# User Location Time and Entropy (ULTE) based Salt generation for Password Based Key Derivation Function (PBKDF) in Cloud Computing

Md. Alam Hossain, Ahsan-Ul-Ambia, Md. Al-Amin, Nazmul Hossain

**Abstract**— Virtualization is the biggest issue in our modern life with information technology. For accessing data globally and timely people are storing data, information, and programs into Cloud Storage Space. Cloud service providers (CSPs) maintain authentication and authorization mechanism but there are no mechanisms for data security. If the users' credentials are compromised then information could be misused, or stolen. Client Side Encryption Tool (CSET) is used widely with their own features and security schemes to overcome the security problems in CSPs. CSET encrypts each data file of client's with encryption key which are derived by a specialized algorithm called Password Based Key Derivation Function (PBKDF), and Salt, a random value plays vital role for making each encryption key unique and different from other keys. Through analysis, we found that: generation of Salt by using any Pseudo random number generation algorithms could lead the intruders like hackers to crack the encryption keys for multiple files stored in CSPs. Therefore, a new random number generate algorithm (ULTE) for Salt generation which is based on users' Location Information, Time Information, and Entropy Data (bit strings of system noise sources like thermal noise or HDD seek time) is proposed. Simulation result shows the whole processes of Salt generation according to the ULTE Salt Generation Algorithm.

**Index Terms**—Cloud Computing, Cloud Service Providers, Client Side Encryption Tools, Encryption, Decryption, Encryption Key, Decryption Key, PBKDF, Location, Latitude, Longitude, Entropy, Thermal Noise, Dropbox, OneDrive, Google Drive, CloudMe.

## 1 Introduction

In the past, physical local computers or servers are used by the people to store information and run applications software or programs, whereas cloud computing allows people to do the same kinds of activities through Internet. Cloud Computing enhances efficiency of system performances, helps improve cash flow and offers many more benefits such as Flexibility, Disaster Recovery, Automatic Software Updates, Capital-Expenditure Free Increased Collaboration, Work from anywhere, Document Control, Security, Competitiveness, and Environmentally Friendly working environment and etc. Cloud Computing is a set of combination of hardware, networks, applications software, storages and interfaces. Cloud Computing services are furnished by third party organizations like Dropbox, OneDrive, Google Drive, CloudMe etc [1-15]. Today people using cloud computing mainly for storing information and application programs services where application programs services are not provided sufficiently by the all cloud service providers.

For authentication and authorization, usernames and passwords are mandatory for users, and usernames and passwords are picked by users during account opening at first time use. Cloud service providers strongly maintain the policies during opening a new account with username and password. Among the service providers few service providers offer also two-way authentication mechanism for ensuring more security where a token or one time temporary PIN or Password is sent with a short life time to user's mobile number or e-mail address just after login into account with username and password. Although CSPs provide authentication and authorization, as well as two-way authentication mechanism for ensuring users security, but there is a great problem with data security since no mechanisms are provided by the CSPs for keeping data files secured. Cloud Service Providers store users' data files in original form. If a user's username and password as well as two-way authentication token are compromised by the intruders like malicious users or hackers then the data files may be stolen and could be used maliciously. As all the communications are occurred in a real time communication system which means through Internet, users' credentials' could be compromised easily by the hackers with special kinds software like malware [ 16-33].

- *Md. Alam Hossain is currently serving as Chairman in Computer Science and engineering Department in Jessore University of Science & Technology, Bangladesh, E-mail:* alamcse_iu@yahoo.com
- *Ahsan-Ul-Ambia is currently serving as Professor in Computer Science and engineering Department in Islamic University, Kushtia, Bangladesh.E-mail:* ambia@cse.iu.ac.bd
- *Md. Al-Amin is currently serving as Lecturer in Computer Science and engineering Department in Jessore University of Science & Technology, Bangladesh. E-mail:* malamin.ali@gmail.com
- *Md. Jahangir Alamis currently serving as Manager inInternal IT Audit Department in NRB Commercial Bank Limited, Dhaka, Bangladesh. E-mail:* jahangircsebd@gmail.com
- *Nazmul Hossain is currently serving as Lecturer in Computer Science and engineering Department in Jessore University of Science & Technology, Bangladesh, PH-042172058. E-mail:* nazmul.justcse@gmail.com

For ensuring data security and avoiding these kinds of security threats new sorts of software called Client Side Encryption Tools (CSETs) are developed. CSETs are widely used by clients with their security schemes and own multiple useful and essential features [34-45]. Client Side Encryption tool is application software which encrypts the data files through the process called Encryption to be uploaded in the CSPs storage where the original data files converted to a special form called cipher-text data files. During encryption the data files are encrypted with encryption algorithms and encryption keys and security of encrypted data files are mainly dependent on these encryption algorithms and encryption keys. Cipher-text data files are not useable or understandable to anyone without decrypting the data files through Decryption. Decryption performs the opposite operation of encryption process that does the translation the cipher text into the plain text data files or original data files. Decryption process depends on the decryption algorithms and decryption keys. So, the client side encryption tools are mainly used for encrypting user data files and uploading the encrypted data files in the cloud storage space and also downloading the data files from the cloud and decrypting the encrypted data files to the original data files.

Encryption and Decryption keys are generated and maintained in Client Side Encryption Tools with a special algorithm called Password-Based Key Derivation Functions (PBKDF) [46-48]. In PBKDF no keys are generated directly from any random generator. Since it is vulnerable making keys using random generator functions and which helps hackers cracking keys. For enhancing randomness and making more secure each key generated from user's credential password with additional input. Each PBKDF is defined by a Pseudorandom Function (PRF) and a fixed iteration count. A password, an uncertain random number called salt, and an indication of the desired length of the key in bits are included as input for the execution of PBKDF. Guessing user's password could increase the key cracking probability but not completely cracking. Hackers have to put unlimited time for cracking key which made from PBKDF. A key will be cracked accurately if Pseudorandom Function (PRF), number of iteration count, password, salt and desired length of the key are compromised. If one of the above parameters is not compromised then it would be very difficult key cracking. All the parameters of PBKDF are important for making unique and random key. Except salt other parameters could be guessed since salt is a random number input. So salt should be generated as true random number.

