The *k*-BDH Assumption Family: Bilinear Cryptography from Progressively Weaker Assumptions

Karyn Benson (UCSD)

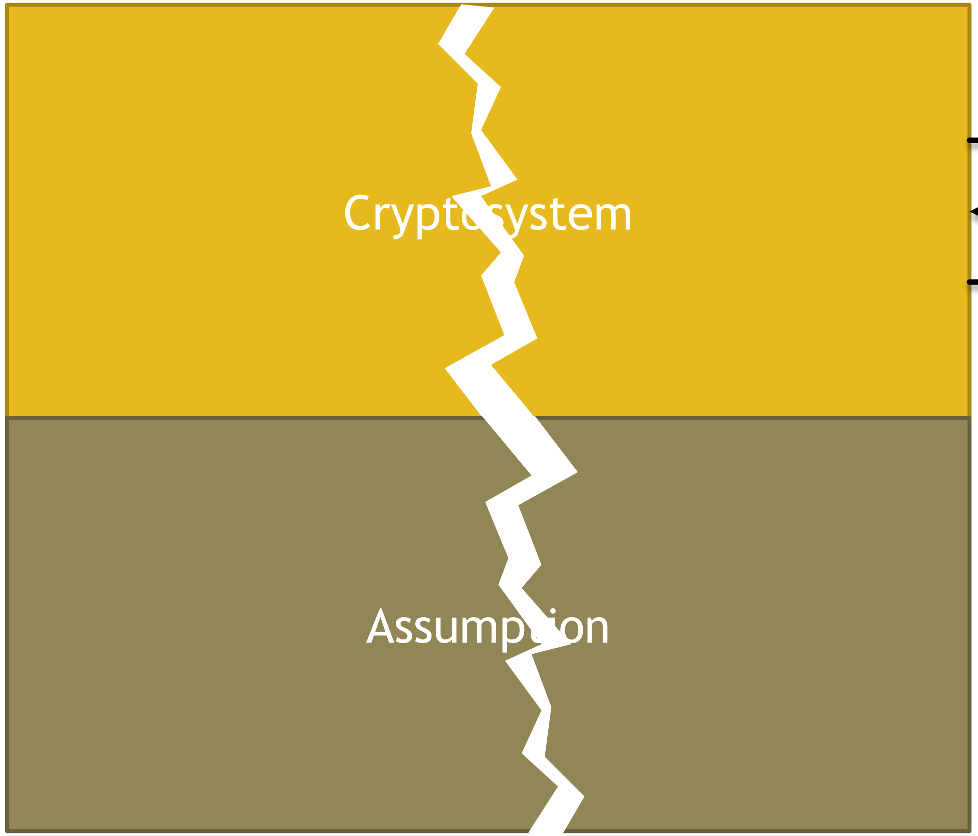
Hovav Shacham (UCSD)

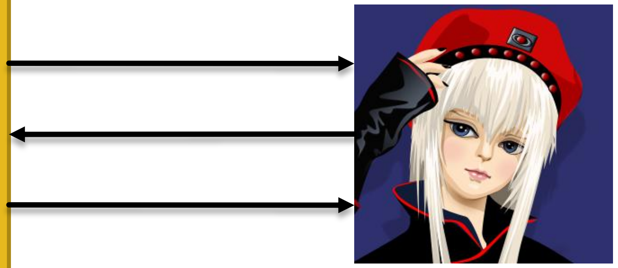
Brent Waters (UT-Austin)

CT-RSA 2013

Provable Security

How to show your cryptosystem is secure:

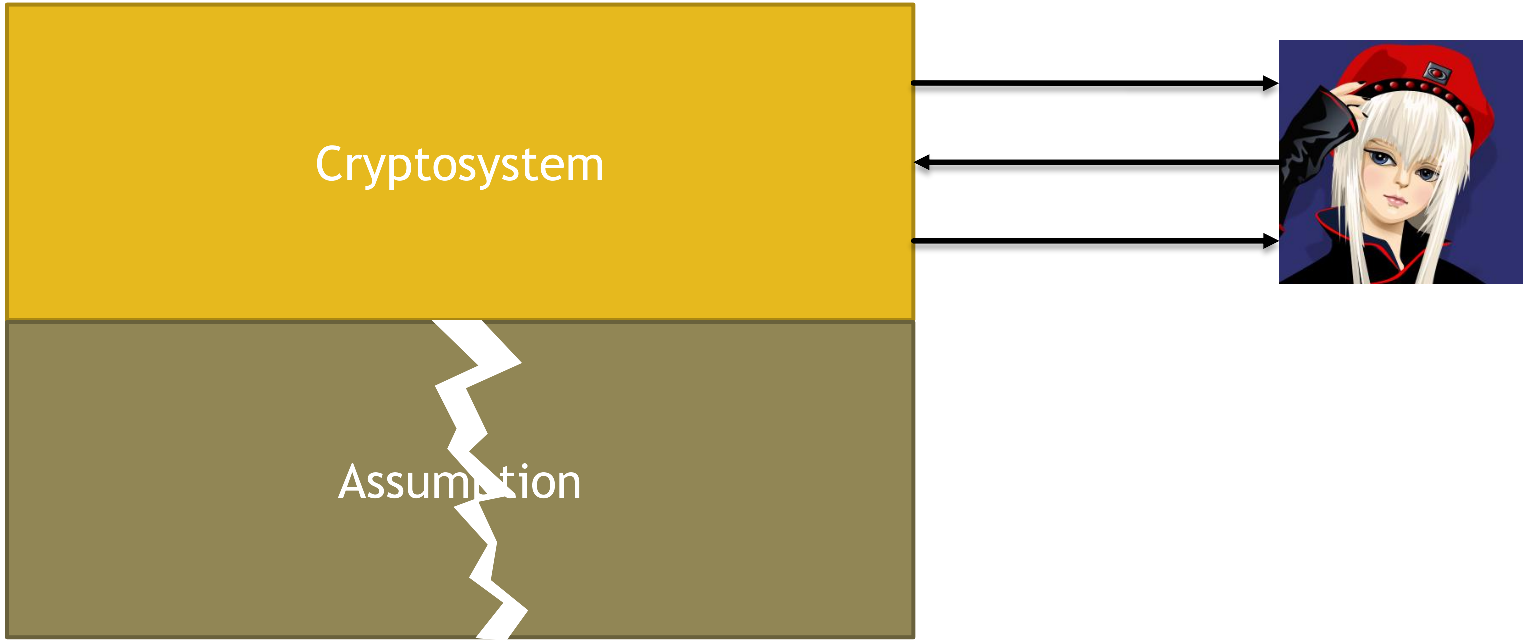




If **Assumption Holds** then **Cryptosystem Secure**

Prove contrapositive

What if the Assumption is False?

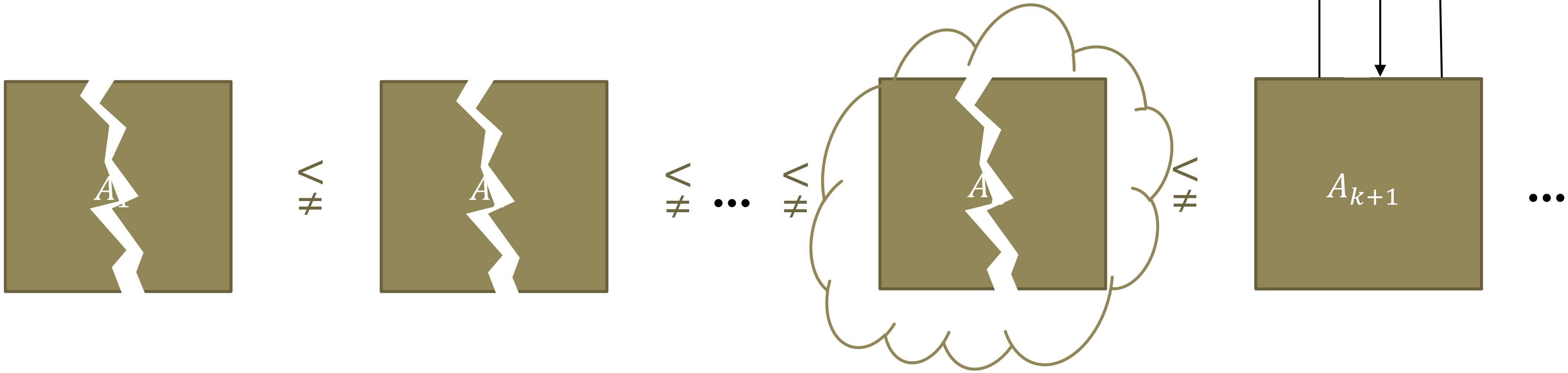


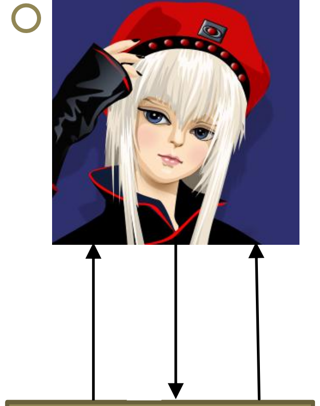
- Cannot reason about security
- Adversary can use the attack on assumption to break cryptosystem

How to Pick a Good Assumption?

- Increase the size of parameters
  - e.g., RSA assumes factoring a large number into 2 primes is hard
  - Factor: 77, 3869, 702619, …
- Use a family of assumptions
  - As you increase a parameter *k* you become more confident in the security of the assumption
  - Example: *k*-Linear [HK07, Sha07]

Family of Strictly Weaker Assumptions





- An assumption A_{k+1} is **weaker** than assumption A_k, if
  - If A_k holds then so does A_{k+1} (Breaking A_{k+1} also breaks A_k)
- The assumption, A_{k+1}, is **strictly weaker** than assumption, A_k, if
  - A_{k+1} is weaker than A_k
  - And an oracle for A_k does not help break A_{k+1}

DDH Assumption

- Given $< g, g^a, g^b, T >$
  - For some $g, g^a, g^b, T \in G$
- Does $T \stackrel{?}{=} g^{ab}$

> It is hard to compute a discrete log in a finite cyclic group G

- No polynomial time algorithm can achieve non-negligible advantage deciding

Bilinear Maps

- $\boldsymbol{e}$: $G \times G \rightarrow G_T$
  - Bilinear: $\boldsymbol{e}(g^a, g^b) = \boldsymbol{e}(g,g)^{ab}$ for all $a, b \in Z_p$
  - Non-Degenerate: If g generates G, then $\boldsymbol{e}(g,g) \neq 1$
  - Computable: $\boldsymbol{e}$ is efficiently computable on all input
- DDH does not hold for groups in which bilinear maps can be computed
  - $< g, g^a, g^b, T \stackrel{?}{=} g^{ab} >$
  - $\boldsymbol{e}(g^a, g^b) \stackrel{?}{=} \boldsymbol{e}(g, T)$

DBDH Assumption

- How can we use DDH in bilinear groups?
- Given $< g, g^a, g^b, g^c, T >$
  - For some $g, g^a, g^b, g^c \in G$ and $T \in G_T$
- Does $T \stackrel{?}{=} \boldsymbol{e}(g, g)^{abc}$

- Hard to compute discrete log: in G and G_T
- Bilinear maps have 2 inputs
- Can't undo a bilinear map

- No polynomial time algorithm can achieve non-negligible advantage deciding

Decision Linear (DLIN) Assumption

- How can we use DDH in settings where bilinear maps exist?
- Given $< g, g^{s_1}, g^{s_2}, g^{s_1 r_1}, g^{s_2 r_2}, T >$
  - $g, g^{s_1}, g^{s_2}, g^{s_1 r_1}, g^{s_2 r_2}, T \in G$
- Does $T \stackrel{?}{=} \boxed{g^{r_1}}\boxed{{}^{+r_2}}$

This is like 2 DDH problems:
$g, g^{s_1}, \boxed{g^{r_1}}, g^{s_1 r_1}$

- It is hard to compute discrete logs
- Bilinear maps only pair 2 elements (not 3: $g^{s_1}, g^{s_2}, g^{r_1+r_2}$)

- No polynomial time algorithm can achieve non-negligible advantage deciding, even in generic bilinear groups [BBS04]
- Only a decisional problem – computationally same as DDH

k-Linear Family of Assumptions

This is like k DDH problems

- k-Linear generalizes the Linear Assumption
  - 1-Linear is DDH
  - 2-Linear is Linear Assumption
- For $k \geq 1$ Given $< g, g^{s_1}, \ldots, g^{s_k}, g^{s_1 r_1}, \ldots, g^{s_k r_k}, T >$
  - $g, g^{s_1}, \ldots, g^{s_k}, g^{s_1 r_1}, \ldots, g^{s_k r_k}, T \in G$
- Does $T \stackrel{?}{=} g^{r_1 + \cdots + r_k}$
- No polynomial time algorithm can achieve non-negligible advantage deciding
- Only a decisional problem – computationally same as DDH

How are DLIN and DBDH Related?

- If DLIN holds, then so does DBDH

$$< g, g^{s_1}, g^{s_2}, g^{s_1 r_1}, g^{s_2 r_2}, T \stackrel{?}{=} g^{r_1+r_2} >$$

DLIN Instance

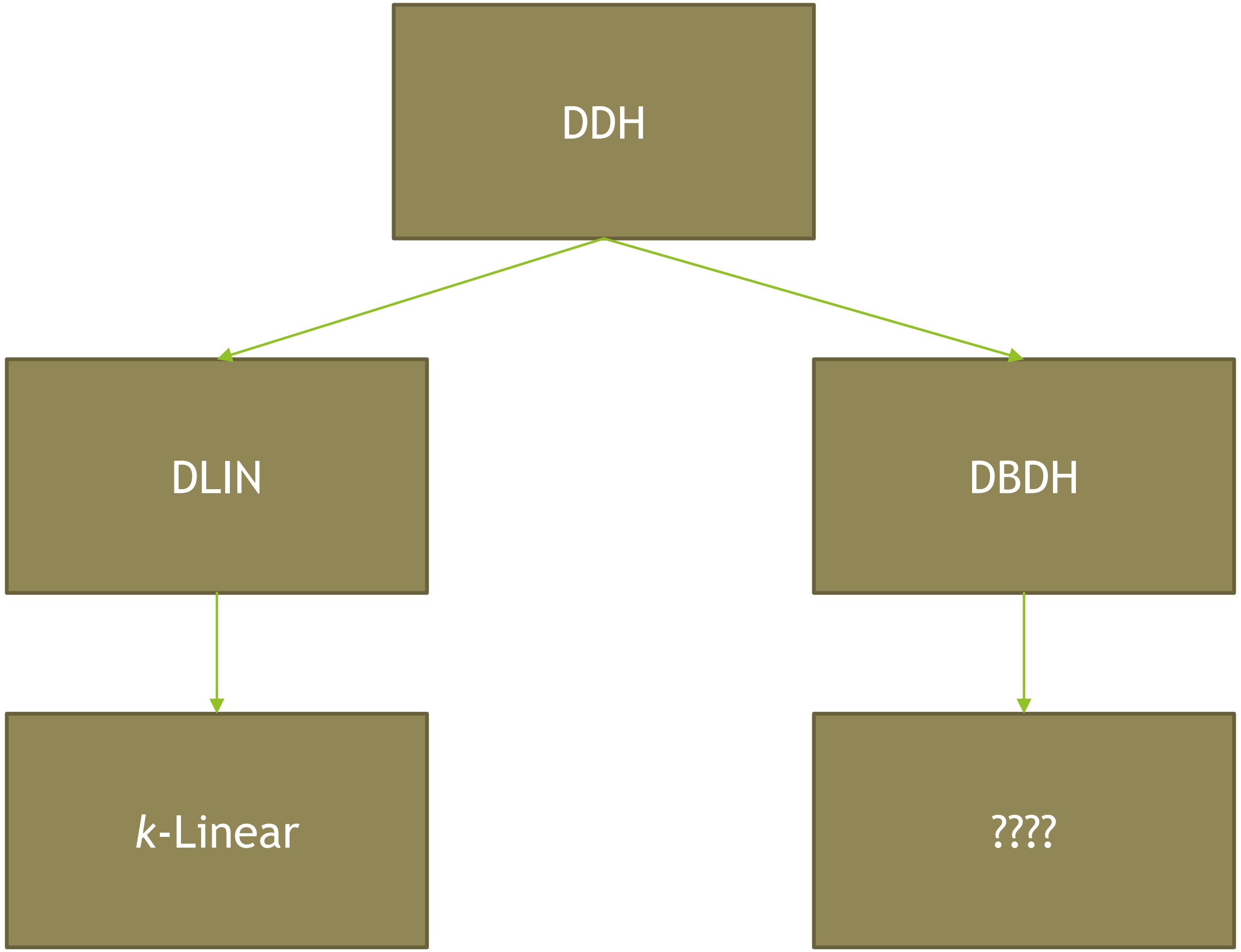
$$< g, g^{s_1}, g^{s_2}, T \stackrel{?}{=} g^{r_1+r_2}, \boldsymbol{e}(g^{s_1 r_1}, g^{s_2}) \cdot \boldsymbol{e}(g^{s_2 r_2}, g^{s_1}) > =$$

$$< g, g^{s_1}, g^{s_2}, T \stackrel{?}{=} g^{r_1+r_2}, \boldsymbol{e}(g, g)^{(s_1 s_2)(r_1+r_2)} >$$

DBDH Decider

[BW06]

Can extend DBDH to a Family of Assumptions?



