RESEARCH ARTICLE | SEPTEMBER 06 2017

Encrypted data inquiries using chained perfect hashing (CPH) FREE

Khalid Kaabneh; Hassan Tarawneh; Issam Alhadid

AIP Conf. Proc. 1872, 020004 (2017)
https://doi.org/10.1063/1.4996661

Encrypted Data Inquiries Using Chained Perfect Hashing (CPH)

Prof. Khalid Kaabneh[1,a)] and Dr. Hassan Tarawneh[2,b)] and Dr. Issam Alhadid[3,c)]

1) Department of Computer Science

College of Computer Science and Informatics. Amman Arab University, Amman, Jordan.

2) Department of Mobile Computing

College of Computer Science and Informatics. Amman Arab University, Amman, Jordan.

3) Department of Business Information Technology, College of Information Technology.

Jordan University, Jordan.

a) kaabneh@aau.edu.jo

b) hassan@aau.edu.jo

c) i.alhadid@ju.edu.jo

Abstract. Cryptography is the practice of transforming data to indecipherable by a third party, unless a particular piece of secret information is made available to them. Data encryption has been paid a great attention to protect data. As data sizes are growing, so does the need for efficient data search while being encrypted to protect it during transmission and storage. This research is based on our previous and continuous work to speed up and enhance global heuristic search on an encrypted data. This research is using chained hashing approach to reduce the search time and decrease the collision rate which most search techniques suffers from. The results were very encouraging and will be discussed in the experimental results section.

Keywords. Data Security, Encryption, Searching Encrypted Data, Secure Data Transfer.

Introduction

With the improvement of today's technology, data retrieval such as emails and data has increased and relied on by many users. Emails are fully connected with mail servers, although servers must be trusted and authorized, at the same time emails should be encrypted to reduce security risks.

In order to retrieve a specific email, the ordinary way would be retrieving then decrypting the whole content of the mailbox messages; this approach isn't practical and has its drawbacks. A new approach can be described by searching a specific keyword in encrypted document to provide both integrity and confidentiality issues.

During our research on an encrypted data, we have proposed a technique called Heuristic Search over Encrypted Data (HSED), which handles a large data keyword search in an encrypted documents using public key encryption stored in un-trusted servers without releasing information about the document and the search of specific query or the stored mails [1]. HSED has a number of drawbacks related to search time. The search time was decreased in a modified approach called General Heuristic Search over Encrypted Data (GHSED) [2]. GHSED uses a general

Mathematical Methods and Computational Techniques in Science and Engineering
AIP Conf. Proc. 1872, 020004-1–020004-9; doi: 10.1063/1.4996661
Published by AIP Publishing. 978-0-7354-1552-2/$30.00

table or (Global Heuristic Table GHT) to decrees the search time, but the above two approached suffers from high collision rates and a high memory overhead.

The above techniques suffer from long search time, high collision rate, security level and high memory overhead. Chained Perfect Hashing (CPH), is proposed as a solution for these problems. It's a cryptographic protocol for searching on encrypted messages without revealing any information to the un trusted server or any loss of data confidentiality, it will satisfy: security, controlled searching, support hidden queries and query isolation.

Our proposed solution has two main goals. First, to decrease the access time in an encrypted document, and to build an index with a minimal collision rate. To solve those issues, we propose a two-level perfect hashing. The expected results of this approach would have better results in reference to the current applied techniques.

Related Work

We have started our novel research using a new idea called Heuristic Search on Encrypted Data (HSED), which reduces communication overhead on the e-mail server, and it requires no additional computation except for simple calculation of a hash function that serves as the address for an entry in the Hash Table (HT). One drawback we face is the collision rate and construction time that is related to the file size, and it deals with each document a lone for searching on specific keyword, which will need a lot of search time [1].
GHSED is another research idea by applying advanced search technique on an encrypted data aimed to solve the problem handling each document alone by making a global heuristic table (GHT) which contains all documents. This technique solved the problem of repeated words in each document, but at the time needed to embed heuristic table into GHT which increased among file size, also it can not deal with compressed files, and have high collision rate increasing of search time.

