Integrity Preserving Multi-keyword Searchable Encryption for Cloud Computing

Reporter: Li Yuxi
Email: eliyuxi@gmail.com

Fucai Zhou (Northeastern University)
Yuxi Li (Northeastern University)
Alex X. Liu (Michigan State University)
Muqing Lin (Northeastern University)
Zifeng Xu (Northeastern University)

Outline

- **Motivation**
- **Our Contribution**
- **Definition and Security Model**
- **Integrity Preserving Multi-keyword Searchable Encryption Scheme**
  - Dynamic Searchable Encryption
  - Making Result Verifiable
- **Security Analysis**

Outline

- **Motivation**
- **Our Contribution**
- **Definition and Security Model**
- **Integrity Preserving Multi-keyword Searchable Encryption Scheme**
  - Dynamic Searchable Encryption
  - Making Result Verifiable
- **Security Analysis**

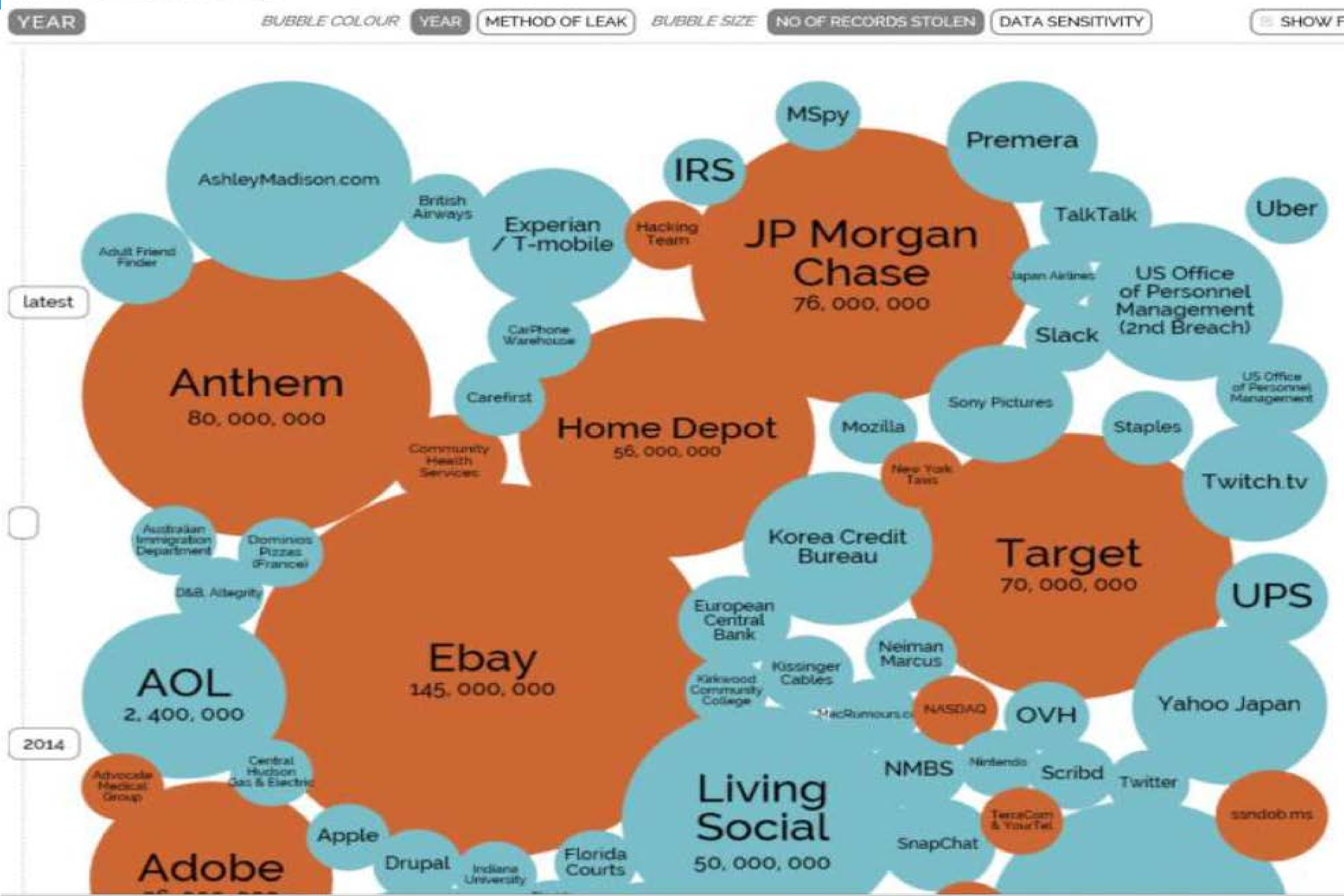
The Snowden disclosures

World's Biggest Data Breaches

Selected losses greater than 30,000 records

(updated 2nd October 2015)

