International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017), e-ISSN: 2455-2585
Volume 4, Issue 09, September-2018

ENHANCING THE EFFICACY OF INEQUALITY QUERY AUDITING AND IMPROVED IN DEVIATION AUDITING USING CASTLE ALGORITHM

K.Vinotha[#1], Dr. G.Kesavaraj[#2],

#1M.Phil Scholar (Full-Time), #2 Associative Professor,

Department of Computer Science,
Vivekanandha College of Arts and Sciences for Women, (Autonomous)
Tiruchengode, Namakkal-DT, TamilNadu, INDIA

ABSTRACT:- We reconsider the various query auditing problem in set of sensitive data is outsourced to a cloud server. Castle query auditing scheme audits aggregate queries including sum, max, min, deviation, etc. These are submitted into an often manner, on the method to protect inference disclosure. It audits currently arrived queries on a single attribute, If any answering it may compromise any individual privacy that query will be rejected. This method analyzes risk of answering a query based on the query history. In additionally we propose relax CASTLE method for enhancing the utility by returning answers with slender perturbations. Our method can be applied into audit intermingled equality queries with extension. Experiments are conducted to evaluate the efficiency and effectiveness of our methods.

Keywords: Query Auditing, Deviation Auditing Privacy-Preserving Query, Denial Threats.

I. INTRODUCTION

In the era of huge statistics, growing amounts of personal data are being accumulated and shared among various parties through the cloud. Therefore, it's remote vital to increase a secured facts sharing, accessing, or exchange and selling mechanism. Taking the medical health insurance Portability and accountability Act (HIPAA) as an example, to keep away from re-identification, entities need to take away or perturb at the least 18 PHI statistics factors once they percentage sensitive data. But, sanitization or perturbation reduces the software of the dataset significantly. Numerous works had been proposed that discover the trade-off between utility and privateness. Their closing aim is to guard people' private facts from unauthorized get entry to. One feasible answer, which is query auditing; that is also the point of interest of our paintings.

We revisit the traditional query auditing hassle in the cloud platform, which serves the cloud as an auditor. Query auditing proposed query will cause privacy compromise when given a sensitive dataset as well as a sequence of formerly responded queries on a free characteristic, and the corresponding solutions. Inequality query auditing, in which the query is of the form f(X), is a subset of the sensitive dataset, and the answer is either 'yes' or 'no'. The characteristic f can be any polynomial l- time computable function (e.g., logistic, linear regression), that is greater general than the aggregate query.

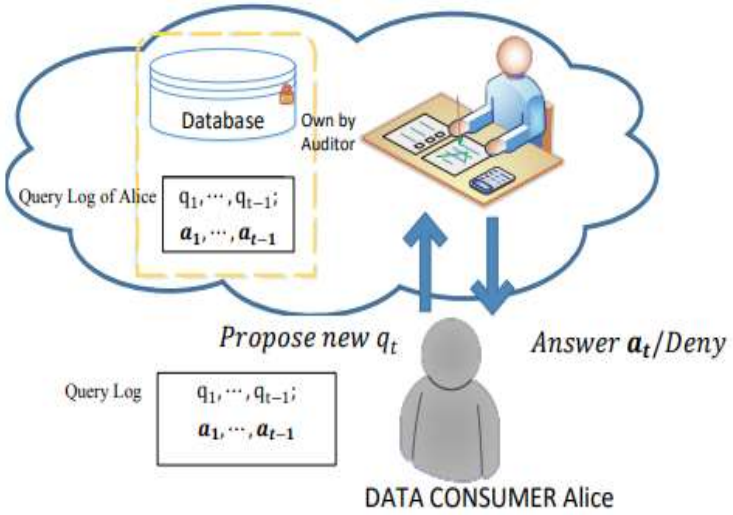

Fig 1: Query Auditing System Model

II. EXISTING SYSTEM

In this existing system proposed statistical info auditing for audits combination queries (sum, max, min, etc.), that are submitted during a continuous manner, to prevent reasoning revelation. Auditing is often accomplished by examining responses to past queries to see whether or not a replacement question will be answered. it's been recognized that question denials un harness data and might cause information revealing. Explanation the whole data leak from every query denial and carrying the whole leaked data derived from past answered and denied queries to audit every new question. The data leak explanation drawback will be developed as a collection of constant optimization programs, and therefore the whole auditing method will be sculptural as a series of protrusive optimization issues.
Statistically auditing/evaluating a selected individual's personal records (e.g., medical data). The dataset of n people defines a n-dimensional area, wherever every individual corresponds to at least one dimension. Observe that a difference question over this dataset, in addition to its answer ('yes' or 'no') defines an n-dimensional valid region. Given a collection of previous queries and their answers, an aggressor will slender down the potential values for the dataset by computing the intersection of all valid regions such as all antecedently answered queries (called the answer area to any or all such queries).
Then a naive auditing mechanism is to deny this question if there's a novel resolution for all historical inequalities with this one (which reveals that the dataset is re-identified).

