Duality in ABE:

Converting Attribute Based Encryption for Dual Predicate and Dual Policy via Computational Encodings

Nuttapong Attrapadung, Shota Yamada
AISTf, Japan

@CT-RSA 2015

Our Main Results in One Slide

Generic dual conversion for ABE

Instantiations:

The first fully secure

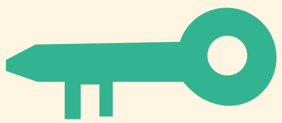
- CP-ABE with short key
- CP-ABE all-unbounded

(for boolean formulae, span programs)

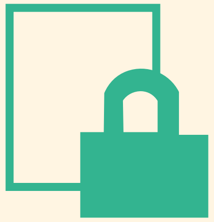
1 Introduction

Attribute Based Encryption (ABE)

ABE for predicate $R: X \times Y \rightarrow \{0,1\}$



Key for
$x \in X$



Ciphertext for
$y \in Y$
(encrypt m)

Decrypt

m if $R(x,y)=1$
? if $R(x,y)=0$

A Predicate

$R: X \times Y \to \{0,1\}$

Its Dual Predicate

$\bar{R}: Y \times X \to \{0,1\}$

$\bar{R}(y,x) := R(x,y)$

Key-Policy ABE

$R: P \times A \rightarrow \{0,1\}$

Key for policy p∈P

Ciphertext for attribute a∈A

R(p,a) iff p satisfies a

Ciphertext-Policy ABE

$\bar{R}: A \times P \rightarrow \{0,1\}$

Key for attribute a∈A

Ciphertext for policy p∈P

$\bar{R}$(a,p) iff p satisfies a

Motivation

- KP-ABE, CP-ABE
  - Definitions: directly related.
  - Constructions: NO known relation.
- **Can we generically convert an ABE to its dual?**
  - So that we would only construct KP, and get also CP.
  - Might be difficult? Historically, CP [BSW07, Waters11] was harder to achieve than KP [GPSW06].

Related Work for Dual Conversion

- Converting KP-ABE for boolean formulae predicate
  - Small classes of predicates
  - Its dual CP: only for bounded-size formulae [GJPS08].
- Converting KP-ABE for all boolean circuits
  - Implies general predicates, but must start with ABE for circuits.
  - Its dual CP: only for bounded-size circuits [GGHSW13].
    - Due to the use of universal circuits.
- Summary: **less expressivity**, and much **less efficient.** ☹

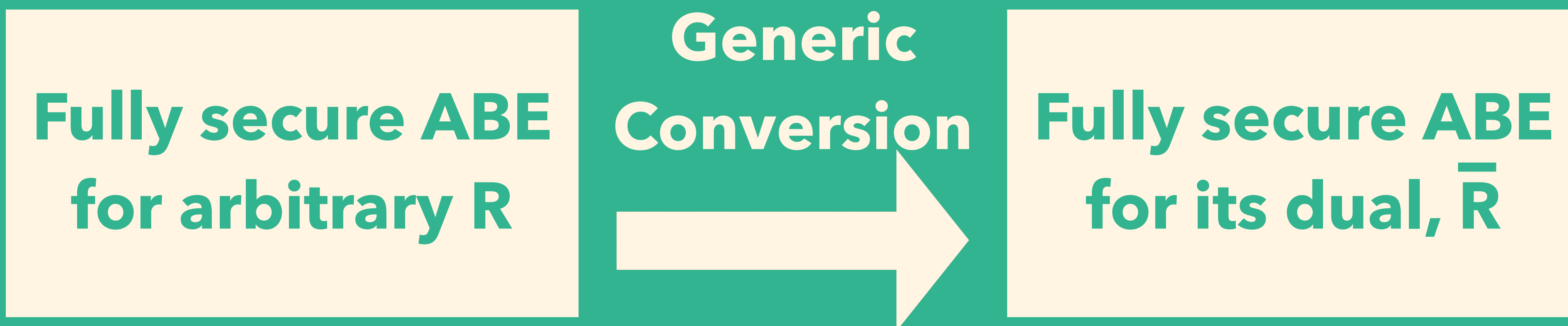
Our Focus

- Goal: Generic dual conversion for any predicate.
  - Preserving full security, expressivity, and efficiency.
- Tool: Use a generic ABE framework of [A14].
  - An abstraction of dual-system encryption [Waters09] for achieving fully-secure ABE.

[A14] N. Attrapadung, "Dual System Encryption via Doubly Selective Security: Framework, Fully-secure Functional Encryption for Regular Languages, and More", *Eurocrypt 2014*.

2 Our Result Overview

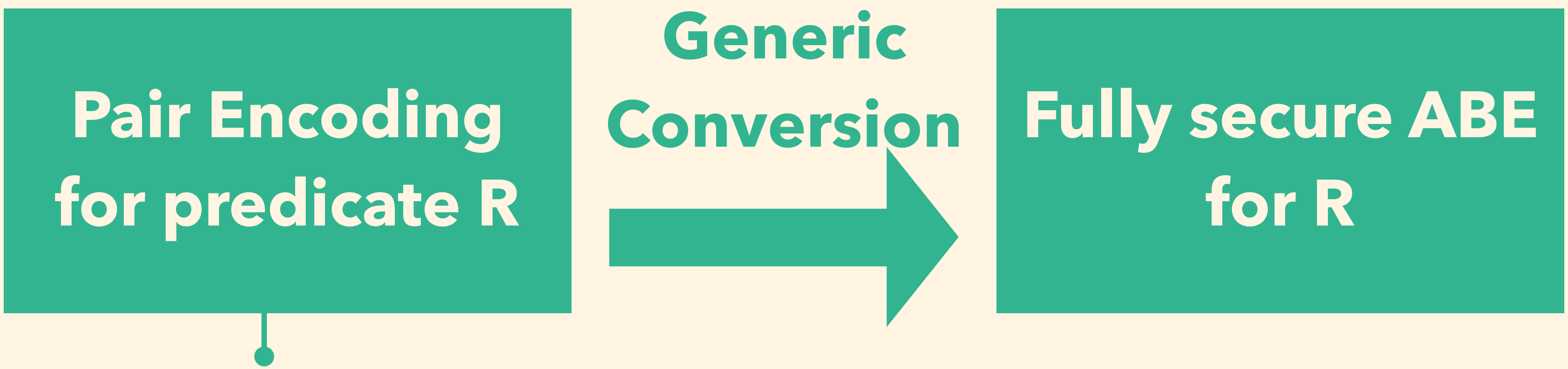
Our Main Result: Dual Conversion



Restricted to ABE In the “pair encoding” framework [A14].

Recall The “Pair Encoding” Framework

Main Theorem in [A14]



If pair encoding is

- **“Perfectly secure” or**
- **“Doubly selectively secure”.**

Our Main Result: More Precisely

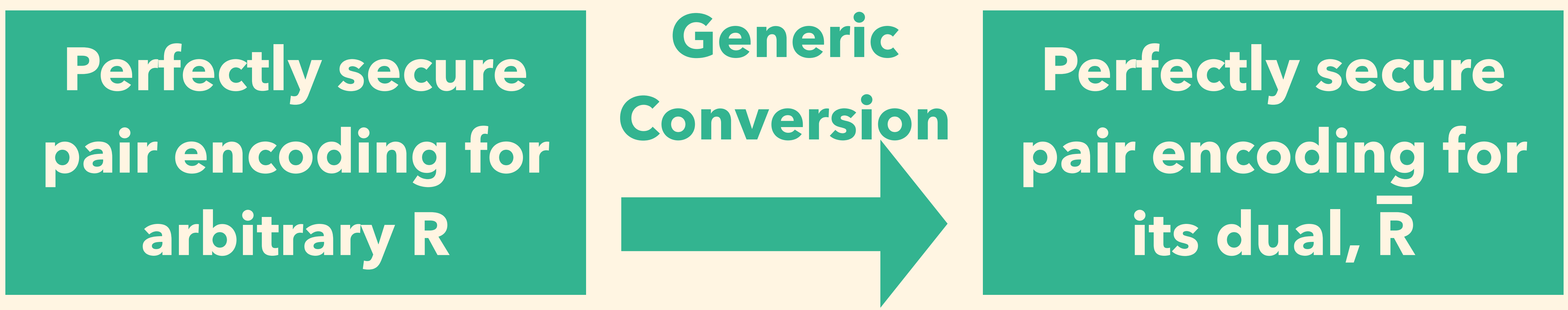
Doubly selective pair encoding for arbitrary R

Generic Conversion →

Doubly selective pair encoding for its dual, $\overline{R}$

The Only Previous Dual Conversion

A Side Result in [A14]



Implications: Solving Open Problems

No fully-secure ABE known before

[NEW! all implied by this work]

- CP unbounded boolean formula
- CP short-key for boolean formula
- CP over doubly-spatial

Doubly selective encodings known

- KP unbounded boolean formula
- KP short-ciphertext for boolean formula
- KP over doubly-spatial
- KP regular languages
- CP regular languages

[all in A14]

Perfectly secure encodings known

- KP, CP boolean formula with some bounds [LOSTW10, W14, A14]
- spatial, inner-product, …

3 Recall Pair Encoding

Recall Pair Encoding and ABE [A14]

Pair Encoding for R	$\longrightarrow$	ABE for R
Param $\longrightarrow$ $\boldsymbol{h}$		$PK=(g_1^{\boldsymbol{h}}, e(g_1,g_1)^{a})$, $MSK=a$
Enc1(x) $\longrightarrow$ $\boldsymbol{k}_x(a,\boldsymbol{r},\boldsymbol{h})$		$SK=g_1^{\boldsymbol{k}_x(a,\boldsymbol{r},\boldsymbol{h})}$
Enc2(y) $\longrightarrow$ $\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h})$		$CT=(g_1^{\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h})}, e(g_1,g_1)^{as_0}M)$
Pair($\boldsymbol{k}_x,\boldsymbol{c}_y$) $\longrightarrow$ as_0 if $R(x,y)=1$		$Dec \longrightarrow e(g_1,g_1)^{as_0}$ if $R(x,y)=1$

- s_0 = first entry in $\boldsymbol{s}$.
- Require some linearity properties.

- Use composite-order bilinear groups.
- (Neglect details here).

Security Definitions of Pair Encoding

Perfect security

Identical (info-theoretic) $\begin{cases} \boldsymbol{k}_x(0, \boldsymbol{r}, \boldsymbol{h}) \\ \boldsymbol{k}_x(a, \boldsymbol{r}, \boldsymbol{h}) \end{cases}$ $\quad \boldsymbol{c}_y(\boldsymbol{s}, \boldsymbol{h})$ for $R(x,y)=0$

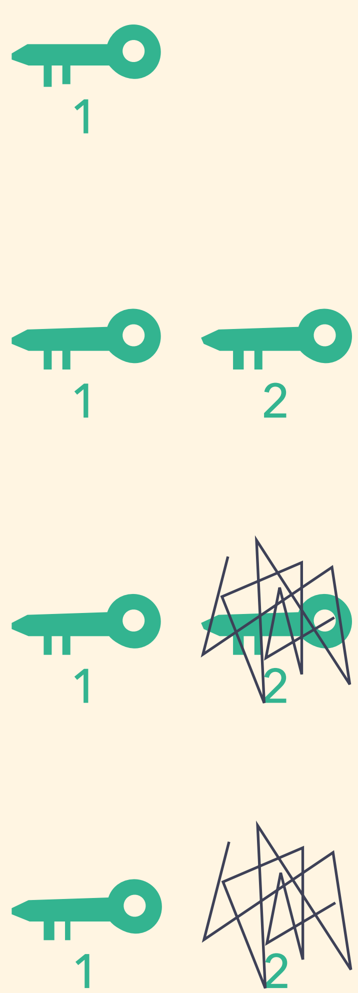
Doubly selective security

Cannot distinguish ⟵ $\begin{cases} g_2^{\boldsymbol{k}_x(0, \boldsymbol{r}, \boldsymbol{h})} \\ g_2^{\boldsymbol{k}_x(a, \boldsymbol{r}, \boldsymbol{h})} \end{cases}$ $\quad g_2^{\boldsymbol{c}_y(\boldsymbol{s}, \boldsymbol{h})}$ for $R(x,y)=0$

- **Selective notion:** queries ***c*** before ***k***. (then)
- **Co-selective notion:** queries ***k*** before ***c***. (then)

Intuition Behind Pair Encoding Security

Switch Keys from Normal to Semi-functional [A14]



normal: K — $g_1^{\mathbf{k}(a,\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\mathbf{0},\mathbf{0})}$

↓ Subgroup Decision

semi-1: K1 — $g_1^{\mathbf{k}(a,\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(0,\hat{\mathbf{r}},\hat{\mathbf{h}})}$

↓ Security of encoding

semi-2: K2 — $g_1^{\mathbf{k}(a,\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{a},\hat{\mathbf{r}},\hat{\mathbf{h}})}$

↓ Subgroup Decision

semi-3: K3 — $g_1^{\mathbf{k}(a,\mathbf{r},\mathbf{h})} \cdot g_2^{\mathbf{k}(\hat{a},\mathbf{0},\mathbf{0})}$

- Only for self-containment, will not use here.