interesting sto



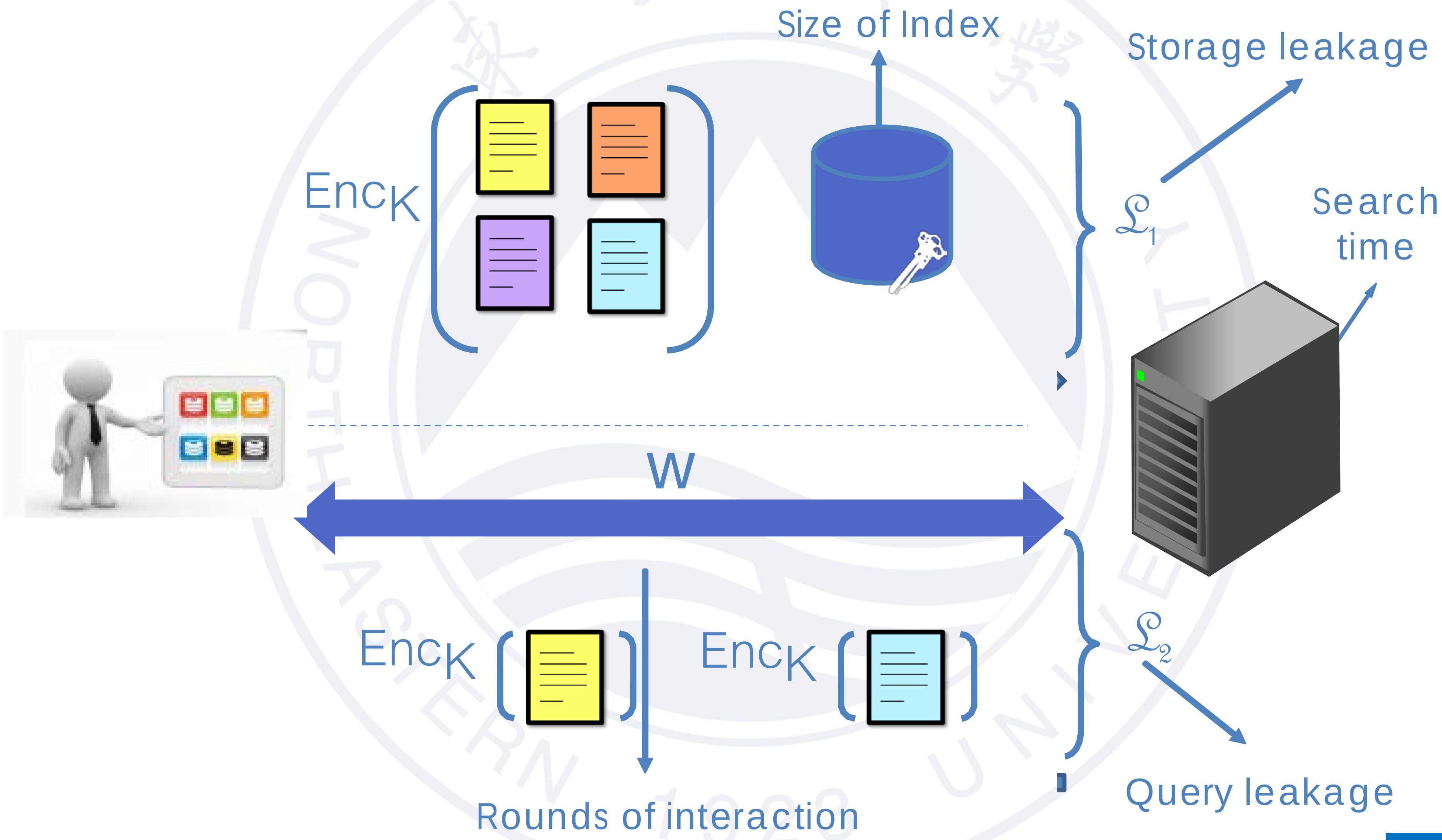
Motivation

- The data we produce—and is produced about us—is not properly secured
  - At best, data is encrypted "at rest" with the server's keys and decrypted upon use

Q: Why not encrypt it with your (data owner) own keys?
A: Utility, e.g. allow the cloud to search the data (e.g. gmail)

Can we keep the data encrypted and searchable too?

Motivation


Size of Index
Storage leakage
EncK
$\mathcal{L}_1$
Search time
W
EncK
EncK
$\mathcal{L}_2$
Query leakage
Rounds of interaction

- General Search
  - One the most basic computational operations
  - Since the 90’s, is arguably the most important functionality in information technology
- Searchable Encryption
  - Enhance end-to-end encrypted cloud storage, email and chat services with private search capabilities
  - In non end-to- end settings, add search to the encrypted back-end systems of cloud providers; support queries over databases that remain encrypted even in memory

All these applications would have a positive impact on the privacy and security of consumers and enterprises.

Motivation

- Drawbacks

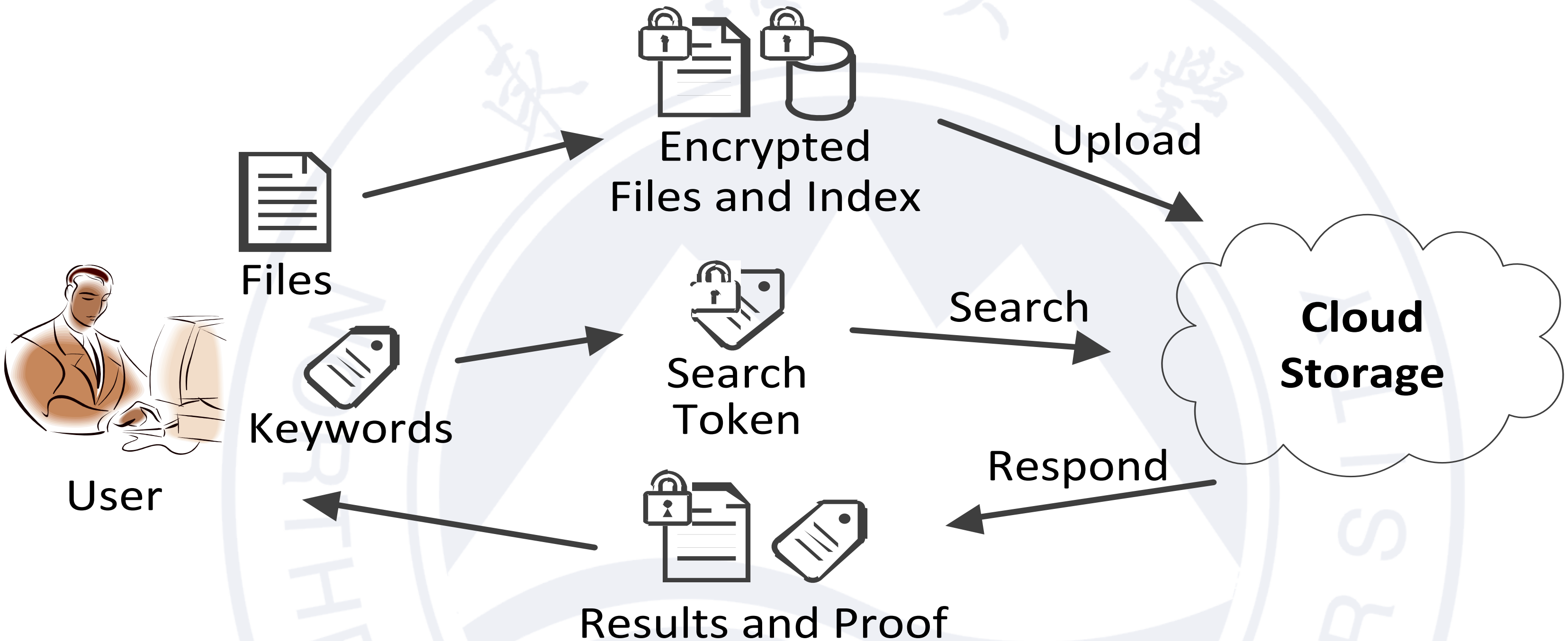
  1) The solutions are single-keyword oriented
     - Inefficient in practice since the searches may return a very large number of files
     - The communication complexity is linear