- Why?
  - For $k > 2$ unclear how *k*-Linear and DBDH are related
  - *k*-Linear only operates in the source group
  - Example: Boneh-Boyen IBE – the message is hidden in target group

This is should be like *k* DBDH problems

Failed Attempt

- Given $g, g^a, g^b, g^{s_1}, \ldots, g^{s_k}, g^{s_1 r_1}, \ldots, g^{s_k r_k} \epsilon$ and $T \epsilon G_T$
- Does $T \stackrel{?}{=} \prod_i \boldsymbol{e}(g, g^{s_i})^{abr_i} = \prod_i \boldsymbol{e}(g, g\ )^{abs_i r_i} = \boldsymbol{e}(g, g)^{ab(s_1 r_1 + \cdots + s_k r_k)}$
- Embeds k DBDH instances: $(g, g^a, g^b, g^{s_i r_i}, \boldsymbol{e}(g, g)^{ab(s_i r_i)})$
- … But is equivalent to DBDH:
  $(g, g^a, g^b, \prod_i g^{s_i r_i} = g^{(s_1 r_1 + \cdots + s_k r_k)}, T \stackrel{?}{=} \prod_i \boldsymbol{e}(g, g\ )^{abs_i r_i})$

k-BDH Assumption

- Given $g, g^a, g^b, g^{s_1}, \dots, g^{s_k}, g^{s_1 r_1}, \dots, g^{s_k r_k} \epsilon\ G$ and $T \epsilon\ G_T$
- Does $T \stackrel{?}{=} \prod_i \boldsymbol{e}(g,g)^{abr_i} = \prod_i \boldsymbol{e}(g^{s_i}, g^{s_i})^{(a/s_i)(b/s_i)r_i} = \boldsymbol{e}(g,g)^{ab(r_1+\cdots+r_k)}$
- Embeds k DBDH instances: $(g^{s_i}, g^a, g^b, g^{s_i r_i}, \boldsymbol{e}(g^{s_i}, g^{s_i})^{(a/s_i)(b/s_i)r_i})$
- … And is a family of strictly weaker assumptions!

A Family of Weaker Assumptions

- If the *k*-BDH assumption holds, so does the *(k+1)*-BDH assumption

$$g, g^x, g^y, v_1, \dots, v_k, v_1^{r_1}, \dots, v_k^{r_k}, T \stackrel{?}{=} \prod_{1\le i\le k} \boldsymbol{e}(g,g)^{xyr_i}$$

k-BDH Instance

$$v_{k+1}, r_{k+1}$$

Random Values

$$< g, g^x, g^y, v_1, \dots, v_{k+1}, v_1^{r_1}, \dots, v_{k+1}^{r_{k+1}},$$

$$T \cdot \boldsymbol{e}(g^x, g^y)^{r_{k+1}} \stackrel{?}{=} \prod_{1\le i\le k+1} \boldsymbol{e}(g,g)^{xyr_i} >$$

(k+1)-BDH Decider

Evidence of a Family of Strictly Weaker Assumptions

- An oracle for k-BDH does not help in deciding a (k+1)-BDH instance
- Similar to the separation proof of k-Linear [Sha07]
- Generic Group Model [BS84,Nechaev94,Shoup97]
  - Interact with adversary using an idealized version of groups
  - Bound the probability of finding an inconsistency if actual groups were used

- Oracle to k-BDH is implemented as a modified k-multilinear map
  - maps k elements in G and one element G_T to an element group G_M

Application: IBE

- Fits in the Boneh-Boyen Framework
- Base switching techniques needed

- Setup:
  - Public parameters: $g, u = g^x, v_1 = g^{s_1}, \ldots, v_k = g^{s_k}, v_1^{\hat{r}_1}, \ldots, v_k^{\hat{r}_k}, w_1, \ldots, w_k$
  - Master key: $s_1, \ldots, s_k, \hat{r}_1, \ldots, \hat{r}_k, x$
- KeyGen(ID):
  - Select random $n_1, \ldots, n_k \in Z_p^*$
  - For each $1 \le i \le k$ output $\left(K_{A,i}, K_{B,i}\right) = (g^{x\hat{r}_i}\left(w_i u^{\mathrm{ID}}\right)^{n_i}, v_i^{n_i})$
- Encrypt (m, ID):
  - Select random $y_1, \ldots, y_k \in Z_p^*$
  - Output $C_0 = m \prod_{1 \le i \le k} \boldsymbol{e}\left(g^x, v_i{}^{\hat{r}_i}\right)^{y_i}$
  - For each $1 \le i \le k$ output $\left(C_{A,i}, C_{B,i}\right) = (v_i^{y_i}, \left(w_i u^{\mathrm{ID}}\right)^{y_i})$
- Decrypt(c):
  - $\frac{C_0 \cdot \prod_{1 \le i \le k} \boldsymbol{e}(K_{B,i}, C_{B,i})}{\prod_{1 \le i \le k} \boldsymbol{e}(K_{A,i}, C_{A,i})} = \frac{m \prod_{1 \le i \le k} \boldsymbol{e}\left(g^x, v_i{}^{\hat{r}_i}\right)^{y_i} \cdot \prod_{1 \le i \le k} \boldsymbol{e}(v_i^{n_i}, \left(w_i u^{\mathrm{ID}}\right)^{y_i})}{\prod_{1 \le i \le k} \boldsymbol{e}(g^{x\hat{r}_i}\left(w_i u^{\mathrm{ID}}\right)^{n_i}, v_i^{y_i})} = m$

Conclusions

- Goal: Introduction the *k*-BDH Family of Assumptions
  - Relationship to standard assumptions (DDH, *k*-Linear, DBDH)
- It is a family of strictly weaker assumptions
- Usable: We construct an IBE in the Boneh-Boyen Framework

- Future Work
  - IBE construction grows with *k* (public parameters, keys, encryption)
  - Different applications
- http://eprint.iacr.org/2012/687

Security in knowledge

Efficient Delegation of Key Generation and Revocation Functionalities in Identity-Based Encryption

