Hindawi
Journal of Applied Mathematics
Volume 2023, Article ID 5187602, 10 pages
https://doi.org/10.1155/2023/5187602

Research Article

SQLite Encryption Method for Embedded Databases Based on Chaos Algorithm

Junlong Shi

Information and Data Management Center, Bengbu University, Bengbu 233000, China

Correspondence should be addressed to Junlong Shi; bbxysjl@bbc.edu.cn

Received 31 October 2022; Revised 14 January 2023; Accepted 16 January 2023; Published 17 February 2023

Academic Editor: Wei-Chiang Hong

With the widespread use of embedded systems, chaos is a nonlinear system with certainty and complexity. It is an important topic in the field of information security at present, and it is an effective way to apply to embedded systems. It has great practical value in theory and in practice. This research mainly focuses on the encryption technology of SQLite embedded database and proposes an improved sparrow algorithm (Logistic Chaos Sparrow Search Algorithm, LCSSA) based on Logistic Chaos Map. It shows that the security level of SQLite in web development is higher than that of conventional Access. The population is initialized by the logistic chaotic mapping method, which improves the quality of the initial solution, increases the diversity of the population, and reduces the risk of premature maturity of the algorithm. The initial value y_0 determines the encryption method of the nonlinear function. Taking the integer variable (int) as an example, the value range is -2^{31}~231. It can be seen that the key space is sufficient to prevent various conventional attacks. When the key is the wrong key, decryption will not yield any data. It can be found that encryption and decryption are very sensitive to the key, which is also determined by the sensitivity of chaotic encryption system to the initial value. The benchmark function compares the performance of the improved algorithm with the algorithm before the improvement and compares it with the SSA. The LCSSA has better convergence performance, higher accuracy, and better stability.

1. Introduction

With the advent of the 21st century, embedded technology has been widely used and has gradually become a development trend in communication and consumer electronic devices, where analogue technology has been completely replaced by digital technology in communication [1]. European DVB technology has been commonly adopted internationally. Dab has started commercial trials, and software, integrated circuits, and new electronic components are gaining prominence in industrial development [2], all of which are inextricably linked to embedded technology. In individual applications, embedded products will be the means of transmission and delivery of personal information [3]. Chaos is an inherently stochastic and nonlinear system that will not be swayed by any external factors [4]. The most important feature of chaotic systems is their sensitivity to initial conditions, which makes them more uncertain. Chaotic signals have a strong influence on the initial values, with good pseudorandomness and spectral characteristics, and the simple structure and fast computation of chaotic communication systems make them a more desirable solution for the design of symmetric key ciphers [5]. Based on this, the study is aimed at proposing a chaos algorithm-based SQLite encryption method for embedded databases, thus improving it based on logistic chaotic sequences, adding optimal nonlinear operations, increasing the complexity of the algorithm, and greatly enhancing the security of chaotic ciphers.

2. Related Work

Chaotic cryptography is an important branch in the field of chaos, which is developing rapidly and has made great progress both theoretically and practically. Especially in recent years, a large number of new ideas and methods have emerged in terms of both practicality and security, thus accelerating its application. Mabrouki et al. proposed an

embedded system based on Internet technology, whose main goal is to sense climate parameters such as temperature, humidity, and the presence of certain gases based on sensors and then send the captured values to a remote application or database for visualization [6]. Yang et al. proposed a novel class-level joint representation framework to exploit the uniqueness of different facial features and class-level commonality in the proposed class-level joint representation with region-adaptive convolutional features, leveraging robust representations of discriminative facial features with robustness to facial variation and classification with generic facial variation. Experimental results show that the proposed method has higher robustness and effectiveness compared with state-of-the-art methods [7]. Rezaei et al. studied the production of miRNAs in the F8 gene by bioinformatic prediction and experimental validation and hoped to confirm the exogenous and endogenous expression of the predicted miRNAs by changing the experimental conditions, designing new primers, or changing the expression of cell lines as well as vectors [8]. Mama and Machkour proposed a simple and intelligent method to extend the SQL language to allow us to write flexible conditions in queries without translation in order to use fuzzy language values in all clauses of the select statement [9]. Espinosa et al. analyzed Smac/DIABLO mRNA expression in 57 frozen tissues by qPCR, while immunohistochemistry was used to assess 82 protein levels in paraffin-embedded tissues. It shows that their mRNA and protein levels of Smac/DIABLO are elevated in ER-positive breast tumours compared with ER-negative samples, but the mechanism of this regulation is unclear. Public databases suggest that the association may have clinical relevance [10].

