

CloudKeyBank: Privacy and Owner Authorization Enforced Key Management Framework

Author: Xiuxia Tian (xxtian@shiep.edu.cn)

Ling Huang

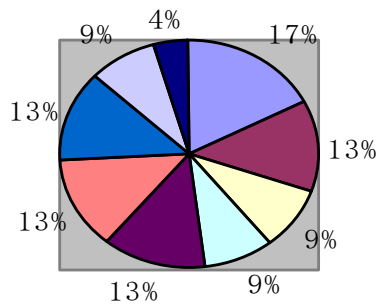
Tony Wu

Xiaoling Wang

Aoying Zhou



Background



- 邮件
- 社交网络
- 电子银行
- 即时通讯
- 在线购物
- 在线课程
- 云存储
- 协同工作
- 公共服务



Keys

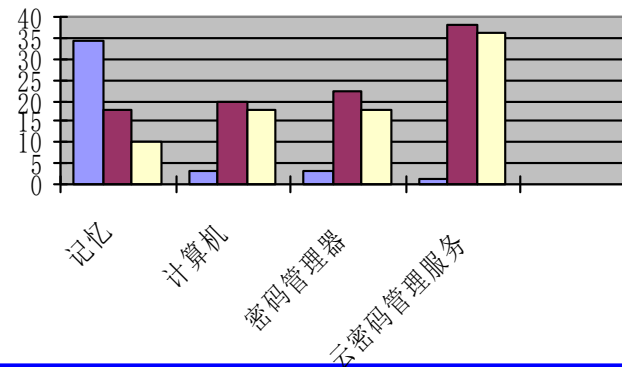


LastPass ****



Outsourced Keys

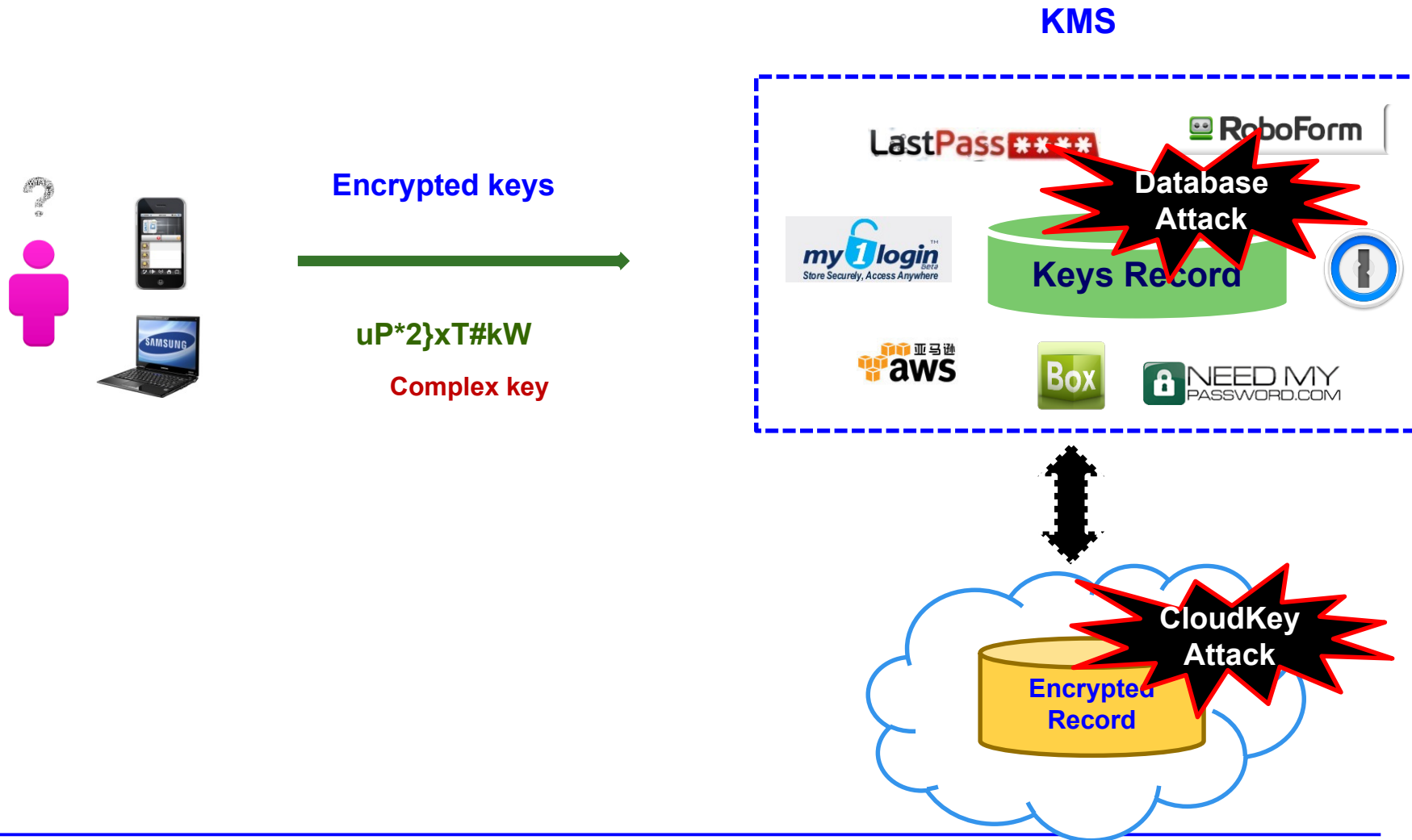
Memory



- 人员
- 口令 (密码)
- 不同的口令 (密码)

