

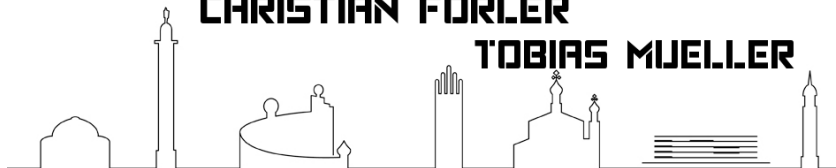


MRMCD2015
SCHNELLER. HÖHER. WEITER.
4.-6. SEP. HS DARMSTADT

THE MAGIC WORLD OF SEARCHABLE ENCRYPTION

CHRISTIAN FORLER

TOBIAS MUELLER

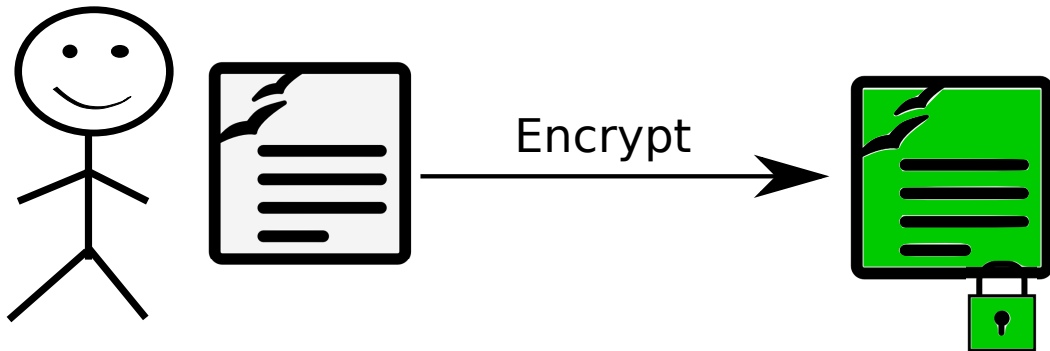


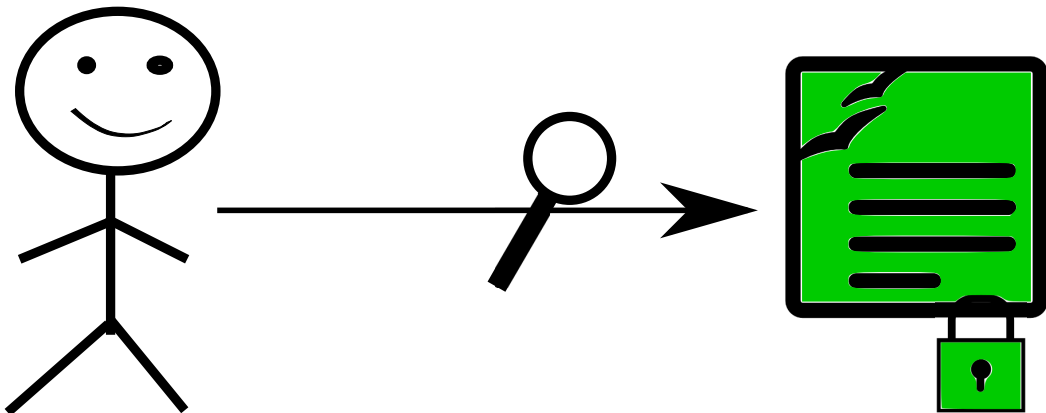
- 1 General Scenario
- 2 Why...? - Ideas?
- 3 Approaches
- 4 Can we do better?
- 5 Index based
- 6 Outlook
- 7 Conclusions

General Scenario

General Scenario

User encrypts data, sends it to a server, forgets about it, then wants to search it for, e.g. substrings





Why...? - Ideas?

