A MATHEMATICAL MODEL OF ACCESS CONTROL IN BIG DATA USING CONFIDENCE INTERVAL AND DIGITAL SIGNATURE

Amine RAHMANI, Abdelmalek AMINE and Mohamed Reda HAMOU

GeCoDe Laboratory, Department of Informatics,
Dr. TAHAR Moulay university of Saida – Algeria-
Aminerahmani2091@gmail.com, amine_abd1@yahoo.fr,
hamoureda@yahoo.fr

ABSTRACT

Nowadays, the concept of big data grows incessantly; recent researches proved that 90% of the whole data existed on the web had been created in last two years. However, this growing bumped by many critical challenges resides generally in security level; the users care about how could providers protect their privacy on their data. Access control, cryptography, and de-identification are the main search areas grouped under a specific domain known as Privacy Preserving Data Publishing. In this paper, we bring in suggestion a new model for access control over big data using digital signature and confidence interval; we first introduce our work by presenting some general concepts used to build our approach then presenting the idea of this report and finally we evaluate our system by conducting several experiments and showing and discussing the results that we got.

KEYWORDS

Access control, standard deviation, privacy preserving, big data, numeric signature, confidence interval

1. INTRODUCTION

Privacy, timeless, scalability of data is the most important problems that big data recognize starting from the first step of data acquisition; in fact, one of the most disturbed principle that are used in big data is the fact of losing control on data. This concept led to a lot of criticism from clients, losing control on your own data means losing everything related to the control even the access control.

Before the coming of the concept big data, controlling access on such data was done locally using the known models such as mandatory models (MAC), discriminatory models (DAC) or role based models (RBAC) but those last cannot be used because of some impediments; in case of DAC models the users defines the right access by himself while in the use of big data the user lose the entire control on his data; in case of MAC models the right access are defined by a major

Jan Zizka et al. (Eds) : ICAITA, SAI, CDKP, Signal, NCO - 2015
pp. 183–198, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51515

entity like military direction and this does not satisfy the users wishes in big data; moreover, in case of RBAC models, the right access are defines in form of roles where the can have the right from a major entity and can also give the rights on his own data to others which is bumped into reality of losing control. For that many works and propositions are passed by many researches such as in [20] and [7] using cryptography concepts and also in some of the works basing on users' identities.

In this report, we suggest a new model using some complex mathematical concepts such as standard deviation, confidence interval and primitive root to protect access control using users' identities and groups; for that we first introduce some backgrounds and definitions of the mathematical concepts that we practice, and so we introduce the main hypothesis under our good example, talking about its theoretical efficiency and carrying a set of experimentations on a set of information.

2. Related Works

The access control presents a sensitive domain in informatics security where it consists of defining such policy that allows or not for such user to get the access to such object; with the coming of concepts of big data and data sharing, this domain became a real challenge in research area. Many works are done within this highly active topic where the most of these works use a promising technique called Attribute Based Encryption such as in [32] [26] [29] and [17]; in [7] the author presented his approach of controlling hierarchical access using multiple key assignment in cryptography where he proposed four schemes, in other world four extensions of his work: bounded, unbounded, synchronous and asynchronous in order to give the general idea under temporal access control; in [2] the authors show their new approach of controlling access on resource-deprived environment in sensor data by integrating the Ladon Security Protocol that offers a secure access using end-to-end authentication, authorisation and key establishment mechanisms in PrivaKERB user privacy framework of KERBEROS environment; in [27] the authors introduced a purpose of using Elliptic Curve Cryptography (ECC) to control the access to data over sensor networks so that they presented their implementation of ECC in TelosB sensor network platform and evaluated their results by comparing it with the results of [18] and [19]; in [25] the paper is addressed to introduce the idea of SafeShare that consists of controlling the access by encapsulation of shared data so that their point of view consists of using the ABE to encrypt, encapsulate, audit and log the data in order to define a perform access control policy; other works go to the fact of using data content to control the access such as it is pointed out in [30] and [33].

3. Preliminaries

Before going far in our work, we like to give you a complete grasp about some general concepts that we used in this report.

3.1. Standard deviation

It is a mathematical concept that gives the measure of dispersion of a specific population starting from its mean which can be regarded as the average of the population's values, however, the standard deviation is linearly related to the multiplication of the individuals over the population space; the more the individuals are spread the higher is the deflection; the following formula is the used one to calculate the standard deviation of such population of size n.

$$S= \sqrt{\frac{\sum_{i=0}^{n}(X_i - \bar{X})^2}{n-1}} \qquad (1)$$

Where the $\bar{X}$ presents the mean that is calculated using the following formula

$$\overline{X} = \frac{\sum_{i=0}^{n} X_i}{n} \qquad (2)$$

And X_i is an individual of the population.