Jae Hong Seo

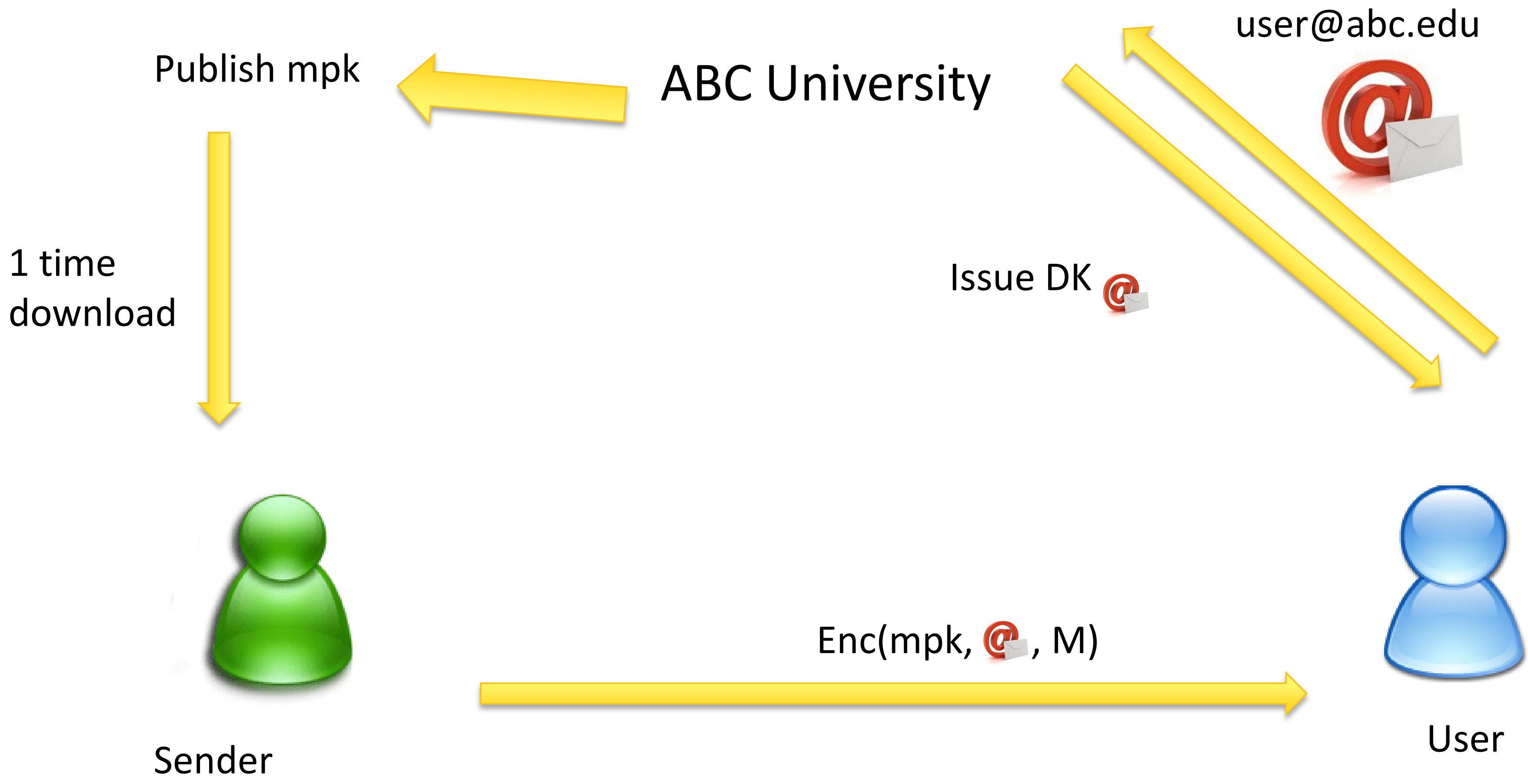
Myongji University, Republic of Korea

RSACONFERENCE2013

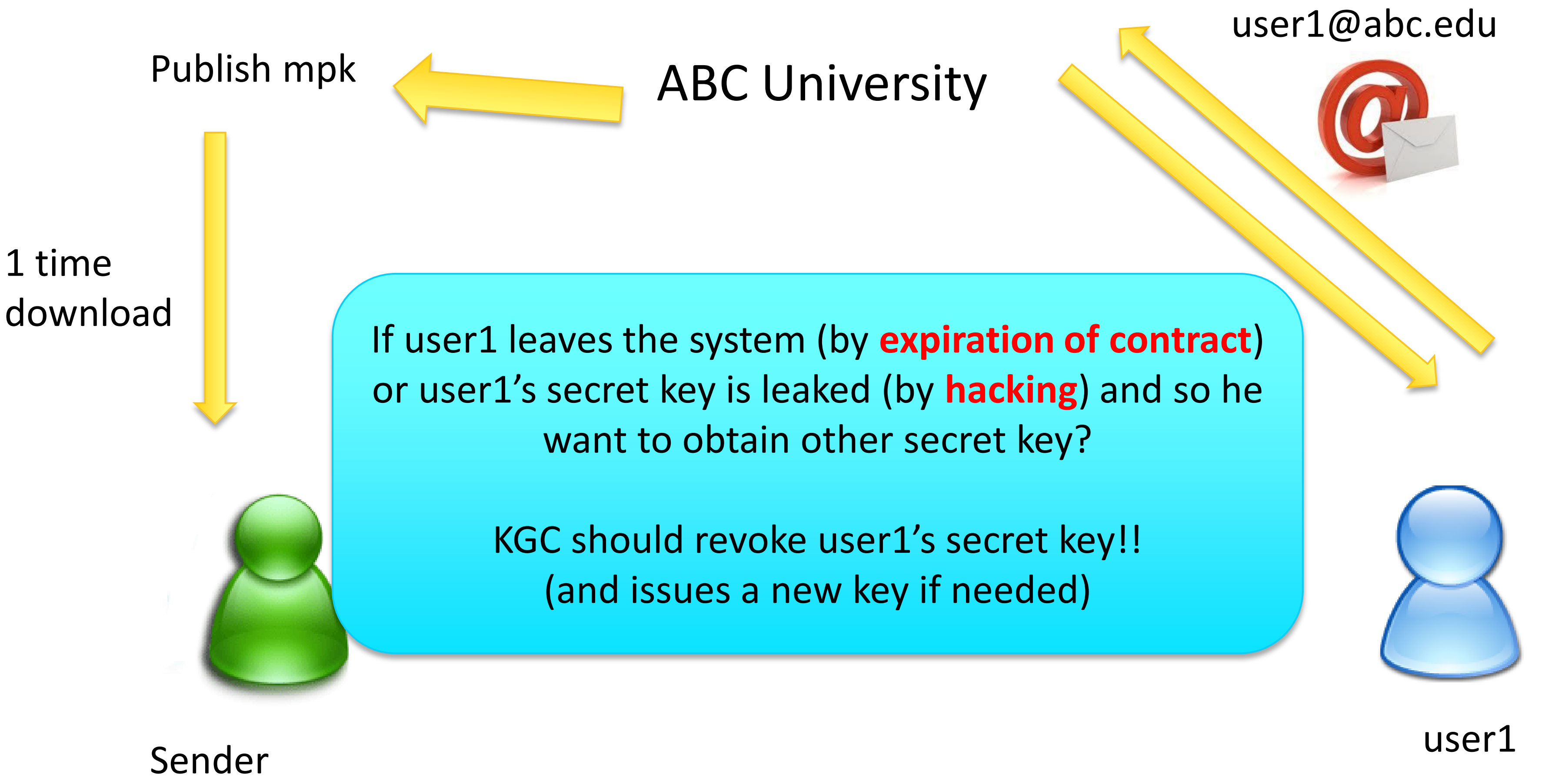
Session ID: CRYP-F41

Session Classification: Intermediate

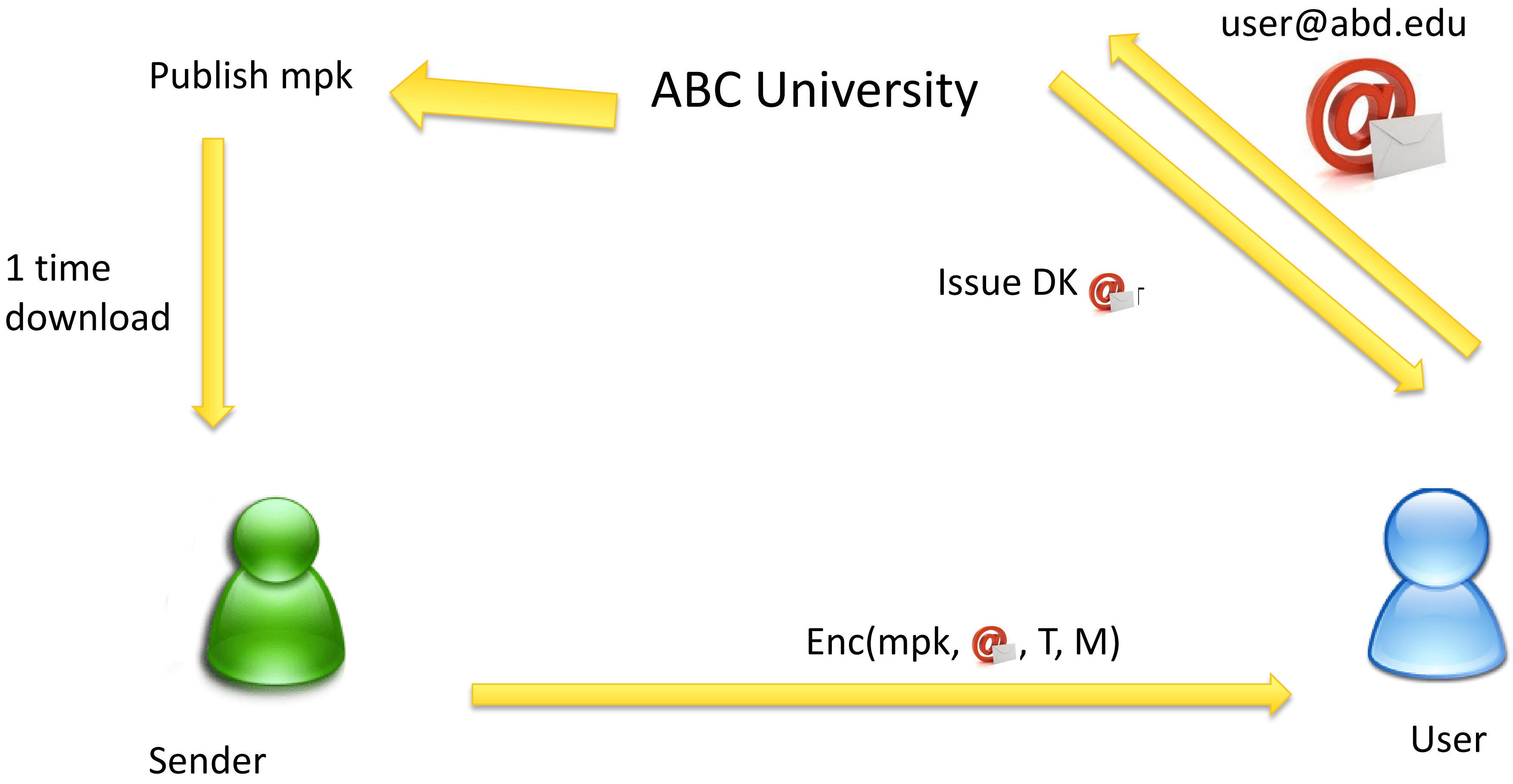
Identity Based Encryption



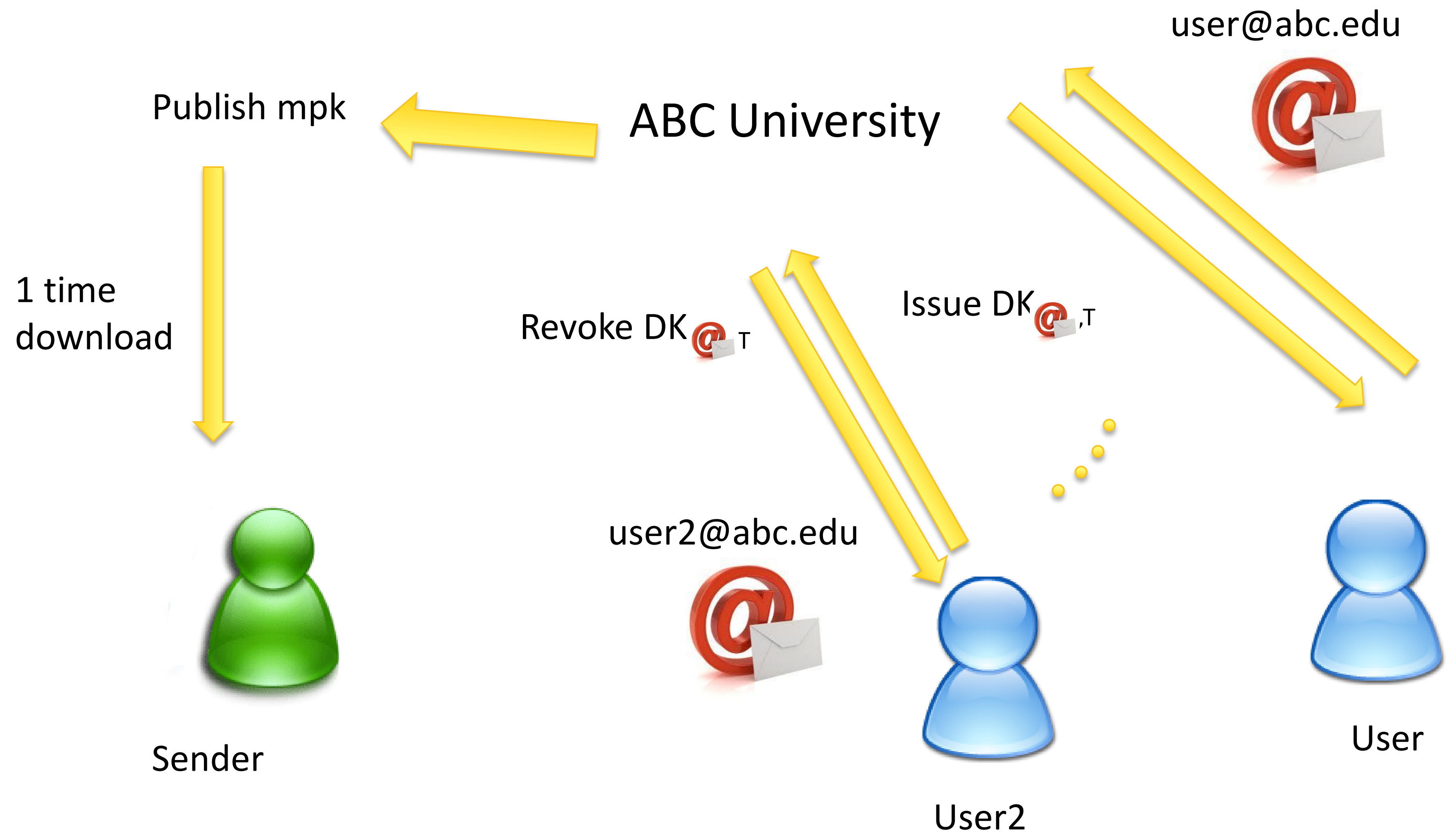
Revocation Functionality in Identity-Based Encryption



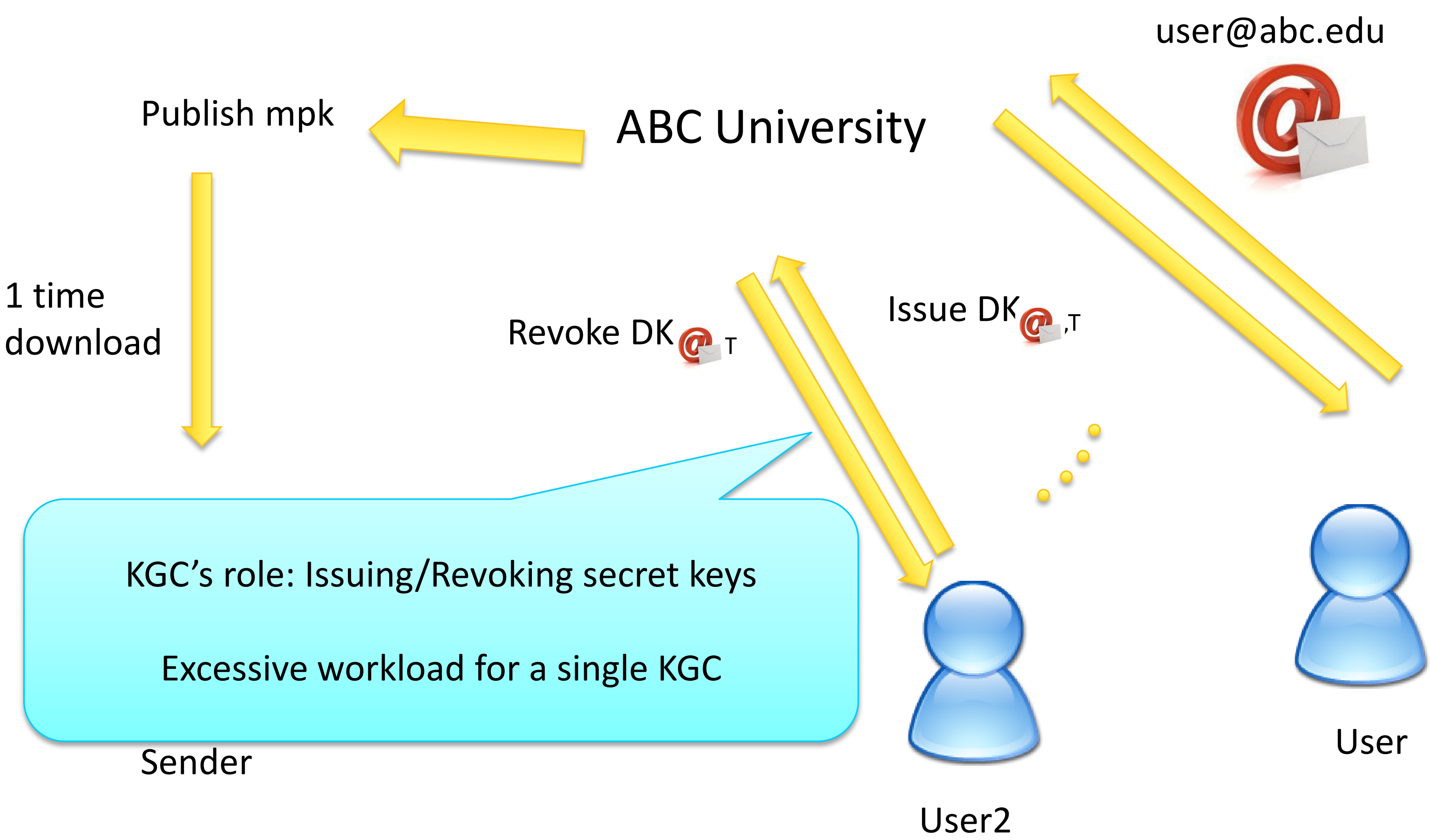
Trivial Approach for Revocation Functionality in IBE



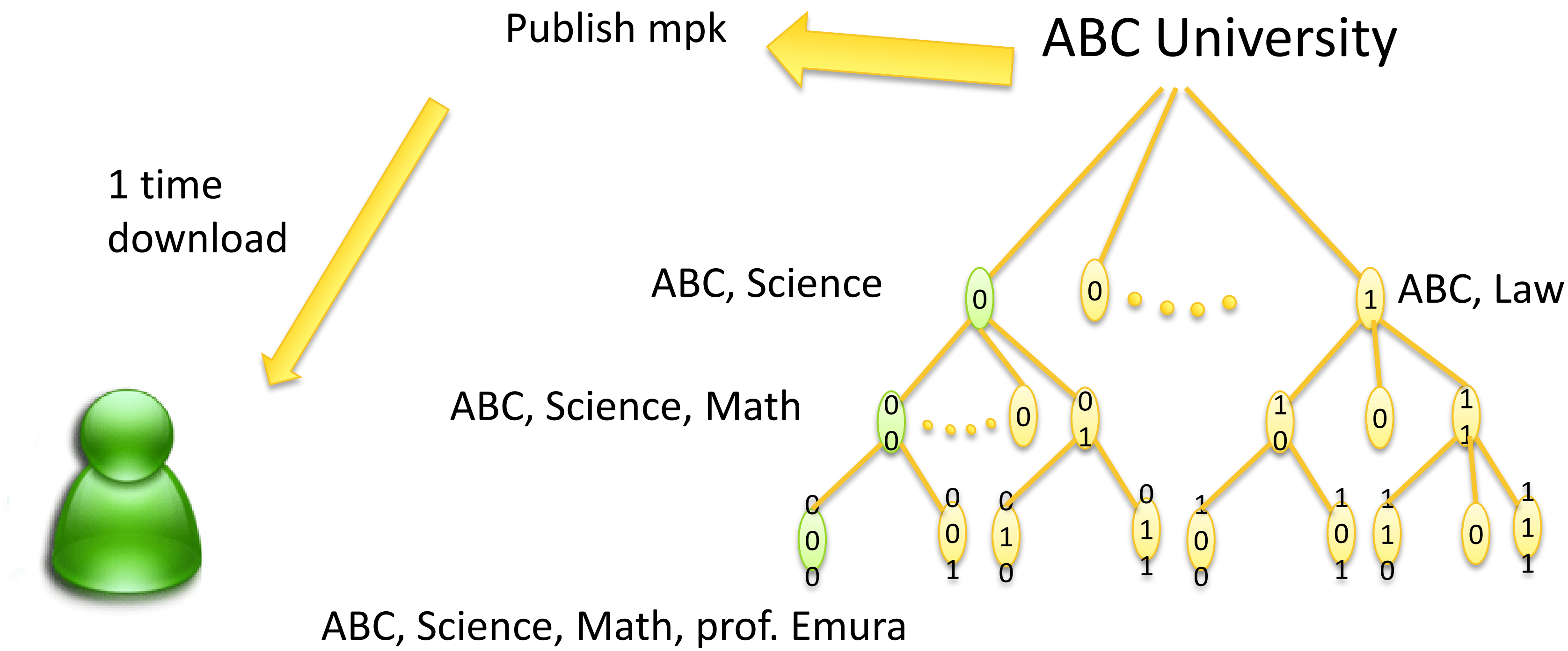
Trivial Approach for Revocation Functionality in IBE



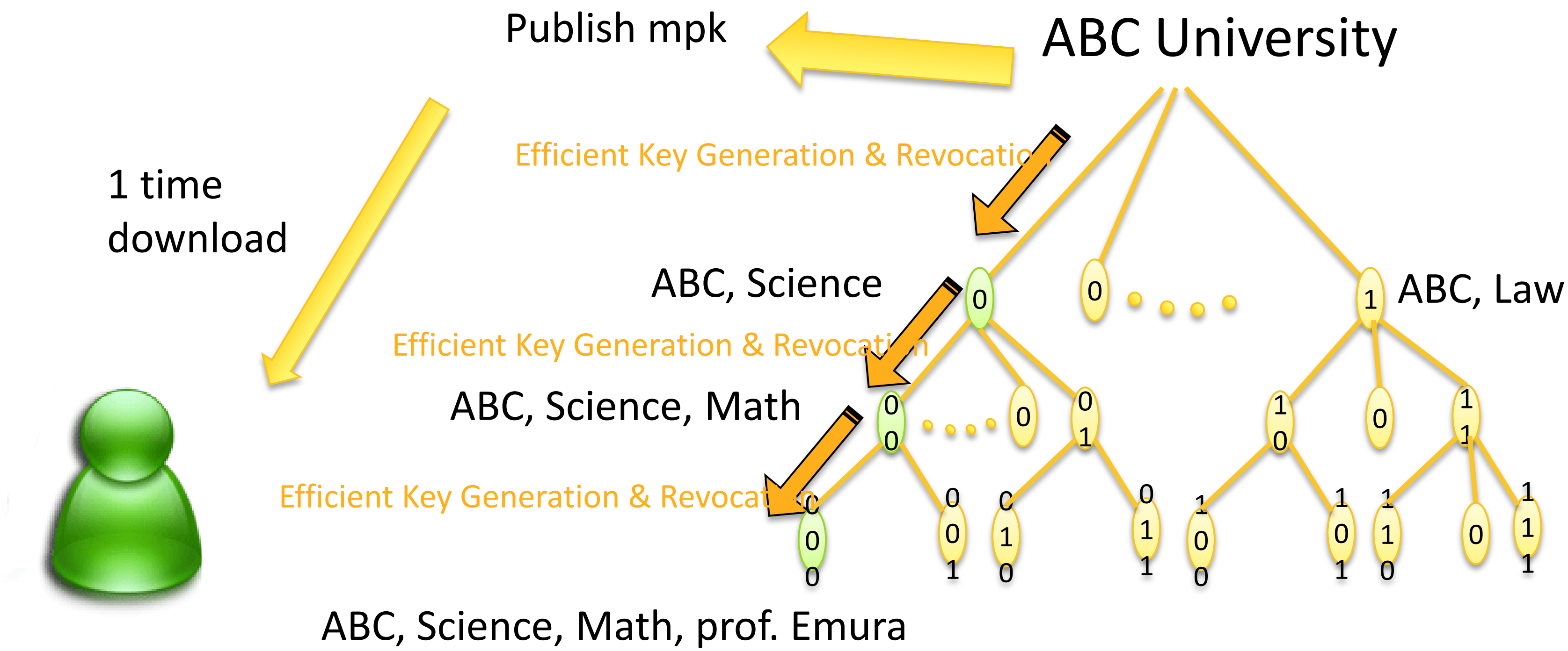
Trivial Approach for Revocation Functionality in IBE



Our Goal:
Delegation of KGC's Roles (Key Generation & Revocation)



Our Goal:
Delegation of KGC's Roles (Key Generation & Revocation)