Background: Key Management Service

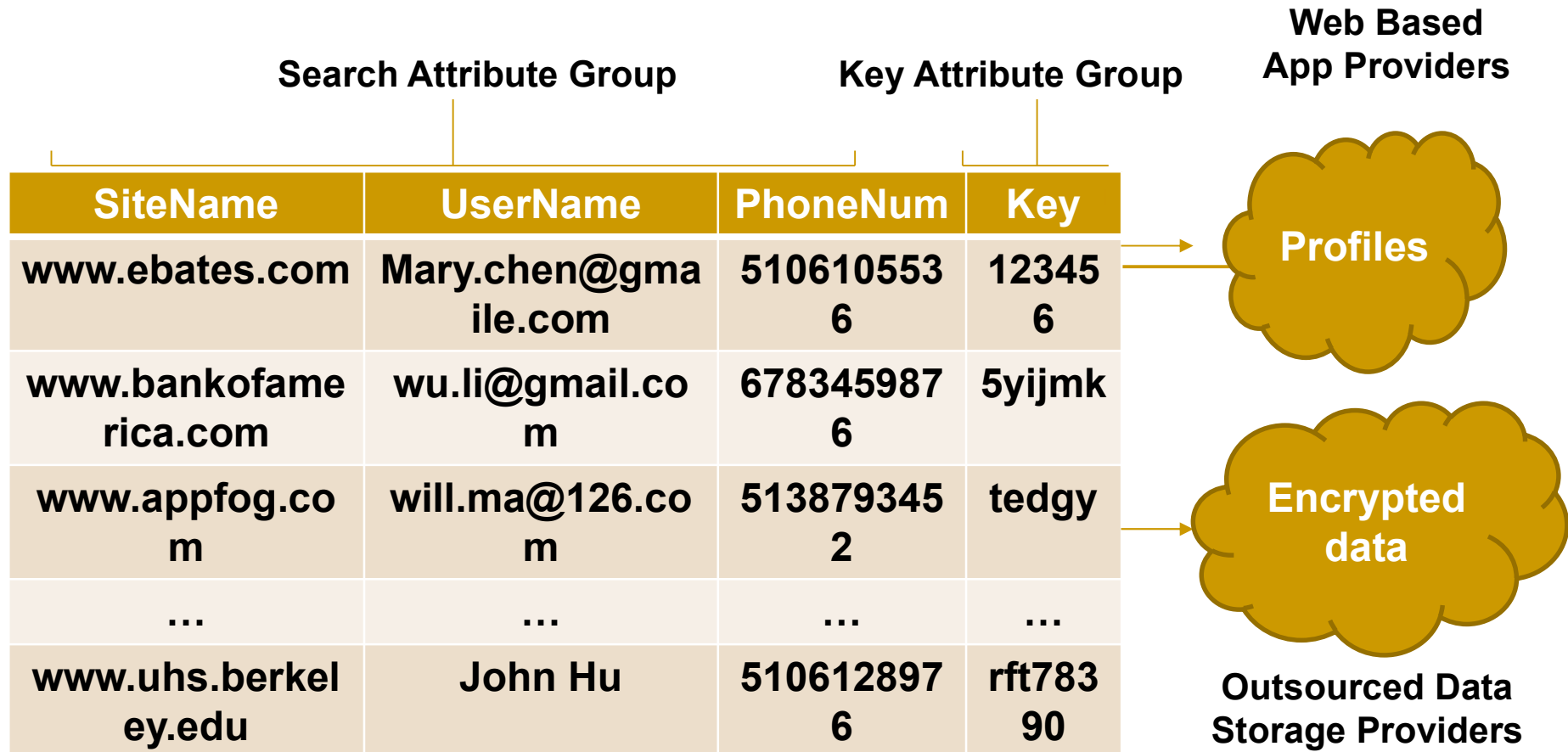
3



Each key tuple is divided into two groups, one is the Search attribute group, denoted as a vector.

Assume $t_1 = \{\vec{x}_1, \vec{k}_1\}$, where

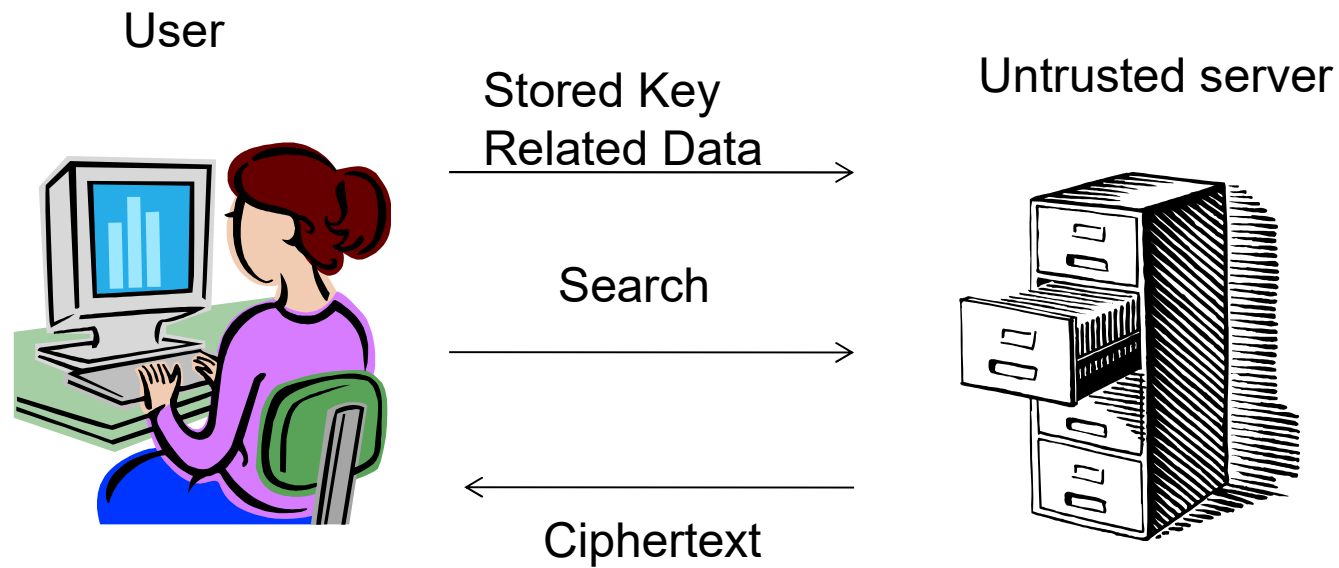
$\vec{x}_1 = \{www.ebates.com, mary.chen@gmail.com, 5106105536\}$, $\vec{k}_1 = \{123456\}$



Problem1

- The service providers are not fully trusted.
 - Keys could be disclosed if there exists a misbehaving internal employee or broken server.
 - The privacy requirements of **key attributes** in the Key attribute group is higher than that of **identity attributes** in the Search attribute group.
 - The resulted information leakage by the former is much larger than that by the latter.
-

Problem2



Identity Privacy

Key Privacy

Key Authorization

LastPass ****

Motivation

- Confidentiality and Privacy of Keys:
 - Only the authorized users can derive the shared keys of the key owner through the authorized decryption computation.
 - Search privacy on multi-identity attributes tied with keys
 - The honest-but-curious key service provider can not derive any identity attribute tied with keys from the submitted search query, but can evaluate the query from the encrypted key database correctly.
 - Owner controllable authorization over his/her shared keys
 - Only the key owner can specify and control in a fine-grained way who has the rights to access his/her shared keys through authorization on key attributes (key authorization) and authorization on submitted search query (query authorization).
-

