Database Encryption Using Fuzzy Chaotic

Saad M. Darwish, Adel A. El-Zoghabi, and Mohammed A. Abdewi

***Abstract*—Database encryption is a fundamental technique in the security mechanisms of database that is characterized by both the fast speed of the conventional encryption and the convenience of key distribution of public key encryption. There are two problems with traditional database encryption schemes. They show tradeoff between efficiency and security. Furthermore, these schemes can't solve the problem of storing multi-level encrypted elements into database besides having no ability for effective key management. In this paper, a new paradigm for database encryption is proposed in which database encryption can be provided as a service to applications with seamless access to encrypted database. The proposed system utilizes a chaotic encryption method based on cellular automata to realize higher complexity of crypt-analytical attacks. Cellular Automata rules are defined based on chaos mapping to generate a symmetric key. Furthermore, a fuzzy observer based scheme for synchronizing chaotic keys of encrypted signal is employed to enhance key distribution. The suggested system have some advantages such as confusion, diffusion, very large number of passwords helped in building of symmetric private key, key-dependent mapping and increasing system complexity with the impact of indefinite rules and chaos mapping. Simulation results obtained from some database demonstrate the strong performance of the proposed encryption system.**

***Index Terms*—Database encryption, fuzzy chaotic, cellular automata, security.**

I. Introduction

Today's marketing database contains data that is vital for all aspects of the business such as marketing, sales, finance, customer service, and product development. However, a new emerging option in this era is illustrated by Database as a Service (DbaaS) structure [1]. Based on this paradigm, data owner manage the DBMS and answer to user's query directly, rather than traditional client-server architecture. One of the most serious obstacles to the prevalent use of DbaaS is related to security issues [2]. Attackers for data stored in database are not only from external parties but also from internal and their vulnerability to external attack increases. Therefore, to properly maintain the integrity and confidentiality of the data, database security becomes one of the most urgent challenges in database research.

Generally, database security methods could be divided into four layers: physical security, operation system security, DBMS security and database encryption [3]. These layers protect database in different aspects. But only first three layers are inadequate to protect the confidential data in database satisfactorily, because data is still stored in readable form. So without database encryption, it makes no sense to guarantee that the sensitive information in plaintext will be protected against a malicious user who has super-user power, such as a database administrator (DBA).

Database security includes topic such as statistical database security, intrusion detection (ID) and most recently privacy preserving data mining [4]. Firewall and ID only provide network layer protection. Access control is based upon the concept of privilege and it is a basic for many security features. One of the requirements for database security is database encryption. With database encryption, the valuable information in database becomes more secure since the encrypted data ensure the confidentiality of the data. Furthermore, database encryption can be employed to maintain the data integrity, ensuring that even a little modification made on the data can be detected [1]. Thus, this paper will focus specifically on some of the details on cryptographic algorithm used to implement the database encryption.

There are two main approaches for database encryption which is whether performing encryption and decryption inside or outside database [3]. After reviewing the recent database encryption algorithms, the best ways to secure the information stored in database is to apply it at outside the database i.e. application level encryption. By using this approach, encryption will be on the column and row basis. Hence, not all data stored in the database will be encrypted. However, encryption done by application itself also poses some challenges, if data is encrypted at the application, then all applications that access the encrypted data must be changed to support the encryption/decryption model. The scholars get an idea that "encryption as a service" to application for solving the problem. They use a special Encryption/Decryption engine outside the database as the service provider [4].

There are two basic dimensions of encryption support in databases to consider [5]. One is the granularity of data to be encrypted or decrypted. The field, the record and the table are the alternatives. A compromise solution between performance and security can be achieved by only encrypting the sensitive field. The second dimension is the choice of encryption algorithm. There are two types of it: the symmetric and the public key encryption. The symmetric algorithm is faster whereas safety of private key storage is also hard to guarantee in public key scheme. Generally, public key encryption schemes are not used frequently for database encryption [1].

A good encrypted database system should meet the

Manuscript received August 11, 2014; revised October 25, 2014.

Saad M. Darwish and Adel A. El-Zoghabi is with the Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, 163 Horreya Avenue, El-Shatby 21526, P.O. Box 832, Alexandria, Egypt (e-mail: saad.saad@alexu.edu.eg, zoghabi@gmail.com).

Mohammed A. Abdewi is with the Department of Computer and Information, Ministry of Education, Iraq (e-mail: mohammed.eyfan@yahoo.com).

DOI: 10.7763/IJFCC.2014.V3.343

following requirements, which are also our scheme aims to meet [1], [3]: 1) sensitive data at rest are encrypted, thus if the database becomes available to the adversary, its contents are not semantic, therefore not directly usable by the adversary ; 2) the process of encryption or decryption is transparent to application. That is to say, user (application) thinks that he interacts with the original non-encrypted database, posing his original query against database; 3) the storage after encrypting doesn't increase much. Generally encrypted data is much longer than the original data in-the-clear. Length of encrypted data depends on the properties of the encryption algorithm itself, which are out of concern in this paper; 4) safe key management. Key management includes key generation, key distribution, key destruction and key sharing. Independently of the encryption strategy, the security of the encrypted data depends on the encryption algorithm, the encryption key size and its protection.