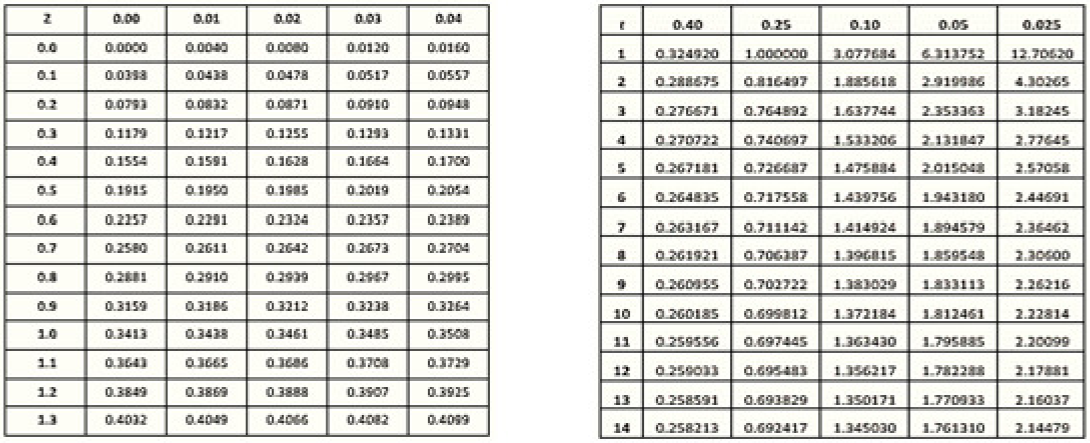
The standard deviation that is used in many cases such as in [16] where the authors proposed a new approach for selection of best threshold where the goal is to obtain better results for image segmentation and evaluated their results by comparing it with other conventional methods in term of several criterions such as the number of misclassified pixels; in [8] the authors proposed and evaluated a new query performance predictor for retrieval models using the standard deviation by testing several confidence levels; another use of standard deviation in information sciences is presented in [34] where the authors presented a standard deviation model to answer the problem of failure data in software reliability that presents a major problems in money costs and costumer satisfactions.

3.2. Confidence interval

It is an inferential statistical measurement that represents an interval of probability that such population individual can fall in basing on three essential parameters: the population's mean, the standard deviation and a specific percentage called confidence level. The confidence interval is calculated as follows:

$$CI= \text{mean} \pm \text{marge\_error} \qquad (3)$$

Where the merge error presents the remainder between the mean and the extremities of the interval, the equation used to compute the merge error has two different cases: the case of a sample which has a size less than 30 and the one which accepts a size more than 30, the initial difference resides in special value called t-value in the first case and z-value in the second one, these two values are pulled from two different tables as shown the figure 1 bellow:



z	0.00	0.01	0.02	0.03	0.04
0.0	0.0000	0.0040	0.0080	0.0120	0.0160
0.1	0.0398	0.0438	0.0478	0.0517	0.0557
0.2	0.0793	0.0832	0.0871	0.0910	0.0948
0.3	0.1179	0.1217	0.1255	0.1293	0.1331
0.4	0.1554	0.1591	0.1628	0.1664	0.1700
0.5	0.1915	0.1950	0.1985	0.2019	0.2054
0.6	0.2257	0.2291	0.2324	0.2357	0.2389
0.7	0.2580	0.2611	0.2642	0.2673	0.2704
0.8	0.2881	0.2910	0.2939	0.2967	0.2995
0.9	0.3159	0.3186	0.3212	0.3238	0.3264
1.0	0.3413	0.3438	0.3461	0.3485	0.3508
1.1	0.3643	0.3665	0.3686	0.3708	0.3729
1.2	0.3849	0.3869	0.3888	0.3907	0.3925
1.3	0.4032	0.4049	0.4066	0.4082	0.4099

a) A sample from Z-table

r	0.40	0.25	0.10	0.05	0.025
1	0.324920	1.000000	3.077684	6.313752	12.70620
2	0.288675	0.816497	1.885618	2.919986	4.30265
3	0.276671	0.764892	1.637744	2.353363	3.18245
4	0.270722	0.740697	1.533206	2.131847	2.77645
5	0.267181	0.726687	1.475884	2.015048	2.57058
6	0.264835	0.717558	1.439756	1.943180	2.44691
7	0.263167	0.711142	1.414924	1.894579	2.36462
8	0.261921	0.706387	1.396815	1.859548	2.30600
9	0.260955	0.702722	1.383029	1.833113	2.26216
10	0.260185	0.699812	1.372184	1.812461	2.22814
11	0.259556	0.697445	1.363430	1.795885	2.20099
12	0.259033	0.695483	1.356217	1.782288	2.17881
13	0.258591	0.693829	1.350171	1.770933	2.16037
14	0.258213	0.692417	1.345030	1.761310	2.14479

b) A sample from T-table

Figure. 1. A samples from t-table and z-table

However, the extraction of values from these two tables is different, meanwhile, in our work we use z-table because of our population has 2 000 individuals in which are divided into 10 groups each one has more than 150 individuals, the computation of error marge passes by two steps:

- Extracting z-value from the table; for that, we must compute the α/2 value where the α is the confidence level, let's get an example of confidence level of 90%, the α/2 value is 0.90/2= 0.45, after that we search the closest value in the table, we find 0.4495 and 0.4505, then for each one of these values we calculate the corresponding row + the corresponding column and we get 1.64 and 1.65, finally the z-value equals to (1.64+1.65) /2= 1.645

- Now we have the z-value, the merge error is calculated using the following formula:

$$\text{Error\_marge= z-value x } \frac{S}{\sqrt{n}} \quad (4)$$

Where the S is the standard deviation and the n is the size of the sample

Another value could be derived from the standard deviation called the standard error that represents the distribution of the sample and it is figured using the formula 5 as follows:

Standard error = $\frac{S}{\sqrt{n}}$ (5)

