2016 IEEE 2nd International Conference on Collaboration and Internet Computing

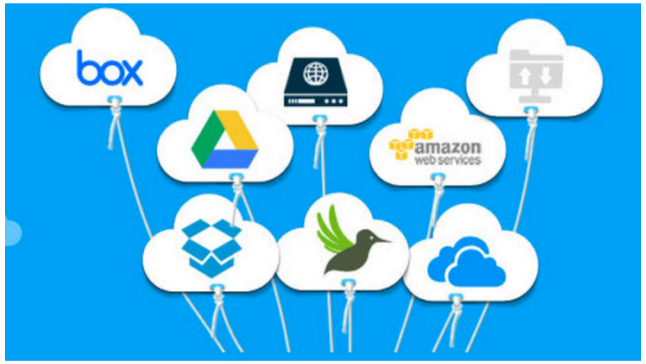
UNIVERSITY OF PITTSBURGH
School of
Information Sciences

Enabling Attribute Based Encryption as an Internet Service

Runhua Xu and James Joshi

University of Pittsburgh

runhua.xu@pitt.edu

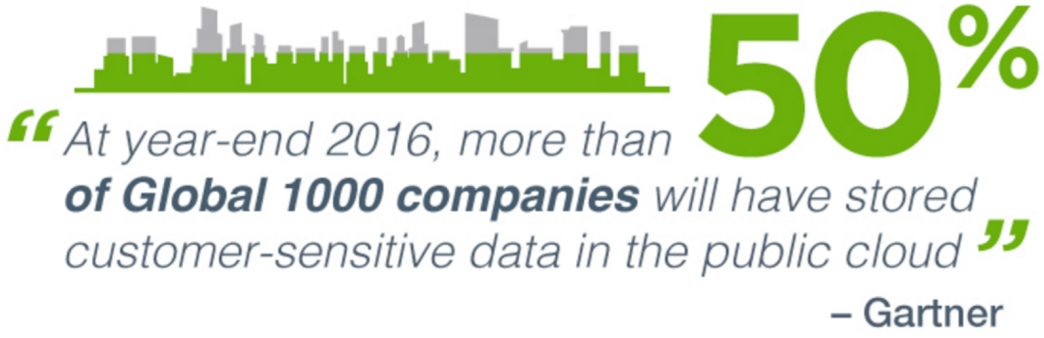


Cloud Storage Service

- It has been gaining significant success
  - *potential “infinite” storage size*
  - *convenience of synchronization*
  - *ease of access (at anytime, from anywhere)*
- Users/Organizations
  - *increasingly utilize/rely on the cloud storage services*

Security & Privacy Concerns

Recent advances have enabled applications that generate/collect *huge amounts of* *personal data*.



“At year-end 2016, more than 50% of **Global 1000 companies** will have stored customer-sensitive data in the public cloud”

– Gartner

Source: http://www.gartner.com/newsroom/id/1862714

Cloud Storage Providers

Honest-but-Curious

-- *run the programs and algorithms correctly*

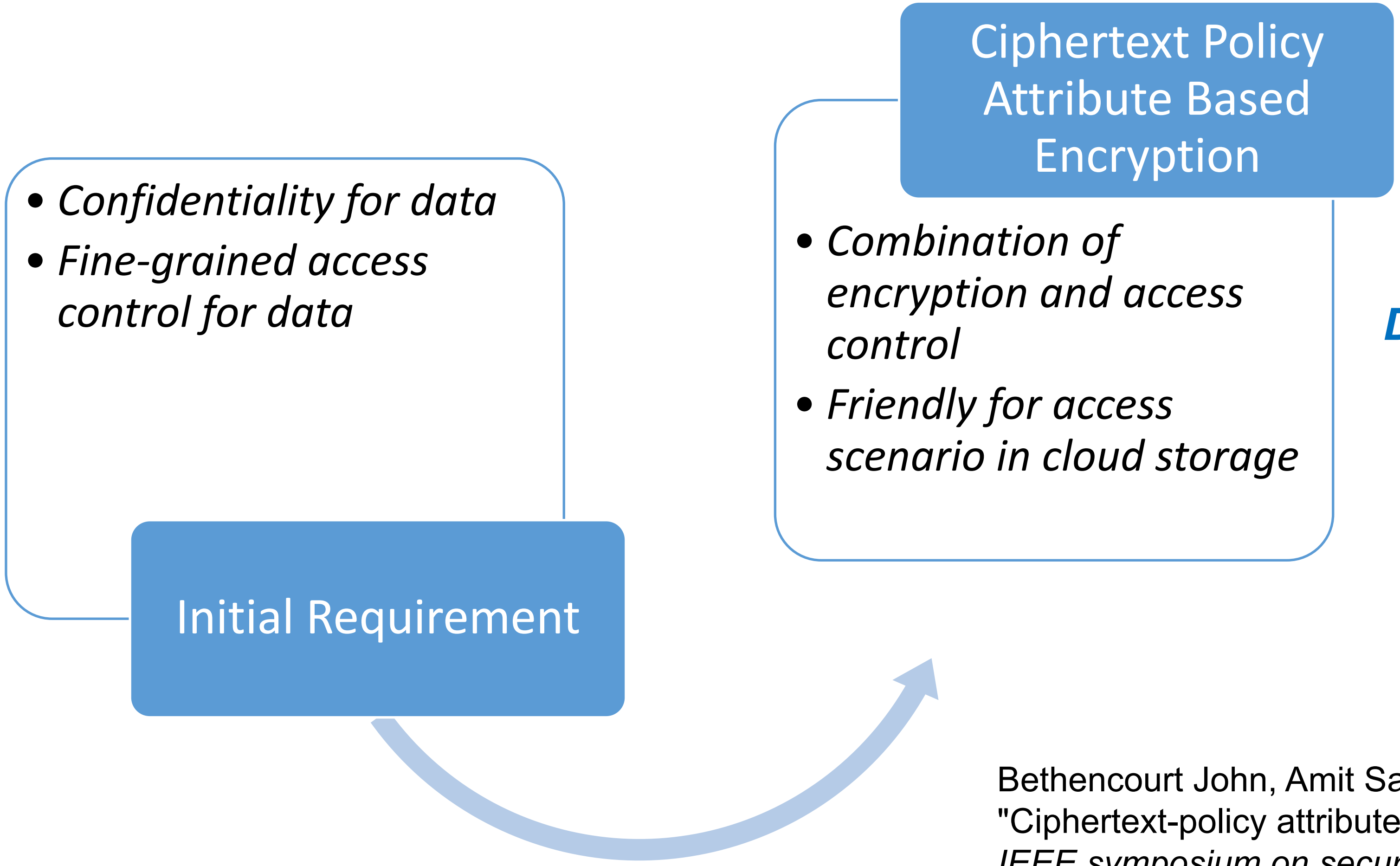
but gather information related to the stored data.

