(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0067538
(43) 공개일자 2021년06월08일

(51) 국제특허분류(Int. Cl.)
G06F 21/62 (2013.01) *G06F 21/60* (2013.01)
(52) CPC특허분류
G06F 21/6227 (2013.01)
G06F 16/24564 (2019.01)
(21) 출원번호 10-2019-0157284
(22) 출원일자 2019년11월29일
심사청구일자 2019년11월29일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
허준범
경기도 용인시 기흥구 보정로 30, 119동 201호(보정동, 동아솔레시티아파트)
한창희
인천광역시 남동구 호구포로 924, 109동 2204호 (만수동, 햇빛마을벽산아파트)
(74) 대리인
김홍석

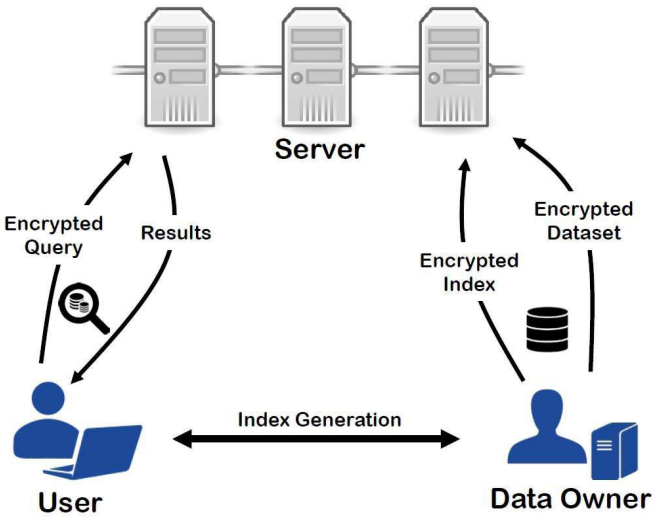
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **다수 개체를 위한 확장성 있고 안전한 유사도 검색 장치 및 방법**

(57) 요 약

다수 개체를 위한 확장성 있고 안전한 유사도 검색 시스템과 방법이 개시된다. 상기 유사도 검색 시스템은 사용자 단말, 데이터 소유자의 단말인 소유자 단말, 및 서버를 포함하고, 상기 서버는 상기 소유자 단말로부터 데이터(M)의 암호화된 데이터를 수신하고, 상기 사용자 단말로부터 트랩도어(trapdoor)를 포함하는 쿼리(query)를 수신하여 상기 사용자 단말의 데이터(M')와 상기 데이터(M)의 유사도를 판단하고, 유사도 판단 결과를 상기 사용자 단말로 송신하고, 상기 소유자 단말은 상기 사용자 단말과의 통신을 통해 상기 데이터(M)를 검색하기 위한 인덱스를 생성하고, 상기 인덱스를 상기 서버로 송신하고, 상기 사용자 단말은 상기 데이터(M')와 상기 데이터(M) 사이의 유사도를 판단하기 위한 트랩도어를 생성하고, 상기 쿼리는 상기 트랩도어를 포함하고, 상기 서버는 상기 인덱스와 상기 트랩도어를 이용하여 상기 데이터(M)와 상기 데이터(M')의 유사도를 판단한다.

대 표 도 - 도2



(52) CPC특허분류
G06F 21/602 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 20180002690021001
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기술진흥센터
연구사업명 SW컴퓨팅산업원천기술개발
연구과제명 [IITP]개인정보를 안전하고 편리하게 빅데이터 처리할 수 있는 방법
기 여 율 1/1
과제수행기관명 고려대학교산학협력단
연구기간 2019.01.01 ~ 2019.12.31

명 세 서

청구범위

청구항 1

사용자 단말, 데이터 소유자의 단말인 소유자 단말, 및 서버를 포함하는 유사도 검색 시스템에 있어서,

상기 서버는 상기 소유자 단말로부터 데이터(M)의 암호화된 데이터를 수신하고, 상기 사용자 단말로부터 트랩도어(trapdoor)를 포함하는 쿼리(query)를 수신하여 상기 사용자 단말의 데이터(M')와 상기 데이터(M)의 유사도를 판단하고, 유사도 판단 결과를 상기 사용자 단말로 송신하고,

상기 소유자 단말은 상기 사용자 단말과의 통신을 통해 상기 데이터(M)를 검색하기 위한 인덱스((δ, c))를 생성하고, 상기 인덱스((δ, c))를 상기 서버로 송신하고,

상기 사용자 단말은 상기 데이터(M')와 상기 데이터(M) 사이의 유사도를 판단하기 위한 트랩도어((t, t'))를 생성하고,

상기 쿼리는 상기 트랩도어((t, t'))를 포함하고,

상기 서버는 상기 인덱스((δ, c))와 상기 트랩도어((t, t'))를 이용하여 상기 데이터(M)와 상기 데이터(M')의 유사도를 판단하는,

유사도 검색 시스템.

청구항 2

제1항에 있어서,

상기 소유자 단말은 $l + 2$ (l 은 2 이상의 자연수) 개의 임의의 값들($r_1, ..., r_l, a, b \in \mathbb{Z}_p^*$)을 선택한 후 모든 $i \leq l$ 에 대하여 $\delta_{pre}^i = g_1^{f_i r_i ab}$ 와 $c_{pre}^i = g_2^{r_i}$ 를 계산하여 제1 인덱스 $(\delta_{pre}, c_{pre}) = \{\delta_{pre}^i, c_{pre}^i\}_{i \leq l}$)를 생성하고, 상기 제1 인덱스를 상기 사용자 단말로 송신하고,

상기 f_i 는 상기 데이터(M)로부터 추출된 서브특징(subfeature)이고,

상기 g_1 과 g_2 각각은 곱셈 순환군 $\mathbb{G}_1$ 과 $\mathbb{G}_2$ 의 생성원인,

유사도 검색 시스템.

청구항 3

제2항에 있어서,

상기 사용자 단말은 l 개의 임의의 값들($r_1', ..., r_l' \in$)을 선택한 후 모든 $i \leq l$ 에 대하여 $(\delta_{pre}^i)^{r_i' k_u k_u'} = g_1^{f_i r_i r_i' k_u k_u' ab}$ 와 $(c_{pre}^i)^{r_i'} = g_2^{r_i r_i'}$ 를 계산하여 제2 인덱스 $((\delta_{pre}^i)^{r_i' k_u k_u'}, (c_{pre}^i)^{r_i'})$ 를 생성하고, 상기 제2 인덱스를 상기 소유자 단말로 송신하고,

상기 k_u와 상기 k'_u는 상기 사용자 단말의 비밀키 쌍(secret key pair)인,

유사도 검색 시스템.