Salt has a vital role for making key unique and random. No pseudorandom generator should be used for making salt because it could help hackers for cracking key. Inputs should be from truly unguessable sources [49]. For this we propose an algorithm in this paper for generating Salt by using client's real time environmental information which is User Location Information, User Time Information, and Entropy Data at key generation time where Entropy Data is measured from thermal noise or HDD seek time or any kind of noise sources. A mobile user change location frequently and which could be a great source for making salt uncertain. Also time at key generation make salt harder for cracking. Finally entropy sources like thermal noise or HDD seek time make salt true random number.

The remaining part of this paper is structured as following. Section 2 discusses about Password-Based Key Derivation Functions (PBKDF). Section 3 presents Terrestrial Coordinate System. Section 4 describes about Entropy. Section 5 derives ULTE Algorithm, Section 5.1 describes Environment Record Information for Salt, Section 5.2 shows the Salt Generation Process, and Section 5.3 describes the Pseudo Code of ULTE Salt Generation Algorithm. Section 6 tabulates the Simulation results. And finally in Section 7 the decision of this research work is made and future works are mentioned.

## 2 Password-Based Key Derivation Functions (PBKDF)

For gaining access to a restricted resource or system a user has to choose a password or a passphrase which is a secret string of characters. Most of the users usually use their name, date of birth, phone number, birth place, pet name, and others as passwords or passphrase that are easily vulnerable by social engineer. User chosen passwords have weak statistical randomness properties and if these passwords used as cryptographic keys directly then this weakness leads hackers to crack or guess the cryptographic keys easily. There are some situations such as ensuring security in storage devices; the password or passphrase may be the only secret information that is available to the cryptographic algorithm to protect the data from malicious users or hackers.

Key Derivation Functions (KDFs) are deterministic algorithms that are used to derive cryptographic keying material from a user's password or passphrase [49]. Password-Based Key Derivation Functions (PBKDF) is a well-known and widely used algorithm for generating cryptographic keys by using KDFs. Each PBKDF [45-47] in the family is defined by the choice of a Pseudorandom Function (PRF) and a fixed iteration count, denoted as C. The input to an execution of PBKDF includes a password, denoted as P, a salt, denoted as S, and an indication of the desired length of the MK in bits, denoted as kLen. The kLen value shall be at least 112 bits in length. A generic diagram of the PBKDF is given in (Fig.1), and symbolically:

$$\boldsymbol{mk}\mathbf{=\ PBKDF}_{(\mathbf{PRF,\ C})}\ (\boldsymbol{P},\ \boldsymbol{S},\ \boldsymbol{kLen}) \qquad \textbf{(1)}$$

The main idea of a PBKDF is to slow dictionary or brute force attacks on the passwords by increasing the time needed to test each password. An attacker with a list of likely passwords can evaluate the PBKDF using the known iteration counter and the salt. Generation of Salt by any pseudo random number generation algorithms make it easy for malicious users for cracking encryption keys. In this paper we propose an algorithm for generating Salt by using client's real time environmental information which includes User Location Information, User Time Information, and Entropy Data at the key generation time.

Since an attacker has to spend a significant amount of computing time for each try, it would become harder to apply the dictionary or brute force attacks. Randomly-generated passwords generally have much higher security strength than user-chosen passwords of the same length. The strength of a password is related to its length and its randomness properties.

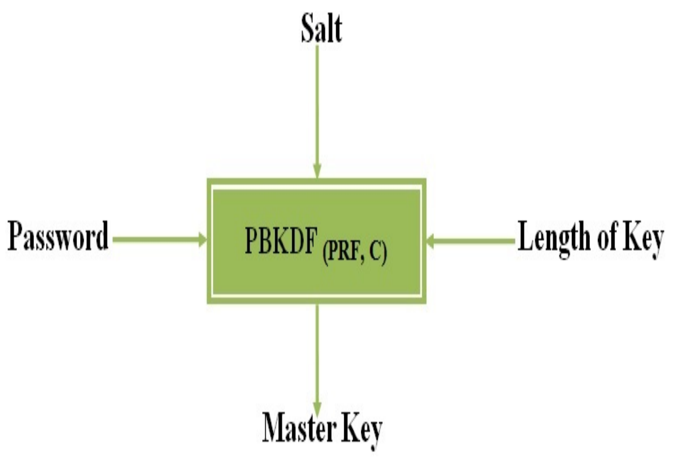


Fig.1. Generic diagram of the Password-Based Key Derivation Functions (PBKDF)

## 3 Terrestrial Coordinate System

The position of an observer on the earth's surface can be specified by the terrestrial coordinate system, which includes latitude and longitude. The "latitude" of a point on the Earth's surface is the angle between the equatorial plane and the straight line that passes through that point and through (or close to) the center of the Earth. Lines of "longitude" are imaginary lines which run in a north-south direction, from the North Pole to the South Pole. They are also measured in degrees (°).

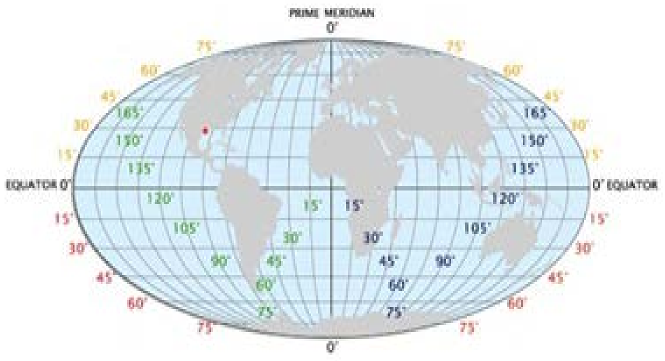


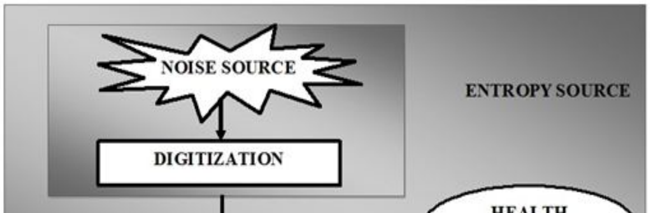
Fig.2. Longitude (shown as a horizontal line) is the angular distance, in degrees, minutes, and seconds of a point north or south of the Equator