In [5], they used the idea of extendible hashing (EH) to search for encrypted keyword over a remote encrypted e-mails in un-trusted servers, it is fast and secure with isolation query search but it did not solve the extra time if there was two operations at the same time, for example insert and search, also there is the problem of skewed data, multi records have the same search key so multi search key might have been assigned the same bucket, which will cause overflow problems, also it cause memory overhead.

Another novel technique was introduced to solve the problem of trying the user to retrieve some encrypted files from server by using encrypted keywords, in a way that ensures security but it will increase the overhead in terms of bandwidth and storage, because we are dealing with user that have mobile cell phones [3].

[4] proposed a study of privacy-preserving access to the database, by suggestion of general solutions rely on a new connection between keyword search and oblivious pseudorandom

functions, but in the whole paper the author have not show any results for the implementation, so there is no idea about how it will affect the search time and the level of security.

Eu-Jin Goh uses the idea of bloom filter. If the query contain the word(x), then the search time O(1) only if the index contain x. it's security depends on the right use of trap door, for generating the secret key. The security model known as semantic security against adaptive chosen keyword attack (IND-CKA), they also develop an efficient IND-CKA secure index construction called Z-IDX using pseudo-random function and bloom filters. The computational cost was low, but their indexes cannot securely handle updates [6].

A secure approach is used when two parities share data but do not trust each others, it propose a search scheme based on bloom filter and pohling-hellamn encryption. there will be a need of a trusted third party that can transform one party's database, but in a way that neither the third party nor the database owner can see the original query [7]. Also the encryption keys are hidden from the third party, which are used to construct the bloom filter, but how much does the third part trusted, they did not measure that.

Brent R. Waters, Dirk Balfanz, Glenn Durfee, and D. K. Smetters, said that the content of the audit log are very sensitive information, so if an organization whishes to search for a certain information, it might need to search all the entries to match some keywords, so it need to be done in a secure way. The server generating and it logs entries encrypt entries with the public key corresponding to the keywords that are derived from those entries. The escrow agent, a third trust party, will construct a search capability for a given keyword as the private key corresponding to the given keyword, and the adversary can not tell which public key was used to create the cipher text, so when an encrypted audit log entry is created, even if it's search keyword are hidden but it introduces considerable overhead [8].

[9] provide more than one technique that have feature of high speed, where the time needed to encryption and search is *O(n)* for a document of size n. all theses methods or scheme take the form of probabilistic searching, to control the number of errors for some parameters in the encryption algorithm, but their indexes can not securely handle updates.

Another approach focuses on the idea if the user was interested in documents containing the same searched keywords, the cost of the proposed model depend on the number of searched keywords, the server should learn nothing other than the results of the conjunctive query, they depend on the features of Decisional Diffie-Hellman(DDH) and Bilinear Decisional Deffie-Hellman(BDDH) to proof the security of there model, the model contain two scenarios which depend on the type of bandwidth connection, if it was high then the user pre-compute a lot of proto-capabilities and send them to the sever which saves it, beside it's belong documents until they are used, which can only used once. If the user has only access to a low-bandwidth connection, he will combine the two parts and access it, but it all depend on the reliability and availability of the connected network, their proposed idea suffer from the high communication cost in order of the number of the keywords, and it's security relies a new hardness assumption [10].

Proposed Scheme

1. Perfect Hashing

The most common use of hashing is to organize the database files. A sequential search of 100,000 records for customers will require too much disk I/O. An indexed file scheme (like B-tree) would probably make the search faster, but not faster enough to the responsiveness of customer's requests. Hashing is an efficient solution to this problem, it's a mathematical function that transforms the key to an integer value), the key here is the customer's record; the hashing function uses the record key to calculate a location for the record. An advantage of hashing is speed, and the most obvious disadvantage is collisions, for example e have two customers, two records, first one is 192 and second is 195, but the two of them have the hash value 105, means for the same position. A simple solution to collision is to store the record to the next empty position. Have function have the most important feature: speed and the B-tree have the unlimited storage ability, these can be combined together to have Extendible Hashing, which views the hash keys as an index into a page of pointer. A hash function that produces no collisions for a given set of keys called perfect hashing function. If the hash function also maps those keys onto consecutive numbers with no gaps, then it is called a minimal perfect hashing function.