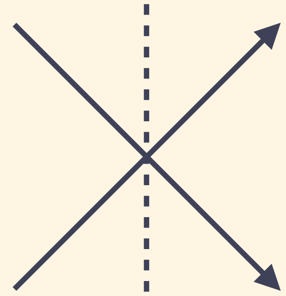
4 Our Conversion

Basic Idea for Dual Conversion

Encoding for R

Enc1 maps $x \in X$

Enc2 maps $y \in Y$



$\overline{\textbf{Encoding}}$ **for** $\overline{R}$

$\overline{\text{Enc1}}$ maps $y \in Y$ defined using Enc2

$\overline{\text{Enc2}}$ maps $x \in X$ defined using Enc1

Our Dual Conversion

Encoding for R

$\text{Param} \longrightarrow \boldsymbol{h}$

$\text{Enc1} \longrightarrow \boldsymbol{k}_x(a, \boldsymbol{r}, \boldsymbol{h})$

$\text{Enc2} \longrightarrow \boldsymbol{c}_y(\boldsymbol{s}, \boldsymbol{h})$

$\overline{\textbf{Encoding}}$ **for** $\bar{R}$

$\overline{\text{Param}} \longrightarrow \bar{\boldsymbol{h}} = (\boldsymbol{h}, \bar{b})$

$\overline{\text{Enc1}} \longrightarrow \bar{\boldsymbol{k}}_y(\bar{a}, \bar{\boldsymbol{s}}, \bar{\boldsymbol{h}}) = (\boldsymbol{c}_y(\boldsymbol{s}, \boldsymbol{h}),\ \bar{a} + \bar{b} s_0)$

$\overline{\text{Enc2}} \longrightarrow \bar{\boldsymbol{c}}_x(\bar{\boldsymbol{r}}, \bar{\boldsymbol{h}}) = (\boldsymbol{k}_x(\bar{b} \bar{s}_0, \boldsymbol{r}, \boldsymbol{h}),\ \bar{s}_0)$

where $\bar{\boldsymbol{s}} = \boldsymbol{s}$ $\quad \bar{\boldsymbol{r}} = (\bar{s}_0, \boldsymbol{r})$

Our Dual Conversion

Encoding for R

$\mathsf{Param} \rightarrow \boldsymbol{h}$

$\mathsf{Enc1} \rightarrow \boldsymbol{k}_x(a, \boldsymbol{r}, \boldsymbol{h})$

$\mathsf{Enc2} \rightarrow \boldsymbol{c}_y(\boldsymbol{s}, \boldsymbol{h})$

$\mathsf{Pair}(\boldsymbol{k}_x, \boldsymbol{c}_y) = as_0$

$\overline{\text{Encoding}}$ for $\overline{R}$

$\overline{\mathsf{Param}} \rightarrow \overline{\boldsymbol{h}} = (\boldsymbol{h}, \overline{b})$

$\overline{\mathsf{Enc1}} \rightarrow \overline{\boldsymbol{k}}_y(\overline{a}, \overline{\boldsymbol{s}}, \overline{\boldsymbol{h}}) = (\boldsymbol{c}_y(\boldsymbol{s}, \boldsymbol{h}),\ \overline{a} + \overline{b}s_0)$

$\overline{\mathsf{Enc2}} \rightarrow \overline{\boldsymbol{c}}_x(\overline{\boldsymbol{r}}, \overline{\boldsymbol{h}}) = (\boldsymbol{k}_x(\overline{b}\overline{s}_0, \boldsymbol{r}, \boldsymbol{h}),\ \overline{s}_0)$

$\overline{\mathsf{Pair}}$: $\mathsf{Pair}(\boldsymbol{k}_x, \boldsymbol{c}_y) = \overline{b}\overline{s}_0 s_0$

$(\overline{a} + \overline{b}s_0)(\overline{s}_0) - \overline{b}\overline{s}_0 s_0 = \overline{a}\overline{s}_0$

where $\overline{\boldsymbol{s}} = \boldsymbol{s}$ $\quad \overline{\boldsymbol{r}} = (\overline{s}_0, \boldsymbol{r})$

Our Dual Conversion

- The same conversion as in [A14].
- [A14] only proved for the **perfectly secure encodings.**
- We make it work also for **doubly secure encodings.**

Our New Theorems

Encoding for R is selective then → **$\overline{\text{Encoding}}$ for $\overline{R}$ is co-selective**

Encoding for R is co-selective then → **$\overline{\text{Encoding}}$ for $\overline{R}$ is selective**

Intuition:

- Swap key/cipher encodings $\rightarrow$ Query order is reversed.
- Hence selective becomes co-selective (and vice versa).

Difficulty in Proving Theorems

Encoding for R | $\overline{\textbf{Encoding for } R}$

Enc1: $\boldsymbol{k}_x(a,\boldsymbol{r},\boldsymbol{h})$

Enc2: $\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h})$

$\overline{\text{Enc1}}$: $\overline{\boldsymbol{k}}_y(\overline{a},\overline{\boldsymbol{s}},\overline{\boldsymbol{h}}) = (\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h}),\ \overline{a}+\overline{b}s_0)$

$\overline{\text{Enc2}}$: $\overline{\boldsymbol{c}}_x(\overline{\boldsymbol{r}},\overline{\boldsymbol{h}}) = (\boldsymbol{k}_x(\overline{b}\overline{s_0},\boldsymbol{r},\boldsymbol{h}),\ \overline{s_0})$

(all terms over the exponent)

Difficulty in Proving Theorems

Encoding for R

Enc1: $\left\{ \begin{array}{l} \boldsymbol{k}_x(0,\boldsymbol{r},\boldsymbol{h}) \\ \boldsymbol{k}_x(a,\boldsymbol{r},\boldsymbol{h}) \end{array} \right.$

Given IND here

Enc2: $\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h})$

$\overline{\textbf{Encoding}}$ **for** $\overline{R}$

$\overline{\text{Enc1}}$: $\overline{\boldsymbol{k}}_y(\overline{a},\overline{\boldsymbol{s}},\overline{\boldsymbol{h}}) = (\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h}),\ \overline{a}+\overline{b}s_0)$

$\overline{\text{Enc2}}$: $\overline{\boldsymbol{c}}_x(\overline{\boldsymbol{r}},\overline{\boldsymbol{h}}) = (\boldsymbol{k}_x(\overline{bs_0},\boldsymbol{r},\boldsymbol{h}),\ \overline{s_0})$

(all terms over the exponent)

Difficulty in Proving Theorems

Encoding for R

Enc1: $\left\{ \begin{array}{l} \boldsymbol{k}_x(0,\boldsymbol{r},\boldsymbol{h}) \\ \boldsymbol{k}_x(a,\boldsymbol{r},\boldsymbol{h}) \end{array} \right.$

Given IND here

Enc2: $\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h})$

$\overline{\text{Encoding}}$ for $\overline{R}$

$\overline{\text{Enc1}}$: $\left\{ \begin{array}{l} \overline{\boldsymbol{k}}_y(0,\overline{\boldsymbol{s}},\overline{\boldsymbol{h}}) \\ \overline{\boldsymbol{k}}_y(\overline{a},\overline{\boldsymbol{s}},\overline{\boldsymbol{h}}) = (\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h}),\ \overline{a}+\overline{b}s_0) \end{array} \right.$

Goal: to prove IND here.
But it totally differs from $\boldsymbol{k}_x$.

$\overline{\text{Enc2}}$: $\overline{\boldsymbol{c}}_x(\overline{\boldsymbol{r}},\overline{\boldsymbol{h}}) = (\boldsymbol{k}_x(\overline{b}\overline{s_0},\boldsymbol{r},\boldsymbol{h}),\ \overline{s_0})$

(all terms over the exponent)

Proof Idea

Encoding for R	$\overline{\textbf{Encoding for } R}$
Enc1: $\boldsymbol{k}_x(a,\boldsymbol{r},\boldsymbol{h})$	$\overline{\text{Enc1}}$: $\overline{\boldsymbol{k}}_y(\overline{a},\overline{\boldsymbol{s}},\overline{\boldsymbol{h}}) = (\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h}),\ \overline{a}+\overline{b}s_0)$
Enc2: $\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h})$	$\overline{\text{Enc2}}$: $\overline{\boldsymbol{c}}_x(\overline{\boldsymbol{r}},\overline{\boldsymbol{h}}) = (\boldsymbol{k}_x(\overline{b}\overline{s_0},\boldsymbol{r},\boldsymbol{h}),\ \overline{s_0})$

(all terms over the exponent)

Proof Idea

Encoding for R | **$\overline{\text{Encoding for}}$ $\overline{R}$**

Enc1: $\boldsymbol{k}_x(a,\boldsymbol{r},\boldsymbol{h})$

$\overline{\text{Enc1}}$: $\overline{\boldsymbol{k}}_y(\overline{a},\overline{\boldsymbol{s}},\overline{\boldsymbol{h}}) = (\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h}),\ \overline{a}+\overline{b}s_0)$

Enc2: $\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h})$

$\overline{\text{Enc2}}$: $\overline{\boldsymbol{c}}_x(\overline{\boldsymbol{r}},\overline{\boldsymbol{h}}) = (\boldsymbol{k}_x(\overline{b}\overline{s_0},\boldsymbol{r},\boldsymbol{h}),\ \overline{s_0})$

Sim

attacks Enc

Adv

attacks $\overline{\text{Enc}}$

(all terms over the exponent)

Proof Idea

Encoding for R

$\overline{\text{Encoding}}$ for $\overline{R}$

define $\bar{a}$ on unknown a

Enc1: $\boldsymbol{k}_x(a, \boldsymbol{r}, \boldsymbol{h})$

$\overline{\text{Enc1}}$: $\bar{\boldsymbol{k}}_y(\bar{a}, \bar{\boldsymbol{s}}, \bar{\boldsymbol{h}}) = (\boldsymbol{c}_y(\boldsymbol{s}, \boldsymbol{h}),\ \bar{a} + \bar{b} s_0)$

Enc2: $\boldsymbol{c}_y(\boldsymbol{s}, \boldsymbol{h})$

$\overline{\text{Enc2}}$: $\bar{\boldsymbol{c}}_x(\bar{\boldsymbol{r}}, \bar{\boldsymbol{h}}) = (\boldsymbol{k}_x(\bar{b}\bar{s}_0, \boldsymbol{r}, \boldsymbol{h}),\ \bar{s}_0)$

Sim

attacks Enc

Adv

attacks $\overline{\text{Enc}}$

(all terms over the exponent)

Proof Idea

Encoding for R

$\overline{\text{Encoding}}$ for $\overline{R}$

define $\overline{a}$ on unknown a

Enc1: $\boldsymbol{k}_x(a,\boldsymbol{r},\boldsymbol{h})$

$\overline{\text{Enc1}}$: $\overline{\boldsymbol{k}}_y(\overline{a},\overline{\boldsymbol{s}},\overline{\boldsymbol{h}}) = (\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h}),\ \overline{a}+\overline{b}s_0)$

But this becomes unknown, too.

Enc2: $\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h})$

$\overline{\text{Enc2}}$: $\overline{\boldsymbol{c}}_x(\overline{\boldsymbol{r}},\overline{\boldsymbol{h}}) = (\boldsymbol{k}_x(\overline{b}\overline{s_0},\boldsymbol{r},\boldsymbol{h}),\ \overline{s_0})$

Sim

attacks Enc

Adv

attacks $\overline{\text{Enc}}$

(all terms over the exponent)

Proof Idea: Cancellation Trick

Encoding for R

$\overline{\textbf{Encoding}}$ **for** $\overline{R}$

define $\bar{a}$ on unknown a

Enc1: $\mathbf{k}_x(a,\mathbf{r},\mathbf{h})$

$\overline{\text{Enc1}}$: $\overline{\mathbf{k}}_y(\bar{a},\bar{\mathbf{s}},\bar{\mathbf{h}}) = (\mathbf{c}_y(\mathbf{s},\mathbf{h}),\ \bar{a}+\bar{b}s_0)$

Enc2: $\mathbf{c}_y(\mathbf{s},\mathbf{h})$

$\overline{\text{Enc2}}$: $\overline{\mathbf{c}}_x(\bar{\mathbf{r}},\bar{\mathbf{h}}) = (\mathbf{k}_x(\bar{b}\bar{s}_0,\mathbf{r},\mathbf{h}),\ \bar{s}_0\ )$

Sim

attacks Enc

Adv

attacks $\overline{\text{Enc}}$

(all terms over the exponent)

Proof Idea: Cancellation Trick

Encoding for R	$\overline{\textbf{Encoding for}\ R}$

define $\bar{a}$ on unknown a

Enc1: $\boldsymbol{k}_x(a,\boldsymbol{r},\boldsymbol{h})$ $\quad$ $\overline{\text{Enc1}}$: $\bar{\boldsymbol{k}}_y(\bar{a},\bar{\boldsymbol{s}},\bar{\boldsymbol{h}}) = (\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h}),\ \bar{a}+\bar{b}s_0)$

define $\bar{b}$ on unknown a

Enc2: $\boldsymbol{c}_y(\boldsymbol{s},\boldsymbol{h})$ $\quad$ $\overline{\text{Enc2}}$: $\bar{\boldsymbol{c}}_x(\bar{\boldsymbol{r}},\bar{\boldsymbol{h}}) = (\boldsymbol{k}_x(\bar{b}s_0,\boldsymbol{r},\boldsymbol{h}),\ s_0)$

Sim

attacks Enc

Adv

attacks $\overline{\text{Enc}}$

(all terms over the exponent)

Proof Idea: Cancellation Trick

Encoding for R | $\overline{\textbf{Encoding for } R}$

define $\bar{a}$ on unknown a

cancellation of two unknowns!

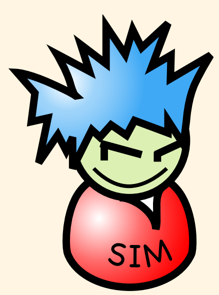
Enc1: $\mathbf{k}_x(a,\mathbf{r},\mathbf{h})$

$\overline{\text{Enc1}}$: $\bar{\mathbf{k}}_y(\bar{a},\bar{\mathbf{s}},\bar{\mathbf{h}}) = (\mathbf{c}_y(\mathbf{s},\mathbf{h}),\ \bar{a}+\bar{b}s_0)$

define $\bar{b}$ on unknown a

Enc2: $\mathbf{c}_y(\mathbf{s},\mathbf{h})$

$\overline{\text{Enc2}}$: $\bar{\mathbf{c}}_x(\bar{\mathbf{r}},\bar{\mathbf{h}}) = (\mathbf{k}_x(\bar{b}s_0,\mathbf{r},\mathbf{h}),\ \bar{s}_0)$



Sim

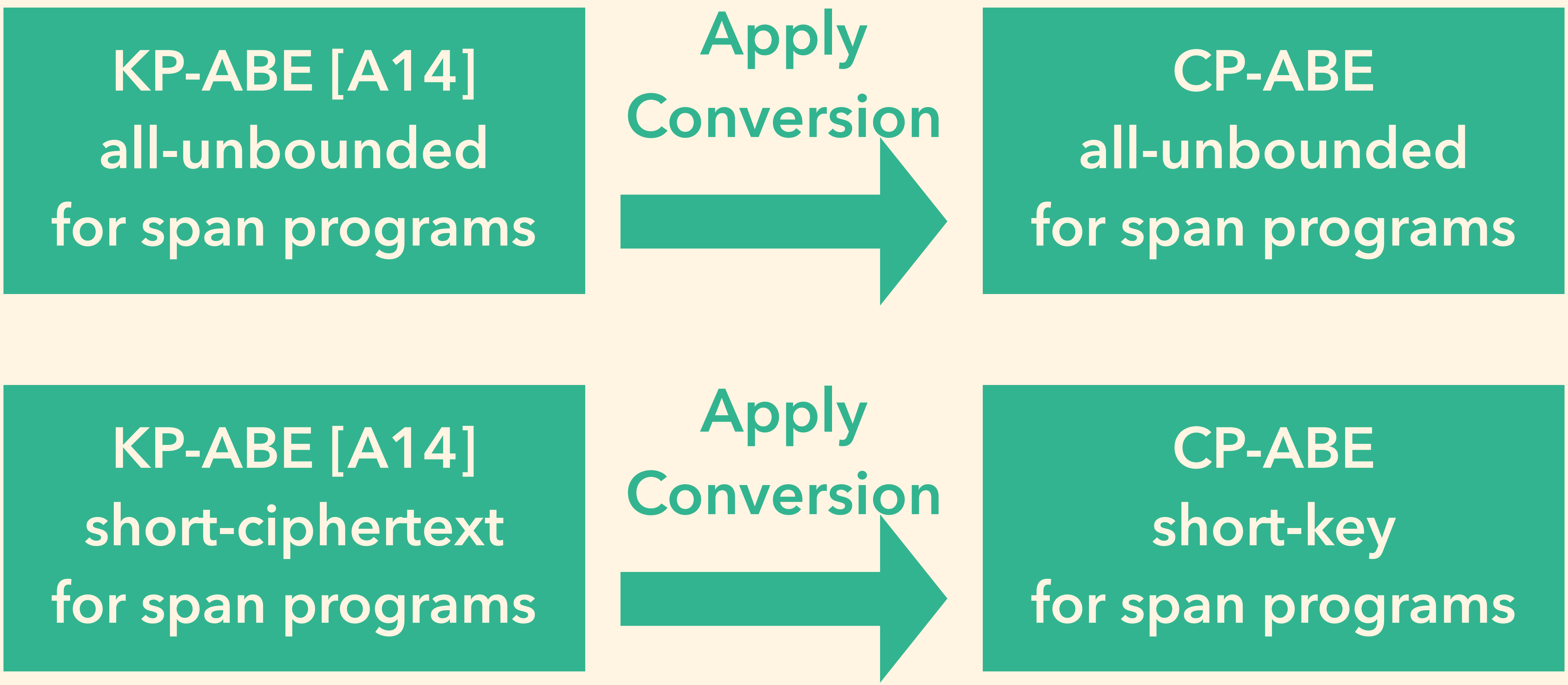
attacks Enc

Adv

attacks $\overline{\text{Enc}}$

(all terms over the exponent)

New Instantiations



Doubly selective secure under
some Extended DH Exponent assumptions [A14].