The attractiveness of encryption technology comes out in more pronounced way when there is no absolute relation between cipher and original data and it is possible to rebuild the original message in much easier way [1]. As chaotic systems are known to be more random and non-predictable, they can be made utilized in achieving the encryption. In recent years, the chaos based cryptographic algorithms have suggested some new and efficient ways to develop secure encryption techniques. The idea of chaotic masking is to directly add the message in a noise-like chaotic signal at the transmitter, while chaotic modulation is done by injecting the message into a chaotic system as in spread spectrum transmission. The main advantages of the chaotic encryption approach include: high flexibility in the encryption system design, good privacy due to both nonstandard approach and vast number of variants of chaotic systems, large, complex and numerous possible encryption keys and simpler design. Furthermore, the transposition technology of encryption systems requires scrambleness behavior in order to achieve the encryption of the data. This scrambleness behavior can be derived from the randomness property of chaos which can be better utilized in the techniques like transposition system [6].

Fuzzy chaotic systems provide original heuristic research achievements and insightful ideas on the interactions or intrinsic relationships between fuzzy logic and chaos theory [7]. Chaotic behavior in nonlinear dynamical systems is very difficult to detect and control. Part of the problem is that mathematical results for chaos are difficult to use in many cases, and even if one could use them there is an underlying uncertainty in the accuracy of the numerical simulations of the dynamical systems. For this reason, current researches are moving toward modeling the uncertainty of detecting the range of values where chaos arises using fuzzy sets theory [8]. However, it has been shown that most of these fuzzy chaotic methods have a low level of security because of single keying concept.

Cellular Automata (CA) is a discrete model that consists of grids of cells in which each cell can exist in finite number of states. Every cell can change its state based on the states of neighboring cells by following a prescribed rule. CA with its inherent properties like parallelism, homogeneity, and unpredictability, as well as it being easily implementable in both software and hardware systems, has become an important tool to develop cryptographic methods. Cryptography of symmetric key systems based on CA, were studied by scholars. The encryption process is based on the generation of pseudorandom bit sequences, and CAs can be effectively used for this purpose. The quality of pseudorandom numbers highly depends on the set of applied CA rules [9].

This paper presents a new effort to involve fuzzy chaos theory and cellular automata technique for database encryption. CA is used to generate a symmetric key based on a new set of rules that makes the system very resistant to attempts of breaking the cryptography key. The cryptosystem uses Takagi-Sugeno fuzzy (*T*–*S* fuzzy) models to exactly represent discrete-time chaotic systems into separate linear systems in spite of a small number of implications of rules for synchronizing chaotic keys of encrypted signal is employed to enhance key distribution. The scalar transmitted signal is designed in such a way that the chaotic carrier masks the encrypted password, which in turn hides the message signal. The proposed encryption algorithm is characterized by both the fast speed of the encryption process and the convenience of key distribution, which enables the encrypted data to be shared conveniently. It reduces the cost of managing users and facilitates privacy managements.

The rest of the paper is organized as follows: Section II discusses related work on database encryption. In Section III we present our database encryption scheme, showing our architecture in detail. In Section IV, simulation results for many cases are presented to evaluate the performance of our system. Finally in Section V, we make the conclusion of our research.

II. Background

In terms of encryption of relational database management system, many creative and efficient schemes have been proposed. These schemes can be classified into: storage-level, database-level and application-level encryption [1], [10]. From a database perspective, storage-level encryption has the advantage to be transparent, thus avoiding any changes to existing applications. On the other side, since the storage subsystem has no knowledge of database objects and structure, the encryption strategy cannot be related with user privileges (e.g., using distinct encryption keys for distinct users), nor to data sensitivity. Thus, selective encryption – i.e., encrypting only portions of the database in order to decrease the encryption overhead – is limited to the file granularity.

Inside database-level encryption, the encryption strategy can thus be part of the database design and can be related with data sensitivity and/or user privileges [4]. Selective encryption is possible and can be done at various granularities, such as tables, columns, rows. Depending on the level of integration of the encryption feature and the DBMS, the encryption process may incur some change to applications. Moreover, it may cause DBMS performance degradation since encryption generally forbids the use of index on encrypted data. For both strategies, data is decrypted on the database server at runtime. Thus, the encryption keys must be transmitted or kept with the encrypted data on the server side; thereby providing a limited

protection against the server administrator or any intruder usurping the administrator identity. Indeed, attackers could spy the memory and discover encryption keys or plain text data [5].

Application-level encryption moves the encryption and decryption process to the applications that generate the data. This approach has the benefit to separate encryption keys from the encrypted data stored in the database since the keys never have to leave the application side. However, applications need to be modified to adopt this solution. Finally, such a strategy induces performance overheads (index on encrypted data are useless) and forbids the use of some advanced database functionalities on the encrypted data, like stored procedures and triggers. In terms of granularity and key management, application-level encryption offers the highest flexibility since the encryption granularity and the encryption keys can be chosen depending on application logic [1], [10].

Some encryption systems are based on record-oriented cryptosystem that enables encryption at the level of rows and decryption at the level of cells [1]. Other systems extend the sub-key encryption by supporting multilayer access control to improve the encryption algorithms and make it more efficient, but they didn't give an effective way to manage keys and ensure that the keys are safekeeping. Some academics researched how to integrate modem cryptography technology into a relational database management system from the view of cryptographic [11]-[14]. They introduced a security dictionary to solve the problems linked to key management, which is very effective to protect security related data. But in their schemes the key pairs are stored at a directory server, so every time the database server needs a key to encrypt or decrypt data, the key must be retrieved from the directory server first, which decreases the performance of operation.

