Mukti PADHYA, Devesh C. JINWALA, 2019. MULKASE: a novel approach for key-aggregate searchable encryption for multi-owner data. *Frontiers of Information Technology & Electronic Engineering*, 20(12):1717-1748. https://doi.org/10.1631/FITEE.1800192

MULKASE: a novel approach for key-aggregate searchable encryption for multi-owner data

Key words: Searchable encryption; Cloud storage; Key-aggregate encryption; Data sharing

Corresponding author: Mukti PADHYA
E-mail: mukti.padhya@yahoo.in
ORCID: http://orcid.org/0000-0002-0498-4188

Motivation

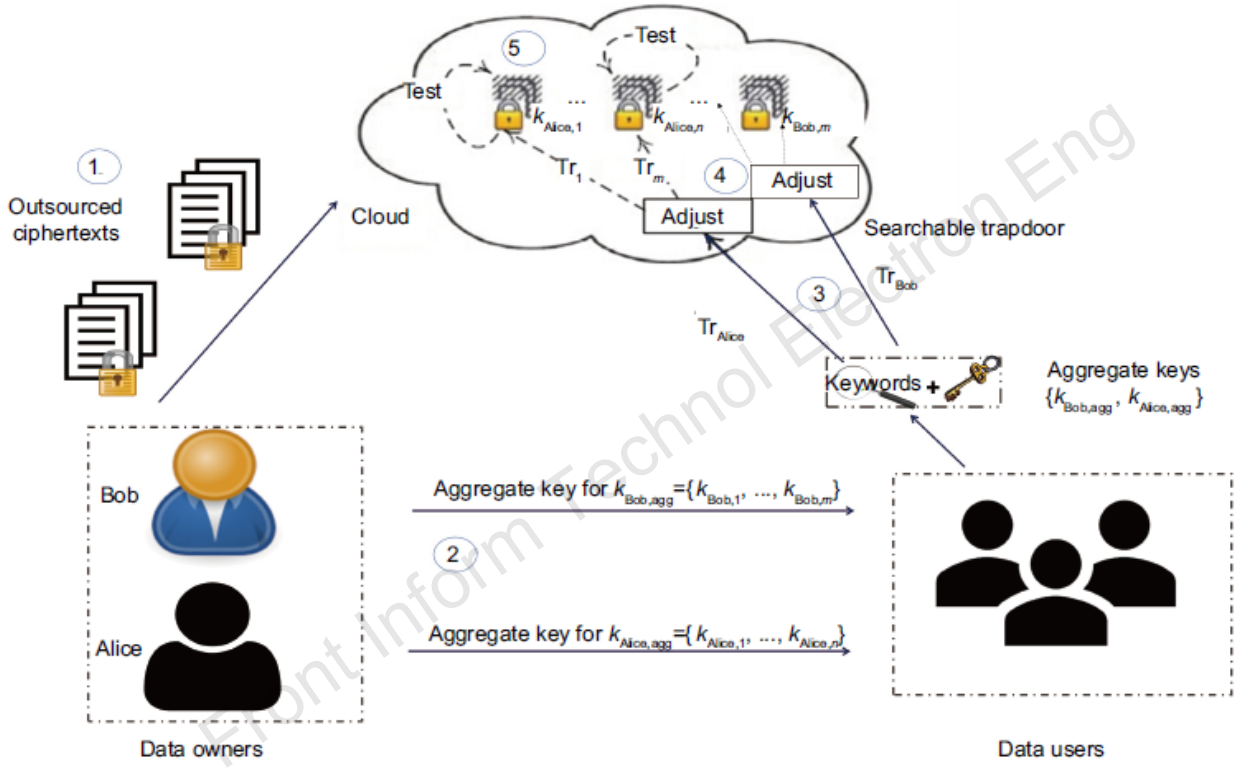

Fig. 4 Keyword search across the set of data owned by multiple users: the existing KASE approach