  2) Weak security model
     - Few works consider the searchable encryption and the search authentication together
       - Kamara et al. : a cryptographic cloud storage system
       - Kurosawa et al. : UC-security ; a verifiable SSE scheme

Even today, efficient integrity preserving multi-keyword search over encrypted data remains a challenging problem.

Outline

- Motivation
- **Our Contribution**
- Definition and Security Model
- Integrity Preserving Multi-keyword Searchable Encryption Scheme
  - Dynamic Searchable Encryption
  - Making Result Verifiable
- Security Analysis



- Our approach meet the following requirements:
  - The server is able to take multiple keywords as input, and give the final result directly;
  - For the server that honestly executes the search algorithm, a valid proof can be formed and pass the verification

Our Contribution

Basic Ideas

Dynamic Searchable Encryotion	Invertible index
	List-based search table
	Homomorphic encryption
Making Result Verifiable	Merkle Tree
	Bilinear map accumulator

Theoretical basis of proposed solution is inspired by **kamara's authenticated data structure** to verify set operations on outsourced sets.

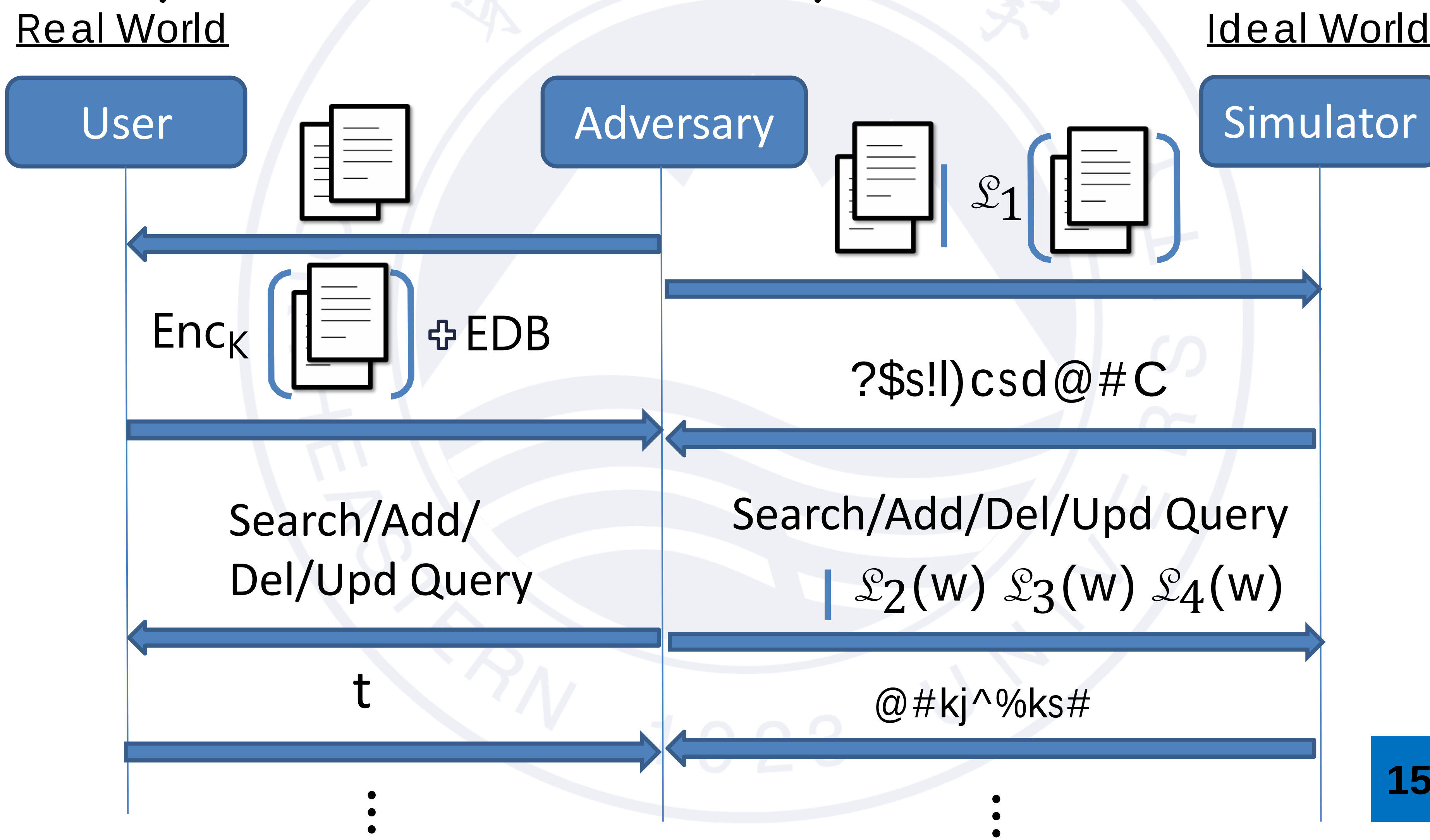
Reference: S. Kamara, C. Papamanthou, and T. Roeder, "CS2: A Searchable Cryptographic Cloud Storage System," TechReport MSR-TR-2011-58, Microsoft Research, 2011.