Insider threat

-- *secretly analyzing or leaking customers’ sensitive data*

How users are able to fully trust the CSP regards to their sensitive data

Initial Solution



Initial Requirement

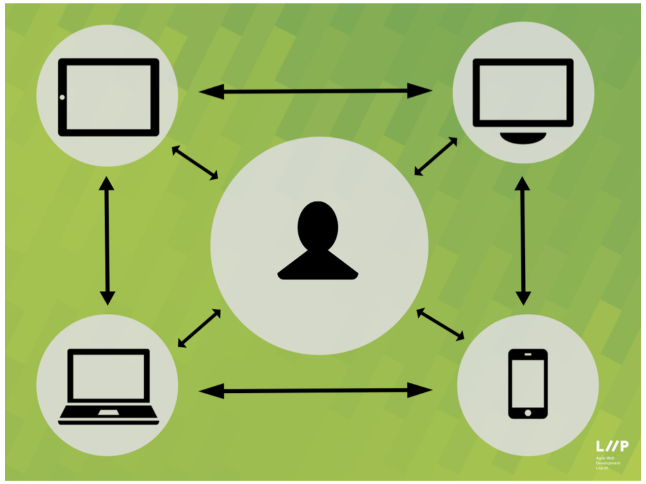
- *Confidentiality for data*
- *Fine-grained access control for data*

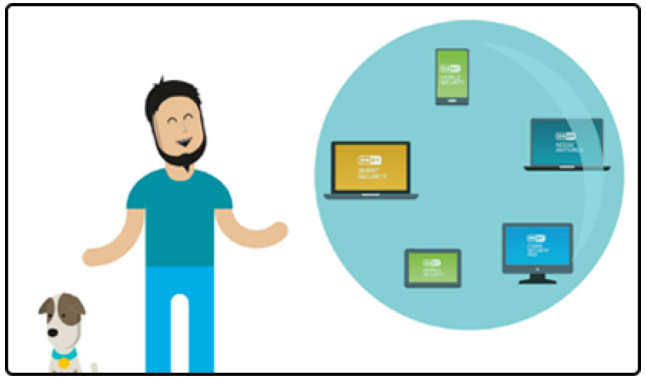
Ciphertext Policy Attribute Based Encryption

- *Combination of encryption and access control*
- *Friendly for access scenario in cloud storage*

Data : self-protection

Bethencourt John, Amit Sahai, and Brent Waters. "Ciphertext-policy attribute-based encryption." *2007 IEEE symposium on security and privacy (S&P'07)*. IEEE, 2007.





Multiple-device Scenarios

- Increasing popularity and adoption of mobile devices
  - *pads*
  - *cell phones*
  - *IoT sensors*
- Traditional application

→***Multiple-device application***

*When **ABE schemes** meet **Multiple-device application**,*
what's the situation?

Desktop, laptop, workstation… → fine
Cell phone, pad, IoT sensor…. → not good as expected

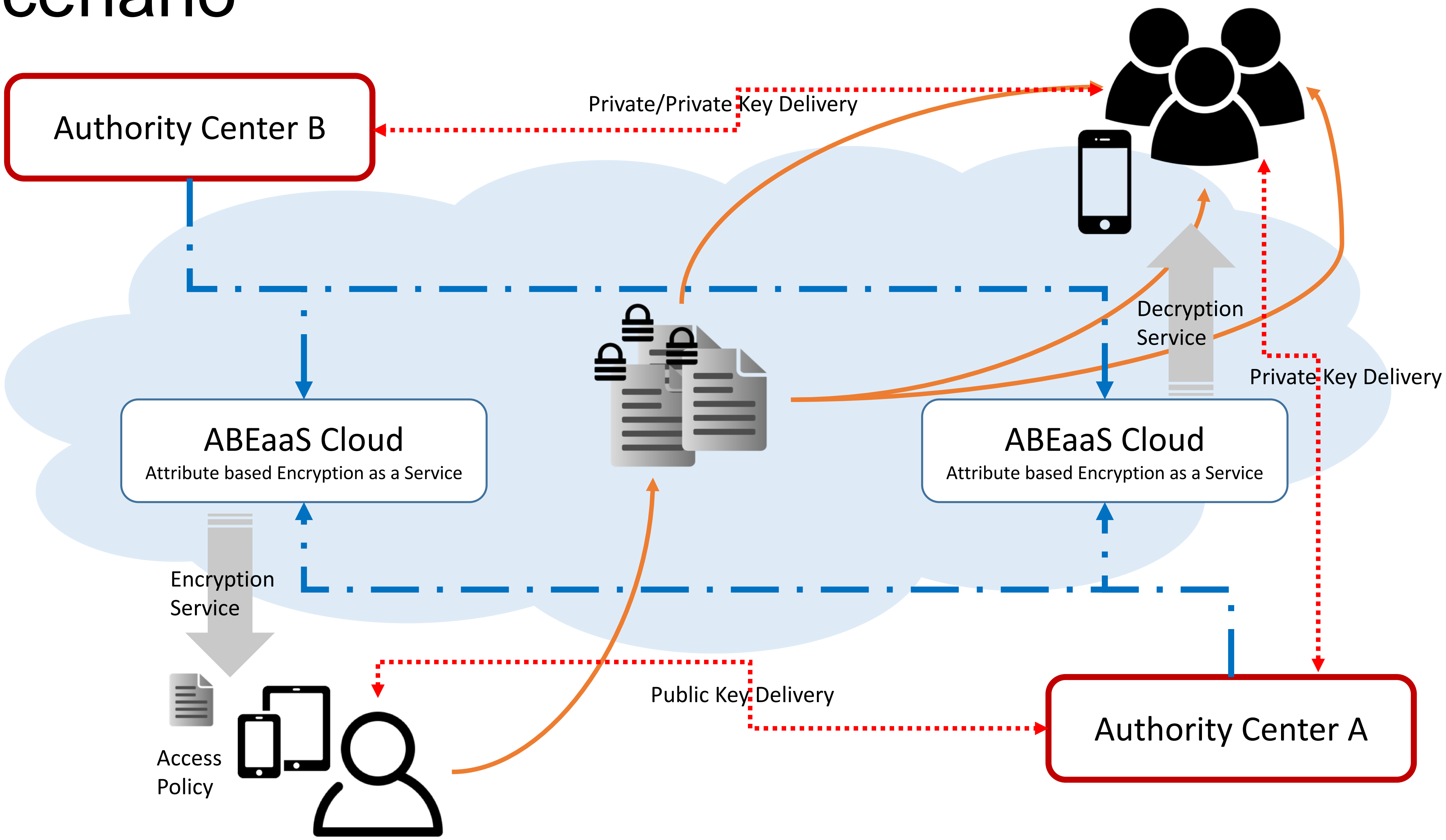
Challenges of ABE Adoption

- Global authority center
  - *hard to deploy a global authority center trusted by all Internet users*
- Multi-device scenarios are pervasive
  - *Put ABE adoption into Multi-device scenarios → limitations*
    - Computational resources for ABE
    - Battery power for ABE

States

- The lack of an effective deployment approach
    - *to make ABE available broadly as a service*
    - *to support a broad set of mobile cloud applications*
- An attribute based encryption as a service
    - *mechanism to deploy ABE widely over various cloud platforms*