The work presented in [10] focused on the design of database encryption at application level using enhanced affine block cipher. This improvement has been made because of the weakness found in the original affine cipher. The enhanced affine block cipher was developed and implemented where the selected sensitive data is encrypted outside the database (application level) and then it is inserted into database. The new encoding schema and modification Cipher Block Chaining (CBC) mode of operation for block cipher was designed for the new algorithm and then the prototype of the system was built and implemented into existing system for protecting user password.

The authors in [15] presented a new scheme for database encryption at multi-granularity level besides protecting index information as well. This system provides multiple encryption algorithms and many encryption granularities to overcome the drawback of traditional database encryption schemes that exhibit tradeoff between security and efficiency. It makes use of a special index server and a session key to protect index information. The authors created B+ tree index for the data to be inserted just before encrypting it besides storing the encrypted data in to database. They also take care of managing keys effectively besides having access control mechanisms over each and every element.

A new database indexing scheme that does not reveal any information on the database plaintext values was proposed by E. Shmueli *et al.*, [16]. In this scheme index values are encrypted with a unique number (the row-*id* of the database value) in order to eliminate patterns matching attacks and any correlation between index and database values. Ensuring index integrity is possible if an index position can be attached to each index value by simply using a technique similar to the one used for table encryption. Their schemes do not impose any changes on the database structure, thus enabling a DBA to manage the encrypted database as any other non-encrypted database. Furthermore, implementing the new scheme in existing applications does not entail modifying the queries.

One disadvantage of all the above schemes is that the basic element in the database is a row and not a cell, thus the structure of the database is modified. In addition, all of those schemes require re-encrypting the entire row when a cell value is modified. Thus, in order to perform an update operation, all the encryption keys should be available. A number of schemes which encrypt each cell in the database individually together with its cell coordinates (table name, column name and row-*id*) are suggested. In this way static leakage attacks are prevented since equal plaintext values are encrypted to different cipher-text values. Furthermore, splicing attacks are prevented since each cipher-text value is correlated with a specific location, trying to move it to a different location will be easily detected. Further security analysis and fixes to these schemes can be found in [2], [10].

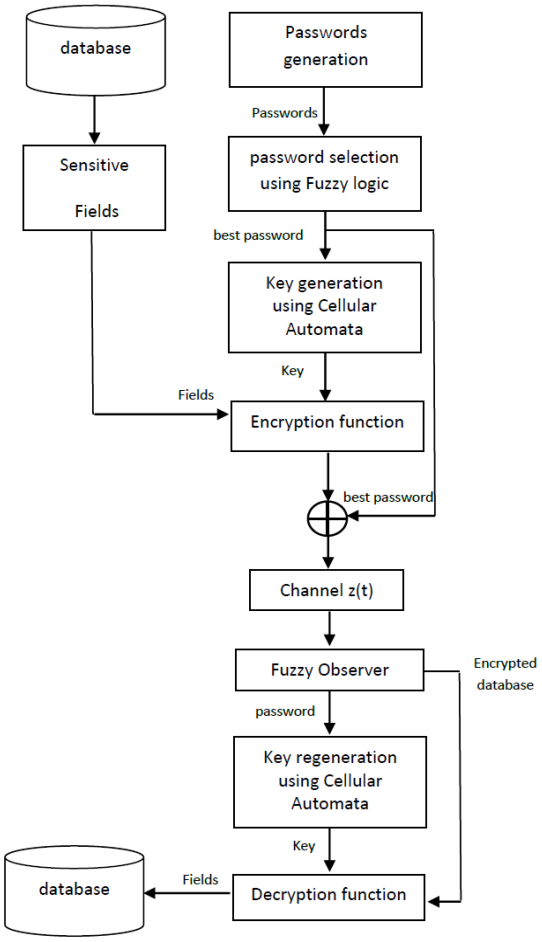


Fig. 1. Main system workflow.

While all existing commercial database products adopt classical encryption algorithms for database encryption, specific encryption schemes have attracted much attention in the academic field, specifically in the Database as a Service paradigm. Towards this objective, this paper suggests a

specific database encryption scheme with flexible data granularity and safe key management for high security. The suggested scheme utilizes CA-based chaos systems to accomplish the demand of reliable and secure protection, storage, and transmission of digital data (password in our case) through public networks. This is due to the fact that the chaotic signals have cryptographically desirable features such as high sensitivity to initial conditions and parameters, long periodicity, high randomness and mixing [7]. These features (or properties) make chaos-based data cryptosystems tremendous and robust against statistical attacks. The properties like high randomness, confusion and diffusion needed in conventional cryptographic algorithms are achieved using states of chaotic maps obtained on iterative processing. Furthermore, by employing the concept of CA to generate a pseudorandom signal, fast encryption with multi-key is justified for database records.

III. Methodology

The proposed database encryption scheme comes with possible improvement in the security properties is shown in Fig. 1. It adds strong encryption to both data storage and communication. Based on cellular automata for building symmetric keys, the system introduces a new high speed chaotic cryptographic scheme that requires a little memory capacity, and also appears to be very secure. The chaotic signals possess many desirable features enable chaos based encryption to achieve better confusion and diffusion. The robustness of the system to opponents' attack is enhanced by using a fuzzy–based passwords selection scheme in which the user would be allowed and/or expected to slightly change their password with each attempt. The detailed descriptions of the system's main constituents are discussed below.