Equality Predicate

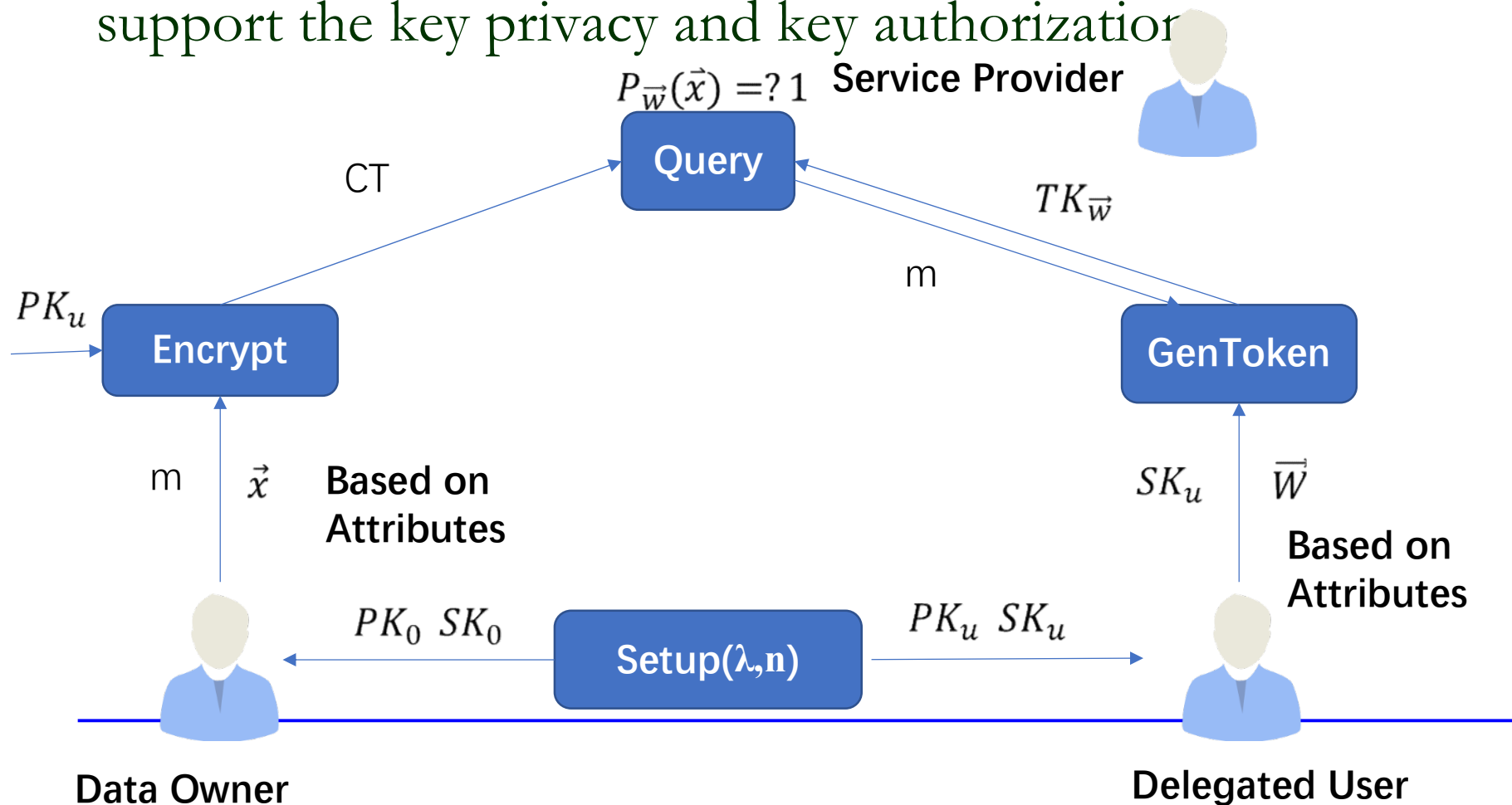
Assume there exist any two ℓ -dimensional vectors $\vec{x} \in \mathbf{x}$ and $\vec{w} \in \mathbf{w}$: encryption vector $\vec{x} = (x_i[1], \dots, x_i[\ell]) \in W^\ell$ and token vector $\vec{w} = (w_i[1], \dots, w_i[\ell]) \in (W_*)^\ell$.

$$P_{\vec{w}} = \begin{cases} 1 & \text{if for all } j \in |\vec{w}|, w[j] \neq * \\ & \text{and } w[j] = x[j] \\ 0 & \text{otherwise} \end{cases}$$

