ISSN:0975-9646

Enriching Forensic Analysis process for Tampered Data in Database

Pallavi D Abhonkar, Ashok Kanthe

Department of Information technology
SIT, Lonavalva, Pune, India

Abstract- The need for secure data storage has become a necessity of our time. Medical records, financial records, and legal information are all in need of secure storage. In the era of globalization and dynamic world economies, data outsourcing is inevitable. Security is major concern in data outsourcing environment, since data is under the custody of third party service provider. In present systems, third party can access & view data even though they are not authorized to do so or even when the data is outsourced to the auditors or allow the employee of the organization to do the updating in the database. This may lead to the serious data theft, data tampering & even data leakages causing severe business impact to data owner.

There are certain many such cases occurred in financial & insurance sector where the data is been tampered by the auditors or by the employees of the organization itself. In this paper we have proposed a novel solution to overcome the problem of tamper detection by notarizing the original data. A heuristics approach is presented in our model where a validator system always authenticate the data for its originality using strong one way hash key functions like MD5 with authorized notarizer. By providing different digital signatures for different data owners, the proposed system provides a strong notarization & validation schemes to maintain high data security and integrity requirements.

Keywords

Security, hash key,MD5,SHA-1,DSA, Data outsourcing, Notarization, Efficient Notarizer, Digital Signature, Validator,Forensic Analysis, Data Set, Avalanche Effect, Secure Data Server.

1. INTRODUCTION

Secure data storage is an everyday requirement for public businesses, government agencies and many institutions. For many organizations, if data were to be maliciously changed, whether by an outsider or by an inside intruder, it could cause severe consequences for the company. Possibly even for their clients as well.

There are many reasons why someone might want to tamper with data. For example, an unsatisfied student who receives a “D” grade in his mathematics subject, in which he needed at least a “B”, could be highly tempted to try to dishonestly change his grade to a “B” in the school’s database. This would be an example of someone who would have to hack into the system from the outside, unless of course the student somehow had access to the database containing the grade.

A similar example, wherein the intruder is an insider rather than someone hacking in from the outside, could be that of an employee at a large company who is trying to meet his sales requirements for a fiscal year. He might attempt to change the dates of transactions to make it appear that they had transpired within the previous fiscal year when, in reality, they had not.

Data outsourcing is an emerging paradigm that allows users & companies to give their (potentially sensitive) data to the external servers that then become responsible for the storage, management and dissemination.

By outsourcing organization can concentrate on their core business activity rather than incurring the substantial hardware, software and personnel cost involved in maintaining applications in-house. Although data outsourcing provides many benefits especially for parties with limited data operators for managing an evermore increasing amount of data, it also introduces new privacy and security concerns.

As promising as it is, this paradigm also brings many new challenges for data security. When business organizations outsource sensitive data for sharing on servers, which are not within the same trusted domain as data owners. In data outsourcing scenario access to data is selective. With different users enjoying the different views over the data. When the data is outsourced there is therefore the problem of enforcing possible data theft or data tampering by the inside employees or by the third party data auditors or it may be from any other form of internal or external threats.

In the data outsourcing scenario the data operators are under the strict custody of trusted party which monitors each access request to verify if it is compliant with the specified client or not.

This approach requires some additional measures to be considered. There is need for data owner (business organization) to manage access to legitimate users. To achieve this, owners can use digital signatures to identify the persons for whom they allow to access data. This actually leads to a system called notarization, which is been used by another system called validator to check for the data redundancy and data originality.

Outsourcing healthcare Insurance services is extremely popular today. However, there are several concerns being voiced about data security and adhering to standard quality norms. The Health Insurance Portability and Accountability Act (HIPAA) are widely acknowledged as the norm for healthcare services and Indian companies are well versed with the Act and other regulatory bodies. Some other standards/acts relevant for data security are:

- The Information Technology Act 2000 (ITA-2000),
- Payment Card Industry Data Security Standard (PCI DSS)

- ISO 27001, ISO27001 Information Security Standard

At present, few technologies used to protect sensitive data using Notarization are

In this paper we presented an approach where in the System we are providing Panel to the data owner where he/she can create a Digital signature and by using this digital signature he/ she can always notarize(authenticate the client during the performing each and every database transactions) the transactions. This Operation will be performing by a System called Validator to check the integrity of the data for each transaction. Legitimate users will be checked by the respective digital signature provided by corresponding data owner by the notarizer service. In this way, the confidentiality & integrity of information does not rely on an implicit assumption of trust on the server for on the legal protection offered by specific service contracts, but instead relies on the technical guarantees provided by Notarizer and validator.

Moreover in this project we are implementing one way MD5 hash key function to ensure data integrity.

Where an avalanche effect between master database and application data produces a tamper detection Scenario dynamically. Then a secured and perfectly weaved Forensic analysis System helps to find who, when and where of the tampered data.