Zhao et al. developed a method for solving the 3D irregular packing problem. A three-mesh approximation technique was first introduced to approximate the irregular objects, and then, a hybrid heuristic was developed to embed chaotic search into the firefly algorithm for each individual object placement and compression to enhance the versatility of the algorithm to optimize the packing order and orientation [11]. Tuerxun et al. used a Supervisory Control and Data Acquisition (SCADA) system to obtain operation and maintenance data, which contains rich information related to the operating characteristics of wind turbines. Experiments showed that the SSA-SVM diagnostic model can be used for fault diagnosis in practical engineering applications [12]. Wang et al. used Bernoulli chaos method to solve for the economics of microgrid clusters, and experiments proved that this method has fast convergence, high computational accuracy, and better overall performance. Finally, the effectiveness of the improvement was verified by calculating examples that yielded an improvement of nearly 20% in the benefits of microgrid clustering [13]. Liu et al. proposed a computer-aided system for the automatic diagnosis of brain tumours, where the CNN was optimized by a newly designed Sparrow Search Algorithm classification (ESSA). Finally, the results of this method were compared with three state-of-the-art techniques on the Whole Brain Atlas (WBA) database, which were shown to be more efficient [14]. Yuan et al. proposed a distributed maximum power point tracking (DMPPT) method based on the Improved Sparrow Search Algorithm (ISSA) for the problem of power mismatch loss under local shading in photovoltaic microgrid system (DMPPT) method. The model results in MATLAB show that ISSA is more accurate than the regenerative observation (P&O) and particle swarm optimization (PSO) algorithms [15].

The Sparrow Search Algorithm (SSA) iterates through individual sparrows searching for food and antipredation and has the advantages of few adjustment parameters, fast convergence, and simple computation; however, like other swarm intelligence algorithms, it tends to be "premature" when solving complex engineering optimization problems, resulting in poor convergence accuracy and easy to fall into local optimum solutions. When solving complex engineering optimization problems, resulting in low convergence accuracy and easy to fall into the local optimal solution, the update mode of SSA can be roughly divided into two kinds: (1) close to the current optimal position and (2) close to the origin. Through simulation experiments, it can be seen that when performing the optimal trajectory solution of the trajectory model, each convergence is a direct jump to the vicinity of the current optimal solution, which is easy to lose the global optimal trajectory solution, and it is only possible to obtain a feasible solution that satisfies the constraints, and there is a probability that no feasible solution will be obtained. It can be seen through the research of scholars at home and abroad on embedded databases and chaotic algorithms that there is relatively little research on combining the two. Therefore, this study focuses on the SQLite encryption method for embedded databases based on chaotic algorithms. A linear decreasing weighting method is used, which can effectively reduce the risk of premature maturity of the group intelligence algorithm and reduce the vibrations generated near the global optimal solution.

3. Construction of SQLite Encryption Method for Embedded Database Based on Chaos Algorithm

3.1. Establishment of the SQLite Architecture for Embedded Databases. SQLite is a powerful embedded relational database management system invented by D. Richard Shipp in 2000. Firstly, the SQL database engine is a fully, integrable, zero-configuration SQL database engine with the following features, even after a system failure or power failure. Secondly, there is zero configuration, which means that there is no need for installation and administration. Thirdly, the system uses most of the standards of SQL92. Each individual hard disk file can hold a complete database and can be freely shared between different computers. We can see that SQLite builds well on the above features for embedded development. SQLite is licensed under copyright without any restrictions, including commercial products. Compiled software for SQLite can be downloaded from the official SQLite website (http://www.sqlite.org) [16].

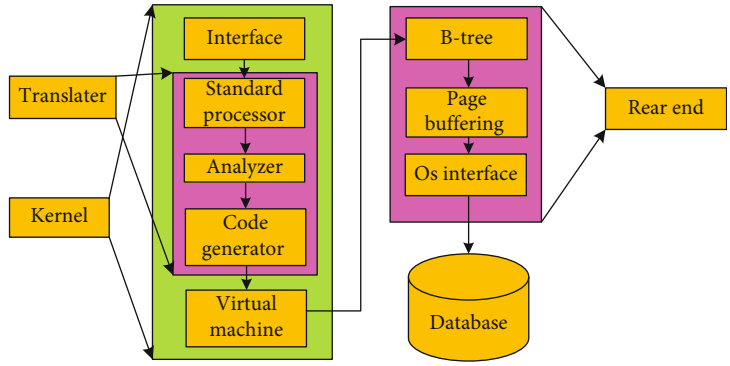


Figure 1: Structure of an SQLite database.