Outline

- Motivation
- Our Contribution
- **Definition and Security Model**
- Integrity Preserving Multi-keyword Searchable Encryption Scheme
  - Dynamic Searchable Encryption
  - Making Result Verifiable
- Security Analysis

- A dynamic MSE scheme is a tuple of polynomial-time algorithms and protocols such that:
  - $K \leftarrow \mathsf{Gen}(1^k)$
  - $(\gamma, \boldsymbol{c}, st, \alpha) \leftarrow \mathsf{Setup}(K, \delta, \mathbf{f})$
  - $\tau_s \leftarrow \mathsf{SrchToken}(K, W)$
  - $(\mathbf{I}_W, \pi) \leftarrow \mathsf{Search}(\gamma, \boldsymbol{c}, \tau_s, \alpha)$
  - $b \leftarrow \mathsf{Verify}(K, st, \tau_s, \mathbf{I}', \boldsymbol{\pi})$
  - $f \leftarrow \mathsf{Dec}(K, \boldsymbol{c})$
  - $(U{:}\,st'; S{:}\,\gamma', \mathbf{c}', \alpha') \leftarrow \mathsf{Add/Update}(U{:}\,K, \delta, f, st; S{:}\,\gamma, \boldsymbol{c}, \alpha)$

- Dynamic CKA2-Security



- Dynamic CKA2-secure
  - Game_real

$$K \leftarrow \mathsf{Gen}(1^k)$$
$$(\delta, \mathbf{f}) \leftarrow \mathcal{A}(1^k)$$
$$(\gamma, \mathbf{c}, st, \alpha) \leftarrow \mathsf{Setup}(K, \delta, \mathbf{f})$$
$$\text{for } 1 \le i \le q$$
$$\{W_i, f_i, f_i'\} \xleftarrow{\text{one query each time}} \mathcal{A}(\alpha, \gamma, \mathbf{c}, \tau_1, \ldots \tau_{i-1}, c_1, \ldots c_{i-1})$$
$$\tau_i \xleftarrow{\mathcal{A}} \mathsf{SrchToken}(K, W_i), \text{ or}$$
$$(U : st'; \mathcal{A} : \tau_i, c_i) \xleftarrow{\mathcal{A}} \mathsf{Add/Update}(U : K, \delta_f, f, st; \mathcal{A}), \text{or}$$
$$(U : st'; \mathcal{A} : \tau_i) \xleftarrow{\mathcal{A}} \mathsf{Del/Update}(U : K, \delta_f, f, st; \mathcal{A})$$
$$\text{output } b \leftarrow \mathcal{A}(\alpha, \gamma, \mathbf{c}, \tau_1, \ldots, \tau_q, c_1, \ldots c_q)$$

- Dynamic CKA2-secure
  - Game_ideal

$$
\begin{aligned}
&(\delta,\mathbf{f}) \leftarrow \mathcal{A}(1^k) \\
&\quad (\tilde{\alpha},\tilde{\gamma},\tilde{\mathbf{c}}) \leftarrow \mathcal{S}^{\mathcal{L}_1(\delta,\mathbf{f})}(1^k) \\
&\quad \text{for } 1 \le i \le q \\
&\qquad \{W_i, f_i, f_i'\} \xleftarrow{\text{one query each time}} \mathcal{A}(\tilde{\alpha},\tilde{\gamma},\tilde{\mathbf{c}},\tilde{\tau}_1,\ldots\tilde{\tau}_{i-1},\tilde{c}_1,\ldots\tilde{c}_{i-1}) \\
&\qquad \tilde{\tau}_i \xleftarrow{\mathcal{A}} \mathcal{S}^{\mathcal{L}_2(\delta,\mathbf{f},W_i)}(1^k), \text{ or} \\
&\qquad (\mathcal{S}: st'; \mathcal{A}: \tilde{\tau}_i, \tilde{c}_i) \xleftarrow{\mathcal{A}} \text{Add/Update}(\mathcal{S}^{\mathcal{L}_3(\delta,\mathbf{f},f_i)}(1^k); \mathcal{A}), \text{or} \\
&\qquad (\mathcal{S}: st'; \mathcal{A}: \tilde{\tau}_i) \xleftarrow{\mathcal{A}} \text{Del/Update}(\mathcal{S}^{\mathcal{L}_4(\delta,\mathbf{f},f_i')}(1^k); \mathcal{A}) \\
&\quad \text{output } b \leftarrow \mathcal{A}(\tilde{\alpha},\tilde{\gamma},\tilde{\mathbf{c}},\tilde{\tau}_1,\ldots,\tilde{\tau}_q,\tilde{c}_1,\ldots\tilde{c}_q)
\end{aligned}
$$

- Dynamic CKA2-secure
  - Dynamic CKA2-secure is satisfied if there exists a simulator such that the real game ≈ the ideal game

Formal definition:

$$\left|\mathbf{Pr}\left[\mathrm{Real}_{\mathcal{A}}(1^k)=1\right]-\mathbf{Pr}\left[\mathrm{Ideal}_{\mathcal{A},\mathcal{S}}(1^k)=1\right]\right|\le \mathbf{negl}(1^k)$$

- Unforgeability
  - Game_forge

$K \leftarrow \text{Gen}(1^k)$

$(\delta, \mathbf{f}) \leftarrow \mathcal{A}(1^k)$

$(\gamma, \mathbf{c}, st, \alpha) \leftarrow \text{Setup}(K, \delta, \mathbf{f})$

for $1 \le i \le q$

$$\{W_i, f_i, f_i'\} \xleftarrow{\text{one query each time}} \mathcal{A}(\alpha, \gamma, \mathbf{c}, \tau_1, \ldots \tau_{i-1}, c_1, \ldots c_{i-1})$$

$$\tau_i \xleftarrow{\mathcal{A}} \text{SrchToken}(K, W_i), \text{ or}$$

$$(\tau_i, c_i) \xleftarrow{\mathcal{A}} \text{Add/Update}(U : K, \delta_{f_i}, f_i, st; \mathcal{A}), \text{ or}$$

$$\tau_i \xleftarrow{\mathcal{A}} \text{Del/Update}(U : K, \delta_{f_i'}, f_i', st; \mathcal{A})$$

$$\boxed{(W, \mathbf{I}', \pi) \leftarrow \mathcal{A}(\alpha, \gamma, \mathbf{c}, \tau_1, \ldots, \tau_q, c_1, \ldots c_q)}$$

$\tau_s \leftarrow \text{SrchToken}(K, W)$

output $b \leftarrow \text{Verify}(K, st', \tau_s, \mathbf{I}', \pi)$

- Unforgeability
  - The unforgeability requires that, all PPT adversaries have at most negligible probability to let the game output 1.