The combination of these two components specifies the position of any location on the surface of the Earth, without consideration of altitude or depth. The facts of latitude and longitude are shown in (Fig. 2). Degrees of latitude and longitude can be further subdivided into minutes and seconds: there are 60 minutes (') per degree, and 60 seconds (") per minute. For example, a coordinate might be written 65° 32' 15". Degrees can also be expressed as decimals: 65.5375, degrees and decimal minutes: 65° 32.25', or even degrees, minutes, and decimal seconds: 65° 32' 15.275". All these notations allow us to locate places on the Earth quite precisely to within inches. A degree of latitude is approximately 69 miles, and a minute of latitude is approximately 1.15 miles. A second of latitude is approximately 0.02 miles, or just over 100 feet. A degree of longitude varies in size. At the equator, it is approximately 69 miles, the same size as a degree of latitude. The size gradually decreases to zero as the meridians converge at the poles. At latitude of 45 degrees, a degree of longitude is approximately 49 miles. Because a degree of longitude varies in size, minutes and seconds of longitude also vary, decreasing in size.

## 4 Entropy

Entropy is defined relative to one's knowledge of X prior to an observation, and reflects them certainty associated with predicting its value the larger the entropy, the greater the uncertainty in predicting the value of an observation. Entropy source may include a noise source (e.g., thermal noise or hard drive seek times), and it plays an important role for generating random salt. It is classified into two categories such as Full entropy and Min entropy when entropy is generated from noise sources. Let xi be a digitized sample from the noise source that is represented in one or more bits, let x1, x2, ..., xM be the outputs from the noise source, and let p(xi) be the probability that xi is produced at any given sampling time. The min-entropy of the outputs is: log2 (max p(xi)). This represents the best-case work for an adversary who is trying to guess an output from the noise source. Entropy Rate is the rate at which a digitized noise source (or entropy source) provides entropy; it is computed as the assessed amount of entropy provided by a bit string output from the source, divided by the total number of bits in the bit string (yielding assessed bits of entropy per output bit). This will be a value between zero (no entropy) and one (full entropy).

Entropy source is a source of random bit strings. The entropy source includes a noise source (e.g., thermal noise or hard drive seek times) [50], health tests, and an optional conditioning component. (Fig. 3) illustrates the model that uses to describe an entropy source, including the components that an entropy source developer shall implement. The noise source is the root of security for the entropy source and for the Random Bit Generator (RBG) as a whole [51]. Fundamentally, the noise source provides random bits in the form of digital samples obtained from a non-deterministic process. The sample values obtained from a noise source consist of fixed-length bit strings, which determine the output space of the component. The optional conditioning component is responsible for reducing bias and/or increasing the entropy rate of the resulting output bits. Health tests are an integral part of the entropy source design; the health test component ensures that the noise source and the entropy source as a whole continue to operate as expected.



## 5 ULTE Algorithm

### 5.1 Environment Record Information for Salt

In cloud computing a client needs to encrypt multiple files at a time. For encrypting multiple files, client side encryption tool needs to generate multiple different and unique encryption keys. Keys are generated based on client's password with a specialized algorithm called Password Based Key Derivation Function (PBKDF) [52], for ensuring uniqueness and randomness of encryption keys, a random value called Salt is used in PBKDF at each time of encryption key generation process. Generation of Salt by any pseudo random number generation algorithms make it easy for intruders like hackers or malicious users for cracking encryption keys. In this paper we propose an algorithm for generating Salt by using client's real time environmental information which is User Location Information, User Time Information time, and Entropy Data which is calculated from thermal noise or HDD seek time or any kind of noise sources at the key generation time.

**Location Information:** Location information is unpredictable for the mobile terminal users. Besides, the history of location information of a user's cannot be guessed or predicted through long time tracing of the user's movement, because it would be needed only at generation time. The recorded geography environmental information includes: latitude and longitude. Since mobile user change its location respect with time and location information could be a great source for making random salt. Now almost the modern mobile devices come with GPS antenna and service. Position of a mobile device can be calculated in many ways like Network-based, Handset-based, SIM-based, Wi-Fi network based, and Hybrid mode in which includes two or more methods in combined [53].

**Time Information:** At a given time, if the user is stationary or need to generate multiple keys at the same session with short period for processing multiple files, its location information remains the same, while the time information is changeable. Each moment we have got a single unique time and cannot be repeated. A time can be different from another time by a year, by a month, by a day, by an hour, by a minute, by a second, even by a millisecond or microsecond or nanosecond.

**Entropy Data:** Entropy data is measured from thermal noise or HDD seek time of the user's system. It is assumed that this data cannot be predicted since hackers could not have the system access directly always. It is a great source of random bit strings. The noise source is the root of security for the entropy source and for the Random Bit Generator (RBG). Basically, the noise source provides random bits in the form of digital samples obtained from a non-deterministic process. The sample values obtained from a noise source consist of fixed-length bit strings, which determine the output space of the Entropy data [54].

### 5.2 Cloud Salt Generation Process

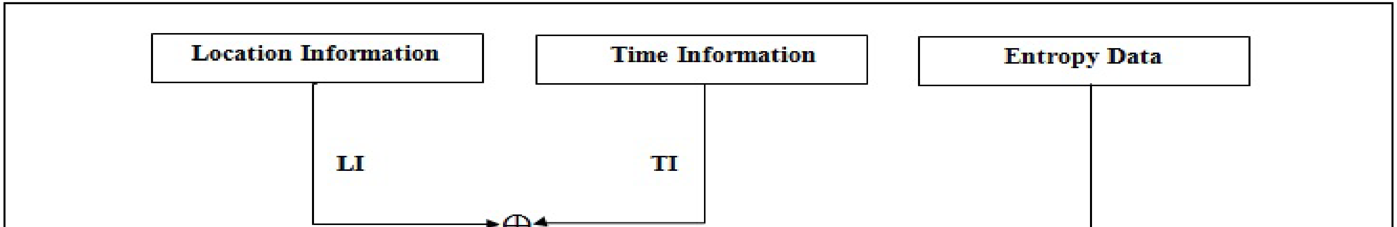
The attacker could acquire location and timing information of clients by tracking and following continuously. So, it is needed to add the interference related information in the Salt generation process to ensure randomness of the generated Salt. In this situation, we add thermal noise or HDD seek time as Entropy so that attacker could not guess the salt although in the meantime location and time information are compromised by attackers. Salt generation process is shown in the following for a single Salt. For multiple Salts then the whole process will be repeated for multiple times. For calculation purpose all the information is converted hexadecimal before processing. The whole process of Salt generation using ULTE Salt Generation Algorithm is drawn in the Figure 4.

