Encryption Domain Text Retrieval System

"It used to be expensive to make things public and cheap than to make them private. Now it's expensive to make things private and cheap than to make them public."

Introduction

Cloud is a concept of providing services over the internet. Inside a cloud there are a lot of computing and storage devices.

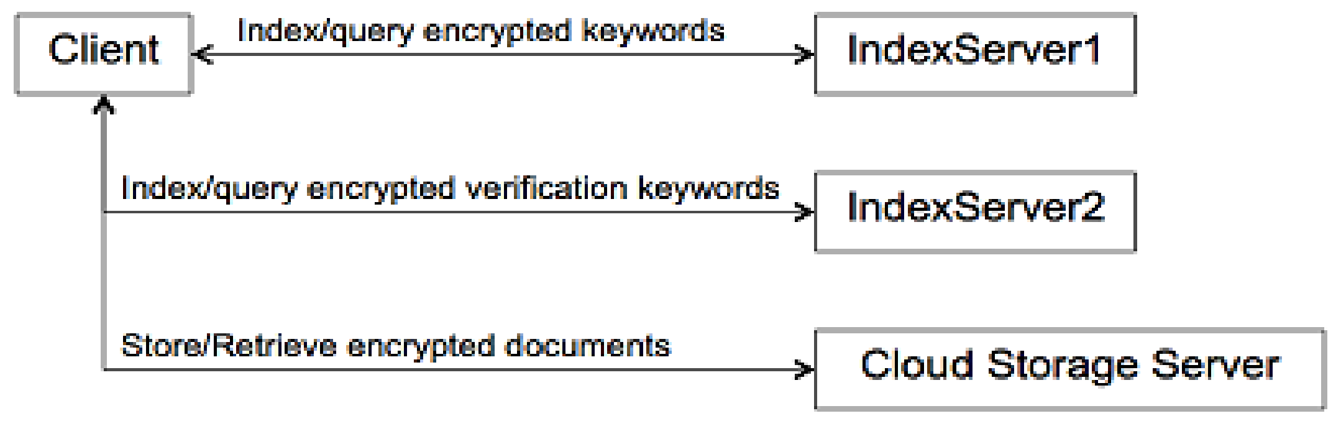
- Store a plain text document on cloud, searching on it is easy, but what about the security of your document?
- Lets propose encryption, but then what happens to the easy searching over text document?

A *trivial solution* is downloading all the documents, decrypting them locally and searching contents. This means we are giving up one thing over another(easy search and Security of documents) in either case for efficient storage we are doing on clouds.

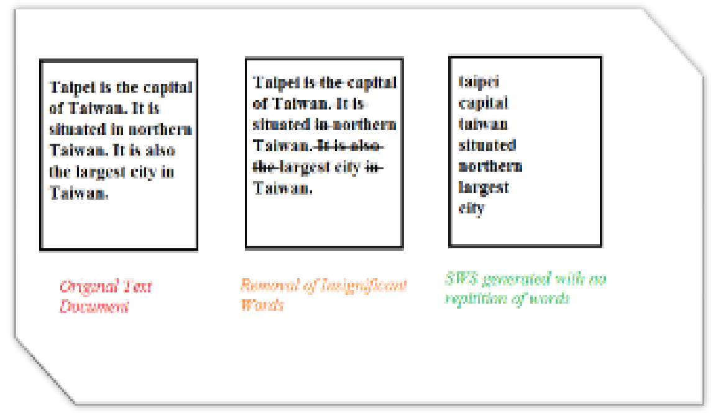
Thus, to provide effective and efficient text retrieval in the cloud, we implement a simple **Encryption Domain Text Retrieval System** while keeping enough privacy protection at the same time

System Architecture and Working

There are four components in the EDTR system as shown in Figure.