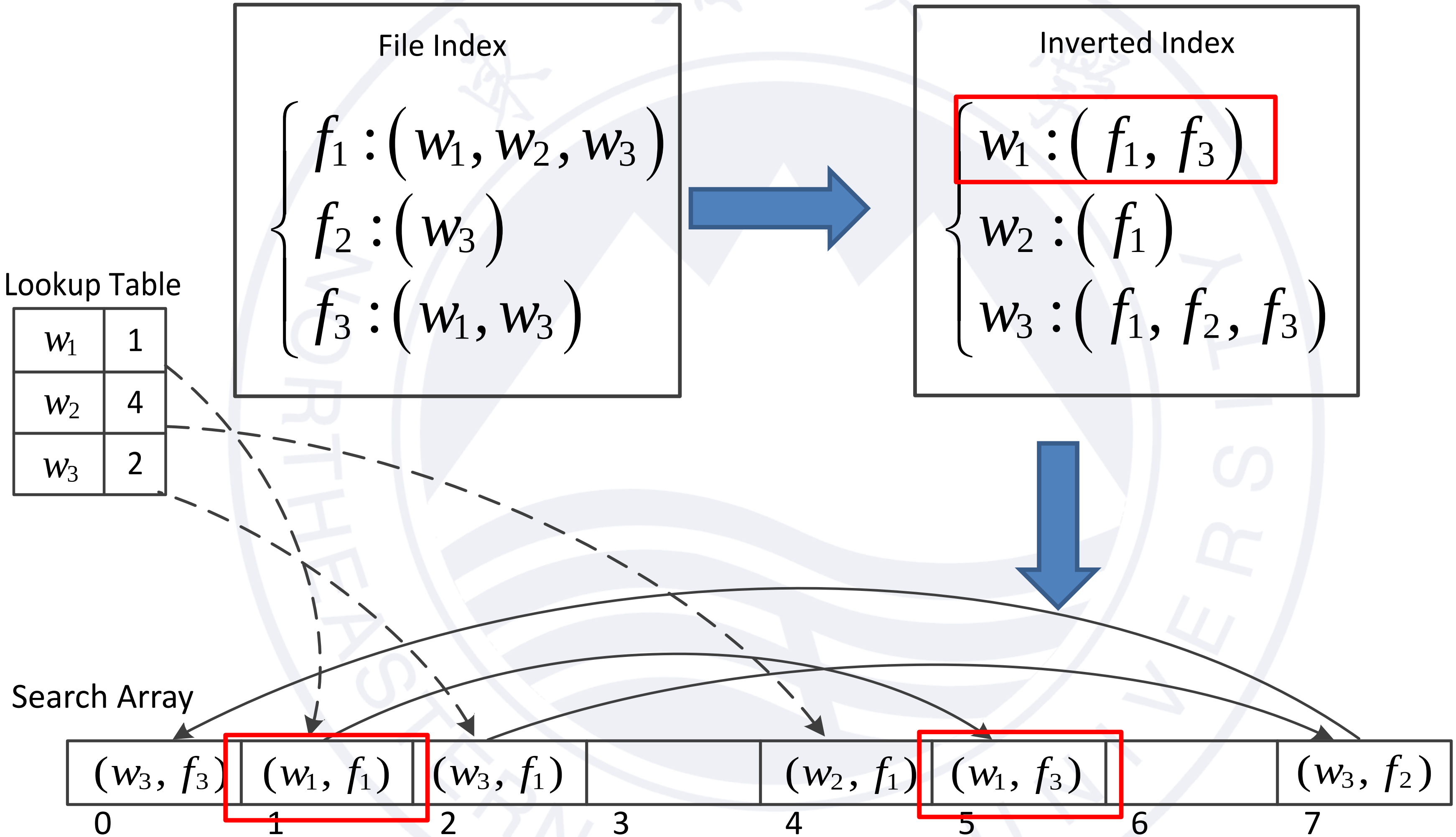
Formal definition:

$$\mathbf{Pr}[\mathsf{Forge}_{\mathcal{A}}(1^k) = 1] \le \mathbf{negl}(1^k)$$

Outline

- Motivation
- Our Contribution
- Definition and Security Model
- **Integrity Preserving Multi-keyword Searchable Encryption Scheme**
  - Dynamic Searchable Encryption
  - Making Result Verifiable
- Security Analysis

Our Scheme


File Index
$f_1 : (w_1, w_2, w_3)$
$f_2 : (w_3)$
$f_3 : (w_1, w_3)$
Inverted Index
$w_1 : (f_1, f_3)$
$w_2 : (f_1)$
$w_3 : (f_1, f_2, f_3)$
Lookup Table
w_1 1
w_2 4
w_3 2
Search Array
(w_3, f_3)
(w_1, f_1)
(w_3, f_1)
(w_2, f_1)
(w_1, f_3)
(w_3, f_2)
0
1
2
3
4
5
6
7