Figure 1 shows the SQLite architecture. It consists of interface, which is a library in C that accepts instructions from an interface and then passes them to the compiler. It consists of three distinct steps: a token processor, a parser, and a code generator. The virtual machine is an abstract computer engine for running database files. SQLite uses a layer of abstraction in the virtual machine and the lower memory reduction to complete the B-tree, the web buffer, and the OS interface. The third part is the database for data storage. SQLite is a lightweight database for embedded use and is currently used in many embedded products. SQLite is used in other applications that are also widely used. For example, MySQL is no longer supported by default in PHP5, but SQLite is supported by default. The widespread use of SQLite provides ample assurance of database security, and SQLite takes a smart and efficient approach to securing data in Windows [17]. With an understanding of the Windows API, it can be seen that a strong cryptographic feature can easily be added to the application through a Cryptography API provided by Microsoft, without having to take into account the underlying algorithms. SQLite uses this approach to protect databases under Windows. In effect, CryptoAPI is a set of features that provides the programmer with an interface to access cryptographic algorithms that pass through an operating system and finally through a CSP, which is a standalone component that enables real cryptographic operations. At the operating system level, Microsoft has also introduced a CSP, the public key cryptographic algorithm from RSA, which allows more CSPs to be added to the program. At present, Microsoft has developed hardware products compatible with the CryptoAPI for several companies, such as the uKey300 USB port from Quandian and the iSecure XCSP [18].

Figure 2 above shows the architecture of CryptoAPI. Using CryptoAPI, operations such as encryption, exchange of public keys, message aggregation, and electronic signatures can be performed on data. It is also possible to provide advanced management operations, such as selecting a set of town energy CSPs [19]. Since SQLite does not perform any type of inspection, it makes automatic determinations based on user input. When users use SQLite to create data tables, it is best to set the data category to communicate with other designers. SQLite supports the following five data types, NULL, INTEGER, TEXT, REAL, and BLOB, to conduct most common data processing in SQL statements. SQLite is a document-based database. SQLite enables access to the file system through the operating system, and its access methods have certain security issues. Password-based authentication technology is the primary guarantee of security management of the database. To access the database, users must enter the appropriate password to modify, delete, insert, and query the database before accessing it. Figure 3 shows the basic principles of internal and external encryption and decryption of an embedded database management system.

In order to ensure the security of embedded databases, it is essential to choose appropriate encryption techniques. The security of a password is determined by the robustness of the cryptographic algorithm. A good encryption algorithm must be secure and effective, otherwise the user will not be able to afford a sluggish response to data processing, and the key management of a good encryption algorithm must not be overly complex and must have simplicity, otherwise the key management issues will become very tricky [20].

3.2. Embedded Database SQLite Encryption Algorithm Design and Implementation. When communication is carried out, it is inevitable that data will be invaded, so encryption of the product's performance is a must. Due to the large amount of data transmission and high speed, encryption algorithms must be carried out on the basis of ensuring basic security. Because they cannot be too complex, complex encryption techniques such as DES and asymmetric public key ciphers are not suitable for inclusion in embedded systems. However, the traditional chaotic encryption method is a relatively simple encryption method with poor security performance, so it becomes a new encryption method based on the existing chaotic cryptosystem and optimizing it. The study proposes a sparrow search encryption method based on logistic chaos optimization, which calculates the chaotic sequence, the key, and the encrypted ciphertext. Then, to improve the randomness and security of the cipher, the data is further encrypted, and the main encryption flow is shown in Figure 4.

As shown in Figure 4, the algorithm uses chaotic sequences and nonlinear operations of the cipher to achieve

