Pien

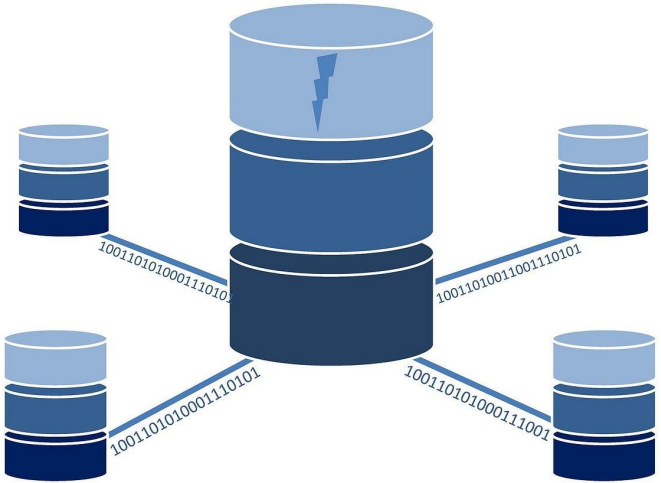
Searching in encrypted databases

Pien van den Abeele	s1044362
Dirk Doesburg	s1040211
Ties Speel	s1020150

Radboud University

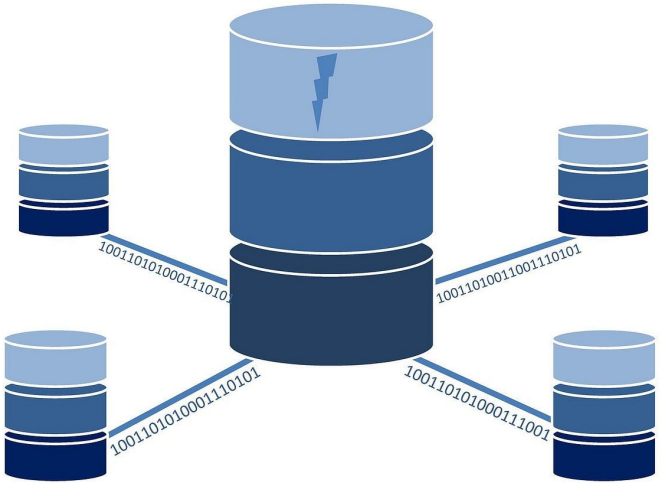
Encrypted databases

Why encrypt?



Encrypted databases

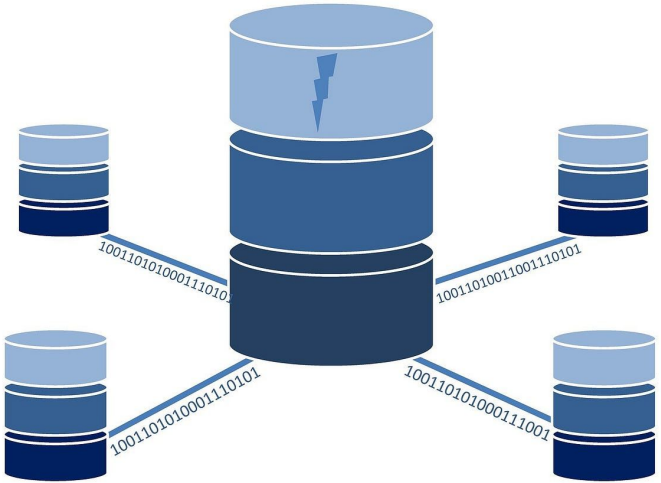
- Third parties



Encrypted databases

- Third parties

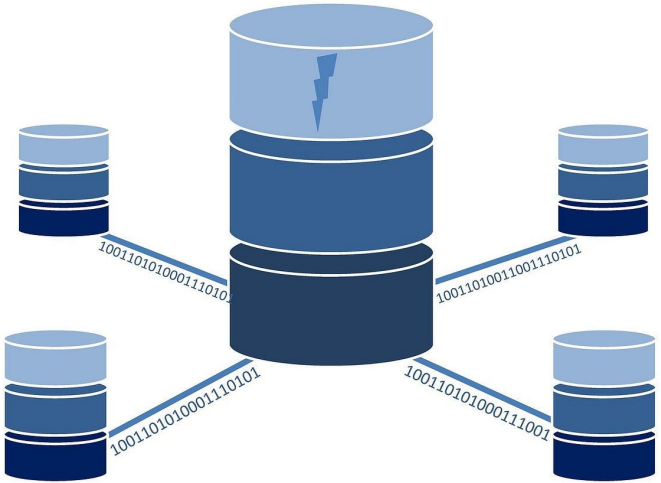
  Databases



Encrypted databases

- Third parties

  Databases
  Partners



Encrypted databases

- Third parties

  Databases
  Partners

- Examples

  Hospital
  - Threats (e.g. unauthorised access, physical theft)

Encrypted databases

- Third parties

  Databases
  Partners

- Examples

  Hospital
  Law-enforcement
    - Threats (e.g. cyberattacks, insider threats)

Encrypted databases

- Third parties

  Databases
  Partners

- Examples

  Hospital
  Law-enforcement
  E-Commerce
    - Threats (e.g. cyberattacks, data leaks)

Back in time - Nowadays

- Secure Data Exchange System (SDES), 1976
    - Symmetric key encryption
    - Key management techniques

Back in time - Nowadays

- Secure Data Exchange System (SDES), 1976
    - Symmetric key encryption
    - Key management techniques
- Nowadays
    - Searchable encryption
    - Secure multi-party computation
        - Homomorphic encryption

GDPR on databases

- Storage and Processing of Encrypted Personal Data

- Right to Access

- Data Breach Notification

- Data Protection Impact Assessment (DPIA)

Problems?

- Searching in encrypted databases is expensive and time-consuming.

- Key management

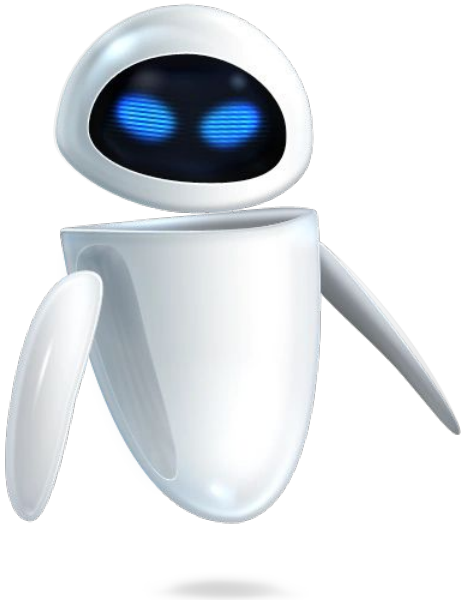
- Privacy vs efficiency & costs
    - Law-Enforcement?
    - Bol.com?
    - Health care?

Ties

Setting

Alice

Eve



Bob?

Defining goal features

Defining goal features

Secrecy

Eve cannot learn anything about the plaintext when given only the ciphertext.

Defining goal features

Secrecy

Eve cannot learn anything about the plaintext when given only the ciphertext.

Query isolation

When Alice searches for some query, Eve cannot learn about anything other than the search result.

Defining goal features

Secrecy

Eve cannot learn anything about the plaintext when given only the ciphertext.

Query isolation

When Alice searches for some query, Eve cannot learn about anything other than the search result.

Controlled searching

Eve cannot search for an arbitrary query herself, she needs Alice to be able to search for something.

Defining goal features

Secrecy

Eve cannot learn anything about the plaintext when given only the ciphertext.

Query isolation

When Alice searches for some query, Eve cannot learn about anything other than the search result.

Controlled searching

Eve cannot search for an arbitrary query herself, she needs Alice to be able to search for something.