Client
Index/query encrypted keywords
IndexServer1
Index/query encrypted verification keywords
IndexServer2
Store/Retrieve encrypted documents
Cloud Storage Server

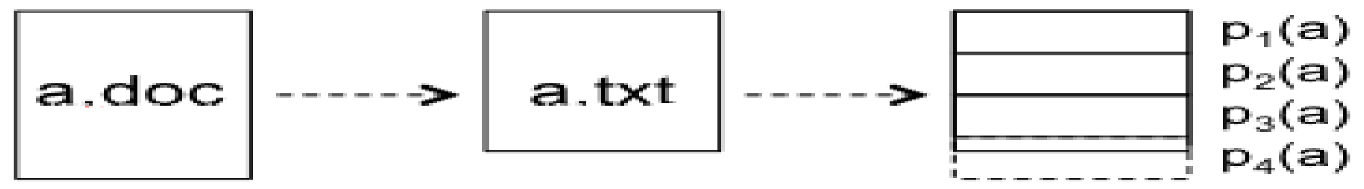
- ***Significant Word Set* (SWS) -** Set of words describing the document, after removing large number of insignificant words like is, the, of etc., that unnecessarily reduce the efficiency of operations like searching where size does matter.
  Word Count ~ 10% of original w.c.


Taipei is the capital of Taiwan. It is situated in northern Taiwan. It is also the largest city in Taiwan.
Taipei is the capital of Taiwan. It is situated in northern Taiwan. ~~It is also the largest city in~~ Taiwan.
taipei capital taiwan situated northern largest city
Original Text Document
Removal of Insignificant Words
SWS generated with no repitition of words

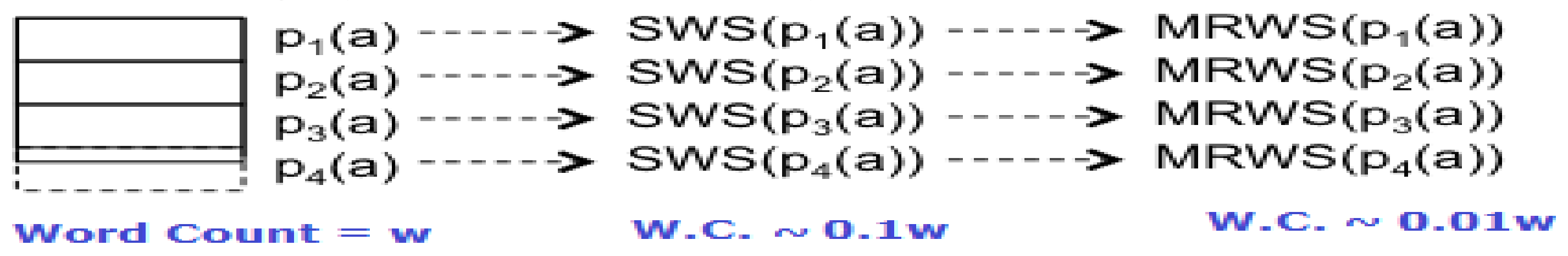
- **Most Relevant Word Set (MRWS) –** best describe our document.
  Word count ~ 1% of original w.c.,
  Main purpose –*integrity check* against server attacks

Client Side:

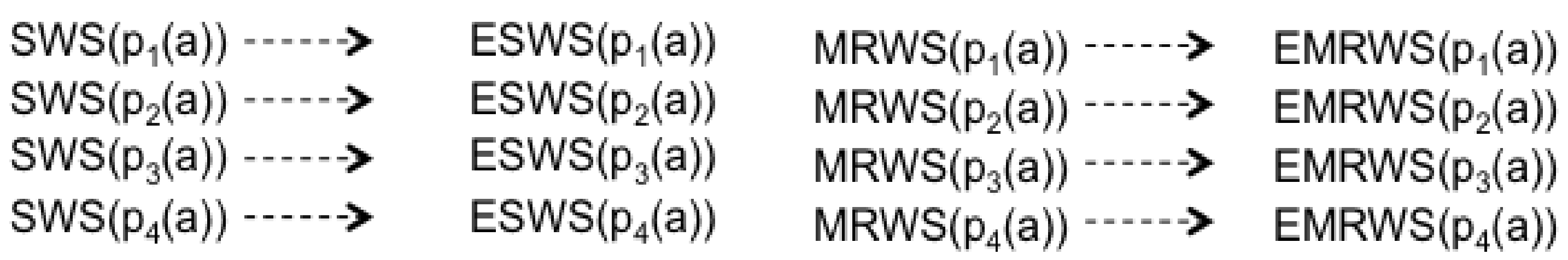
- Extract metadata and text content from documents using Apache Tika Toolkit
- Divide extracted text documents into pages of fixed size