Steps: (1) Data owners (Alice, Bob, and other people) upload a ciphertext onto the cloud server; (2) Each data owner sends the aggregate key to the data user to share set S of documents; (3) The data user generates searchable trapdoors using each aggregate key received from the different data owners and a query keyword, and then submits all the trapdoors to the server; (4) The cloud server transforms each input aggregate trapdoor Tr to several Tr_i for each $i \in S$ (each Tr_i is an actual trapdoor to perform a keyword search across the i^{th} document); (5) The cloud server checks whether the i^{th} document contains the query keyword or not using the individual Tr_i

Motivation (Cont'd)

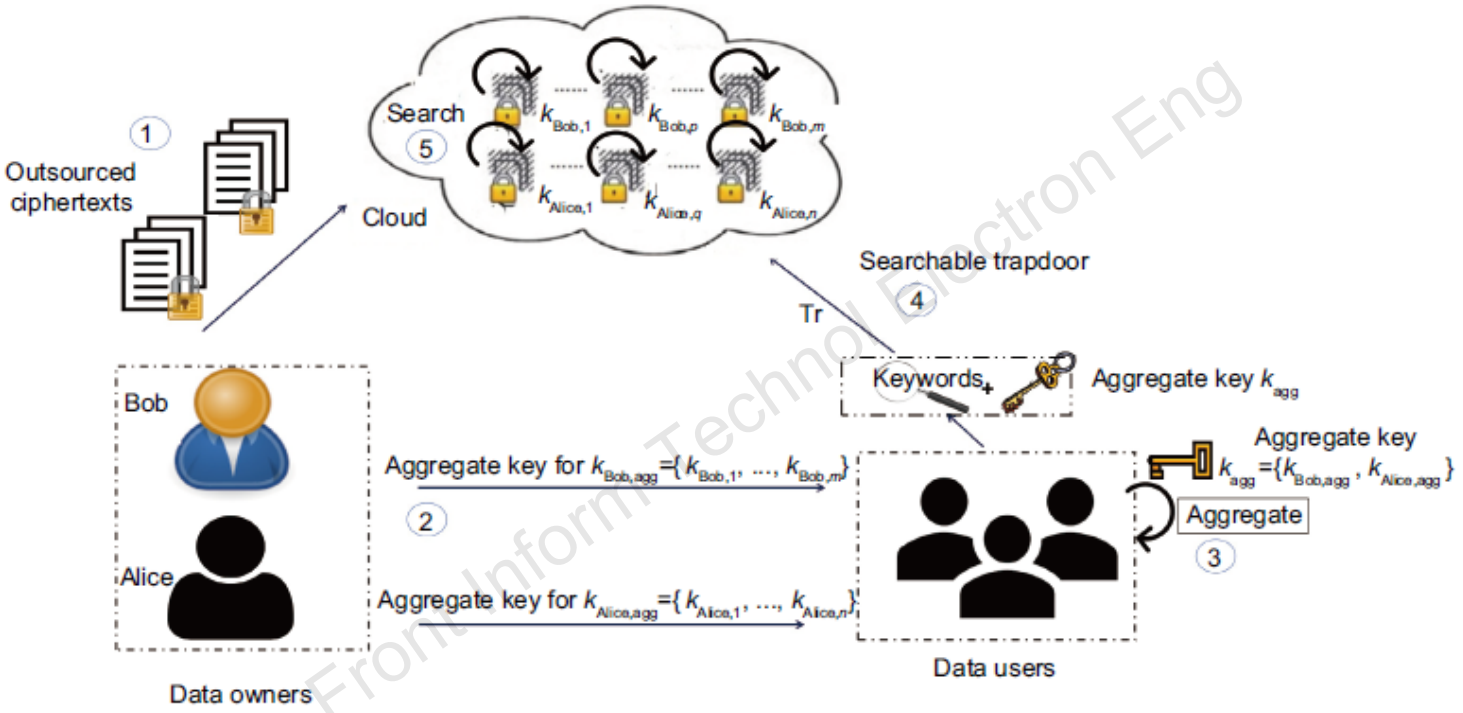

Fig. 5 Keyword search across the set of data owned by multiple users: the MULKASE approach