The rest of the paper is organized as follows: section 2 will introduce about related work done so far. Section 3 will give proposed work, section 4 will give idea about evaluation and our approach, and section 5 gives result analysis process, and Section 6 will gives Conclusion, future work and also provides comparisons of algorithms through flow chart and in the end Section 7 provides Reference that we used.

2. RELATED WORK

Widespread news coverage of collusion between auditors and companies they audit[1], A recent FBI study indicates that almost half of attacks were by insiders [2].It is assumed that the notarization and validation services remain in a trusted computing base. This can be done by making them geographically and perhaps organizationally separate from the DBMS and the database [3], thereby effecting correct tamper detection even when the tampering is done by highly motivated insiders. scenario, like discusses tampering event in which in U.S., all patients are required to sign an authorization under HIPAA [4]. Computer forensics is now an active field, with more than 50 books published in the last 10 years. There are few computer tools for these tasks, in part due to the heterogeneity of the data. One substantive example of how computer tools can be used for forensic analysis is Mena's book [5]. Goodrich et al. introduce new techniques for using main-memory indexing structures for data forensics [6].

In the database context, previous papers introduced the approach of using cryptographic hash functions to detect database tampering [7] and of introducing additional hash chains to improve forensic analysis [7]. Previously, there has been proposed the Monochromatic, RGB, and Polychromatic forensic analysis algorithms [8].

If an adversary modifies even single byte of data or its timestamp, the independent validator will detect a mismatch with the notarized document, thereby detecting the tampering. The adversary could simply re-execute transactions, making whatever changes he/she wanted, and then replace original database with his/her altered one. However, the notarized document would not match in time. Avoiding tamper detection comes down to inverting the cryptographically strong one way hash function. An extensive presentation of an approach, performance limitations, tamper detection, threat model and other forensic analysis algorithms is discussed in paper[7],[9].Hash chain linking is discussed in more detail in paper[7].

Tiled bitmap algorithm is refinement of polychromatic algorithm. The advantage of the Tiled Bitmap Algorithm is that it lays down a regular pattern (a "tile") of such chains over contiguous segments of the database. The other advantage of the Tiled Bitmap Algorithm is that it can detect multiple corruption events that other previous algorithms can-not. On the other hand it suffers from false positives while the previous algorithms do not.

There are many models have been proposed to find the tamper detection process like

2.1 Monochromatic Algorithm

The Monochromatic Algorithm uses only the cumulative (black) hash chains we have seen so far, and as such it is the simplest algorithm in terms of implementation.

2.2 RGB Algorithm

In the RGB Algorithm, three new types chains are added, denoted with the colors red, green, and blue, to the original (black) chain in the so-called Monochromatic Algorithm. These hash chains can be computed in parallel; all consist of linked sequences of hash values of individual transactions in commit order. While additional hash values must be computed, no additional disk reads are required. The additional processing is entirely in main memory. The RGBY Algorithm retains the red, green, and blue chains and adds a yellow chain. This renders the new algorithm more regular and more powerful.

2.3 RGBY Algorithm

The RGBY Algorithm is an improvement of the original RGB Algorithm. The main insight of the previously presented Red-Green-Blue forensic analysis algorithm (or simply, the RGB Algorithm) is that during notarization events, in addition to reconstructing the entire hash chain (illustrated with the long right-pointed arrows in prior corruption diagrams), the Validator can also rehash portions of the database and notarize those values, separately from the full chain.

2.4 A3D Algorithm

The a3D Algorithm is the most advanced algorithm in the sense that it does not lay repeatedly a "fixed" pattern of hash chains over the database. Instead, the lengths of the partial hash chains change (decrease or increase) as the transaction time increases, in such as way so that at each point in time a complete binary tree (or forest) of hash chains exists on top of the database. This enables forensic analysis to be speed up significantly.

In all the above mentioned algorithms they differ in the amount of work necessary during normal processing . As

we seen in Monochromatic algorithm we use an array Black Chains of Boolean values to store the results of validation during forensic analysis.
Computing additional hash chains during periodic validation) and the precision of the when and what estimates produced by forensic analysis.
The Boolean results are indexed by the subscript of the notarization event considered: the result of validating is stored at a given index. Since we do not wish to pre-compute all this information, the validation results are computed lazily, i.e., whenever needed. This can give rise to corruption easily.
The RGBY Algorithm was designed so that it attempts to find more than one Corruption Event. However, the main disadvantage of the algorithm is that it cannot distinguish between three contiguous corruptions and two corruptions with an intervening notarization interval between them. The a3D Algorithm is working on the recursive pattern for the call of notarization service. Where if the Chain is having lager tree then it performs faster but fails to get desired result for all the intervals.