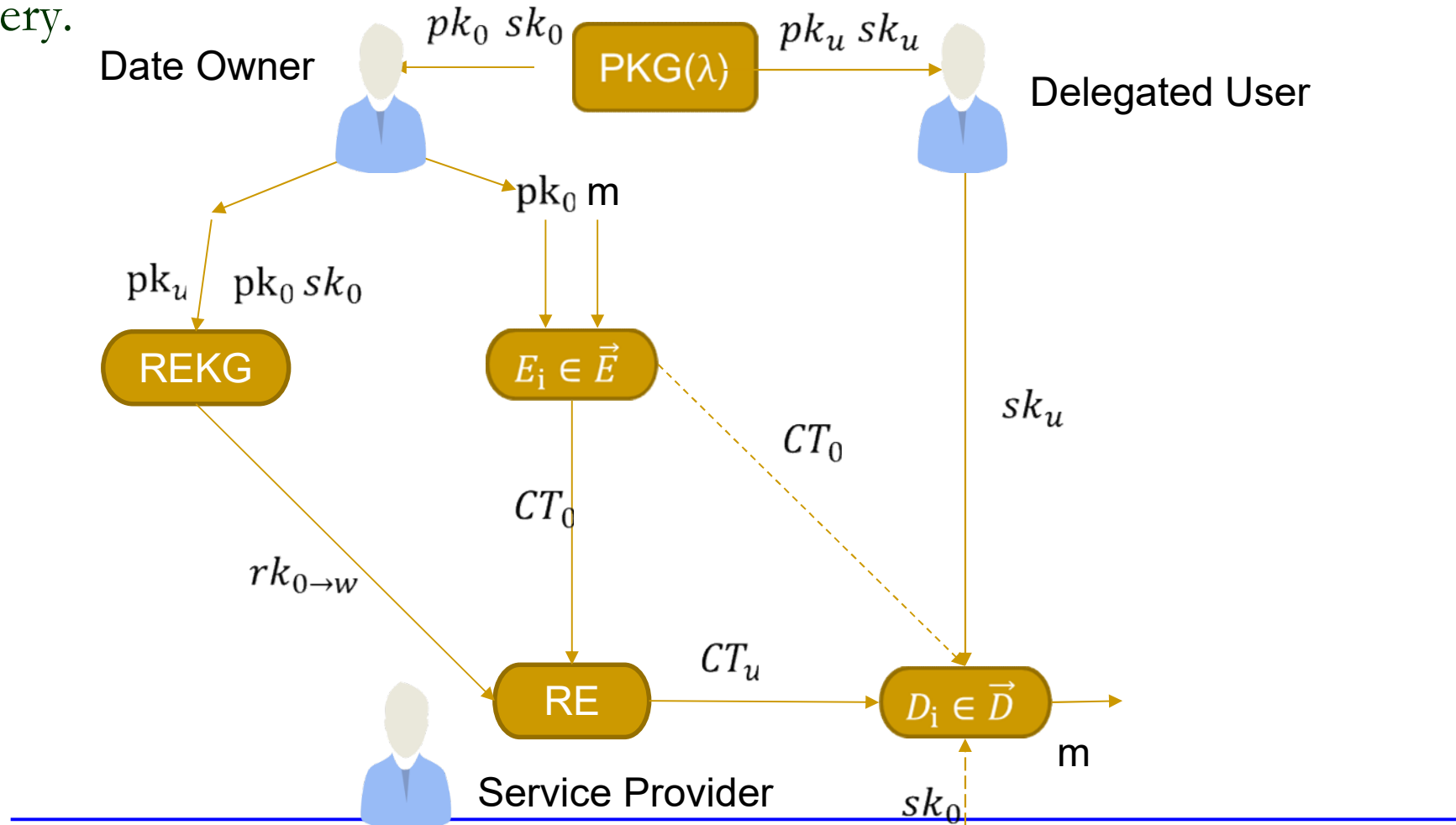
A search query denoted as $P_{\vec{w}}$ is specified by token vector $\vec{w} \in W_*$ and the boolean formula P on \vec{w} . $DB(P_{\vec{w}})$ denotes the set of tuples in Key DB which satisfies $P_{\vec{w}}$. $\vec{k} \subseteq DB(P_{\vec{w}})$ if Equation(1) succeeds where \vec{k} is a vector from vector set $\mathbf{k} = \{\vec{k}_1, \vec{k}_2, \dots, \vec{k}_n\}$, $\vec{k}_i = (k_i[1], k_i[2], \dots, k_i[|\vec{k}_i|])$ is a vector corresponding to key attributes in the i^{th} key tuple t_i .

Existing Solutions

HVE[J.Hwan Park 2011,2013] can be used to guarantee the search privacy on identity attributes, but it does not support the key privacy and key authorization



PRE[Blaze 1998, Ateniese 2005, Weng 2010] can be used to guarantee the confidentiality of keys and key authorization, but it does not support the search privacy on identity attributes and query authorization on submitted query.

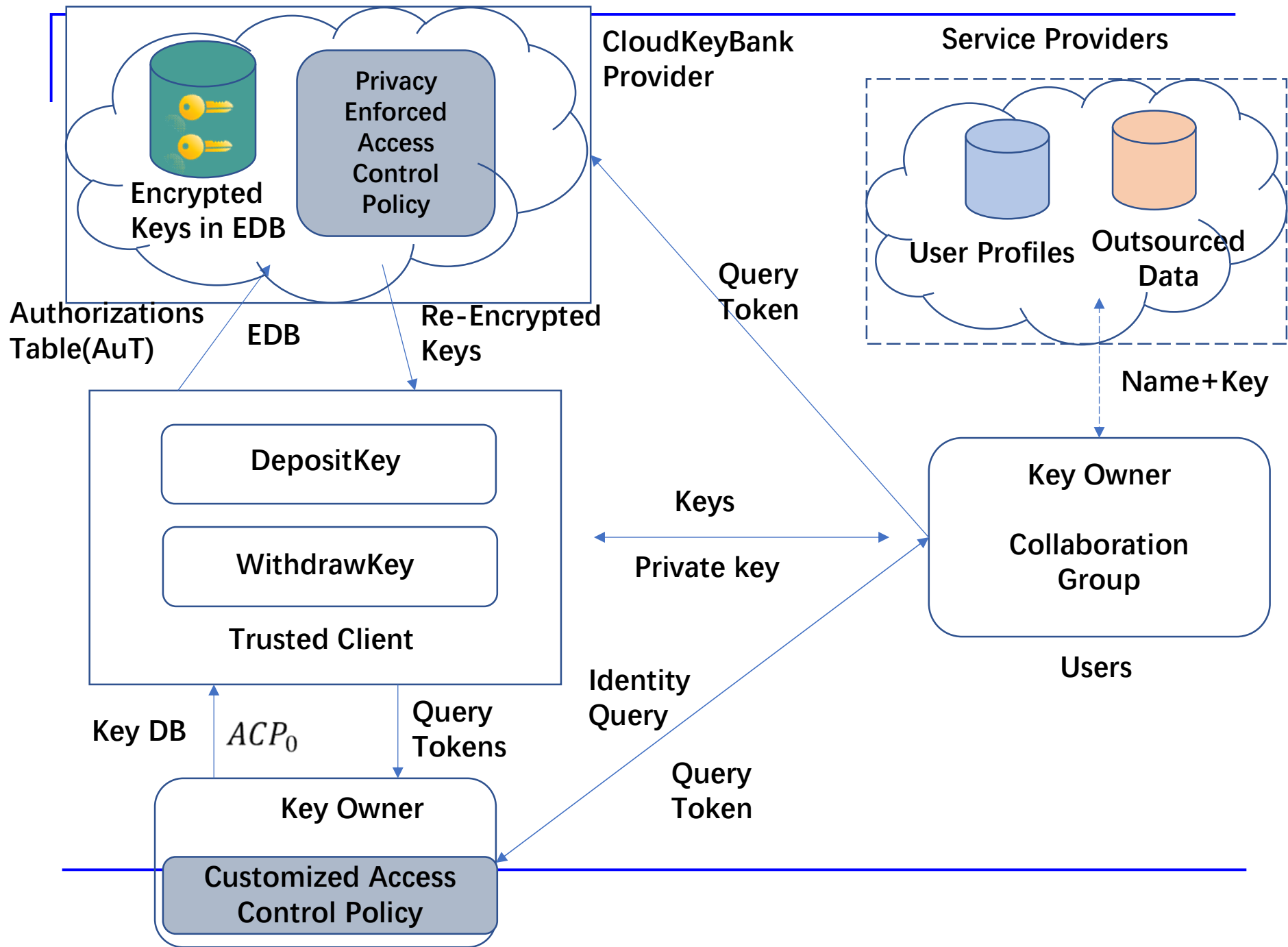


Problems of Existing Solutions

1. The increasing burden of users by remembering at least two pairs of public/private parameters;
 2. The increasing security and cost caused by at least two times of public/private key distribution;
 3. The increasing communication and computation cost between the key owner and the delegated users by encrypting and decrypting at least two times.
-