5 Concluding Remarks

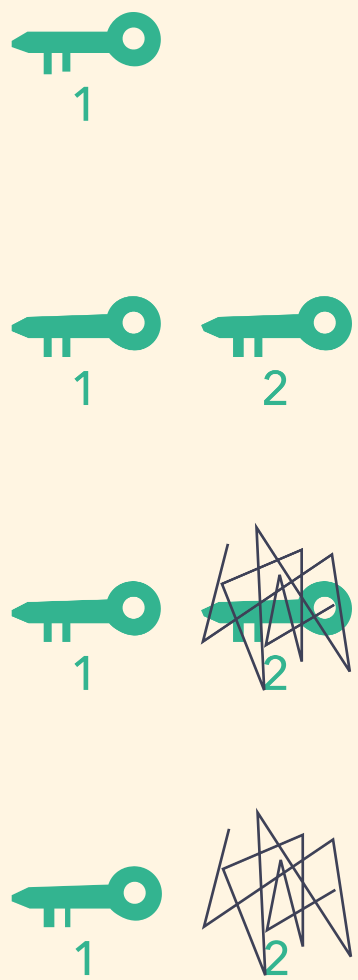
More Results

- Dual-Policy ABE
  - Conjunctively combine ABE and its dual [AI09].
  - We also provide a conversion from ABE to DP-ABE.
- More refinement:
  - New specific CP-ABE with tighter reduction.
- Full version at http://eprint.iacr.org/2015/157.

Thank you

Intuition Behind Pair Encoding Security

Switch Keys from Normal to Semi-functional [A14]



normal
K
$g_1^{\boldsymbol{k}(a,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(0,\boldsymbol{0},\boldsymbol{0})}$

Subgroup Decision

semi-1
K1
$g_1^{\boldsymbol{k}(a,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(0,\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})}$

Security of encoding

semi-2
K2
$g_1^{\boldsymbol{k}(a,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{a},\hat{\boldsymbol{r}},\hat{\boldsymbol{h}})}$

Subgroup Decision

semi-3
K3
$g_1^{\boldsymbol{k}(a,\boldsymbol{r},\boldsymbol{h})} \cdot g_2^{\boldsymbol{k}(\hat{a},\boldsymbol{0},\boldsymbol{0})}$

- Only for self-containment, will not use here.

RSAConference2015

San Francisco | April 20-24 | Moscone Center

Revocable Hierarchical Identity-Based Encryption: History-Free Update, Security Against Insiders, and Short Ciphertexts

Jae Hong Seo[1] and Keita Emura[2]

1. Myongji University, Korea
2. NICT, Japan

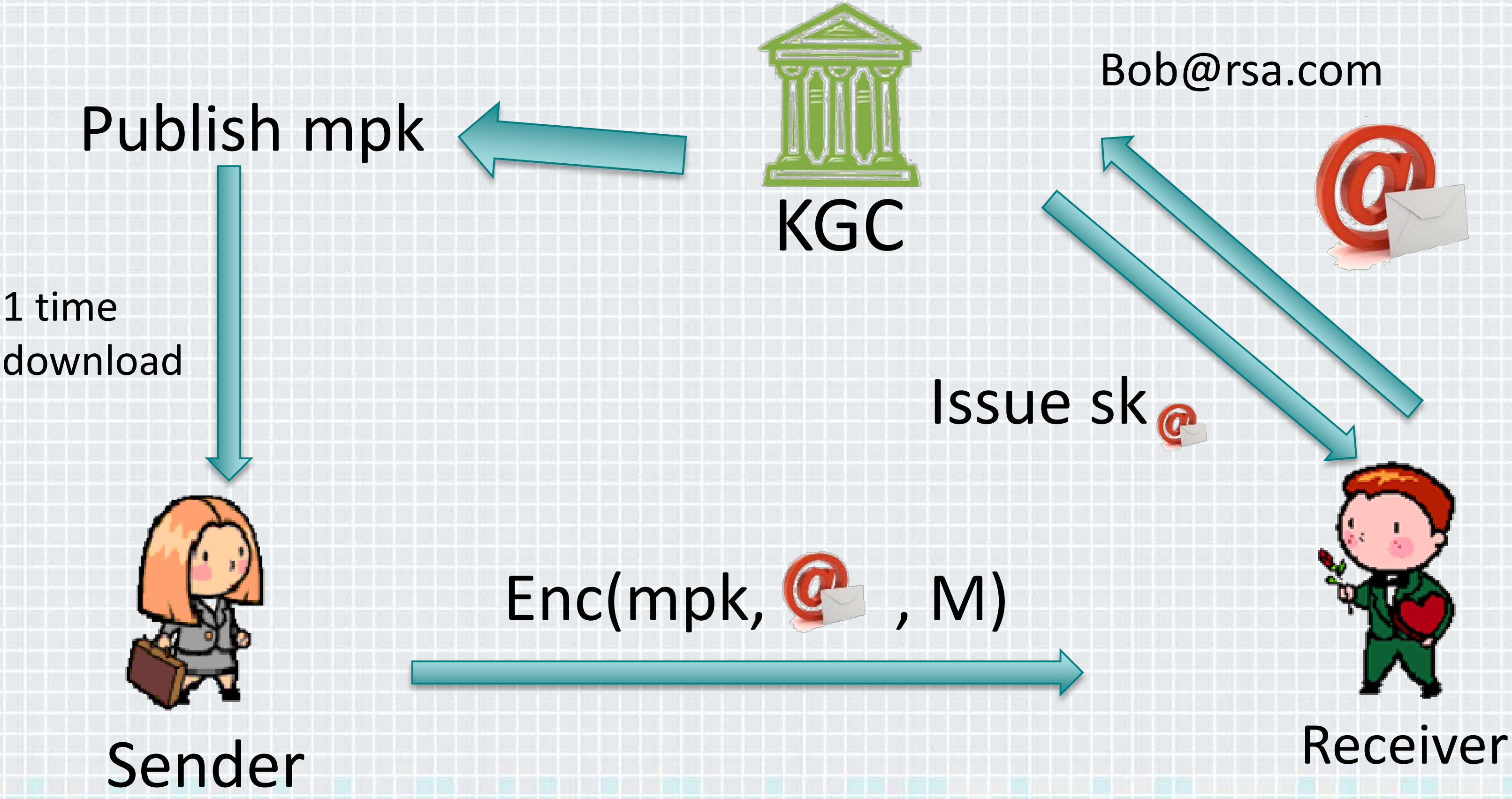
#RSAC

Contents

- Identity-based encryption with revocation (RIBE)
  - Trivial Way (by Boneh and Franklin 2001)
  - Scalable construction (by Boldyreva, Goyal, and Kumar, 2008)
- Revocable Hierarchical IBE (RHIBE): CT-RSA 2013, Seo and Emura
  - History-preserving updates approach
  - Security against outsider
  - Long-size ciphertext (ciphertext size depends on the level of hierarchy)
- Our RHIBE Constructions
  - History-free updates approach
  - Security against insider
  - Constant-size ciphertext (in terms of the hierarchy level)

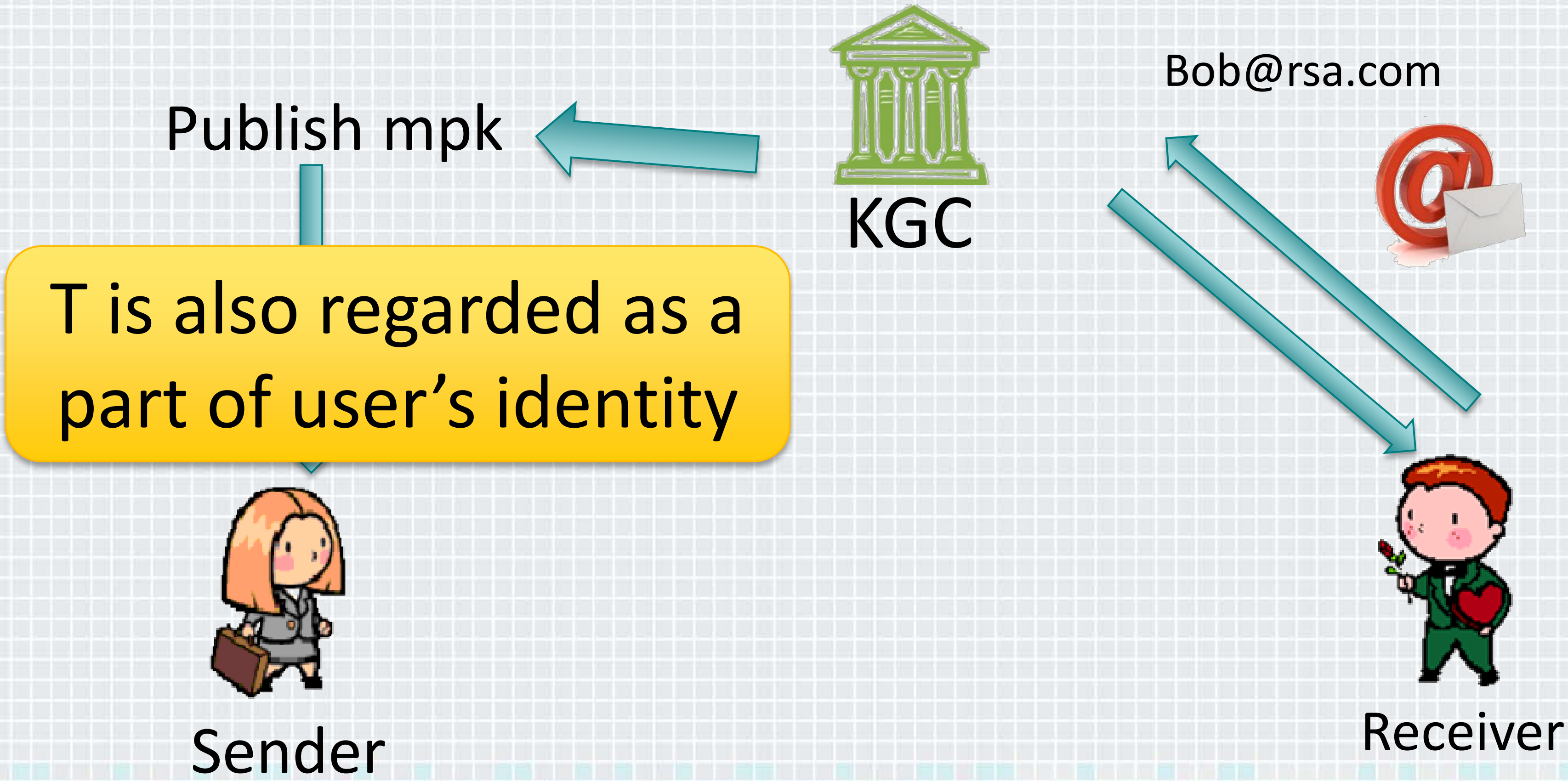
Identity-Based Encryption and Revocation

Identity-Based Encryption (IBE)

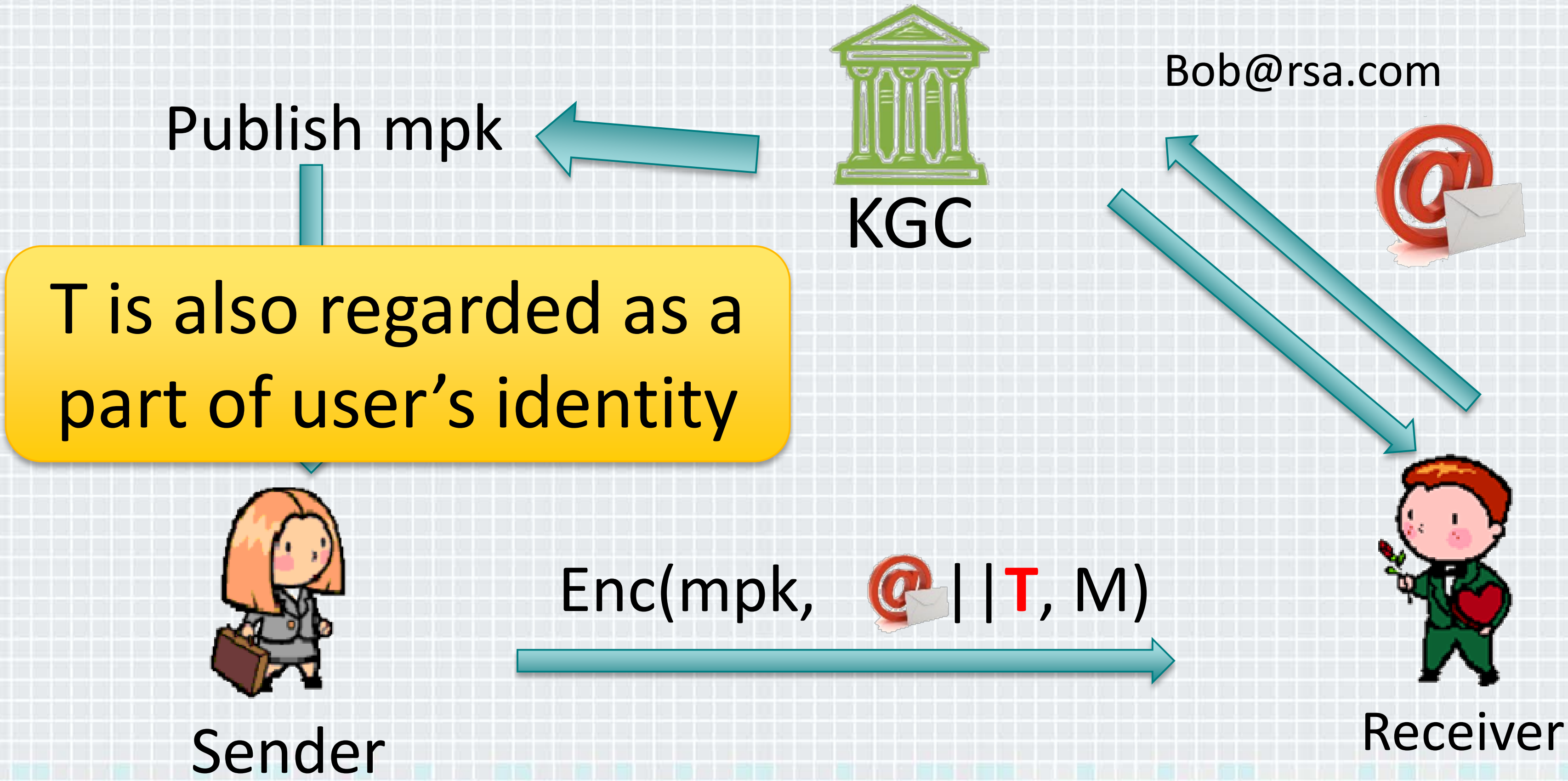


Identity-Based Encryption (IBE)