2.5 Tiled Bitmap Algorithm

This algorithm introduces the notion of a candidate set (all possible locations of detected tampering(s)) and provides a complete characterization of the candidate set And its cardinality. An optimal algorithm for computing the candidate set is also presented. Finally, the implementation of the Tiled Bitmap Algorithm is discussed, along with a comparison to other forensic algorithms in terms of space/time complexity and cost.
Where candidateSet Function is to arrange values of targeted binary array in reverse order and renumber function is to re arrange values of targeted binary array in perfect order.
So in our proposed System the DBMS computes a cryptographically strong one-way hash function for each tuple inserted and then notarizes it using a notarization service. This made it possible to check the consistency of the data by comparing it to the values stored with the notarization service. In continuation with this method, algorithms were designed to further analyze an intrusion of a database.

3. PROPOSED METHOD

In this section, we describe our approach of Tamper Detection and Forensic Analysis according to the steps shown in figure 2.As shown in figure there are 13 main steps in our approach.
Step 1, 2, and 3: In this approach we have taken a scenario whose architecture is shown in **Fig-1** where the data owner who outsourced his data at the 3rd party server is creating his digital signature by using any text file of him which consists of any secret document of his own. This signature created by using SHA-1 algorithm with DSA (Digital Signature Algorithm) produces a digital certificate file which is totally in unreadable format. This also produces a private key for verification.
Step 4, 5 and 6: In these steps we upload both the master data on which operations need to be performed & also a digital signature. This digital signature is created by using SHA-1 hash algorithm provide high featured security. This SHA-1 with DSA algorithm provides a strong digital signature which acts as a unique signature of the respective owner for the respective data which is to be placed & handled at the 3rd party server end by the employees & auditors.
By using uploading panel provided to the data owner in our web application owner will upload both master data and digital certificate which stores in a specific location of the web server of data ware house system.(This is the scenario which we designed and implement to perform tamper detection and forensic analysis process)
Step 7: Here in this step whenever a transaction is happening related to database then all the data field entering through the web application either by a inside employee or outside auditors by using their panel is also stored temporally in the java bean classes.
Step 8: The digital signature which is uploaded in the data ware house third party server by the data owner in step 4 acts as a notarizer element for every transaction carried out by the employees & auditors. Whenever a transaction is happening for each of those transactions the notarizer authenticates for the private key. If this private key is same as provided by the data owner then it notarizes the transaction positively. Else it denies the transaction.
Step 9: Validator is the system which actually brings the transacting details to the notarizer to validate the document or data.
Step 10: If the validator authenticates the data then a strong cryptographically one way hashing is performed by using MD5 algorithm which gives a sixteen byte hash value for both the master data & also for the transaction data either by the auditor or by the employee of the organization itself.
Step 11 and 12: In this step the MD5 hash value of the master data & the transaction data are both checked for avalanche effect. If the avalanche affect produces any positive changes then we consider there is some tampering is been happened at the transaction data then we note that data set as an infected one or a tampered one. In this way we are going to detect the data tampering for the numerous data owners for their respective data with the respective applications with their respective signatures. The proposed model provides an exact privacy ness for the data owner with their own private signatures so we can assure that in our taken scenario both the data and the tamper detection process will prevail for a long go.

We propose the above Discussed Tamper detection method by using following Algorithm:

1. Convert a text file into a Digital Signature using SHA-1 with DSA
2. Validating Digital Signature
3. **if** (Validates) **then**

Apply MD5 for Original and Transaction Data to get 16 byte hash keys

4. Checking for Avalanche Effect for both Hash keys

if(Avalanche effect) **then**

Data is tampered

else

Data is Safe

Step 13: In this step once the data tuple is considered as tampered then the forensic analysis operation is performed on this tuple to find who is the person tampered the data, when is tampering happened and what are the exact field where tampering happened.

Here in our proposed algorithm it accepts Master data set and transaction data set then we create a another set called D_{set} which actually consists of array of data field index whose value will be setting initially as "0". This indicates that the fields are not yet tampered.

Then for each master data set M_{set} and for each transaction data set T_{set} our algorithm takes each data fields of these two sets and compare both of them. If they are not equal then that data field is considered as tampered and then d_i that belongs to D_{set} is set to value "1". This way the complete tuple is keep checking for the exact tampered fields. Then this array of tampered fields is rearranged and also checking for more granularities to print the result.

The tampered person name can be identify using servlet which actually set the user name as he/ she login into the system and by using date and time operation on the same instance we can calculate the exactly at time data tampering is been happened.