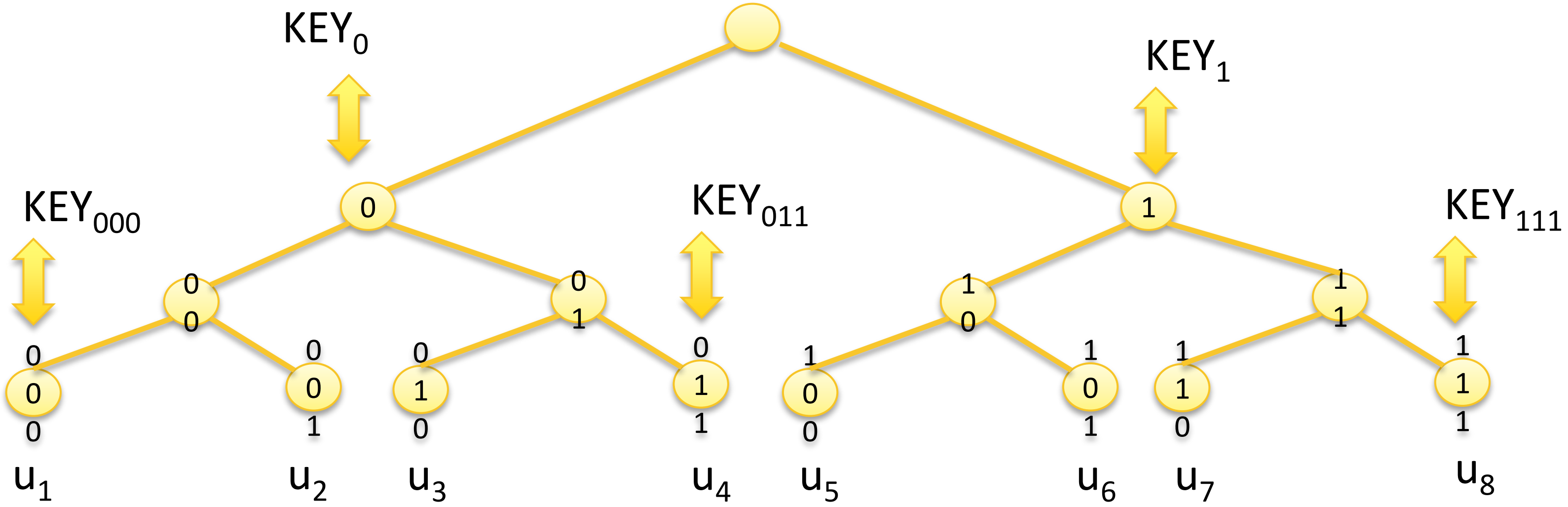


Outline

- Previous Approaches
  - Revocable Symmetric Key Encryption: Broadcast Encryption
  - Revocable Identity-Based Encryption
- Trivial Approach – Exponentially large secret key
- Our Approach – Asymmetric trade
- Further Study

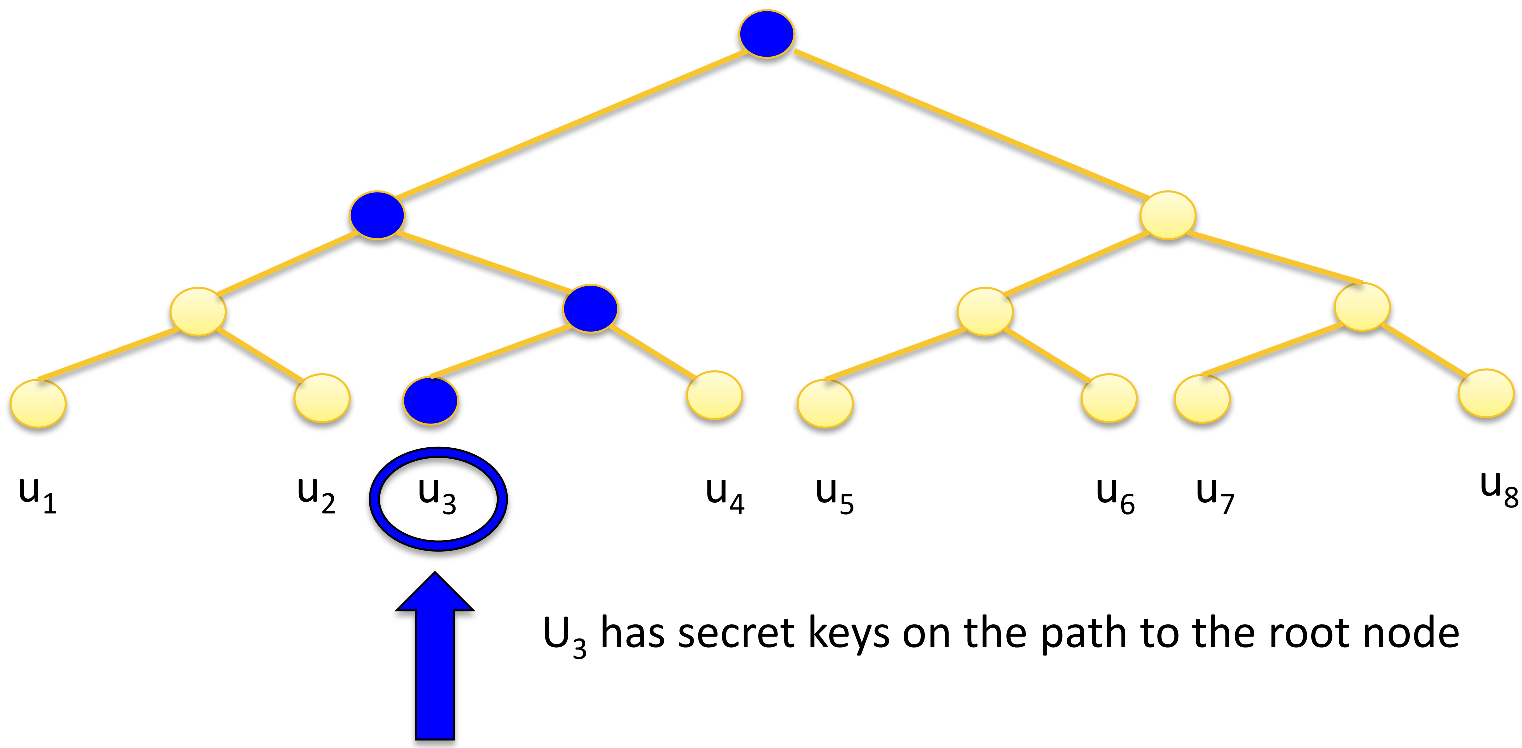
Broadcast Encryption (BE) Technique

Each node has the corresponding key.



We consider a binary tree kept by KGC

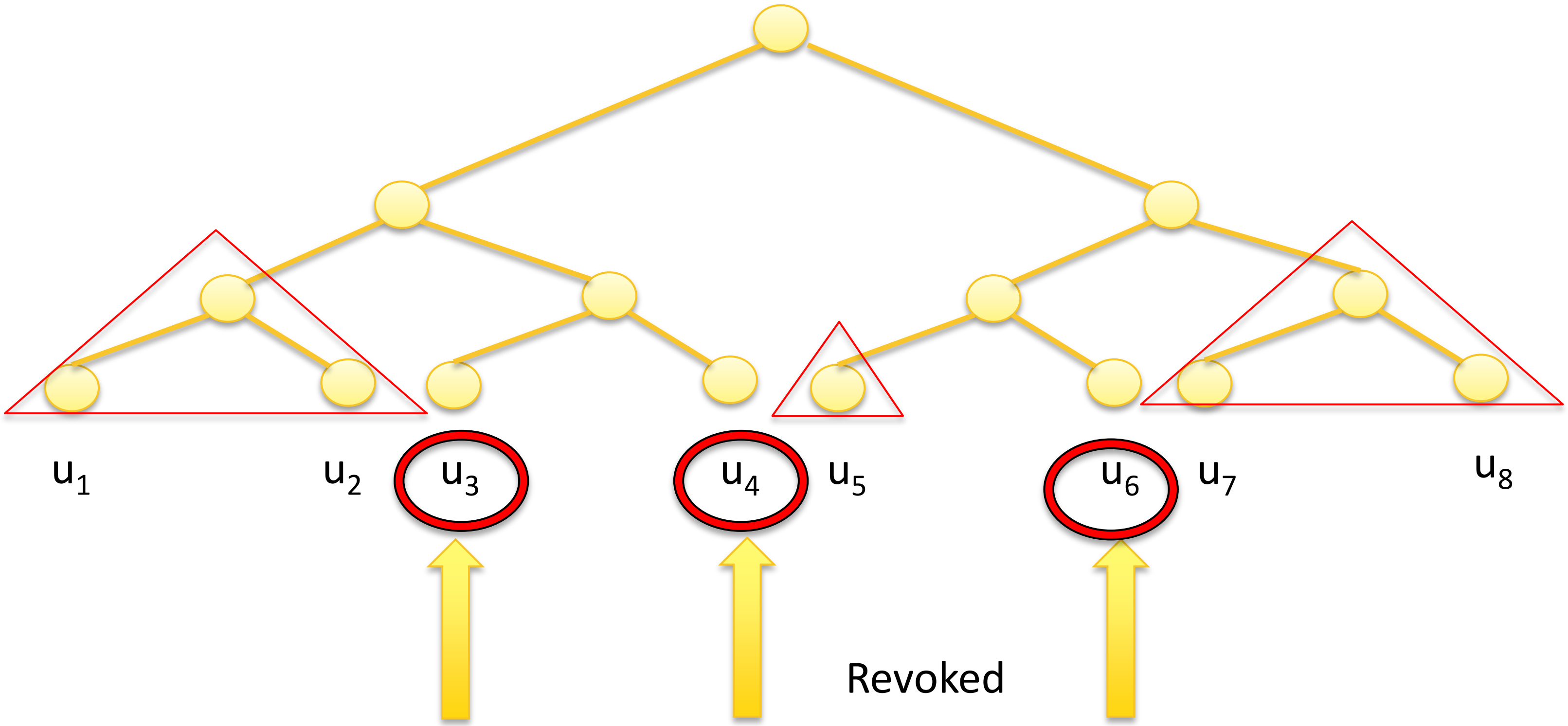
Broadcast Encryption (BE) Technique



U_3 has secret keys on the path to the root node

Broadcast Encryption (BE) Technique

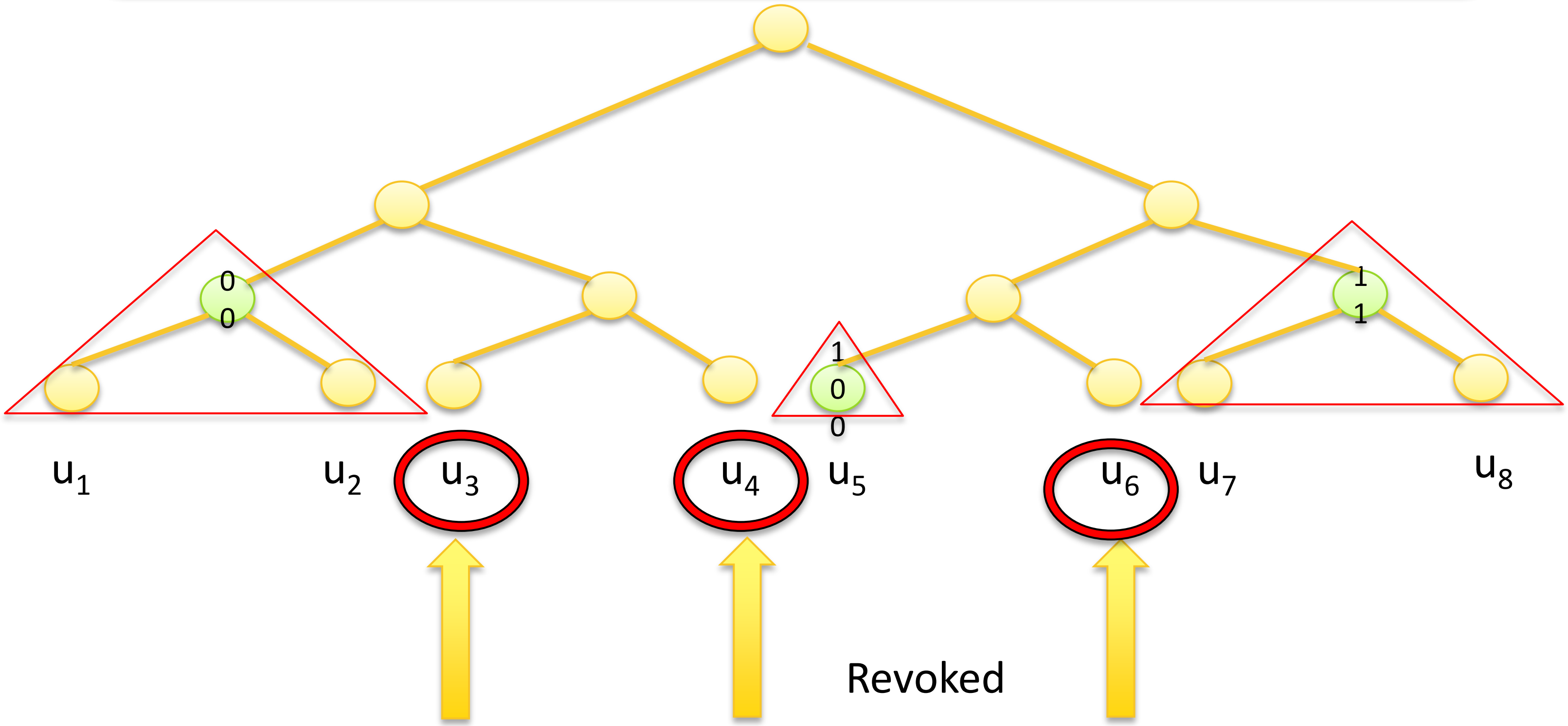
If u_3, u_4, and u_6 are revoked, first compute triangles containing only non-revoked users.



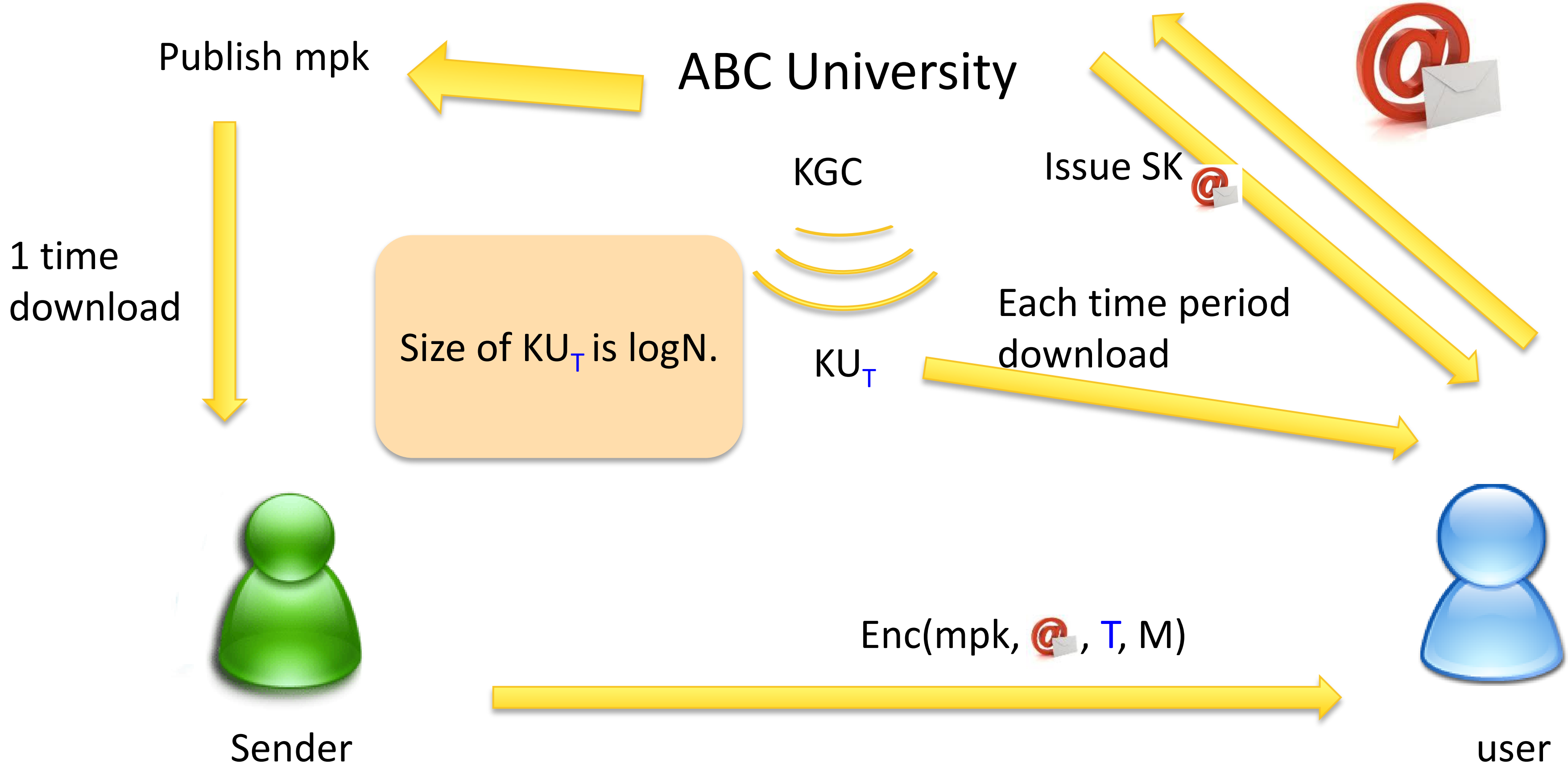
Broadcast Encryption (BE) Technique

Encrypt a session key using KEY_{00}, KEY_{101}, KEY_{11}.
Then, only non-revoked user can recover the session key.