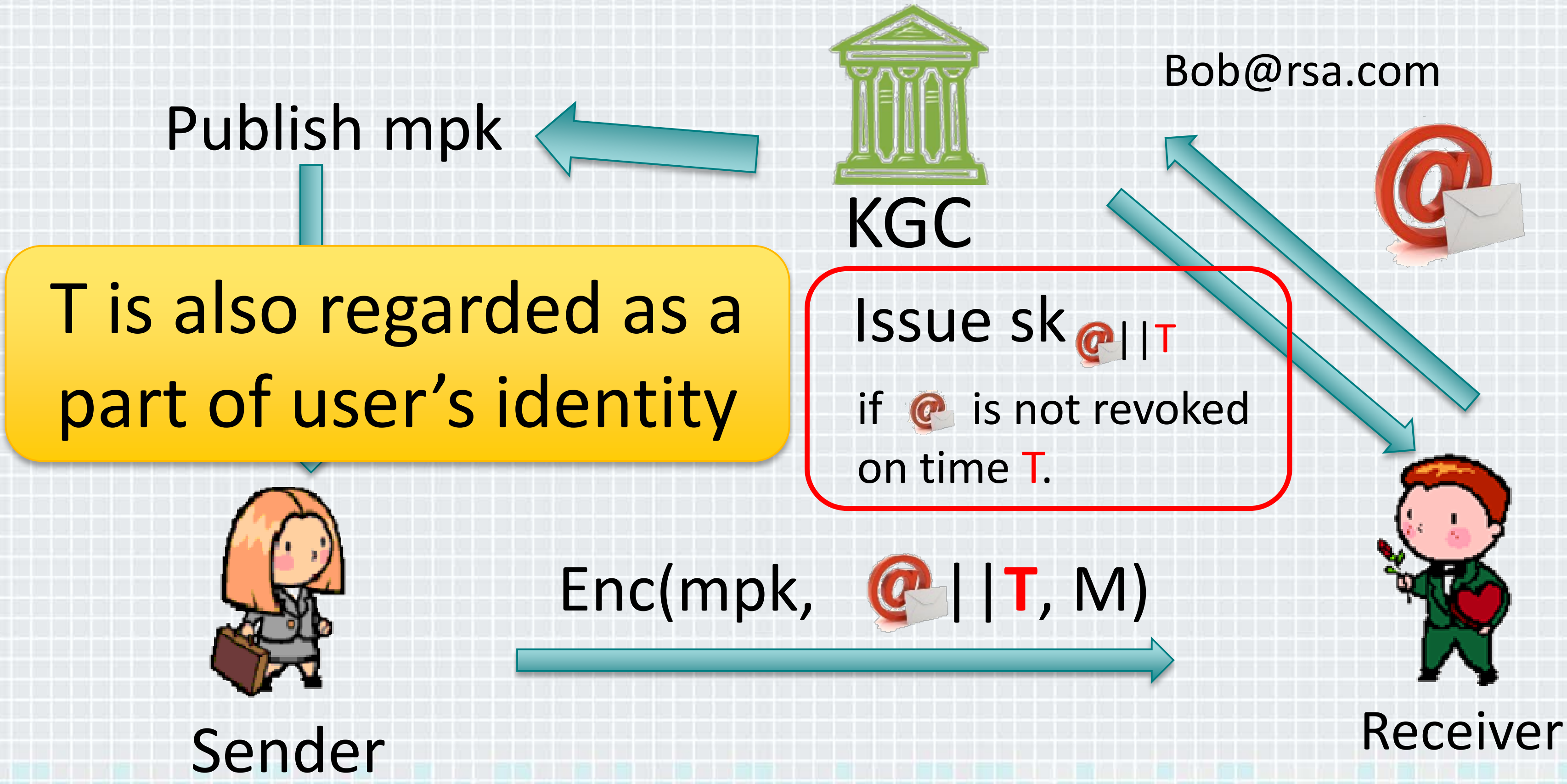
Revocation Capability in IBE: Boneh-Franklin



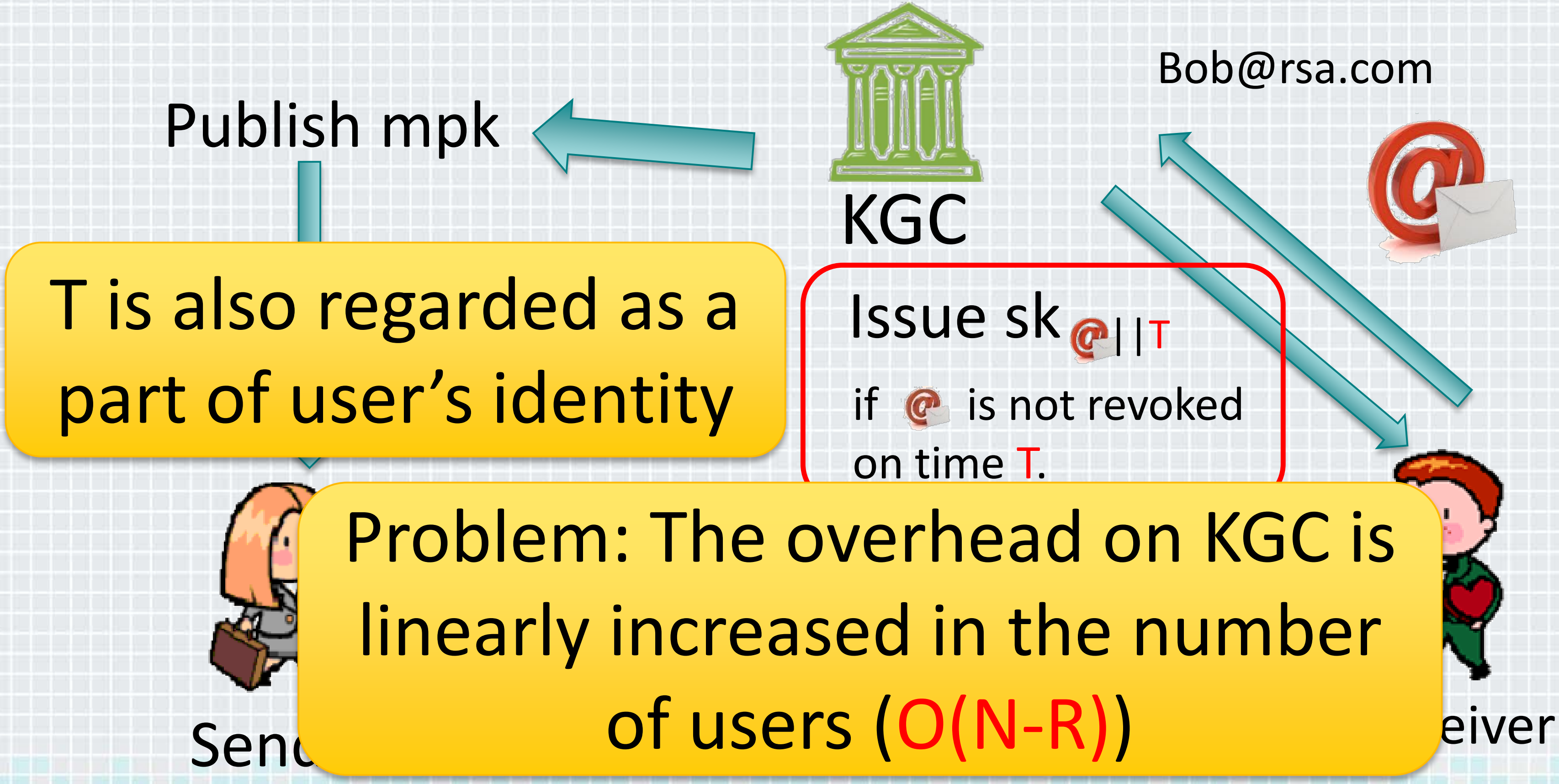
Revocation Capability in IBE: Boneh-Franklin



Revocation Capability in IBE: Boneh-Franklin



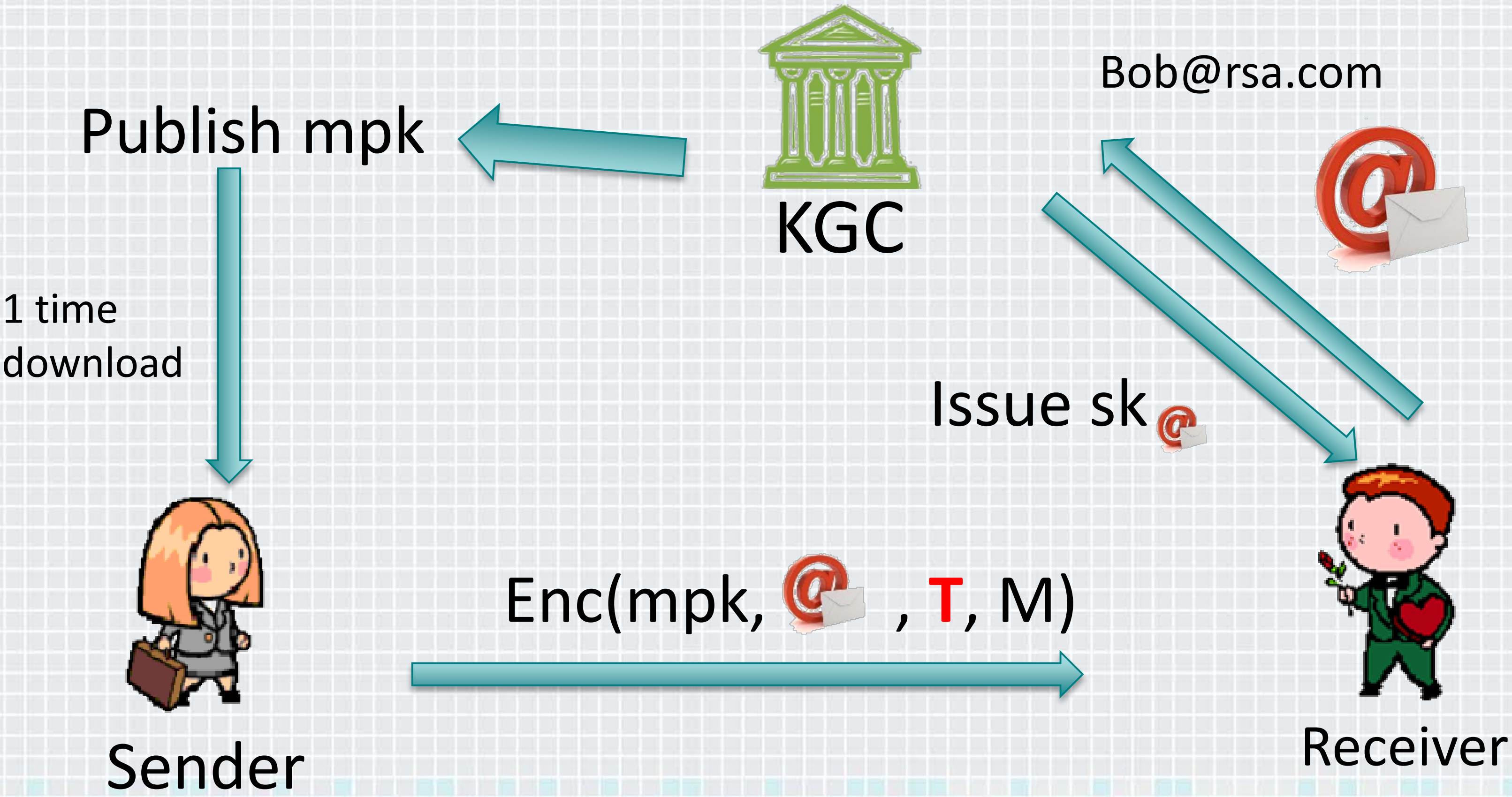
Revocation Capability in IBE: Boneh-Franklin



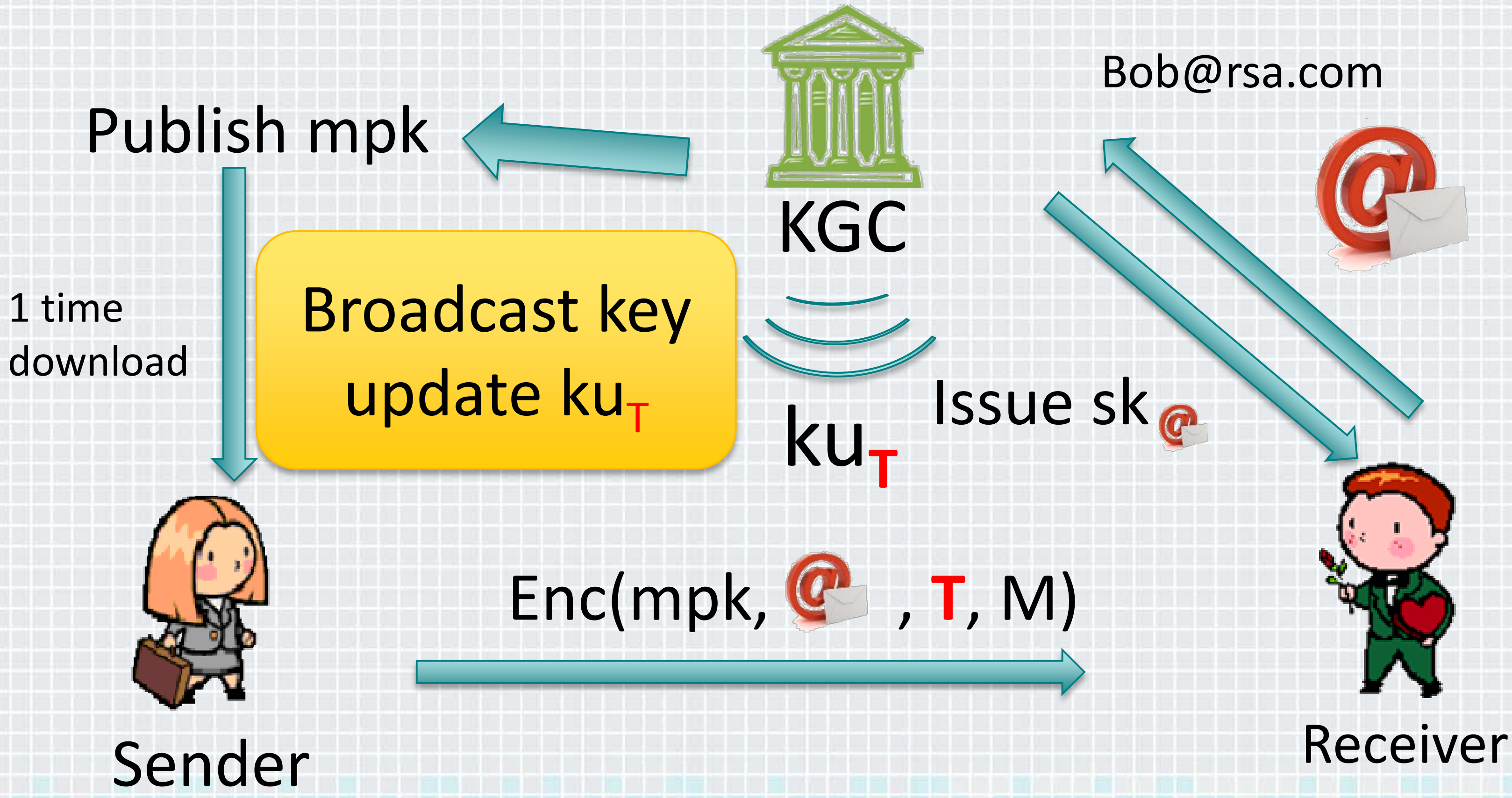
T is also regarded as a part of user's identity