We propose the above Discussed Forensic Analysis method by using following Algorithm:

// input: M_{set} is the set of master database
// T_{set} is the set of Transaction database
// D_{set} is the set of Data Field Index
// Un is Username
// td is date and time
// R_{set} is the set of Result
// output: R_{set} the set of Result
function forensicAnalysis (M_{set}, T_{set}, D_{set} ,Un, td, R_{set})
1: d_i =0 // index data field
2: $R_{set=}$ ""
3: for i= 1 to number of data fields
4: t_i data field of T_{set}
5: m_i data field of M_{set}
6: if ti and m_i are not equal then
7: d_i =1
8: end of for
9: for i= 1 to number of data fields
10: if d_i =1
11: R_{set} = R_{set} + d_i
12: Return R_{set}

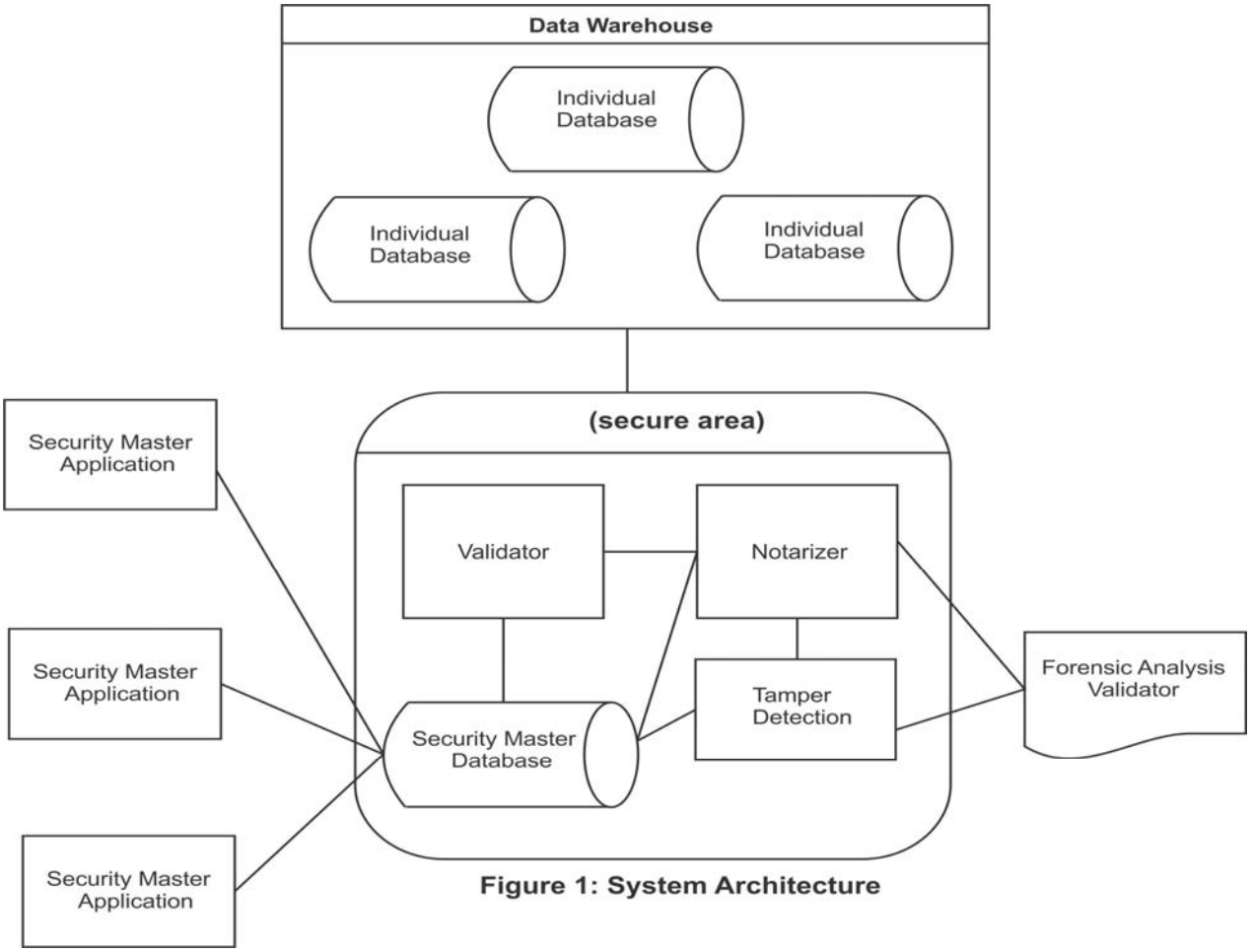


Figure 1: System Architecture

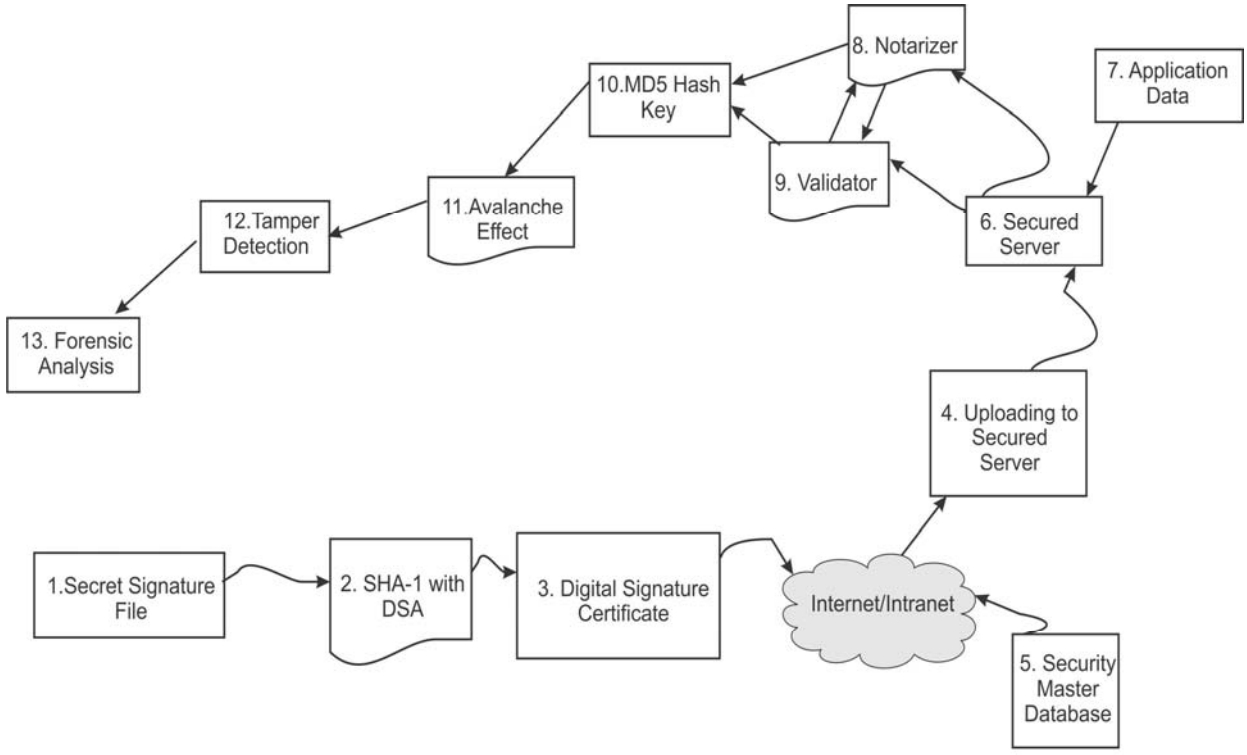


Fig-2 Overview of our Approach

Algorithm	Cost (Rs=1)
Monochromatic	O(lg(D/Iv))
RGB	O(D/Iv)
Tiled Bitmap	O ((D.lg Iv)/Iv+D)
Our Approach	O(log(D/Iv)) where Iv=1 and D=1

TABLE 1
Running Time Complexity of Forensic Analysis Algorithms

Algorithm	Cost (Rs=1)
Monochromatic	O(D)
RGB	O(D)
Tiled Bitmap	O (D.(1+lg Iv) / Iv)
Our Approach	O (D.(lg Iv) / Iv) Where D=1

TABLE 2
Worst Case Cost / space Complexity of Forensic Analysis Algorithms

THE RESULTS OF COMPARISING OF ALL ALGORITHMS WITH OUR APPROACH

Algorithm	Running Time	Cost	Space Complexity	Dynamic Performance Based on Hash Chains	Multiple Corruption Event
Monochromatic	Fast	High	More	No	No
RGB	Low	High	More	No	No
A3D	Low	Medium	Medium	No	Yes
Tiled Bitmap	Medium	Low	Less	No	Yes
Our Approach	Fast	Low	Less	Yes	Yes