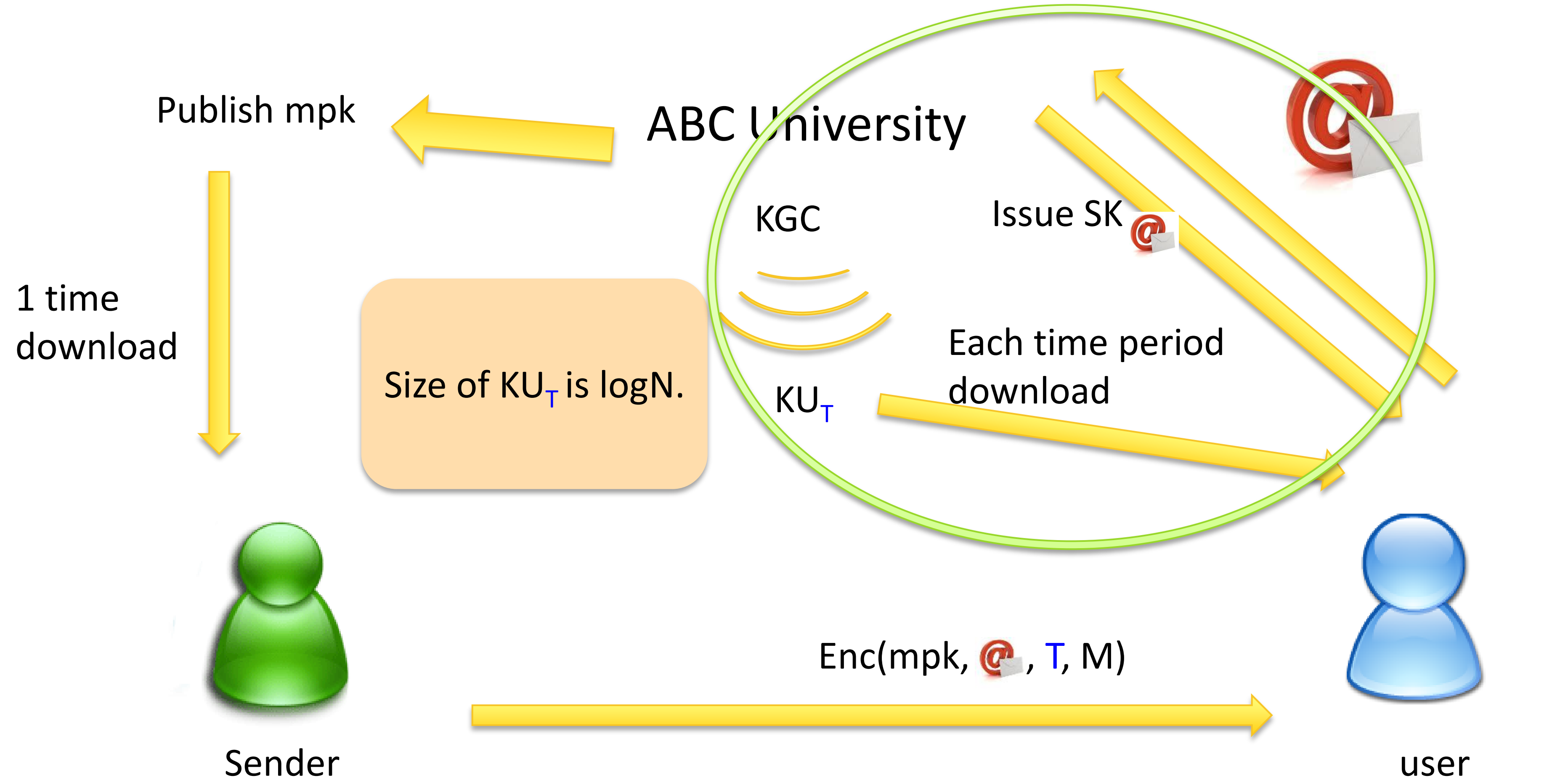
# of triangles ~ log N (where N is # of leaf nodes)



Revocable IBE:
Combining BE technique with IBE scheme

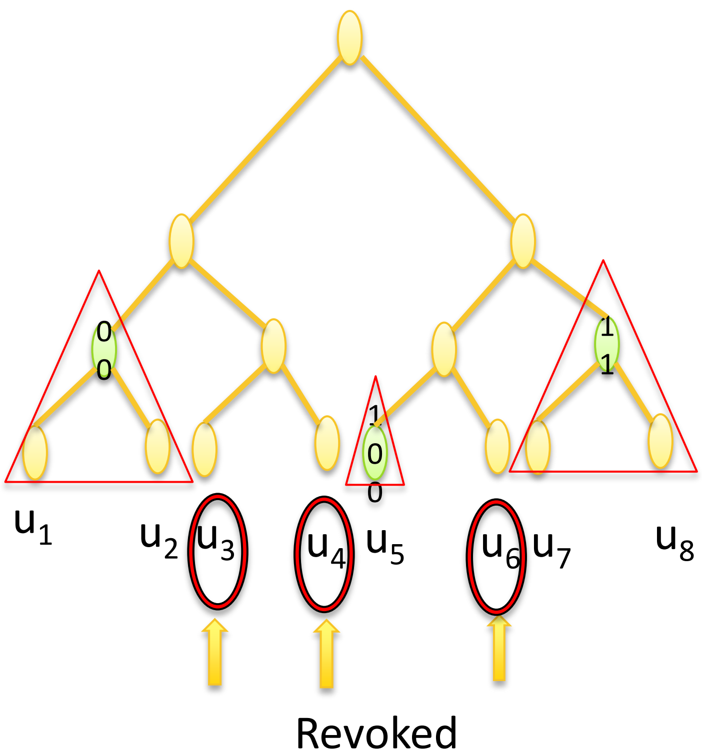


Revocable IBE: Combining BE technique with IBE scheme

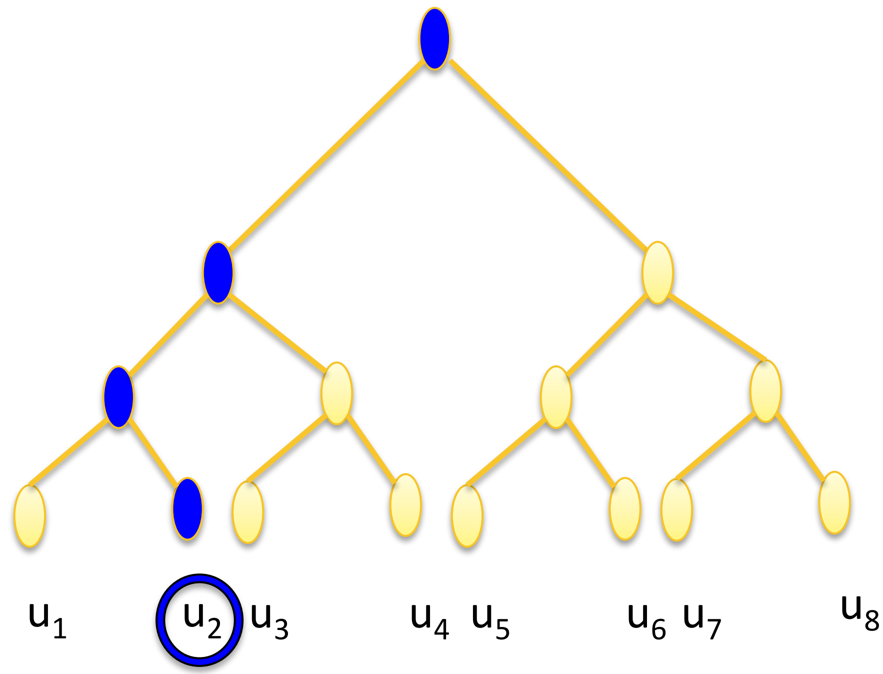


Revocable IBE:
Combining BE technique with IBE scheme

Key Revocation

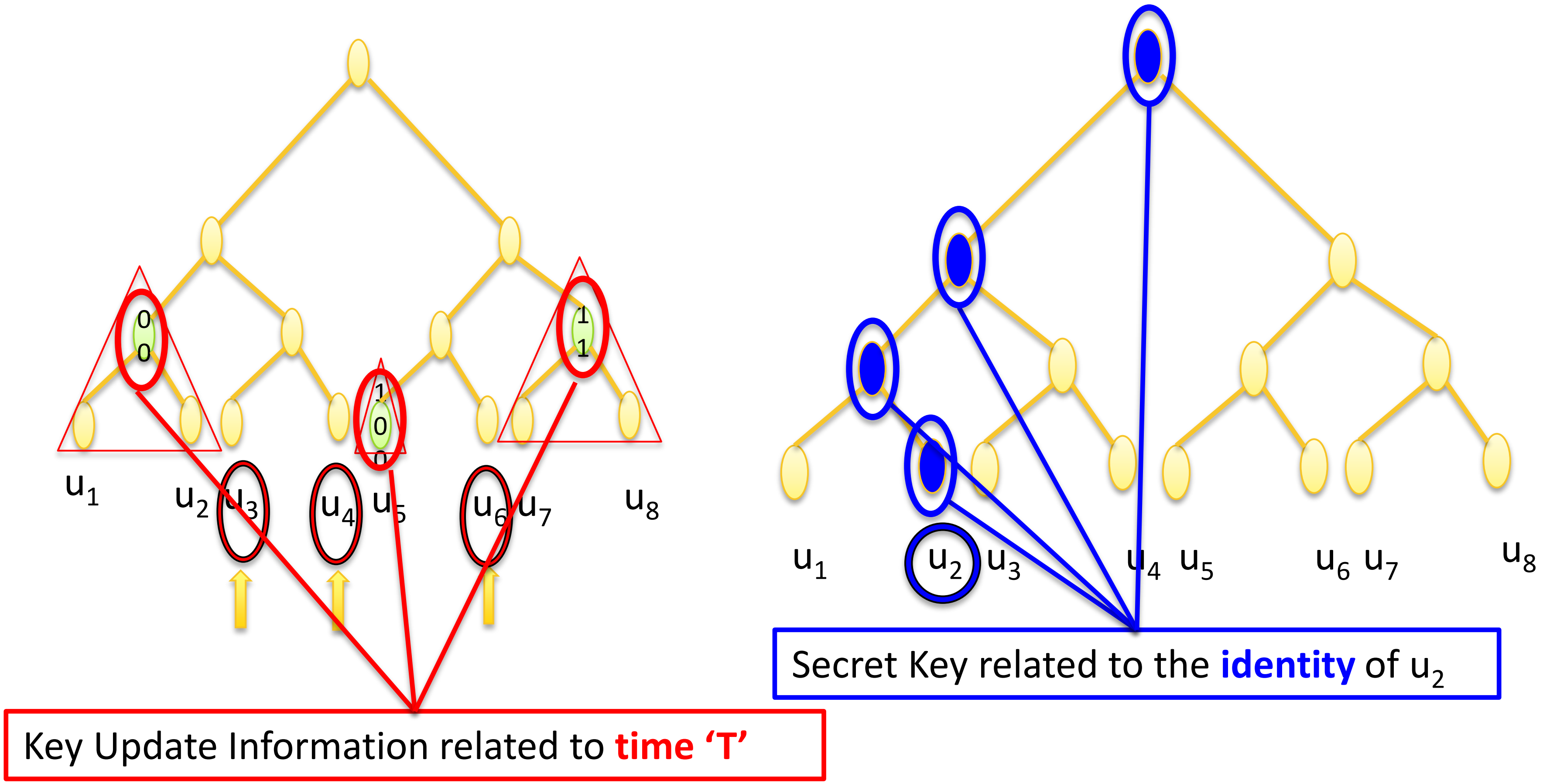


Key Generation



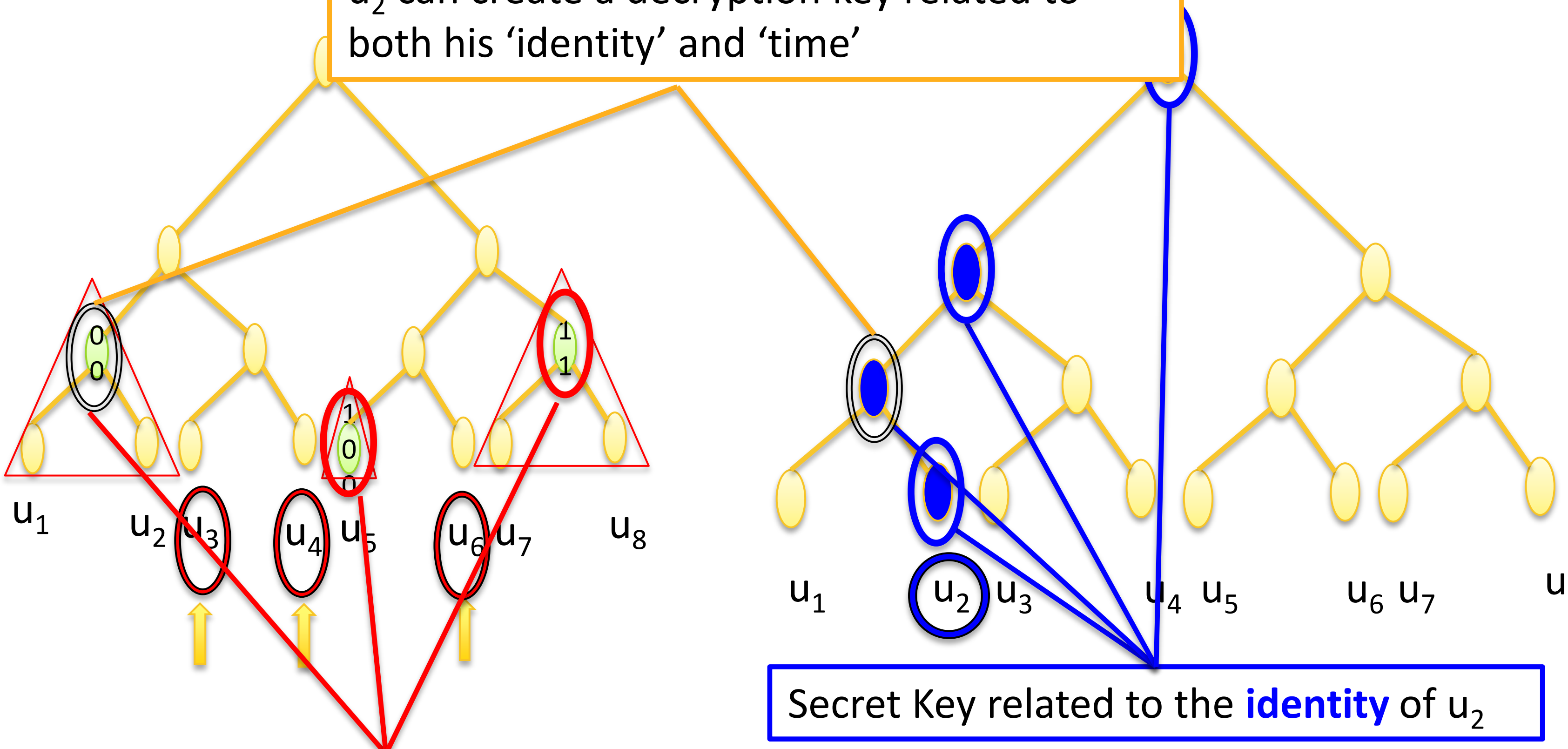
Whenever a user is registered in the system,
the user is assigned in the leaf node of binary tree.