Hashing used to solve the problem of large number f elements that no table could handle, uses a hash function *h(k)* that maps *k* randomly into slots of a hash-table *T*, when two keys hash to the same slot, a collision will be accord, this can be solved by chaining. In chaining, we put all elements that hash to the same slot in a linked list.

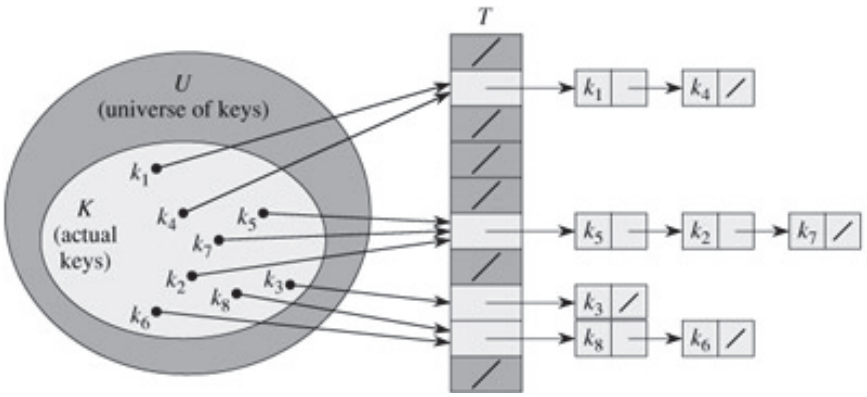

FIGURE 1. Universial Hashing.

But this weakness could lead to denial of serves attack on the application using hashing, that's why it's important to choose a hash function at random, this is called universal hashing.
If we do not want to use the chaining method to solve the collisions problem, we could go back to the universal hashing family and choose the hash function twice in two levels, the first level is essentially the same as for hashing with chaining: the n keys are hashed into m slots using a hash function h carefully selected from a family of universal hash functions. Instead of making a list of the keys hashing to slot j, we use a small secondary hash table S_j with an associated hash

function h_j. In order to guarantee that there are no collisions at the secondary level, we need to let the size of m_j of hash table S_j be the square of the number n_j, ($m_j = n_j^2$).

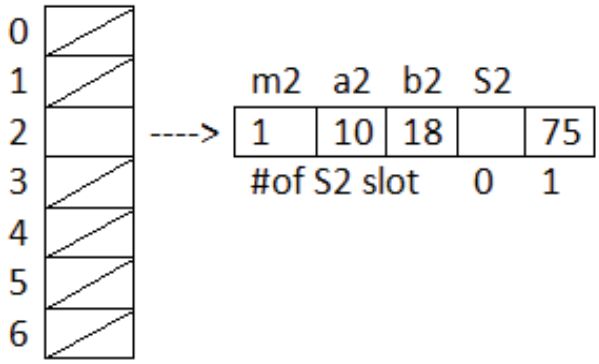

FIGURE 2. Perfect Hashing.

There are no collisions in any secondary hash tables, so searching take constant time in the worst case [11].

2. Secure Hash Algorithm (SHA)

Hash functions also called “digest” or “one way” functions, it has input as string of any length, and the output is a short fixed length string called a “message digest” or “finger print”, which should be collision-resistant. It’s advantage are fast, keyless operation and keyed operations, while it’s disadvantage is applicable to a limited numbed of situations. The uses of hash functions is in password storage, message and file integrity and commitment scheme [12].

Secure hash algorithms-SHA, also known as secure hash standards-SHS this hash algorithm was published by the united states government. This algorithm can produce an output of 160-bit hash value.

Hashing algorithms must be used to ensure the integrity of the message M which required that the SHA be used [13].

SHA is the third form of cryptology, where the two other forms are (enciphering and deciphering), it’s one way encryption. A hash is a cryptographic algorithm that takes a data input of any length an produces as output of a fixed length. The hash output is called a digital signature which used for data integrity. The larger the signature the more secure the hash.

It take an input from a large domain and return and output in a smaller range. It’s easy to compute [14].

