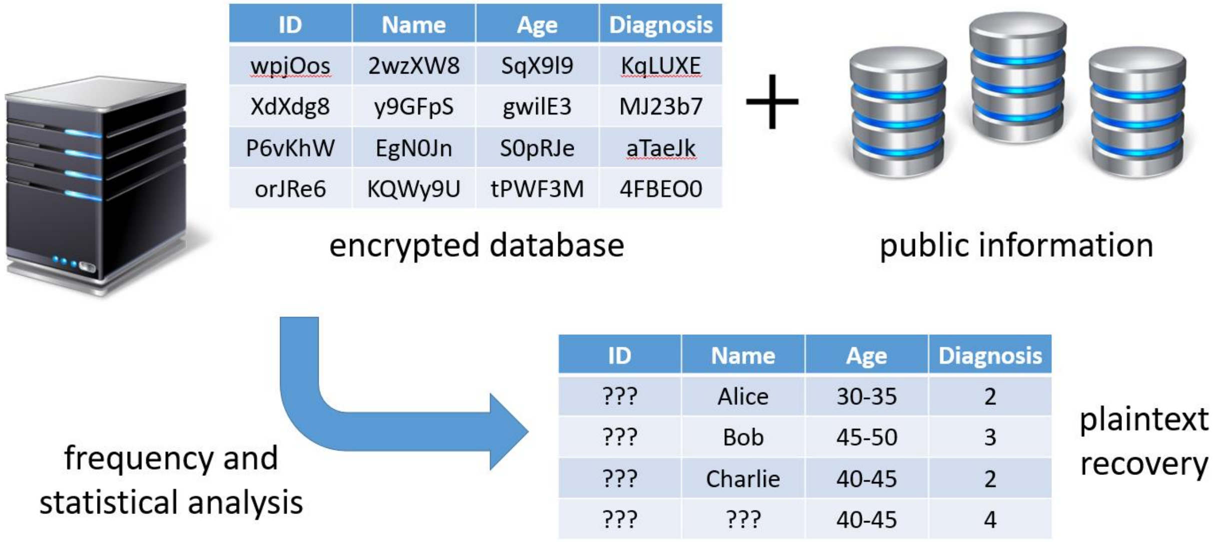
Performance Evaluation

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] 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 legacy-compatible and provides offline semantic security
- New ORE construction that is concretely efficient with strong security
- In paper: new impossibility results for security achievable using OPE

Open Problems

- What kind of inference attacks on possible in the online setting?
    - Indices encrypted separately, so multi-column correlations harder to infer
    - More limited leakage profile (between left and right ciphertexts)
- Can we construct small-domain OREs (with best-possible security) and *sublinear* ciphertext size from PRFs?
- Can we construct left/right ORE (from PRFs) where both left and right ciphertexts are *semantically secure*?

Questions?

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