Steps: (1) Data owners (Alice, Bob, and other people) upload a ciphertext onto the cloud server; (2) Each data owner sends the aggregate key to the data user to share set S of documents; (3) The data user generates a single aggregate key using aggregate keys received from the different data owners; (4) The data user constructs a searchable trapdoor using an aggregate key and a query keyword, and then submits a trapdoor to the server; (5) The cloud server checks whether the i^{th} document contains the query keyword or not using Tr

Main idea

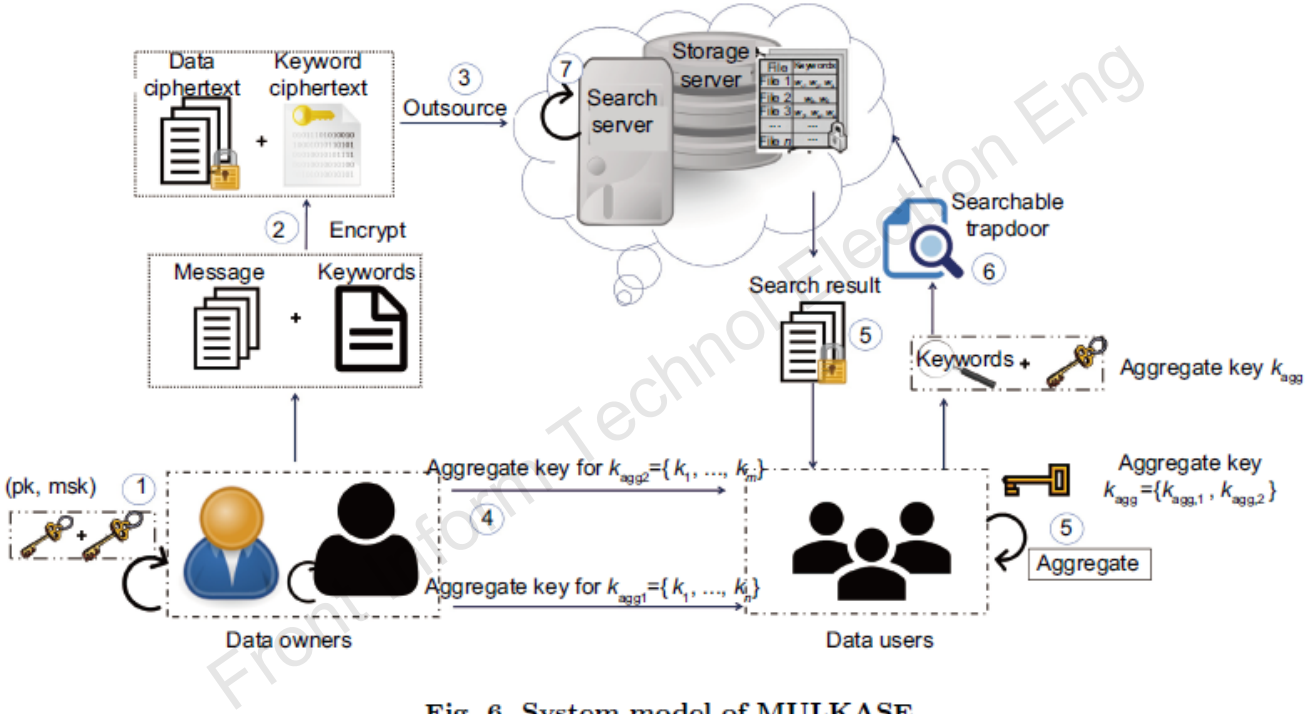

Fig. 6 System model of MULKASE

Steps: (1) The cloud server sets up the system's public parameters and the data owner generates a key pair (public, master-secret); (2) Data owners encrypt the payload message (m) along with the related keyword (KW) using the public key (pk) and searchable encryption key (k_i), and generate the data-ciphertext, keyword-ciphertext, and public information (δ_i) ($C_i = (\delta_i, C_m, C_{\mathrm{KW}})$); (3) Data owners upload ciphertexts (C_i) onto the cloud server; (4) Each data owner sends the aggregate key (k_{agg}) to the data user for sharing set S of their documents; (5) The data user generates a single aggregate key (k_{agg}) using each aggregate key ($k_{\mathrm{agg},1}, k_{\mathrm{agg},2}, \ldots, k_{\mathrm{agg},m}$) received from the different data owners; (6) The data user constructs a searchable trapdoor Tr using an aggregate key (k_{agg}) and a query keyword (w), and submits a trapdoor to the server; (7) The cloud server runs the proposed Test($\cdot$) algorithm to check whether the i^{th} document contains the query keyword (w) or not using the submitted trapdoor Tr

Security analysis

Theorem 1 The proposed MULKASE method is IND-CKA secure under the DDH assumption, if there is no polynomial time adversary *A* who can win the game with a non-negligible advantage Adv_A (l) in security parameter l.

Theorem 2 The proposed MULKASE method is IND-KGA secure, assuming that the DDH problem is hard to solve.

Theorem 3 The proposed MULKASE method can achieve the goal of query privacy.

Lemma 1 An attacker is not able to learn a given keyword in a query from the submitted trapdoor.