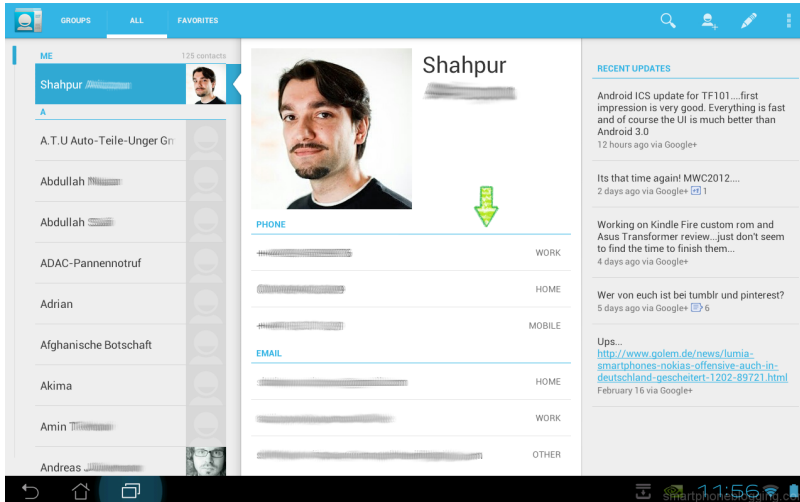
Motivation

- Emails

Motivation

- Emails
- Documents

Example: Contacts



Example: More Concrete

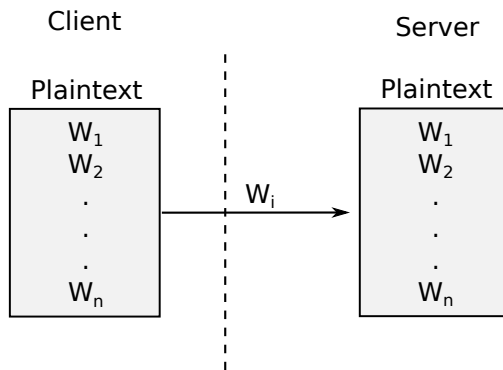
Store

First Name	Last Name	Number
Alice	Foo	123
Bob	Foo	345
Eve	Bar	456

securely in the cloud™

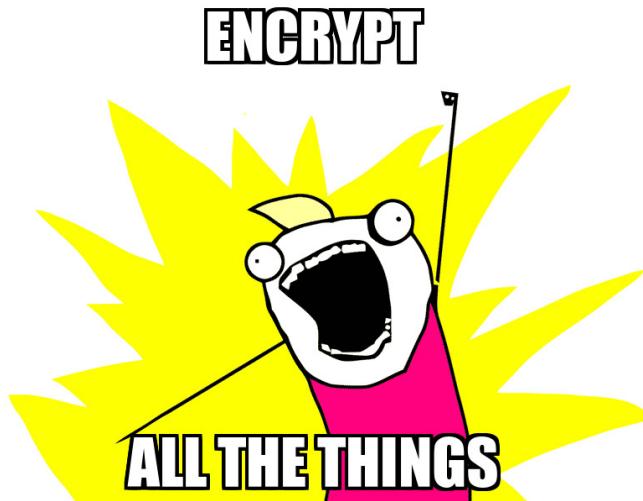
Approaches

Plaintext



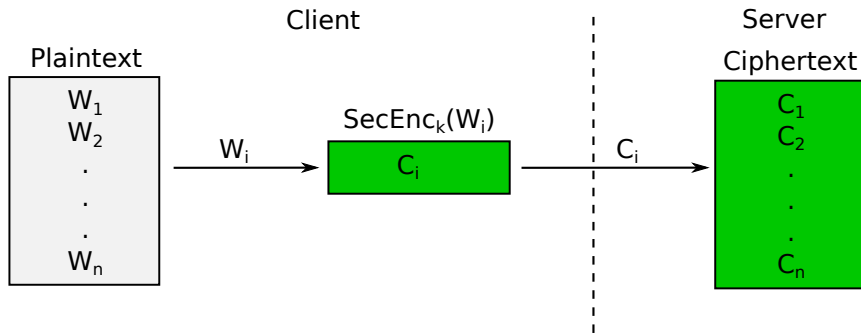
Now, the server knows your contacts. :-)





Simple Crypto

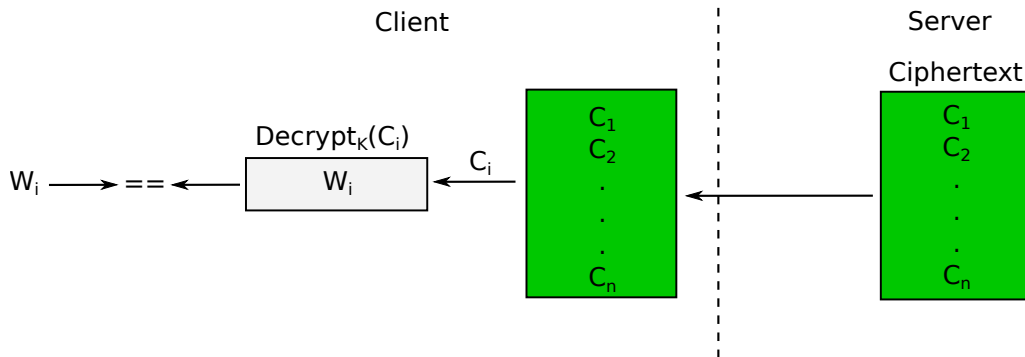
Encrypt all the things!