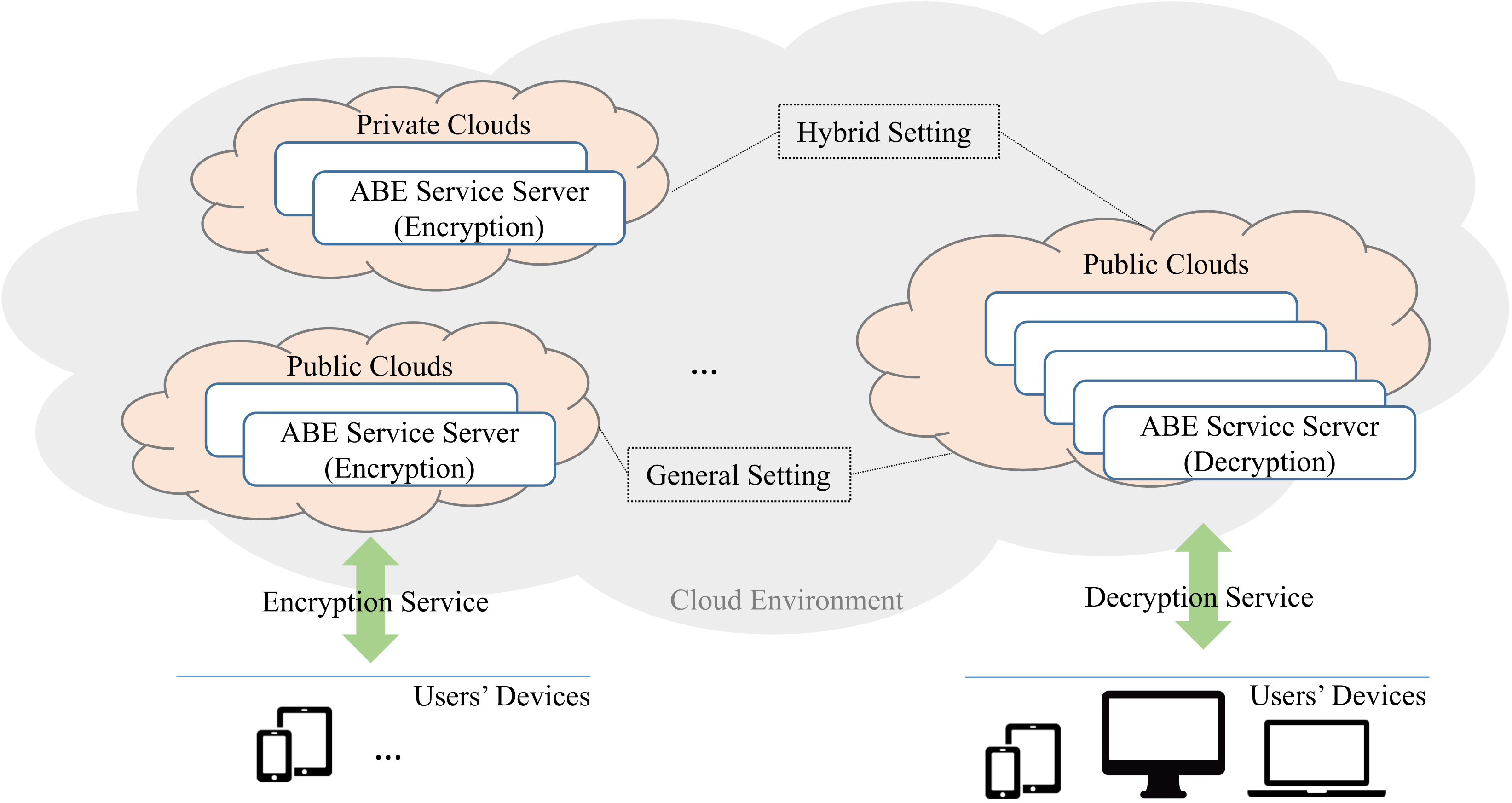
Scenario


Authority Center B
Private/Private Key Delivery
ABEaaS Cloud
Attribute based Encryption as a Service
Decryption Service
Private Key Delivery
ABEaaS Cloud
Attribute based Encryption as a Service
Encryption Service
Access Policy
Public Key Delivery
Authority Center A

Overview

Two setting:

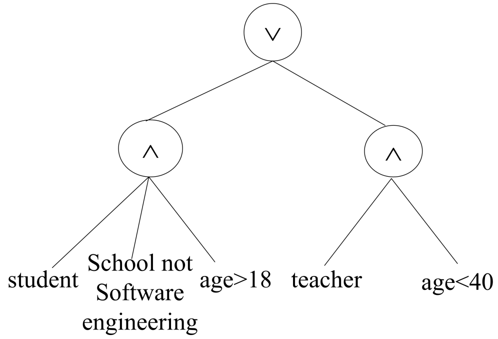
hybrid setting / general setting



From ABE to ABEaaS

- Overview of ABE
  - *Four Algorithms*
    - Setup
    - Key Generation
    - Encryption
    - Decryption
  - *CP-ABE/KP-ABE*
  - *Access Structure*
    - And-gate, Tree, LSSS
  - *Technique to Outsource Computation*
    - Outsource partial computation to a powerful server without impact on the functionality and security of the ABE scheme