Advantage
- Flexibility
- Maximum information
- Reduce information leakage problem

Disadvantages
- Difficulty of removing denial threats
- Lack the analysis of utility
- Lack number of answered queries
- Doesn't provide sufficient security

III. PROPOSED SYSTEM

In this paper, we tend to propose CASTLE, which with success audits the difference queries while not denial threats. To audit a received query, we tend to estimate the amount of the posterior answer area, and leverage a recent work [9], that approximates the amount of a convexo-concave polytope in a very high-dimensional area. Note htat it's troublesome to answer a question once the dataset is sort of pinpointed, as a result of the intersection of all answer areas becomes quite tiny. To enhance the utility, that is described because the variety of answered queries, we tend to additional present a relaxed CASTLE by adding errors to the answers. To keep up the service quality (introducing too several errors seriously degrades the quality), the error budget is outlined because the sum of the further errors. Maximising the quantity of answered queries below the constraint of the error budget, involves another difficult optimisation task – on-line maximization with an unknown question distribution. Moreover, we tend to still fix the privacy problems with the same strategies with differential privacy, with the aim of absolutely protective the knowledge that's provided by the info owners. Finally, we tend to show that our methodology will audit the integrated queries (min, max, and sum) and isn't restricted to the difference question via another extension. To the most effiective of our information this can be the primary paper to check difference question auditing, within which the question perform is any polynomial time estimable perform whereas considering a denial threat.

Advantages
- auditing efficiently
- increase the utility of query auditing
- provide accurate answer
- Avoid denial threat
- Provide sufficient privacy
- Answered large number of questions within few minutes

IV. ALGORITHM

CASTLE: INEQUALITY QUERY AUDITING
Makes "cloud first" realizable for agency. It develops recent application higher and quicker. And cloud computing transforms the legal relationship between individuals and their their individual proceedings. Federal-having (or relevancy a system of states among that many countries of a unity but keep add internal associations. Whether or not the posterior answer residence is finite by the safe zone (i.e., the relation of $S_s^t \cup S_z$ is large) via sampling. Throughout this approach, our theme is applied to any polynomial computation question. Our algorithmic method is freed from denial threats since the answer residence won't be narrowed right all the approach right down to some extent and may well be finite

from below by the safe zone. Specifically, for any non-public dataset $X = \{x_{,\ldots\ldots,}x_n\}$, , to come back to a assessment whether or not or not the recently received query qt is answered, we wish to ascertain whether or not or not the chance P r, that is outlined in equivalent. 1, is a smaller amount than $1 - \delta$. If $Pr \geq 1 - \delta$, qt are properly answered; otherwise, it'll be denied.

Algorithm 1: *Estimation of* ***Pr****(Probability)*

Input: $S_z=\{l_1 \leq x_1 \leq u_{1,\ldots,} l_n \leq x_n \leq u_n\}$,*and* S_s^{t-1},$<q_t, a_t>$ *; n, €,* a_O *,ratio,* r_{steps} *,W;*
Output: Pr $\leftarrow \frac{|N^t|}{|N|}$
Step 1*: Let* $N_t = \emptyset$ *and* $N = \emptyset$
Step 2: *Update* S_s^{t-1} *to* $S_s^{t'}$ *with* $<q_t, a_t>$
Step 3:T = *Round* (S_z, r_{steps})
Step 4: *Set* $S_z^{'} = T \cdot S_z$ *and* $S_s^{t'} = T \cdot S_s^t$
Step 5: $<f_o, \cdots, f_m>$ *= GetAnnealingSchedule* $(S_z^{'}, a_O, ratio)$
Step 6: *for i = 1; i< m; i + + do*
Step 7: *Set converged = false*
Step 8: *while converged = false do*
Step 9: *Sample P based on HitAndRun* (K', f_{i-1})
Step 10: $N \leftarrow N \cup P$
Step 11: *if P is within* $S_s t'$ *then*
Step 12: $N_t \leftarrow N_t \cup P$
Step 13: *end if*
Step 14: *converged = Checkconverged (€/m, W)*
Step 15: *end while*
Step 16: *end for*
Step 17: *Return Pr = |Nt|/|N|*