4. EVALUATION AND APPROACH

4.1 Running Time Complexity of Forensic Analysis Algorithms

Table 1 shows the running time for three of the forensic analysis algorithms (the Polychromatic Algorithm is omitted because it is replaced by the Tiled Bitmap Algorithm) along with our approach. We assume that the spatial detection resolution Rs is equal to 1 for simplicity and D denotes the Number of Multiple corruption events. Observe that the algorithms become progressively slower because of the increased number of chains maintained and used during forensic analysis. The Monochromatic Algorithm, while being the fastest algorithm, suffers from the fact that only the first corruption event can be detected. As noted, the Tiled Bitmap Algorithm can be slightly optimized by retaining the cumulative chain of the Monochromatic in order to locate the first corrupted tile by performing binary search,

Although this refinement does not affect its asymptotic running time. When we compare our approach with all the three of above ours is faster as in our model we never going to do the multiple corruptions in post operation of corruption event. Because in our case we keep finding the corruption at each and every transaction so D value is 1 and also Iv that is validation interval is also 1 as we are performing operation at the same instance.

4.2 Worst Case Cost / space Complexity of Forensic Analysis Algorithms

Table 2 shows the cost for each of the forensic algorithms assuming a spatial detection resolution of one hour (Rs=1) and a single corruption event.