Our Scheme

$$w_2 \rightarrow (F(w_2), G(w_2), P(w_2))$$

Lookup Table

Index	Value
$F(w_1)$	$(2) \oplus G(w_1)$
$F(w_2)$	$(7) \oplus G(w_2)$
$F(w_3)$	$(8) \oplus G(w_3)$

w_3	8

Search Array

	(pre, next, id)
0	

	$(p,n,id) \oplus H_1(P(w),r)$	r
0		
1		
2	$(0,0,id(f_1)) \oplus H(P(w_1),r_1)$	r_1
3		
4	$(7,0,id(f_2)) \oplus H(P(w_2),r_2)$	r_2
5		
6		
7	$(0,4,id(f_1)) \oplus H(P(w_2),r_3)$	r_3
8	$(0,9,id(f_2)) \oplus H(P(w_3),r_4)$	r_4
9	$(8,0,id(f_3)) \oplus H(P(w_3),r_5)$	r_5

- Multi-keyword Searchable Encryption
  - Keywords to search $W = \{w_1, \ldots, w_n\}$
  - Single keyword search result : $S_1, \ldots, S_n$
  - The final search result

$$I_W = S_1 \cap S_2 \cap \ldots \cap S_n$$

Q: How to make result verifiable?

The bilinear-map accumulator

The correct intersection is equivalent to

- The subset condition

$$I \subseteq S_1 \wedge I \subseteq S_2 \wedge \ldots \wedge I \subseteq S_n$$

- The completeness condition

$$(S_1 - I) \cap (S_2 - I) \cap \ldots \cap (S_n - I) = \varnothing$$

Reference : C. Papamanthou, R. Tamassia, and N. Triandopoulos, "Optimal verification of operations on dynamic sets," Advances in Cryptology-CRYPTO 2011, Springer Berlin Heidelberg, pp. 91-110, 2011.

- Main Idea (1)
  - Computes the accumulated value for each set S_i,
  - And construct a Merkle tree using these values :

$$\theta_w = \left\langle F_{K_1}(w), g^{\prod_{f \in \mathbf{f}_w}(s+id(f))} \right\rangle$$

  - In a search, the server returns a file set I, the accumulated value for each node, and a Merkle tree proof to this set

- Main Idea (2)
  - For every S_i , form the polynomial :

$$P_i = \prod_{f \in S_i - \mathbf{I}_W} (s + id(f))$$

  - Send user the subset witness :

$$\mathcal{S} = \{ g^{P_1}, \ldots, g^{P_n} \}$$

  - User perform the subset condition verification by checking :

$$e\left(g^{\prod_{id_k \in \mathbf{I}_W} (s + id_k)}, g^{P_i} \right) = e(\theta_{i,2}, g)$$

- Main Idea (3)
  - Based on $P_1, \ldots, P_n$, use the extended Euclidean algorithm over polynomials to get $q_1, \ldots, q_n$, satisfies :

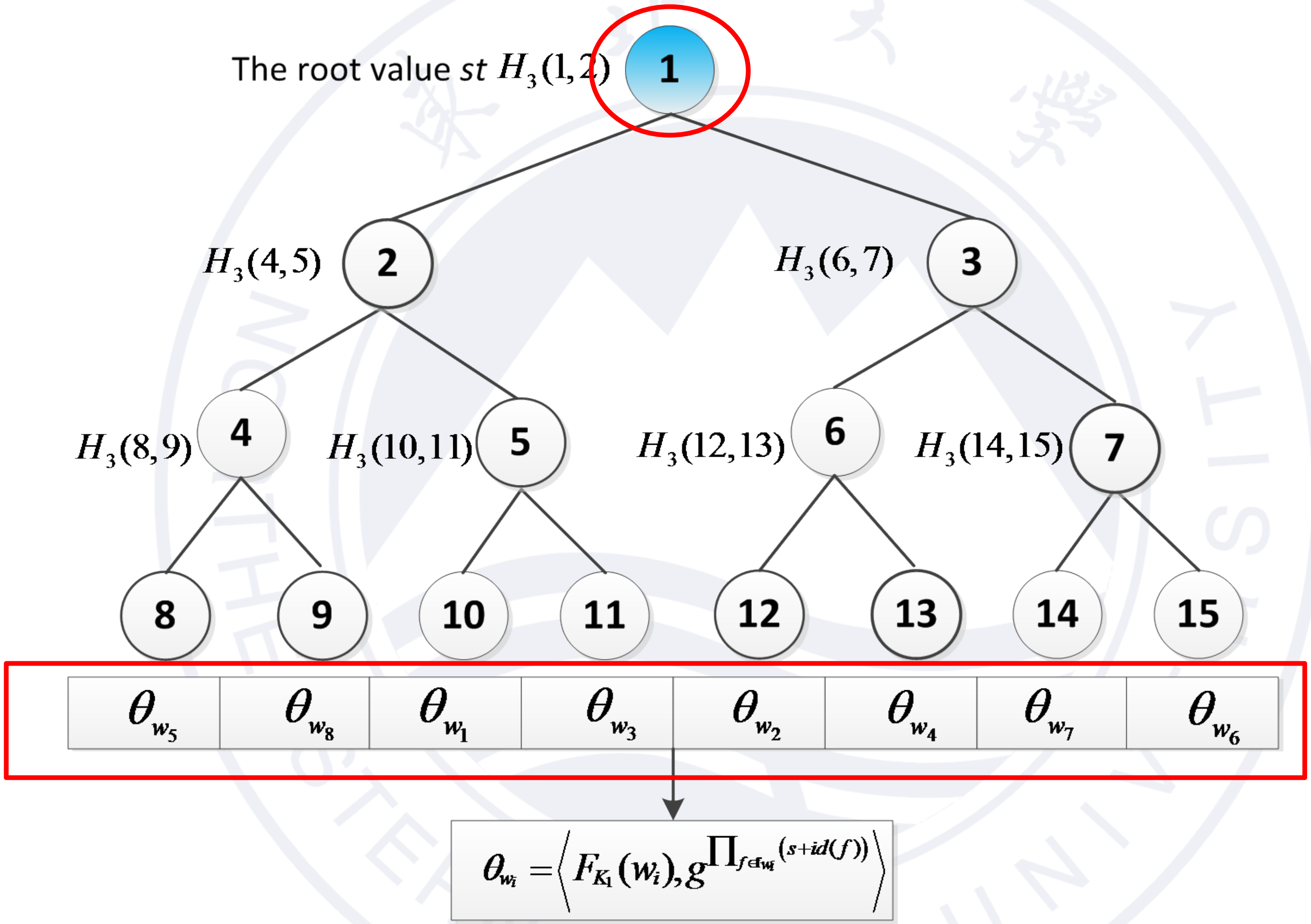
$$q_1 P_1 + q_2 P_2 + \cdots + q_n P_n = 1$$