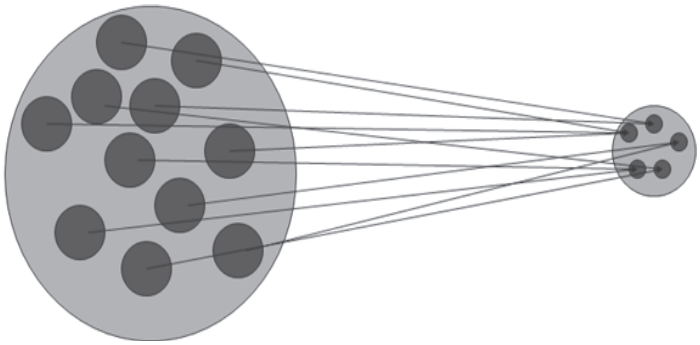

FIGURE 3. Rrange Values of I/O SHA Values.

Hash functions must have the following properties: can be applied to any size of data block, produce fixed length output, easy to compute, not feasible to reverse and not feasible to find two messages that give the same length [15].

3. Chained Perfect Hashing (CPH)

This section will explain the proposed algorithm in detail, but before that there will be comparison between the proposed algorithm and three of the most related ideas(HSED, CHSED and EH).

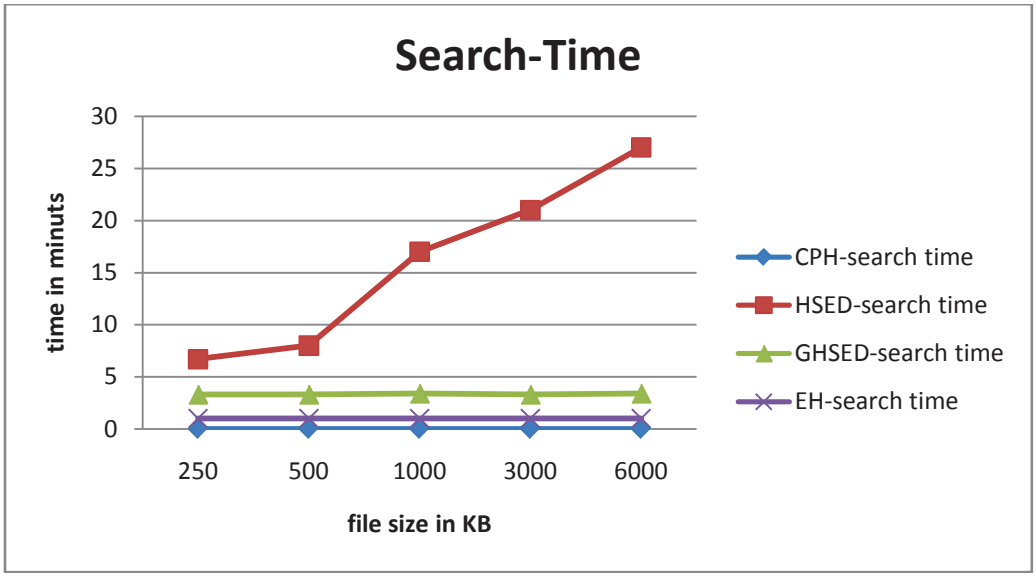

FIGURE 4. Search-Time's Comparison.