Lemma 2 An attacker is not able to learn a keyword in a document from the stored keyword-ciphertexts and the relevant public information.

Security analysis (Cont'd)

Theorem 4 The proposed MULKASE method supports controlled search.

Lemma 3 An authorized user or server is not able to carry out a keyword search across any set of documents, not in the range of the user's aggregate key.

Lemma 4 Even when a malicious authorized user collides with the cloud server, he/she is not able to carry out a keyword search across any document that is not within the range of the user's aggregate key.

Lemma 5 An attacker is not able to generate the new aggregate key for any new set S' of indices from the known aggregate key.

Theorem 5 MULKASE is secure against a cross-pairing attack.

Major results

Table 2 Comparative analysis of significant characteristics

Method	MO	TT	KSI	FS	IND-CPA	IND-CKA	CPrA	SRV
Cui et al. (2016)	×	✓	×	×	×	×	×	×
Li et al. (2016)	✓	✓	×	×	×	×	×	✓
MULKASE	✓	×	✓	✓	✓	✓	✓	×

MO: search across multi-owner data using a single aggregate key; TT: trapdoor transformation required; KSI: the aggregate key size being independent of the maximum number of ciphertexts held by a data owner; FS: support federated search using an aggregate trapdoor; IND-CPA: secure against message indistinguishability attacks; IND-CKA: secure against keyword indistinguishability attacks; CPrA: secure against cross-pairing attacks; SRV: verification of search result using an aggregate key

Major results (Cont'd)

Table 3 Comparison of storage overhead

Method	PubK	k_{agg}	Trapdoor		Ciphertext
			Searching single-owner data	Searching multi-owner data	
Cui et al. (2016)	$(2n+1)\|G\|$	$\|G\|$	$\|G\|$	$U\|G\|$	$2G+ G_T$
Li et al. (2016)	$(2n+1)\|G\|$	$\|G\|$	$\|G\|$	$\|G\| + U \cdot \mathrm{AV}$	$3G+G_T$
MULKASE	$(n+1)\|G\|$	$\|G\|$	$2\|G\|$	$2\|G\|$	$2G+ G_T$

n: the number of documents that belong to the data owner; U: the number of different data owners (i.e., k_{agg} aggregates the power to search across U data owner's documents); AV: the size of the auxiliary value used to search across multi-owner data; $|G|$: the bit length of the element that belongs to G; G and G_T: bilinear groups