Hidden queries

Alice can let Eve search for a query, without exposing the query (or the content of the matching parts of the plaintext) to Eve.

Baby's first scheme

Split into words $W_1 | \ldots | W_L$
Pick key k

Can encrypt words $C_i = W_i \oplus k$ **to get** $C_1 | \ldots | C_L$

Searching:
Compute C_q **for query** W_q **and send** C_q **to Eve.**

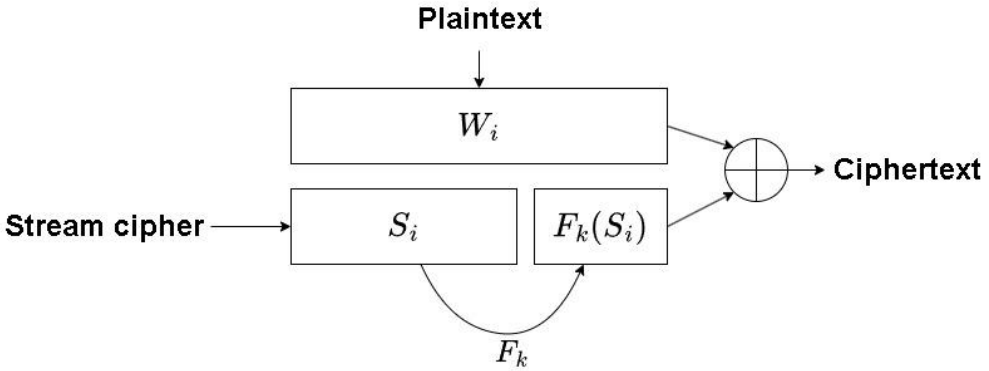
Can also use multiple different keys for search spaces

Source: https://imgur.com/F8yBdTI

Song et al. 2000

Scheme 2

Goal: create a scheme that includes randomness for proper secrecy
Take pseudorandom generator G and generate L random values $S_1 | \ldots | S_L$ (with masterkey)
Create tags $T_i = S_i || F_k(S_i)$ where F is a trapdoor function e.g. keyed hash function
$C_i = W_i \oplus T_i$

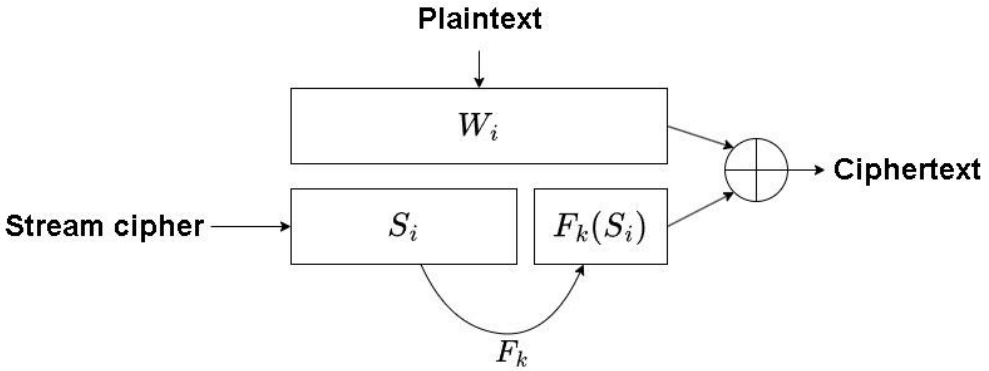


Song et al. 2000

Scheme 2

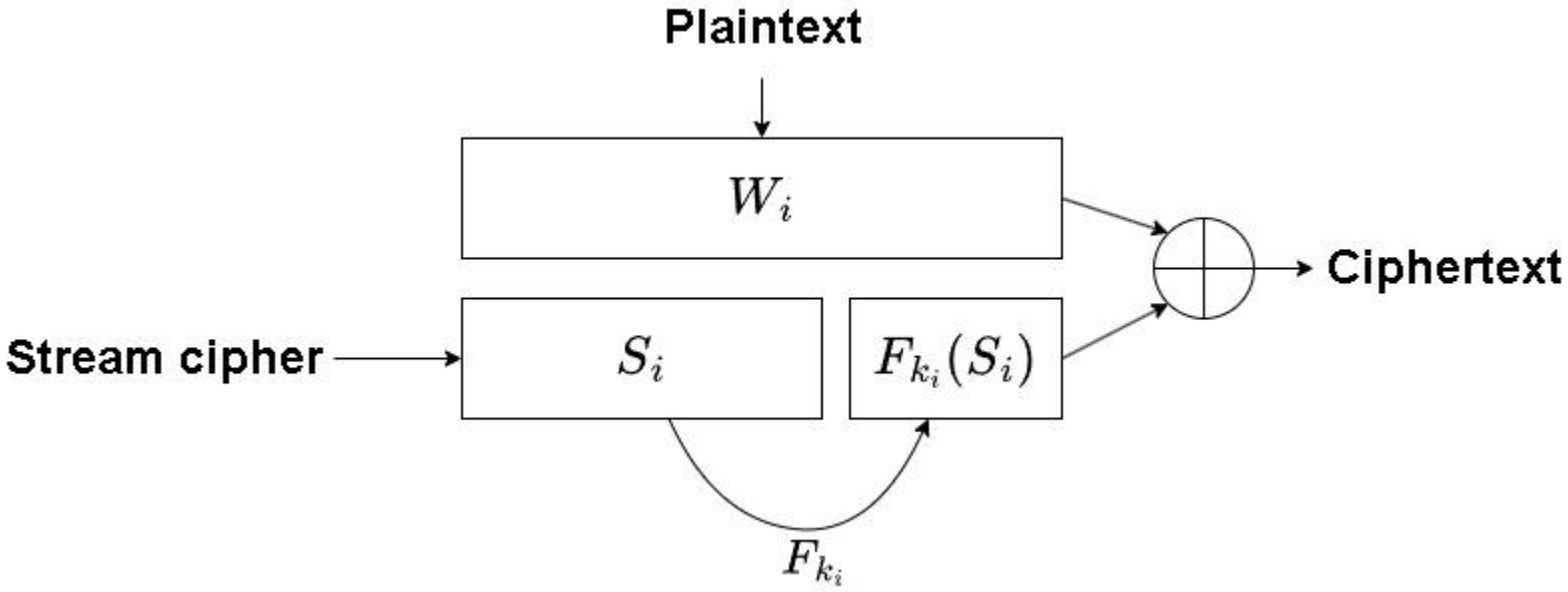
Goal: create a scheme that includes randomness for proper secrecy
Take pseudorandom generator G and generate L random values $S_1 | \ldots | S_L$ (with masterkey)
Create tags $T_i = S_i || F_k(S_i)$ where F is a trapdoor function e.g. keyed hash function
$C_i = W_i \oplus T_i$

Searching:
Give Eve k and query word W_q ->
To decrypt compute $C_i \oplus W_q$ -> Get tag x||y
$W_i == W_q$ if $F_k(x) == y$
Eve cannot learn W_i if $W_i != W_q$



Song et al. 2000

Scheme 2



Song et al. 2000

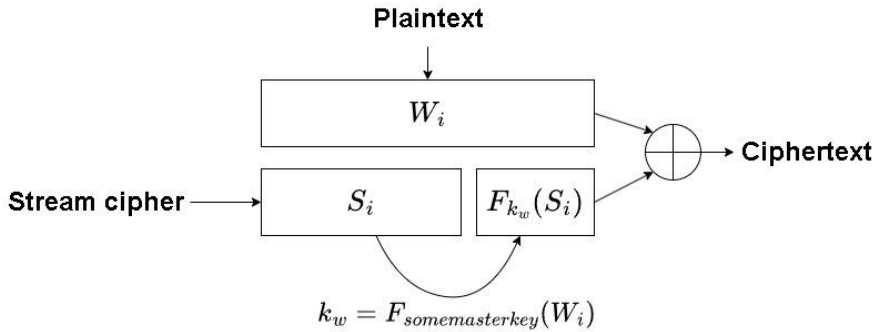
Scheme 3

Goals: We don't want to give our keys to Eve

Take pseudorandom generator G and generate L random values $S_1 | \ldots | S_L$ (with masterkey)