3.3. Primitive root

In informatics security the primitive root is an important concept used in several cases, especially in the case of sharing the keys in public key cryptography schemes; formally a primitive origin of a number P is the number that satisfies the following attribute:

r is a primitive root of P => □ i, j ∈ ℕ, if i ≠j than ri mod P ≠ rj mod P

Nevertheless, in mathematics there is no accurate way to compute a primitive root of a number, instead, there is a method to verify if such number r is a primitive origin of a number P as shown the following code:

```
Procedure isPrimitiveRoot (number r; number P)
Begin
Compute ℓ (P);
Decompose ℓ (p) to a set of prime factors
For each prime factor fi do
        Compute mi= $r^{\ell\ (P)/fi}$ mod P
If all mi ≢ 1 mod P & mi ≢ -1 mod P then r is primitive root of P
End.
```

The most known algorithm which uses the primitive root is the famous Deffie-Helman algorithm for sharing secrete keys because of his special characteristic that is known as discrete logarithm problem where it is proved that for a number r being primitive root of a number P, if we know r, P, and the result of ra mod P we could never conclude the number a

3.4. Hierarchical identification

The big data knows comes with real evolution not only in term of data volume, but also in term of number of users which makes the identification of them a crucial problem and implies the search for new technics of choosing identities; one of these technics is a promising and new method called Hierarchical Identification that aims to benefit from different information concerning the users such as their groups so that the identities are depending on these information using the concatenation process as shows the figure 2 bellow:

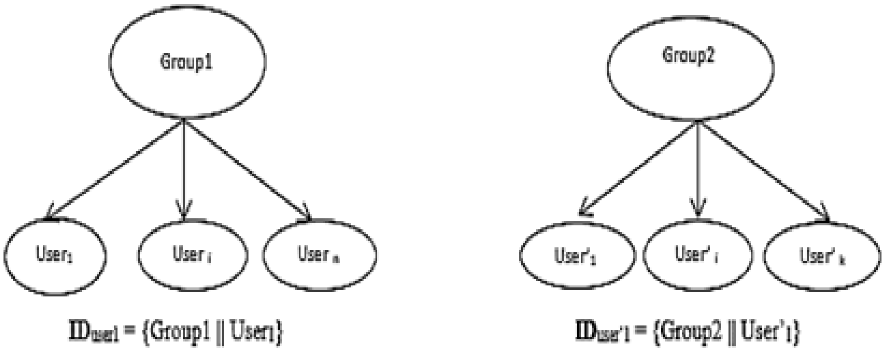


Figure. 2. Hierarchical Identification

This method has a major advantage resides in the ability of using the same identifier for multi users in different groups which allow the identification of big number of users with small size of identities and that can be useful in many cases that are related to the identification such as authentication mechanisms.

4. OUR APPROACH

Our advance is founded on three independent processes: defining access policy by computing the access control matrix and process of sharing the access rights.

4.1. Computing the access control matrix

This process is based, as the figure 3 shows above, on five steps: identification, normalization of identities, calculation of confidence interval for each group, calculation of digital signature for each user and ultimately determine the access rights by defining the matrix of access rights; in the remainder of this section we will detail each one of the stairs:

4.1.1. Identification

In this step, we target to get the identities of users utilizing the hierarchical identification mechanism in society to afford a standard configuration and size of the identities, we pass an address range of 10000 identities for each group using a concatenation operation between the group's ID where the user belongs and the genuine identity of the user, for example, let's consider a user with IDu= 0001 who belongs to a group which has the IDg= 01, the used identity of the user in our organization will be IDfinal= Edge || IDu= 010001= 10001.