Problem: The overhead on KGC is linearly increased in the number of users (O(N-R))

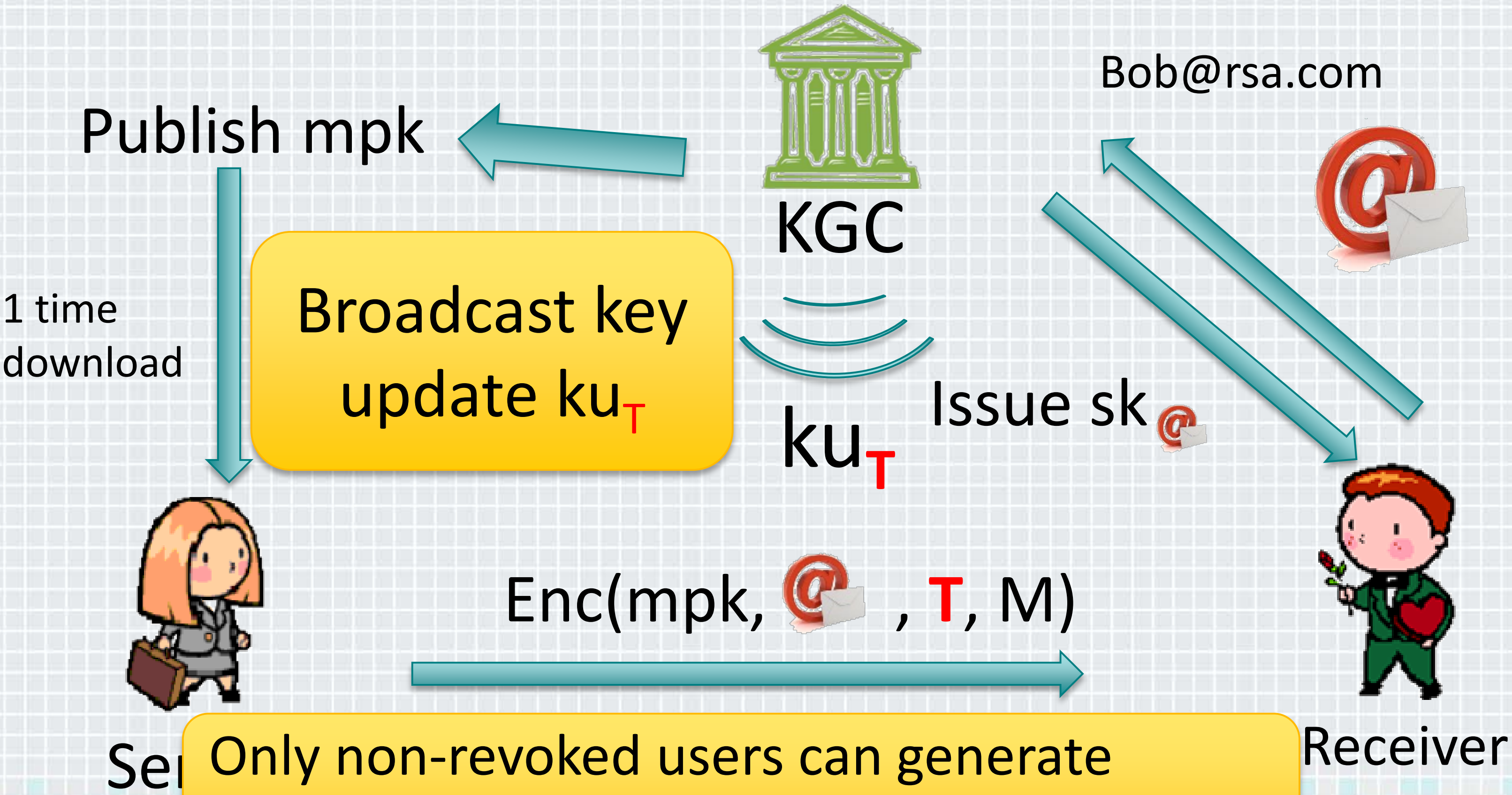
Revocation Capability in IBE: Boldyreva et al.



Revocation Capability in IBE: Boldyreva et al.

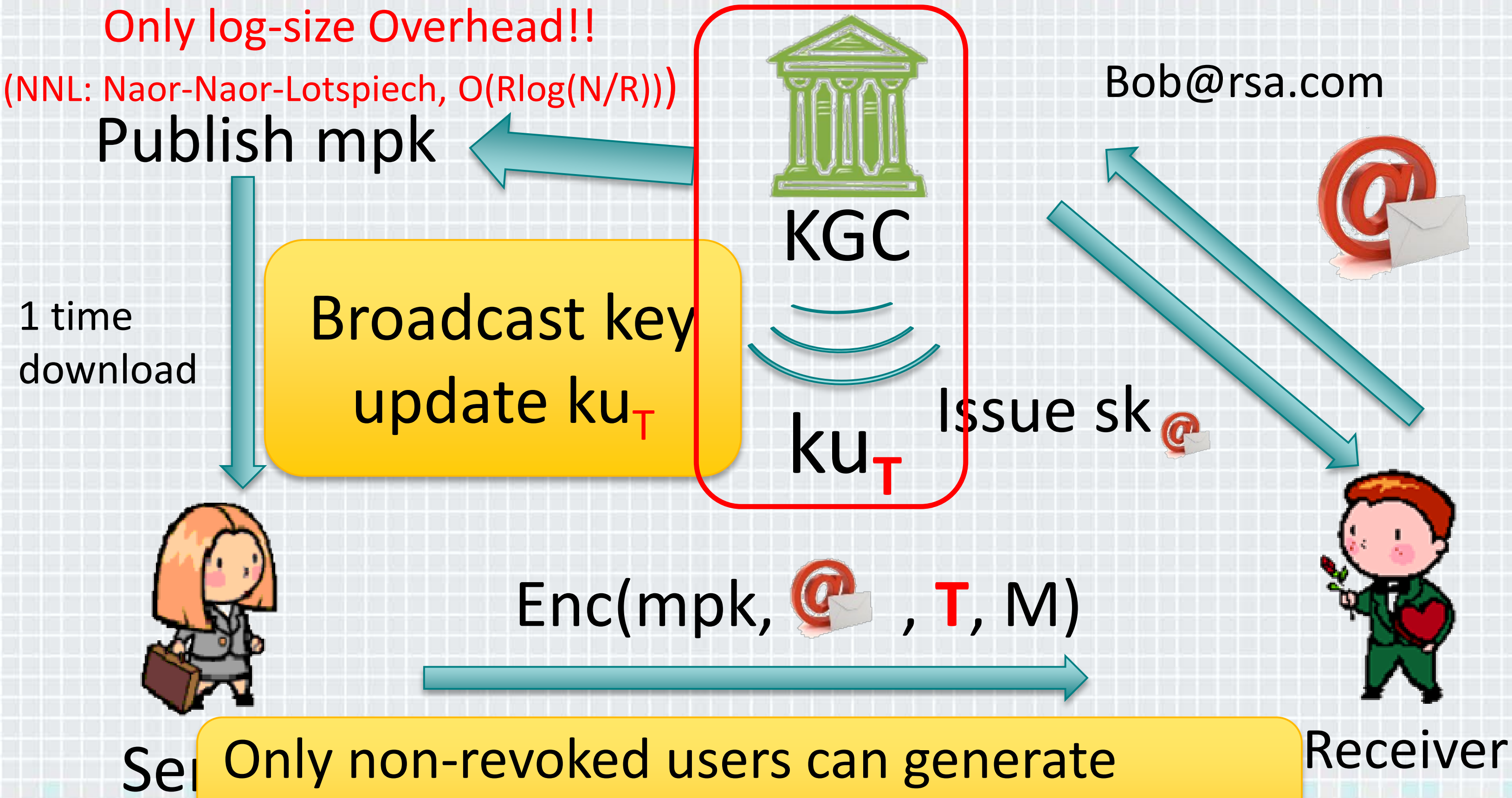


Revocation Capability in IBE: Boldyreva et al.



Only non-revoked users can generate a decryption key dk ,T from ku_T and sk

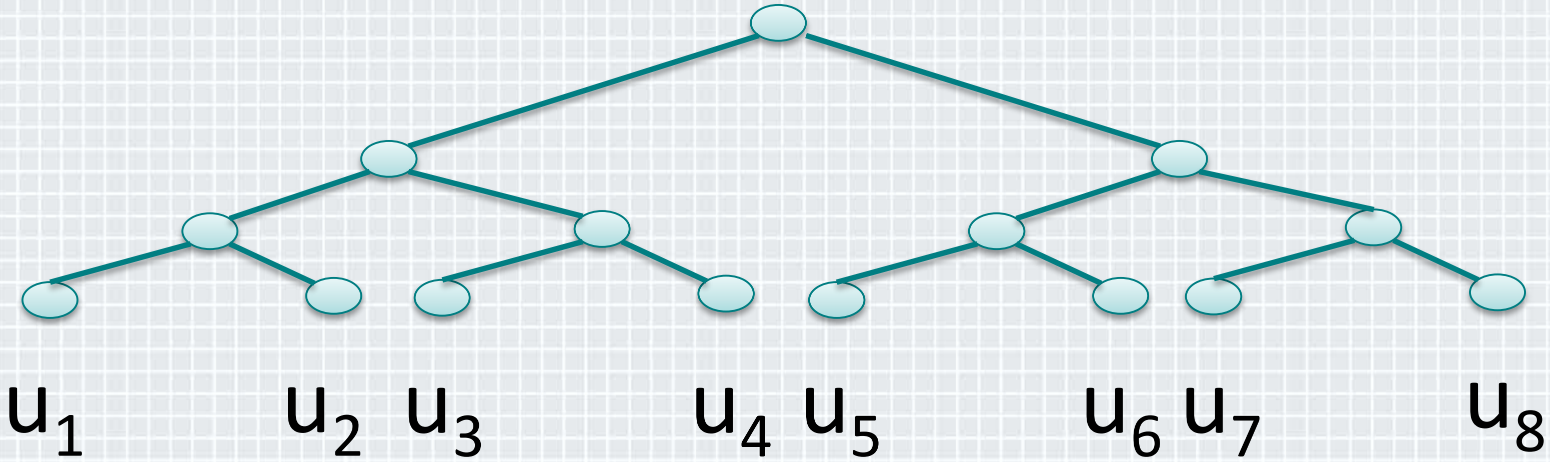
Revocation Capability in IBE: Boldyreva et al.



Only non-revoked users can generate a decryption key dk ,T from ku_T and sk

Broadcast Encryption (BE) Technique
Complete Subtree (CS)

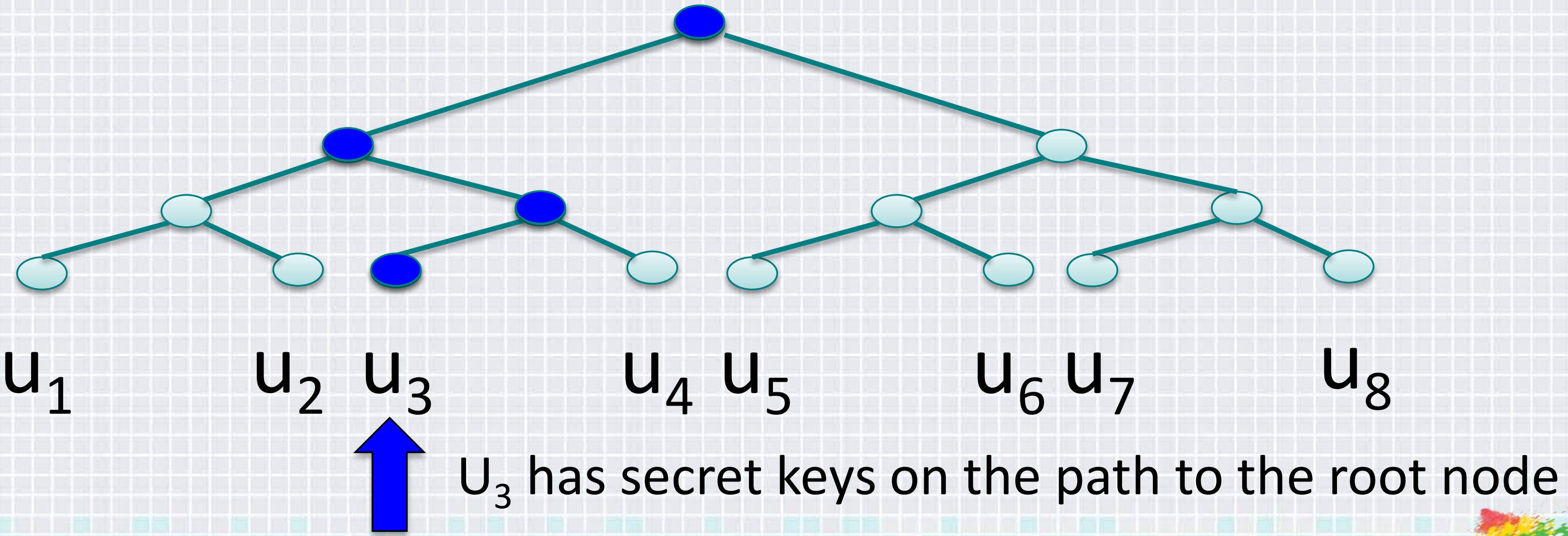
- Each user is assigned to a node



We consider a binary tree kept by KGC

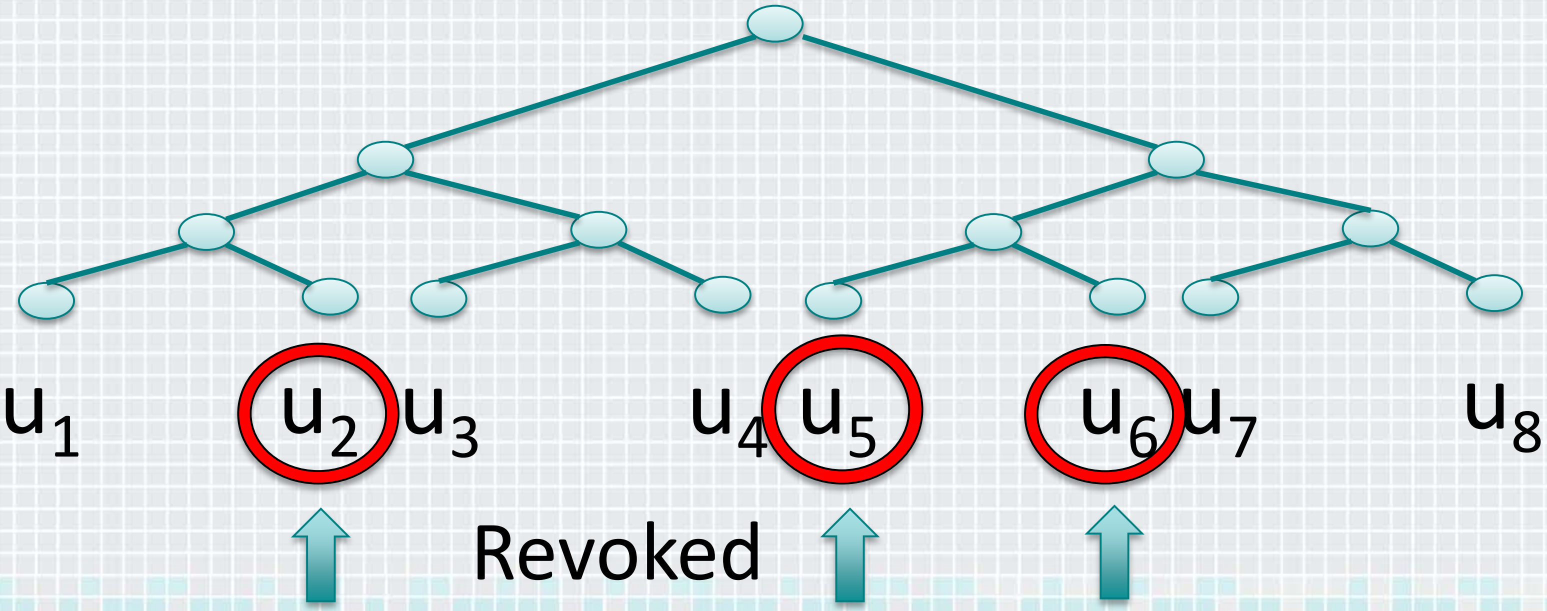
Broadcast Encryption (BE) Technique Complete Subtree (CS)