- For each page, generate the SWS (top 10% words with highest occurrence) and MRWS described before, using Apache Lucene Library


$p_1(a)$ → $SWS(p_1(a))$ → $MRWS(p_1(a))$
$p_2(a)$ → $SWS(p_2(a))$ → $MRWS(p_2(a))$
$p_3(a)$ → $SWS(p_3(a))$ → $MRWS(p_3(a))$
$p_4(a)$ → $SWS(p_4(a))$ → $MRWS(p_4(a))$
Word Count = w
W.C. ~ 0.1w
W.C. ~ 0.01w

- Encrypt the SWS and MRWS sets so generated naming them ESWS and EMRWS respectively

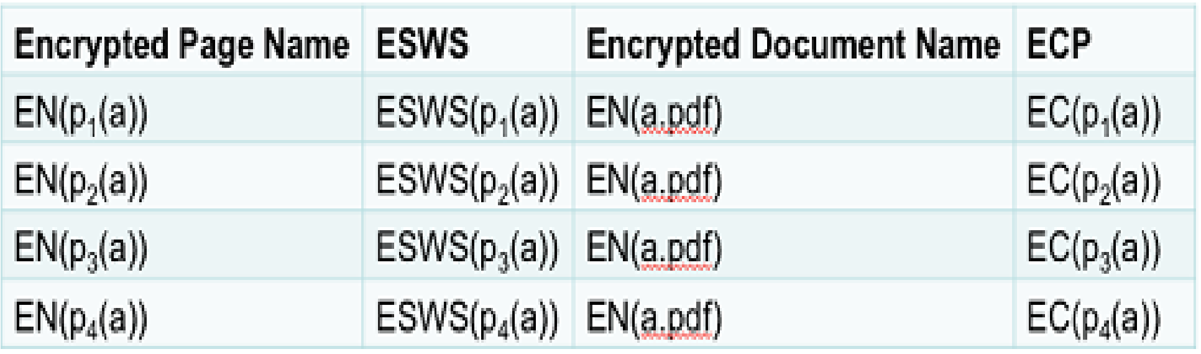


$SWS(p_1(a))$ → $ESWS(p_1(a))$
$SWS(p_2(a))$ → $ESWS(p_2(a))$
$SWS(p_3(a))$ → $ESWS(p_3(a))$
$SWS(p_4(a))$ → $ESWS(p_4(a))$

$MRWS(p_1(a))$ → $EMRWS(p_1(a))$
$MRWS(p_2(a))$ → $EMRWS(p_2(a))$
$MRWS(p_3(a))$ → $EMRWS(p_3(a))$
$MRWS(p_4(a))$ → $EMRWS(p_4(a))$

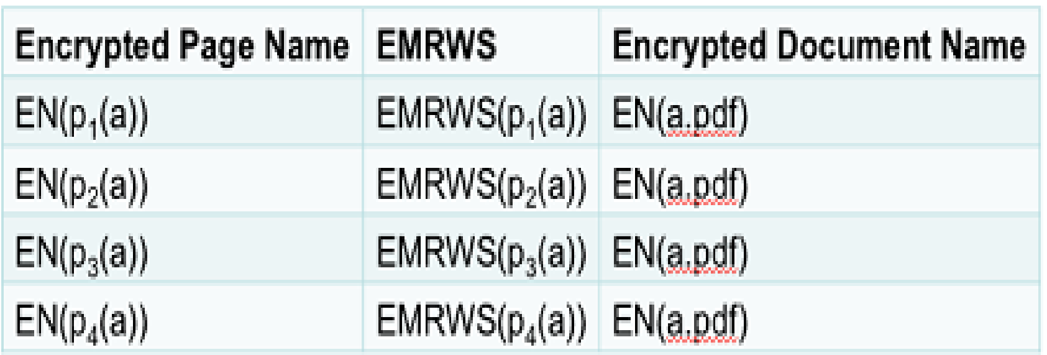
Index Service:

Server Side (only)



Encrypted Page Name	ESWS	Encrypted Document Name	ECP
$EN(p_1(a))$	$ESWS(p_1(a))$	EN(a.pdf)	$EC(p_1(a))$
$EN(p_2(a))$	$ESWS(p_2(a))$	EN(a.pdf)	$EC(p_2(a))$
$EN(p_3(a))$	$ESWS(p_3(a))$	EN(a.pdf)	$EC(p_3(a))$
$EN(p_4(a))$	$ESWS(p_4(a))$	EN(a.pdf)	$EC(p_4(a))$

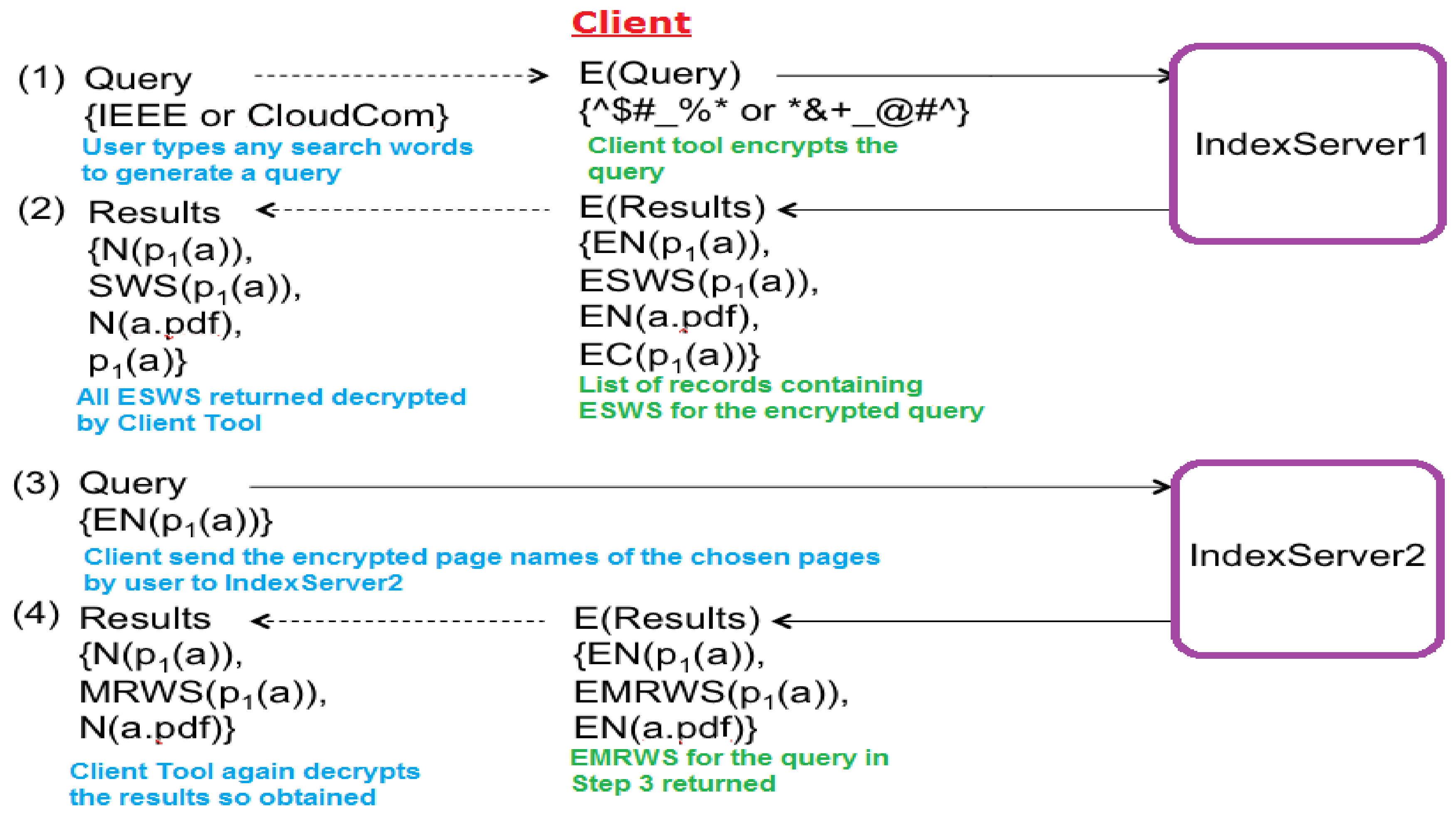
Indexed fields for ESWS



Encrypted Page Name	EMRWS	Encrypted Document Name
$EN(p_1(a))$	$EMRWS(p_1(a))$	EN(a.pdf)
$EN(p_2(a))$	$EMRWS(p_2(a))$	EN(a.pdf)
$EN(p_3(a))$	$EMRWS(p_3(a))$	EN(a.pdf)
$EN(p_4(a))$	$EMRWS(p_4(a))$	EN(a.pdf)

Indexed fields for EMRWS

Query Service:


Client
(1) Query {IEEE or CloudCom}
User types any search words to generate a query
E(Query) {^$#_%* or *&+_@#^}
Client tool encrypts the query
IndexServer1
(2) Results {$N(p_1(a))$, $SWS(p_1(a))$, N(a.pdf), $p_1(a)$}
All ESWS returned decrypted by Client Tool
E(Results) {$EN(p_1(a))$, $ESWS(p_1(a))$, EN(a.pdf), $EC(p_1(a))$}
List of records containing ESWS for the encrypted query
(3) Query {$EN(p_1(a))$}
Client send the encrypted page names of the chosen pages by user to IndexServer2
IndexServer2
(4) Results {$N(p_1(a))$, $MRWS(p_1(a))$, N(a.pdf)}
Client Tool again decrypts the results so obtained
E(Results) {$EN(p_1(a))$, $EMRWS(p_1(a))$, EN(a.pdf)}
EMRWS for the query in Step 3 returned

Integrity Check: Results in 4 should be a subset of Results in 2

Experiment and Results

Software Hardware	CPU	Memory	HD