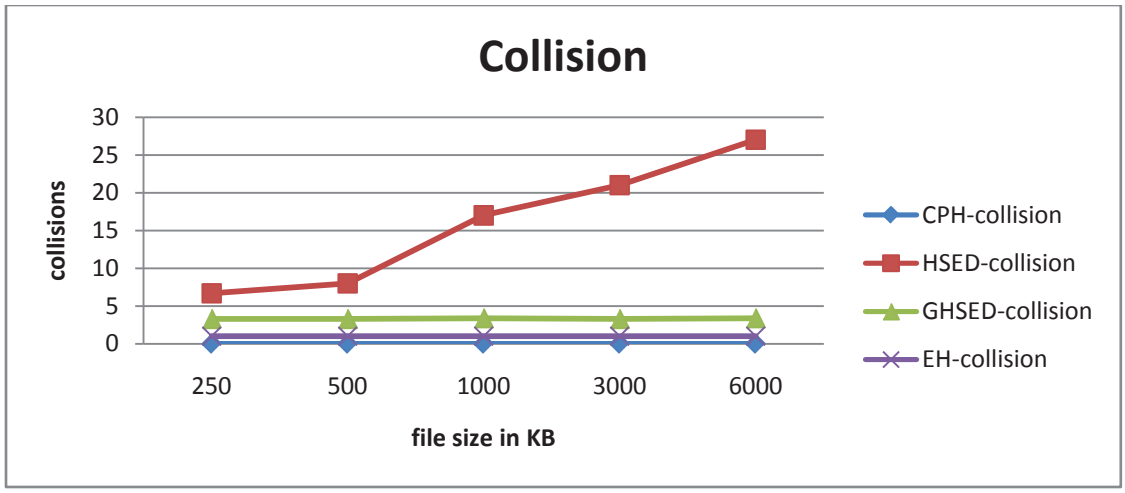

FIGURE 5. Collision Rate Comparison.

From the above, it's obvious that the proposed idea, CPH, have better performance from the others, both in the search time and the collision rates.

CPH-steps:

Sender-Side

Open file to read from it
While (not end of file)
  Read block of sentence
  For each word calculate
    x=ascii(1'st char) , y=sum of ascii(all char), $h_1(x)$ and $h_2(y)$
  Build the list : the parent node will contain ($h_1(x)$) and the child node will contain ($h_2(y)||E(keyword)||doc\#$)
Encrypt the document by SHA-1024

Email server-Side

Receive the encrypted email + list
Update the main list
Save the encrypted email

Searcher-Side

Input keyword: Enc(trapdoor(keyword $_{\text{searcher pub key}}$, $E(H_2(y)_{\text{searcher pub key}})$
If found(T)
  Send documents
Else
  Msg(not found)

Complexity-Time (CPH)

1-Construction time: $O(n^2)$

Loop : read from file
  $H_1(k)$=parent

$H_2(k)$=child
Loop : insert into linked list
If LIST(empty)
add parent & it's child
else if (parent exist)
add it's child
else
add parent & it's child

2-Search time: O(n)
Read word
$H_1(k)$
$H_2(k)$
Loop : scan the list for existence parent
If found then scan for existence child.

CONCLUSION

Searching an encrypted data has been gaining a lot of attention recently. This research is based on our previous and continuous work to speed up and enhance global heuristic search on an encrypted data remotely. This research experiments on data search using chained hashing to reduce the search time and decrease the collision rate which most search techniques suffers from. The results were very encouraging and gave a faster searching time *O(n)* compared to other techniques and a minimal collision rate.

REFERENCES

1. G. Sammour, J. Qaryouti, M. Shareef and K. Kaabneh, "Heuristic Search on Encrypted Data (HSED)", Department of Computer Science, Amman Arab University for Graduate Studies, AMMAN, JORDAN, 2005.
2. Halloush, "Applying Advanced Searching Techniques to Encrypted Data", Master Tesis, Al-Balqa Applied University, Jordan, 2007.
3. Yan-Cheng Chang and Michael Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data", Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA
4. Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold, "Keyword Search and Oblivious Pseudorandom Functions", New York University.
5. Affef, "Privacy-enhanced searches on encrypted emails using secure extendable hash tree", CS, Amman Arabia, 1998.
6. Eu-Jin Goh, Secure Indexes, May 5, 2004
7. Steven M. Bellovin and William R. Cheswick, "Privacy-Enhanced Searches Using Encrypted Bloom Filters", Lumeta.

8. Brent R. Waters, Dirk Balfanz, Glenn Durfee, and D. K. Smetters, “Building an Encrypted and Searchable Audit Log”, Princeton University, Computer Science Department, Princeton, NJ 08544.

9. Dawn Xiaodong Song David Wagner Adrian Perrig, “Practical Techniques for Searches on Encrypted Data”, University of California, Berkeley.

10. Philippe Golle and Jessica Staddon and Brent Waters, “Secure Conjunctive Keyword Search Over Encrypted Data”, Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA.

11. Thomas H. Cormen, Introduction to Algorithms, Second edition, North America, Mc-Grow Hill, 2003.

12. Roman Fail, Cryptography and User Management, Foundstone Professional Services, Java in action.

13. Bowne, Symmetric Cryptography Algorithms, Symmetric Cryptography, Asymmetric Cryptography, and Digital Signatures.

14. Fred Moore, President, Data Encryption Strategies, Horison, Inc, http:// www.horison.com

15. Allam Apparao, Network Security, Data and Computer Communications.