REALLY DOWNLOAD

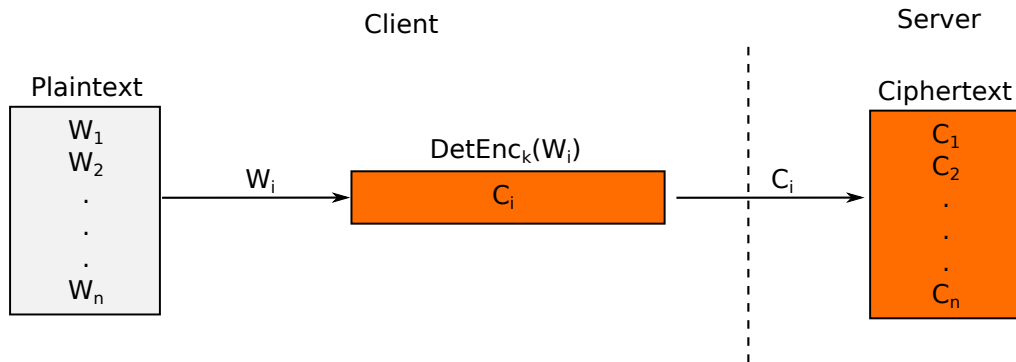


Simple Crypto - Search

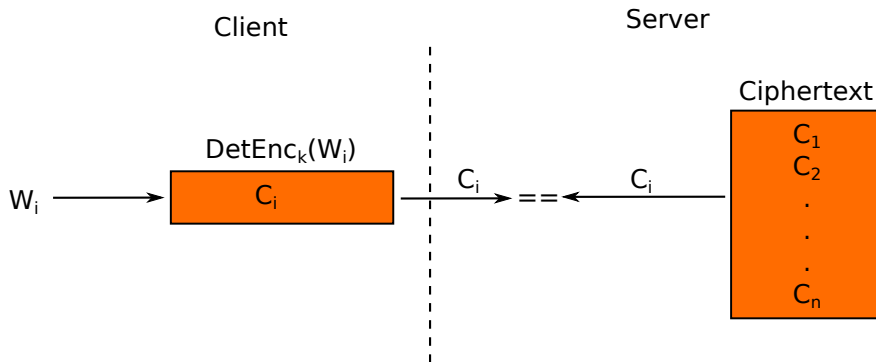


Can we do better?

Deterministic Encryption of Keywords - Setup



Deterministic Encryption of Keywords - Search



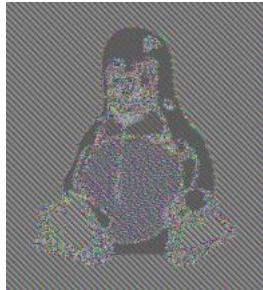
Deterministic Encryption of Keywords - Problem



problem?

Deterministic Encryption of Keywords - Problem

Deterministic encryption sucks!

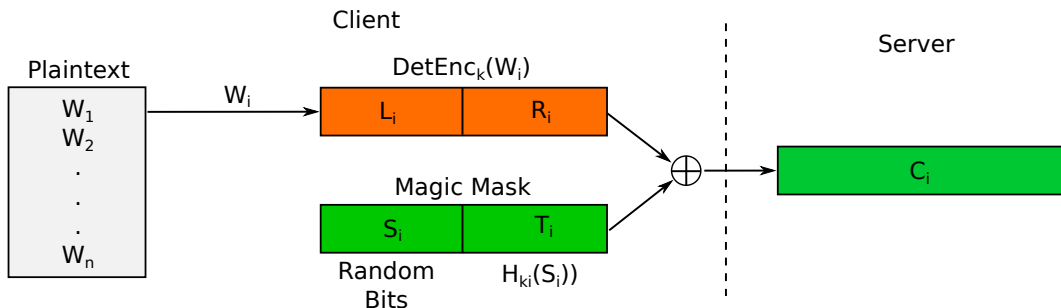


Keyword based - Setup (Song, Wagner, Perrig)