Table 4 Comparison of computational overhead

Method	Encryption*	Aggregate key generation		Trapdoor generation**		Keyword search***	
		Single-owner	Multi-owner	Single-owner	Multi-owner	Single-owner	Multi-owner
Cui et al. (2016)	$2E+3P+2M$	$\|S\|\cdot M$	–	M	$U\cdot M$	$(\|S\|\cdot M)+$ $(\|S\|\cdot M+2P)$	$(\|S\|\cdot M)+$ $U(\|S\|\cdot M+2P)$
Li et al. (2016)	$2E+3P+2M$	$\|S\|\cdot M$	$U\cdot M$	M	$(M+E)+$ $U(M+E)$	$(\|S\|\cdot M)+$ $(\|S\|\cdot M+2P)$	$U(\|S\|\cdot M)+$ $U(\|S\|+2M)+2P$
MULKASE	$2E+2P+M$	$\|S\|\cdot M$	$U\cdot M$	$2E+M$	$2E+M$	$3P+\|S\|\cdot M$	$3P+\|S\|\cdot M$

Single-owner: searching single-owner data; Multi-owner: searching multi-owner data. E: exponentiation; M: scalar multiplication; P: pairing; U: number of data owners; $|S|$: size of set S. *: data-ciphertext+keyword-ciphertext; **: trapdoor+auxilliary value; ***: trapdoor transformation+search

Major results (Cont'd)

Table 5 Comparison of communication overhead

Method	Communication overhead	
	Single-owner	Multi-owner
Cui et al. (2016)	T	$U \cdot T$
Li et al. (2016)	T	$T + U \cdot \mathrm{AV}$
MULKASE	T	T

T: number of trapdoors, U: number of data owners, AV: number of auxiliary values used to search over multi-owner data

Major results (Cont'd)