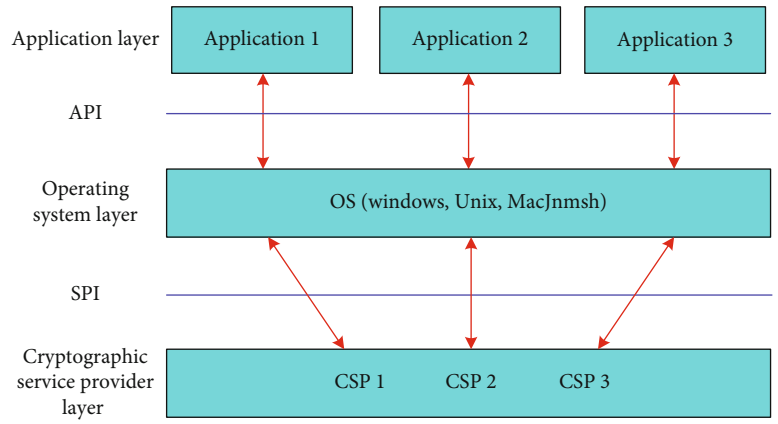
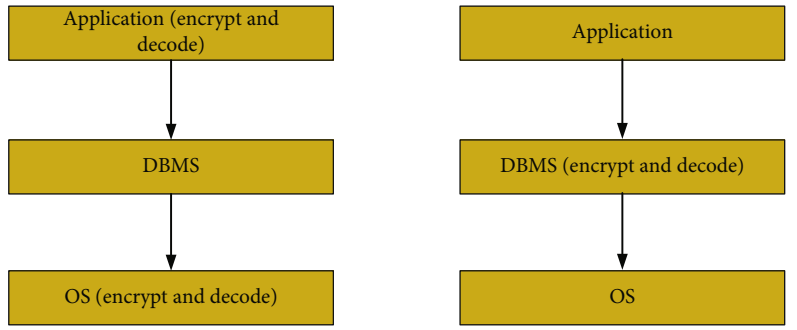


Figure 2: Architecture of CryptoAP1.



(a) DBMS outer layer encryption and decryption (b) DBMS kernel layer encryption and decryption

Figure 3: Schematic diagram of encryption and decryption of the inner and outer layers of DBMS.

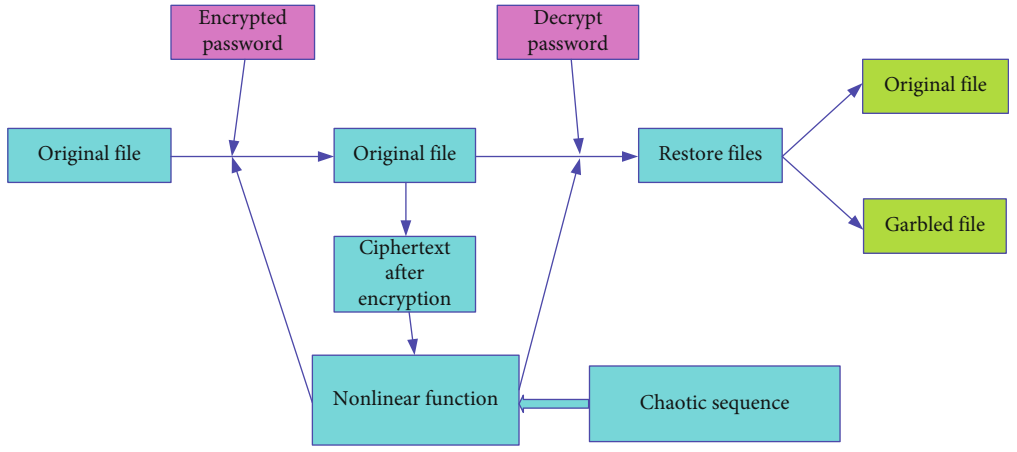


Figure 4: Schematic diagram of encryption process.

better hiding effects. At the beginning of the file, an initial password, i.e., an encrypted password, is set. The key to this method is the selection of an appropriate nonlinear function, which reduces the computational effort of the algorithm and improves the encryption efficiency. The study introduces the basic principles of the Sparrow Search Algorithm (SSA) [21]: (1) the Sparrow Search Algorithm divides the colony into a discovery colony and an input colony. The more resources the discovery colony has, the greater the direction the colony will search, while the join colony will follow the actions of the discovery colony. (2) The sparrow colony contains a certain percentage of warning sparrows. In warning sparrows of prey sightings, or if the warning value exceeds the alert value, discoverers will lead the colony in other possible

directions. Within the safety zone, sparrows on the periphery of the colony will move quickly towards safety, while those in the middle of the colony will wander around as the colony moves. (3) The status of discoverers and joiners changes with each iteration, with those having more resources become discoverers. However, the overall ratio of the two remains unchanged. (4) The fewer resources a participant has, the lower its status in the group and the easier it is to access resources in other areas. (5) During the group search, participants always look for food around the discoverer and compete with the discoverer for resources [22]. A matrix is used to describe a $n \times d$-dimensional vector population that includes n sparrows, as shown in the following equation:

$$X = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1d} \\ X_{21} & X_{22} & \cdots & X_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nd} \end{pmatrix}. \tag{1}$$