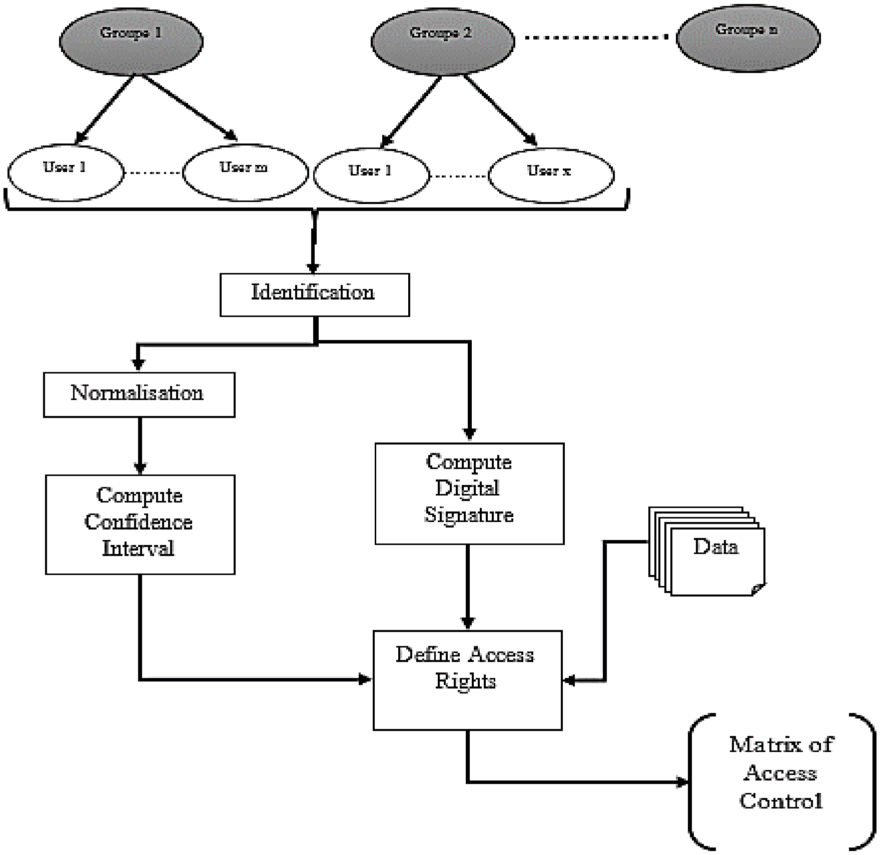


Figure. 3. Process of computing the access control matrix

4.1.2. Normalization

We can notice from the formula 4 that the standard deviation has a role in the process of computing the error marge, from its position in the formula, it's clearly shown that the error merge is linearly related to the standard deviation; otherwise, the more bigger is the standard deviation the more bigger is the error marge, by consequence, the more larger is the confidence interval, for that we suggest a normalisation of identities of the groups in order to create a less propagation rate in the range corresponded to each group so that instead of a maximum difference of 10000 between the extremities of a values of group, we diminish this value to 1 by dividing the identities by 10000. Getting hold of the example of a group in which the ID values go from 00001 to 10000, the normalized values go from 0.0001 to 1.0000; another normalization is used at the level of groups' sizes in order to preclude the influence of the great number of users in the group in the process.

4.1.3. Computing confidence interval

Our system defines for each group a specific confidence interval within the identity range of the group, this interval is estimated using various parameters, leading off from the standard deviation, and so setting the confidence layer, after that computing the merge error, finally utilizing the group means to get the final confidence level; all the computations in this step use the normalized values instead of the real ones.

4.1.4. Computing the digital signatures

This step is independent from the two precedent steps so that it can be executed in parallel with them, however, this step uses the concept of primitive root defined in the section 2.3 in such way that guarantees the unicity of signature for each user; to do that our system, firstly, generates for each group a big prime number P and find one of his primitive roots R, after that, for each user the generated signature equals to RIDfinal mod P; we choose this formula for two reasons: first, since R is primitive root of P then for two different users we will never get the same signature; and the second reason is that to protect the IDfinal of the user because of its major property known as Discrete Logarithm Problem cited in section 2.3, the IDfinal used in this step is in the real form and not the normalised one, thus, our system consists of generating the P as big prime because of two reasons too: first, there is no exact mathematical way to compute the primitive root but instead of that there is a way to verify if such number is primitive root as the procedure in section 2.3 shows using ℓ(P), so that we choose the P as prime to optimise the computation because ℓ(P) in this case equals to P-1; the second reason of choosing P as prime is also an optimisation reason because the researches proved that for P prime, we have a probability of 0.50 to generate a primitive root between 0 and ℓ(P).

4.1.5. Generation of access control matrix

This is the most important step of this process where the access rights are defined starting from the confidence interval of each group, the normalised identity of the user and his own signature; to do that our system conducts a set of tests for each group and each user by verifying if the normalised identity of a user belongs to the confidence interval of the group in order to know in which group the user belongs; once the system defines that, it start comparing the signatures of the data with the one of the user so that if are equals, the user will have full access and all the rights on the data that is considered as his own; else the user will have access on read only on the data that is considered in this case as shared data with him; for the other groups that the user doesn't belong, he will get no access right to their corresponding data; at the end of this process a matrix user x data is generated that resumes the access control policy.

The following code resume the process of creation of access control matrix by our system:

```
Algorithm AccessControlMatrix ( )
Input: $ID_g$ : groups' ID, $ID_u$: users' ID;
Output: M : matrix of access rights;
Begin
For each $group_i$ from IDg do
    For each $user_j$ from IDu that belongs to $group_i$ do
        $IDfinal_j$ ← $group_i$ || $user_j$;
        IDfinal←IDfinal+{$IDfinal_j$};
    End for;
End for;
For each $group_i$ from IDg do
    $sizegroup_i \leftarrow \frac{sizegroup_i}{2000}$
    $CI_i$ = Compute_confidence_interval ();
    Generate big prime number P and primitive root r
End for;
For each $user_j$ in IDfinal do
    $IDuser_j \leftarrow \frac{IDfinal_j}{10000}$
```

```
Compute $signature_j \leftarrow r^{IDfinal_j}$ mod P
End for;
M←new Matrix [number of users] [number of documents]
For each $group_i$ from IDg do
For each $user_j$ in IDfinal do
 If ($IDuser_j \in CI_i$) then
 For each $Document_k$ corresponds to $group_i$ do
  If ($signature_j = signature_k$) then
   M [j] [k] ← "read/write access";
  Else M [j] [k] ← "read only access";
  End if;
 End for;
 Else for each $Document_k$ corresponds to $group_i$ do
    M [j] [k] ← "no access right";
  End for;
 End if;
End for;
End for;
End.
```