Algorithm two is predicated on Algorithm 1, wherever the answer area is barely updated with qt and its correct answer once P r ≥ one − δ; otherwise, it remains an equivalent. In Algorithm two, it takes the subsequent as input: the query and its answer <qt, at>. If the proper answer to a query, e.g., f ( )≤? a, is e.g., „no", then it is stored in the form e.g., <−f(X), −a> or <f(X), a> with correct answer „yes". The safe zone is denoted as Sz; the answer area is shapely by Sst'; The dataset is denoted as X and its size is n; and the privacy parameter δ is defined in Definition one.

Algorithm 2: CASTLE

Input: X = $\{x_1; \ldots., x_n\}$;
$\mathbf{S_z}$ = $\{ l_1 \leq x_1 \leq u_1; \ldots., l_n \leq x_n \leq u_n\}$;
and$\mathbf{S_s^{t-1}}$; $\mathbf{f_t}$; n; δ ; €; a_O; ratio; r_{steps};W;
Output: 'yes=no' or 'denial'
1: Obtain <qt; at> from f_t;
2:
3: Estimate P_r with following inputs via Algorithm 1:
$\mathbf{S_z}$, $\mathbf{S_s^{t-1}}$,q_t; a_t; n; € ; a_O; ratio; r_{steps};W;
4: if Pr ≥1-δ then
5: Update $\mathbf{S_s^{t-1}}$ to $\mathbf{S_s^t}$ with $<q_t; a_t>$
6: Return `a'$_t$
7: end if
8: Return `denial'

RELAXED CASTLE: UTILITY MAXIMIZATION

Relaxed CASTLE Overview
Briefly, any fresh received query are 1st audited by CASTLE. If P r $(D \in S_s^t | D \in S_z) \geq 1 - \delta$, then the auditing takings usually. Otherwise, qt is denoted as Associate in nursing insecure query, which we will appreciate the minimum perturbation that the chances are larger than one − δ. After that, we have a tendency to area unit able to improve the utility by respondent this insecure question with a tiny low perturbation that consumes the error budget. Since the error budget is verboten to E, ideally, forever we continuously always prefer to answer the insecure queries that require the smallest perturbation for the only utility. If the sequence of all queries were well-known ahead, it might be straight

forward to decide on the optimum set of insecure queries for maximising the effectiveness ψ though agreeable the error budget constraint (by covetously and iteratively choosing the insecure query with the smallest error demand). However, the sequence of all queries isn't well-known ahead. Hence, once given the minimum perturbation of AN insecure question, we've got a bent to boost professional to form a choice whether or not or to not perform perturbation or deny the queries to maximize the utility (approximates the optimum set) at intervals the error budget limitation.

Algorithm 3: RELAXED CASTLE

Input: $\mathbf{X} = \{x_1, \cdots, x_n\}$;
$\mathbf{S_z} = \{l_1 \le x_1 \le u_1, \cdots, l_n \le x_n \le u_n\}$;
and$\mathbf{S_s^{t-1}}$,**ft**; $<q_t, a_t>$; n, δ, €$_s$, a_O, ratio, r_{steps}, **W**;
Output: 'answer' or new answer with a'$_t$ or 'denial'
1: Obtain $<q_t, a_t>$ from $\mathbf{f_t}$;
2: Evaluate Algorithm 1 with inputs
X, $\mathbf{S_Z}$, $\mathbf{S_S^{t-1}}$,$\mathbf{f_t}$, n, €$_s$, a_O, ratio, r_{steps}, **W**;
3: **if** answer is 'answer' **then**
4: Update $\mathbf{S_S^{t-1}}$ to $\mathbf{S_s^t}$with $<q_t, a_t>$
5: **else**
6: Resort to Expert with input e_t, t, τ, C_c, **E**, LB, UB
7: where $e_t = \min \{e|a'_t \leftarrow a_t + e; Pr\tilde{} \ge 1 - \delta\}$
8: **if** Expert outputs 'answer' **then**
9: Update S_s^{t-1} to S_s^t with <qt, a'$_t$>
10: **Return** a'$_t$, where $a'_t = a_t + e_t$,
11: **end if**
12: **end if**
13: **Return** 'denial'

EXTENDED CASTLE

The basic approach is analogous to CASTLE: checking whether or not the answer region has enough uncertainties (in the invigorated safe zone) via sampling. The distinction is that the safe zone is projected on the hyperplane that's evoked by the new received queries with its answer, consequently the safe zone can have identical dimension because the posterior resolution residence for every received query.