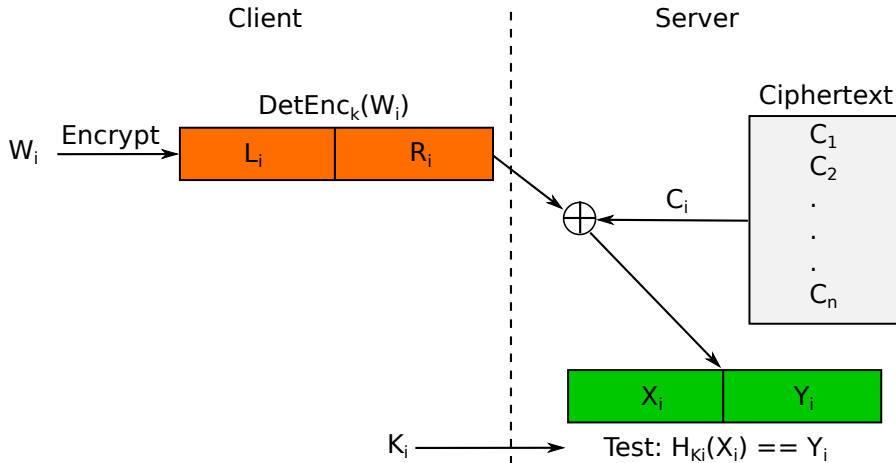
Encrypt-then-Mask

Search key $k_i = H_k(L_i)$

Magic Mask: T_i can be derived from S_i , i.e. $T_i = H_{k_i}(S_i)$



Keyword based - Search



Speed

Plaintext size (King James Bible): 4.3 MB

Ciphertext size: 25 MB

Time to encrypt: 0.211 sec

Search (in seconds): • Foobar 0.181

- God 0.003
- towel: 0.155
- Eve 0.005
- wrath 0.014
- dragon 0.094

Index based

Plaintext Index - Search

Client

Plaintext

1	Alice	Foo
2	Bob	Foo
3	Eve	Bar

Index

Token	Value
H(Foo)	[1, 2]
H(Bar)	[3]

Server

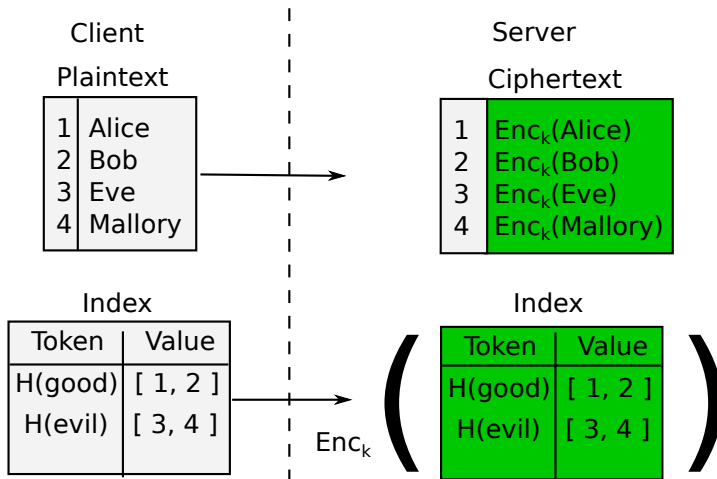
Ciphertext

1	$\text{Enc}_k(\text{Alice}, \text{Foo})$
2	$\text{Enc}_k(\text{Bob}, \text{Foo})$
3	$\text{Enc}_k(\text{Eve}, \text{Bar})$