In Equation (1), X_{ij} is the coordinates of the i sparrow in the j dimension, n is the number of sparrows in the population, and d is a variant dimension of the target function that can be used to describe the fitness of the sparrow population as shown in the following equation:

$$F_X = \begin{pmatrix} f((X_{11} & X_{12} \cdots X_{1d})) \\ f((X_{21} & X_{22} \cdots X_{2d})) \\ & \vdots \\ f((X_{n1} & X_{n2} \cdots X_{nd})) \end{pmatrix}. \tag{2}$$

In Equation (2), f is its sparrow individual fitness. When the SSA is optimized, individuals with higher fitness will preferentially pick food as the found individual gives the whole colony direction to look and find, so the finder's search range is greater than the selected one, and the equation for updating the finder's position at iteration is shown in the following equation:

$$X_{Fi,j}^{t+1} = \begin{cases} X_{Fi,j}^{t} \cdot \exp\left(-\dfrac{i}{\alpha T}\right) & R < \mathrm{ST}, \\ X_{Fi,j}^{t} + Q & R \geq \mathrm{ST}. \end{cases} \tag{3}$$

In Equation (3), t is the current number of iterations; T is the maximum of the iterations; $\alpha \in (0, 1]$ is a uniform random number; $R \in [0, 1]$ and $\mathrm{ST} \in [0.5, 1]$ are the alert value and safety, respectively; and Q is a random variable that follows the normal assignment.

When $R < \mathrm{ST}$, the species is surrounded by no natural predators and its foraging environment is relatively safe, so the finder can conduct an extensive search. When $R \geq \mathrm{ST}$, some sparrows in the group have detected a prey, started to notify the rest of the group, put the group into "antipredator" mode, and must quickly find a safe area [23]. The inductees, on the other hand, influence their behavioral characteristics, and their position update formula is shown in the following equation:

$$X_{Ji,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\dfrac{X_L^t - X_{Ji,j}^t}{i^2}\right), & i > 0.5n, \\ X_P^{t+1} + \left|X_{Ji,j}^t - X_P^{t+1}\right| \cdot L \cdot A^+, & i \leq 0.5n. \end{cases} \tag{4}$$

In Equation (4), X_P is the best position of the person currently found, X_L is the worst position in the current global picture, L is a matrix with the dimension $1 \times d$, and all elements are 1, A is a matrix with the dimension $1 \times d$, and each cell element is either 1 or -1, $A^+ = A^T(AA^T)$. When $i > 0.5n$, the ith joiner is poorly adapted and does not have enough food. It is in a very hungry state and must fly elsewhere to replenish [16]. When, $i \leq 0.5n$, the ith participant will randomly search around X_p. Within the colony, there are sparrows that act as scouts and early warnings, alerting the whole colony and guiding the colony into a new safe zone. 10%-20% of the sparrows are randomly selected for early warning in each generation, and their position update formula is shown in the following equation:

$$X_{Di,j}^{t+1} = \begin{cases} X_B^t + \beta \cdot \left|X_{Di,j}^t - X_B^t\right|, & f_i > f_g, \\ X_{Di,j}^t + K \cdot \left(\dfrac{\left|X_{Di,j}^t - X_L^t\right|}{(f_i - f_w) + \varepsilon}\right), & f_i \leq f_g. \end{cases} \tag{5}$$

In Equation (5), X_B is the current global best position, f_i is the current adaptation of the sparrow, f_g and f_w are the current best and worst individual adaptations of the sparrow, β is a random number that is parametrically controlled for step size and normally distributed with a variance of 1 and a mean of 0, and $K \in [-1, 1]$ is a random number. ε is an infinitesimal constant, mainly to prevent the denominator of the equation from generating 0. When $f_i > f_g$, such sparrows are at the edge of the colony and are vulnerable to predators. When $f_i \leq f_g$, sparrows in the center of the flock feel in danger and must stay close to other sparrows in the flock in order to reduce their chances of being hunted. The logistic chaos algorithm is one of the most representative ones with simple mathematical expressions and wide applications. The encryption steps of the chaos algorithm are shown below, first for logistic mapping, as shown in the following equation:

$$x_{n+1} = \lambda x_n(1 - x_n), \quad \lambda \in (0, 4), x_n \in [0, 1]. \tag{6}$$