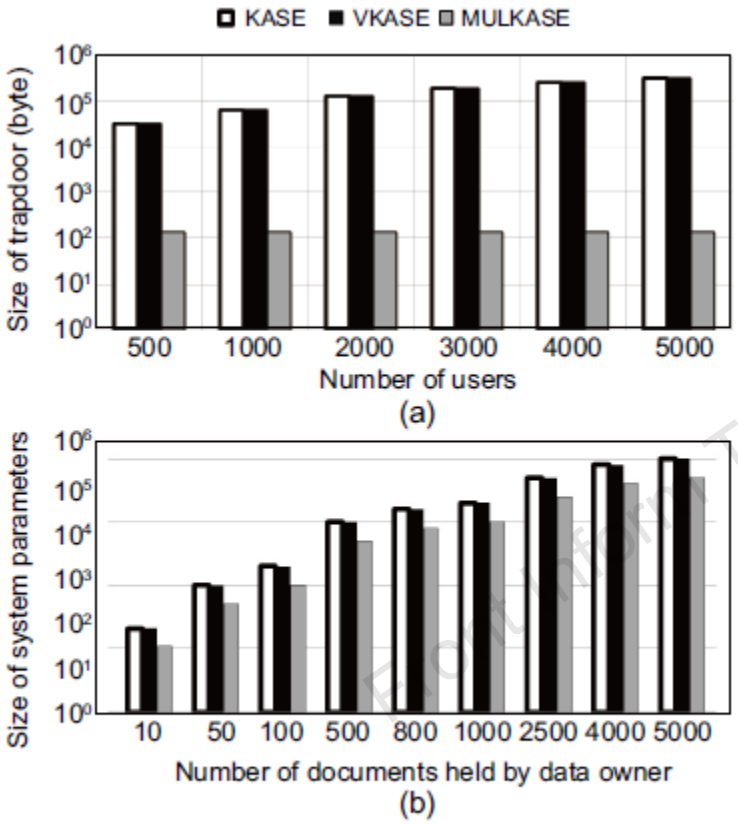

Fig. 10 Storage overhead: (a) size of trapdoors; (b) size of system parameters

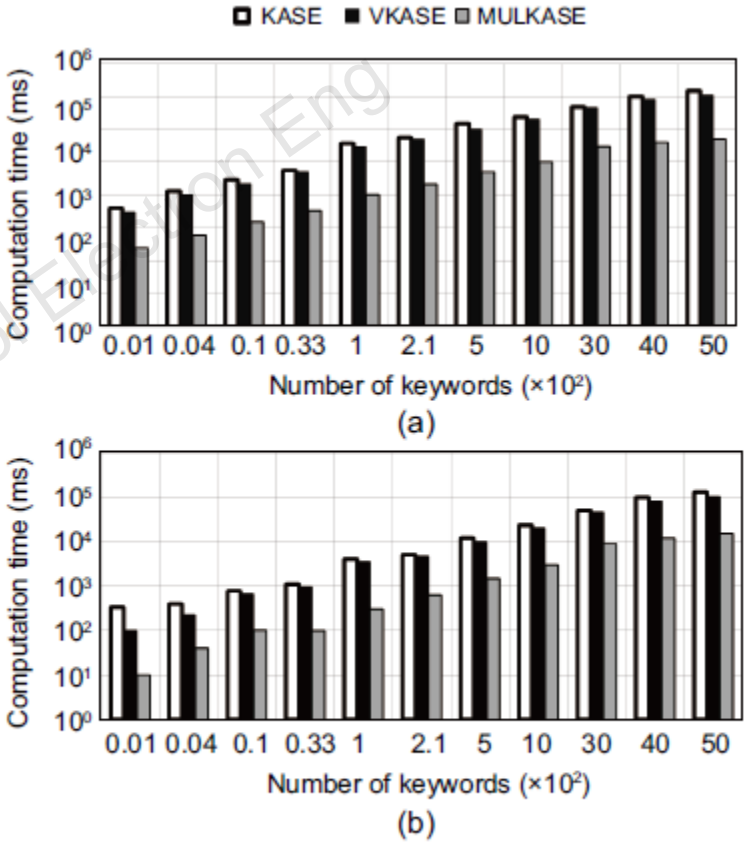

Fig. 11 Computation time of Encrypt(·): (a) mobile device; (b) computer

Major results (Cont'd)

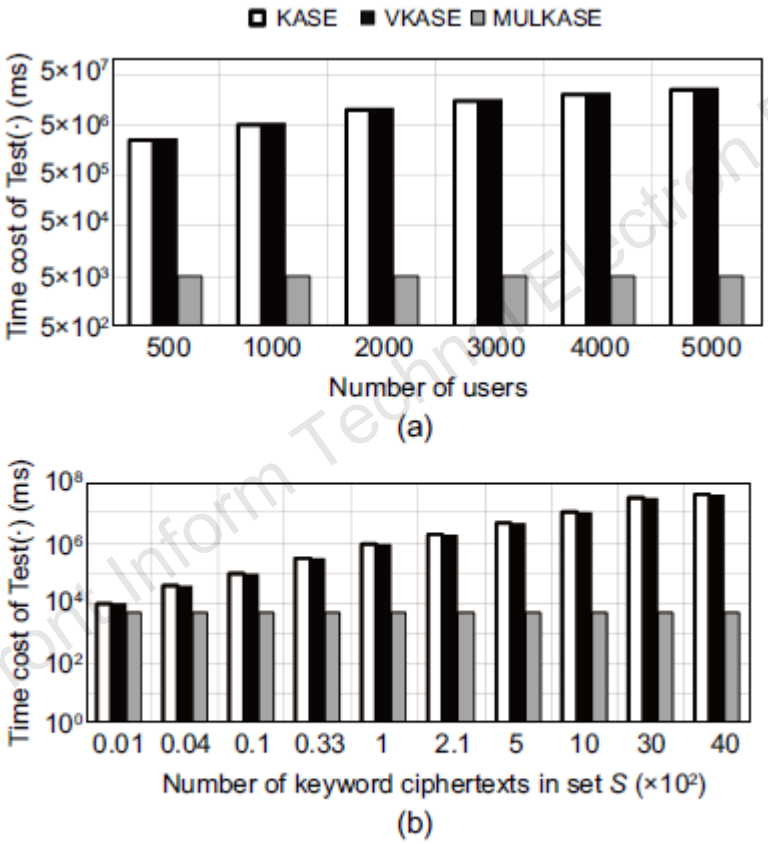

Fig. 12 Computation time of Test(·): (a) number of users; (b) number of keyword ciphertexts in set S