In this case, we observe the opposite trend compared to the one observed for the running times of the algorithms. For a sufficiently large validation interval IV , the Tiled Bitmap Algorithm has the smaller cost. This is because the ratio (1+lgIV)/IV Becomes less than one. When we compare smallest values of tiled bitmap algorithm with our approach, (lgIV)/IV yields even smaller value than tiled bitmap.So we can state that our approach is having smallest cost of all algorithms.

This quantification of cost also reflects the space complexity of the forensic algorithms since each of the contacts with the external notarization service corresponds to a hash value (of chains) which must be initially computed (and recomputed for comparison during validation) and maintained within the system. None of algorithms in Table 2 require extra space over the collection of hash values themselves.

5 RESULT ANALYSIS

As suggested in Tiled Bitmap algorithm [1] database tampering can be done for tiles here in our approach we are using dynamic data.

A robust and well-organized database was created to maintain the system as a whole and provide the central point of interaction for all tools in the system.

This database is easily expandable for future versions of this project. Three applications were created, one for each role, to allow users to interface with the Database. They provide an organized and controlled way for employees and auditors to interact with the system. The Complete System is integrated and developed using Java classes for Apache tomcat server edition. At this point, there are many ideas for additions that can be made to these applications; some ideas are outlined in the next section.

All in all, a large step was made toward securing databases from intrusion and the maintenance of such intrusions. By utilizing a central security master database as part of enterprise architecture for auditing, as well as role-specific GUIs, it is possible to efficiently manage the auditing of databases across an enterprise.

This auditing makes it possible to protect a database from both inside and outside intruders. By using this auditing system, businesses, government agencies, and other institutions can now know that their data is secure and safe from tampering.

5.1 Results for intrusion and Forensic Analysis.

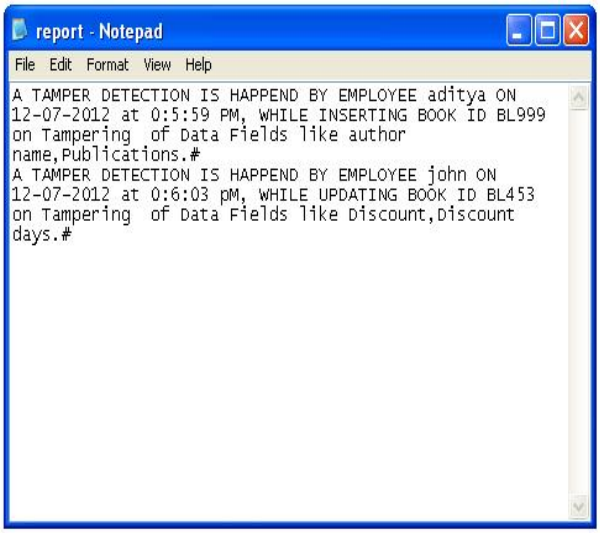


Figure 3: Results for Tamper detection and Forensic Analysis

As we Discussed in section 3 where we have taken the scenario of outsourcing data storage to the third party server and if the tampering is happened then its detected on the instance and forensic analysis is also been done and print the results in a notepad file. This result can be shown in below figure 3.

5.2 Performance Evaluation for Digital Signature Creation

As discussed previously in section no 3 that we provide a panel for data owner to create digital Signature certificate using secret text file by applying SHA-1 with DSA.So generally while creating digital signatures many of them using text file of bigger size, So if the system is perfect then it should not affect with its performance while creating signature and also while verifying signature as the system may have many number of users like employees and auditors of the organization, So to measure the performance of Digital signature creation we have taken the readings for time required for Signature Creation and verification verses the file size on which we are creating digital signature. This is shown below.

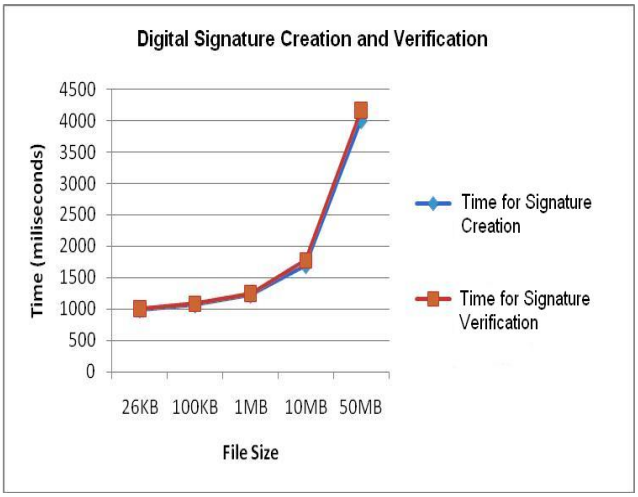


Figure 4:Signature Creation and Verification

5.3Performance Evaluation for Avalanche Effect

As discussed previously in section no 3 that we are hashing both master database tuple and application transaction data using MD5 hashing algorithm. Then to identify Tampered data tuple we check for the Avalanche effect of the MD5 hash values for both master database and application transaction data. As we know MD5 hash keys are 16 byte, So we taken the percentage probability of the avalanche effect of MD5 hash values, which can shown in figure 5.