청구항 4

제3항에 있어서,

상기 소유자 단말은 상기 제2 인덱스로부터 상기 인덱스 $(\delta, c) = \{\delta^i = g_1^{f_i r_i r'_i k_u k'_u k_o^{-1}}, c^i = g_2^{r_i r'_i}\}_{i \le l}$)를 생성하고,

상기 k_o는 상기 소유자 단말의 비밀키인,

유사도 검색 시스템.

청구항 5

제4항에 있어서,

상기 사용자 단말은 l 개의 임의의 값들($b_1, ..., b_l \in \mathbb{Z}_p^*$)을 선택하고, 모든 $i \le l$에 대하여 $t^i = g_1^{f'_i k_u k'_u b_i}$와 $t^{i'} = g_2^{k_o b_i}$를 계산하여 상기 트랩도어($t^{i'} = g_2^{k_o b_i}$)를 생성하고,

상기 f'_i는 상기 데이터(M')로부터 추출된 서브특징인,

유사도 검색 시스템.

청구항 6

제5항에 있어서,

상기 서버는 모든 $i \le l$에 대하여 수학식의 성립 여부를 판단하고,

상기 수학식은 $e(\delta^i, t^{i'}) \stackrel{?}{=} e(t^i, c^i)$이고,

상기 서버는 상기 수학식이 성립한 횟수가 미리 정해진 임계값 이상인 경우 상기 데이터(M)와 상기 데이터(M')가 유사한 것으로 결정하는,

유사도 검색 시스템.

청구항 7

사용자 단말, 데이터 소유자의 단말인 소유자 단말, 및 서버를 포함하는 유사도 검색 시스템 상에서 수행되는 유사도 검색 방법에 있어서,

상기 서버가 상기 소유자 단말로부터 데이터(M)의 암호화된 데이터를 수신하는 단계;

상기 소유자 단말이 상기 사용자 단말과의 통신을 통해 상기 데이터(M)을 검색하기 위한 인덱스((δ, c))를 생성하고, 상기 인덱스((δ, c))를 상기 서버로 송신하는 단계;

상기 사용자 단말이 데이터(M')과 상기 데이터(M) 사이의 유사도 판단을 위한 트랩도어((t, t'))를 생성하고, 상

기 트랩도어를 포함하는 쿼리를 상기 서버로 송신하는 단계; 및

상기 서버가 상기 인덱스와 상기 트랩도어를 이용하여 상기 데이터(M)과 상기 데이터(M')의 유사도를 결정하는 단계를 포함하는,

유사도 검색 방법.

청구항 8

제7항에 있어서,

상기 인덱스를 상기 서버로 송신하는 단계는,

상기 소유자 단말이 $l+2$(l은 2 이상의 자연수) 개의 임의의 값들($r_1,...,r_l,a,b \in \mathbb{Z}_p^*$)을 선택한 후 모든 $i \leq l$에 대하여 $\delta_{pre}^i = g_1^{f_i r_i ab}$와 $c_{pre}^i = g_2^{r_i}$를 계산하여 제1 인덱스 ($(\delta_{pre}, c_{pre}) = \{\delta_{pre}^i, c_{pre}^i\}_{i \leq l}$)를 생성하고, 상기 제1 인덱스를 상기 사용자 단말로 송신하는 단계;

상기 사용자 단말이 l 개의 임의의 값들($r_1', ..., r_l' \in$)을 선택한 후 모든 $i \leq l$에 대하여 $(\delta_{pre}^i)^{r_i' k_u k_u'} = g_1^{f_i r_i r_i' k_u k_u' ab}$와 $(c_{pre}^i)^{r_i'} = g_2^{r_i r_i'}$를 계산하여 제2 인덱스 $((\delta_{pre}^i)^{r_i' k_u k_u'}, (c_{pre}^i)^{r_i'})$를 생성하고, 상기 제2 인덱스를 상기 소유자 단말로 송신하는 단계; 및

상기 소유자 단말이 상기 제2 인덱스로부터 상기 인덱스 ($(\delta, c) = \{\delta^i = g_1^{f_i r_i r_i' k_u k_u' k_o^{-1}}, c^i = g_2^{r_i r_i'}\}_{i \leq l}$)를 생성하는 단계를 포함하고,

상기 f_i는 상기 데이터(M)로부터 추출된 서브특징(subfeature)이고,

상기 k_u와 상기 k_u'는 상기 사용자 단말의 비밀키 쌍(secret key pair)이고,

상기 g_1과 g_2 각각은 곱셈 순환군 $\mathbb{G}_1$과 $\mathbb{G}_2$의 생성원이고,

상기 k_o는 상기 소유자 단말의 비밀키인,

유사도 검색 방법.

청구항 9

제8항에 있어서,

상기 쿼리를 상기 서버로 송신하는 단계는, 상기 사용자 단말이 l 개의 임의의 값들($b_1, ..., b_l \in \mathbb{Z}_p^*$)을 선택하고, 모든 $i \leq l$에 대하여 $t^i = g_1^{f_i' k_u k_u' b_i}$와 $t^{i'} = g_2^{k_o b_i}$를 계산하여 상기 트랩도어($t^{i'} = g_2^{k_o b_i}$)를 생성하는 단계를 포함하고,

상기 f_i'는 상기 데이터(M')로부터 추출된 서브특징인,

유사도 검색 방법.

청구항 10

제9항에 있어서,

상기 유사도를 결정하는 단계는, 상기 서버가 모든 $i \leq l$에 대하여 수학식의 성립 여부를 판단하는 단계를 포함하고,

상기 수학식은 $e(\delta^i, t^{i'}) \stackrel{?}{=} e(t^i, c^i)$이고,

상기 서버는 상기 수학식이 성립한 횟수가 미리 정해진 임계값 이상인 경우 상기 데이터(M)과 상기 데이터(M')가 유사한 것으로 결정하는,

유사도 검색 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 다수 개체를 위한 확장성 있고 안전한 유사도 검색 기법에 관한 것이다.

배 경 기 술

[0003] 데이터 소유자들(data owners)이 다른 사용자들과 데이터를 공유하기 위해 데이터를 서버에 위탁할 때, 데이터 프라이버시(data privacy) 문제가 제기될 수 있다. 위탁 전에 데이터를 암호화하는 것이 효과적이지만, 사용자들이 관심 있는 데이터를 검색하는 것과 같은 몇몇 중요한 기능을 복잡하게 만들다. 이에 대한 해결책으로, 안전한 유사도 검색(secure similarity search)은 단일 사용자로 하여금 암호화된 데이터베이스(encrypted databases)로부터 유사한 데이터를 검색할 수 있도록 한다.