A. Encryption Phase

Step 1. Passwords generation: As we all know, the strength of key used to encrypt data is crucial to the security of the algorithm. To make the keys strong enough, the random numbers used to create keys should be "random" enough so as to resist the attacks. As a matter of fact, it is difficult to gain the absolute random number. This step allows us to generate random passwords. The randomness comes from pseudorandom number algorithms as the basic tools of stochastic modeling. Here, we are hired the strong password generation technique by considering multiple input parameters *a*, *b*, and *m* defined as [12]: *a*, *b*, and *m* are initially chosen at random and then fixed, and the seed is the initial value x_0 . These values are changed dynamically for each password generation. The advantage of linear recurrence generators is that they are fast, and it has been shown that they have good statistical properties for appropriate choices of the parameters *a*, *b*, and *m*.

$$x_{n+1} = (ax_n + b) \bmod m \quad (1)$$

Step 2. Password selection using fuzzy logic: this step utilizes a fuzzy logic approach to select the best password from a pool of passwords generated above. Fuzzy engine depends on two parameters: password's length and password's content (numbers, characters or both). Unlike a static password, this type of dynamic password is a password which changes every time the user logs in. Basically, fuzzy logic (FL) provides an effective means of capturing the approximate, inexact nature of the real world [8]. Advantages of fuzzy logic are: simplify knowledge acquisition and representation; a few rules encompass great complexity; and finally can achieve steady state in a shorter time interval. The essential part of the FL is a set of linguistic control rules related by the dual concept of fuzzy implication and the compositional rule of inference.

The FL controller provides an algorithm which can convert the linguistic control strategy based on expert knowledge into an automatic control strategy. The first step in the design of a fuzzy logic controller is to define membership functions for the inputs. Three fuzzy levels or sets are chosen and defined by the following library of fuzzy-set values for the password's length and password's content as shown in Fig. 2. For a given crisp input, fuzzifier finds the degree of membership in every linguistic variable. The number of fuzzy levels is not fixed and depends on the input resolution needed in an application. The larger the number of fuzzy levels, the higher is the input resolution. The fuzzy controller utilizes trapezoidal membership functions on the controller input. The trapezoidal membership function is chosen due to its simplicity.

In general, the actual shape of a fuzzy set depends completely on the semantics of the concept intended to be represented. In other words, there are no universal or pre-defined fuzzy sets. A fuzzy set makes no sense without the context of a system or model, which means that certain shapes are representative of particular classes of knowledge.

The control rules that associate the fuzzy output to the fuzzy inputs are derived from general knowledge of the system behavior. However, some of the control actions in the rule table are also developed using "trial and error" and from an "intuitive" feel of the process being controlled. The derivation of the fuzzy control rules is heuristic in nature and consists of the following rules:

If (Length is Short) and (Content is Num.) then (Password Invalid)

If (Length is Long) and (Content is Char.) then (Password Invalid)

If (Length is Very Long) and (Content is Num.) then (Invalid)

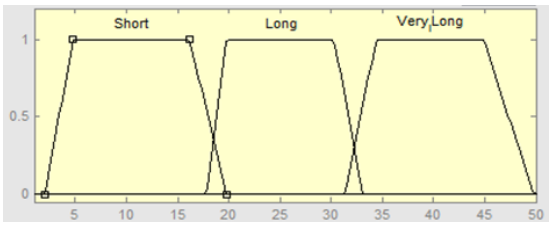
If (Length is Short) and (Content is Num.+ Char.) then (Invalid)

If (Length is Short) and (Content is Char.) then (Password Invalid)

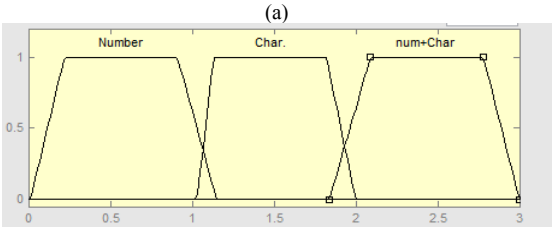
If (Length is Very Long) and (Content is Num.+ Char.) then (Valid)

Step 3. Key generation using Cellular Automata: This step presents a new idea concerning application of chaotic –based cellular automata (CAs) to the secret key using Vernam cipher cryptography that is based on the principle that the plain text of a message is 'mixed' with random text from a One Time Pad (OTP). CA is applied to generate pseudo-random numbers sequence (PNS) based on selected password which is used during the encryption process of the selective records from the database. Cellular Automaton (CA)

is an infinite, regular lattice of simple finite state machines that change their states synchronously, according to a local update rule that specifies the new state of each cell based on the old states of its neighbors. At a particular time each cell in the binary state CA will be in a state 0 or 1 in the simplest case. Two-dimensional, binary states CA that use the nearest neighbors to determine their next state are called elementary cellular automata [9].



(a)



(b)

Fig. 2. Membership functions for (a) password length and (b) password content.

Definition: A homogeneous CA can be represented as a sextuple $\langle \mathrm{T}, S, s, s_0, N, \phi \rangle$ where [17]:

- T is a lattice of a n-dimensional Euclidean space R^n consisting of cells c_i, $i \in N$.
- S is a finite set of k states, often $S \subset N$.
- The output $s : T \times N \to S$ maps the states of cell c_i at discrete time t, i.e., $s(c_i, t)$.
- The function $s_0 : A \to S$ allocates the initial configuration of every cell c_i,i.e., s(c_i)= $s_0(c_i)$.
- The neighborhood function $N : T \to \bigcup_{p=1}^{\infty} T^p$, yields every cell c_i to $N(c_i) = (c_{ij})^{|N(c_i)|}\ j = 1$, with $|N(c_i)|$ distinct call c_i.