Revocable IBE: Combining BE technique with IBE scheme



Revocable IBE:
Combining BE technique with IBE scheme

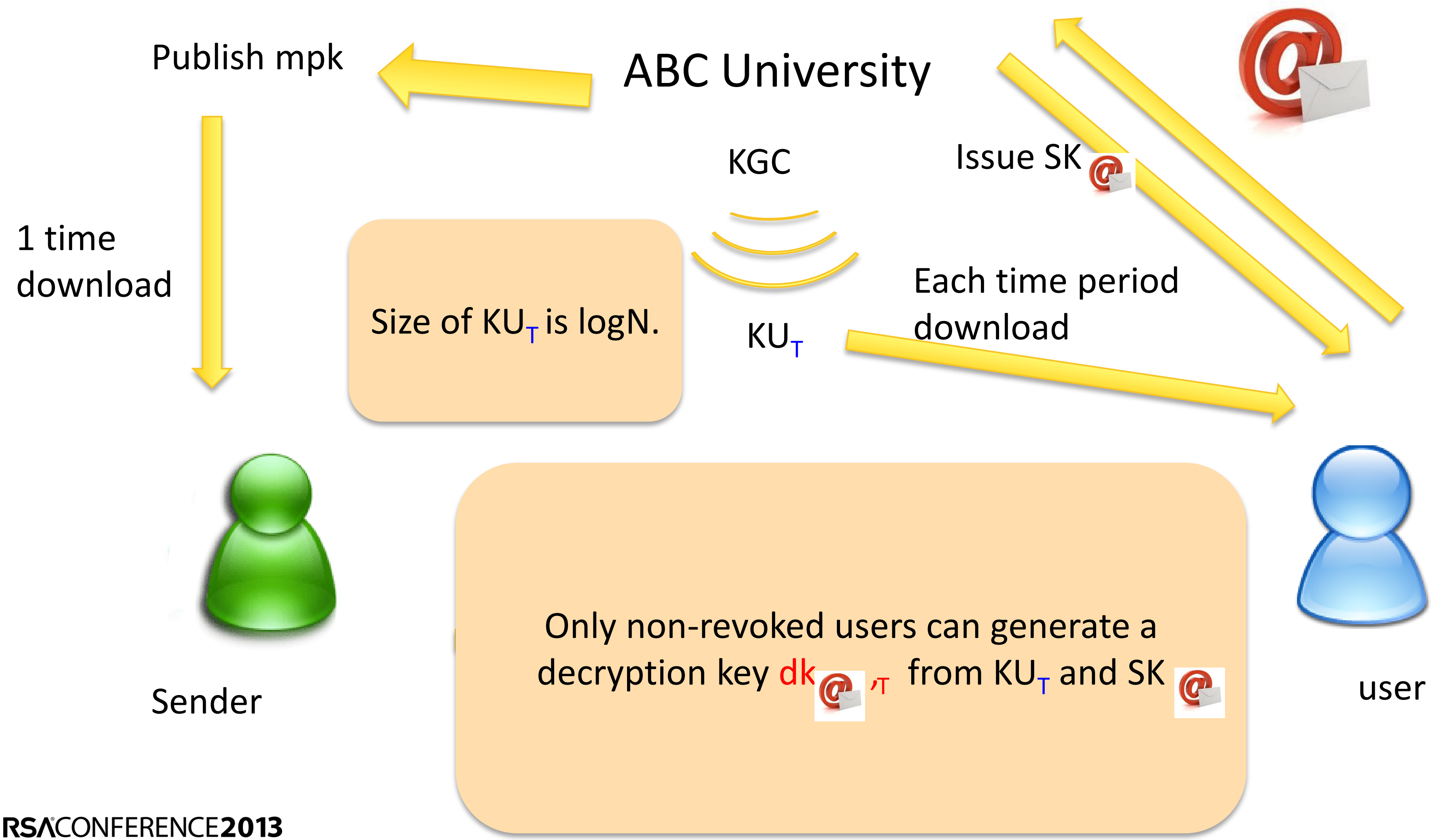
Since u_2 has both related to the same node, u_2 can create a decryption key related to both his 'identity' and 'time'



Secret Key related to the **identity** of u_2

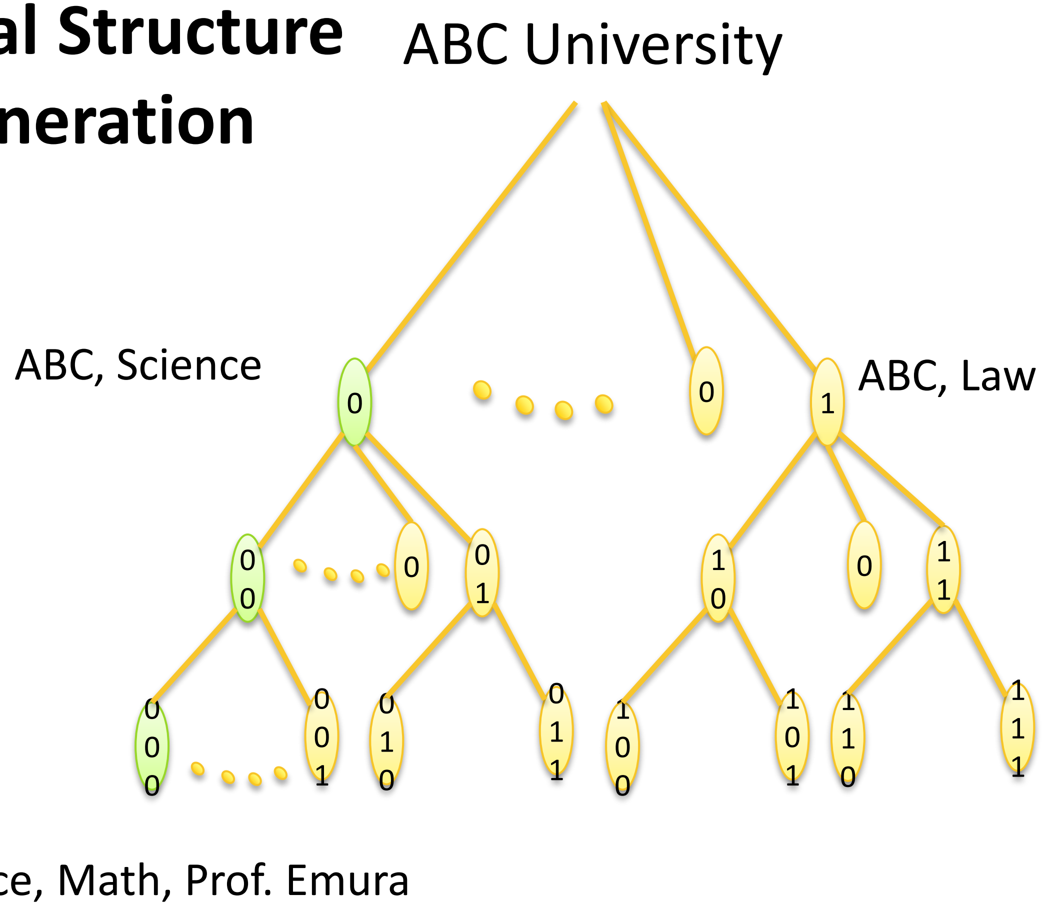
Key Update Information related to **time 'T'**

Revocable IBE: Combining BE technique with IBE scheme

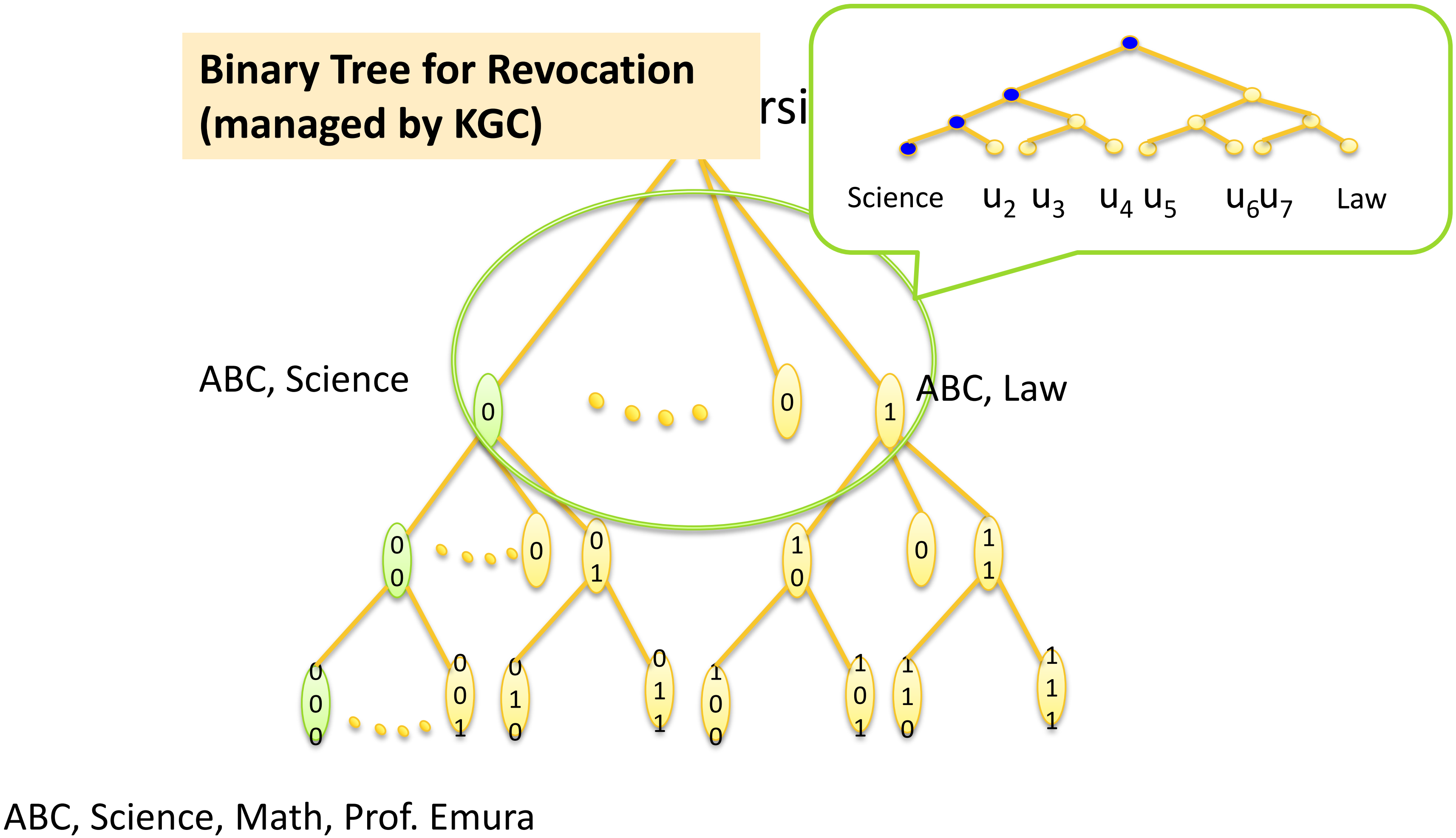


Trivial Approach for Our Goal

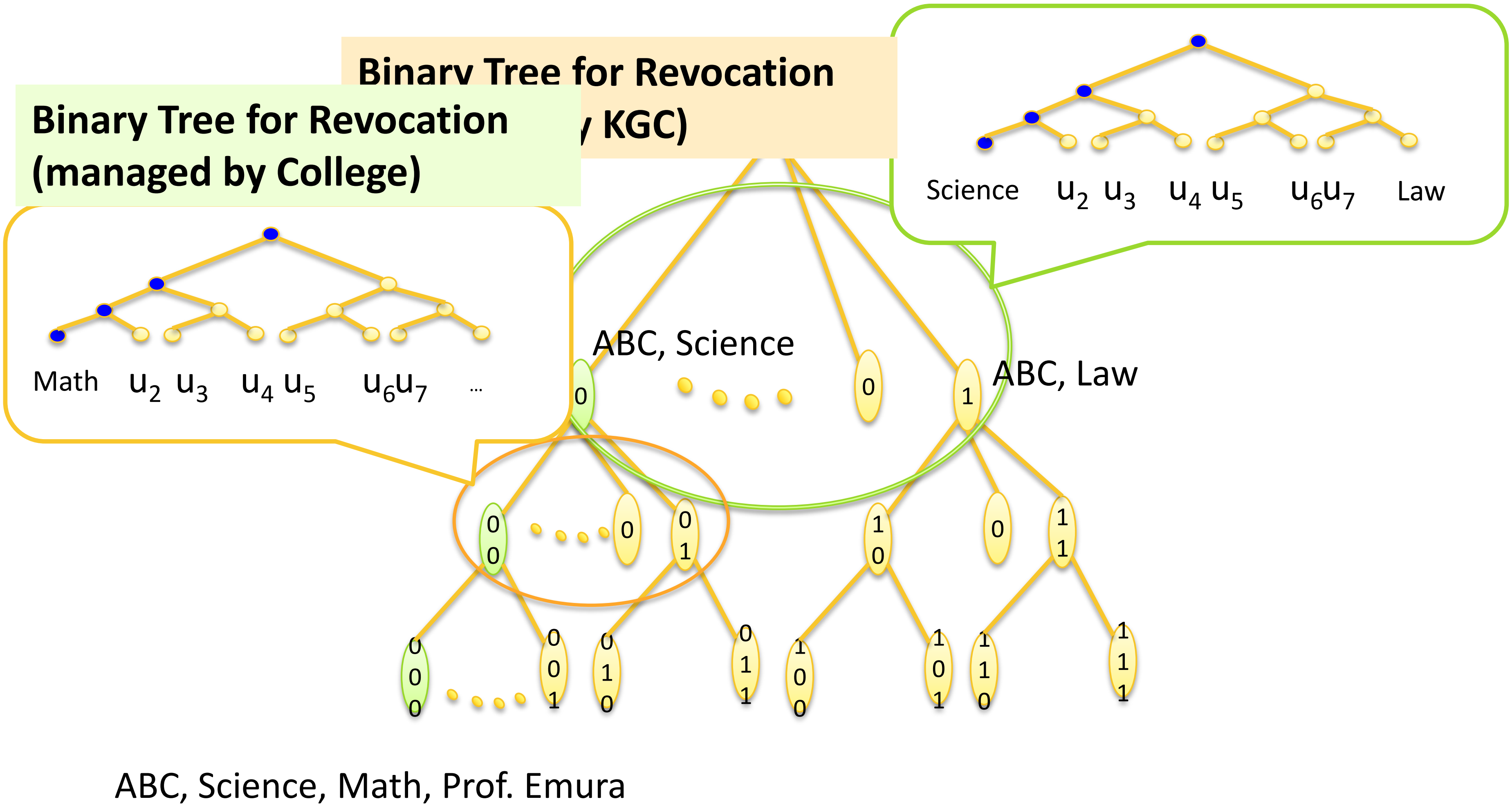
Hierarchical Structure for Key Generation