Create tags $T_i = S_i || F_{kw}(S_i)$ where $k_w = F_{somemasterkey}(W_i)$

$C_i = W_i \oplus T_i$

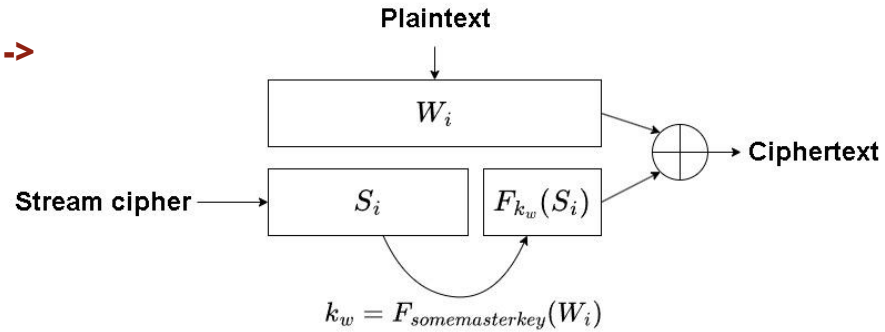


Song et al. 2000

Scheme 3

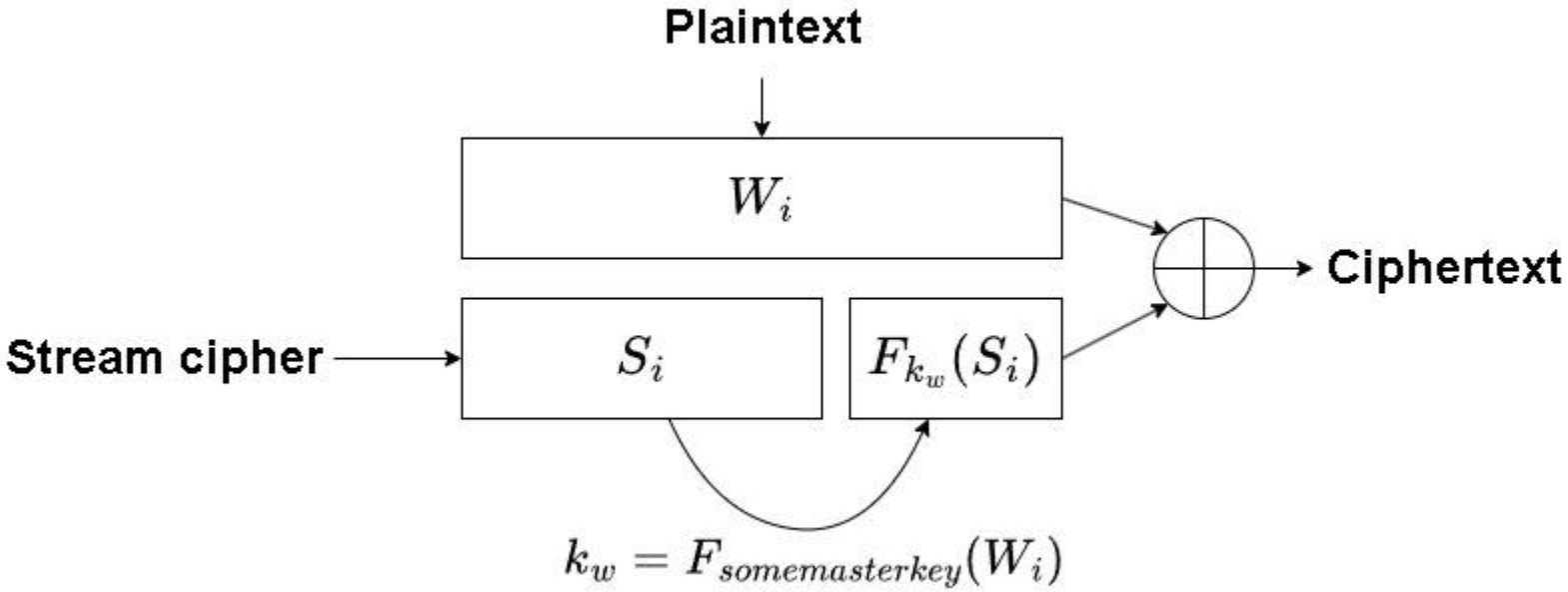
Goals: We don't want to give our keys to Eve
Take pseudorandom generator G and generate L random values $S_1 | \ldots | S_L$ (with masterkey)
Create tags $T_i = S_i || F_{kw}(S_i)$ where $k_w = F_{somemasterkey}(W_i)$
$C_i = W_i \oplus T_i$

Searching:
Give Eve query word W_q and corresponding key k_w ->
To decrypt compute $C_i \oplus W_q$ -> Get tag $x||y$
$W_i == W_q$ if $F_{kw}(x) == y$
Eve cannot search for W_q without knowing k_w



Song et al. 2000

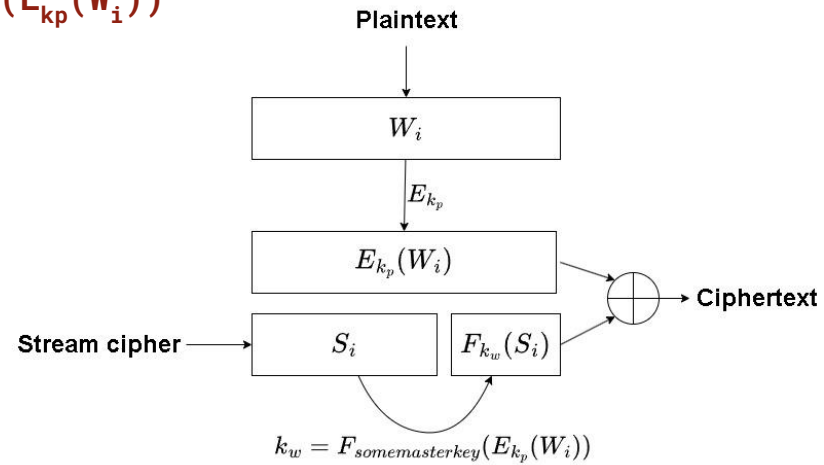
Scheme 3



Song et al. 2000

Scheme 4

Goal: We want to search for a word W_q without revealing it to Eve
We can pre-encrypt the words with an extra key k_p
Words are encapsulated in a layer of ECB -> $E_{kp}(W_i)$
Take pseudorandom generator G and generate L random values $S_1 | \ldots | S_L$ (with masterkey)
Create tags $T_i = S_i || F_{kw}(S_i)$ where $k_w = F_{somemasterkey}(E_{kp}(W_i))$
$C_i = E_{kp}(W_i) \oplus T_i$