As shown in Equation (6), when $\lambda \geq 3.61$, the system is in chaotic mode. Assuming that λ is 3.7 and x_n is 6 significant digits after the decimal point, a particular chaotic sequence can be obtained at X_n. Therefore, in order to

make the cryptographic algorithm better suited to the needs of the system, it is necessary to choose the appropriate function. The simplest quadratic function is first given here as Equation (7); then, the nonlinear function used for this cipher is Equation (8).

$$Y = X^2, \tag{7}$$

$$y_n = \left[(y_{n-1} + x_n{}^*1000000)^2 + a_n\right]\%256. \tag{8}$$

As shown in Equations (7) and (8), y_{n+1} is its currently encrypted data, y_{n-1} is its currently last encrypted data, and x_n is its nth chaotic order. Because x_n is a floating point number with 6 decimal digits, it must be multiplied by 1,000,000 to make it an integer. a_n is its current original data, satisfying the conditions. $a_n \in [0, 255]$ is the data after plaintext encryption. At the start of encryption, each value is first initialized, so$x_n \in [0, 1]$and$x_1 \in [0, 1]$. Assuming $x_1 = 0.5y_{n-1}$ is its last encrypted data, but since the data y_0 has not been encrypted before y_1, an arbitrary data is chosen (0~255 is recommended) and used as the encryption key [24]. Equation (9) and Equation (10) are for reasoning until the end of encryption.

$$\begin{cases} y_1 = \left[(y_0 + x_1{}^*1000000)^2 + a_1\right]\%256, \\ x_2 = \lambda x_1(1 - x_1), \quad \lambda = 3.7, \end{cases} \tag{9}$$

$$\begin{cases} y_n = \left[(y_{n-1} + x_n{}^*1000000)^2 + a_n\right]\%256, \\ x_{n+1} = \lambda x_n(1 - x_n), \quad \lambda = 3.7. \end{cases} \tag{10}$$

Decryption is a kind of antiencryption process that starts with the first data, which otherwise cannot be decrypted. At the same time, decryption requires knowledge of the chaotic sequence used in encryption, the previous encrypted data (containing the key y_0), so the same chaotic sequence needs to be generated at the decryption end as at the encryption end. Namely, the initial data used in encryption x_1 and the parameter λ are necessary to find the corresponding decryption formula as shown in Equation (11), in the case of the above encryption method.

$$\begin{cases} a_n = \left[(y_{n-1} + x_n{}^*1000000)^2 + y_n\right]\%256, \\ x_{n+1} = \lambda x_n(1 - x_n), \quad \lambda = 3.7. \end{cases} \tag{11}$$

Following the above formula, the entire data can be decrypted in sequence until it is all decrypted. Generally, due to hardware resource constraints, applications in embedded systems are unlikely to run a program compiler such as gcc on the target system, and there is a strong desire to write the OS and programs into the embedded environment. In addition, developers are reluctant to develop or purchase special packages to develop a dedicated embedded hardware platform. Therefore, in most embedded systems, crossdevelopment becomes the best solution. Thanks to cross-disciplinary development, programmers can write software for other different hardware systems on the general-purpose PC platform of the X86 [22, 25, 26]. Although the hardware environment of CN1010 allows compilation and debugging on a specific platform, the aim is to simplify, save resources, and facilitate debugging. CN1010 can still be used for application development on a PC. The crosscompilation principle of CN1010 is shown in Figure 5.

It is possible to run a program or an executable on two systems simultaneously in the same development environment if the following conditions are met. The first is that the executable is dynamically linked, and the second is using the same glibc with the -mhard-float compile option turned on for compiling libraries. The third is on the Linux kernel, where floating point operations are emulated. After meeting the above conditions, it is possible to share programmable resources between the embedded system and the general PC.

4. Performance Evaluation of SQLite Encryption Methods for Embedded Databases Based on Chaotic Algorithms

Because the application of SQLite in the web is similar to the function of Access database, here, we compare SQLite and Access to verify the security of SQLite in web development. The test environment is Windows XP, IE6.0, and the network speed is 10 mbps. The test content is a comparison of SQLite and Access databases are compared. The results of the tests are shown in Table 1 (in seconds).