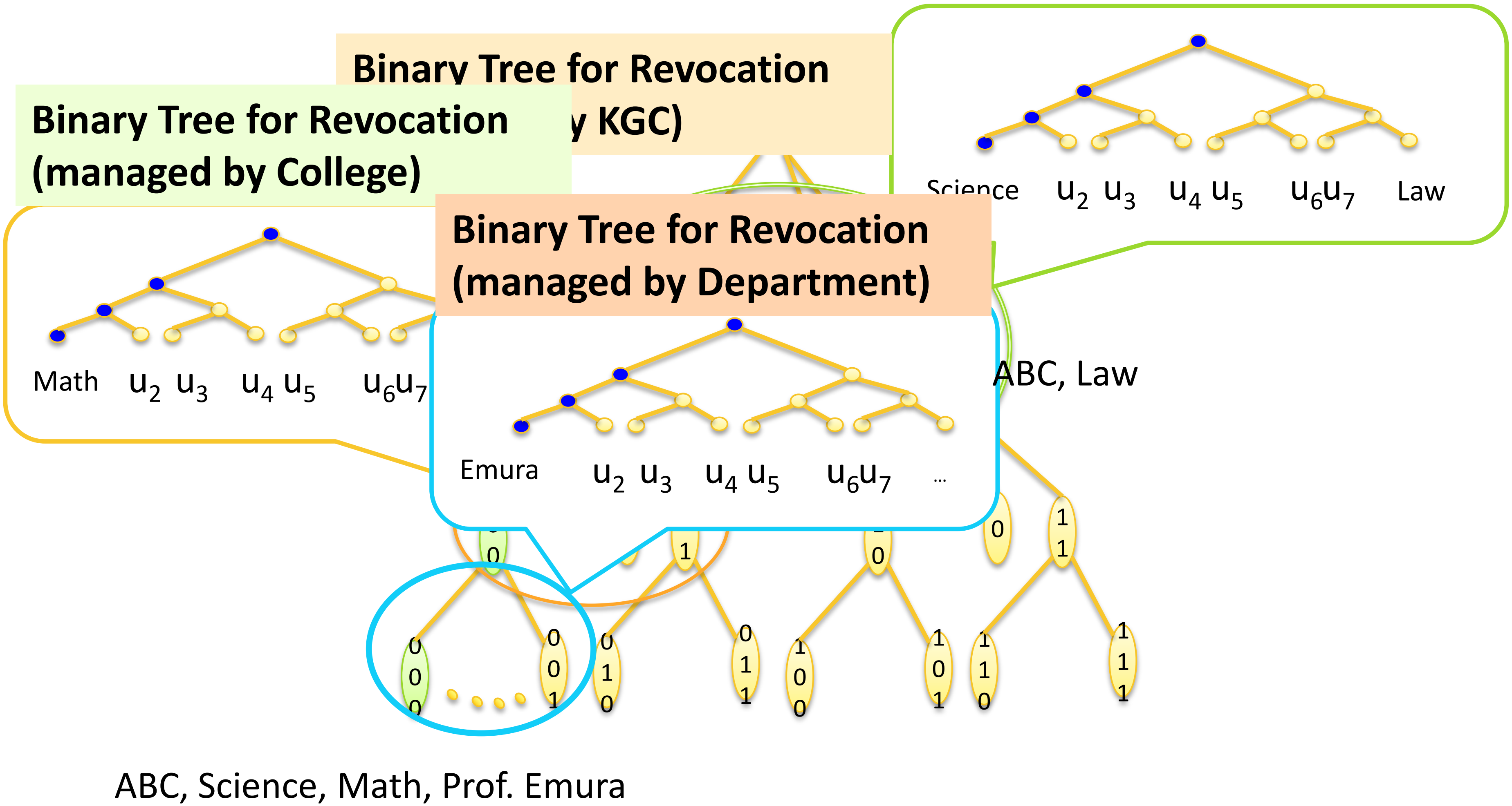
Trivial Approach



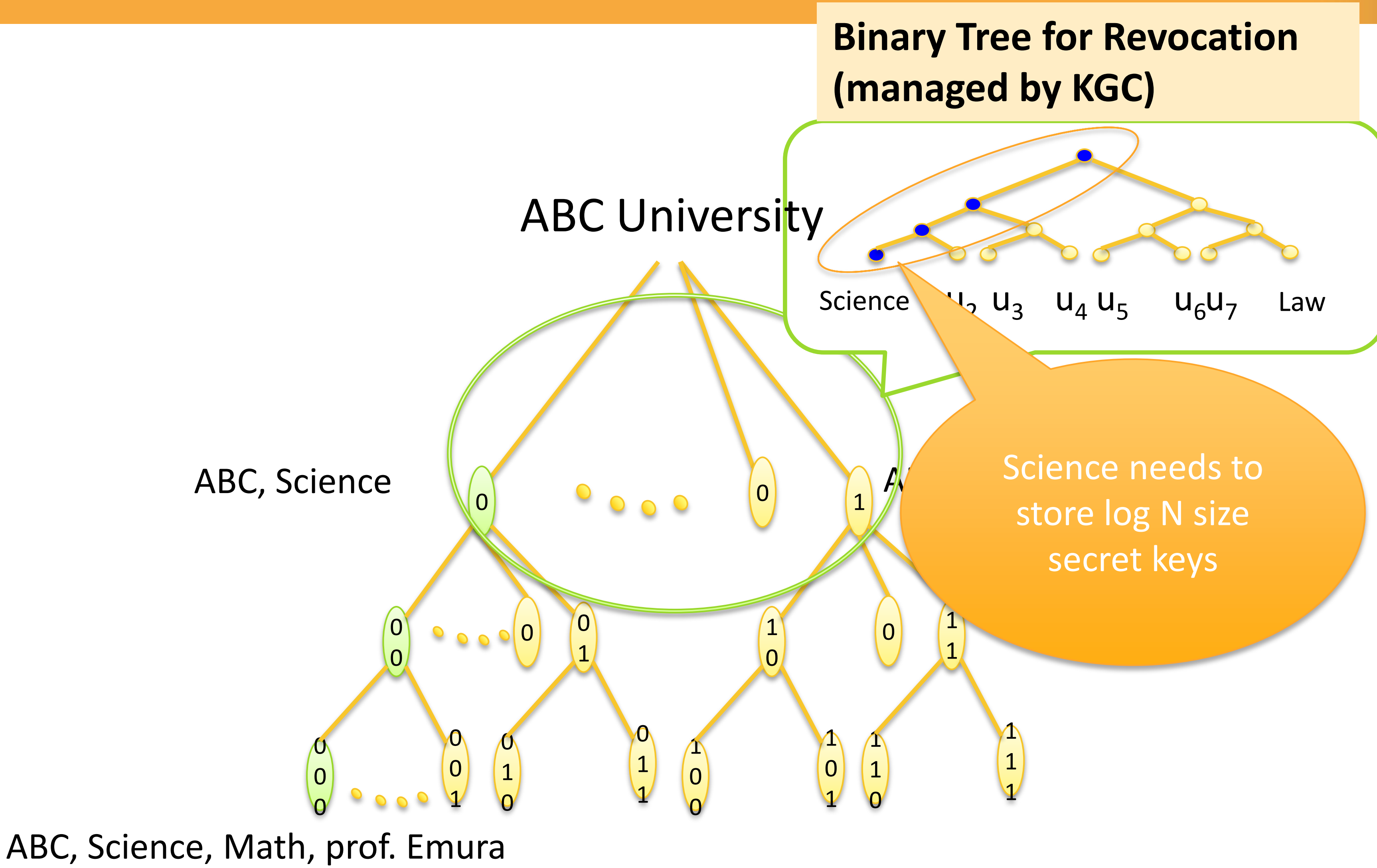
Trivial Approach



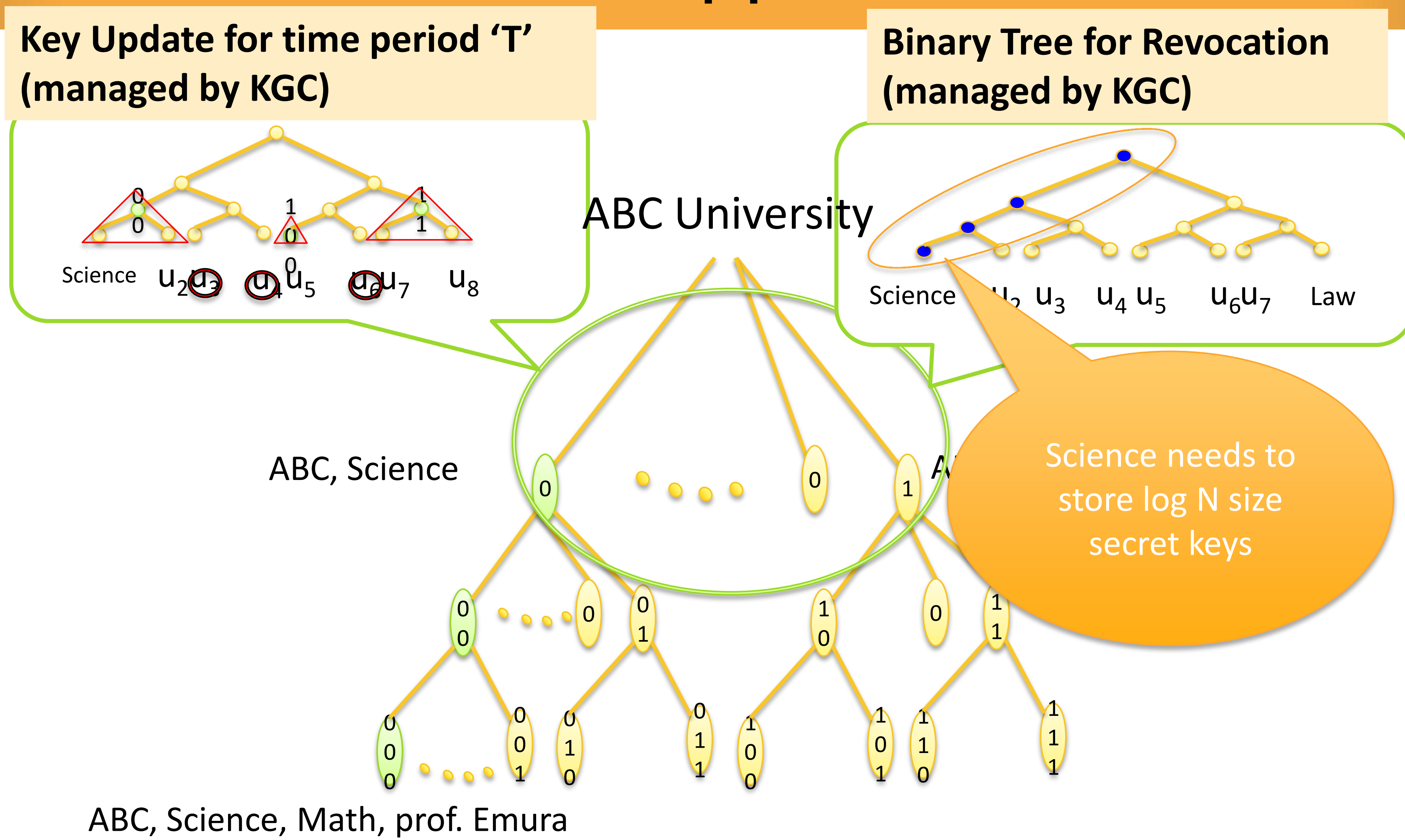
Trivial Approach



Trivial Approach



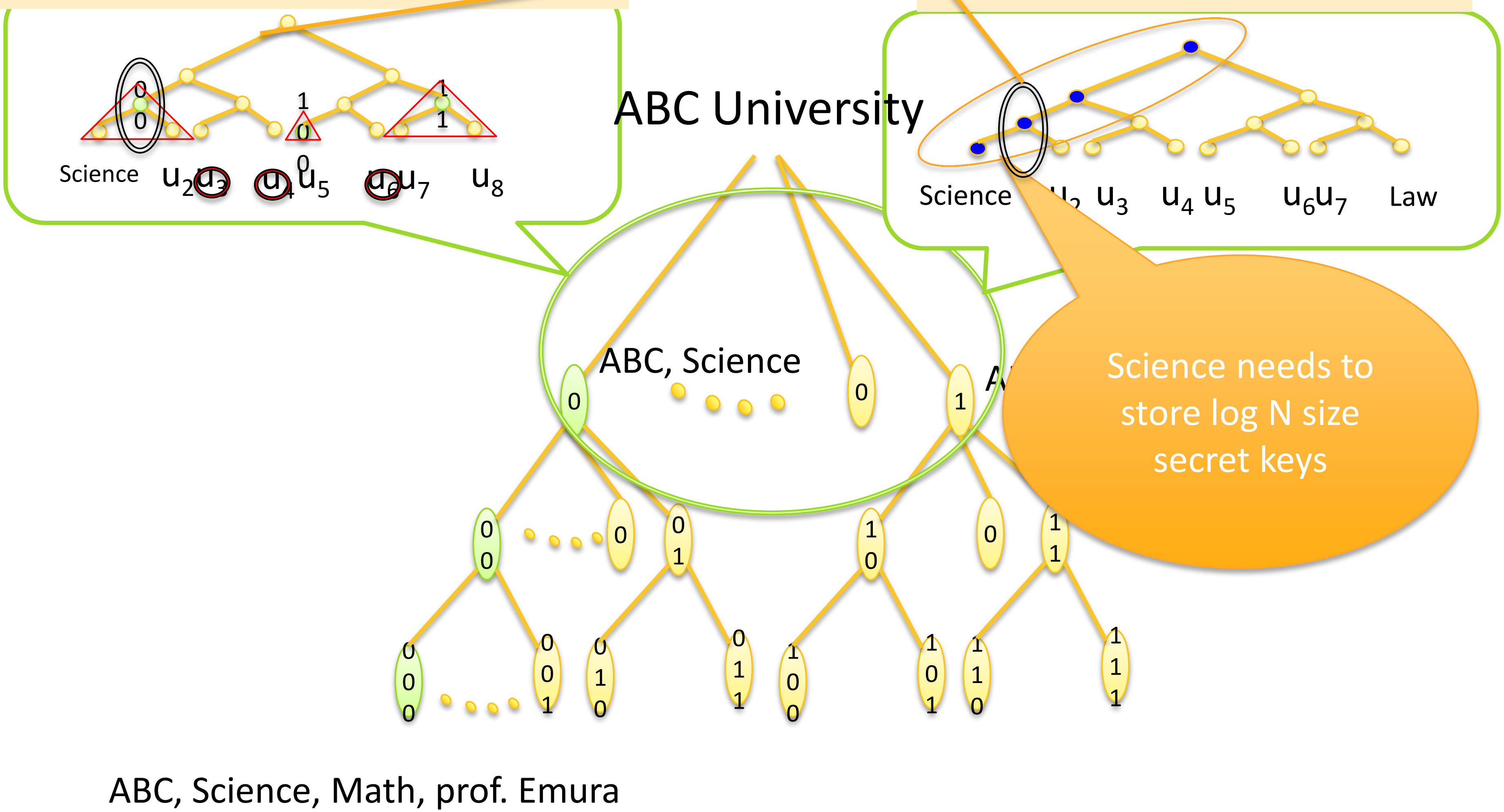
Trivial Approach



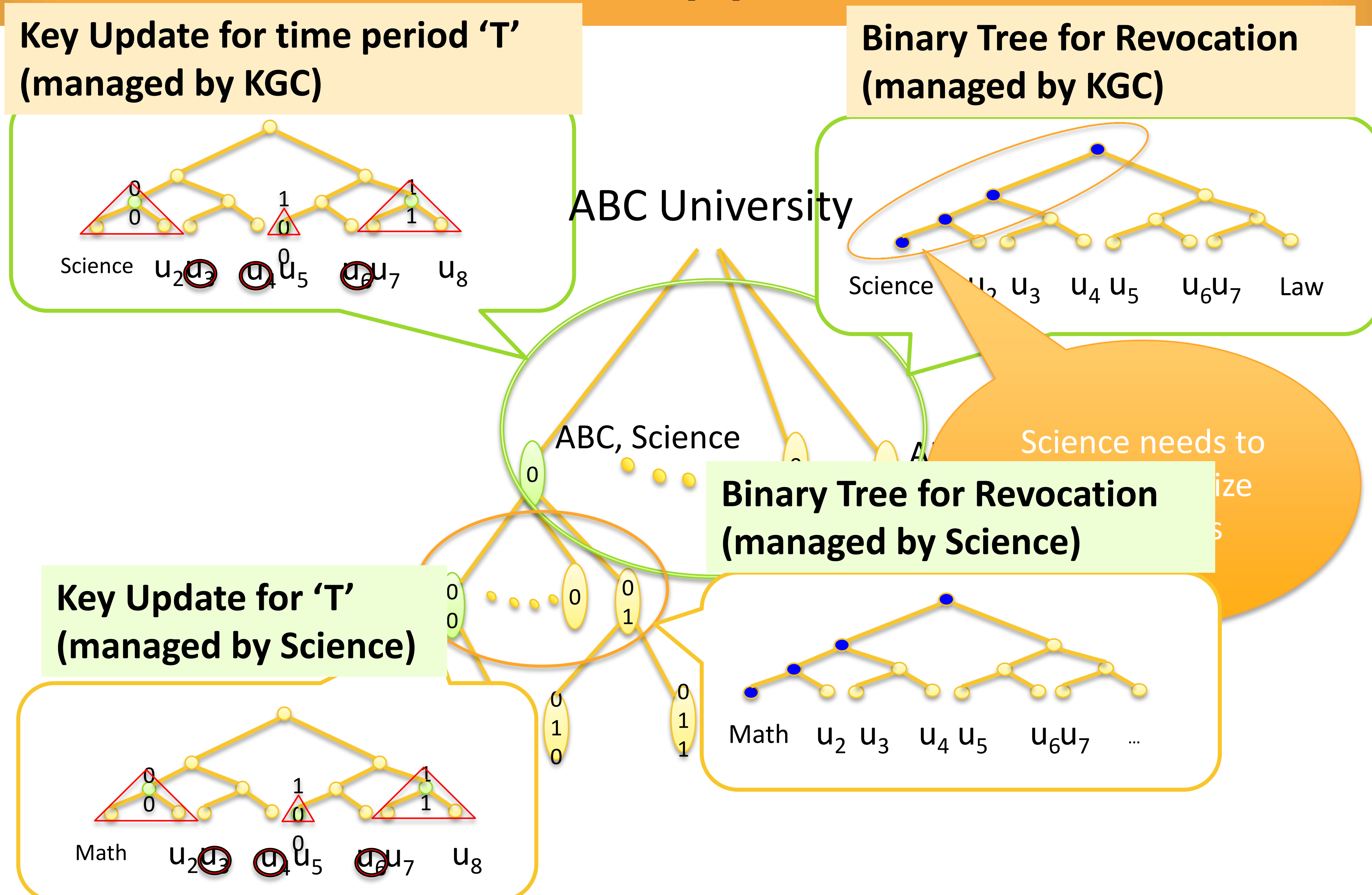
Science is not revoked on time ‘T’.
(1) It can create a decryption key relate to its identity & time ‘T’.
(2) It can generate Key Update for its children.