4.2. Process of sharing the access rights

This procedure has been summed in order to answer some other problem that came to mind; what if such user decide to grant some other user to have write access right to his own data?, Otherwise, we aim by adding this process to allow for users of our system to share the same rights on same datum, to do that, our system uses Deffie-Helman algorithm of sharing cryptographic keys, but in our case to share the signatures between users; first of all, our system generates randomly a big prime number Q and a primitive root r then computed a primary signature for each of the users that will have the same access right S = $r^{ID_{user1}}$ mod Q that corresponds to user1, finally the system computes the final signature S_{final} = $S^{ID_{user2}}$ mod Q and signs the data with it. In this process, our system generates a new big prime Q and his primitive root and utilize them in the stead of the P that corresponds to the group where user1 belongs in order to protect the original signatures of the both users because it is used to sign other data that the users don't want to share rights with each other. The following process resumes this step:

```
Algorithm RightsSharing ()
Input $ID_{user1}$, $ID_{user2}$: users identities
Output $S_{final}$: final shared signature
Begin
Generate randomly a big prime Q and a number r < (Q-1);
While (r is not a primitive root of Q) do
Generate randomly a new r < (Q-1);
End while;
Compute S ← $r^{ID_{user1}}$ mod Q;
Compute $S_{final}$ ← $S^{ID_{user2}}$ mod Q;
Sign data with $S_{final}$;
End.
```

However, in this approach we choose to sign the data independently of its content, unlike the work presented in [30] because of two reasons: first, is to protect the privacy of data by perturbing the name in order to hide the real extension of the documents; and secondly, is to prevent problems of distrust like the famous one that Dropbox had recently because of its policy against violation of copyrights where a client claimed to the company from reading the content of his own data via Tweeter after the company prevent him from storing a document because of copyright violation[1].

5. EXPERIMENTS AND RESULTS

We take a set of experiments by building up a framework consists of 2000 users where each one has ten files stored in our system with a total of 20000 documents which gives an access control matrix of 2000 x 20000 that equals to 40 million right; the users are divided into 10 groups; this section is reserved for the introduction of a set of results using various parameters. But before going to the results, we will present the details of our dataset as shown in table 1

Table. 1. Dataset details used in our system

Group	Number of users	Range of identities	Range of normalised identities	Corresponding normalised Mean	Corresponding normalised standard deviation
Group 01	181	[0001…09999]	[0,0001…0,9999]	0.096	2.511
Group 02	234	[10000…19999]	[1,0000…1,9999]	1.100	3.377
Group 03	205	[20000…29999]	[2,0000…2,9999]	2.081	11.140
Group 04	209	[30000…39999]	[3,0000…3,9999]	3.020	23.819
Group 05	190	[40000…49999]	[4,0000…4,9999]	4.101	25.117
Group 06	184	[50000…59999]	[5,0000…5,9999]	5.098	25.649
Group 07	191	[60000…69999]	[6,0000…6,9999]	6.101	25.311
Group 08	221	[70000…79999]	[7,0000…7,9999]	7.096	23.672
Group 09	193	[80000…89999]	[8,0000…8,9999]	8.067	34.671
Group 10	193	[90000…99999]	[9,0000…9,9999]	9.098	34.774

As we notice in table 1, the mean is entirely related to the distribution of the values in their specific range and does not necessarily show the core of the range, and we notice also that the standard deviation is always out of the range of the sample because of the use of the ability of two during his computation.

5.1. Results

We carry on a set of comparisons organized in two steps: first, we confront a comparison between domains in order to study the influence of the distribution of the sample in our approach, secondly, we study the effect of choosing the confidence level in our approach, and finally, we evaluate our system by comparing it with other conventional work.

[1] http://assoquebecois.com/2014/04/01/dropbox-clarifie-sa-politique-sur-lexamen-des-dossiers-partages-pour-les-questions-dmca/

The next table shows the result of average of the access rate by domain in many experiments on normalized identities and real ones.

Table. 2. Results of comparison between groups in term of normalised and not normalised values and distribution of samples

Group	Standard error (distribution of sample)	Average of Access rate (%)	
		Normalized values	Not normalized values
Group 01	0.187	68.04	39.66
Group 02	0.220	77.48	89.00
Group 03	0.778	98.53	82.33
Group 04	1.647	97.12	114.33
Group 05	1.822	100.00	135.00
Group 06	1.891	100.00	152.66
Group 07	1.831	100.00	321.50
Group 08	1.592	100.00	341.00
Group 09	2.495	122.27	368.00
Group 10	2.503	119.42	364.00
Average	1.497	98.28	200.74

As the table 2 shows, the case of using normalised values gives better results than the one of not normalised values, however, we can see that the access policy exceed the limits in two groups: the Group 09 with an average of 122.27% (about 239 user could access to data instead of 193) and Group 10 with an average of 119.42% (230 users could access to data). Meanwhile the groups 05, 06, 07, and 08 give the best result with no error; meanwhile, the standard error is relatively related to the value of the standard deviation and that's what influence the value of access rate. The more the standard error is big, the more the number of authorised users to access is big. As the results indicates that the normalised values' case is widely better than the other case, we will in the rest of this section focus our experiments in only the normalised case.

In table 4 below, we will detail the results of the admission rate by group in term of the chosen confidence level in which we used many confidence levels and we take the ones that give an excited results in some of our groups, the following board indicates the chosen confidence levels with the corresponding z-value of each ace.