**Users record its location information (LI) as history data:** Location information which includes latitude and longitude is recorded in Degrees-Minutes-Seconds format, portion after the decimal point discarded, for example: 372445, 1083435. This record indicates user locates at latitude 37 degrees 24 minutes 45 seconds, longitude 108 degrees 34 minutes 35 seconds. Latitude and Longitude are combined written, for above information combined information is 3724451083435. Corresponding hexadecimal of 3724451083435 is 3632A9574AB.

**Users record its time information (TI) as history data:** Time information is recorded in Year-Month-Day-Hour-Minute-Second format. For example: 20160424133456. This record indicates: at 56 seconds, 34 minutes, 13 hours, on April 24, 2016. 20150930094927 indicate at 27 seconds, 49 minutes, 9 hours, on September 30, 2015. Corresponding hexadecimal of 20150930094927 is 1253C107BB4F.

**Users record Entropy Data (ED) from system thermal noise or HDD seek time:** Entropy Data must be stored in 32 bit stream. For example 00101011000101101000011000010100 is recorded as thermal noise of user's system as Entropy Data. Corresponding hexadecimal of 00101011000101101000011000010100 is 2B168614..

Compute SHA256 ((LI ⊕ TI) ⊕ ED) to get 256 bits SALT.



**Algorithm 1:** Pseudo Code of ULTE Salt Generation Algorithm

```
Step-01: Start
Step-02: Record Location Information (LI) at the present in the Degrees-Minutes-Seconds format for both latitude and longitude.
Step-03: Record Time Information (TI) from the system in the Year-Month-Day-Hour-Minute-Second format.
Step-04: Compute $(LI \oplus TI)$.
Step-05: Read the Entropy Data (ED).
Step-06: Compute $((LI \oplus TI) \oplus ED)$.
Step-07: Compute SHA256 $((LI \oplus TI) \oplus ED)$ to get 256 bits SALT.
Step-08: Stop.
```

## 6 Simulation

Simulation of salt generation according to the proposed ULTE Salt Generation Algorithm is tabulated in this section. Simulation is performed using some random data of User's Location Information, Time Information, and Entropy Data. Result of each operation is tabulated in the corresponding table. Environmental Information like Location and Time Information is recorded in decimal form but they are converted into hexadecimal for calculation purposes, also Entropy Data is recorded in binary form and converted into hexadecimal.

Recorded decimal value and equal hexadecimal value of user's Location Information are tabulated in (Table I). Location values are recorded when user's need to generate Cryptographic keys as well as Encryption and Decryption keys. Recorded decimal value and equal hexadecimal value of user's Time Information are tabulated in Recorded decimal value and equal hexadecimal values of user's Location Information are tabulated in (Table II).

TABLE I

**User Location Information (Decimal, Hexadecimal).**

| *Case* | *Location Information in Decimal* | *Location Information in Hexadecimal* |
|---|---|---|
| **1** | 3724451083435 | 3632A9574AB |
| **2** | 783245651209 | B65D142D09 |
| **3** | 560324510983 | 8275F18507 |
| **4** | 1095623874312 | FF18459708 |
| **5** | 192376567293 | 2CCA8975FD |
| **6** | 9987541129876 | 91567D75A94 |
| **7** | 709812665285 | A54421BBC5 |
| **8** | 457635981256 | 6A8D3AE7C8 |
| **9** | 1102367563271 | 100AA3A1E07 |
| **10** | 872536908712 | CB2740A3A8 |

TABLE II

**User Time Information (Decimal, Hexadecimal).**

| *Case* | *Time Information in Decimal* | *Time Information in Hexadecimal* |
|---|---|---|
| **1** | 20160515124156 | 1255FC57ABBC |
| **2** | 20160516092534 | 1255FC667276 |
| **3** | 20160519101532 | 1255FC945C5C |
| **4** | 20160521085248 | 1255FCB2A140 |
| **5** | 20160522074536 | 1255FCC1B9A8 |
| **6** | 20160523074556 | 1255FCD0FBFC |
| **7** | 20160524074557 | 1255FCE03E3D |
| **8** | 20160526114735 | 1255FCFF5FAF |
| **9** | 20160527055241 | 1255FD0DB989 |
| **10** | 20160529094835 | 1255FD2CD8B3 |

TABLE III

**Entropy Data (Decimal, Hexadecimal).**

| *Case* | *Entropy Data in Binary* | *Entropy Data in Hexadecimal* |
|---|---|---|
| **1** | 10001010101001101010001001010101 | 8AA6A255 |
| **2** | 00101011000101101000011000010100 | 2B168614 |
| **3** | 10100110101001101010100001010101 | A6A6A855 |
| **4** | 11001000101001101001001001001101 | C8A6924D |
| **5** | 10101010100001000010010101010101 | AA842555 |
| **6** | 10110010100100101000001001010011 | B2928253 |
| **7** | 11001001010001001010100001010101 | C944A855 |
| **8** | 10100010101001100100100101010101 | A2A64955 |
| **9** | 10000100101010001010100100010101 01 | 84A2A455 |
| **10** | 10001010101001010100100100101010 | 8AA5492A |

Entropy data is measured from system thermal noise signal. Thermal noise signal is captured in analog signal and then converted in digital form. Recorded binary value and equal hexadecimal value of Entropy data which recorded from user's system thermal noise are tabulated in (Table III).

XOR operation is performed on Location Information (LI) and Time Information (TI) and results are tabulated in (Table IV). Location information and Time information are converted in hexadecimal for operation and outputs of XOR operation in hexadecimal.