  - Send user the completeness witness

$$C = \{g^{q_1}, \ldots, g^{q_n}\}$$

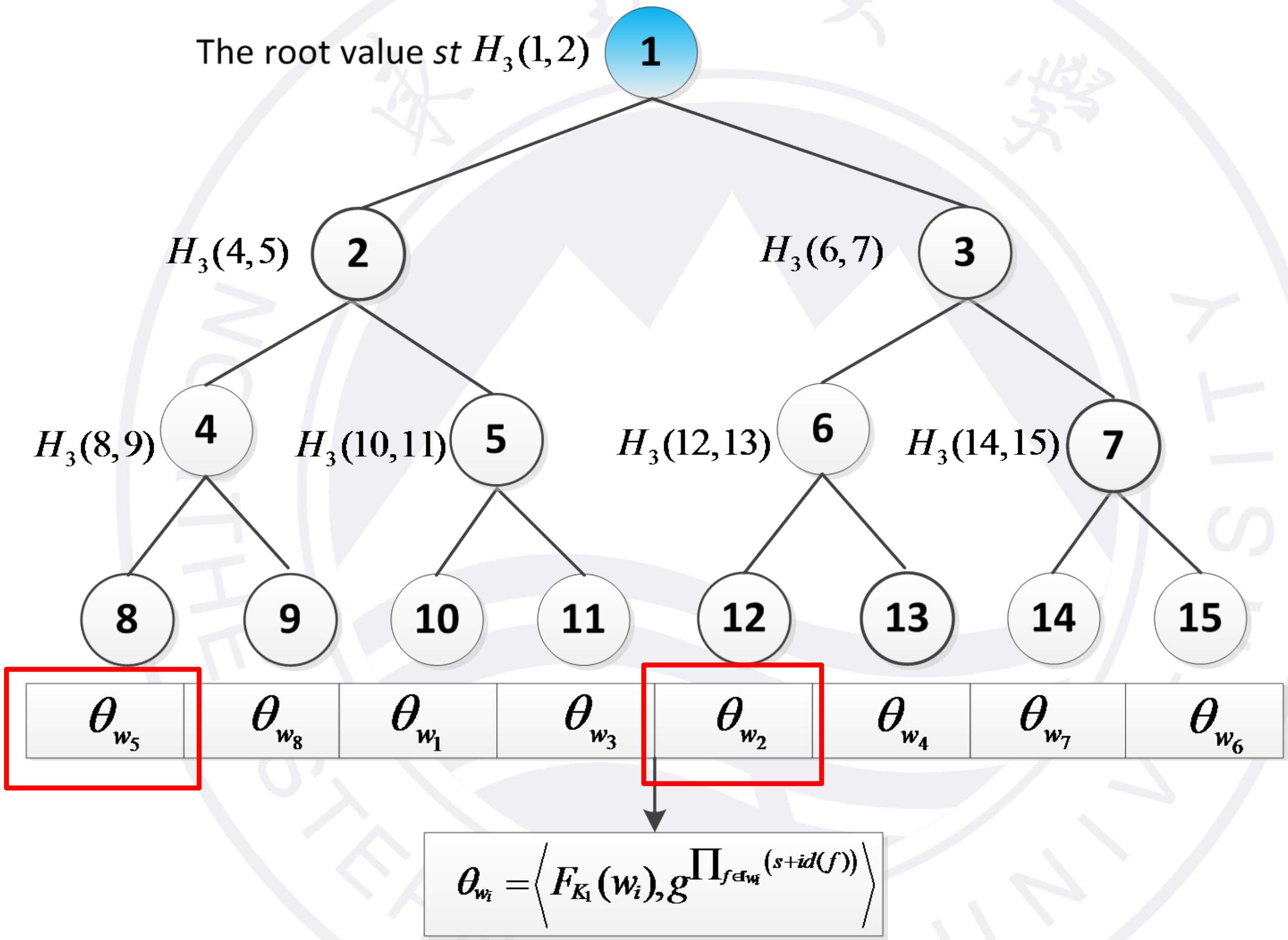
  - User verify the completeness condition by checking