Solution

1. CloudKeyBank, an unified key management framework with enforced privacy and owner controllable authorization.
 2. We propose a new cryptographic primitive named Searchable Conditional Proxy Re-Encryption (**SC-PRE**) which combines the techniques of hidden vector encryption and proxy re-encryption seamlessly.
 3. We also propose a concrete SC-PRE scheme based on the existing schemes.
-



Roles

1. Key Owner

- a) Constructing the customized access control policy (ACP) in terms of his/her practical keys sharing requirements;
- b) Depositing Key DB by using DepositKey protocol under the support of ACP;
- c) Distributing authorized Query tokens to the delegated user based on the user's registered information.

2. CloudKeyBank Provider

- a) To enforce the privacy of identity attributes in the Search attribute group, he/she can perform search query directly by evaluating the submitted **Query token** against the encrypted key tuples in EDB;
 - b) To enforce the key authorization he/she can transform an encrypted key into the authorized re-encrypted key under the corresponding **Delegation token** stored in Authorization Table (AuT).s the wanted query and physical identity.
-

1. Trusted Client

- a) The primary privacy enforced component in CloudKeyBank framework, it mainly includes **DepositKey** and **WithdrawKey**.

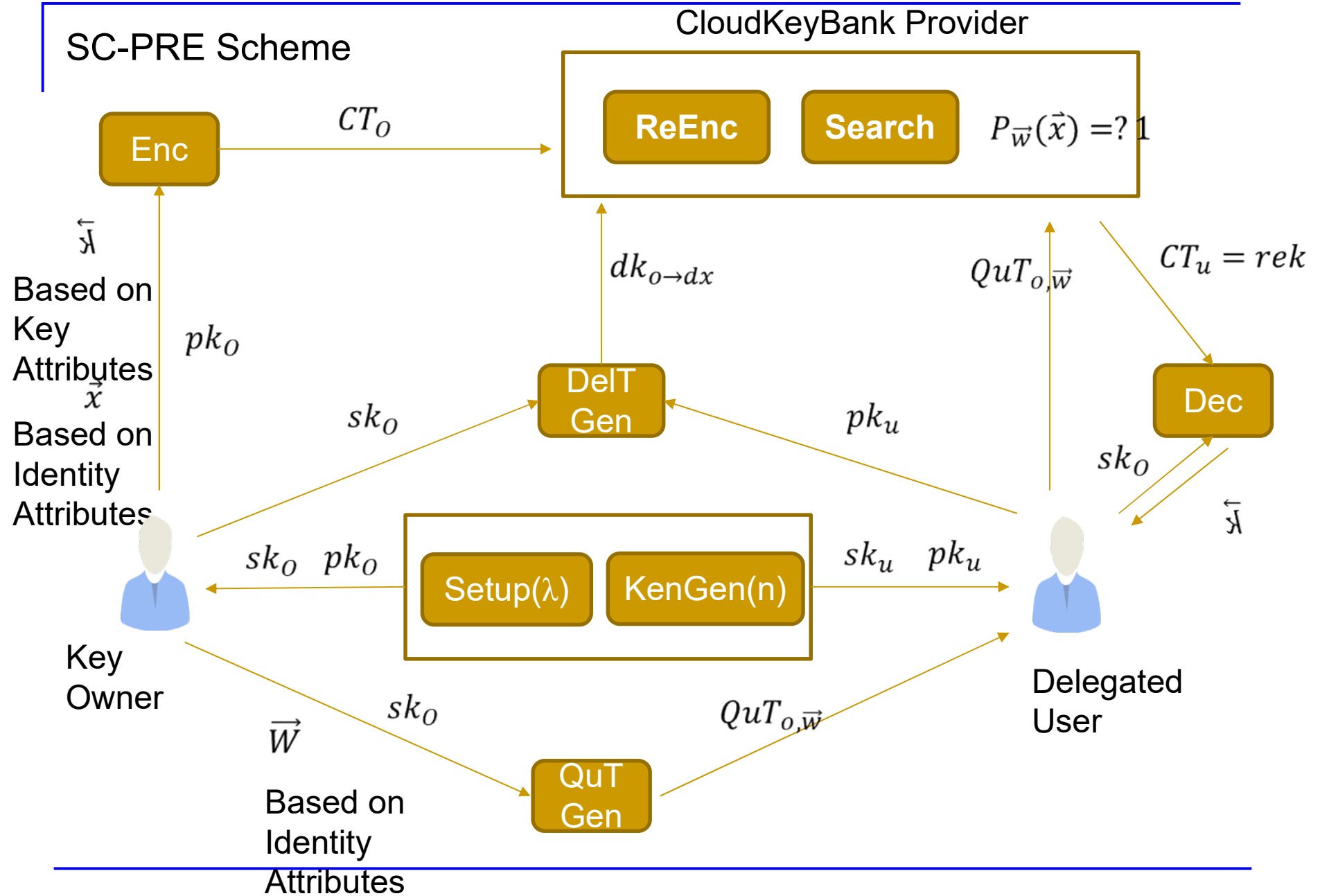
2. User

- a) Key owner and Collaboration group

3. Attack Surface

- a) Key Confidentiality
 - b) Identity Confidentiality and Linkability Privacy
 - c) Key Privacy and Key Authorization
 - d) Search Privacy and Query Authorization
-

SC-PRE Scheme



Definition 3.3(SC-PRE). SC-PRE consists of the following eight polynomial algorithms **Setup**, **KeyGen**, **Enc**, **QuTGen**, **DeITGen**, **ReEnc**, **Search** and **Dec**.

- **Setup** denoted as $\text{Setup}(\lambda) \rightarrow \text{params}$. It takes the security parameter λ as input and outputs system parameters params .

$$\text{params} = (p, G, G_T, e, g, e(g, g), H_1)$$

params is publicly known. G and G_T are two cyclic multiplication groups of prime order p , g is a generator of G and $e(g, g)$ is a generator of G_T . $H_1 : \{0,1\}^* \rightarrow G$.