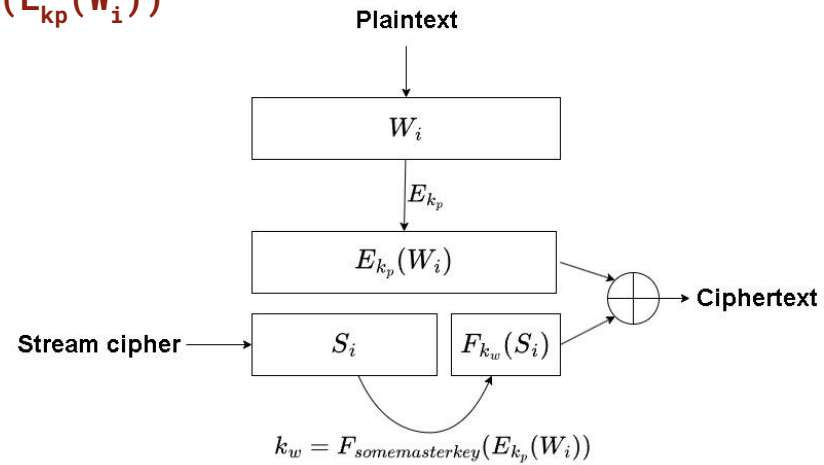


Song et al. 2000

Scheme 4

Goal: We want to search for a word W_q without revealing it to Eve
We can pre-encrypt the words with an extra key k_p
Words are encapsulated in a layer of ECB -> $E_{kp}(W_i)$
Take pseudorandom generator G and generate L random values $S_1 | \ldots | S_L$ (with masterkey)
Create tags $T_i = S_i || F_{kw}(S_i)$ where $k_w = F_{somemasterkey}(E_{kp}(W_i))$
$C_i = E_{kp}(W_i) \oplus T_i$

Searching:
Give Eve query word $E_{kp}(W_q)$ corresponding key k_w ->
To decrypt compute $C_i \oplus E_{kp}(W_q)$ -> Get tag $x||y$
$W_i == W_q$ **if** $F_{kw}(x) == y$
Eve cannot see W_q



Song et al. 2000

Oops… we've added too much privacy!

We have accidentally made our scheme so secure that even with all the keys, Alice can no longer decrypt anything

$C_i = E_{kp}(W_i) \oplus S_i || F_{kw}(S_i)$

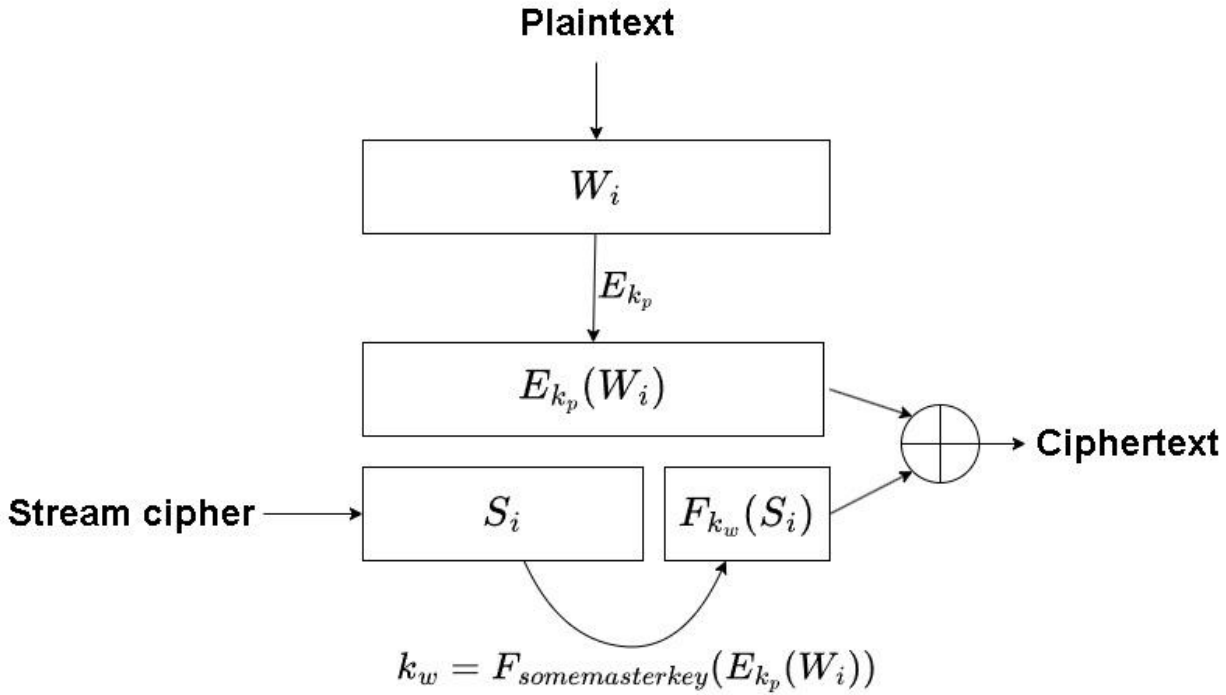
Keys k_w are generated as $F_{somemasterkey}(E_{kp}(W_i))$

This means to decrypt, Alice needs to know $E_{kp}(W_i)$

Alice needs to know the plaintext to decrypt the ciphertext!

Song et al. 2000

Scheme 4



Song et al. 2000

Final scheme

Need to decrypt $C_i = E_{kp}(W_i) \oplus S_i || F_{kw}(S_i)$
Need to know the word W_i...

Split $E_{kp}(W_i)$ into two halves L_i and R_i and switch $k_w = F_{somemasterkey}(E_{kp}(W_i))$
with $k_w = F_{somemasterkey}(L_i)$

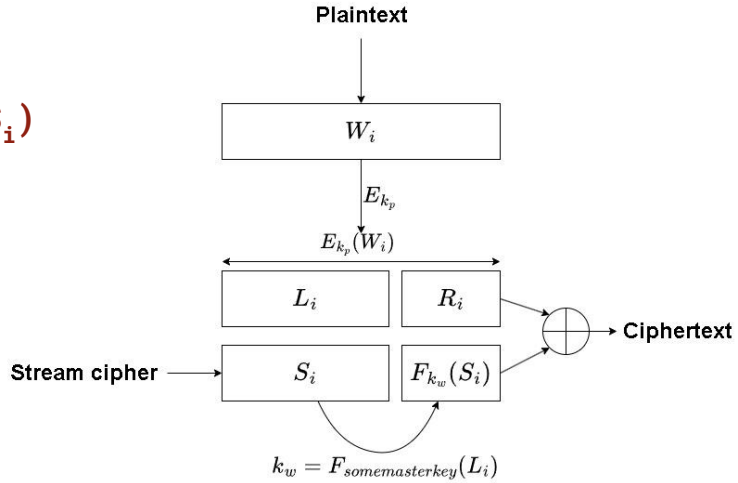
Song et al. 2000

Final scheme

Need to decrypt $C_i = E_{kp}(W_i) \oplus S_i || F_{kw}(S_i)$
Need to know the word W_i...