- Each user is issued secret keys on the path to the root node by KGC (sk)



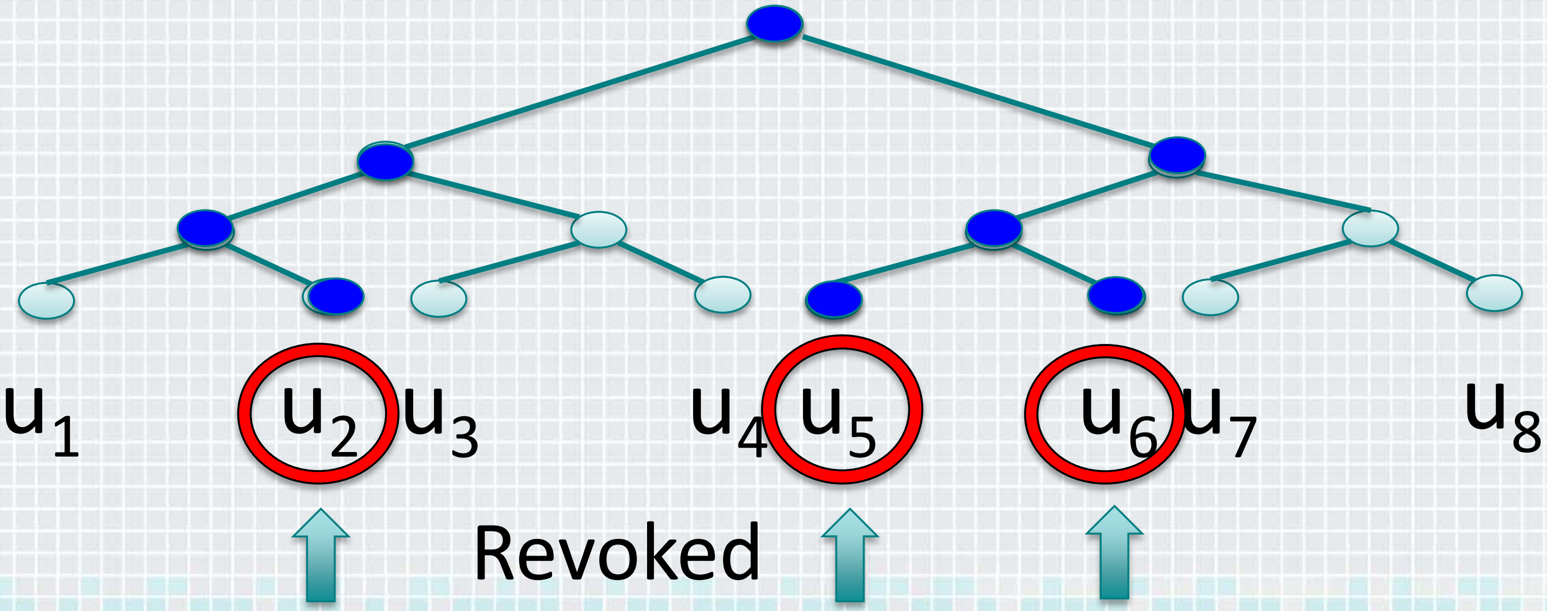
Broadcast Encryption (BE) Technique
Complete Subtree (CS)

- ku_T is computed for nodes which do not have intersection against paths (to the root node) of revoked users



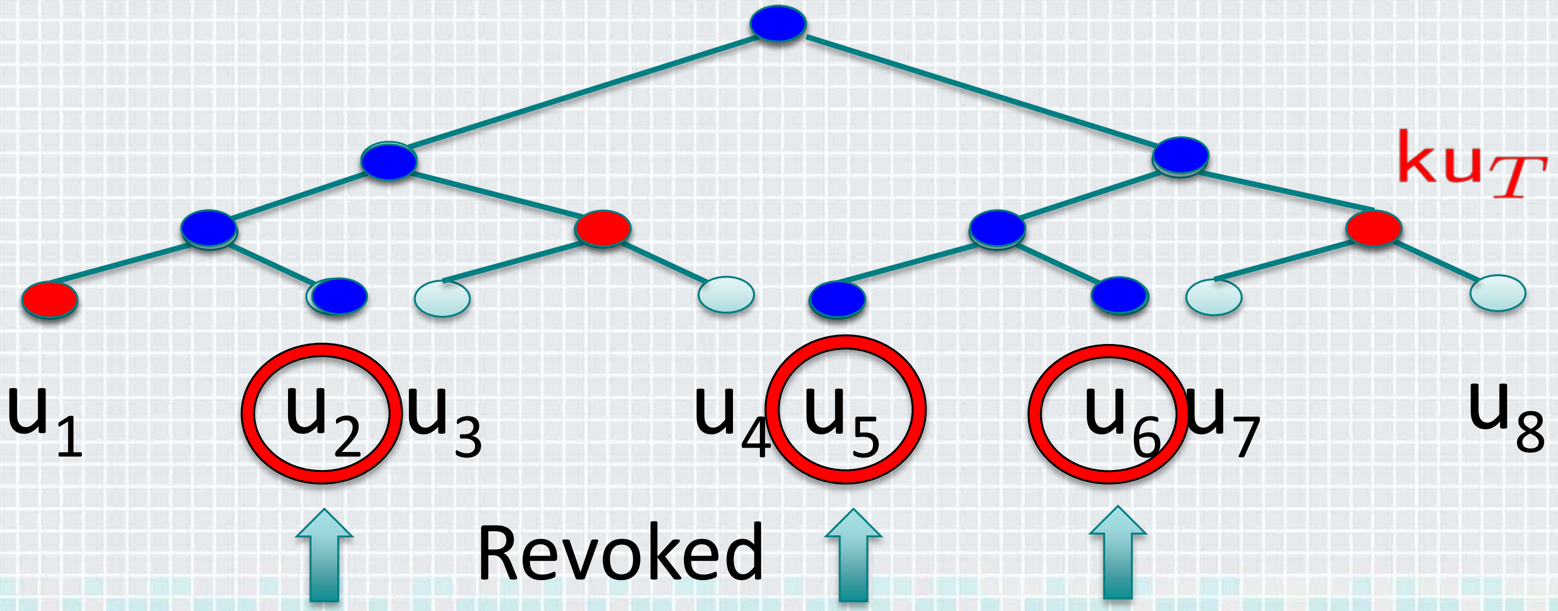
Broadcast Encryption (BE) Technique Complete Subtree (CS)

- ku_T is computed for nodes which do not have intersection against paths (to the root node) of revoked users



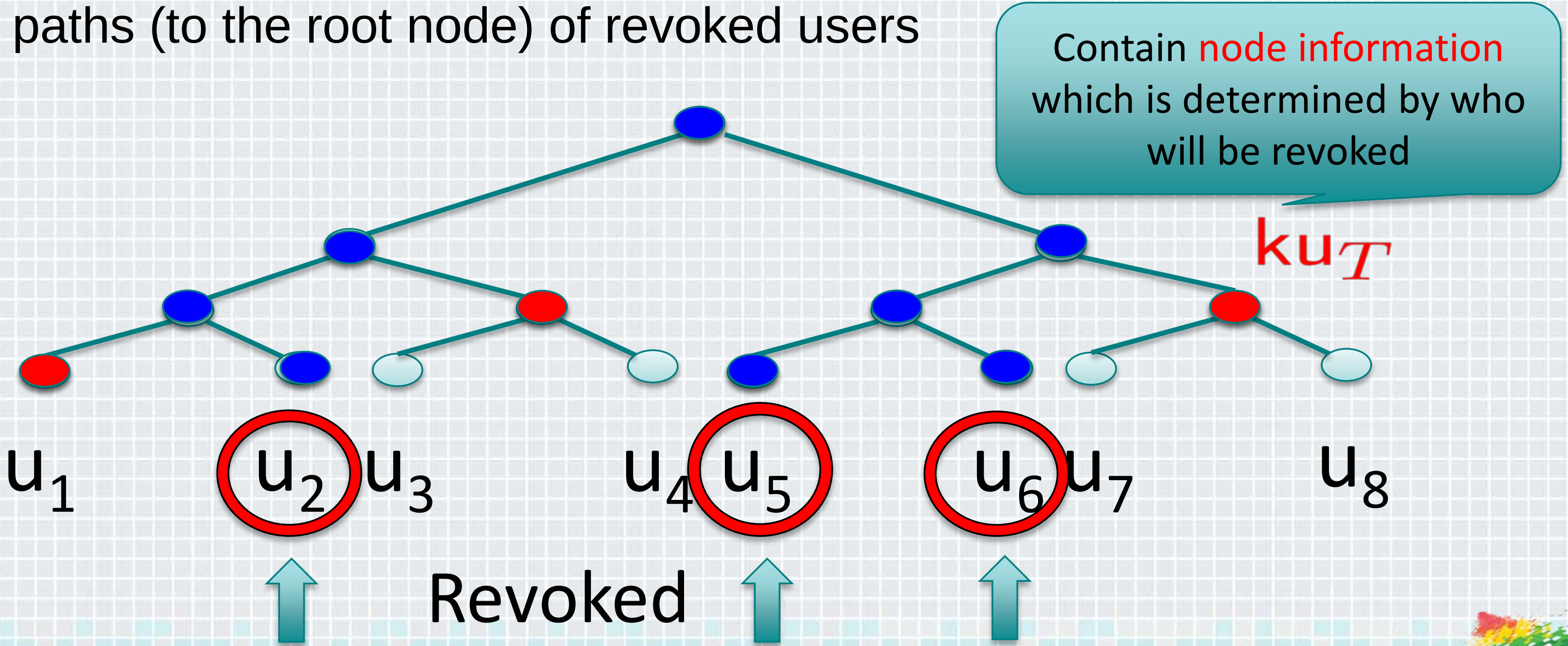
Broadcast Encryption (BE) Technique Complete Subtree (CS)

- ku_T is computed for nodes which do not have intersection against paths (to the root node) of revoked users



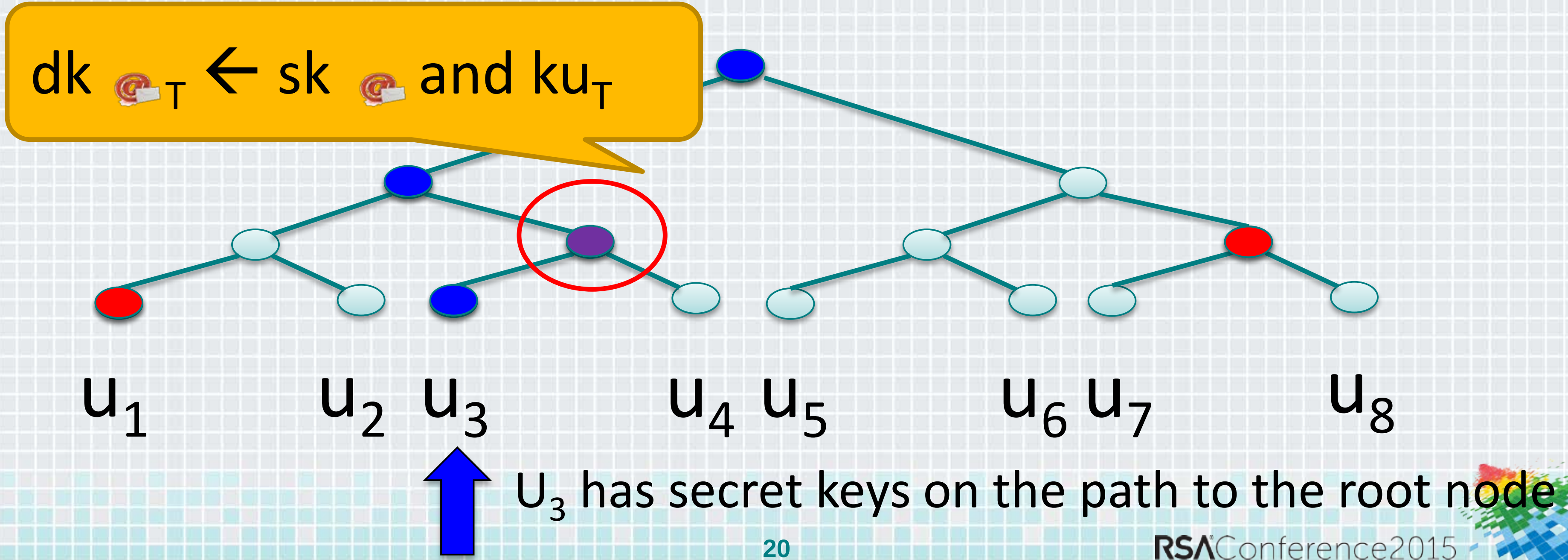
Broadcast Encryption (BE) Technique
Complete Subtree (CS)

- ku_T is computed for nodes which do not have intersection against paths (to the root node) of revoked users



Broadcast Encryption (BE) Technique
Complete Subtree (CS)

- From log N size public information ku_T, only non-revoked users can extract useful information.