TABLE IV

**Results of XOR operations of Location Information and Time Information.**

| *Case* | *LI (Location)* | *TI (Time)* | *(LIi⊕ TIi)* |
|---|---|---|---|
| **1** | 3632A9574AB | 1255FC57ABBC | 1136D6C2DF17 |
| **2** | B65D142D09 | 1255FC667276 | 12E3A1725F7F |
| **3** | 8275F18507 | 1255FC945C5C | 12D78965D95B |
| **4** | FF18459708 | 1255FCB2A140 | 12AAE4F73648 |
| **5** | 2CCA8975FD | 1255FCC1B9A8 | 12793648CC55 |
| **6** | 91567D75A94 | 1255FCD0FBFC | 1B409B07A168 |
| **7** | A54421BBC5 | 1255FCE03E3D | 12F0B8C185F8 |
| **8** | 6A8D3AE7C8 | 1255FCFF5FAF | 123F71C5B867 |
| **9** | 100AA3A1E07 | 1255FD0DB989 | 13555737A78E |
| **10** | CB2740A3A8 | 1255FD2CD8B3 | 129EDA6C7B1B |

**Results of XOR operations of Location Information and Time Information.**

TABLE V

(e) **Results of XOR operation of Entropy Data with the output of XOR operations of Location Information and Time Information.**

| *Case* | *(LIi⊕ TIi)* | *(ED)* | *((LIi⊕ TIi) ⊕ EDi)* |
|---|---|---|---|
| **1** | 1136D6C2DF17 | 8AA6A255 | 11365C647D42 |
| **2** | 12E3A1725F7F | 2B168614 | 12E38A64D96B |
| **3** | 12D78965D95B | A6A6A855 | 12D72FC3710E |
| **4** | 12AAE4F73648 | C8A6924D | 12AA2C51A405 |
| **5** | 12793648CC55 | AA842555 | 12799CCCE900 |
| **6** | 1B409B07A168 | B2928253 | 1B402995233B |
| **7** | 12F0B8C185F8 | C944A855 | 12F071852DAD |
| **8** | 123F71C5B867 | A2A64955 | 123FD363F132 |
| **9** | 13555737A78E | 84A2A455 | 1355D39503DB |
| **10** | 129EDA6C7B1B | 8AA5492A | 129E50C93231 |

XOR operation of Entropy Data with the output of XOR operations of Location Information and Time Information are performed and tabulated in Table V.

In Table VI the results are tabulated after calculating Hash of ((LIi⊕ TIi) ⊕ EDi) using Secure Hash Algorithm –SHA256. Output of SHA256 is 256 bits and they converted in hexadecimal.

TABLE VI

**Results of SHA256 ((LIi⊕ TIi)⊕ EDi) operation.**

| *Case* | *((LIi⊕ TIi)⊕ EDi)* | *SHA256 ((LIi⊕ TIi)⊕ EDi)* |
|---|---|---|
| **1** | 11365C647D42 | 2BE6321D0CF8A235A7FC722AAE41A742B8975862E04A9A1FCD4706461AE4D8CB |
| **2** | 12E38A64D96B | 573AE6543502971E0474E817825293E5864CF82C091AA78EA64FE9C345FD63C0 |
| **3** | 12D72FC3710E | 227D27C30151C78041E0C0C91529247B2D503FBA8B0CCCE016A2144260FEB75C |
| **4** | 12AA2C51A405 | 336A796023B9270AAF2C1838F045009C61135BA19F29F0E6B7D4974B04A6B350 |
| **5** | 12799CCCE900 | 1A247B437CC9513A97C62643CC429E0BE1FB12BA01441CF6016FED9510193CE2 |
| **6** | 1B402995233B | D9AB75606191B597BCD8EA066ADAA41A86DBE9131B6ACF99A0CE8FA16AA371AB |
| **7** | 12F071852DAD | 9E13A02925E338680E79EEC1E6C071EEC23FCFE7764F636B26A75F21E300563D |
| **8** | 123FD363F132 | E63EA1503887831EEA8610CB8798C0B8CAA7D8C566CF2A62474759CA8FE2D507 |
| **9** | 1355D39503DB | CE83F1E8ACB54F44F5F9F5FE610BD92C4B167DE241D35F7DC324372AFC8534D3 |
| **10** | 129E50C93231 | C3F1A9BE2E1EBDD2554C7ECDBDC21F4E3F2B56AB3B776FCF094A736DCD6A8779 |

Results of Hash function SHA256 of ((LIi⊕ TIi)⊕ EDi) is converted into binary value and tabulated in the Table VII.

TABLE VII

**Results of SHA256 ((LIi⊕ TIi)⊕ EDi) operation.**

| *Case* | *SHA256 ((LIi⊕ TIi)⊕ EDi)* | *BINARY VALUE* |
|---|---|---|
| **1** | 2BE6321D0CF8A235A7FC722AAE41A742B8975862E04A9A1FCD4706461AE4D8CB | 0010101111100110001100100001110100001100111110001010001000110101101001111111110001110010001010101010111001000001101001110100001010111000100101110101100001100010111000000100101010011010000111111100110101000111000001100100011000011010111001001101100011001011 |
| **2** | 573AE6543502971E0474E817825293E5864CF82C091AA78EA64FE9C345FD63C0 | 0101011100111010111001100101010000110101000000101001011100011110000001000111010011101000000101111000001001010010100100111110010110000110010011001111100000101100000010010001101010100111100011101010011001001111111010011100001101000101111111010110001111000000 |
| **3** | 227D27C30151C78041E0C0C91529247B2D503FBA8B0CCCE016A2144260FEB75C | 0010001001111101001001111100001100000001010100011100011110000000010000011110000011000000110010010001010100101001001001000111101100101101010100000011111110111010100010110000110011001100111000000001011010100010000101000100001001100000111111101011011101011100 |
| **4** | 336A796023B9270AAF2C1838F045009C61135BA19F29F0E6B7 | 00110011011010100111100101100000001000111011100100100111000010101010111100101100000110000011100011110000010001010000000010 |