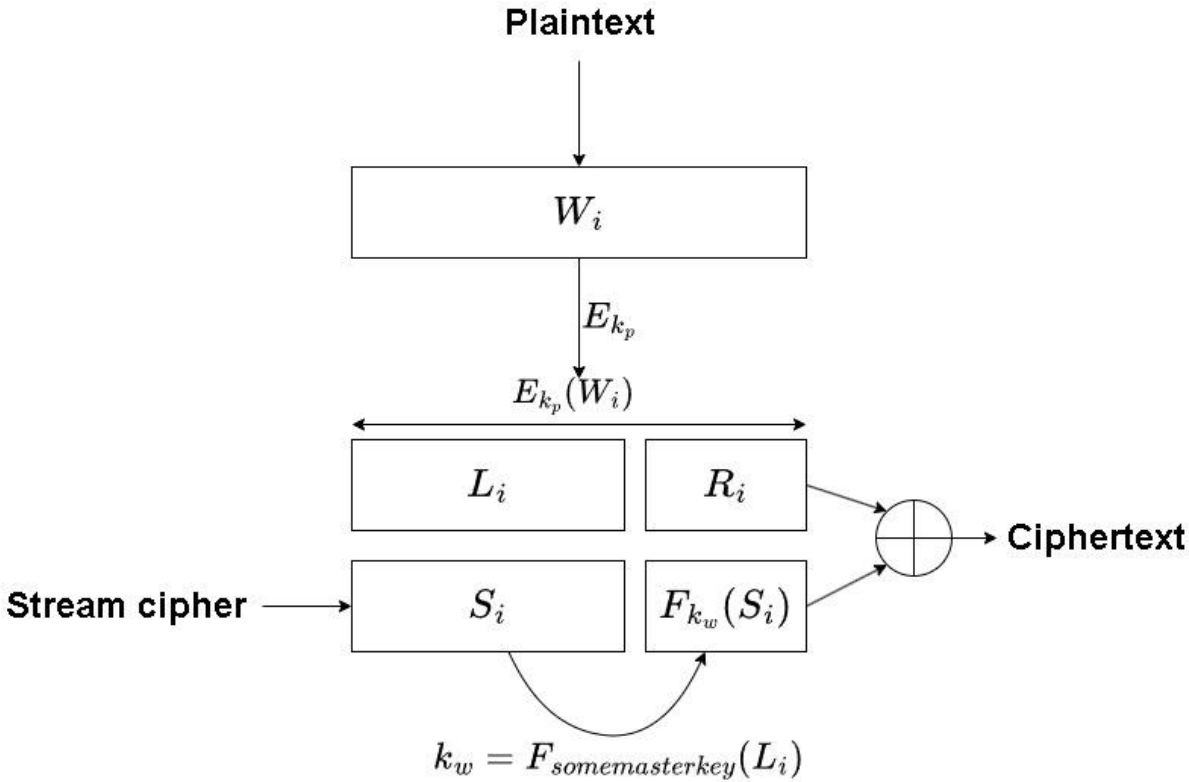
Split $E_{kp}(W_i)$ into two halves L_i and R_i and switch $k_w = F_{somemasterkey}(E_{kp}(W_i))$
with $k_w = F_{somemasterkey}(L_i)$

Alice knows S_i -> Can compute L_i from C_i
Can use L_i to compute k_w and compute tag $T_i = S_i || F_{kw}(S_i)$
Now Alice can compute $E_{kp}(W_i)$ and can decrypt.



Song et al. 2000

Final scheme



Pien

Symmetric Searchable Encryption

- Symmetric-key cryptography

- Three parties:
    - Data owner
    - Search engine
    - User

SSE - Deterministic encryption

- One cipher key

- A plain text gives one encrypted text

- Encrypted query compared to encrypted text

SSE - Probabilistic encryption

- Multiple cipher keys

- Same plain text gives multiple encrypted texts

- Probabilistic search method

- Encrypted query compared to different encrypted texts

PSSE vs DSSE

- DSSE:
    - Less complex than PSSE
    - Less secure than PSSE
    - No false positives

- PSSE:
    - More complex than DSSE
    - More secure than DSSE
    - May generate false positives

Public-key Encryption with Keyword Search

- SSE but with Public-key cryptography

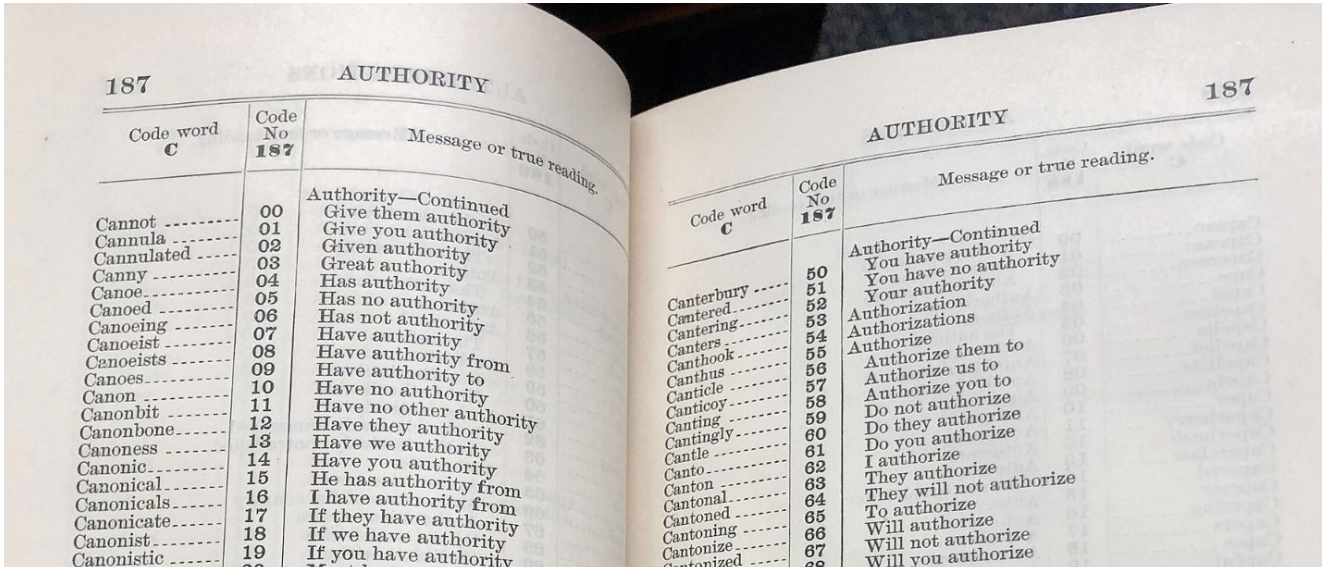
- Three parties:
    - Data owner
    - Search engine
    - User

- Trapdoor used

Dirk

Order-Preserving Encryption

- It's a new field (proposed in 2004)
- But has been used in WW I: one-part codebooks, e.g.:



187 AUTHORITY

Code word C	Code No 187	Message or true reading.
		Authority—Continued
Cannot	00	Give them authority
Cannula	01	Give you authority
Cannulated	02	Given authority
Canny	03	Great authority
Canoe	04	Has authority
Canoed	05	Has no authority
Canoeing	06	Has not authority
Canoeist	07	Have authority
Canoeists	08	Have authority from
Canoes	09	Have authority to
Canon	10	Have no authority
Canonbit	11	Have no other authority
Canonbone	12	Have they authority
Canoness	13	Have we authority
Canonic	14	Have you authority
Canonical	15	He has authority from
Canonicals	16	I have authority from
Canonicate	17	If they have authority
Canonist	18	If we have authority
Canonistic	19	If you have authority

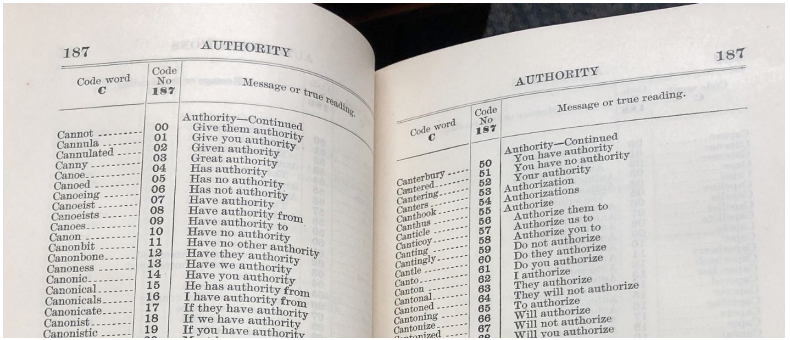
AUTHORITY 187

Code word C	Code No 187	Message or true reading.
		Authority—Continued
Canterbury	50	You have authority
Cantered	51	You have no authority
Cantering	52	Your authority
Canters	53	Authorization
Canthook	54	Authorizations
Canthus	55	Authorize
Canticle	56	Authorize them to
Canticoy	57	Authorize us to
Canting	58	Authorize you to
Cantingly	59	Do not authorize
Cantle	60	Do they authorize
Canto	61	Do you authorize
Canton	62	I authorize
Cantonal	63	They authorize
Cantoned	64	They will not authorize
Cantoning	65	To authorize
Cantonize	66	Will authorize
Cantonized	67	Will not authorize
		Will you authorize

Order-Preserving Encryption

- Why is the codebook order-preserving?