As can be seen from Table 1, SQLite is more secure than regular Access in web development and is very different from large DBMS such as Oracle and IBM DB2, but SQLite is only a small database and it is difficult to achieve such security performance, so the embedded database SQLite architecture proposed in this study is feasible, and the following is a performance analysis of the improved sparrow algorithm based on logistic chaos. The improved sparrow algorithm based on logistic chaos is compared with the original sparrow algorithm, where the number of correlation coefficients between SSA and LCSSA is the same. The algorithm discovery rate is set to 20% for both groups. The probability of warning is 20%, and the warning value is 0.6. The team size is 50 with 100 repetitions. After setting the parameters, reference functions were selected to test the performance of the sparrow algorithm before and after the improvement, and four reference functions were selected to reflect the reasonableness of the test, as a way to conduct a specific performance analysis of the improved algorithm. Each reference function was set as shown in Table 2.

In order to overcome the randomness of the test results, eight reference functions were tested in the same environment. The test was set to I5-8265U CPU@1.60 GHz and 1.80 GHz, with 4.00 GB of memory, and 20 independent runs were performed using MATLAB R2018a to obtain the mean and standard deviation, as shown in Table 3.

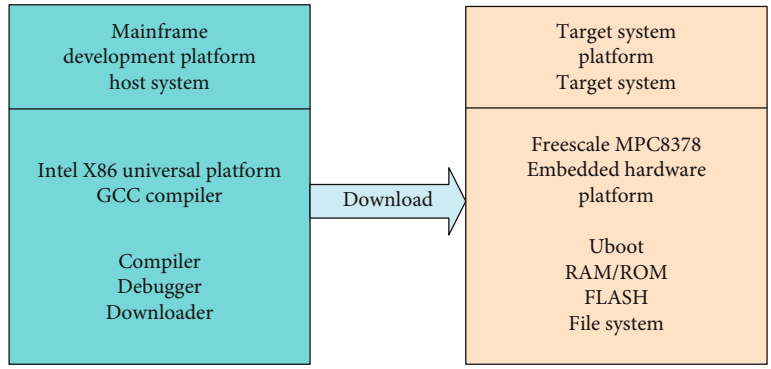


Figure 5: CN1010 crosscompilation diagram.

Table 1: Compare experimental results.

Database	SQLite	Access
Easy to get	Download directly from the Internet	Download directly from the Internet
Easy to decrypt	The encryption mechanism is relatively complete, the password has no fixed rules, and it is difficult to decrypt	The encryption mechanism is relatively simple; even if a password is set, decryption is easy
Data read	Notepad, etc.	Office tools
Data storage	Binary stream	Mdb file, easy to read

Table 2: Eight benchmark function parameter settings.

Benchmark function	Ranges	Dimension	Optimal extreme value
F_1	[-110, 110]	35	0
F_2	[-20, 20]	35	0
F_3	[-110, 110]	35	0
F_4	[-110, 110]	35	0

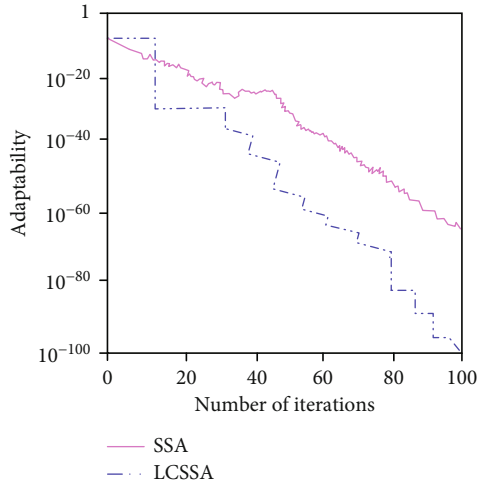
As can be seen from Table 3, LCSSA is more stable than SSA under the same test conditions. After analyzing its performance, for the functions F_1, F_2, F_3, and F_4, the mean and standard deviation of LCSSA are tens of orders of magnitude lower than those of SSA, which indicates that LCSSA has greatly improved its optimization-seeking accuracy and reduced volatility and randomness while finding the overall suboptimal solution. Therefore, LCSSA enhances the algorithm's stability and global optimization-seeking performance. The convergence of the SSA and LCSSA was compared, and their iterative convergence rate curves were derived, as shown in Figure 6.

As can be seen in Figure 6, with reference to the functions F_1, F_2, F_3, and F_4, the LCSSA converges much faster than SSA in the initial stage and can perform better search for optimal solutions in the search space. The algorithm is improved by using logistic chaos mapping and the LCSSA, which makes the global search performance of LCSSA better than that of SSA and also better than SSA in terms of jumping out of local optimization. Moreover, the LCSSA algorithm converges faster than the SSA algorithm.