$A_1 = (1 \wedge 2 \wedge 3 \wedge 4)$

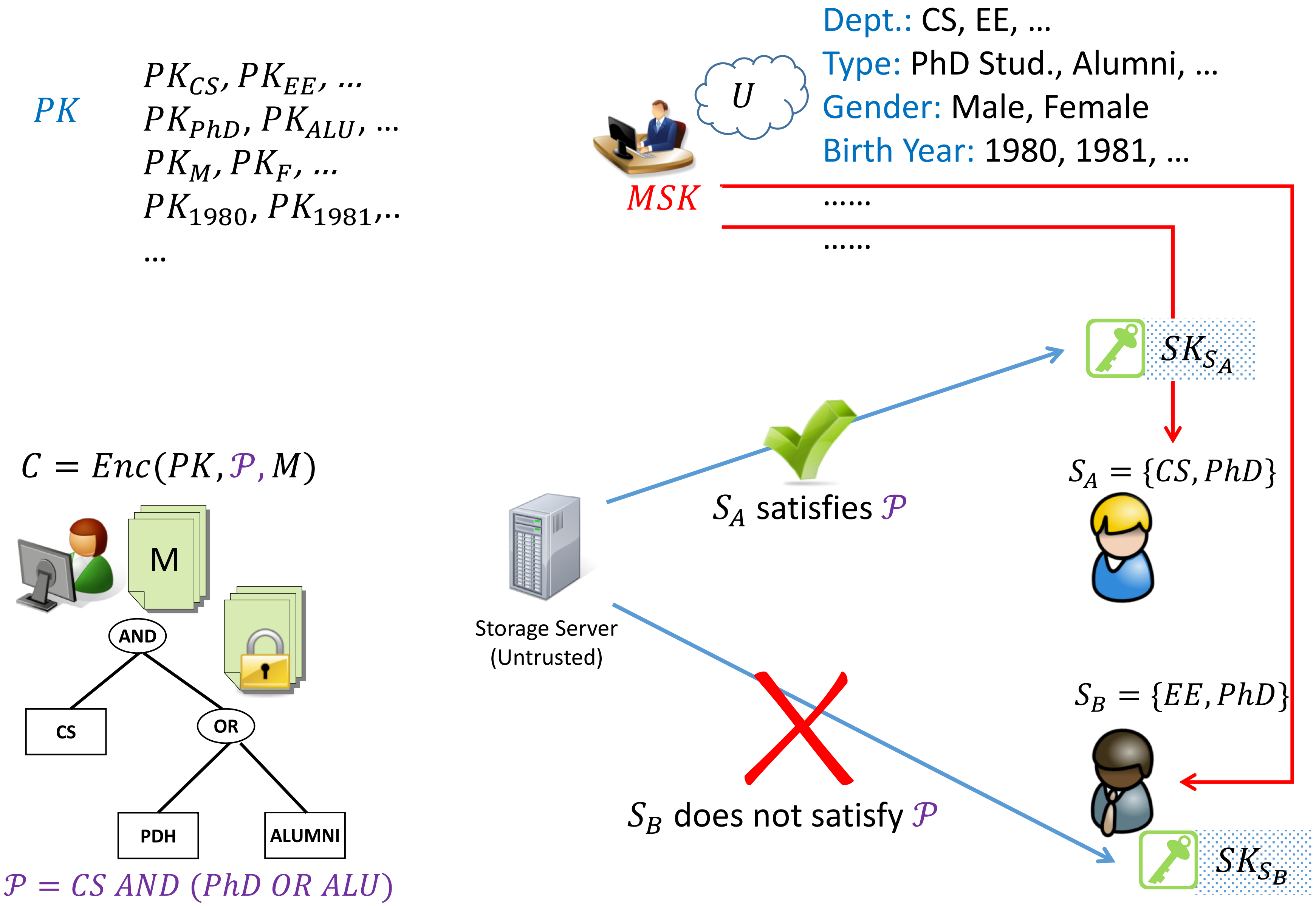


$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \end{pmatrix}$$

$A_3 = (1 \wedge 2 \wedge 3) \vee (1 \wedge 4)$

Preliminaries: What's CP-ABE

CP-ABE in detail



PK

$PK_{CS}, PK_{EE}, \ldots$
$PK_{PhD}, PK_{ALU}, \ldots$
$PK_{M}, PK_{F}, \ldots$
$PK_{1980}, PK_{1981},..$
…

U

Dept.: CS, EE, …
Type: PhD Stud., Alumni, …
Gender: Male, Female
Birth Year: 1980, 1981, …
……
……

MSK

SK_{S_A}

$S_A = \{CS, PhD\}$

S_A satisfies $\mathcal{P}$

Storage Server
(Untrusted)

S_B does not satisfy $\mathcal{P}$

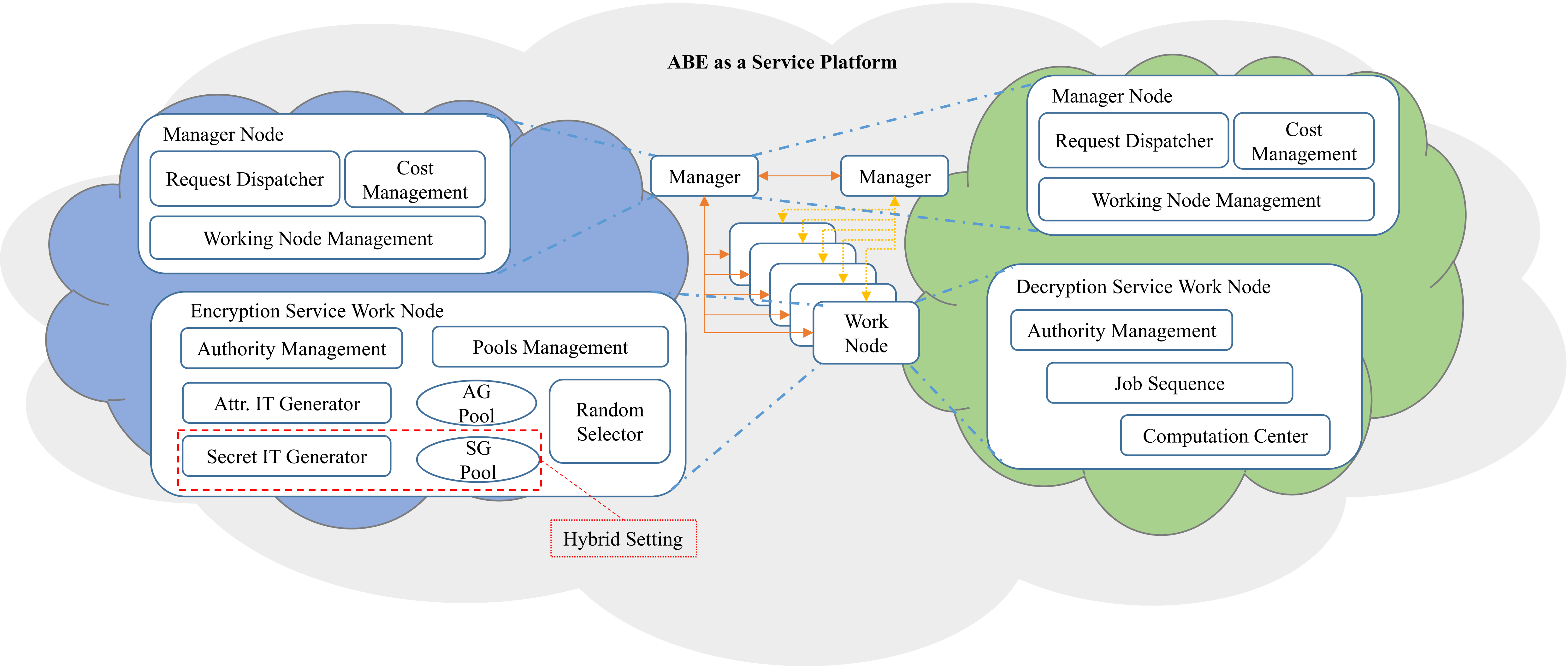
$S_B = \{EE, PhD\}$

SK_{S_B}

$C = Enc(PK, \mathcal{P}, M)$

$\mathcal{P} = CS\ AND\ (PhD\ OR\ ALU)$

Architecture of ABE service platform



Manager Node

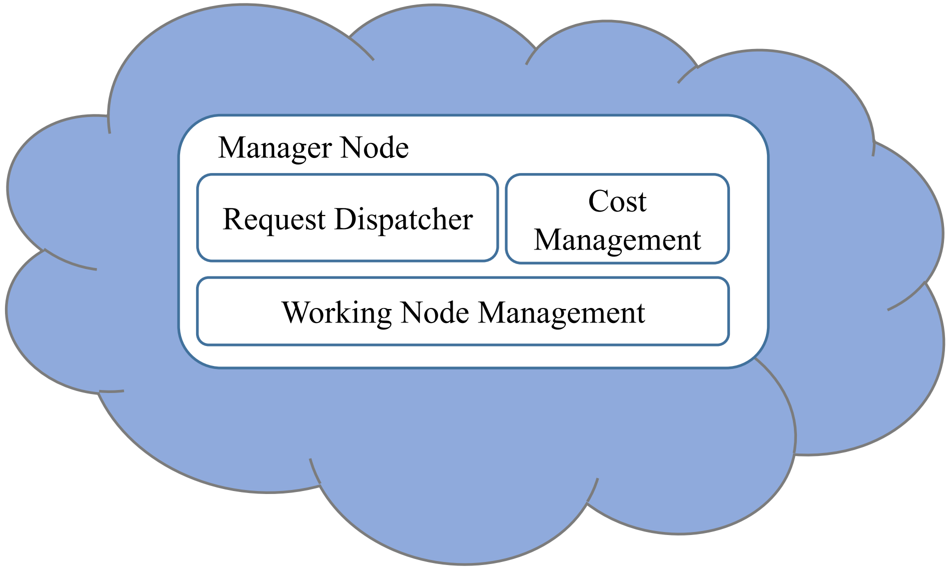
Request Dispatcher (RD)