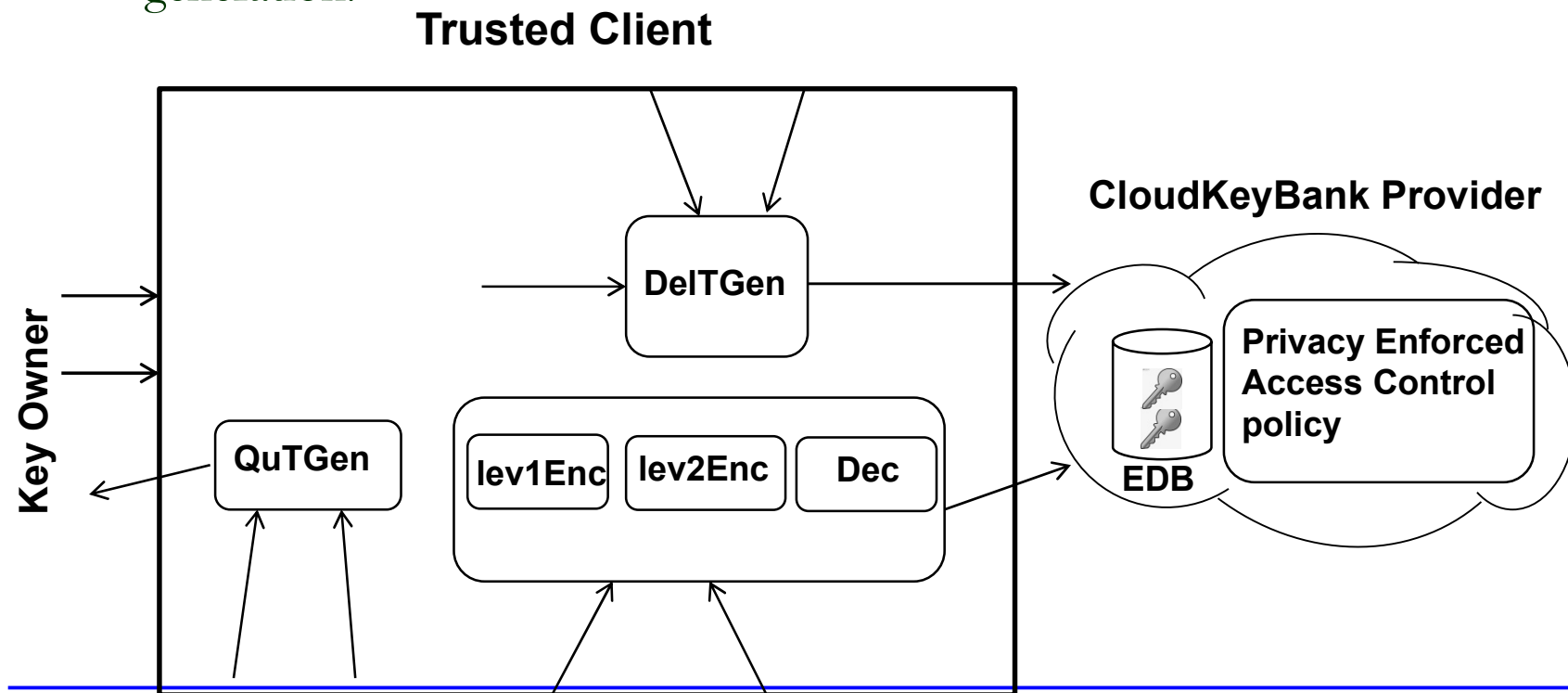
- **KeyGen** denoted as $\text{KeyGen}(|\vec{x}|) \rightarrow (pk, sk)$. It takes as input the number of identity attribute-value pairs $|\vec{x}|$. The key generation algorithm (**KeyGen**) outputs a public and private key pair (pk, sk) for the user. For example (pk_0, sk_0) for the key owner, (pk_{du}, sk_{du}) for the delegated user of the key owner.
- **Enc** denoted as $\text{Enc}(pk_0, \vec{x}, \vec{k}) \rightarrow CT$. It takes as input the public pk_0 of key owner, a tuple $t = (\vec{x}, \vec{k})$, vector $\vec{k} \subseteq K$ of key attributes and vector $\vec{x} \subseteq W$ of identity attributes. The encryption algorithm (**Enc**) based on HVE and PRE outputs the ciphertext CT of key tuple. In this process \vec{k} is first transformed to the corresponding ek so as to hide the real \vec{k} from the CloudKeyBank provider.

- **QuTGen** denoted as $QuTGen(sk_o, \vec{w}) \rightarrow QuT_{o, \vec{w}}$. It takes as input the private key sk_o of key owner and vector $\vec{w} = (w[1], w[2], \dots, w[|\vec{w}|])$ of identity attributes, $\vec{w} \subseteq W_*$. The query token generation algorithm (QuTGen) outputs a query token $QuT_{o, \vec{w}}$ for subsequent privacy search query on encrypted Key DB and query authorization on submitted queries.
- **DelTGen** denoted as $DelTGen(sk_o, pk_{du}) \rightarrow dk_{o \rightarrow du}$. It takes as input the private key sk_o of key owner and the public key pk_{du} of the delegated user. The delegation key generation algorithm (DelTGen) outputs the delegation token $dk_{o \rightarrow du}$ that is computed from the partial parameters of sk_o and pk_{du} .
- **Search** denoted as $Search(CT, QuT_{o, \vec{w}}) \rightarrow ek$. It takes as input the tuple ciphertext CT under public key pk_o and key owner's query token $QuT_{o, \vec{w}}$, and outputs the encrypted key ek on the condition that $P_{\vec{w}} = 1$ succeeds by evaluating $QuT_{o, \vec{w}}$ against the hidden vector \vec{x} in encrypted tuple CT , otherwise outputs \perp .