Algorithm 4: Extended CASTLE

Input:X = $\{x_1,\ldots,x_n\}$;
and S_z^{t-1},$\mathbf{S_s^{t-1}}$, q_t; n, δ, €$_s$, a_O, ratio, r_{steps}, **W**;
Output: answer $a_t$or 'Denial'
1: $a_t = q_t(X)$;
2: Preprocess $\mathbf{S_s^{t-1}}$and $\mathbf{S_z^{t-1}}$ to $\mathbf{S_s^{t'}}$and$\mathbf{S_z^{t'}}$
3: Evaluate Algorithm 1 with inputs
X, $\mathbf{S_z^{t'}}$, $\mathbf{S_s^{t'}}$, n, €$_s$, a_O, ratio, r_{steps}, **W**;
4: **if** P~r< - δ **then**
5: **Return** `denial'
6: **end if**
7: Update $\mathbf{S_s^{t-1}}$to$\mathbf{S_s^t}$ with $<q_t, a_t>$
8: **Return** a_t

V. EVALUATION

GRAPH REPRESENTATION ANALYSIS

Our experiment relies on a Chicago employee earnings dataset that has the names of workers and their salaries. The progressive techniques on volume estimation space polytope. Unit only acceptable for estimating high-dimensional bodies that square measure delineated as H-polytope or P- so, throughout this work, we have an inclination to only price some varieties of distinction queries, admire add and liquid ecstasy.

CASTLE VS SIMULATABLE

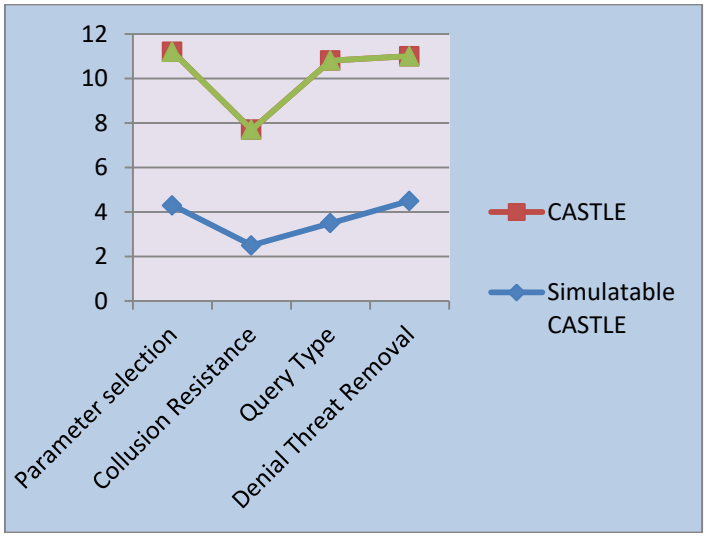

Fig: 2 Castle vs Simulatable

CASTLE VS RELAX CASTLE

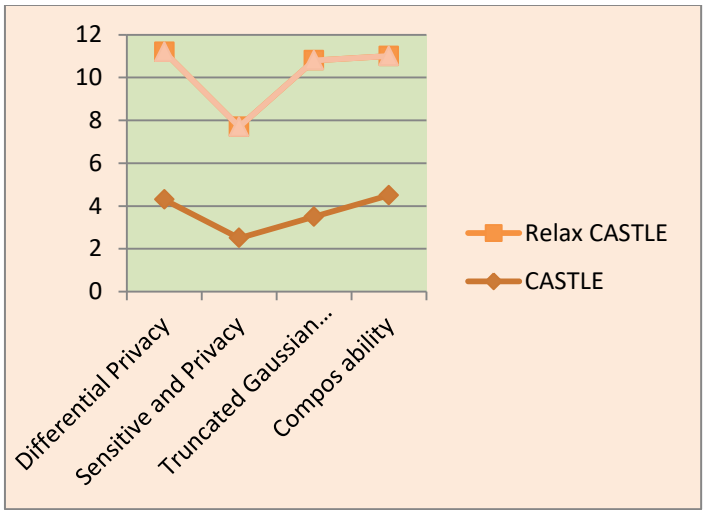

Fig 3: Castle vs Relax Castle

TABLE OUTPUT

Table 1: Efficiency of CASTLE

Data Size	Auditing Time	Setting Up
50	1.12	1760.21
100	4.62	9473.77
200	2548.14	46330.45
300	4385.76	68703.41
400	6490.63	85021.72