Table. 3. Corresponding z-value for each chosen confidence level

Confidence level	20%	28%	28%<<29%	30%	31%
z-value	0.255	0.355	0.365	0.375	0.385

Table. 4. Access rate by domain in term of chosen confidence level

Confidence level / Domain	20%	28%	28%<<29%	29%	30%	31%
Group 01	48.62	68.51	70.72	71.82	72.36	76.24
Group 02	55.55	78.20	79.91	81.20	84.19	85.90
Group 03	98.54	98.54	98.54	98.54	98.54	98.54
Group 04	97.13	97.13	97.13	97.13	97.13	97.13
Group 05	100.00	100.00	100.00	100.00	100.00	100.00
Group 06	100.00	100.00	100.00	100.00	100.00	100.00
Group 07	100.00	100.00	100.00	100.00	100.00	100.00
Group 08	100.00	100.00	100.00	100.00	100.00	100.00
Group 09	99.48	111.39	122.80	140.41	165.80	193.78
Group 10	100.00	100.00	108.81	124.87	136.79	146.63

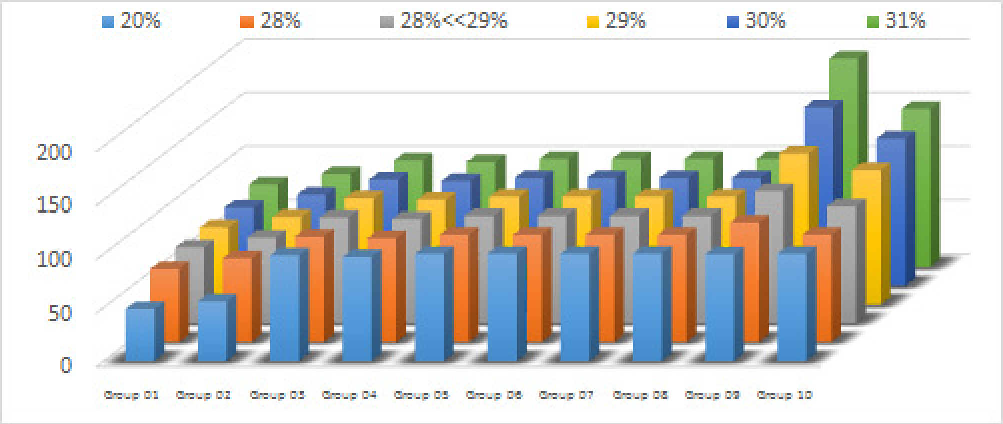


Figure. 4. Access rate by domain in term of chosen confidence level

As the table 4 and figure 4 show, each one of the confidence levels that we choose presents some good results in some groups and in the same time bad results in other groups; the best confidence level for groups group 01 and group 02 is 31% with rate of access of 76.24% in group 01 (138 user from 181) and 85.90% for group 02 (202 users from 234) while this level presents the worst results in programming group with 193.78% of access rate (374 users from 193 authorised), instead of that, the groups programming and security gives best results with less level of confidence using 20% of confidence level with 99.48% for programming (192 users from 193 authorised) and full access rate without error for security; meanwhile, the other groups such as data mining and natural sciences gives excited results without been influenced of the value of confidence level.

The following table presents the results of average of access rate and error rate between domains in term of variation of confidence level in normalised values where the positive value of error rate means that there is less users have access than the authorised ones and negative value means that there is more users that have access than the authorised ones, otherwise, the positive error rate means that there are users who must have access but our system doesn't allow to them to get access while the negative value means that there are some users must not access to data but our system allows to them to have access.

Table. 5. Results of average of access rate and error rate in term of confidence level variation

Confidence level (%)	Average of access rate (%)	Error rate (%)
20	89.92	10.08
28	95.37	4.63
28<<29	97.73	2.27
29	101.39	- 1.39
30	105.47	- 5.47
31	109.81	- 9.81

From the table 6 we can clearly notice that the confidence level between 28% and 29% gives better results with an average rate of access about 97.73% even if it represents some weaknesses in the last two domains where the access rate exceeds the 100 % (122.79% for 9 domain (44 unauthorized users), and 108.29% for 10 (16 unauthorized users)), the reason of why we didn't determine the exact value between 28 and 29 is that because all values within this range gives the same z-value which is about 0.365. The use of confidence level equals to 20% presents a major advantage because of all the values of access rate doesn't exceed the 100%, which means for all data there is no unauthorized access while it presents the worst result in term of error rate with more than 10% of authorized users could not access to data that must get access to because of the less access rate in the first domains where only 48.61% (only 88 users) of authorized users could access in 1 domain and 55.55% (only 130) could access in 2 domain. So, as the table shows, once we defined a confidence level starting from 29% the results became more badly (average of 28 of unauthorized users could access to data for 29%, 110 to 30%, and 197 to 31%).

After introducing a set of outcomes using a variation of parameters, we put our system in confrontation with a set of conventional works in the image of the system presented in [18] named TinyECC, and the one shown in [27] under the name ECC-AC in term of time of generating a signature and time of verifying the signature, however, our system generates a signature of average of size of 128 bits because of the role of a prime number of sizes of 2048 bits and primitive root of 1024 chips; the following table introduces the effects of time of generation and verification of signatures

Table. 6. Comparison of time of generation and verification of signatures

Approach	Time of generation of signature (s)	Time of verification of signature (s)
DSCI-AC	0.031	0.073
ECC-AC	3.35	6.78
TinyECC	8.24	16.28

