Analysis of Request over Possible Clouds without Merged Duplicates

[1] R.Shanthi (Asst prof) [2] V.Raghavi [3] B.Revathy [4] D.Sharon Sharmishtha [5] D.Shunmugapriya
Department of information technology
Valliammai Engineering College

Abstract: **To realize entity resolution, unmerged duplicate, probabilistic database and top k queries in query sanalytics over database. To provide more secure data transmission. To protect unauthorized person from retrieval of data and security to the data packets by using information security system. Entity resolution approaches exhibit benefits when addressing the problem through unmerged duplicates instances describing real world objects are not merged based on thresholds or human intervention, instead relevant resolution information is employed for evaluating resolution decisions during query processing using possible world's semantics. In this paper, we present the first known approach for efficiently handling complex analytical queries over probabilistic databases with unmerged duplicates.**

Keywords: **Entity resolution, unmerged duplicates, multiple cloud's**

I. INTRODUCTION

Cloud-based services are ideal for businesses with growing or fluctuating bandwidth demands. If your needs increase it's easy to scale up your cloud capacity, drawing on the service's remote servers. Entity resolution **is** the task of processing a data set in order to create entities by merging the data set instances that describe the same real-world objects. The Cloud Computing is where it improves accessibility by accessing information anywhere at any time. The typical situation is that the unmerged duplicates are part of the large database that of course contains other tables. Retrieving relevant information to all data by eliminating the merged duplicates. The major issue is securing the data. To overcome this major issue, security is improved by cookie management .HTTP is stateless protocol. Requests to a web server are treated as independent transactions with no relation to each other. More robust alternatives to authentication cookies have been proposed .HTTP cookies ,small pieces of data that keep section state information in the browser. For example most web applications relay on the security provided by HTTP to protect the user's password during the login propose. There's no need to spend big money on hardware, software or licensing fees. Get more work done in less time with less people. Globalize your workforce on the cheap, People worldwide can access the cloud, provided they have an Internet connection. The beauty of cloud computing is that the servers are off-premise, out of sight and out of your hair. Suppliers take care of them for you and roll out regular software updates – including security updates – so you don't have to worry about wasting time maintaining the system yourself. Leaving you free to focus on the things that matter, like growing your business. Entity resolution approaches exhibit benefits when addressing the problem through unmerged duplicates. Instances describing real world objects are not merged based on apriority thresholds or human intervention. The disadvantage of existing system is that relevant resolution information is employed for evaluating resolution decisions during query processing using possible world's semantics. To overcome the disadvantages of existing system. A newly implemented proposed system consists of, to provide a security to the data while transferring or retrieving the data. The practical scenarios that do not require retrieving the huge collection of all possible resolution worlds, but rather analytical or summarized information on the entities. A MAGIC cookie will prevent this session hijacking attack by encrypting the Cookie with MAC address so that attacker cannot do any type of session hijacking.

II. RELATED WORK

Our paper is most related to database.

[1] Considering a database with duplicated instances, probabilistic linkages between duplicated instances, and tables with other related data. The semantics of queries with ENTITY JOIN focusing on two analytical query types: aggregation and top-k/iceberg.

[2].To overcome the security attack we use session management. OTC(one time cookie)prevents attack such as

session hijacking by signing each user request with a session secret securely stored in the browser.

[3]. A survey of uncertain data algorithms and applications, In the field of uncertain data management, examine traditional database management methods such as join processing, query processing, selectivity estimation, OLAP queries, and indexing. In the field of uncertain data mining examine the problems such as frequent pattern mining, outlier detection, classification, and clustering.

[4]. Clean answers over dirty databases**,**An approach that permits declarative query answering over duplicated data, where each duplicate is associated with a probability of being in the clean database. They rewrite queries over a database containing duplicates to return each answer with the probability that the answer is in the clean database.

III. PROPOSED SYSTEM:

A proposed system has been implemented to enhance the security for the data and to overcome the disabilities of the existing system.