Table 2: Efficiency of Relaxed Castle

Generation	Minimum Perturbation	Expert
0.01	5934.84	2.31
0.02	1128.24	4.61
0.03	2246.51	8.27
0.04	4456.26	9.98

Table 3: Statistical Vs. Castle Algorithm

S. No	Description	STA	CASTLE
1	**Data Size**	Average	Excellent
2	**Auditing Time**	Better	Excellent
3	**Security**	Average	Excellent

VI. CONCLUSION

This paper presents a novel query auditing scheme called CASTLE which allows (i) auditing inequality queries without denial threats and (ii) enhancing the utility with slight perturbation. Our scheme presents a more comprehensive and general privacy definition, which is based on the safe zone and considers the correlation among the dataset. It achieves good performance in terms of auditing efficiency. We also propose a relaxed version that improves the utility while satisfying differential privacy. Furthermore, we propose an extended scheme for auditing intermingled equality queries without denial threats, which has not been studied in the past. In Our experiments demonstrate the efficiency of our schemes.

VII. FUTURE ENHANCEMENT

This paper presents a unique question auditing scheme referred to as CASTLE that permits (i) auditing difference queries while not denial threats and (ii) enhancing the utility with slight perturbation. Our theme presents a dditional comprehensive and general privacy definition, that relies on the safe zone and considers the correlation among the dataset. It achieves smart performance in terms of auditing potency. We have a tendency to additionally propose a relaxed version that improves the utility whereas satisfying differential privacy. furthermore, we have a tendency to propose an extended theme for auditing integrated equality queries while not denial threats, that has not been studied within the past. In Our experiments demonstrate the potency of our schemes.

VIII. REFERENCE

[1]Jianwei Qian, Xiang-Yang Li, et al., "Social network de-anonymization and privacy inference with knowledge graph model," IEEE Transactions on Dependable and Secure Computing, Pp.1-14, 2017.
[2] DavidDobkin, Anita K Jones, et al., "Secure databases: Protection against user influence," ACM TODS, vol. 4, no. 1, Pp. 97–106,1979.
[3]Francis Y Chin and GultekinOzsoyoglu, "Statistical database design ACM TODS, vol. 6, no. 1, Pp. 113–139, 1981.
[4] Norman S Matloff, "Another look at the use of noise addition for database security," in IEEE EuroS&P, Pp.173-180, 1986.
[5] KrishnamurtyMuralidhar, Rahul Parsa, et al., "A general additive data perturbation method for database security," Management Science, vol. 45, no. 10, Pp. 1399–1415, 1999.
[6] Francis Chin, "Security problems on inference control for sum, max, and min queries," JACM, vol. 33, no. 3, Pp. 451–464, 1986.
[7] KrishnaramKenthapadi, Nina Mishra, et al., "Simulatable auditing," in PODS. ACM, Pp. 118–127, 2005,
[8] Shubha UNabar, BhaskaraMarthi, et al., "Towards robustness in query auditing,"in VLDB. VLDB Endowment, Pp. 151–162, 2006.
[9] Haibing Lu, Jaideep Vaidya, et al., "Statistical database auditing without query denial threat," INFORMS JOC, vol. 27, no. 1, Pp. 20–34, 2014.
[10] KrishnaramKenthapadi, Nina Mishra, et al., "Denials leak information: Simulatable auditing," JCSS, vol. 79, no. 8, Pp. 1322–1340,2013.
[11] Christian P Robert, "Simulation of truncated normal variables," Statistics and computing, vol. 5, no. 2, Pp. 121–125, 1995.
[12] Agrawal S, Budetti P Physician medical identity theft. JAMA Pp. 459–460, 2012.
[13] J. M. Charnes, D. J. Morriceet.al., "The Hit-And-Run Sampler: A Globally Reaching Markov Chain Sampler For Generating Arbitrary Multivariate Distributions", Proceedings of the Winter Simulation Conference, Pp. 260–264, 2010.
[14] Lee S, Genton MG, Arellano-Valle RB, "Perturbation of numerical confidential data via skew-t distributions", Management Sci.56, Pp. 318–333, 2010.
[15] Li X-B, Sarkar S "Protecting privacy against record linkage disclosure: A bounded swapping approach for numeric data", Inform. Systems Res. 22, Pp. 774–789, 2011.