$$\prod_{i=1}^{n} e(g^{P_i}, g^{q_i}) = e(g, g)$$

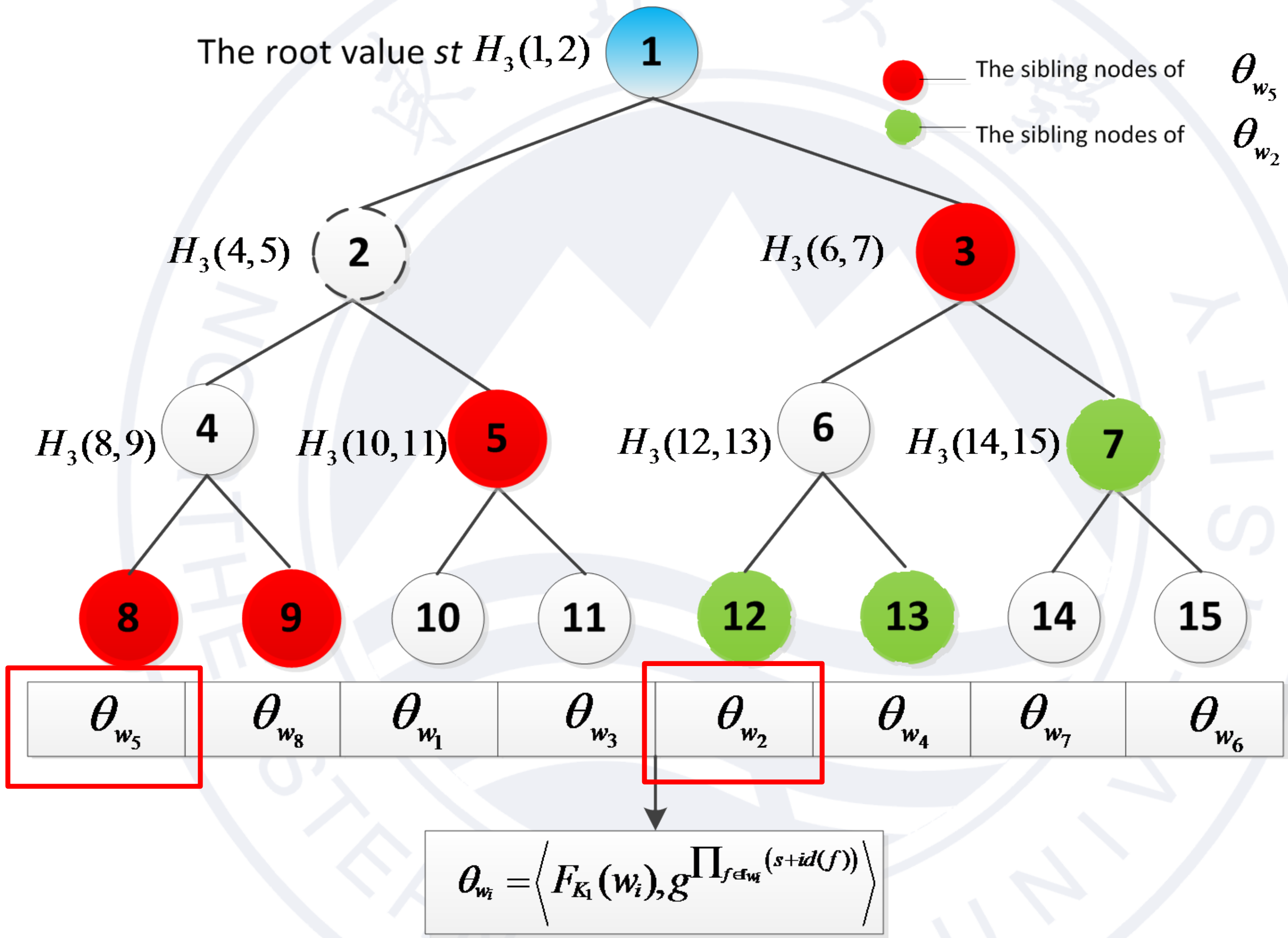

The root value st $H_3(1,2)$
1
$H_3(4,5)$
2
$H_3(6,7)$
3
$H_3(8,9)$
4
$H_3(10,11)$
5
$H_3(12,13)$
6
$H_3(14,15)$
7
8
9
10
11
12
13
14
15
θ_{w_5}
θ_{w_8}
θ_{w_1}
θ_{w_3}
θ_{w_2}
θ_{w_4}
θ_{w_7}
θ_{w_6}
$\theta_{w_i} = \left\langle F_{K_1}(w_i), g^{\prod_{f \in f_{w_i}}(s+id(f))} \right\rangle$

Our Scheme

Making Result Verifiable



Making Result Verifiable



Outline

- Motivation
- Our Contribution
- Definition and Security Model
- Integrity Preserving Multi-keyword Searchable Encryption Scheme
  - Dynamic Searchable Encryption
  - Making Result Verifiable
- **Security Analysis**

- Definition 1 (Dynamic CKA2-secure)

Real game | Ideal game

  - Dynamic CKA2-secure is satisfied if there exists a simulator such that the real game ≈ the ideal game

Theorem 1. *If the SKE scheme is CPA-secure, and F G P are pseudo-random functions, then the dynamic MSE scheme is secure against adaptive chosen-keyword attacks in the random oracle model.*

- Proof Sketch
  - Leakage functions

$$\mathcal{L}_1(\delta,\mathbf{f})=\left(\#A_s,\left[id(w)\right]_{w\in\mathbf{w}},\left[id(f)\right]_{f\in\mathbf{f}},\left[|f|\right]_{f\in\mathbf{f}}\right)$$

$$\mathcal{L}_2(\delta,\mathbf{f},W)=\left([id(f)]_{f\in\mathbf{f}_w},id(w)\right)_{\text{for all } w\in W}$$

$$\mathcal{L}_3(\delta,\mathbf{f},f)=\left(id(f),\left[id(w),\text{apprs}(w)\right]_{w\in\mathbf{w}_f},|f|\right)$$

$$\mathcal{L}_4(\delta,\mathbf{f},f)=\left(id(f),\left[id(w),\text{prev}(f,w),\text{next}(f,w)\right]_{w\in\mathbf{w}_f}\right)$$

Our goal is to prove that, any PPT adversary can obtain no information about the data and queries, except the information in the leakage functions.

- Definition 2 (Unforgeability)
  - Game_forge
    - A interacts with a user that honestly executes the scheme。
    - After making polynomial times queries, the adversary produces a set of keywords, a wrong search result and a proof to this result.
    - If these outputs pass the users verification algorithm, the game outputs 1, otherwise it outputs 0.
  - If A cannot win, then the scheme holds Unforgeability

Theorem 2. *If H3 is collision-resistant hash function and the bilinear q-SDH assumption holds then the dynamic MSE scheme is unforgeable*

- Proof Sketch
  - if there exists a PPT adversary A such that ,

$$\text{Forge}_{\mathcal{A}}(1^k) = 1$$

then there exist a PPT simulator S that breaks at least one of the assumptions blew:

- The collision-resistance property of H3
- Bilinear *q*-SDH assumption.

Conclusion

- A dynamic integrity preserving multi-keyword searchable encryption scheme
  - Enabling search authentication in multi-keyword searchable encryption schemes
  - Secure against the adaptive chosen-keyword attack
  - Unforgeability

Thank you!

Reporter: Yuxi Li
Email: eliyuxi@gmail.com