Entity Resolution is the task of processing a data set in order to create entities by merging the data set instances that describe the same real-world objects. An entity join is the probabilistic join between tables with each of the entities in the possible worlds. Note that a possible world will be considered in the join only when this is valid, the transitivity of the accepted linkages does not violate the transitivity of rejected linkages. This means that if the accepted linkages imply entity then the rejected linkages must also indicate that one instance can be merged with other instance.

The query semantics focus on enabling users to retrieve analytical or summarized information about the entities and related uncertainties. The basis for this task is the set of possible resolved entities that can be created given the duplicated instances and the probabilistic linkage information. The clause, and more specifically the entity join operator, states that the possible resolved entities must be joined with the records of table. Here, we assume a group by operator over each possible merge. Thus, the corresponding result set must include one record per entity. Although this group by operator is not directly included in the query syntax, it is implicitly expressed through the aggregation function.
We now introduce the indexing structure, which forms the basis of the efficient processing of the supported query types. The main goal of the indexing structure is to reduce the complexity of the computations required when processing queries. This is achieved since the indexing structure provides efficient access to the information encoded through the linkages and since it allows easy construction of possible worlds as well as the fast retrieval of their probabilities.

For example

- ❖ The server can login by using username and password.
- ❖ The server seems those who are upload the file to server.
- ❖ The server analyse the database which one is more space and upload the file to drop box or cloud me.
- ❖ The server sends the database upload details to the owner and the owner seem file upload in which database.

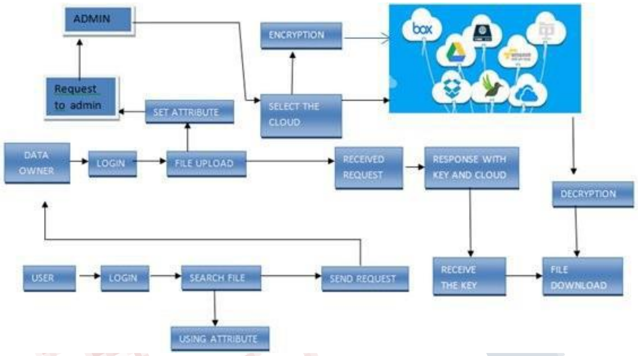
The indexing structure contains the top K linkage combinations for each factor. Iceberg or top k query by navigating in the structure and the factors. The algorithm uses a temporal list that contains the factors with instances satisfying the query, sorted by their upper bound probability. The retrieval of the entities is iterative, and each step processes the linkage combinations in the factor with the highest probability from the temporal list. For this factor, it uses the indexing structure to retrieve combinations, and on each combination it performs its merge. These merges are included in the result list if their probability is higher than that the upper bound of the following factor. Processing continues with the following factor, when we find a merge with a probability lower than the upper bound of the factor.
Where the attacker uses packet sniffing to read network traffic between two parties to steal the session cookie. Many web sites use SSL encryption for login pages to prevent attackers from seeing the password, but do not use encryption for the rest of the site once authenticated. This allows attackers that can read the network traffic to intercept all the data that is submitted to the server or web pages viewed by the client. Since this data includes the session cookie, it allows him to impersonate the victim, even if the password itself is not compromised. Unsecured Wi-Fi hotspots are particularly vulnerable, as anyone sharing the network will generally be able to read most of the web traffic between other nodes and the access to system. A magic cookie, or just cookie for short, is a token or short packet of data passed between communicating programs, where the data is typically not meaningful to the recipient program. The contents are opaque and not usually interpreted until the recipient passes the cookie data back to the sender or perhaps another program at a later time. The cookie is often used like a ticket – to identify a particular event or transaction. The other type of session attack is session fixation. Here, instead of stealing/hijacking the victim's session, the attacker fixes the user's session ID before the user even logs into the target server (that is, before authentication), thereby eliminating the need to obtain the user's session ID afterwards. Before going into detail of session fixation attacks, we must classify two types of sessions managed on Web servers: Permissive sessions allow the client's browser to propose any session ID, and create a new session with that ID if one does not exist.