The transition function $\phi : S^{|N(c_i)|} \to S$, describes the rules governing the dynamics of every cell c_i in which in our implementation, we utilize "Life-Like" automata that is defined as $\langle Z^2, S = \{0,1\}, s, s_0, moor\ neighborhood, \phi : S^9 \to S \rangle$ where Moore neighborhood considers 8 cardinal direction and the state of the center. For instances, for binary cells c_1, ...,c_8, and c_9 we say that the transition function, at any time t (Game-of-Life) is of the form[18]:

$$s(c_i, t+1) = \phi\left(\left(s(c_i, t)\right)_{j=1}^{|N(c_i)|}\right) = \phi(\sigma_i) \tag{2}$$

$$\text{With} \qquad \sigma_i = \sum_{j=1}^{N(c_i)} s(c_i, t) \tag{3}$$

$$\phi \begin{pmatrix} c_1, & c_2, & c_3 \\ c_4 & c_5, & c_6 \\ c_7 & c_8 & c_9 \end{pmatrix} = \begin{cases} 1, & if \sum_{i=1}^{9} s(c_i, t) = 3 \\ 1, if \sum_{i=1}^{9} s(c_i, t) = 2, i \neq 5 \\ 0, & otherwise \end{cases} \tag{4}$$

In general, Encryption, by theory requires highly complex actions such as permuting, flipping and altering data in such a way that it is undecipherable and provides complex relationship with the original text and the keys. This relationship should be non-linear so that decryption process is as tough as possible. The encryption process must be faster in time and cheaper in terms of the components involved [2]. CA provides a basic structure for highly parallel and complex operations upon which a basic encryption scheme can be built. CA based processor can be used to compute and alter data with high degree of linearity and complexity [9]. At the end of this step we have a key for encryption that has been extracted by CA with password generated from step 2. This password will store within the encrypted file in a random manner for extraction later in the process of decryption to regenerate the same key.

Step 4. Encryption Function: given the sensitive database's fields chosen from the user (various levels of granularity), the selective fields encryption is done by Pseudo Random Number Generators (PRNGs) using CA. In the recommended system we assume that the encryption keys are kept per session and that the table is encrypted at the node (field) level of granularity. The generation of new states in two-Dimensional (2D) CA, can be considered as a sequence of random numbers. Let P be a selected record consisting of m bits $P_1 P_2 ... P_m$ and ϕ be a bit stream of a key. Let c_i be the i^{th} of a cipher–text obtained by apply a $\oplus$ (exclusive-or) enciphering operation: the original bit P_i of a message can be recovered by applying the same operation $\oplus$ on with use of the same bit stream key k: $P_i = C_i \oplus \phi_i$ This enciphering algorithm is known to be perfectly safe if the key stream is truly unpredictable and is used only time. The idea of chaotic masking is to directly add the message (password) in a noise-like chaotic signal at the transmitter (selected database fields). In our case, the use of multiple CA based keys can increase the complexity of the transmitted signal and is likely to improve the security of information sending.

$$C_i = P_i \oplus \phi_i \tag{5}$$

B. Decryption Phase

To a great extent, the security of encrypted data depends on the security of keys. It is unadvisable to store the keys and encrypted data in the same server because of the increased possibility of disclosure at the same time. In our scheme, the password is not stored in the server but is joined in a noise-like chaotic signal at the transmitter (encrypted fields). For deciphering, the system first extracts the noise-like chaotic signal (password) for the received encrypted file. The system combines cryptography and the synchronization of chaotic system based on a fuzzy observer with XOR enciphering operation for decryption.

The system utilizes fuzzy observer for synchronization of CA based chaotic systems. T-S fuzzy model can represent a general class of nonlinear system and we employ it for fuzzy modeling of the chaotic drive system [19]. The difference between the received signal $y(t)$ and the predicted value $\hat{y}(t)$ obtained from the fuzzy observer is sent for

decryption. A fuzzy observer can then be derived to estimate the password $\hat{x}(t)$ that is included with the encrypted message (record) in random manner. For simplicity, the premise variable of observer rules and that of fuzzy rules are assumed to be same. In a fuzzy observer design, a chaotic system should be exactly represented by a *T–S* fuzzy model. Consider a general chaotic system as given below:

$$x(t) = f(x(t)), y(t) = h(x(t)) \tag{6}$$

where $x \in R^n$ is the state vector $y \in R_n$ is the system output; $f(.)$and $h(.)$ are the nonlinear functions with appropriate dimensions. The fuzzy representation of equation (6) is given by of the following rules:

$$\begin{aligned} &if\ z_1(t)\ is\ F_{1i}\ \text{and}\ ...\ \text{and}\ z_g(t)\ is\ F_g\ \text{then} \\ &x(t) = A_i x(t) + b_i \\ &y(t) = C_i x(t) \quad for\ i = 1,2,...,r \end{aligned}$$

where, $z_1(t), z_2(t), \ldots, z_g(t)$ are the premise variables which represent the states of the system; $F_{ji}(j = 1,2,\ldots,g)$ are the fuzzy sets; r is the number of fuzzy rules; A_i and C_i are the system and output matrices with appropriate dimensions; and $b_i \in R^n$ denotes the constant bias term, which is generated by the exact fuzzy modeling procedure. The premise variable of the fuzzy rules satisfies $x_i(t) \in [-d \quad d]$ with d=30. For 2D chaotic [19]:

$$F_1(x_1(t)) = 1/2(1 + (x_1(t)/d))$$

$$F_2(x_1(t)) = 1/2(1 - (x_1(t)/d))$$

$$A_1 = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & -d \\ 0 & d & -8/3 \end{bmatrix} \quad A_2 = \begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & -d \\ 0 & d & -8/3 \end{bmatrix}$$