- Why would we like order now?



187 AUTHORITY

Code word ℂ	Code No. 187	Message or true reading.
		Authority—Continued
Cannot	00	Give them authority
Cannula	01	Give you authority
Cannulated	02	Given authority
Canny	03	Great authority
Canoe	04	Has authority
Canoed	05	Has no authority
Canoeing	06	Has not authority
Canoeist	07	Have authority
Canoeists	08	Have authority from
Canoes	09	Have authority to
Canon	10	Have no authority
Canonbit	11	Have no other authority
Canonbone	12	Have they authority
Canoness	13	Have we authority
Canonic	14	Have you authority
Canonical	15	He has authority from
Canonicals	16	I have authority from
Canonicate	17	If they have authority
Canonist	18	If we have authority
Canonistic	19	If you have authority

AUTHORITY 187

Code word ℂ	Code No. 187	Message or true reading.
		Authority—Continued
Canterbury	50	You have authority
Cantered	51	You have no authority
Cantering	52	Your authority
Canters	53	Authorization
Canthook	54	Authorizations
Canthus	55	Authorize
Canticle	56	Authorize them to
Canticoy	57	Authorize us to
Canting	58	Authorize you to
Cantingly	59	Do not authorize
Cantle	60	Do they authorize
Canto	61	Do you authorize
Canton	62	I authorize
Cantonal	63	They authorize
Cantoned	64	They will not authorize
Cantoning	65	To authorize
Cantonize	66	Will authorize
	67	Will not authorize
		Will you authorize

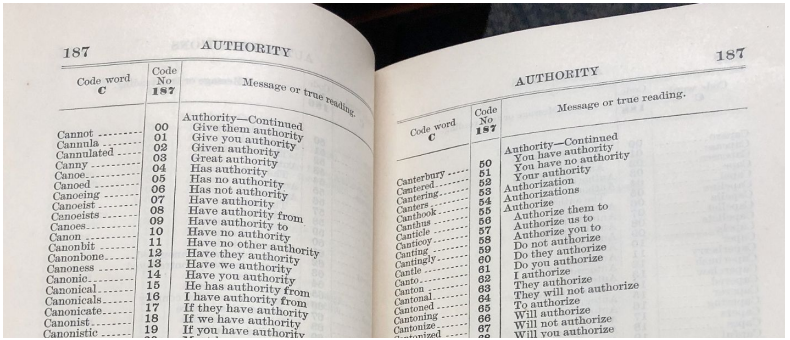
Order-Preserving Encryption

- Why is the codebook order-preserving?

- Why would we like order now?

  - Efficient lookups (binary search)
  - Range queries

- Is it secure?



187 AUTHORITY

Code word C	Code No. 187	Message or true reading.
		Authority—Continued
Cannot	00	Give them authority
Cannula	01	Give you authority
Cannulated	02	Given authority
Canny	03	Great authority
Canoe	04	Has authority
Canoed	05	Has no authority
Canoeing	06	Has not authority
Canoeist	07	Have authority
Canoeists	08	Have authority from
Canoes	09	Have authority to
Canon	10	Have no authority
Canonbit	11	Have no other authority
Canonbone	12	Have they authority
Canoness	13	Have we authority
Canonic	14	Have you authority
Canonical	15	He has authority from
Canonicals	16	I have authority from
Canonicate	17	If they have authority
Canonist	18	If we have authority
Canonistic	19	If you have authority

AUTHORITY 187

Code word C	Code No. 187	Message or true reading.
		Authority—Continued
Canterbury	50	You have authority
Cantered	51	You have no authority
Cantering	52	Your authority
Canters	53	Authorization
Canthook	54	Authorizations
Canthus	55	Authorize
Canticle	56	Authorize them to
Canticoy	57	Authorize us to
Canting	58	Authorize you to
Cantingly	59	Do not authorize
Cantle	60	Do they authorize
Canto	61	Do you authorize
Canton	62	I authorize
Cantonal	63	They authorize
Cantoned	64	They will not authorize
Cantoning	65	To authorize
Cantonize	66	Will authorize
	67	Will not authorize
		Will you authorize

Order-Preserving Encryption

- Why is the codebook order-preserving?

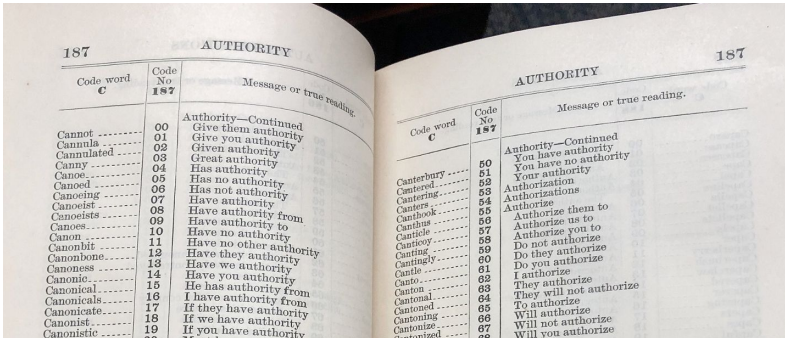
- Why would we like order now?

  - Efficient lookups (binary search)
  - Range queries

- Is it secure?

  - IND-CPA?

  - But still useful?



187 AUTHORITY

Code word C	Code No. 187	Message or true reading.
		Authority—Continued
Cannot	00	Give them authority
Cannula	01	Give you authority
Cannulated	02	Given authority
Canny	03	Great authority
Canoe	04	Has authority
Canoed	05	Has no authority
Canoeing	06	Has not authority
Canoeist	07	Have authority
Canoeists	08	Have authority from
Canoes	09	Have authority to
Canon	10	Have no authority
Canonbit	11	Have no other authority
Canonbone	12	Have they authority
Canoness	13	Have we authority
Canonic	14	Have you authority
Canonical	15	He has authority from
Canonicals	16	I have authority from
Canonicate	17	If they have authority
Canonist	18	If we have authority
Canonistic	19	If you have authority

AUTHORITY 187

Code word C	Code No. 187	Message or true reading.
		Authority—Continued
Canterbury	50	You have authority
Cantered	51	You have no authority
Cantering	52	Your authority
Canters	53	Authorisation
Canthook	54	Authorizations
Canthus	55	Authorize
Canticle	56	Authorize them to
Canticoy	57	Authorize us to
Canting	58	Authorize you to
Cantingly	59	Do not authorize
Cantle	60	Do they authorize
Canto	61	Do you authorize
Canton	62	I authorize
Cantonal	63	They authorize
Cantoned	64	They will not authorize
Cantoning	65	To authorize
Cantonize	66	Will authorize
	67	Will not authorize
		Will you authorize

OPE: Boldyreva et al. 2009

- Symmetric

- Ciphertext space >> Plaintext space

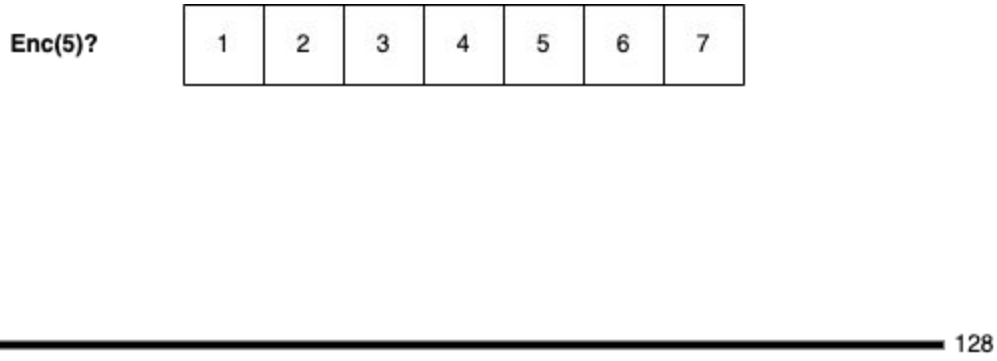
- Pick a ‘random’ but ordered encryption using binary search

- ‘Randomness’ deterministically generated with a block cipher.