[0004] Cui 등은 MKSE(multi-key searchable encryption)를 이용하여 확장성 있는 방식으로 다수 개체를 서포트하는 방법을 제안하였다(비특허문헌 [15], [24] 참조). MKSE는 데이터 소유자들과 사용자들의 각각 데이터와 쿼리들(queries)을 암호화하는 경우라도, 개별적으로 생성한 상이한 키들을 이용하여 데이터 검색을 가능케 한다. 따라서, 쿼리를 통해 상이한 데이터 소유자들에게 종속되는 데이터를 검색할 수 있다. 그러나, 이러한 기능은 데이터 소유자들이 유사한 데이터 아이템들을 공유하고 있다는 것을 유출시키는 문제점이 있다.

[0005] 게다가, 멀티-유저 환경에서의 데이터 검색은 공모 공격(collusion attack)에 취약하다. 공모 공격에서, 몇몇 사용자들과 공모하는 서버는 다른 사용자들의 쿼리를 유출할 수 있다. 최근, Hamlin 등은 공모 공격을 방지하기 위한 방법을 제안하였다(비특허문헌 [26] 참조). 그러나, Hamlin 등의 접근법은 선택적인 허가받지 않은 검색 문제(selective unauthorized search problem)를 초래할 수 있음이 관찰되었다.

[0006] 따라서, 본 명세서에서는 다수 개체를 위한 확장성 있고 안전한 유사도 검색 기법을 제안하고자 한다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) 대한민국 공개특허 제2019-0063204호 (2019.06.07. 공개)

(특허문헌 0002) 대한민국 공개특허 제2009-0031079호 (2009.03.25. 공개)

(특허문헌 0003) 미합중국 공개특허 US 2018/0157703 A1 (2018.06.07. 공개)

비특허문헌

[0009] (비특허문헌 0001) [1] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE InternetComputing, vol. 16, no. 1, pp. 69-73, 2012.

(비특허문헌 0002) [2] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the

Masses," Computer, vol. 45, no. 1, pp. 39-45, 2012.

(비특허문헌 0003) [3] Bosch, C., Hartel, P., Jonker, W., and Peter, A., "A survey of provably secure searchable encryption", ACM Computing Surveys (CSUR), 47(2), 2014.

(비특허문헌 0004) [4] Zhang, Y., Katz, J., and Papamanthou, C., "All your queries are belong to us: the power of file-injection attacks on searchable encryption", In Usenix Security, (2016)

(비특허문헌 0005) [5] Boneh, D., Di Crescenzo, G., Ostrovsky, R., and Persiano, G., "Public key encryption with keyword search", In International Conference on the Theory and Applications of Cryptographic Techniques, pp. 506-522, 2004.

(비특허문헌 0006) [6] Song, D. X., Wagner, D., and Perrig, A., "Practical techniques for searches on encrypted data", In Security and Privacy, IEEE Symposium on, pp. 44-55, 2000.

(비특허문헌 0007) [7] Kiayias, A., Oksuz, O., Russell, A., Tang, Q., and Wang, B., "Efficient encrypted keyword search for multi-user data sharing," In European Symposium on Research in Computer Security, pp. 173-195, 2016.

(비특허문헌 0008) [8] Sun, S. F., Liu, J. K., Sakzad, A., Steinfeld, R., and Yuen, T. H., "An efficient non-interactive multi-client searchable encryption with support for boolean queries," In European Symposium on Research in Computer Security, pp. 154-172, 2016.

(비특허문헌 0009) [9] Patel, S., Persiano, G., and Yeo, K., "Symmetric searchable encryption with sharing and unsharing," In European Symposium on Research in Computer Security, pp. 207-227, 2018.

(비특허문헌 0010) [10] Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., and Lou, W., "Fuzzy keyword search over encrypted data in cloud," In INFOCOM, 2010 Proceedings IEEE, pp. 1-5, (2010)

(비특허문헌 0011) [11] Kuzu, M., Islam, M. S., and Kantarcioglu, M., "Efficient similarity search over encrypted data", In Data Engineering (ICDE), 2012 IEEE 28th International Conference on, pp. 1156-1167, 2012.

(비특허문헌 0012) [12] X. Yuan, X.Wang, C.Wang, A. Squicciarini, and K. Ren, "Enabling privacy-preserving image-centric social discovery," In Distributed Computing Systems (ICDCS), pp. 198-207, 2014.

(비특허문헌 0013) [13] X. Yuan, X. Wang, C. Wang, and C. Yu, "Privacy-Preserving Similarity Joins Over Encrypted Data," IEEE Transactions on Information Forensics and Security, pp. 2763-2775, 2017.

(비특허문헌 0014) [14] Strizhov, M. and Ray, I. "Multi-keyword similarity search over encrypted cloud data," In IFIP International Information Security Conference, pp. 52-65, Springer, Berlin, Heidelberg, 2014.

(비특허문헌 0015) [15] H. Cui, X. Yuan, Y. Zheng, and C. Wang, "Enabling secure and effective near-duplicate detection over encrypted in-network storage," In the 35th International Conference on Computer Communications (INFOCOM), 2016.

(비특허문헌 0016) [16] http://lear.inrialpes.fr/people/jegou/data.php#copydays

(비특허문헌 0017) [17] Y. Ke, R. Sukthankar, L. Huston, Y. Ke, and R. Sukthankar, "Efficient near-duplicate detection and sub-image retrieval," In ACM Multimedia, 2004.

(비특허문헌 0018) [18] Gionis, A., Indyk, P., and Motwani, R., "Similarity search in high dimensions via hashing" In VLDB, pp. 518-529, 1999.

(비특허문헌 0019) [19] Python Software Foundation, https://pypi.python.org/pypi/ImageHash.

(비특허문헌 0020) [20] Garg, S., Mohassel, P., and Papamanthou, C, "TWORAM: Round-Optimal Oblivious RAM with Applications to Searchable Encryption," IACR Cryptology ePrint Archive, 2015, 1010.

(비특허문헌 0021) [21] Stefanov, E., Papamanthou, C., and Shi, E, "Practical Dynamic Searchable

Encryption with Small Leakage," In NDSS, Vol. 14, pp. 23-26, 2014.

