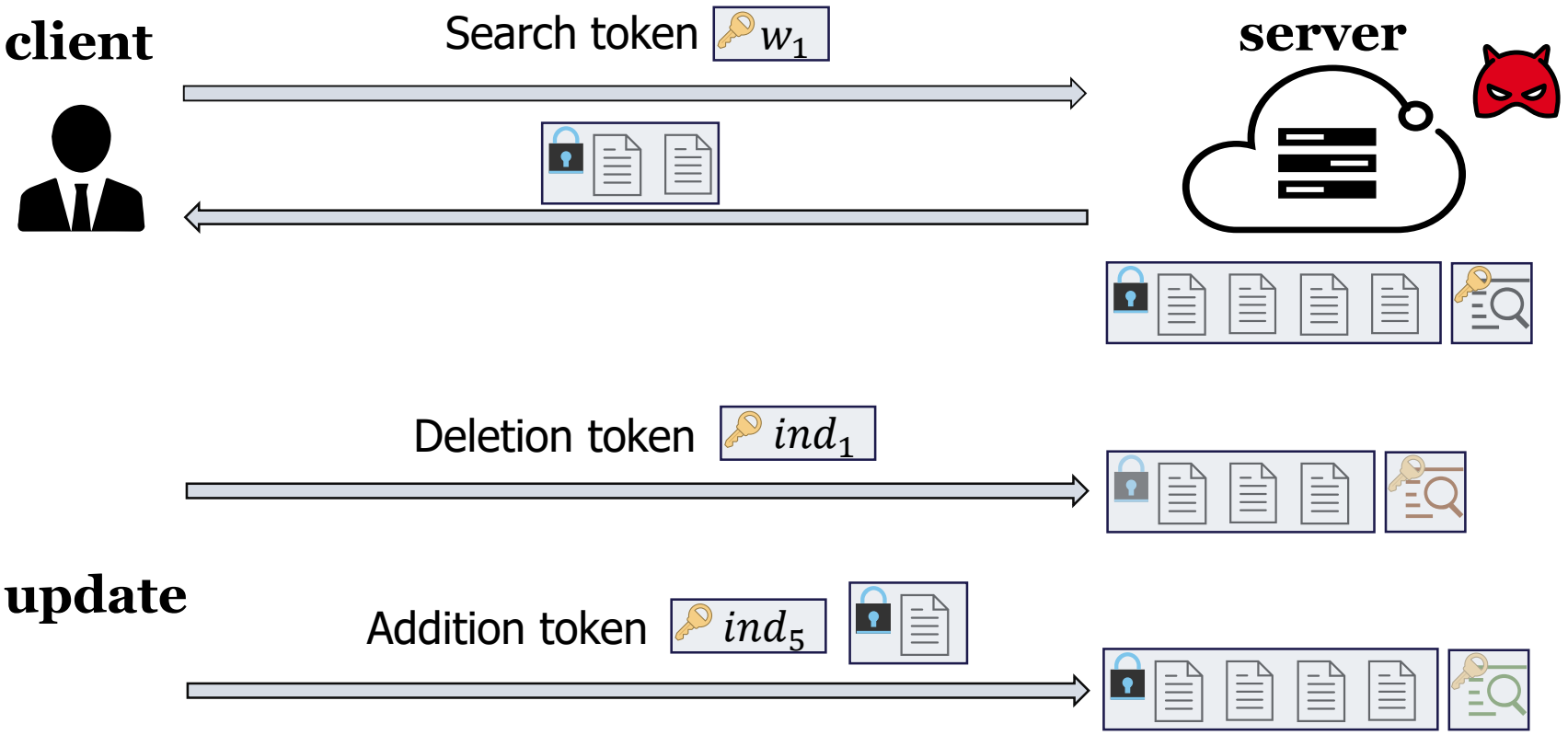
Dynamic Searchable Symmetric Encryption with Forward and Backward Privacy

Yu Peng[a], Qin Liu[a], Yue Tian[a], Jie Wu[b], Tian Wang[c],
Tao Peng[d], and Guojun Wang[d]

[a] Hunan University
[b] Temple University
[c] Beijing Normal University & UIC
[d] Guangzhou University