-
- To achieve the minimum information leakage in the process of privacy and owner controllable authorization enforcement, we introduce dual authorization tokens including the **Query token** and the **Delegation token**.
 - By using the dual authorization tokens the key owner can encrypt the Key attribute group in such a way that only the user with the appropriate tokens can gain access to the shared key of key owner.
 - Both the delegated user and the CloudKeyBank provider can not derive the private key of key owner from the submitted Query token, but the CloudKeyBank still can perform efficient search queries by evaluating the Query token from each encrypted key tuple.
-

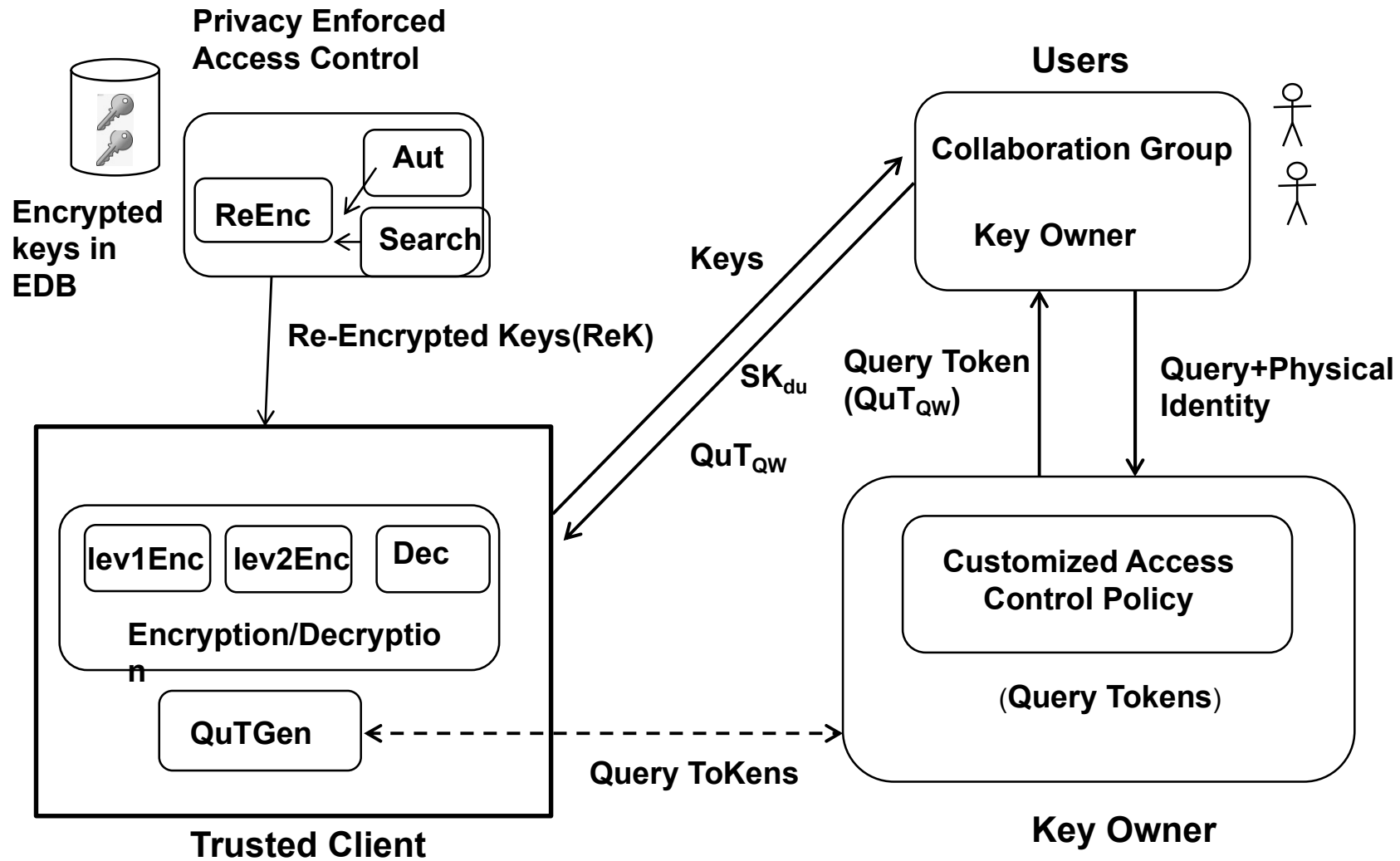
Trusted Client

- DepositKey protocol is used to help the key owner outsource his/her keys to the CloudKeyBank provider with minimum information leakage such as the leakage of tuple identifier.
 - DB encryption, Query token generation and Delegation token generation.



-
- WithdrawKey protocol is used to help the delegated users obtain the shared keys of key owner with minimum privacy leakage such as the leakage of Query token.
 - Search query on EDB, Conditional delegation and Key derivation.
-

CloudKeyBank Provider



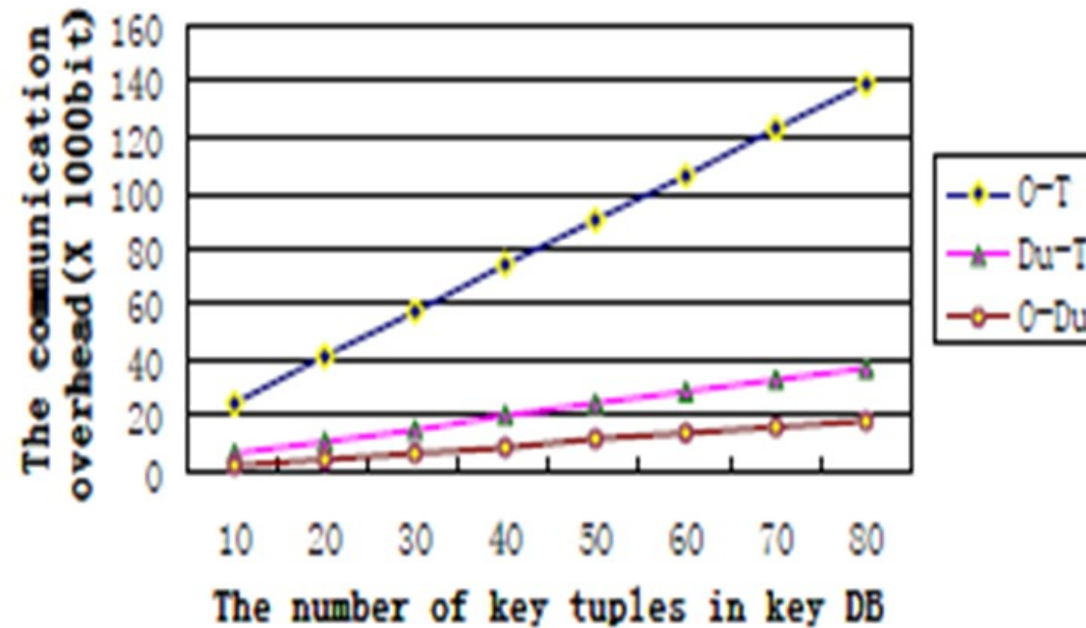
Security analysis

PWManager	LoginDomain	Name	Password	Collaboration	SearchPrivacy
LastPass	X	√	√	√	X
My1login	X	√	√	√	X
PassworBox	X	X	√	√	X
RoboForm	X	X	X	X	X
NeedMyPassword	X	X	X	X	X
CloudKeyBank	√	√	√	√	√