(비특허문헌 0022) [22] Bost, R. "Forward Secure Searchable Encryption," In ACM CCS, Vol. 16, pp. 1143-1154, 2016.

(비특허문헌 0023) [23] Bost, R., Minaud, B., and Ohrimenko, O, "Forward and backward private searchable encryption from constrained cryptographic primitives," In ACM CCS, 2017.

(비특허문헌 0024) [24] Popa, R. A. and Zeldovich, N., "Multi-Key Searchable Encryption," IACR Cryptology ePrint Archive, 2013.

(비특허문헌 0025) [25] Grubbs, P., McPherson, R., Naveed, M., Ristenpart, T., and Shmatikov, V., "Breaking web applications built on top of encrypted data," In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1353-1364, 2016.

(비특허문헌 0026) [26] Hamlin, A., Shelat, A., Weiss, M., and Wichs, D., "Multi-Key Searchable Encryption, Revisited," In IACR International Workshop on Public Key Cryptography, pp. 95-124, 2018.

(비특허문헌 0027) [27] Van Rompay, C., Molva, R., and Onen, M., "A leakage-abuse attack against multi-user searchable encryption," Proceedings on Privacy Enhancing Technologies, pp. 168-178, 2017.

(비특허문헌 0028) [28] Cui, B., Liu, Z., and Wang, L., "Key-aggregate searchable encryption (KASE) for group data sharing via cloud storage," IEEE Transactions on computers, 65(8), pp. 2374-2385, 2016.

(비특허문헌 0029) [29] Liu, Z., Li, T., Li, P., Jia, C., and Li, J., "Verifiable searchable encryption with aggregate keys for data sharing system," Future Generation Computer Systems, 78, pp. 778-788, 2018.

(비특허문헌 0030) [30] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," Computers and Communications (ISCC), IEEE Symposium on, pp. 850-855, 2011.

(비특허문헌 0031) [31] Uzunkol, O. and Kiraz, M. S., "Still wrong use of pairings in cryptography," Applied Mathematics and Computation, 333, pp. 467-479, 2011, 2018.

(비특허문헌 0032) [32] Stein, B. and Zu Eissen, S. M., "Near similarity search and plagiarism analysis," In From data and information analysis to knowledge engineering, pp. 430-437, Springer, Berlin, Heidelberg, 2006.

(비특허문헌 0033) [33] Lv, Q., Josephson, W., Wang, Z., Charikar, M., and Li, K., "Multiprobe LSH: efficient indexing for high-dimensional similarity search," In Proceedings of the 33rd international conference on Very large data bases, pp. 950-961, VLDB Endowment, 2007.

(비특허문헌 0034) [34] Enron email dataset, https://www.cs.cmu.edu/ ./enron/. Accessed: 2019-05-30.

발명의 내용

해결하려는 과제

[0010] 본 발명이 이루고자 하는 기술적인 과제는 다수 개체를 위한 확장성 있고 안전한 유사도 검색 장치 및 방법을 제공하는 것이다.

과제의 해결 수단

[0012] 본 발명의 일 실시예에 따른 유사도 검색 시스템은 사용자 단말, 데이터 소유자의 단말인 소유자 단말, 및 서버를 포함하고, 상기 서버는 상기 소유자 단말로부터 데이터(M)의 암호화된 데이터를 수신하고, 상기 사용자 단말로부터 트랩도어(trapdoor)를 포함하는 쿼리(query)를 수신하여 상기 사용자 단말의 데이터(M')와 상기 데이터

(M)의 유사도를 판단하고, 유사도 판단 결과를 상기 사용자 단말로 송신하고, 상기 소유자 단말은 상기 사용자 단말과의 통신을 통해 상기 데이터(M)를 검색하기 위한 인덱스를 생성하고, 상기 인덱스((δ, c))를 상기 서버로 송신하고, 상기 사용자 단말은 상기 데이터(M')와 상기 데이터(M) 사이의 유사도를 판단하기 위한 트랩도어를 생성하고, 상기 쿼리는 상기 트랩도어를 포함하고, 상기 서버는 상기 인덱스와 상기 트랩도어를 이용하여 상기 데이터(M)와 상기 데이터(M')의 유사도를 판단한다.

[0013] 본 발명의 일 실시예에 따른 유사도 검색 방법은 사용자 단말, 데이터 소유자의 단말인 소유자 단말, 및 서버를 포함하는 유사도 검색 시스템 상에서 수행되고, 상기 서버가 상기 소유자 단말로부터 데이터(M)의 암호화된 데이터를 수신하는 단계, 상기 소유자 단말이 상기 사용자 단말과의 통신을 통해 상기 데이터(M)을 검색하기 위한 인덱스를 생성하고, 상기 인덱스를 상기 서버로 송신하는 단계, 상기 사용자 단말이 데이터(M')과 상기 데이터(M) 사이의 유사도 판단을 위한 트랩도어를 생성하고, 상기 트랩도어를 포함하는 쿼리를 상기 서버로 송신하는 단계, 및 상기 서버가 상기 인덱스와 상기 트랩도어를 이용하여 상기 데이터(M)과 상기 데이터(M')의 유사도를 결정하는 단계를 포함한다.

발명의 효과

[0015] 본 발명의 실시예에 따른 유사도 검색 기법에 의할 경우, 데이터 소유자의 프라이버시를 보장하면서도 확장성 있고 안전한 유사도 검색을 가능케 하는 효과가 있다.

도면의 간단한 설명

[0017] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.

도 1은 본 명세서에서 사용되는 기호의 설명을 나타내는 표를 도시한다.

도 2는 본 발명의 일 실시예에 따른 유사도 검색 시스템을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0018] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시예들에 대해서 특정한 구조적 또는 기능적 설명들은 단지 본 발명의 개념에 따른 실시예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시예들은 다양한 형태들로 실시될 수 있으며 본 명세서에 설명된 실시예들에 한정되지 않는다.

[0019] 본 발명의 개념에 따른 실시예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물, 또는 대체물을 포함한다.

[0020] 제1 또는 제2 등의 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만, 예컨대 본 발명의 개념에 따른 권리 범위로부터 벗어나지 않은 채, 제1 구성 요소는 제2 구성 요소로 명명될 수 있고 유사하게 제2 구성 요소는 제1 구성 요소로도 명명될 수 있다.

[0021] 어떤 구성 요소가 다른 구성 요소에 '연결되어 있다'거나 '접속되어 있다'고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성 요소가 다른 구성 요소에 '직접 연결되어 있다'거나 '직접 접속되어 있다'고 언급된 때에는 중간에 다른 구성 요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성 요소들 간의 관계를 설명하는 다른 표현들, 즉 '~사이에'와 '바로 ~사이에' 또는 '~에 이웃하는'과 '~에 직접 이웃하는' 등도 마찬가지로 해석되어야 한다.