**Results of SHA256 ((LIi⊕ TIi)⊕ EDi) operation.**

| *Case* | *SHA256 ((LIi⊕ TIi)⊕ EDi)* | *BINARY VALUE* |
|---|---|---|
| | D4974B04A6B350 | 0111000110000100010011010110111010000110011111001010011111000<br>0111001101011011111010100100101110100101100000100101001101011<br>001101010000 |
| **5** | 1A247B437CC9513A97C62643<br>CC429E0BE1FB12BA01441CF6<br>016FED9510193CE2 | 0001101000100100011110110100001101111100110010010101000100111<br>0101001011111000110001001100100001111001100010000101001111000<br>0010111110000111111011000100101011101000000001010001000001110<br>0111101100000000101101111111011011001010100010000000110010011<br>110011100010 |
| **6** | D9AB75606191B597BCD8EA06<br>6ADAA41A86DBE9131B6ACF<br>99A0CE8FA16AA371AB | 1101100110101011011101010110000001100001100100011011010110010<br>1111011110011011000111010100000011001101010110110101010010000<br>0110101000011011011011111010010001001100011011011010101100111<br>1100110011010000011001110100011111010000101101010101000110111<br>000110101011 |
| **7** | 9E13A02925E338680E79EEC1E<br>6C071EEC23FCFE7764F636B26<br>A75F21E300563D | 1001111000010011101000000010100100100101111000110011100001101<br>0000000111001111001111011101100000111100110110000000111000111<br>1011101100001000111111110011111110011101110110010011110110001<br>1011010110010011010100111010111110010000111100011000000000101<br>011000111101 |
| **8** | E63EA1503887831EEA8610CB8<br>798C0B8CAA7D8C566CF2A62<br>474759CA8FE2D507 | 1110011000111110101000010101000000111000100001111000001100011<br>1101110101010000110000100001100101110000111100110001100000010<br>1110001100101010100111110110001100010101100110110011110010101<br>0011000100100011101000111010110011100101010001111111000101101<br>010100000111 |
| **9** | CE83F1E8ACB54F44F5F9F5FE6<br>10BD92C4B167DE241D35F7DC<br>324372AFC8534D3 | 1100111010000011111100011110100010101100101101010100111101000<br>1001111010111111001111101011111111001100001000010111101100100<br>1011000100101100010110011111011110001001000001110100110101111<br>1011111011100001100100100001101110010101011111100100001010011<br>010011010011 |
| **10** | C3F1A9BE2E1EBDD2554C7EC<br>DBDC21F4E3F2B56AB3B776FC<br>F094A736DCD6A8779 | 1100001111110001101010011011111000101110000111101011110111010<br>0100101010101001100011111101100110110111101110000100001111101<br>0011100011111100101011010101101010101100111011011101110110111<br>1110011110000100101001010011100110110110111001101011010101000<br>011101111001 |

## 7 Conclusion and Future Work

In this paper we use user's environmental information such as Location Information, Time Information, and Entropy Data for generating Salt which is a random number for Password-Based Key Derivation Function (PBKDF) Algorithm which widely used in Cloud Computing for generating Cryptographic keys for both Encryption and Decryption operations. We use that environmental information for Salt generation so that hackers or intruders or malicious users could not guess the sequences of Salt. Random number generation from any mathematical equation will help hackers for guessing random number sequences easily due to the radip growth of computer and information technology. It is not possible to guess Location, Time and Entropy information without tracking and having user's system physical access.

We design ULTE Salt Generation Algorithm for mobile devices users where users' location changes with time to time. It won't work when user's location does not change and users have got to be placed at a fixed location for all time. We will work for designing new algorithm for Salt generation for users when user's location is static with respect to time. We will try to find and include some others real time environmental information for users so that from those environmental information system could generate Salt or Random number for Encryption purposes and hackers would not be able to guess or crack the random number sequences.

In future we will perform Parameter test of this ULTE algorithm as random number sequence generator compared with other random number generator like Prime Modulus Multiplicative Linear Congruential Generator

(PMMLCG), Generalized Feedback Shift Register Generator (GSFRG), Super-prime method and etc. Parameter test of the random number sequence generator mainly includes mean test, variance test and second moment test.

## REFERENCES

[1] T. Dillon, C. Wu and E. Chang. "Cloud Computing: Issues and Challenges," 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, WA, Apr 2010, pp 27 – 33.

[2] Anitha Y. (2013, Dec.). "Security Issues in Cloud Computing- A Review." International Journal of Thesis Projects and Dissertations (IJTPD). [On-line]. 13, pp. 1-6. Available:http://www.researchpublish.com/journal/IJTPD/Issue-1-October-2013-December-2013/0 [Jan. 12, 2016].

[3] B. Pring, R. H. Brown, A. Frank, S. Hayward, L. Leong. (2009, Mar.). "Forecast: Sizing the cloud; understanding the opportunities in cloud services", Gartner Inc., Tech. Rep.G00166525. [On-line]. 27(3). Available: https://www.gartner.com/doc/914826/forecast-sizing-cloud-understanding-opportunities [Feb. 17, 2016].

[4] W. Zeng, Y. Zhao, K. Ou, and W. Song. "Research on cloud storage architecture and key technologies," Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, Seoul, Korea, Nov 2009, pp 1044-1048.

[5] J. Weinman "The Future of Cloud Computing." IEEE Technology Time Machine Symposium on Technologies Beyond 2020 (TTM), Hong Kong, Jun. 2011, pp 1-2.

[6] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, A. Vakali "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", IEEE Internet Computing Journal, Vol. 13, pp 10-13. Sep. 2009.

[7] M. Malekimajd, D. Ardagna, M. Civotta, A. M. Rizzi, and M. Passacantando "Optimal Map Reduce Job Capacity Allocation in Cloud Systems", ACM SIGMETRICS Performance Evaluation Review, Vol. 42, pp 51-61. Mar. 2015.

[8] T. Harmer, P. Wright, C. Cunningham, and R. Perrott. "Provider Independent Use of the Cloud," 15th International European Conference on Parallel and Distributed Computing, Delft, The Netherlands, Aug 2009, pp 454-465.