Federated search

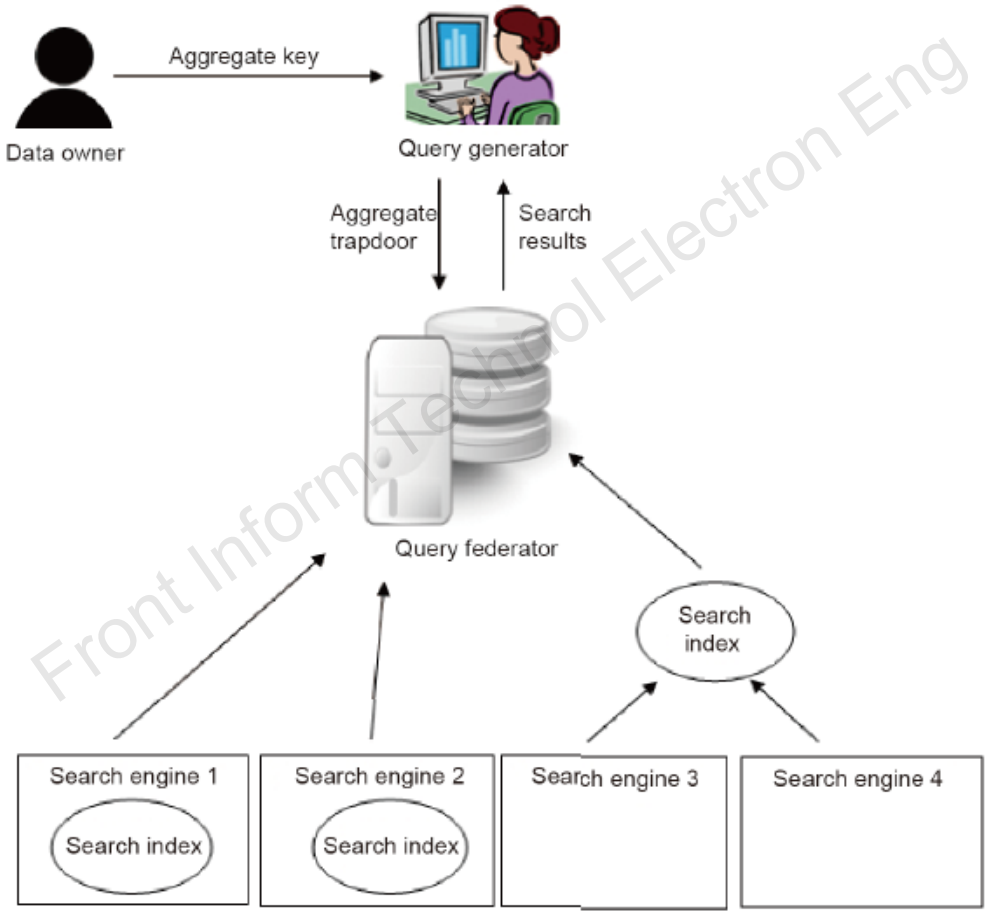

Fig. 13 System model for a federated search using MULKASE

Conclusions

1. The searchable group data sharing scheme that allows search of multi-owner data using a single trapdoor is designed.
2. A novel MULKASE method that allows an authorized user to search for a keyword across multi-owner dataset using a single aggregate trapdoor is presented.
3. Security and efficiency analyses confirm that the proposed MULKASE provides an efficient, secure, and searchable data sharing system for public cloud storage.
4. In addition, the proposed MULKASE method is applicable for carrying out a federated search.