$C_1 = C_2 = [1\ 0\ 0]$ *and* $b_1 = b_2 = 0$. The output of the fuzzy observer, estimated password, is used to generate the key following the same approach as in Step 3 using the same CA parameters. For the Decryption mode:

$$\begin{aligned} &P_i = E_i^{-1}(C_i \oplus Y_i) \oplus C_{i-1}, \\ &C_0 = Y_i \oplus P_i \quad for\ i = 1,2,...,m \end{aligned} \tag{7}$$

IV. Simulation Results

In this section, the efficiency of proposed encryption method is analyzed. Many different sets of experimental were done to determine the variation in robustness of the proposed system. These cover various aspects including security, encryption efficiency, and system analysis. The simulation results were executed based on the "Northwind_Plaintext" database inside Microsoft SQL Server 2005, which contains seven tables. In the experiments, the algorithm encrypts different fields with various records range from 77 tuples to 1000 tuples from different tables in the database. All programs are implemented in C# and MATLAB. Experiments are performed on a processor Intel(R), Core(TM) i3 CPU, M380 @ 2.53GHz with 2.53GHz. RAM: 4GB in Microsoft windows 7 Ultimate as running operating system, Service pack1

A. Performance Evaluation (Evaluation Criteria)

A comparison has been conducted for those encryption algorithms at encryption and decryption time. The encryption time is considered the time that an encryption algorithm takes to produce a ciphertext from plaintext. It indicates the speed of encryption. The decryption time is considered the time that decryption algorithm takes to produce a plaintext from ciphertext. The correctness of the received passwords in the decryption side (against attacks and channel' noises) is measured through accuracy that is defined as [20]:

$$\text{Accuracy} = \frac{\sum True\ \ Positive + \sum True\ \ Negative}{\sum Total\ Population} \tag{8}$$

where, true positive rate measures the proportion of actual positives which are correctly identified and true negative rate measures the proportion of negatives which are correctly identified. A perfect predictor would be described as 100% true positive and 100% true negative rate; however, theoretically any predictor will possess a minimum error bound known as the Bayes error rate.

Furthermore, encryption Ratio (ER) is utilized to study the impact of the proposed encryption system on the size of the output file (decrypted file). This criterion measures the ratio between the size of the encrypted part and the whole data size. Encryption ratio has to be minimized by selective encryption.

B. Computational Time

The first experiment, shown in Fig. 3 measures the consumed time for generating the key used for encryption and decryption process inside Employee table. It is clarified that time consumed increases when the database size increases and vice versa. Our system requires 0.78 second at maximum for generating the key when the database size is 100*k* (514 record / 4 sensitive fields).

The second experiment was performed to test the applicability of the proposed system. The experiment No. 2 illustrated in Table I and Table II assess the time required for the encryption process for employee and customer tables under different size. With the advised system the time necessary for the encryption process is about 90 and 23 seconds for employee and customer tables respectively at 100k size. For real-time database encryption, the proposed system is with an acceptable speed since it needs chaotic and key generation for encryption.

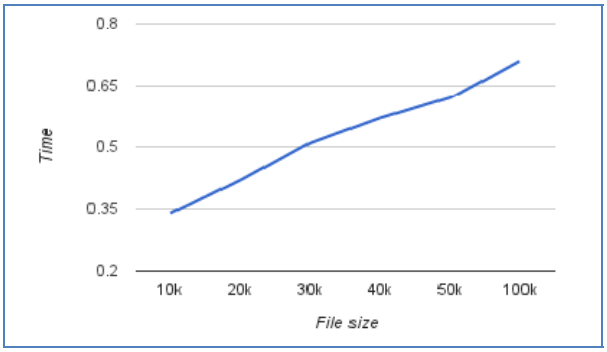


Fig. 3. Consumed time for generating the key within Employee table with different size.

The experiment No. 3 illustrated in Table III and Table V evaluates the time required for the decryption process for employee and customer tables under different size. With our system the time necessary for the decryption process is about

92 and 24 seconds for employee and customer tables respectively at 100k size. For real-time database decryption, the proposed system is with an acceptable speed since it needs CA-based chaotic and fuzzy observer for decryption.