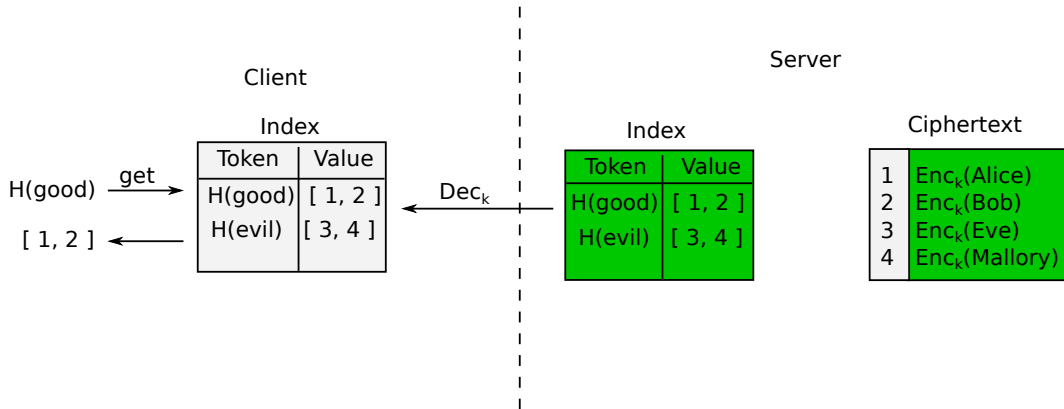
Plaintext Index - Hell of Synchronisation



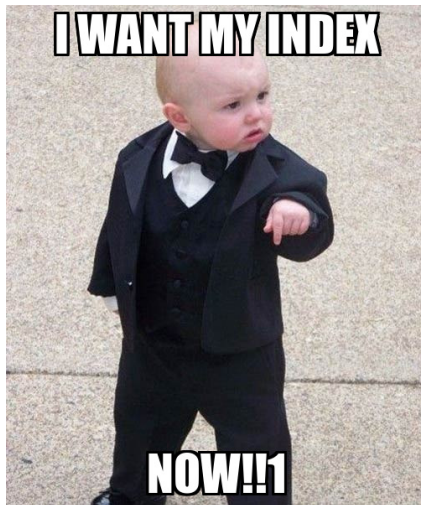
Enc. Index based - Setup



Enc. Index based - Search



Communication Cost



Searchable Enc. Index

Client

Plaintext

1	Alice	Foo
2	Bob	Foo
3	Eve	Bar

$sk1 = H_k(\text{search} || \text{Foo})$

$sk2 = H_k(\text{search} || \text{Bar})$

$ik1 = H_k(\text{index} || \text{Foo})$

$ik2 = H_k(\text{index} || \text{Bar})$

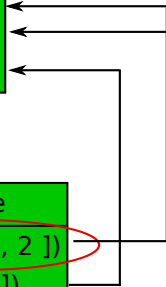
Server

Ciphertext

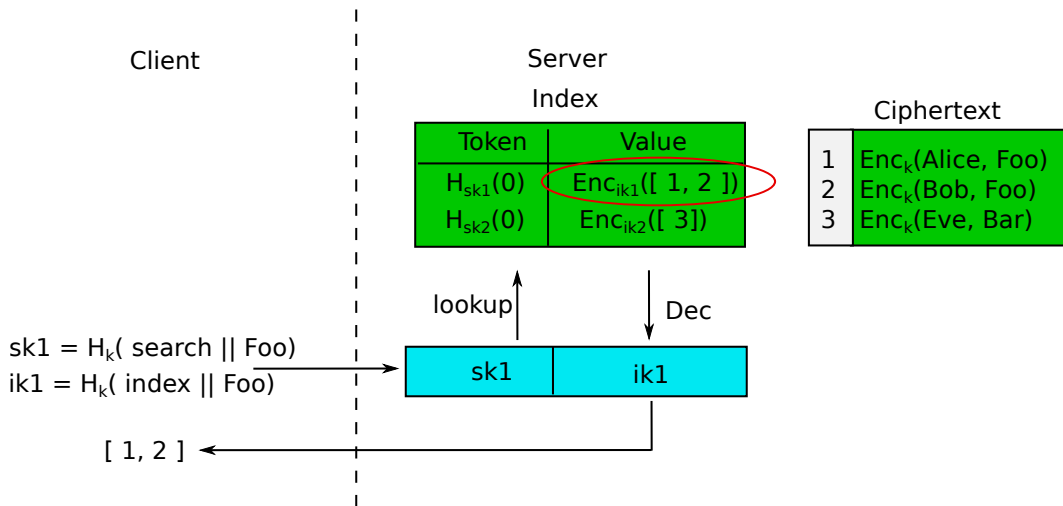
1	$Enc_k(\text{Alice}, \text{Foo})$
2	$Enc_k(\text{Bob}, \text{Foo})$
3	$Enc_k(\text{Eve}, \text{Bar})$