After that, the server continues to authenticate the client with the given ID.Strict sessions allow only server-side-generated session ID values.

A successful session fixation attack is generally carried out in three phases:

1.Phase I or session set-up: In this phase, the attackers set up legitimate session with the Web application, and obtain their session ID. However, in some cases the established trap session needs to be maintained (kept alive) by repeatedly sending requests referencing it, to avoid idle session time-out.
2. Phase II or fixation phase: Here, attackers need to introduce their session ID to the victim's browser, thereby fixing the session.
3. Phase III or entrance phase: Finally, the attacker waits until the victim logs into the Web server, using the previous session ID.

IV. ARCHITECTURE DIAGRAM



V.BACKGROUND EXPERIMENTAL ALOGRITHM

A. *Attribute Based Encryption*

Attribute-based encryption is a type of public- key encryption in which the secret key of a user and the cipher text are dependent upon attributes (e.g. the country he lives, or the kind of subscription he has). In such a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text. A crucial security aspect of Attribute-Based Encryption is collusion-resistance: An adversary that holds multiple keys should only be able to access data if at least one individual key grants access.

B. *Top-K Quries:*

Top-*k* queries retrieve the objects that best match the user requirements by employing user-specified scoring functions that result in an ordered set of objects containing the best *k* objects only [30, 75].
Efficient processing of top-*k* queries in peer-to- peer systems is studied. To this end, the applicability of the skyline operator is investigated for efficiently answering top- *k* queries for a wide class of scoring functions, indicating user-specified preferences, in large P2P networks.

C. *Advanced Encryption Standards:*

STEP1: Key expansion-round keys are derived from cipher keys. AES need 128bit round key block for every round with one more.
STEP2: Initial round –1.AddRoundkey-each bit is added with block of round key.
SETP3: Other round- Subbytes, shiftrows, mixcloumns, addroundkeys. STEP 4: In final round expect the mix columns other 3 rounds takes place.

VI.CONCLUSION

In this paper we address problem through the processing complex queries over unmerged duplicates. Our approaches mainly based on database with duplicate instances and providing security to the data through information security. We have introduce magic cookie for securing the data using session management. Moreover OTC offers another security layer through web applications.

REFERENCES

[1]Query Analytics over Probabilistic Databases with Unmerged Duplicates Ekaterini Ioannou and Minos Garofalakis. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 8, AUGUST 2015.

[2]One-Time Cookies: Preventing Session Hijacking Attacks with Stateless Authentication Tokens .IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 6, JUNE 2015.

[3] R. Fink, L. Han, and D. Olteanu, "Aggregation in probabilistic databases via knowledge compilation," Proc. VLDB Endowment ,vol. 5, no. 5, pp. 490–501, 2012. [4]The Open Web Application Security Project (OWASP). Cross-siteScripting (XSS).http://www.owasp.org/index.php/Cross-site Scripting 2010.

[5] D. Wang, M. Franklin, M. Garofalakis, and J. Hellerstein,"Querying probabilistic information extraction," Proc. VLDB Endowment, vol. 3, no. 1, pp. 1057–1067, 2010.

[6]C. Aggarwal and P. Yu, "A survey of uncertain data algorithms and applications," IEEE Trans. Knowl. Data Eng., vol. 21, no. 5, pp. 609–623, May 2009.

[7]C. Jackson and A. Barth. Force https: protecting high-security web sites from network attacks. In Proceeding of the

ACM international conference on World Wide Web (WWW), 2008.

[8]Duplicate Record Detection: A Survey Ahmed K. Elmagar mid, Senior Member, IEEE, Panagiotis G .Ipeirotis, Member, IEEE Computer Society, and Vassilios S. Verykios, Member, IEEE Computer Society,2007.

[9]P. Andrit os, A. Fuxman, and R. Miller, “Clean answers over dirty databases: A probabilistic approach,” in Proc. 22nd Int. Conf. DataEng., 2006, p. 30.

[10]Jehiah. XSS - Stealing Cookies 101. http://jehiah.cz/a/xss-stealing-cookies-101, 2006.