TABLE I: TIME NECESSARY FOR THE ENCRYPTION PROCESS IN EMPLOYEE TABLE (4 SENSITIVE FIELDS)

Table Size	10k	20k	30k	40k	50k	100k
Number of Records	51	103	153	204	257	514
Time (Sec)	1.48	5.28	11.39	21.23	29.78	90.88

TABLE II: TIME NECESSARY FOR THE ENCRYPTION PROCESS OF CUSTOMER TABLE (3 SENSITIVE FIELDS)

Table Size	10k	20k	30k	40k	50k	100k
Number of Records	37	66	94	139	176	352
Time (Sec)	0.88	1.68	3.27	7.57	12.23	23.53

TABLE III: TIME NECESSARY FOR THE DECRYPTION PROCESS IN EMPLOYEE TABLE (4 SENSITIVE FIELDS)

Table Size	10k	20k	30k	40k	50k	100k
Number of Records	51	103	153	204	257	514
Time (Sec)	1.88	5.96	11.89	22.52	30.02	92.61

TABLE V: TIME NECESSARY FOR THE DECRYPTION PROCESS OF CUSTOMER TABLE (3 SENSITIVE FIELDS)

Table Size	10k	20k	30k	40k	50k	100k
Number of Records	37	66	94	139	176	352
Time (Sec)	1.14	1.94	3.54	8.22	13.12	24.62

C. Encryption Ratio

The second set of experiments was conducted to determine encryption ratio of the introduced system to measure the extent of the increase in the size of the encrypted file. The findings in Table IV estimate the encryption ratio for the encryption process. The size of the encrypted file is greater than the size of plain file as a result of adding new symbols from encryption and attached password. It is also clarified that the encryption ratio increases when the database size increase and vice versa.

TABLE IV: ENCRYPTION RATIO FOR EMPLOYEE TABLE (4 SENSITIVE FIELDS)

Table Size	10k	20k	30k	40k	50k	100k
Number of Records	51	103	153	204	275	514
Before sensitive fields encryption (K)	1.84	3.73	5.55	7.43	8.33	18.6
After sensitive fields encryption(K)	2.13	4.33	6.45	8.22	9.88	27.4
Encryption ratio (%)	**1.15**	**1.16**	**1.17**	**1.11**	**1.18**	**1.47**

D. Accuracy

The last set of experiments address the ability of the system to recover the password that is randomly inserted inside the encrypted file. Without any attacks (modifying the encrypted file), our algorithm has 100 % accuracy that showing its capability of truly regenerate the encryption key from the encrypted data in all cases (all fields sizes). Another experiment is done to show the accuracy of our system to regenerate the key at the receiver but this time with automated modification in the encrypted database to simulate the attack behavior. The results are given in Fig. 4, where the accuracy curve for different modification percentage is plotted. The proposed system performs very well up to 2% modification resulting 100 % accuracy, but this ratio decreases when modification percentage increases. For 15% alteration in the encrypted file the accuracy is shifted to 0.

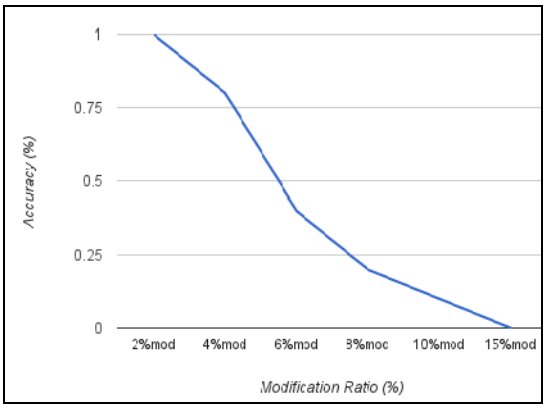


Fig. 4. Accuracy percentage.

E. System Analysis

The new database encryption scheme satisfies the encryption requirements mentioned before as follows:

- The system security relies on the security of the used encryption algorithm. In order to reveal some database value it has to be decrypted using the correct key.
- Encryption and decryption are fast operations and are mandatory in any database encryption scheme. The proposed implementation adds the overhead of a XOR operation which are negligible compared to CA –based key generation.
- The basic element of reference is a database node (field). Operations on a node do not depend on or have any effect on other cells.
- The new scheme prevents patterns matching attacks since there exist a little correlation between a plaintext value and a ciphertext value that is achieved by using chaotic based encryption.
- Unauthorized manipulation on the encrypted data without the encryption key would be noticed at decryption time.
- As the basic element of reference is a database field, it is possible to recover information from partially completed records in the same way as it is recovered from full records.
- The new scheme complies with the structure preserving requirements as the basic element of reference is a database field.
- Although the proposed system is not secure against level-3 attacks where the attacker observes a set of tuples in database and he knows the corresponding encrypted values of those tuples; we can show that it is resilient to common level-2 attacks in which the attacker knows a set of plain tuples in database but he does not know the corresponding encrypted values of those tuples in encrypted database. This information without the knowledge of the key is impossible. In the

proposed system, the key generation depends on password and CA-based chaotic machine. So it is too hard for attacker to use signature linking attack to guess the key.

V. Conclusions

This paper presents a new scheme for database encryption at multi-granularity level. It provides multiple keys and many encryption granularities to overcome the drawback of traditional database encryption schemes that exhibit tradeoff between security and efficiency. In this paper, a fuzzy logic based chaotic encryption method based on the "Life-Like" CA for database encryption was proposed. The cryptosystem we have described is based on two-dimensional. The nonlinear chaotic dynamics are represented as the T-S fuzzy model. The proposed system achieves better result than others based on CA compared in literature. The new schemes do not impose any changes on the database structure, thus enabling a DBA to manage the encrypted database as any other non-encrypted database. The empirical results revealed that the proposed scheme is effective and can be used.

The main advantages of the proposed encryption algorithm are 1) it requires relatively small time for generating the key used for the encryption process. 2) The key length with our method is somewhat long that means that our algorithm is more resistant to cracking and it is more secure. 3) Simplicity and low cost of implementation for the all encryption process. 4) Compression ratio with this system is smaller than other approaches reported in the literature. 5) It achieves robustness performance against the external disturbance. The future work has to be carried for improving our proposed system for resistance against unwanted modification to the data.