[9] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. "The eucalyptus open-source cloud computing system." Proceedings of 12th International Conference on Cloud Computing and Its Applications", Washington, DC, USA, Feb. 2008, pp 1-8.

[10] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Petterson, A. Rabkin, I. Stoica, M. Zaharica. (2010, Apr.). "A View of Cloud Computing. Communications of the ACM." Communications of the ACM - ACM Digital Library. [On-line]. 53, pp. 50-58. Available: 10.1145/1721654.1721672 [Feb. 23, 2016].

[11] P. Mell and T. Grance. (2011, Sep.). "The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology." Computer Security Division, IT Laboratory, National Institute of Standards and Technology Special Publication 800-145, Gaithersburg. [On-line]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf [Jan. 06, 2016].

[12] N. Sinha, and L. Khreisat. "Cloud computing security, data, and performance issues," 23rd Wireless and Optical Communication Conference (WOCC), Newark, NJ, May 2014, pp 1 - 6.

[13] A. Lenk, M. Klems, J. Nimis, S. Tai and T. Sandholm. "What's Inside the Cloud? An Architectural Map of the Cloud Landscape," Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, Washington DC, 2009, pp 23-31.

[14] J. Shamsi, M. Khojaye and M. Qasmi. (2013, June.). "Data-Intensive Cloud Computing: Requirements, Expectations, Challenges, and Solutions." Journal of Grid Computing. [On-line]. 11, pp. 281-310, Available: http://link.springer.com/article/10.1007/s10723-013-9255-6. [Mar. 13, 2016].

[15] R. Maggiani. "Cloud Computing is Changing How we Communicate," in IEEE International Professional Communication Conference, Waikiki, HI, USA, Jul 2009, pp 1-4.

**[16]** F. Sabahi. "On Technical Security Issues in Cloud Computing," in IEEE 2nd International Conference on Cloud Computing, Bangalore, India, Sep 2009, pp 109 – 116.

[17] F. Sabahi. "Cloud Computing Security Threats and Responses," in IEEE 3rd International Conference on Communication Software and Networks (ICCSN), Xi'an, May 2011, pp 245 – 249.

[18] A. S.Ibrahim, J. Hamlyn-Harris and J. Grundy. "Emerging Security Challenges of Cloud Virtual Infrastructure," in APSEC 2010 Cloud Workshop, Sydney, Australia, 2012, pp 1054-1057.

[19] F. Sabahi. "Virtualization-Level Security in Cloud Computing," in IEEE 3rd International Conference on Communication Software and Networks (ICCSN), Xi'an, May 2011, pp 250 – 254.

[20] E. Mathisen. "Security Challenges and Solutions in Cloud Computing," in Proceedings of 5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011), Daejeon, May 2011, pp 208-212.

[21] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacon. "On technical Security Issues in Cloud Computing," in Proceedings of IEEE International Conference on Cloud Computing (CLOUD-II, 2009), Bangalore, India, Sep 2009, pp 109-116.

[22] S. Kamara, and K. Lauter. "Cryptographic cloud storage," Lecture Notes in Computer Science, Financial Cryptography and Data Security, 6054, 2010, pp. 136- 149.

[23] L. M. Kaufman (2009, Aug.). "Data security in the world of cloud computing." IEEE Security and Privacy Journal. [On-line]. 7(4), pp. 61-64. Available: DOI: 10.1109/MSP.2009.87 [Apr. 11, 2016].

[24] K. Hashizume, D. G. Rosado, E. F. Medina, and E. B. Fernandez (2013, Feb.). "An analysis of security issues for cloud computing." Journal of Internet Services and Applications. [On-line]. 4, pp. 1-13. Available: **DOI:** 10.1186/1869-0238-4-5 [Apr. 19, 2016].

[25] R. P. Padhy, M. R. Patra, S. C. Satapathy (2011, Dec.). "Cloud Computing: Security Issues and Research Challenges." International Journal of Computer Science and Information Technology & Security (IJCSITS). [On-line]. 1(2), pp. 136-146. Available: DOI: http://ijcsits.org/papers/Vol1no22011/13vol1no2.pdf [May. 02, 2016].

[26] K. Chandrahasan, R. Kalaichelvi, S. S. Priya, and L. Arockiam (2012, Mar.). "Research Challenges and Security Issues in Cloud Computing." International Journal of Computational Intelligence and Information Security. [On-line]. 3, pp. 42-48. Available: DOI: 10.4236/cn.2014.61003 [Jun. 22, 2016].

[27] P. A. L. Rego, E. F. Coutinho , A. S. Lima , and J. N. D. Souza (2015, Aug.). "Using Processing Features for Allocation of Virtual Machines in Cloud Computing." IEEE Latin America Transactions. [On-line]. 13(8), pp. 2798 - 2812. Available: DOI: 10.1109/TLA.2015.7332165 [May. 16, 2016].

[28] S. Subashini and V. Kavitha (2011, Jan.). "A Survey on Security Issues in Service Delivery Models of Cloud Computing." Journal of Network and Computer Applications. [On-line]. 34 (1), pp. 1-11. Available: doi:10.1016/j.jnca.2010.07.006 [Apr. 19, 2016].

[29] M. T. Khorshed, A. B. M. Shawkat Ali, and S. A. Wasimi. "Trust Issues that create threats for Cyber-attacks in Cloud Computing," in IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS), Tainan, Dec 2011, pp 900-905.

[30] B. Grobauer, T. Walloschek and E. Stocker (2011, Jun.). "Understanding Cloud Computing Vulnerabilities." IEEE Security Privacy. [On-line]. 9 (2), pp. 50-57. Available: doi: 10.1109/MSP.2010.115 [Apr. 11, 2016].

[31] Xiaopeng G, Sumei W, Xianqin C. "VNSS: a Network Security sandbox for virtual Computing environment," in IEEE youth conference on information Computing and telecommunications (YC-ICT). Washington DC, USA, Nov 2010, pp 395–398.

[32] K. Popovic and Z. Hocenski. "Cloud Computing Security Issues and Challenges," in Proceedings of the 33rd International Convention in MIPRO, Opatija, Croatia, 2010, pp 344-349.