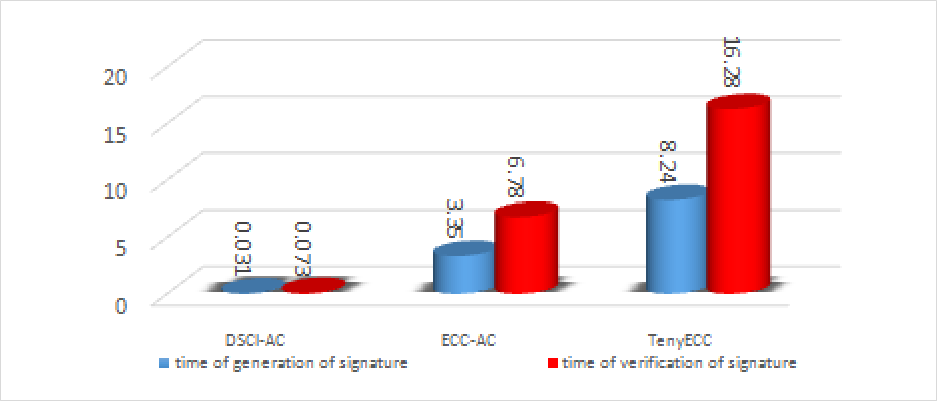


Figure. 5. Results of comparison of time of generation and verification of signatures

As the table 6 and figure 5 indicate, our system doesn't take much time for generating or verifying the signatures and that's due to the fact of using a simple computations to generate signatures and also the signature are used to perturb the data via their names which make its verification low costs because the system doesn't need to treat the data content to have signature, thus, in our approach only the server is the responsible of computing and verifying the signatures which leads us to eliminate the time of connection for users. In the other hand; we notice clearly that the time of generation of a signature is less than the one of its verification because the generation is based only on one and one only operation while the verification takes more time because of the number of operations resides on searching if the user belongs to the group in order to define if he has already the access or not then compare the two signatures to define the right that he has.

5.2. Limitations and weaknesses

As a security issue, our scheme could not present a safety sure state, nonetheless, our scheme has some restrictions that we could resume below:

- Confidence level: the use of a unique confidence level for all groups presents a limitation as shows the results above where the same confidence level prevent some authorized users from accessing their shared data in some groups and allows in the same time unauthorized users to access to data in other groups.

- Causing the server the only one who can generate and deploy signatures, making it easy to take on the role of man in the middle if we count the server as honest attacker, in fact, the server knows every signature used in the organization which allows for it to have broad access to all information, and that may be considered as assault of privacy requirements at some higher degrees of protection.

6. CONCLUSION

In this paper, we ushered in a new glide path of applying digital signature and the confidence interval in order to answer three essential questions: how could we control the approach to information that we don't hurt even the control on?, To answer it, we first divided the users into groups by their domains then compute for each group its own confidence interval that we used in our system in parliamentary procedure to ascertain who has access to data and who doesn't, after determining which user has access to the data, another question came to mind; for the users who have access to data, which right should they have, is that full access or read only access? We answered this question by using the digital signature generated using another mathematical concept called primitive root basing on prime numbers and random theories in order to precisely which access right each user must take; then by assisting these two questions we could define the last access control matrix; the final question that we answered in this study is that if such user decide to afford full access on his data for another user, how could we ensure that? To respond that we offered the use of Deffie-Hellman algorithm of sharing cryptographic keys in order to permit users to partake in the same signature by consequence have the same access right on the same data.

As future work, we will usher in new models of using meta-heuristics technics to improve the results of this work by searching for the appropriate assurance level for each group we also will give other models using cryptography whose purpose is to prevent the server from recognizing the genuine signatures of the users. In the final stage, it only remains to mention that the security in Big Data is all grounded on trust then that no trust no security.

REFERENCES

[1] Arunkumar, S., Raghavendra, A., Weerasinghe, D., Patel, D., & Rajarajan, M. (2010, October). Policy extension for data access control. In Secure Network Protocols (NPSec), 2010 6th IEEE Workshop on (pp. 55-60). IEEE.

[2] Astorga, J., Jacob, E., Huarte, M., & Higuero, M. (2012). Ladon 1: end-to-end authorisation support for resource-deprived environments. Information Security, IET, 6(2), 93-101.

[3] Altman, D. G., & Bland, J. M. (2005). Standard deviations and standard errors.Bmj, 331(7521), 903.

[4] Bagheri, E., Babaei, S., & Khayyambashi, M. R. (2009, August). A new method for consistency of access control in web services. In Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on (pp. 567-569). IEEE.

[5] Camenisch, J., Mödersheim, S., Neven, G., Preiss, F. S., & Sommer, D. (2010, June). A card requirements language enabling privacy-preserving access control. In Proceedings of the 15th ACM symposium on Access control models and technologies (pp. 119-128). ACM.

[6] Chen, Y. R., Chu, C. K., Tzeng, W. G., & Zhou, J. (2013, January). Cloudhka: A cryptographic approach for hierarchical access control in cloud computing. InApplied Cryptography and Network Security (pp. 37-52). Springer Berlin Heidelberg.

[7] Crampton, J. (2009). Cryptographically-enforced hierarchical access control with multiple keys. The Journal of Logic and Algebraic Programming, 78(8), 690-700.

[8] Cummins, R., Jose, J., & O'Riordan, C. (2011, July). Improved query performance prediction using standard deviation. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (pp. 1089-1090). ACM.

[9] Curé, O., Kerdjoudj, F., Le Duc, C., Lamolle, M., & Faye, D. (2012, September). On the potential integration of an ontology-based data access approach in NoSQL stores. In Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on (pp. 166-173). IEEE.