[0022] 본 명세서에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로서, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, '포함하다' 또는 '가지다' 등의 용어는 본 명세서에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구

성 요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0023] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0024] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시예들을 상세히 설명한다. 그러나, 특허출원의 범위가 이러한 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.

[0026] 이하에서는, 본 발명에서 이용되는 배경 지식에 대하여 설명한다.

[0027] LSH(Locality Sensitive Hashing)

[0028] LSH는 상이한 아이템들 사이의 유사도(similarity)를 측정하기 위해 데이터로부터 식별 가능한 특징들(identifiable features)을 추출하는 근사 알고리즘(approximation algorithm)이다(비특허문헌 [18] 참조). LSH의 기본적인 아이디어는 높은 확률로 유사한 아이템들을 동일한 값들로 매핑시키는 해쉬 함수들(hash functions)을 이용하여 고차원의 데이터(high-dimensional data)의 차원(dimensionality)을 감소시키는 것이다. 데이터 아이템에 대하여, LSH 함수는 서브특징 집합(subfeature set) $\{f_1, ..., f_l\}$를 추출한다. 그런 다음, 두 개의 아이템들의 특징 집합들 간의 유사도를 측정하기 위해 동일성 테스트(equality testing)가 수행되고, 이 과정은 LSH-기반 유사도 검색의 검색 정확도(search accuracy)를 결정한다. 본 제안 기법에서는, 데이터 특징들을 추출하기 위하여 사용자들과 데이터 소유자들은 기존의 LSH 함수들을 사용하는 것으로 가정한다(비특허문헌 [18], [19] 참조). 그러나, 본 발명의 권리범위가 LSH 함수들의 종류나 유형에 제한되는 것이 아님은 명백하다.

[0030] LSH를 이용한 유사도 검색

[0031] 유사도 검색의 핵심 아이디어는 LSH 매치들(LSH matches)의 개수가 미리 정의된 임계값을 초과하는지 여부를 체크하는 것이다. 검색 가능한 인텍스들(searchable indices)의 집합으로써 $\{f_i | f_i := LSH(M)\}_{i \in \{1,...,l\}}$가 주어지고, 트랩도어들(trapdoors)의 집합으로써 $\{f'_i | f'_i := LSH(M')\}_{i \in \{1,...,l\}}$가 주어졌을 때, M과 M'이 유사한지 여부를 결정할 수 있다. 이를 위해, 카운터 변수(counter variable) $count$가 0으로 초기화되고, 모든 $i \in \{1, ..., l\}$에 대하여 $f_i = f'_i$이 성립하면 $count$가 1 만큼 증가된다. 결국, 미리 결정된 임계값 th에 대하여 $count \geq th$이 성립하면, M과 M'은 유사한 것으로 고려된다. 즉, 인덱스와 인덱스와 동일한 값을 갖는 트랩도어의 개수가 임계값 th보다 크거나 같은 경우, M과 M'은 유사한 것으로 결정될 수 있다.

[0032] 키워드들을 LSH 값들과 교체함으로써 멀티-키워드 검색(multi-keyword searches)을 사용한 유사도 검색이 가능함이 보고되었다(비특허문헌 [15] 참조). 또한, 그 역도 마찬가지다. 따라서, 제안 기법은 키워드 검색에도 적용이 가능하다.

[0034] Bilinear Maps(겹선형 함수)

[0035] $\mathbb{G}_1$, $\mathbb{G}_2$, 및 $\mathbb{G}_T$를 소수 위수(prime order) p의 곱셈 순환 군들(multiplicative cyclic groups)이라 하고, $\mathbb{Z}_p^*$를 곱셈 모듈로(multiplication modulo) p 하의 군(group)이라 하자. 또한, $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$를 다음 특성을 갖는 겹선형 함수(bilinear maps)라 하자.

[0036] · Bilinearty: 모든 $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, 및 $a, b \in \mathbb{Z}_p^*$에 대하여 $e(P^a, Q^b) = e(P, Q)^{ab}$이 성립한다.

[0037] · Nondegeneracy: g_1과 g_2 각각을 $\mathbb{G}_1$과 $\mathbb{G}_2$의 생성원이라 할 때, $e(g_1, g_2) \neq 1$이 성립한다.

[0038] $\mathbb{G}_1$과 $\mathbb{G}_2$ 내에서의 그룹 연산과 겹선형 함수 $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$가 효율적으로 계산가능하다면 $\mathbb{G}_1$과 $\mathbb{G}_2$를 겹선형 그룹(bilinear group)이라 한다.

[0040] Algorithm Definitions of Similarity Search(유사도 검색 알고리즘 정의)

[0041] 본 명세서에서 사용되는 기호에 대한 설명은 도 1에 도시된 표와 같고, 유사고 검색 알고리즘은 다음과 같이 구성된다.

[0042] 1) $BuildIndex(k_u, k'_u, k_o, f_1, ..., f_l) \rightarrow (\delta, c)$. 인덱스 생성 알고리즘(index generation algorithm, 인덱스 생성 단계로 명명될 수도 있음)은 사용자 $u \in U_o$와 소유자 O에 의해 수행되는 인터랙티브 알고리즘(interactive algorithm, 대화식 알고리즘)이다. 사용자의 비밀키(secret key) (k_u, k'_u), 소유자의 비밀키 k_o, 및 M으로부터 추출된 서브특징들(subfeatures) $f_1, ..., f_l$을 입력으로 받고, 암호화된 검색 가능한 인덱스(encrypted searchable index) (δ, c)를 출력한다.

[0043] 2) $Trapdoor(k_u, k'_u, g_2^{k_o}, f'_1, ..., f'_l) \rightarrow (t, t')$. 트랩도어 생성 알고리즘(trapdoor generation algorithm, 트랩도어 생성 단계로 명명될 수도 있음)은 사용자의 비밀키 (k_u, k'_u), 소유자의 공개키(public key) $g_2^{k_o}$, 사용자에 의해 소유되는 M'으로부터 추출된 서브특징들 $f'_1, ..., f'_l$을 입력으로 받고, 트랩도어 (t, t')를 출력한다.