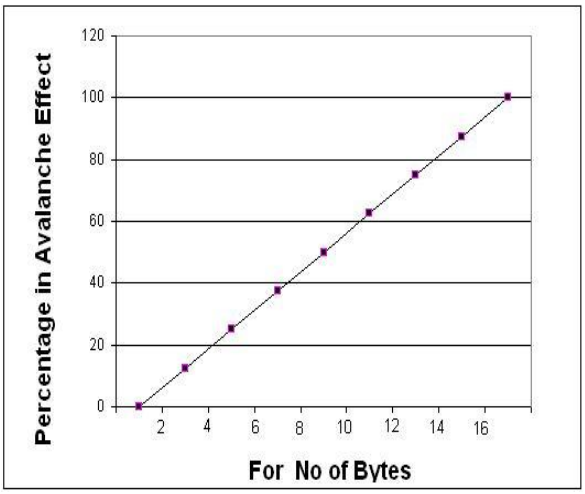


Figure 5:Percentage Probability of Avalanche Effect for MD5 Hash keys.

6 CONCLUSION, FUTURE WORK AND COMPARISON

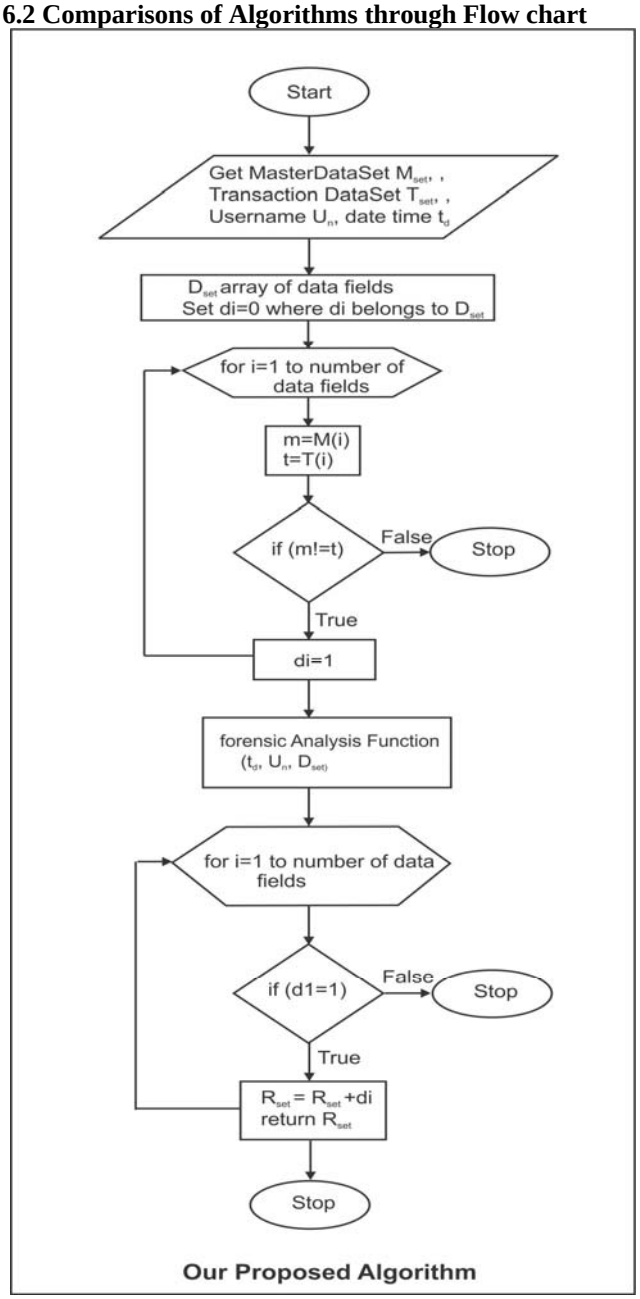
6.1 Conclusions and Future work

Forensic analysis commences when a crime has been detected, in this case the tampering of a database. Such analysis endeavors to ascertain when the tampering occurred, and what data were altered. The present paper expands upon that work by presenting the Tiled Bitmap Algorithm, which is cheaper and more powerful than prior algorithms. This algorithm employs a logarithmic number of hash chains within each tile to narrow down the when and what. Checking the hash chain values produces a binary number; it is the task of the algorithm to compute the pre image of bitwise We also note that previous algorithms do not handle multiple corruption events well, whereas the Tiled Bitmap Algorithm can independently analyze corruption events occurring both in different tiles and multiple corruption events occurring within a single tile.