[10] Di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S., & Samarati, P. (2007, September). Over-encryption: management of access control evolution on outsourced data. In Proceedings of the 33rd international conference on Very large data bases (pp. 123-134). VLDB endowment.

[11] Guo, J., Baugh, J. P., & Wang, S. (2007). A group signature based secure and privacy-preserving vehicular communication framework. Mobile Networking for Vehicular Environments, 2007, 103-108.

[12] Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006, October). Attribute-based encryption for fine-grained access control of encrypted data. InProceedings of the 13th ACM conference on Computer and communications security (pp. 89-98). ACM.

[13] Keathley, E. F. (2014). Big Data and Bigger Control Issues. In Digital Asset Management (pp. 99-115). Apress.

[14] Kelani Bandara, K. B. P. L. M., Wikramanayake, G. N., & Goonethillake, J. S. (2007, August). Optimal selection of failure data for reliability estimation based on a standard deviation method. In Industrial and Information Systems, 2007. ICIIS 2007. International Conference on (pp. 245-248). IEEE.

[15] Khalil, I., Khreishah, A., & Azeem, M. (2014). Consolidated Identity Management System for secure mobile cloud computing. Computer Networks,65, 99-110.

[16] Li, Z., Cheng, Y., Liu, C., & Zhao, C. (2010, March). Minimum Standard Deviation Difference-Based Thresholding. In Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on (Vol. 2, pp. 664-667). IEEE.

[17] Li, M., Yu, S., Zheng, Y., Ren, K., & Lou, W. (2013). Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. Parallel and Distributed Systems, IEEE Transactions on, 24(1), 131-143.

[18] Liu, A., & Ning, P. (2008, April). TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. In Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on (pp. 245-256). IEEE.

[19] Malan, D. J., Welsh, M., & Smith, M. D. (2004, October). A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on (pp. 71-80). IEEE.

[20] Miklau, G., & Suciu, D. (2003, September). Controlling access to published data using cryptography. In Proceedings of the 29th international conference on Very large data bases-Volume 29 (pp. 898-909). VLDB Endowment.

[21] Ortiz, P., Lazaro, O., Uriarte, M., & Carnerero, M. (2013, June). Enhanced multi-domain access control for secure mobile collaboration through Linked Data cloud in manufacturing. In World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a (pp. 1-9). IEEE.

[22] Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Sensing as a service model for smart cities supported by internet of things. Transactions on Emerging Telecommunications Technologies, 25(1), 81-93.

[23] Shtok, A., Kurland, O., & Carmel, D. (2009). Predicting query performance by query-drift estimation. In Advances in Information Retrieval Theory (pp. 305-312). Springer Berlin Heidelberg.

[24] Stevens, G., & Wulf, V. (2009). Computer-supported access control. ACM Transactions on Computer-Human Interaction (TOCHI), 16(3), 12.

[25] Thilakanathan, D., Calvo, R., Chen, S., & Nepal, S. (2013, December). Secure and Controlled Sharing of Data in Distributed Computing. In Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on(pp. 825-832). IEEE.

[26] Tu, S. S., Niu, S. Z., Li, H., Xiao-ming, Y., & Li, M. J. (2012, May). Fine-grained access control and revocation for sharing data on clouds. In Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International (pp. 2146-2155). IEEE.

[27] Wang, H., Sheng, B., & Li, Q. (2006). Elliptic curve cryptography-based access control in sensor networks. International Journal of Security and Networks, 1(3), 127-137.

[28] Wang, Z. H., Zhi, S. S., & Liu, H. M. (2012, July). MSHS: The mean-standard deviation curve matching algorithm in HSV space. In Machine Learning and Cybernetics (ICMLC), 2012 International Conference on (Vol. 3, pp. 1064-1069). IEEE.Yang, Y., & Zhang, Y. (2011, September). A generic scheme for secure data sharing in cloud. In Parallel Processing Workshops (ICPPW), 2011 40th International Conference on (pp. 145-153). IEEE.

[29] Yu, S., Wang, C., Ren, K., & Lou, W. (2010, March). Achieving secure, scalable, and fine-grained data access control in cloud computing. InINFOCOM, 2010 Proceedings IEEE (pp. 1-9). Ieee.

[30] Zeng, W., Yang, Y., & Luo, B. (2013, October). Access control for big data using data content. In Big Data, 2013 IEEE International Conference on (pp. 45-47). IEEE.

[31] Zhang, X., Liu, C., Nepal, S., Dou, W., & Chen, J. (2012, November). Privacy-Preserving Layer over MapReduce on Cloud. In Cloud and Green Computing (CGC), 2012 Second International Conference on (pp. 304-310). IEEE.

[32] Yang, Y., & Zhang, Y. (2011, September). A generic scheme for secure data sharing in cloud. In Parallel Processing Workshops (ICPPW), 2011 40th International Conference on (pp. 145-153). IEEE.

[33] Nabeel, M., Bertino, E., Kantarcioglu, M., & Thuraisingham, B. (2011, October). Towards privacy preserving access control in the cloud. InCollaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on (pp. 172-180). IEEE.

[34] Bandara, K., Wikramanayake, G. N., & Goonethillake, J. S. (2007, August). Optimal selection of failure data for reliability estimation based on a standard deviation method. In Industrial and Information Systems, 2007. ICIIS 2007. International Conference on (pp. 245-248). IEEE.