[0044] 3) $Search((\delta, c), (t, t'), th) \rightarrow (true \text{ or } \emptyset)$. 유사도 검색 알고리즘(similarity search algorithm, 유사도 검색 단계로 명명될 수도 있음)은 (δ, c), (t, t'), 및 유사도 임계치 th를 입력으로 받고, th를 이용하여 M과 M' 사이의 유사도를 테스트한다. M과 M'이 유사하면 true를 출력하고, M과 M'이 매치되지 않으면 공집합(empty set) $\emptyset$를 출력한다. 즉, 유사도 검색 알고리즘은 소정의 입력을 이용하여, 데이터 M과 데이터 M'의 유사 여부를 판단한 후 판단 결과를 출력할 수 있다.

[0046] 도 2는 본 발명의 일 실시예에 따른 유사도 검색 시스템을 도시한다.

[0047] 도 2를 참조하면, 제안하는 유사도 검색 시스템은 데이터 소유자들(data owners, 데이터 소유자들 각각의 단말을 의미할 수 있음), 사용자들(users, 사용자들 각각의 단말을 의미할 수 있음), 및 서버(유사도 검색 서버라 명명될 수도 있음)로 구성된다. 각 엔티티의 역할은 다음과 같다.

[0048] 1) Data owners(데이터 소유자들). 데이터 소유자들은 공개적으로 접근 가능한 서버로 업로드 되는 데이터의 소유자들이다. 소유자 O는 데이터 셋(data set) D_o를 소유한다. 각 소유자는 선택적으로 자신의 데이터를 자신이 선택한 특정 사용자들과 공유할 수 있다. 소유자는 사용자와 대화식 공유 프로세스(interactive sharing process)를 수행한 후 오직 정해진 사용자만이 검색을 수행할 수 있는 검색 가능한 인덱스들(searchable indices)을 출력한다.

[0049] 2) Users(사용자들). 사용자들은 검색 쿼리들(search queries)을 생성하는 클라이언트들이다. 사용자들은 검색 가능한 인덱스들을 생성하기 위해 소유자들과 상호 작용한다. 또한, 사용자들은 프라이버시를 보호하기 위하여 비밀키들과 특정 소유자의 공개키 하에서 자신의 평문 쿼리 데이터(plain query data)를 암호화한다.

[0050] 3) Server(서버). 서버는 복수의 데이터 소유자들에 의해 업로드된 데이터를 저장하는 스토리지 서비스 제공자

(storage service provider)이다. 서버는 n 명의 소유자들로부터 수신된 암호화된 코퍼스(corpus) $D = \{D_1, \cdots, D_n\}$를 소유한다. 또한, 서버는 암호화된 데이터 상에서 사용자들로부터 검색 쿼리들을 수신하여 유사도 검색을 수행한다. 유사도 검색 수행 결과는 쿼리를 전송한 사용자에게 전송될 수 있다.

[0051] 상술한 소유자들, 사용자들, 및 서버 각각은 각종 연산 처리 및 신호 생성이 가능한 적어도 하나의 전자 장치를 포함할 수 있다. 여기서, 적어도 하나의 전자 장치는 프로세서 및/또는 프로세서가 설치된 컴퓨팅 장치를 포함할 수 있으며, 상술한 동작들 중 적어도 일부는 상기 프로세서의 동작을 의미할 수 있다. 여기서, 프로세서는 중앙 처리 장치(CPU, Central Processing Unit), 마이크로 컨트롤러 유닛(MCU, Micro Controller Unit), 마이컴(Micom, Micro Processor), 애플리케이션 프로세서(AP, Application Processor), 전자 제어 유닛(ECU, Electronic Controlling Unit), 그래픽 처리 장치(GPU, Graphic Processing Unit) 및/또는 각종 연산 처리 및 제어 신호의 생성이 가능한 처리 장치 등을 포함할 수 있다. 이들 처리 장치는 예를 들어 하나 또는 둘 이상의 반도체 칩 및 관련 부품을 이용하여 구현될 수 있다. 또한, 컴퓨팅 장치는, 예를 들어, 데스크톱 컴퓨터, 랩톱 컴퓨터, 서버용 컴퓨터, 스마트 폰, 태블릿 피씨, 스마트 시계, 두부 장착형 디스플레이(HMD, Head Mounted Display) 장치, 휴대용 게임기, 내비게이션 장치, 개인용 디지털 보조기(PDA, Personal Digital Assistant), 인공지능 스피커 장치, 디지털 텔레비전, 셋톱 박스, 로봇, 가전 기기, 기계 장치 및/또는 이외 정보 처리 기능을 수행할 수 있는 적어도 하나의 전자 장치를 포함할 수 있다.

[0053] 이하에서는, 제안하는 유사도 검색 기법을 상세하게 설명한다. 우선, $\mathsf{GroupGen}$을 보안 상수 1^λ을 입력으로 받아 소수 $p = \Theta(\lambda)$를 위수로 하고 겹선형 함수 $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$를 구비하는 3 개의 그룹들(곱셈 순환군) $\mathbb{G}_1$, $\mathbb{G}_2$, 및 $\mathbb{G}_T$를 생성하는 그룹 생성기(symmetric pairing group generator)라 하자. $\mathsf{GroupGen}$는 또한 $\mathbb{G}_1$과 $\mathbb{G}_2$ 각각의 생성자 g_1과 g_2를 출력한다. $H : \{0,1\}^\lambda \rightarrow \mathbb{Z}_p^*$를 공모-저항 해쉬 함수(collision-resistant hash function)이라 하자. 또한, M으로부터 추출된 서브특징들의 집합 $\{f_1, \cdots, f_l\}$은 $f_i := H(LSH(M))$와 같이 생성되며, 여기서 $f_i = 0$가 성립할 확률은 무시할 수 있으며, l은 2이상의 자연수일 수 있다. $F : \{0,1\}^\lambda \times \{0,1\}^\lambda \rightarrow \mathbb{Z}_p^*$를 수도 랜덤 함수(pseudo-random function)이라 하자. 소유자 O는 비밀키 $k_o = F(K_o, seed)$와 공개키 $g_2^{k_o}$를 유도하기 위한 마스터키(master key) $K_o \xleftarrow{R} \{0,1\}^\lambda$를 갖는다. 여기서, $seed$는 랜덤하게 샘플링된 씨드(seed)이다. 사용자 $u \in U_o$는 비밀키 쌍(secret key pair) $(k_u = F(K_u, seed), k'_u = F(K_u, seed'))$와 $1, ..., l$의 재배치된 순서(permuted order) ord를 유도하기 위한 마스터키 $K_u \xleftarrow{R} \{0,1\}^\lambda$를 갖는다. 여기서, $seed$와 $seed'$는 랜덤하게 샘플링된 씨드들(seeds)이다. 소유자는 유사도 임계치 th를 정의하고 이를 서버로 송신한다. 또한, 상술한 변수, 파라미터들 중 연산과정에 필요한 변수나 파라미터들은 사전에 엔티티 사이에서 공유될 수 있다. 구체적인 알고리즘은 다음과 같다.