Scalable Revocable IBE

- First construction
  - A. Boldyreva, V. Goyal, and V. Kumar. Identity-based encryption with efficient revocation. In ACM CCS 2008
- First adaptive secure scheme
  - B. Libert and D. Vergnaud. Adaptive-ID secure revocable identity-based encryption. In CT-RSA 2009.
- Considering decryption key exposure resistance
  - J. H. Seo and K. Emura. Revocable identity-based encryption revisited: Security model and construction. In PKC 2013.
  - An adversary is allowed to obtain

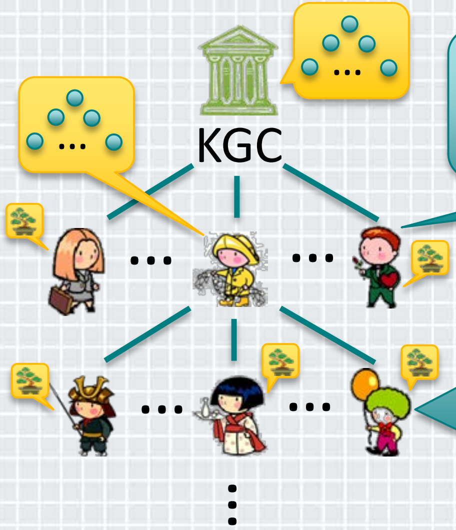
$$dk_{ID,T} \text{ if } (ID, T) \neq (ID^*, T^*)$$

- SD-based construction
  - K. Lee, D. H. Lee, and J. H. Park. Efficient revocable identity-based encryption via subset difference methods, eprint.iacr.org/2014/132, 2014.

Revocable Hierarchical IBE (RHIBE)

Revocable Hierarchical IBE (RHIBE)

- A low-level user can stay in the system only if her parent also stays in the current time period.



First-level users have O(log N)-size secret key (as in RIBE)

Second-level users have log N subkeys for "each" parent's subkey (in total $(\log N)^2$-size subkeys)

$skID_1 = \{ subkey_1, \ldots, subkey_{logN} \}$

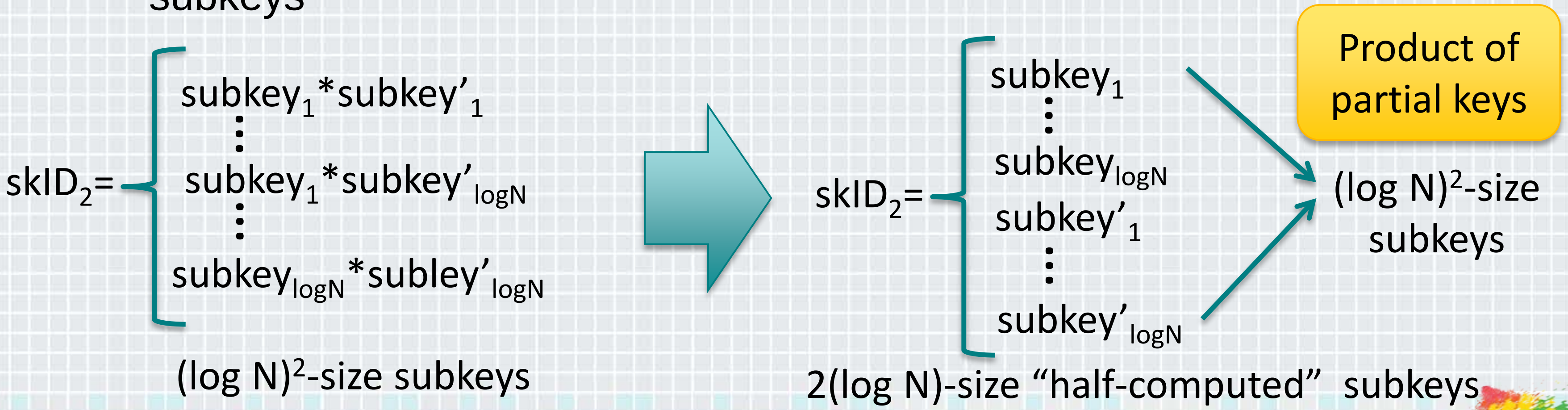
One of subkeys will be used for computing kuT.

$skID_2 = \{ subkey_1 * subkey'_1, \ldots, subkey_1 * subkey'_{logN}, \ldots, subkey_{logN} * subley'_{logN} \}$

- Trivial combination of RIBE and HIBE will result in an impractical scheme with an exponential number of secret keys

Revocable Hierarchical IBE (RHIBE)

- The first RHIBE scheme with polynomial size secret keys (Seo-Emura, CT-RSA 2013)
  - Asymmetric trade between secret key size and generating time for secret key
    - A parent gives “half-computed” subkeys, and children generate suitable subkeys



Revocable Hierarchical IBE (RHIBE)

- The first RHIBE scheme with polynomial size secret keys (Seo-Emura, CT-RSA 2013)

size and generating time for

, and children generate suitable

History-preserving key updates

- For the calculation, a child needs to know which partial key of the ancestor was used in each time period.
- Such information is also announced in the key updates

$$ku_{ID_{|\ell-1},T} := \{\{Lv_i\}_{i \in [1,\ell-1]}, \vec{f}_{ID_{|\ell-1},\theta}\}$$

$(\log N)^2$-size subkeys

=
subkey$_1$
⋮
subkey$_{\log N}$
subkey'$_1$
⋮
subkey'$_{\log N}$

$(\log N)^2$-size subkeys

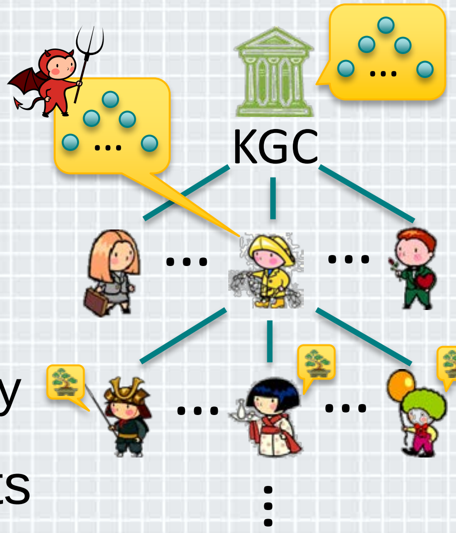
Product of partial keys

2(log N)-size “half-computed” subkeys

#RSAC

Our Contribution

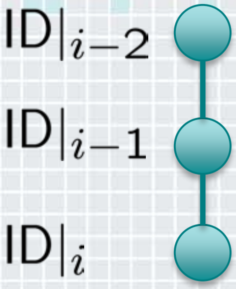
- History-Free Update
  - Low-level users do not need to know what ancestors did during key updates.
- Security Against Insiders
  - An adversary is allowed to obtain state information
- Short Ciphertexts
  - Constant-size ciphertext in terms of the level of hierarchy
- Two constructions: Shorter secret keys and ciphertexts
  - Complete Subtree (CS)
  - Subset Difference (SD)



Main Idea for History-Free Update

- R(H)IBE:
  - KGC (or a parent user) issues a long-term secret key sk_{ID} using msk (or $sk_{parent\text{-}ID}$).
  - KGC (or a parent user) broadcasts key update information ku_T which is computed by msk (or $sk_{parent\text{-}ID}$).
  - A (child) user can generate the decryption key $dk_{ID,T}$ from sk_{ID} and ku_T if he/she is not revoked at time T.
- Two situations are equivalent:
  - A user ID is not revoked at time T
  - The user can generate the decryption key $dk_{ID,T}$
- Re-define the key update algorithm

Main Idea for History-Free Update



- Previous syntax

$$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, N, \ell)$$

$$\mathsf{sk}_{\mathsf{ID}|_i} \leftarrow \mathsf{SKGen}(\mathsf{sk}_{\mathsf{ID}|_{i-1}}, \mathsf{st}_{\mathsf{ID}|_{i-1}}, \mathsf{ID}|_i)$$

$$\mathsf{ku}_{\mathsf{ID}|_{i-1},T} \leftarrow \mathsf{KeyUp}(\mathsf{sk}_{\mathsf{ID}|_{i-1}}, \mathsf{ku}_{\mathsf{ID}|_{i-2},T}, \mathsf{st}_{\mathsf{ID}|_{i-1}}, RL_{\mathsf{ID}|_{i-1}})$$

$$\mathsf{dk}_{\mathsf{ID}|_i,T} \leftarrow \mathsf{DKGen}(\mathsf{sk}_{\mathsf{ID}|_i}, \mathsf{ku}_{\mathsf{ID}|_{i-1},T})$$

- Our modification

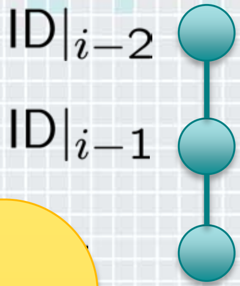
$$\mathsf{sk}_{\mathsf{ID}|_i} \leftarrow \mathsf{SKGen}(\mathsf{st}_{\mathsf{ID}|_{i-1}}, \mathsf{ID}|_i)$$

$$\mathsf{ku}_{\mathsf{ID}|_{i-1},T} \leftarrow \mathsf{KeyUp}(\mathsf{dk}_{\mathsf{ID}|_{i-1},T}, \mathsf{st}_{\mathsf{ID}|_{i-1}}, RL_{\mathsf{ID}|_{i-1}})$$

No parent secret key is required
(for history-free approach)
State information takes a role of the
delegation key

dk is used instead of sk and ku

Main Idea for History-Free Update



- Previous syntax

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, N, \ell)$

$\mathsf{sk}_{\mathsf{ID}|_i} \leftarrow \mathsf{S}$

$\mathsf{ku}_{\mathsf{ID}|_{i-1},T}$

$\mathsf{dk}_{\mathsf{ID}|_i,T} \leftarrow$

- Our mod

$\mathsf{sk}_{\mathsf{ID}|_i} \leftarrow$

$\mathsf{ku}_{\mathsf{ID}|_{i-1},T}$

The secret key is used only for generating the decryption key dk.

Low-level users do not need to know what ancestors did during key updates.

ired
n)
of the

dk is used instead of sk and ku

Proposed RHIBE Scheme (CS)

- Based on the BBG HIBE scheme
  - BBG HIBE (ID) + Boneh-Boyen IBE (Time)
  - Give a reduction to the BBG HIBE scheme.
    - [BBG05] Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. EUROCRYPT 2005.

$$\mathsf{mpk} = \{N, g, h, u_1, \ldots, u_\ell, g_1, g_2, \underline{u', h'}\}$$

$$\mathsf{msk} = \{g_2^\alpha\}$$

(Boneh-Boyen hash)

$$\mathsf{sk}_{\mathsf{ID}|_k} = \left\{P_\theta (u_1^{I_1} \cdots u_k^{I_k} h)^{r_\theta},\ g^{r_\theta},\ u_{k+1}^{r_\theta}, \ldots, u_\ell^{r_\theta}\right\}_{\theta \in \mathsf{Path}(\mathsf{ID}|_k)}$$

$$\mathsf{ku}_{\mathsf{ID}|_{k-1},T} = \left\{P_\theta^{-1} \cdot g_2^\alpha (u_1^{I_1} \cdots u_k^{I_k} h)^{r_\theta} (u'^T h')^{t_\theta},\ g^{r_\theta},\ g^{t_\theta},\ u_k^{r_\theta}, \ldots, u_\ell^{r_\theta}\right\}_{\theta \in \mathsf{KUNode}(\mathsf{BT}_{\mathsf{ID}|_{k-1}}, RL_{\mathsf{ID}|_{k-1}}, T)}$$

$$\mathsf{dk}_{\mathsf{ID}|_k,T} = (g_2^\alpha (u_1^{I_1} \cdots u_k^{I_k} h)^r (u'^T h')^t, g^r, g^t, u_{k+1}^r, \ldots, u_\ell^r)$$

$$\mathsf{CT} = (\underline{M \cdot e(g_1, g_2)^s, g^s, (u_1^{I_1} \cdots u_k^{I_k} h)^s}, \underline{(u'^T h')^s})$$

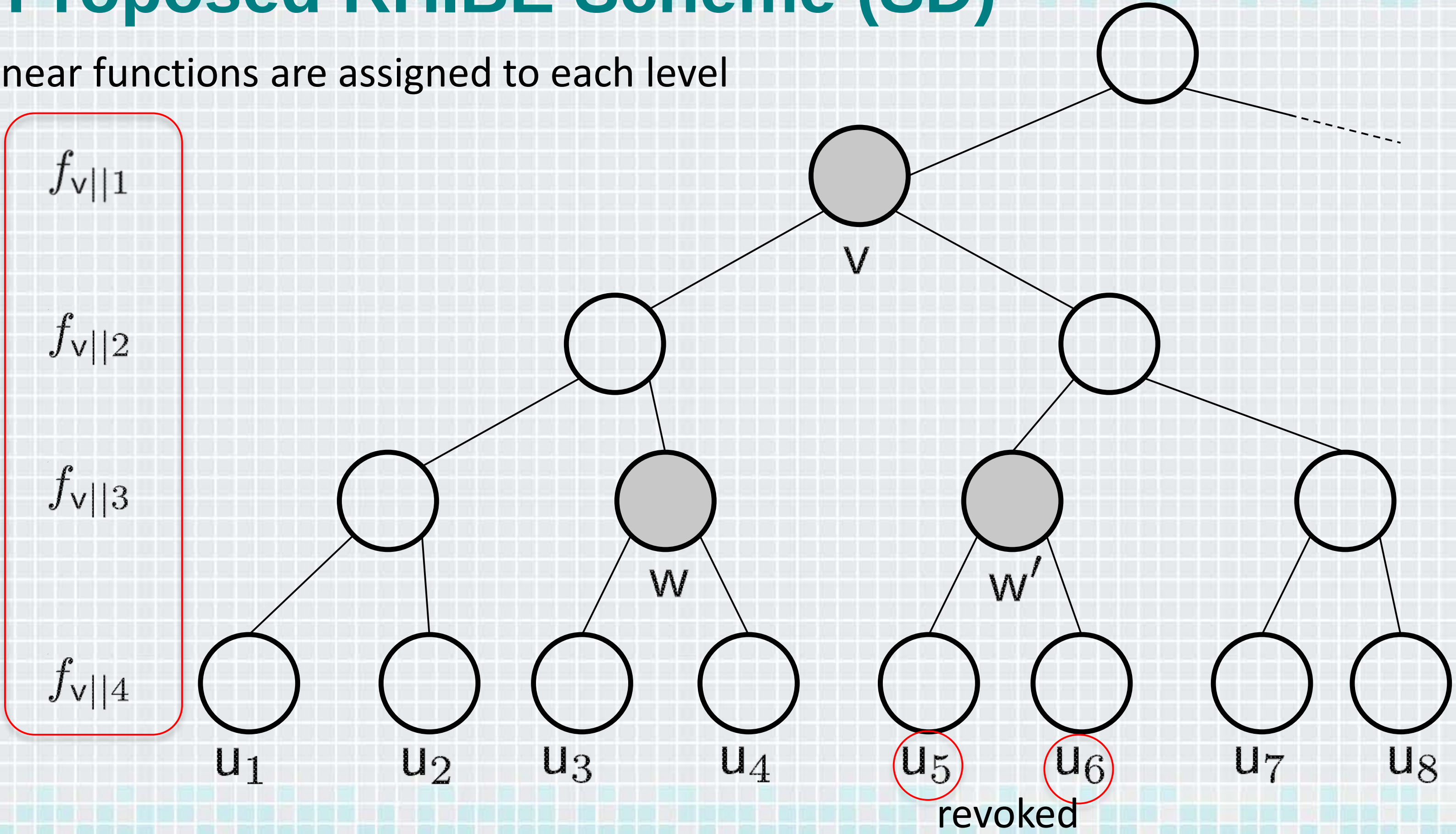
BBG HIBE (ID) — BB IBE (Time)