References

[1] E. Shmueli, R. Vaisenberg, Y. Elovici, and C. Glezer, "Database encryption – an overview of contemporary challenges and design considerations," *Journal of the Sigmod Record*, vol. 38, no. 3, 2009, pp. 29-34.

[2] R. Ravan, N. Idris, and Z. Mehrabani, "A survey on querying encrypted data for database as a service," in *Proc. International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing*, Oct. 2013, pp. 14-18 .

[3] L. Bong and Y. Guo, "Database encryption," *Encyclopedia of Cryptography and Security*, 2009, pp. 307 -312.

[4] A. P. Deshmukh and R. Qureshi, "Transparent data encryption-solution for security of database contents," *International Journal of Advanced* Computer *Science and Applications*, vol. 2, no. 3, March 2011, pp. 25-28.

[5] G. Chen, K. Chen, and J. Dong, "A database encryption scheme for enhanced security and easy sharing," in *Proc. the 10th International Conference on Computer Supported Cooperative Work in Design*, China, 2006, pp. 995-1000.

[6] M. Machicao and A. Marco, "Chaotic encryption method based on life-like cellular automata," *International Journal of Expert Systems with* Applications, vol. 39, no. 16, November, 2012, pp. 12626–12635.

[7] G. Yu, "Robust chaotic cryptosystems based on T-S fuzzy model," in *Proc. the 17th World Congress the International Federation of Automatic Control Seoul*, Korea, July 2008, pp. 6-11.

[8] K. Y. Lian, P. Liu, T. C Wu, and W. C. Lin, "Chaotic control using fuzzy model-based methods," *International Journal of Bifurcation and Chaos*, vol. 12, no. 8, 2002, pp. 1827-1841.

[9] S. Billings and Y. Yang, "Identification of probabilistic cellular automata," *IEEE Trans. on Systems Man and Cybernetics*, Part B: Cybernetics, 2003, vol. 33, no. 2, pp. 225-236.

[10] M. Patil1 and S. Ingale, "Privacy control methods for anonymous & confidential database using advance encryption standard," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 8, pp. 224-229, August 2013.

[11] Y. Elovici1 and R. Waisenberg1, "A structure preserving database encryption scheme," in *Proc. the Secure Data Management (VLDB)*, Canada, August 30, 2004, pp. 28-40.

[12] P. Bhagat and K. Satpute, "Reverse encryption algorithm: a technique for encryption & decryption, "*International Journal of Latest Trends in Engineering and Technology*, vol. 2, no. 1, January 2013, pp. 90-95.

[13] L. Liu, and J. Gai, "A new lightweight database encryption scheme transparent to applications," in *Proc. the IEEE International Conference on Industrial Informatics*, Korea, July 2008, pp. 135-140.

[14] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure KNN computation on encrypted databases," in *Proc. the International Conference on Management of Data* , USA, 2009, pp. 139-152.

[15] R. Pabboju and S. Rao, "A modular and extendable approach to database encryption at multi-granularity level," *International Journal of Computer Trends and Technology*, vol. 3, no. 6, pp. 769-773, 2012.

[16] E. Shmueli, R. Waisenberg, Y. Elovici, and E. Gudes, "Designing secure indexes for encrypted databases," in *Proc. the Working Conference on Data and Applications Security*, USA, 2005, pp. 54-68.

[17] S. Bandin, "Cellular automata: future generation," *Computer Systems*, vol. 18, 2002.

[18] S. Chattopadhyay, S. Adhikari, S. Sengupta, and M. Pal, "Highly regular modular, and cascadable design of cellular automata-based pattern Classifier," *IEEE Transaction on VLSI Systems*, vol. 8, no. 6, pp. 724–735, December 2000.

[19] V. Natarajan and P. Kanagasabapathy, "Fuzzy observer design with n-shift multiple key for cryptography based on 3d hyper chaotic oscillator," *Iranian Journal of Fuzzy Systems*, vol. 3, no. 2, 2006, pp. 21-32.

[20] S. Jacob, "Cryptanalysis of a fast encryption scheme for databases and of its variant," in *Proc. Cryptology Reprint Archive (IACR),* vol. 3, no. 4, 2010, pp. 1-7.

Adel A. El-Zoghabi received his B.Sc. in computer engineering from Alexandria University in 1987, he received his M.Sc. and Ph.D. in information technology from Alexandria University and Old Dominion University in 1991 & 1994 respectively. His research and professional interests include intelligent systems and machine learning, internet working and routing protocols, and distributed systems. He has published many papers in international journals and conferences worldwide during the past three decades. Currently he is a professor of computer science and IT and the head of Dept. of Information Technology since August 2012.

Saad M. Darwish received his Ph.D. degree from the Alexandria University, Egypt. His research and professional interests include image processing, optimization techniques, security technologies, and machine learning. He has published in journals and conferences and severed as TPC of many international conferences. Since Feb. 2012, he has been an associate professor in the Department of Information Technology, Institute of Graduate Studies and Research, Egypt.

Mohammed A. Abdewi received the B.Sc. degree in computer science from the Department of Computer Science, the College of Computers, University of Alanbar, Iraq in 2005. Currently he is a M.Sc. student in the Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Egypt. His research and professional interests include database encryption and intelligent systems.