receive the request

dispatch the request to an available work node

Work Node Management (WNM)

manage a number of work nodes



Encryption Service Work Node

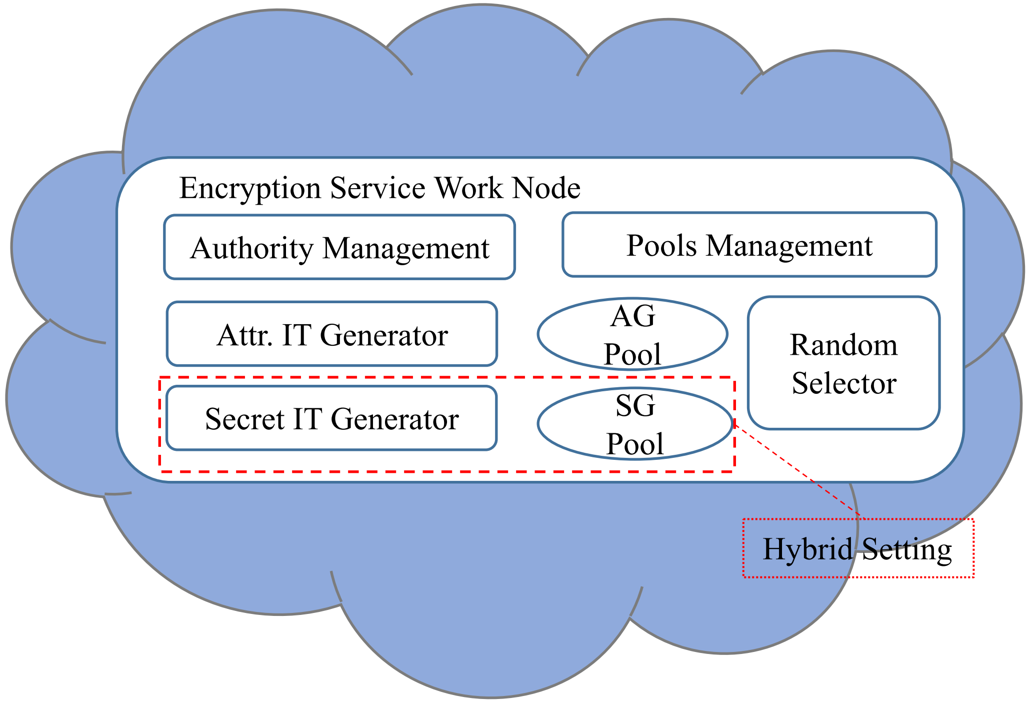
Authority Management (AM)

Secret/Attribute Intermediate Ciphertext Generator

Pools Management (PM)

Secret Intermediate Ciphertext Pool (SICP)

Attribute Intermediate Ciphertext Pool (AICP)

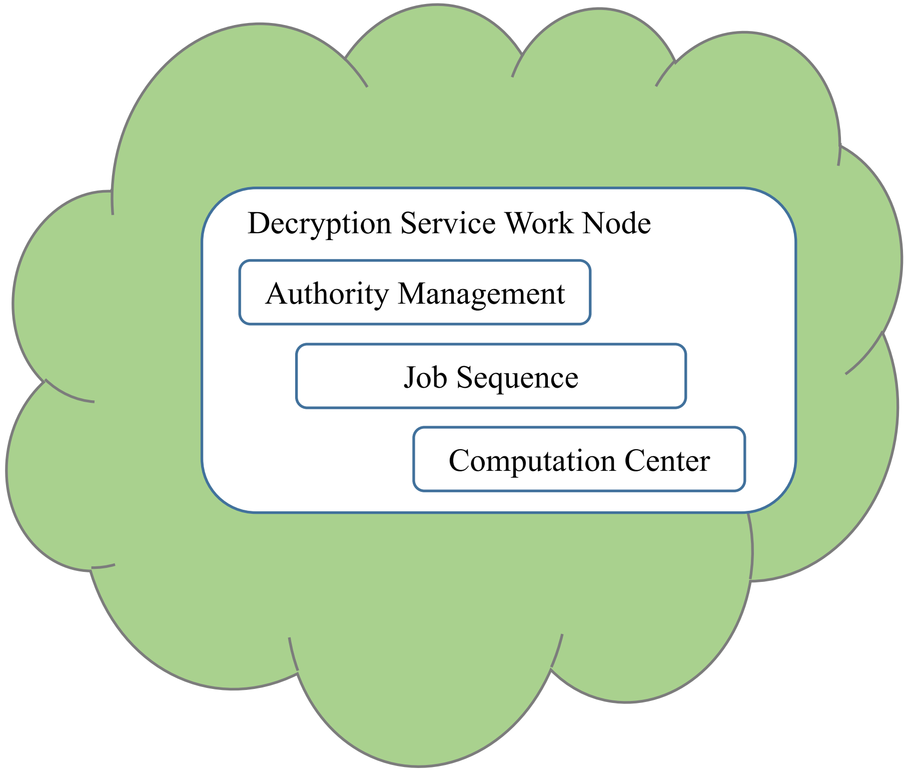


Decryption Service Work Node

The frequency of using decryption service is several times than the frequency of using encryption service.

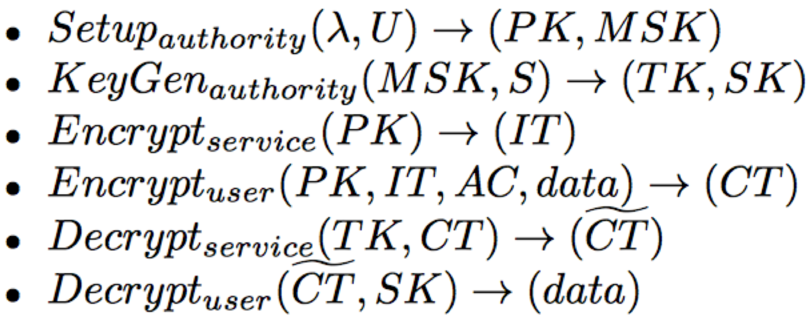
Job Sequence

Computation center will calculate the job in parallel



ABEaaS Implementation

- Prototype model of ABE used in ABEaaS
  - *Extend from [2] & [6]*
  - *ABE Instance*