Key Update for tim
(managed by KGC)

(managed by KGC)

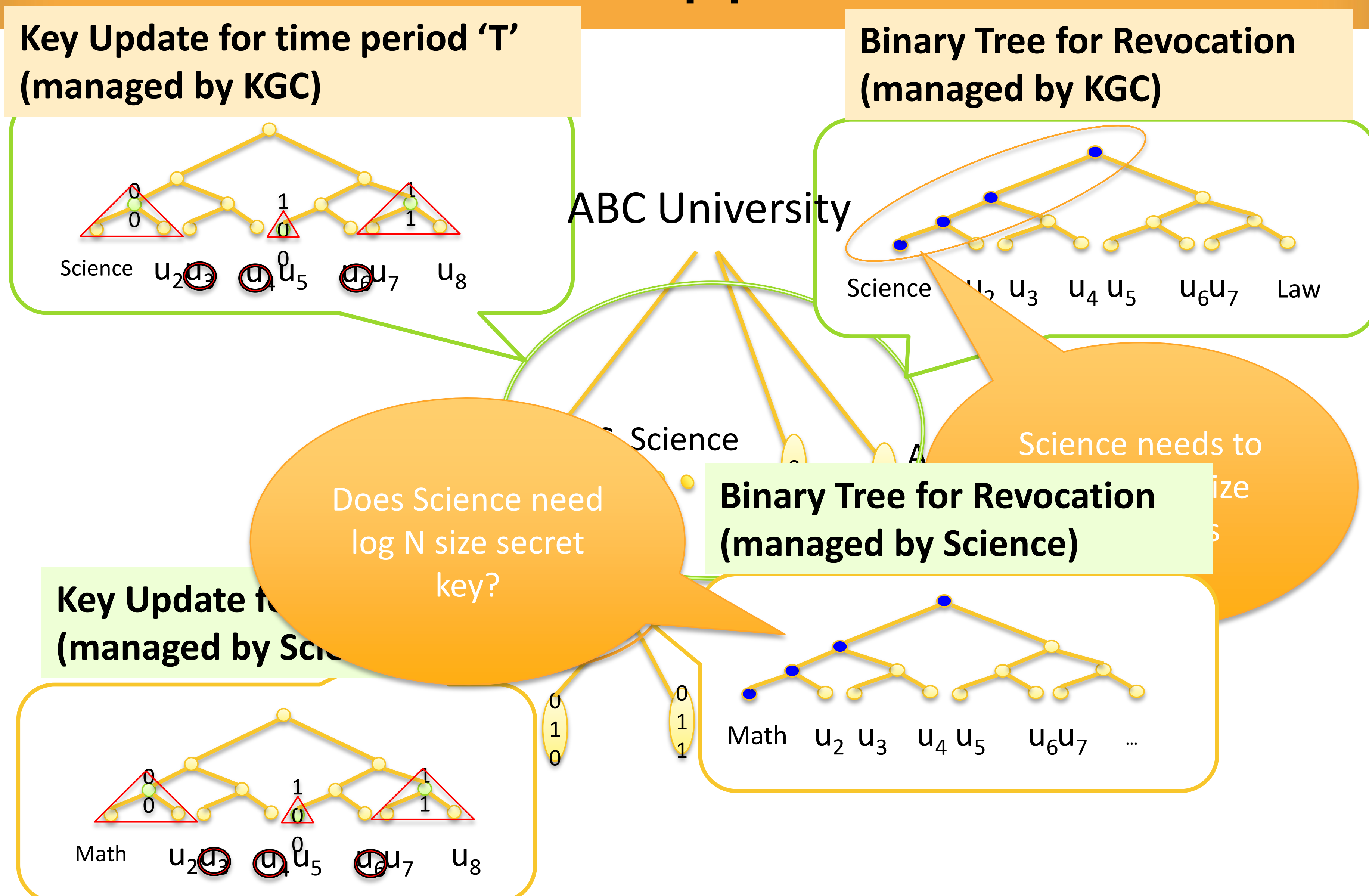


Trivial Approach



RSA CONFERENCE 2013

Trivial Approach



Approach

No!

The parent (Science) has log N size secret key and one subkey is used for each time period.

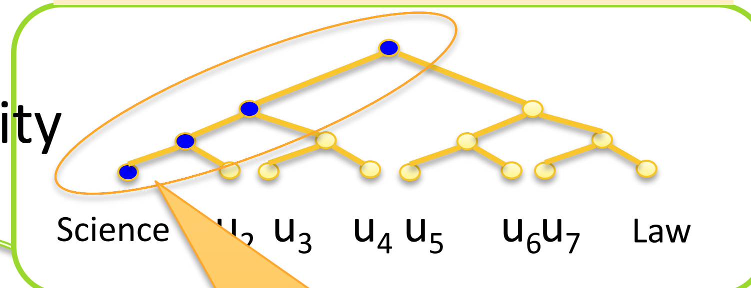
A child (Math) does not know which subkey will be used for each time period.

Therefore, children should have $(\log N)^2$ subkeys.

…

n-th level user has $(\log N)^n$ size secret keys

Binary Tree for Revocation (managed by KGC)



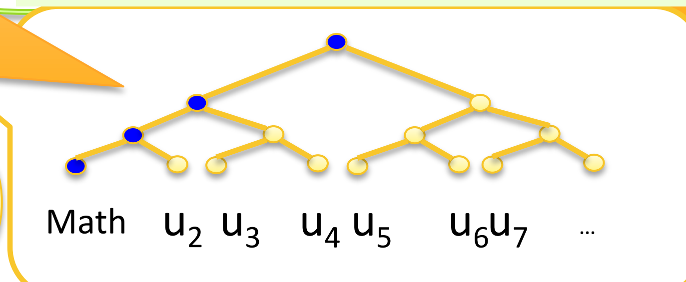
University

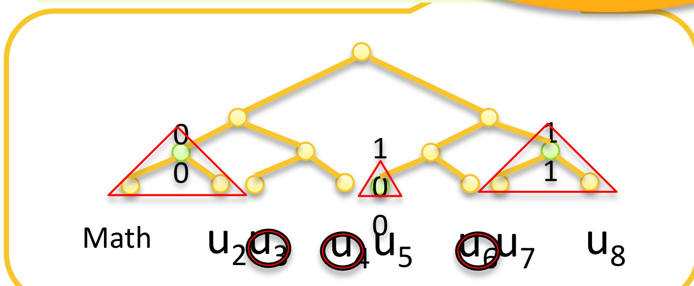
cience

Science needs to

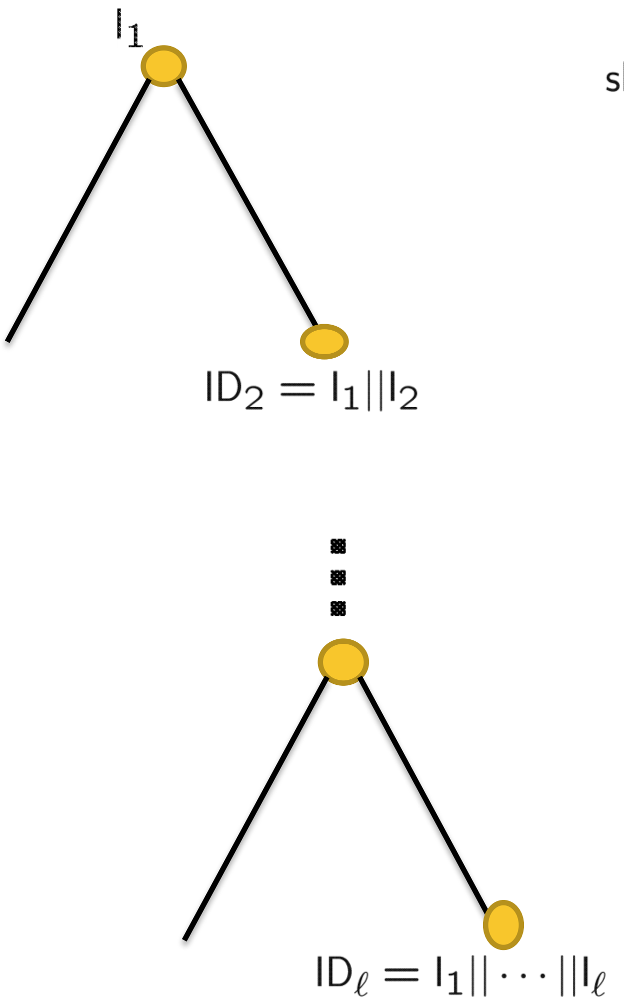
ize

Binary Tree for Revocation (managed by Science)





Trivial Approach

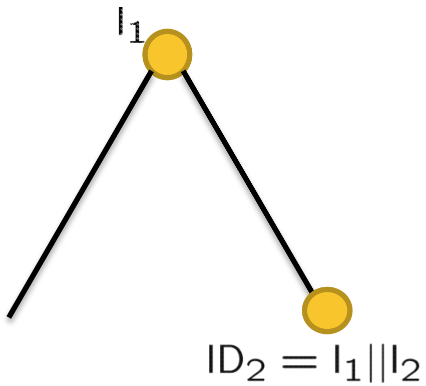


$$\mathsf{sk}_{\mathsf{ID}_{|1}} = \left\{ \begin{array}{l} \mathsf{d}_{\mathsf{I}_1}^{1} \\ \vdots \\ \mathsf{d}_{\mathsf{I}_1}^{\log N} \end{array} \right\} \log N \text{ subkeys}$$

$$\mathsf{sk}_{\mathsf{ID}_{|2}} = \left\{ \begin{array}{l} \mathsf{d}_{\mathsf{I}_1}^{1} \cdot \mathsf{d}_{\mathsf{I}_2}^{1} \\ \mathsf{d}_{\mathsf{I}_1}^{1} \cdot \mathsf{d}_{\mathsf{I}_2}^{2} \\ \vdots \\ \mathsf{d}_{\mathsf{I}_1}^{1} \cdot \mathsf{d}_{\mathsf{I}_2}^{\log N} \\ \mathsf{d}_{\mathsf{I}_1}^{2} \cdot \mathsf{d}_{\mathsf{I}_2}^{1} \\ \vdots \\ \mathsf{d}_{\mathsf{I}_1}^{\log N} \cdot \mathsf{d}_{\mathsf{I}_2}^{\log N} \end{array} \right\} (\log N)^2 \text{ subkeys}$$

$$\mathsf{sk}_{\mathsf{ID}_{|\ell}} = \left\{ \vdots \right\} \quad (\log N)^\ell \text{ subkeys}$$

Our Approach – Asymmetric Trade



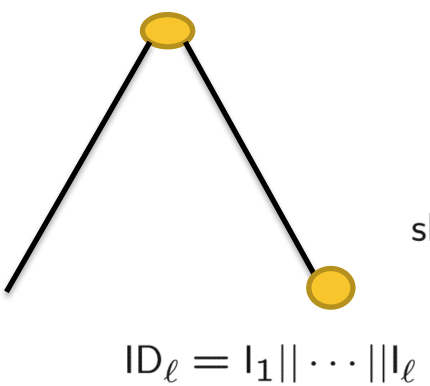
$$sk_{ID_{|1}} = \left\{ \begin{array}{c} d_{I_1}^{1} \\ \vdots \\ d_{I_1}^{\log N} \end{array} \right\}$$

$\log N$ subkeys

$$sk_{ID_{|2}} = \left[\begin{array}{c} \left\{ \begin{array}{c} d_{I_1}^{1} \\ d_{I_1}^{2} \\ \vdots \\ d_{I_1}^{\log N} \end{array} \right\} \\ \\ \left\{ \begin{array}{c} d_{I_2}^{1} \\ d_{I_2}^{2} \\ \vdots \\ d_{I_2}^{\log N} \end{array} \right\} \end{array} \right] \rightarrow \text{Product} \Rightarrow (\log N)^2 \text{ subkeys}$$

$2 \log N$ subkeys

$\vdots$



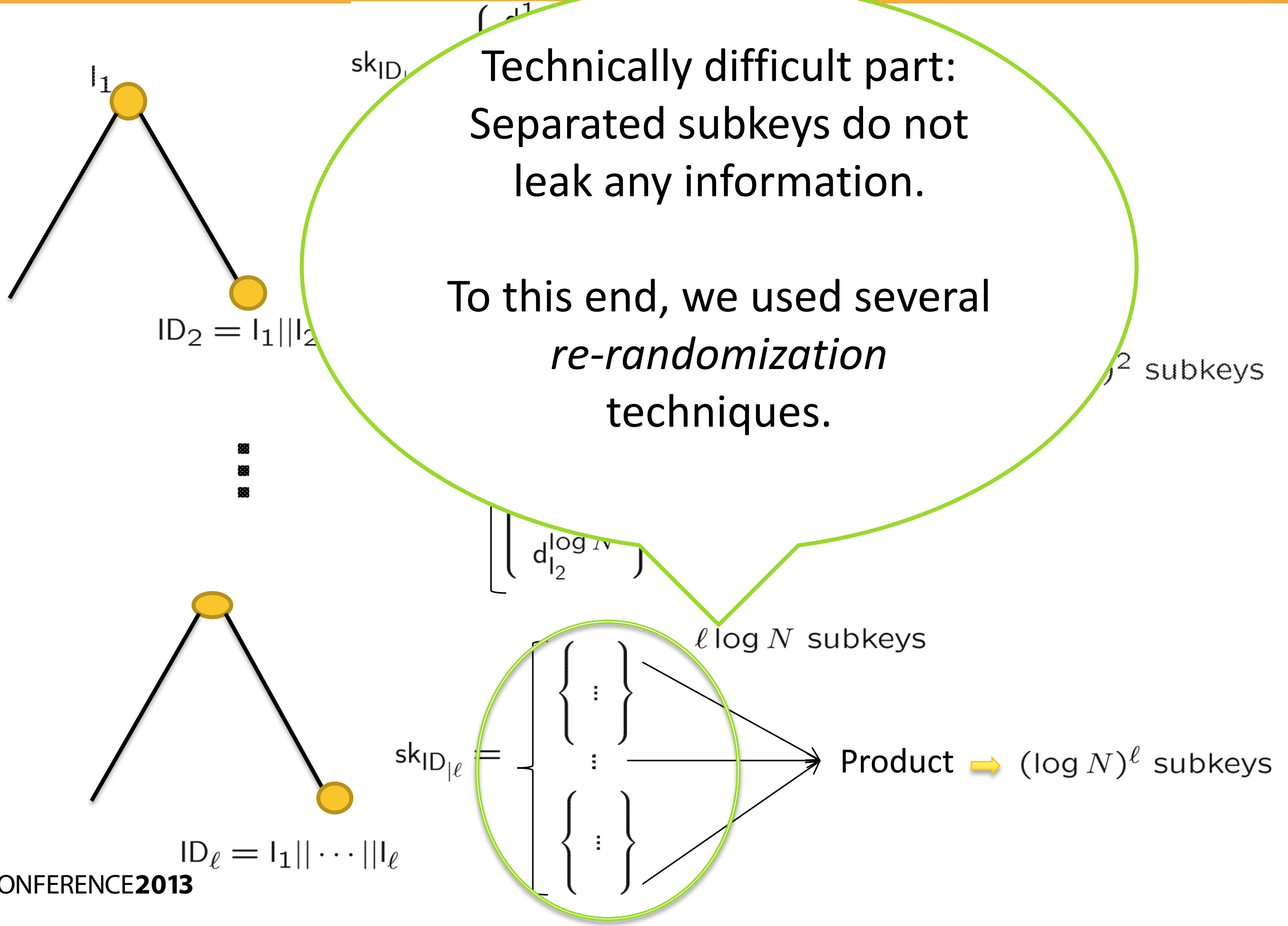
$\ell \log N$ subkeys

$$sk_{ID_{|\ell}} = \left[\begin{array}{c} \left\{ \vdots \right\} \\ \vdots \\ \left\{ \vdots \right\} \end{array} \right] \rightarrow \text{Product} \Rightarrow (\log N)^\ell \text{ subkeys}$$

Our Approach – Asymmetric Trade

Technically difficult part:
Separated subkeys do not leak any information.

To this end, we used several *re-randomization* techniques.



Our Result

- We propose the first practical RHIBE scheme
  - Our scheme is based on Boneh-Boyen HIBE scheme
  - The size of secret key is $O(l^2 \log N)$, where l is user's level.
  - We proved that the proposed scheme satisfies a weaker security notion such as *selective* security notion.

Further Study

- Fully secure RHIBE
- Different revocation method, such as *Subset Difference*
- Revocation methodology in functional encryption

- Thanks!

Security in knowledge