Client, IndexServer1, IndexServer2, CSS	2.53 GHz, 16 Cores	48 GB	2 Tb, 7200 RPM, 64 MB Cache

Tests performed both for indexing and query.

- **Index Time (in Table 1)** calculated from the time, client extracts text content to client gets response from IndexServer1.
- Indexing performed on 200 pdf, 200 doc and 200 ppt files.

	w/o encryp.	with encryp.
PDF	2.352	2.868
Doc	1.417	1.863
PPT	0.829	1.020

Table 1: Indexing Time

	w/o encryp.	with encryp.
OR	0.514	0.882
AND	0.160	0.194
OR+ AND	0.307	0.376

Table 2: Query Time

- **Query Time (in Table 2)** is calculating from Step 1 to Step 4 in the last figure.
- Query performed with Boolean Operators like OR, AND, NOT, Grouping of OR and AND.
- From Tables 1 and 2, we see that both our Index and Query Services are efficient.

Conclusion

Our EDTR System is well suitable as an enterprise cloud service, the most beneficial being Healthcare Services, where users need not worry about security issues and are willing to place their data in the Cloud.

References

[1] *Encryption Domain Text Retrieval System*, by Dr. Tzi Cker Chiueh, Alankar Saxena, Saurabh Bhola, Dilip N Simha, Dept of Comp. Sci., Stoney Brooke, NY, USA; Ping-Hung Lin and Cheng-en Pang, CCMA, ITRI, Taiwan

Presented by: **Alankar Saxena** and **Saurabh Bhola**, Computer Science and Engineering Department, IIT Bombay

We thank **Dr. Tzi Cker Chieuh** and **CCMA, ITRI, Republic of China(Taiwan)** for their support in our research