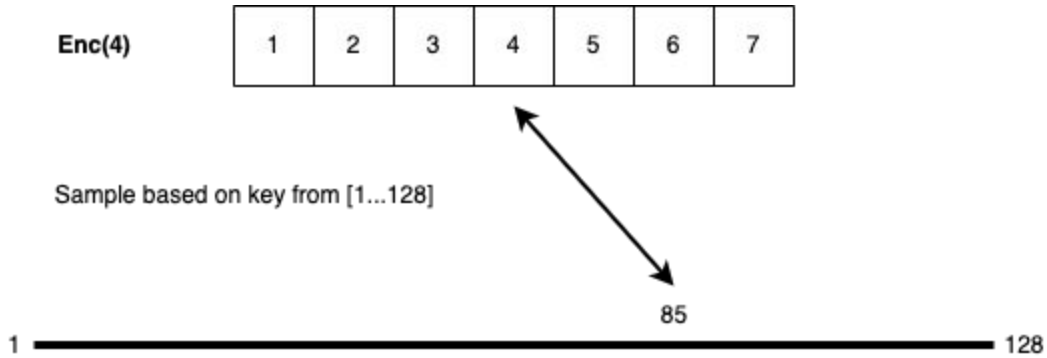
- Decrypt also with binary search

Let’s try with:
Plaintext space [1…7]
Ciphertext space [1…128]
Encrypt 5

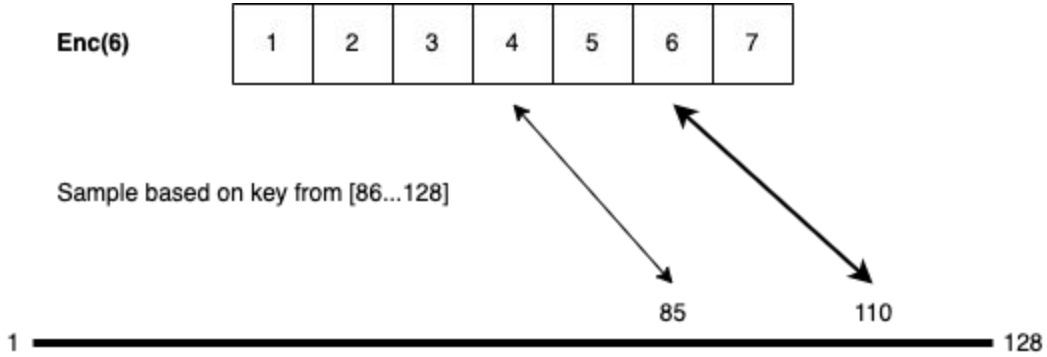
OPE: Boldyreva et al. 2009



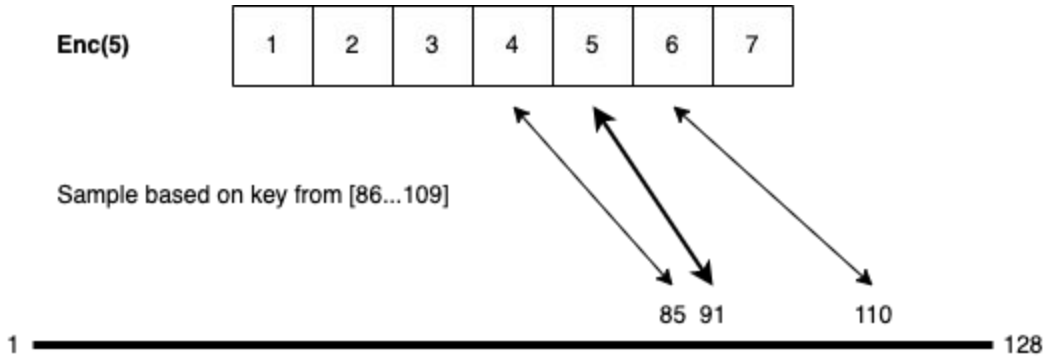
OPE: Boldyreva et al. 2009



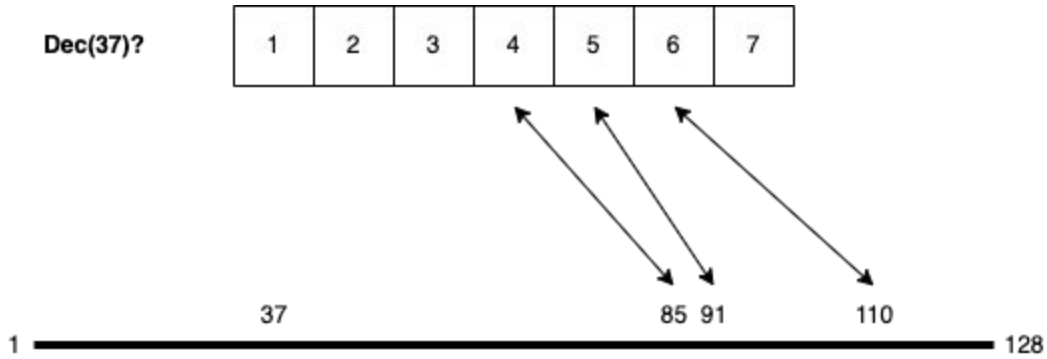
OPE: Boldyreva et al. 2009



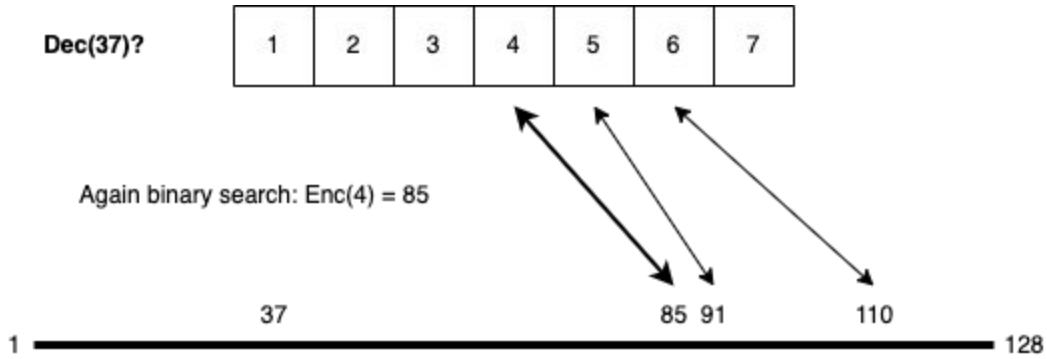
OPE: Boldyreva et al. 2009



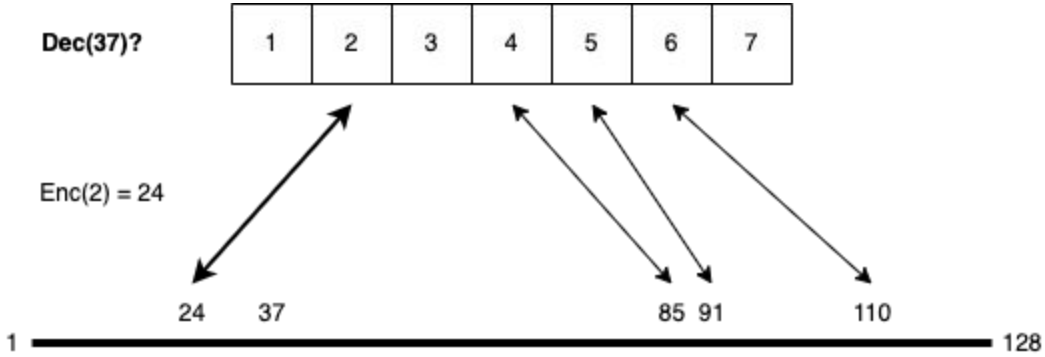
OPE: Boldyreva et al. 2009



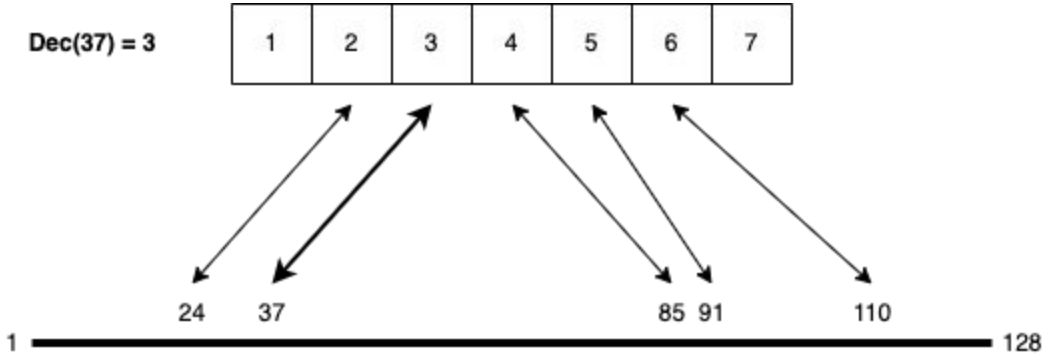
OPE: Boldyreva et al. 2009



OPE: Boldyreva et al. 2009



OPE: Boldyreva et al. 2009



OPE: Limitations

- Public key setting?

OPE: Limitations

- Public key setting?

    - Attacker could encrypt
    - So attacker could decrypt with binary search

OPE: Limitations

- Public key setting?
    - Attacker could encrypt
    - So attacker could decrypt with binary search
- Leaks order (intentionally), is this a problem?

OPE: Limitations

- Public key setting?

    - Attacker could encrypt
    - So attacker could decrypt with binary search

- Leaks order (intentionally), is this a problem?

    Does ‘size’ matter?