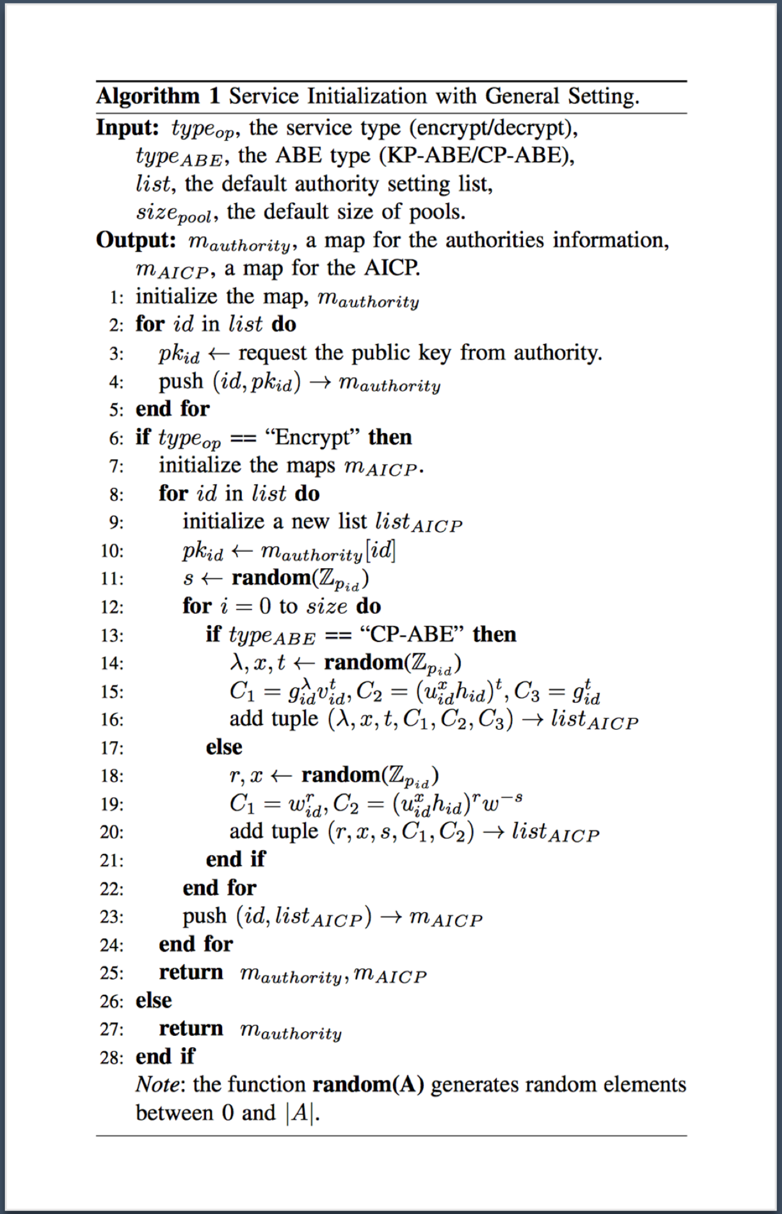
- $Setup_{authority}(\lambda, U) \rightarrow (PK, MSK)$
- $KeyGen_{authority}(MSK, S) \rightarrow (TK, SK)$
- $Encrypt_{service}(PK) \rightarrow (IT)$
- $Encrypt_{user}(PK, IT, AC, data) \rightarrow (CT)$
- $Decrypt_{service}(TK, CT) \rightarrow (\widetilde{CT})$
- $Decrypt_{user}(\widetilde{CT}, SK) \rightarrow (data)$

[2] Susan Hohenberger and Brent Waters. Online/offline attribute-based encryption. *In Public-Key Cryptography–PKC 2014*, pages 293–310. Springer, 2014.

[6] Matthew Green, Susan Hohenberger, and Brent Waters. Outsourcing the decryption of abe ciphertexts. *In USENIX Security Symposium*, volume 2011, 2011.

Initialization

- Check the authority list
  - *preload the authority information*
- Initialization of pool
  - *precompute the intermediate components*
  - *store the intermediate components into the pool*



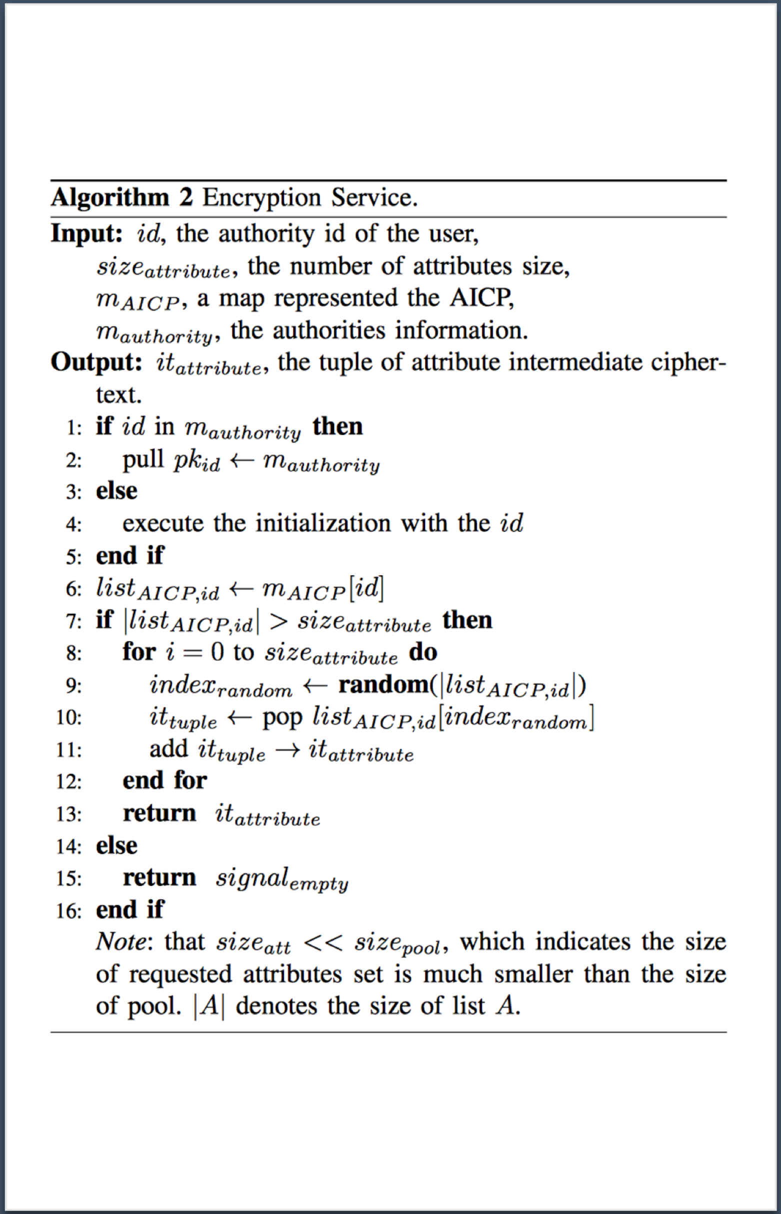
Algorithm 1 Service Initialization with General Setting.

```
Input: $type_{op}$, the service type (encrypt/decrypt),
    $type_{ABE}$, the ABE type (KP-ABE/CP-ABE),
    $list$, the default authority setting list,
    $size_{pool}$, the default size of pools.
Output: $m_{authority}$, a map for the authorities information,
    $m_{AICP}$, a map for the AICP.
initialize the map, $m_{authority}$
for $id$ in $list$ do
  $pk_{id}$ ← request the public key from authority.
  push $(id, pk_{id}) \rightarrow m_{authority}$
end for
if $type_{op}$ == "Encrypt" then
  initialize the maps $m_{AICP}$.
  for $id$ in $list$ do
    initialize a new list $list_{AICP}$
    $pk_{id} \leftarrow m_{authority}[id]$
    $s \leftarrow$ random($\mathbb{Z}_{p_{id}}$)
    for $i = 0$ to $size$ do
      if $type_{ABE}$ == "CP-ABE" then
        $\lambda, x, t \leftarrow$ random($\mathbb{Z}_{p_{id}}$)
        $C_1 = g_{id}^{\lambda} v_{id}^{t}, C_2 = (u_{id}^{x} h_{id})^{t}, C_3 = g_{id}^{t}$
        add tuple $(\lambda, x, t, C_1, C_2, C_3) \rightarrow list_{AICP}$
      else
        $r, x \leftarrow$ random($\mathbb{Z}_{p_{id}}$)
        $C_1 = w_{id}^{r}, C_2 = (u_{id}^{x} h_{id})^{r} w^{-s}$
        add tuple $(r, x, s, C_1, C_2) \rightarrow list_{AICP}$
      end if
    end for
    push $(id, list_{AICP}) \rightarrow m_{AICP}$
  end for
  return $m_{authority}, m_{AICP}$
else
  return $m_{authority}$
end if
```