[0054] 1) $BuildIndex(k_u, k'_u, k_o, f_1, ..., f_l) \rightarrow (\delta, c)$. 검색 가능한 인덱스 생성 알고리즘(searchable index generation algorithm, 검색 가능한 인덱스 생성 단계로 명명될 수도 있음)은 사용자와 소유자의 상호 작용으로 수행된다. 알고리즘의 구체적인 내용은 다음과 같다.

[0055] a) 소유자는 $l + 2$ 개의 임의의 값들 $r_1, ..., r_l, a, b \in \mathbb{Z}_p^*$을 선택한다. 모든 $i \leq l$에 대하여, 소유자는 $\delta_{pre}^i = g_1^{f_i r_i ab}$와 $c_{pre}^i = g_2^{r_i}$를 계산한다. 소유자는 $(\delta_{pre}, c_{pre}) = \{\delta_{pre}^i, c_{pre}^i\}_{i \leq l}$를 사용자에게 전송한다. 여기서, (δ_{pre}, c_{pre})를 인덱스 (δ, c)와 구분하기 위하여 제1 인덱스로 명명할 수 있다.

[0056] b) 사용자는 l 개의 임의 값들 $r'_1, ..., r'_l \in \mathbb{Z}_p^*$를 선택한다. 모든 $i \leq l$에 대하여, 사용자는

$(\delta_{pre}^{i})^{r_i' k_u k_u'} = g_1^{f_i r_i r_i' k_u k_u' ab}$와 $(c_{pre}^{i})^{r_i'} = g_2^{r_i r_i'}$를 계산한다. 사용자는 ord에 따라 l개의 쌍들 $((\delta_{pre}^{i})^{r_i' k_u k_u'}, (c_{pre}^{i})^{r_i'})$을 재배치(shuffle)한다. 그리고, 사용자는 재배치 결과를 소유자에게 반환한다. ord에 따라 재배치된 l개의 쌍들 $((\delta_{pre}^{i})^{r_i' k_u k_u'}, (c_{pre}^{i})^{r_i'})$을 인덱스 (δ, c)와 구분하기 위하여 제2 인덱스로 명명할 수 있다.

[0057] c) 소유자는 $(\delta_{pre}^{i})^{r_i' k_u k_u'}$로부터 a와 b를 제거한다. 결과는 k_o^{-1}의 거듭제곱이 된다. 소유자는 $(\delta, c) = \{\delta^i = g_1^{f_i r_i r_i' k_u k_u' k_o^{-1}}, c^i = g_2^{r_i r_i'}\}_{i \le l}$를 서버로 업로드한다. 즉, 소유자는 사용자로부터 수신한 제2 인덱스로부터 인덱스 (δ, c)를 생성하고, 생성된 인덱스 (δ, c)를 서버로 송신할 수 있다. 물론, 소유자의 데이터 M은 사전에 암호화되어 미리 서버로 업로드되어 저장되어 있을 수 있다.

[0058] 2) $Trapdoor(k_u, k_u', g_2^{k_o}, f_1', ..., f_l') \rightarrow (t, t')$. 사용자는 l개의 임의의 값들 $b_1, ..., b_l \in \mathbb{Z}_p^*$를 선택한다. 모든 $i \le l$에 대하여, 사용자는 $t^i = g_1^{f_i' k_u k_u' b_i}$와 $t^{i'} = g_2^{k_o b_i}$를 계산한다. 사용자는 ord에 따라 $(t^1, t^{1'}), ..., (t^l, t^{l'})$를 재배치하고, 재배치된 트랩도어 $(t, t') = \{t^i, t^{i'}\}_{i \le l}$를 서버로 전송한다.

[0059] 3) $Search((\delta, c), (t, t'), th) \rightarrow (true \text{ or } \emptyset)$. 서버는 임시 변수 $count$를 0으로 설정한다. 모든 $i \le l$에 대하여, 서버는 $e(\delta^i, t^{i'}) \stackrel{?}{=} e(t^i, c^i)$의 성립 여부를 체크한다. 위 수학식이 성립하는 경우, 서버는 $count$를 1 만큼 증가시킨다. 결국, $count \ge th$이 성립하면 서버는 $true$를 반환하고, 그렇지 않으면 M과 M'이 유사하지 않음을 의미하는 $\emptyset$를 반환한다. 즉, 서버는 유사도 검색 결과(유사 또는 비유사)를 쿼리를 송신한 사용자에게 전송할 수 있다.

[0061] Correctness(정확성)

[0062] 상술한 제안 기법의 검색 정확성은 아래와 같이 증명될 수 있다. 2 개의 i번째 서브특징들 f_i, f_i'이 동일하다고 가정하면, 주어진 i번째 검색 가능한 인덱스 (δ^i, c^i)와 i번째 트랩도어 $(t^i, t^{i'})$에 대하여 검색 알고리즘은 다음을 계산한다.

[0063]
$$\begin{aligned} e(\delta^i, t^{i'}) &= e(g_1^{f_i r_i r_i' k_u k_u' k_o^{-1}}, g_2^{k_o b_i}) \\ &= e(g_1, g_2)^{f_i r_i r_i' k_u k_u' b_i} \\ &= e(g_1^{f_i k_u k_u' b_i}, g_2^{r_i r_i'}) \\ &= e(t^i, c^i). \end{aligned}$$

[0065] Efficient revocation(효율적인 폐기)

[0066] 소유자 O가 소유하는 데이터 아이템 M에 대한 사용자 $u \in U_o$의 검색 능력을 폐기하기 위하여, 소유자는 M에 대응하는 (δ, c)를 제거할 것을 요청할 수 있다. (δ, c)가 제거되었지만, 사용자 u는 여전히 M에 대한 유효한 트랩도어를 생성할 수 있다. 그러나, (δ, c)가 제거되었기 때문에 트랩도어는 M에 대한 유사도 테스트를 위해 사용될 수 없다. (δ, c)의 제거는 사용자 집합 U_o에 포함되는 다른 사용자들에게 영향을 미치지 않으며, 이는 다른

사용자들은 그들의 (δ, c)가 유효한 동안 검색을 통해 M을 복구할 수 있음을 의미한다.