        - Age, income, weight
        - UUID
        - Phone number
        - Credit card number?

Attacks on OPE

Using only an encrypted column and statistics:

Given an estimated distribution of the plaintexts

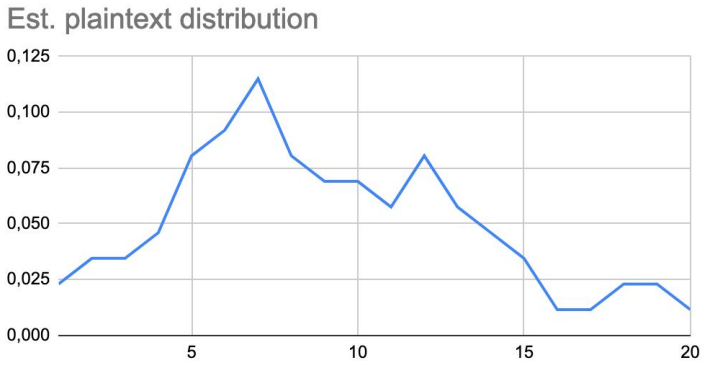
For each ciphertext:

Get percentage of ciphertexts that are smaller

Match that to the estimated distribution to get a likely plaintext

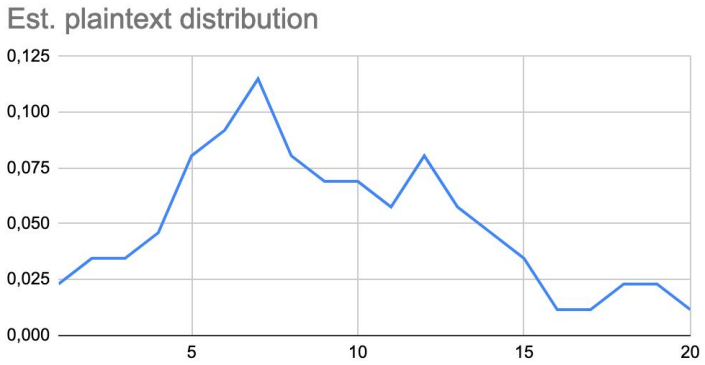
Attacks on OPE

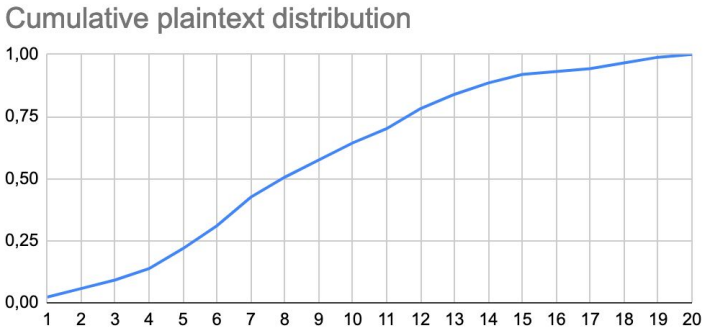
Ciphertext	?	Plaintext?
25		
30		
41		
42		
56		
82		
205		
234		



Attacks on OPE

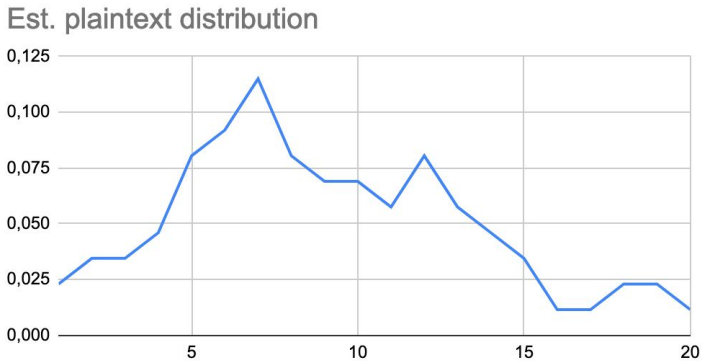
Ciphertext	?	Plaintext?
25		
30		
41		
42		
56		
82		
205		
234		

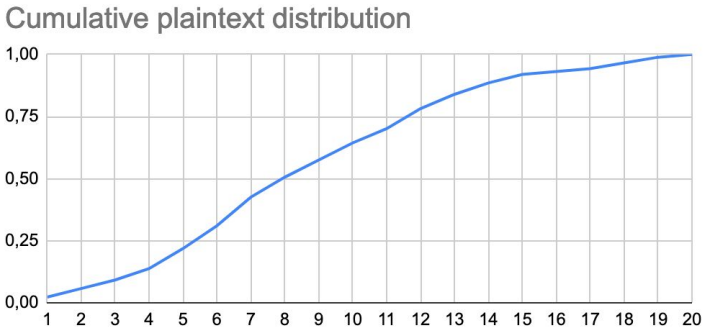




Attacks on OPE

Ciphertext	CDF	Plaintext?
25	0	
30	0.125	
41	0.25	
42	0.375	
56	0.5	
82	0.625	
205	0.75	
234	0.875	





Conclusion

- Order is a lot of information
    - Especially if just an estimation is already privacy-sensitive

- OPE leaks much more than SSE from before

- Can we still use searchable encryption?

    - In some cases
    - But need to be very careful allowing more than exact-match queries

    - Active area of research!