> If ID|$_k$ is not revoked, then there exists the same θ (CS method)

> With re-randomization for decryption key exposure resistance

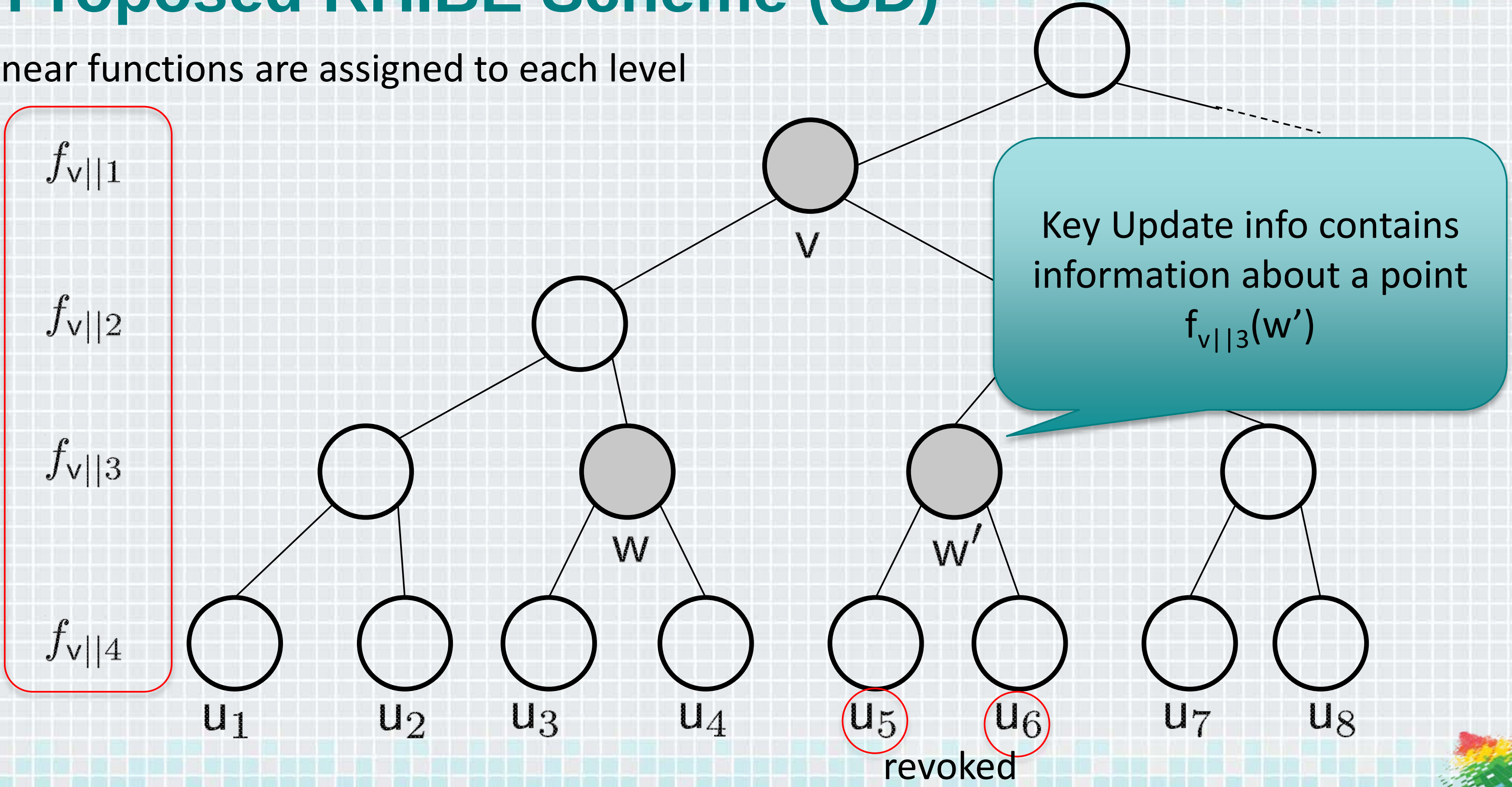
Proposed RHIBE Scheme (SD)

Linear functions are assigned to each level



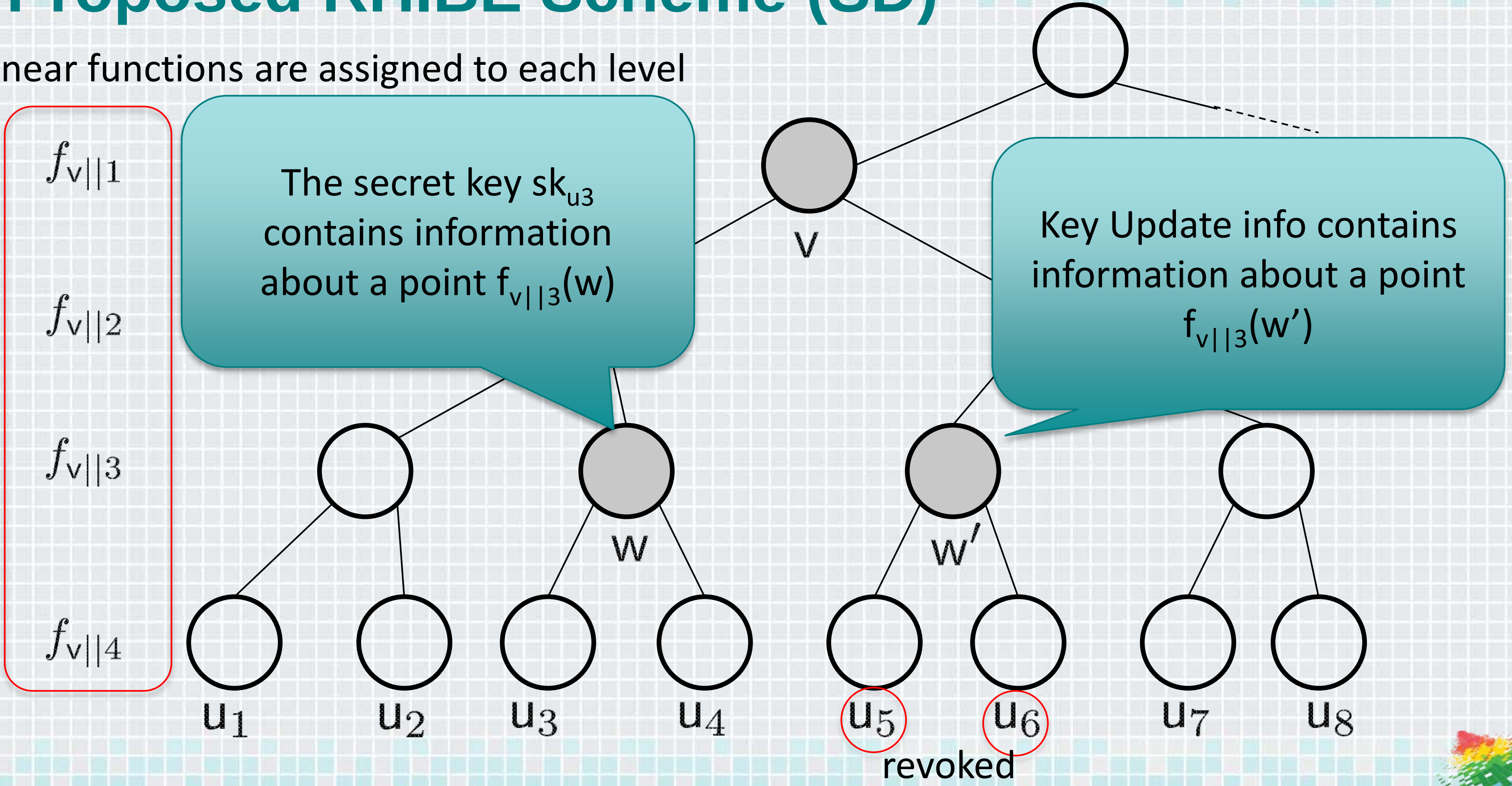
Proposed RHIBE Scheme (SD)

Linear functions are assigned to each level



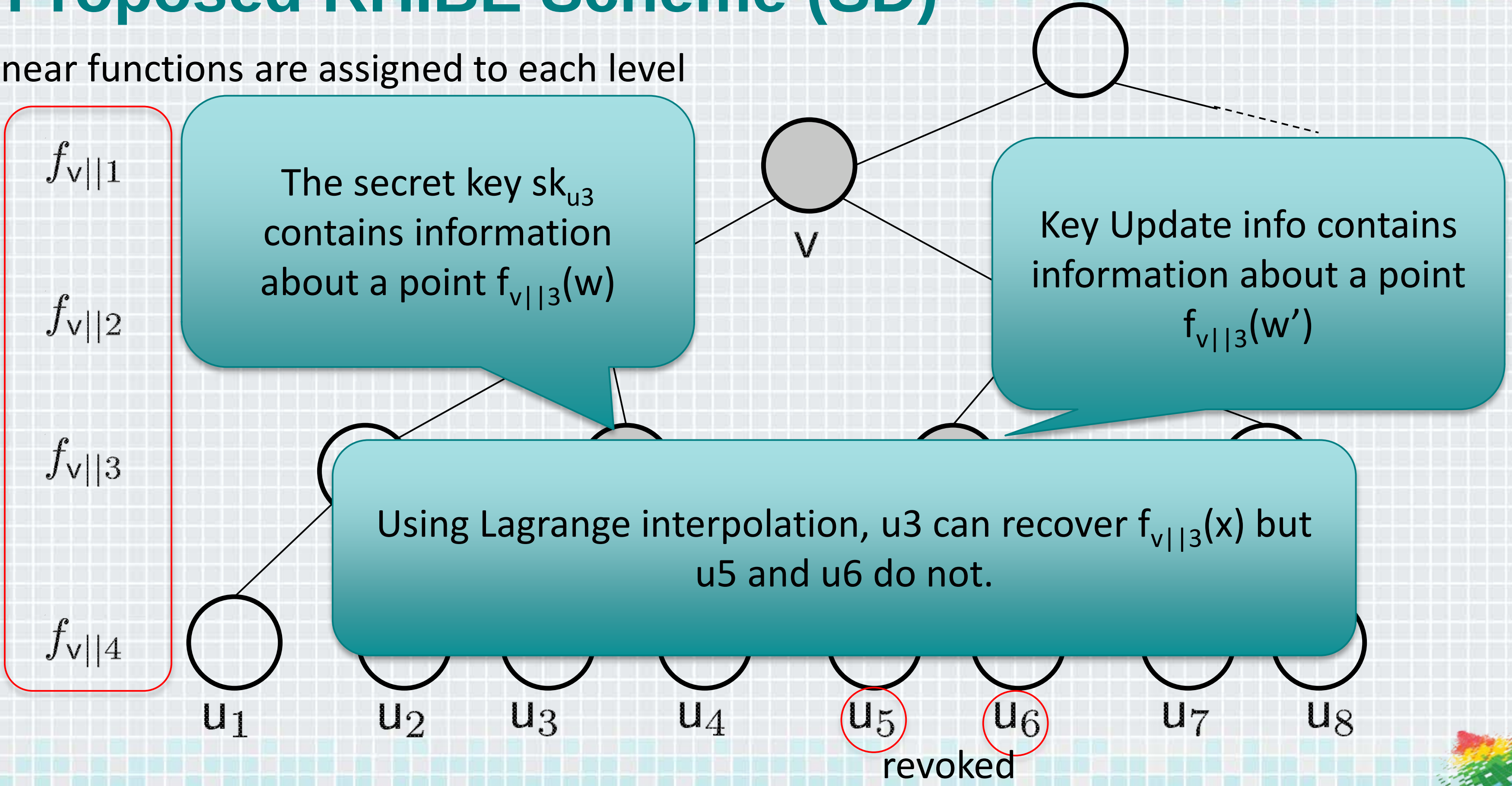
Proposed RHIBE Scheme (SD)

Linear functions are assigned to each level



Proposed RHIBE Scheme (SD)

Linear functions are assigned to each level



Proposed RHIBE Scheme (SD)

- The main part is the same as that of the LLP RIBE scheme.
  - K. Lee, D. H. Lee, and J. H. Park. Efficient revocable identity-based encryption via subset difference methods, eprint.iacr.org/2014/132, 2014.
- One difference is: we introduce the false master key for history-free construction so that sk does not contain the master key α

$$f_y(x) := \mathsf{PRF}_k(y)x + \beta$$

See the paper for details

Comparison

Table 1: Revocable Hierarchical Identity-Based Encryption schemes

	SK size	CT size	KU size	Model	Sec. ag. insiders	DKE resist.	Assum.
Trivial	$\omega(2^\ell)$						
SE13	$O(\ell^2 \log N)$	$O(\ell)$	$O(r \log \frac{N}{r})$	Std., Sel.	✘	✘	static
CS const.	$O(\ell \log N)$	$O(1)$	$O(\ell r \log \frac{N}{r})$	Std., Sel.	✔	✔	q-type
SD const.	$O(\ell (\log N)^2)$	$O(1)$	$O(\ell r)$	Std., Sel., SRL	✔	✔	q-type

Std.: standard model, Sel.: selective security, SRL: selective revocation list [BGK08,LLP14]

ℓ: maximum hierarchical level, N: maximum number of users in the system, r: number of revoked users.

Comparison

Table 1: Revocable Hierarchical Identity-Based Encryption schemes

	SK size	CT size	KU size	Model	Sec. ag. insiders	DKE resist.	Assum.
Trivial	$\omega(2^\ell)$						
SE13	$O(\ell^2 \log N)$	$O(\ell)$	$O(r \log \frac{N}{r})$	Std., Sel.	✘	✘	static
CS const.	$O(\ell \log N)$	$O(1)$	$O(\ell r \log \frac{N}{r})$	Std., Sel.	✔	✔	q-type
SD const.	$O(\ell (\log N)^2)$	$O(1)$	$O(\ell r)$	Std., Sel., SRL	✔	✔	q-type

Std.: standard model, Sel.: selective security, SRL: selective revocation list [BGK08,LLP14]

ℓ: maximum hierarchical level, N: maximum number of users in the system, r: number of revoked users.

DBDH

q-weak Bilinear Diffie-Hellman Inversion

Conclusion and Future work

Table 1: Revocable Hierarchical Identity-Based Encryption schemes

	SK size	CT size	KU size	Model	Sec. ag. insiders	DKE resist.	Assum.
Trivial	$\omega(2^\ell)$						
SE13	$O(\ell^2 \log N)$	$O(\ell)$	$O(r \log \frac{N}{r})$	Std., Sel.	✘	✘	static
CS const.	$O(\ell \log N)$	$O(1)$	$O(\ell r \log \frac{N}{r})$	Std., Sel.	✔	✔	q-type
SD const.	$O(\ell(\log N)^2)$	$O(1)$	$O(\ell r)$	Std., Sel., SRL	✔	✔	q-type

Std.: standard model, Sel.: selective security, SRL: selective revocation list [BGK08,LLP14]

ℓ: maximum hierarchical level, N: maximum number of users in the system, r: number of revoked users.

- RHIBE:
  - History-free update, insider security, short ciphertext, and DKER
- The reduction to the underlying HIBE requires the challenge identity for the security proof.
  - Adaptive-ID secure RHIBE under a static assumption with these desirable properties