[0068] Accelerating search via pre-computation(사전 연산을 통한 검색의 가속)

[0069] 제안 기법의 중요한 특징은 트랩도어 생성과 검색 시간은 사전 연산(pre-computation)을 통하여 급격하게 감소될 수 있다는 것이다. 우선, 트랩도을 사전 연산하는 방법을 기술한다. 트랩도어는 2 개의 파트로 구성된다. 첫 번째 파트는 쿼리 데이터 암호화 $\{t^1, ..., t^l\}$이며, 두번째 파트는 소유자의 공개키 $\{t^{1'}, ..., t^{l'}\}$이다. 여기서, 두 개의 파트는 임의의 변수 $\{b_1, ..., b_l\}$를 이용하여 연관되어 있다. 트랩도어 생성 시간을 가속하기 위하여, 사용자는 $\{b_1, ..., b_l\}$를 선택하고 $\{t^{1'}, ..., t^{l'}\}$를 미리 연산할 수 있다. 사용자는 $\{b_1, ..., b_l\}$와 $\{t^{1'}, ..., t^{l'}\}$를 자신의 로컬 스토리지(사용자 또는 사용자 단말에 포함되어 있는 저장장치(또는 저장부)를 의미함)에 저장할 수 있다. 이후에, 사용자가 어떤 데이터를 검색하고자 할 때, 사용자는 $\{t^1, ..., t^l\}$을 계산하기 위하여 미리 저장되어 있는 $\{b_1, ..., b_l\}$를 이용할 수 있으며, 연산 결과로 생성된 $\{t^1, ..., t^l\}$와 미리 연산된 값인 $\{t^{1'}, ..., t^{l'}\}$를 트랩도어로써 서버로 송신할 수 있다. 이와 같은 사전 연산은 트랩도어 생성 시간을 약 89% 감소시킬 수 있다.

[0070] 다음으로, 검색 시간을 감소시키는 방법을 기술한다. 이를 위해, 사용자는 소유자의 공개키 부분 $\{t^{1'}, ..., t^{l'}\}$을 사전 연산하고 이를 미리 서버로 전송할 수 있다. 서버가 $\{t^{1'}, ..., t^{l'}\}$를 수신함에 따라, 서버는 유사도 검색식의 좌변인 $e(\delta^i, t^{i'})$를 사전 연산하고, 이를 자신의 로컬 스토리지(서버에 포함되어 있는 저장장치(또는 저장부)를 의미함)에 저장할 수 있다. 후에, 사용자가 $\{t^1, ..., t^l\}$로 구성된 트랩도어를 서버로 전송하면, 서버는 유사도 검색식의 우변인 $e(t^i, c^i)$를 연산할 수 있다. 서버는 미리 연산되어 저장되어 있는 값인 $e(\delta^i, t^{i'})$와 $e(t^i, c^i)$를 비교함으로서 유사도 검색을 수행할 수 있다. 실험에 의하면, 검색을 수행하기 위해 필요한 전체 동작의 약 50%가 미리 계산될 수 있다.

[0072] 이상에서 설명된 장치는 하드웨어 구성 요소, 소프트웨어 구성 요소, 및/또는 하드웨어 구성 요소 및 소프트웨어 구성 요소의 집합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성 요소는, 예를 들어, 프로세서, 콘트롤러, ALU(Arithmetic Logic Unit), 디지털 신호 프로세서(Digital Signal Processor), 마이크로컴퓨터, FPA(Field Programmable array), PLU(Programmable Logic Unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(Operation System, OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술 분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(Processing Element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(Parallel Processor)와 같은, 다른 처리 구성(Processing Configuration)도 가능하다.

[0073] 소프트웨어는 컴퓨터 프로그램(Computer Program), 코드(Code), 명령(Instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(Collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성 요소(Component), 물리적 장치, 가상 장치(Virtual Equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(Signal Wave)에 영구적으로, 또는 일시적으로 구체화(Embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매

체에 저장될 수 있다.

[0074] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM, DVD와 같은 광기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-optical Media), 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

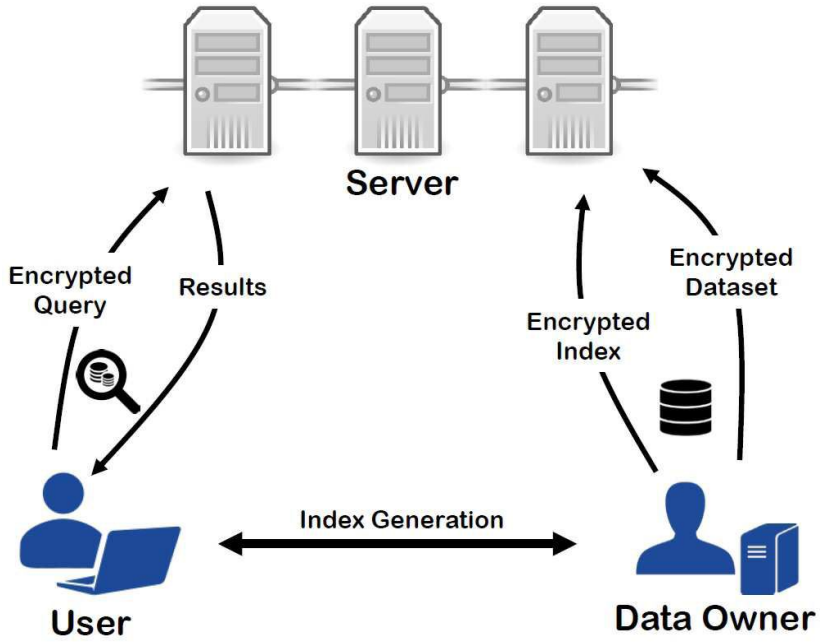
[0076] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성 요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

도면

도면1

λ	Security parameter used in the setup phase
M	Owner's data needed to generate a searchable index
M'	User's data needed to generate a trapdoor
$LSH(\cdot)$	Locality sensitive hash function which, given data, returns subfeatures
l	Number of subfeatures
f_i	i-th subfeature
O	A set of data owners
$o \in O$	owner who belongs to O
D_o	A data set owned by o
U_o	A set of users who can access the data owner o's data
Q_o	A set of queries made for o
$u \in U_o$	User who belongs to U_o
(δ, c)	Searchable index corresponding to M, created by the data owner
(t, t')	Trapdoor corresponding to M', created by the user
th	Threshold which determines the similarity criterion for the two files

도면2


Server
Encrypted Query
Results
Encrypted Index
Encrypted Dataset
Index Generation
User
Data Owner