Note: the function **random(A)** generates random elements between 0 and $|A|$.

Encryption Service

- Find required authority information from DB
- "calculate" the *intermediate components* (IC)
  - *check the current pool*
  - *if no enough IC*
    - return signal to change to another node
  - *if having enough IC*
    - randomly select from the pool
    - remove the selected IC from the pool



Algorithm 2 Encryption Service.

Input: id, the authority id of the user,
$size_{attribute}$, the number of attributes size,
m_{AICP}, a map represented the AICP,
$m_{authority}$, the authorities information.

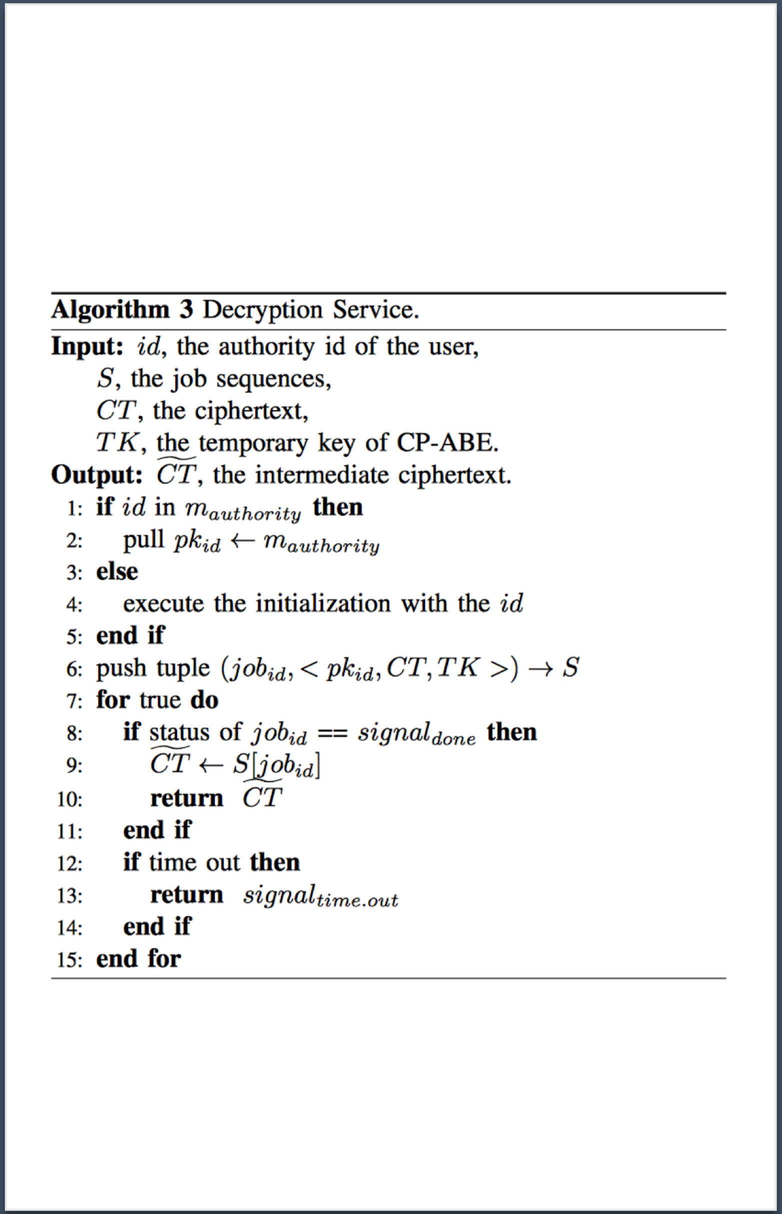
Output: $it_{attribute}$, the tuple of attribute intermediate cipher-text.

```
if $id$ in $m_{authority}$ then
    pull $pk_{id} \leftarrow m_{authority}$
else
    execute the initialization with the $id$
end if
$list_{AICP,id} \leftarrow m_{AICP}[id]$
if $|list_{AICP,id}| > size_{attribute}$ then
    for $i = 0$ to $size_{attribute}$ do
        $index_{random} \leftarrow$ random($|list_{AICP,id}|$)
        $it_{tuple} \leftarrow$ pop $list_{AICP,id}[index_{random}]$
        add $it_{tuple} \rightarrow it_{attribute}$
    end for
    return $it_{attribute}$
else
    return $signal_{empty}$
end if
```

Note: that $size_{att} << size_{pool}$, which indicates the size of requested attributes set is much smaller than the size of pool. $|A|$ denotes the size of list A.

Decryption Service

- Find required authority information from DB
  - *if no, query from the authority and store it*
- Computation job
  - *add delegation computing job to job sequence*
  - *(multiple processing in parallel)*
  - *return the intermediate computing result*



Algorithm 3 Decryption Service.

```
Input: $id$, the authority id of the user,
       $S$, the job sequences,
       $CT$, the ciphertext,
       $TK$, the temporary key of CP-ABE.
Output: $\widetilde{CT}$, the intermediate ciphertext.
if $id$ in $m_{authority}$ then
    pull $pk_{id} \leftarrow m_{authority}$
else
    execute the initialization with the $id$
end if
push tuple $(job_{id}, < pk_{id}, CT, TK >) \rightarrow S$
for true do
    if status of $job_{id}$ == $signal_{done}$ then
        $\widetilde{CT} \leftarrow S[job_{id}]$
        return $\widetilde{CT}$
    end if
    if time out then
        return $signal_{time.out}$
    end if
end for
```

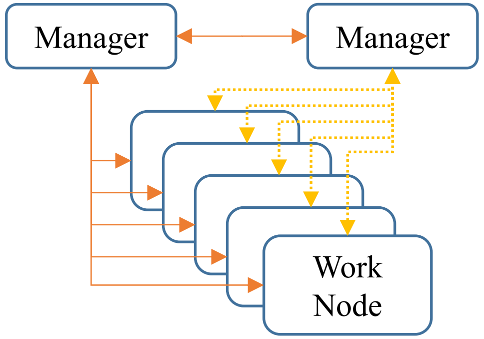
Security Discussion

- Security of Encryption Service Node
  - *sensitive modules*
    - general setting: AICG, AICP
    - hybrid setting: additional SICG, SICP

    produce intermediate components for the encryption
  - *The AIC does not include any secret*
  - *The SIC includes secret information*
    - only used in the hybrid setting
  - *The AIC/SIC is disposable*
    - when the intermediate component is used, it will be destroyed immediately
  - *The intermediate component is randomly selected*
- Security of Decryption Service Node
  - *we does not change the structure of delegation computation algorithm*