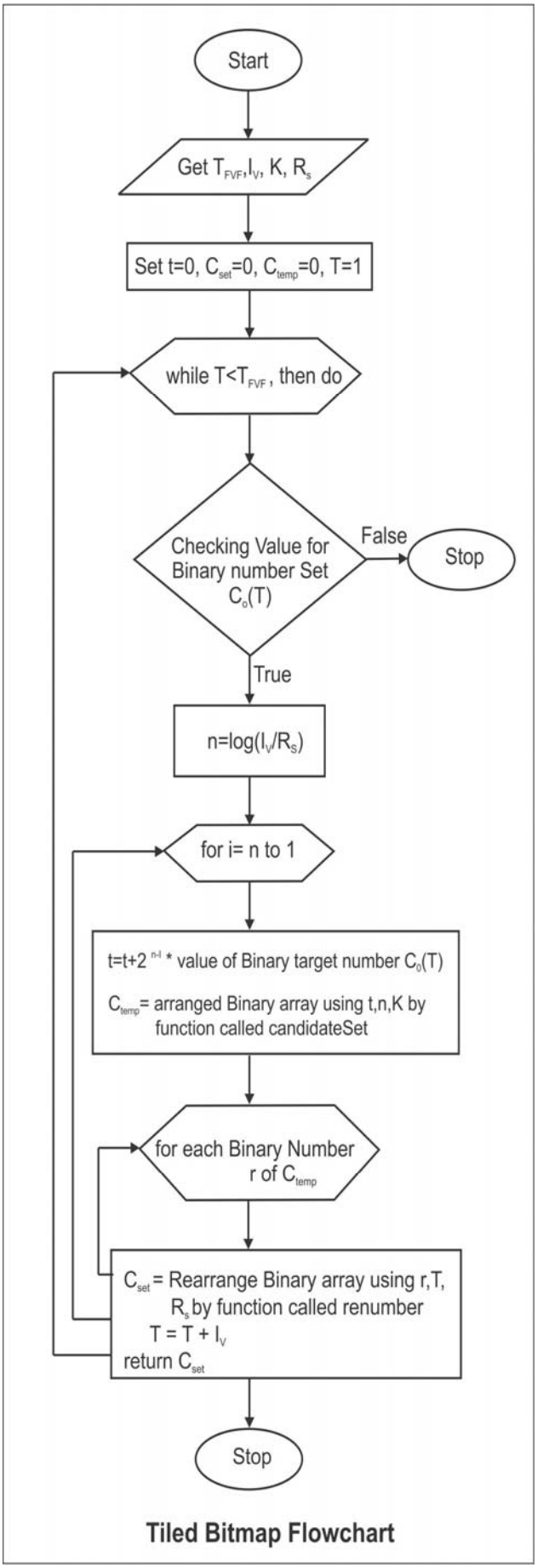
By creating a central database for all of the tools in the system to interact with it made it possible for the notarizer and validator to perform their operations successfully.

They can now store their data in this central database as well as use the information stored in it to schedule future executions. The three role-specific applications allow auditing or data updataion to be started on individual databases and then be maintained. The necessary tools for auditing a database are in place. It is now possible for Doctor's offices, companies, and government agencies to protect their information from threats by implementing this Enriched System.

6.2 Comparisons of Algorithms through Flow chart



6.2.1 Our approach flow



6.2.2 Tiled Bitmap Algorithm Flow

REFERENCES

[1] "The tiled bitmap forensic analysis algorithm", K.E. Pavlou and R.T. Snodgrass, IEEE transaction on knowledge and data engineering, Vol. 22, pp no.590-601, April 2010

[2] CSI/FBI, "Tenth Annual Computer Crime and Security Survey," http://www.cpppe.um.edu/Book store/Documents/2005CSISurvey.pdf, 2009.

[3] "An Infrastructure for Database Tamper Detection and Forensic Analysis", M. Malmgren, honors thesis, Univ. of Arizona, http://www.cs.arizona.ed u/projects/tau/tbdb/MelindaMalmgrenThesis.pdf, 2009.

[4] U.S. Dept. of Health & Human Services, The Health Insurance Portability and Accountability Act (HIPAA), http://www.cms.hhs. Gov/HIPAAGenInfo/, 2009.

[5] Investigative Data Mining for Security and Criminal Detection. J. Mena, Butterworth Heinemann, 2003.

[6] "Indexing Information for Data Forensics", M.T. Goodrich, M.J. Atallahand, and R. Tamassia, Proc. Conf. Applied Cryptography and Network Security, pp. 206-221, 2005.

[7] "Tamper Detection in Audit Logs", R.T. Snodgrass, S.S. Yao, and C. Collberg, Proc. Int'l Conf. Very Large Databases, pp. 504-515, Sept. 2004.

[8] "Forensic Analysis of Database Tampering", K.E. Pavlou and R.T. Snodgrass, Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 109-120, June 2006.

[9] "Forensic Analysis of Database Tampering", K.E. Pavlou and R.T. Snodgrass, ACM Trans. Database Systems, vol. 33, no. 4, pp. 1-47, Nov. 2008.