2021-10-21
IEEE TrustCom'21, Shenyang, China

Dynamic Searchable Symmetric Encryption

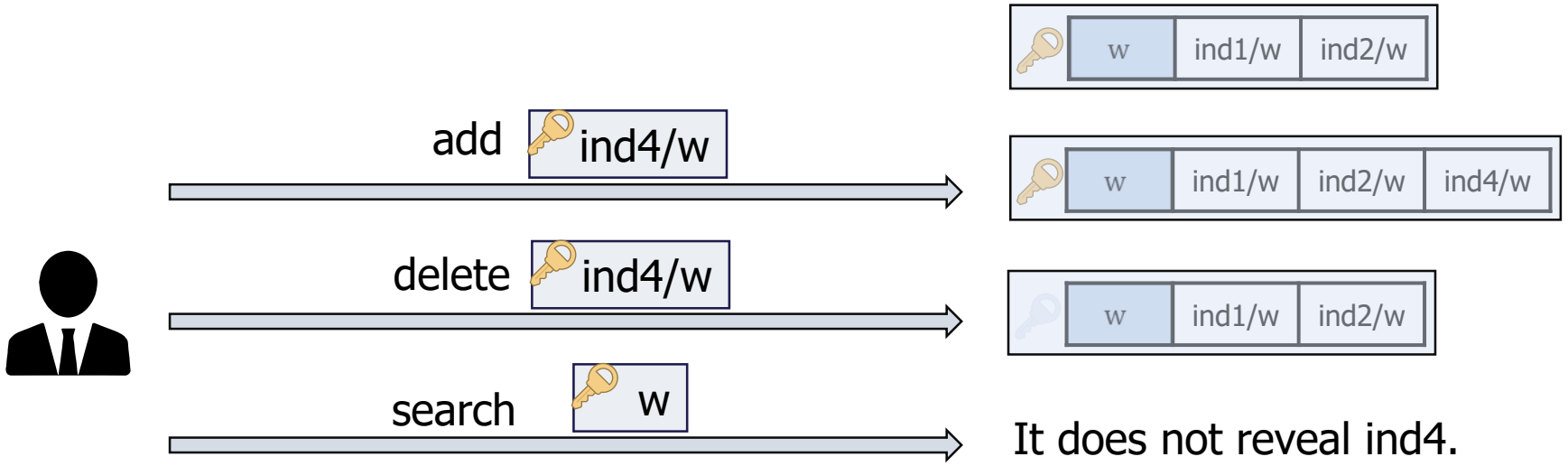


With the trade of security and efficiency, most of the existing DSSE schemes leak some information (e.g., search pattern) in data search and update phases.

Forward and Backward Privacy

- Forward privacy (FP) requires that the newly added files cannot be linked to previous search tokens.

- Backward privacy (BP): the deleted files cannot be searched any more.



Research Status

- Keyword-level updates
  - The client needs to know all keywords contained in the document to be updated in advance
  - CCS'15 Sophos, CCS'17 Janus, CCS'18 Janus++
- Document-level deletions
  - The client requires only an identifier of the document in the deletion operation
  - TDSC'21 Khons (the storage cost at the client is liner with the number of documents.)
- Actual deletion
  - Many FP/BP schemes perform the deletion in the same way as the addition
  - CCS'15 Sophos, TDSC'20 FAST

Document-based Backward Privacy (DBP)

Threat model: the server is considered to be honest-but-curious (HBC).

Document	Keywords
ind_1	w_1, w_2
ind_2	w_2, w_3

Client:

secret key: k_1

T_C	
keyword	key
w_1	$key^{w_1'}$
w_2	key^{w_2}
w_3	key^{w_3}

Server:

T_e	
keyind	Value set (e)
~~$F_{k_1}(ind_1)$~~	~~$H_1(F_{k_1}(ind_1), key^{w_1}) \oplus < ind_1, \mathbf{0} >$~~ ~~$H_1(F_{k_1}(ind_1), key^{w_2}) \oplus < ind_1, \mathbf{0} >$~~
$F_{k_1}(ind_2)$	$H_1(F_{k_1}(ind_2),\ key^{w_2}) \oplus < ind_2, \mathbf{0} >$ $H_1(F_{k_1}(ind_2),\ key^{w_3}) \oplus < ind_2, \mathbf{0} >$

T_P	
keyword	ind
key^{w_1}	~~ind_1~~

search token: $(F_{k_1}(w_1), key^{w_1})$

$mask \leftarrow (\text{keyind}, F_{k_1}(w_1))$ $mask \oplus e \rightarrow$ [redacted] **0?**

deletion token: $(F_{k_1}(\text{ind}_1), ind_1)$

Advanced DBP: 2 Round Trip

Client:

secret key: k_1

T_C	
keyword	key
w_1	$key^{w_1'}$
w_2	key^{w_2}
w_3	key^{w_3}

Server:

T_e	
keyind	Value set (e)
$F_{k_1}(ind_1)$	~~$H_1(F_{k_1}(ind_1), key^{w_1}) \oplus < ind_1, \mathbf{0} >$~~ $H_1(F_{k_1}(ind_1), key^{w_2}) \oplus < ind_1, \mathbf{0} >$
$F_{k_1}(ind_2)$	$H_1(F_{k_1}(ind_2), key^{w_2}) \oplus < ind_2, \mathbf{0} >$ $H_1(F_{k_1}(ind_2), key^{w_3}) \oplus < ind_2, \mathbf{0} >$

T_P	
keyword	ind
key^{w_1}	ind_1

The server may be attacked by the external attackers, who can obtain control over the server.