[33] D. Jamil and H. Zaki (2011, Jun.). "Cloud Computing Security." International Journal of Engineering Science and Technology. [On-line]. 3 (4), pp. 3478-3483. Available: doi: 10.1109/CLOUD.2011.9 [Jun. 22, 2016].

[34] S. K. Das, M. A. Hossain., M. A. Sardar, R. K. Biswas, P. D. Nath (2014) "Performance Analysis of Client Side Encryption Tools." International Journal of Advanced Computer Research. [On-line]. 4 (3), pp. 888-897. Available: http://accentsjournals.org/PaperDirectory/Journal/IJACR/2014/9/20.pdf [Jul. 13, 2016].

[35] A. Aleem and C. R. Sprott (2013, Feb.). "Let Me in the Cloud: Analysis of the Benet and Risk Assessment of Cloud Platform." Journal of Financial Crime. [On-line]. 20 (1), pp. 6-24. Available: doi: http://dx.doi.org/10.1108/13590791311287337 [Jul. 24, 2016].

[36] Y. Ye, L. Xiao, I. L. Yen, and F. Bastani. "Secure, Dependable, and High Performance Cloud Storage," in 29th IEEE International Symposium on Reliable, New Delhi, India, 2010, pp 194 – 203.

[37] S. Ramgovind, M. Elo and E. Smith. "The Management of Security in Cloud Computing," in Proceedings of Information Security for South Africa, Sandton, 2010, pp 1-7.

[38] C. Soghoian (2010, Spr.). "Caught in the Cloud: Privacy, Encryption, and Government Back Doors in the Web 2.0 era." Journal on Telecommunications and High Technology Law. [On-line]. 8 (2), pp. 359-424. Available: http://www.jthtl.org/articles.php?volume=8 [Jul. 24, 2016].

[39] D. Chen and H. Zhao. "Data Security and Privacy Protection Issues in Cloud Computing," in International Conference on Computer Science and Electronics Engineering, Hangzhou, Mar 2012, pp 647-651.

[40] B. Sotomayor, R. Montero, I. Llorente, and I. Foster (2009, Oct.). "Virtual Infrastructure Management in Private and Hybrid Clouds." IEEE Internet Computing. [On-line]. 13 (5), pp. 14-22. Available: doi: 10.1109/MIC.2009.119 [May. 16, 2016].

[41] D. Zissis and D. Lekkas (2012, Mar.). "Addressing Cloud Computing Security Issues." Future Generation Computer Systems. [On-line]. 28 (3), pp. 583-592. Available: doi:10.1016/j.future.2010.12.006 [Apr. 19, 2016].

[42] C. Wang, Q. Wang, K. Ren and W. Lou. "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in Proceedings IEEE in INFOCOM, San Diego, 2010, pp 1-9.

[43] S. Pearson. "Taking account of privacy when designing cloud computing services," in CLOUD '09 Proceedings of ICSE Workshop on Software Engineering Challenges of Cloud Computing, IEEE Computer Society, Washington, DC, USA, 2009, pp 44-52.

[44] H. C. Lin, S. Babu, J. S. Chase, and S. S. Parekh. "Automated Control in Cloud Computing: Opportunities and Challenges," in Proceedings of the 1st Workshop on Automated control for data centres and clouds, New York, NY, USA, 2009, pp 13-18.

[45] S. Chen; S. Nepal, and R. Liu. "Secure Connectivity for Intra-cloud and Inter-cloud Communication," in 40th International Conference on Parallel Processing Workshops (ICPPW), Sep 2011, pp 154-159.

[46] M. S. Turan, E. Barker, W. Burr, and L. Chen (2010, Dec.). "Recommendation for Password-Based Key Derivation, Part 1: Storage Applications." Computer Security Division, Information Technology Laboratory, NIST, SP 800-132. [On-line]. Available: http://dx.doi.org/10.6028/NIST.SP.800-132 [Jun. 24, 2016].

[47] E. Barker, J. Kelsey (2015, Jun.). "Recommendation for Random Number Generation Using Deterministic Random Bit Generators-Revision 1." Computer Security Division, Information Technology Laboratory, NIST SP 800-90A. [On-line]. Available: http://dx.doi.org/10.6028/NIST.SP.800-90Ar1 [Jul. 24, 2016].

[48] L. Chen (2009, Oct.). "Recommendation for Key Derivation Using Pseudorandom Functions." Computer Security Division, Information Technology Laboratory, NIST SP 800-108. [On-line]. Available: http://csrc.nist.gov/publications/nistpubs/800-108/sp800-108.pdf [Jun. 21, 2016].

[49] "Latitude Information of a Location." Internet: https://www.coursehero.com/file/p506990/Results-1-What-information-does-the-latitude-of-a-location-tell-you-In/. Jul, 2013 [Jul. 07, 2016].

[50] M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish, and M. Boyle (2012, Aug.). "Recommendation for the Entropy Sources Used for Random Bit Generation." Computer Security Division, Information Technology Laboratory, NIST DRAFT Special Publication 800-90B. [On-line]. Available: http://dx.doi.org/10.6028/NIST.SP.XXX [Jul. 14, 2016].

[51] Z. Li, L. O'Brien, and H. Zhang "BOOSTING METRICS: MEASURING CLOUD SERVICES FROM THE HOLISTIC PERSPECTIVE", International Journal of Cloud Computing (IJCC), 2(4), 2014, pp. 12-22.

[52] S. Nepal, C. Friedrich, C. Wise, R. O. Sinnott, J. J. Jaccard, and S. Chen (2016, Jun.). "KEY MANAGEMENT SERVICE: ENABLING SECURE SHARING AND DELETING OF DOCUMENTS ON PUBLIC CLOUDS." Services Transactions on Cloud Computing (STCC), 4(2), 2016, pp. 15-31.

[53] "Mobile Phone Tracking." Internet: https://en.wikipedia.org/wiki/Mobile_phone_tracking. Jul. 1, 2016 [Jul. 24, 2016].

[54] N. Hossain, M. A. Hossain, A. K. M. F. Islam P. Banarjee, and T. Yasmin (2016, Aug.). "Research on Energy Efficiency in Cloud Computing." International Journal of Scientific & Engineering Research, Volume 7, Issue 8, pp. 358-367.