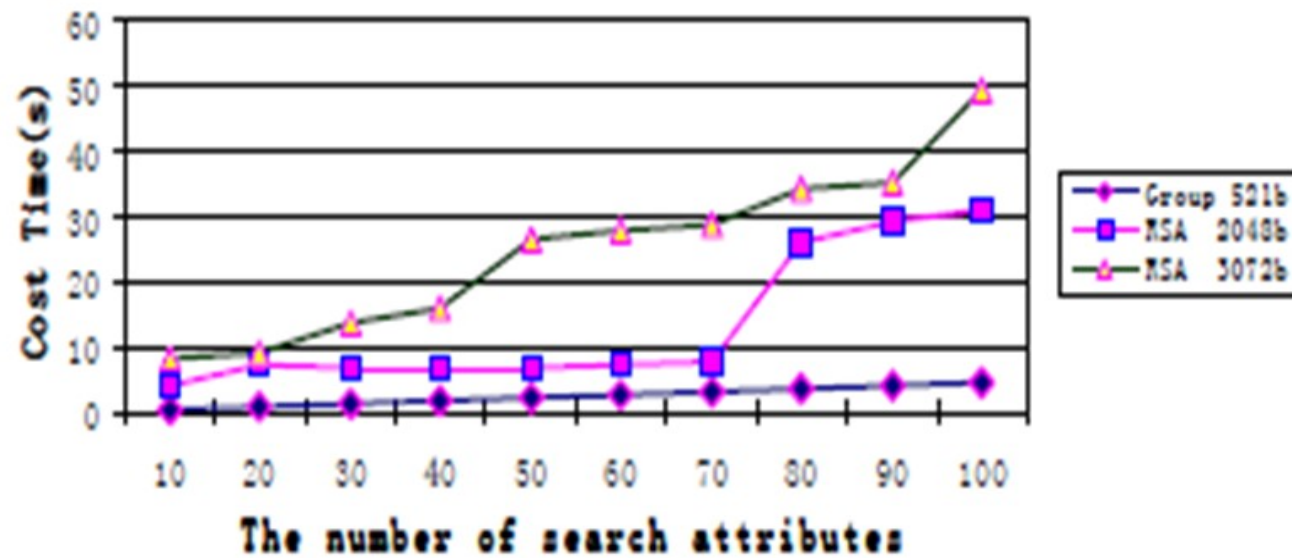
Security analysis

Security Scheme	Key Confidentiality and Privacy	Search Privacy (on Identity Attributes)	Owner Controllable Authorization	
			Key Authorization (on Key Attributes)	Query Authorization
Hacigumus[13,14]	√			
Cash[15,16]		√		
Shang[18]	√			
Nabeel[19]	√			
Tian[22,25]	√		√	
Li[27]		√		√
Our Solution	√	√	√	√

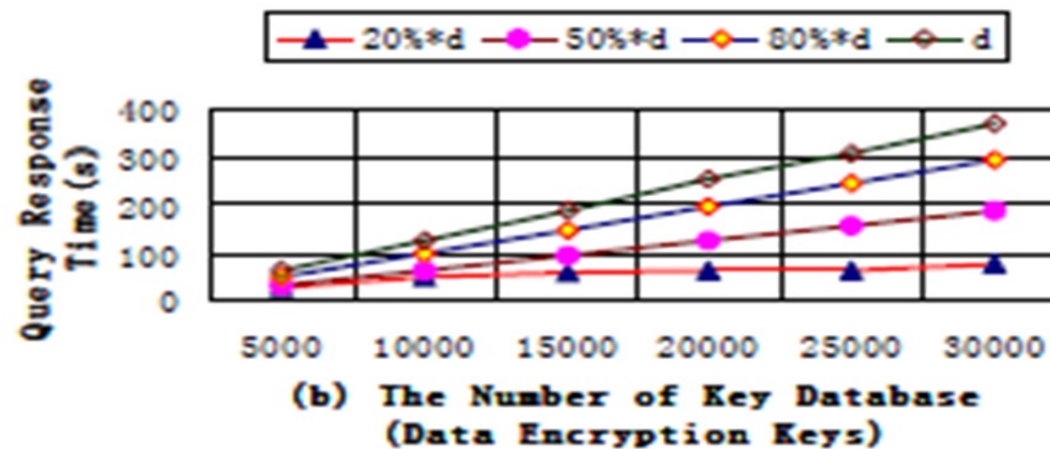
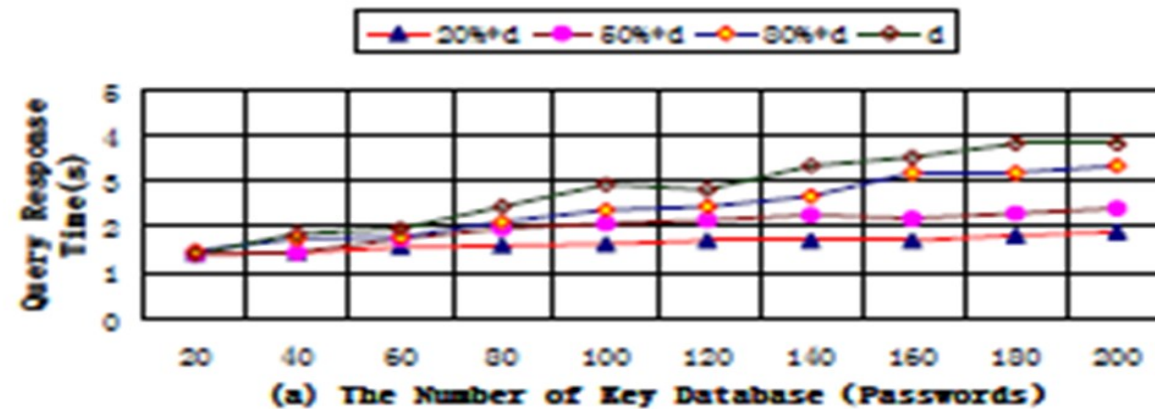
Performance analysis



Performance analysis



Performance analysis



■ Xiuxia Tian, Ling Huang, Tony Wu, Xiaoling Wang, Aoying Zhou. CloudKeyBank: Privacy and Owner Authorization Enforced Key Management Framework. IEEE Transactions on Knowledge and Data Engineering(IEEE TKDE), 27(12): 3217-3230, 2015.

■ Amazon的密钥管理服务: <https://aws.amazon.com/cn/kms/>

■ 阿里云的密钥管理服务:
<https://help.aliyun.com/product/28933.html>

■ 腾讯云的密钥管理服务:
<https://cloud.tencent.com/product/kms>

谢谢！

Q & A