Performance Analysis

- Scalability and Availability
  - *dual-master multi-slave architecture*
    - a backup manager node with real-time synchronization
    - multiple work nodes
    - computing of each work node
- Efficiency of using ABEaaS
  - *Efficiency estimates*
    - theoretical analysis
  - *Experiment Result*



Efficiency Estimates

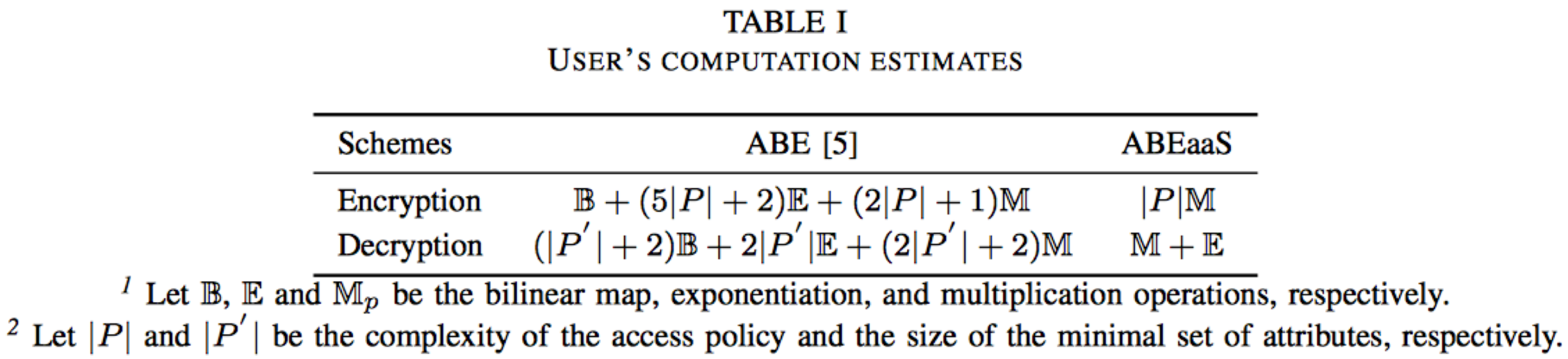


TABLE I
USER'S COMPUTATION ESTIMATES

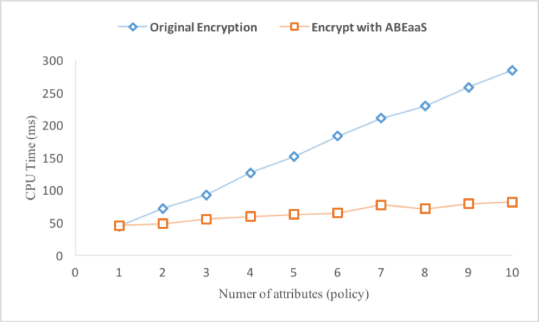
Schemes	ABE [5]	ABEaaS
Encryption	$\mathbb{B} + (5\|P\| + 2)\mathbb{E} + (2\|P\| + 1)\mathbb{M}$	$\|P\|\mathbb{M}$
Decryption	$(\|P^{'}\| + 2)\mathbb{B} + 2\|P^{'}\|\mathbb{E} + (2\|P^{'}\| + 2)\mathbb{M}$	$\mathbb{M} + \mathbb{E}$

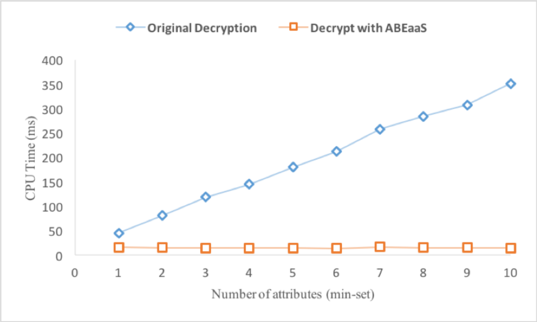
[1] Let $\mathbb{B}$, $\mathbb{E}$ and $\mathbb{M}_p$ be the bilinear map, exponentiation, and multiplication operations, respectively.

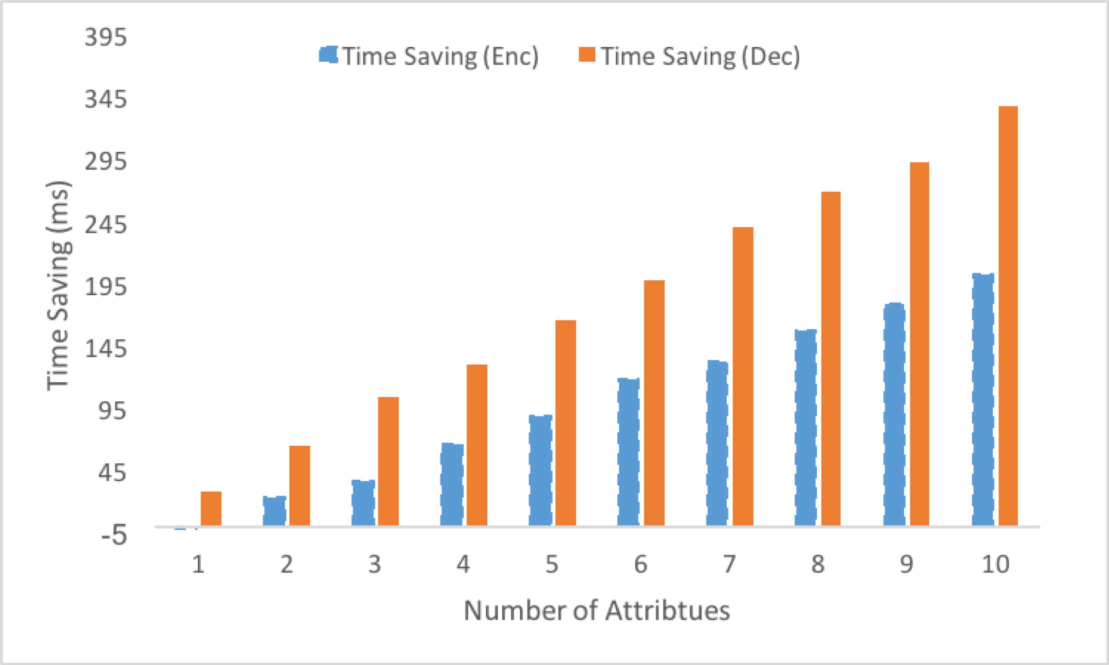
[2] Let $|P|$ and $|P^{'}|$ be the complexity of the access policy and the size of the minimal set of attributes, respectively.

Users’ operation time

- Users’ operation time
  - *Original ABE scheme v.s. ABEaaS scheme in General Setting*
- More attributes, more time saving







Thanks

Q & A