Index

Token	Value
$H_{sk1}(0)$	$Enc_{ik1}([1, 2])$
$H_{sk2}(0)$	$Enc_{ik2}([3])$



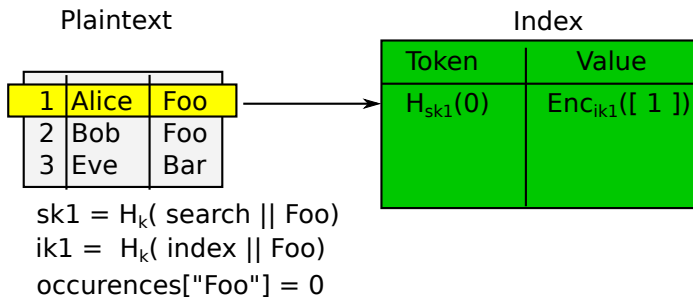
Searchable Enc. Index - Search



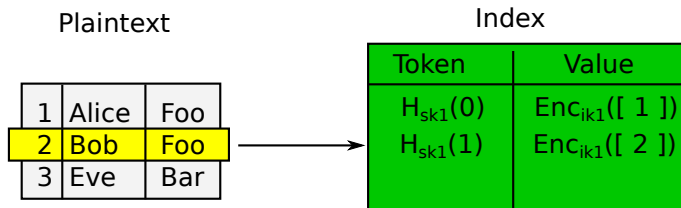
Size matters



Index based - Cash et al. - Setup



Index based - Cash et al. - Setup (contd.)

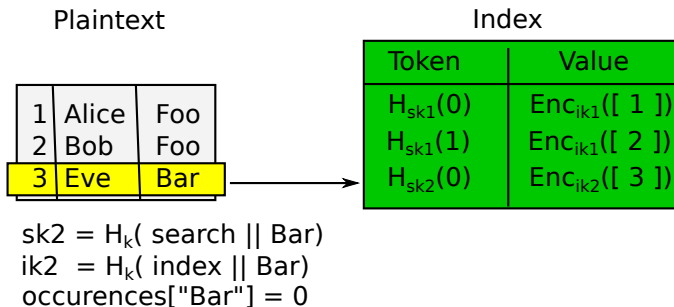


$sk1 = H_k(\text{search} || \text{Foo})$

$ik1 = H_k(\text{index} || \text{Foo})$

$occurrences["Foo"] = 1$

Index based - Cash et al. - Setup (contd.)



Index based - Cash et al.

Client

Plaintext

1	Alice	Foo
2	Bob	Foo
3	Eve	Bar

 $sk1 = H_k(\text{search} \parallel \text{Foo})$
 $sk2 = H_k(\text{search} \parallel \text{Bar})$
 $ik1 = H_k(\text{index} \parallel \text{Foo})$
 $ik2 = H_k(\text{index} \parallel \text{Bar})$

Server

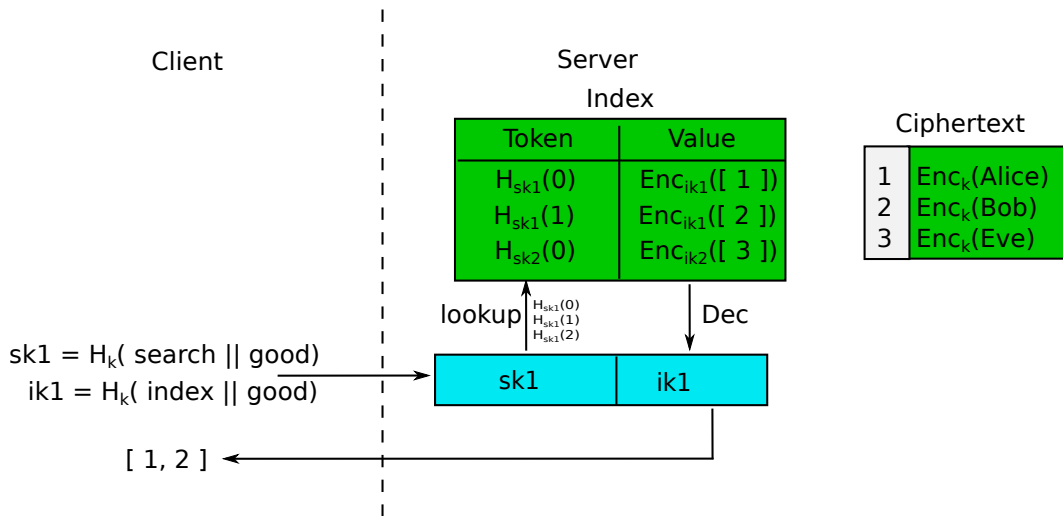
Ciphertext

1	$Enc_k(\text{Alice}, \text{Foo})$
2	$Enc_k(\text{Bob}, \text{Foo})$
3	$Enc_k(\text{Eve}, \text{Bar})$