$H_1(F_{k_1}(ind_1), key^{w_1}) \oplus < ind_1, \mathbf{0} >$ ⇒ $H_1(F_{k_1}(ind_1), key^{w_1}) \oplus < Enc(k_2, ind_1), \mathbf{0} >$

DBP*: Improved Search Speed

T_e	
keyind	Value set (e)
$F_{k_1}(ind_1)$	$H_1(F_{k_1}(ind_1), key^{w_1}) \oplus < ind_1, \mathbf{0} >$ $H_1(F_{k_1}(ind_1), key^{w_2}) \oplus < ind_1, \mathbf{0} >$
$F_{k_1}(ind_2)$	$H_1(F_{k_1}(ind_2), key^{w_2}) \oplus < ind_2, \mathbf{0} >$ $H_1(F_{k_1}(ind_2), key^{w_3}) \oplus < ind_2, \mathbf{0} >$

$mask \leftarrow (\text{keyind}, F_{k_1}(w_1))$

$mask \oplus e \rightarrow$ 

$Dual - key/value: (< key_1, key_2 >, value)$

$key_1 = F_{k_1}(ind_1)$

$key_2 = e_{\lambda+1 \cdots 2\lambda}$

$value = e_{1 \cdots \lambda}$

- **Search**

$mask \leftarrow (\text{keyind}, F_{k_1}(w_1))$

$< F_{k_1}(ind_1), mask_{\lambda+1 \cdots 2\lambda} >$

$ind \leftarrow mask_{1 \cdots \lambda} \oplus value$

Experiments

- **Instantiation**
  - Implementation: Java
  - Intel(R) Xeon(R) Gold 5218 CPU of 2.3GHz and 128GB RAM

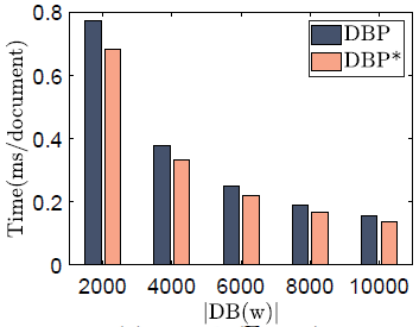
Dataset	Data size(GB)	Documents number	Keywords number	Pairs number
Enron[1]	1.56	577,744	54,021	2,888,720
Wikipedia[2]	10.6	3,506,761	957,920	17,533,805

- **Setup and update**

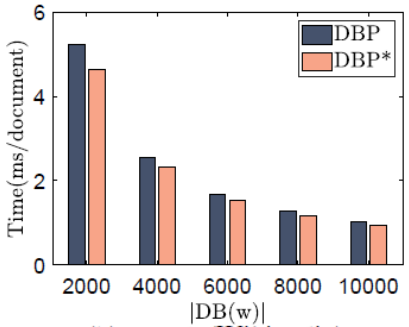
Dataset	DBP-B					DBP-E				
	Setup			Addition	Deletion	Setup			Addition	Deletion
	Time(ms)	Size1(MB)	Size2(MB)	Time(μs)	Time(μs)	Time(ms)	Size1(MB)	Size2(MB)	Time(μs)	Time(μs)
Enron	8727	2.78	274	15.11	2.25	11471	2.78	274	19.85	8.57
Wikipedia	60073	50.07	1662	18.18	2.30	75955	50.07	1662	21.66	9.16

Size1: the storage cost (σ) at the client side, Size2: the storage cost (EDB) at the server side.

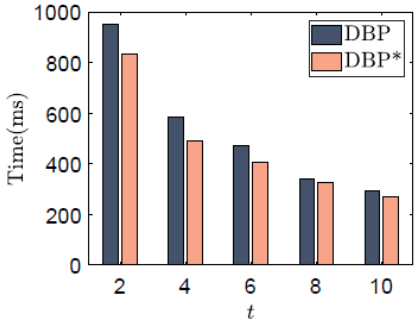
- **Search**



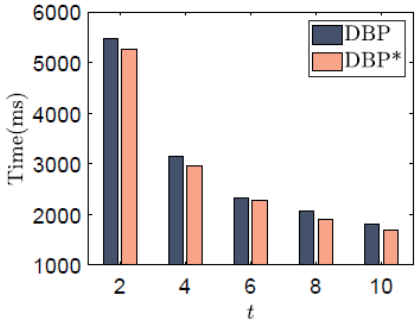
(a) $t = 1$ (Enron)



(b) $t = 1$ (Wikipedia)



(c) DB(w) = 10000 (Enron)



(d) DB(w) = 10000 (Wikipedia)

Conclusion

In this work:

- Devise an efficient forward and backward private scheme, which supports document-based immediate deletion
- Implement our scheme on two real datasets and show its practicality

Thanks for your attention!

Questions & Answers?