Index

Token	Value
$H_{sk1}(0)$	$Enc_{ik1}([1])$
$H_{sk1}(1)$	$Enc_{ik1}([2])$
$H_{sk2}(0)$	$Enc_{ik2}([3])$

Index based - Cash et al. - Search



Speed

Plaintext size (King James Bible): 4.3 MB

Ciphertext size: 4.3 MB

Time to encrypt: 0.108 sec

Time to search: 0.001 sec

Outlook

Outlook

- So far: deterministic search token \rightarrow statistical analysis

Outlook

- So far: deterministic search token \rightarrow statistical analysis
- Making existing approaches practical is a challenge (e.g. FHE)

Outlook

- So far: deterministic search token \rightarrow statistical analysis
- Making existing approaches practical is a challenge (e.g. FHE)
- Implement and adapt!!1

Conclusions

Conclusions

- Presented some schemes and their properties

Conclusions

- Presented some schemes and their properties
 - Det

Conclusions

- Presented some schemes and their properties
 - Det
 - Fast setup

Conclusions

- Presented some schemes and their properties
 - Det
 - Fast setup
 - search insecure

Conclusions

- Presented some schemes and their properties
 - Det
 - Fast setup
 - search insecure
 - Keyword (Song, Wagner, Perrig)

Conclusions

- Presented some schemes and their properties
 - Det
 - Fast setup
 - search insecure
 - Keyword (Song, Wagner, Perrig)
 - Search is in $O(n)$

Conclusions

- Presented some schemes and their properties
 - Det
 - Fast setup
 - search insecure
 - Keyword (Song, Wagner, Perrig)
 - Search is in $O(n)$
 - Index (Cash et al.)

Conclusions

- Presented some schemes and their properties
 - Det
 - Fast setup
 - search insecure
 - Keyword (Song, Wagner, Perrig)
 - Search is in $O(n)$
 - Index (Cash et al.)
 - Search is in $O(1)$

Conclusions

- Presented some schemes and their properties
 - Det
 - Fast setup
 - search insecure
 - Keyword (Song, Wagner, Perrig)
 - Search is in $O(n)$
 - Index (Cash et al.)
 - Search is in $O(1)$
 - Index maintenance needed (think: Update)

Conclusions

- Presented some schemes and their properties
 - Det
 - Fast setup
 - search insecure
 - Keyword (Song, Wagner, Perrig)
 - Search is in $O(n)$
 - Index (Cash et al.)
 - Search is in $O(1)$
 - Index maintenance needed (think: Update)
- slightly different features

Conclusions

- Presented some schemes and their properties
 - Det
 - Fast setup
 - search insecure
 - Keyword (Song, Wagner, Perrig)
 - Search is in $O(n)$
 - Index (Cash et al.)
 - Search is in $O(1)$
 - Index maintenance needed (think: Update)
- slightly different features
- more exist!

Conclusions

- Presented some schemes and their properties
 - Det
 - Fast setup
 - search insecure
 - Keyword (Song, Wagner, Perrig)
 - Search is in $O(n)$
 - Index (Cash et al.)
 - Search is in $O(1)$
 - Index maintenance needed (think: Update)
- slightly different features
- more exist!
- Searching on encrypted data is practical

Thanks!

References:

Dawn Xiaodong Song, David Wagner, Adrian Perrig: Practical Techniques for Searches on Encrypted Data. IEEE Symposium on Security and Privacy 2000: 44-55

David Cash, Joseph Jaeger, Stanislaw Jarecki, Charanjit S. Jutla, Hugo Krawczyk, Marcel-Catalin Rosu, Michael Steiner: Dynamic Searchable Encryption in Very-Large Databases: Data Structures and Implementation. NDSS 2014