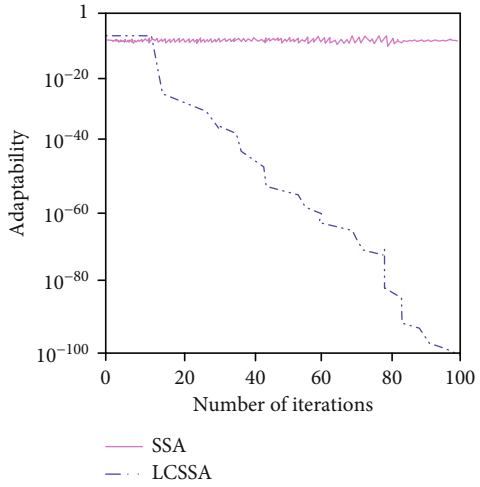
Figure 7 shows the phase space of the logistic mapping and the improved algorithm for comparison. From Figure 7(a), it can be seen that the phase space of the logistic mapping is constructed as a single-peak structure and therefore can be used for approximate analysis and prediction of chaotic signals using neural networks, reconstructed phase space, nonlinear regression, etc. The study has improved the algorithm, so that LCSSA has a global search in better performance and thus overcomes this challenge. The improved algorithm has a larger key space and mapping space compared to the logistic mapping. In the encryption algorithm of this paper, the initial values of chaotic encryption y_0 and the parameter λ are used as the key of the encryption system. The parameter determines the difference of the generated chaotic sequence. The value range is 3.57~4.0, valid after six decimal places. The initial value y_0 determines the encryption method of the nonlinear function. Taking the integer variable (int) as an example, the value range is -2^{31}~231. It can be seen that the key space is sufficient to prevent various conventional attacks. When the key is the wrong key, decryption will not yield any data. It can be seen that encryption and decryption are very sensitive to the key, which is also determined by the sensitivity of chaotic encryption system to the initial value. Finally, the study compared the accuracy of the random

Table 3: Test results of SSA and LCSSA under 8 benchmark functions.

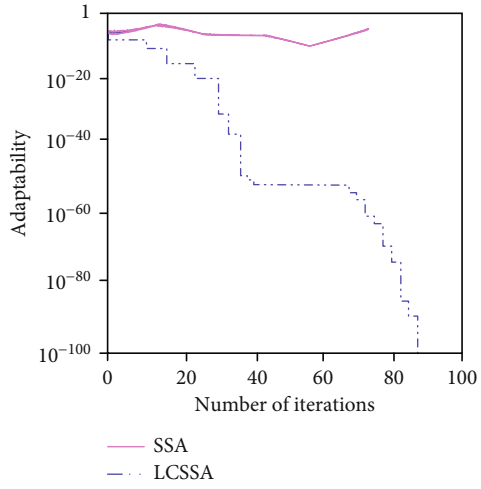
Benchmark function	SSA		LCSSA	
	Average value	Standard deviation	Average value	Standard deviation
F_1	5.945E-32	2.631E-31	0.001E + 01	0.000E + 00
F_2	4.895E-08	4.130E-06	4.695E-50	7.533E-42
F_3	2.135E-13	9.453E-13	2.693E-91	3.963E-82
F_4	1.485E-09	2.698E-06	1.315E-48	7.931E-41



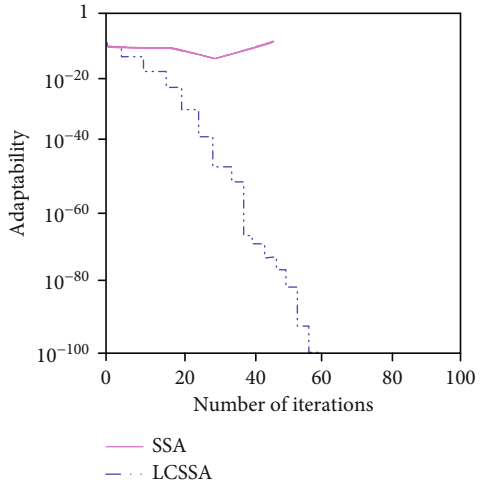
(a) Iterative convergence curve when F_1 is used as the benchmark function



(b) Iterative convergence curve when F_2 is used as the benchmark function



(c) Iterative convergence curve when F_3 is used as the benchmark function

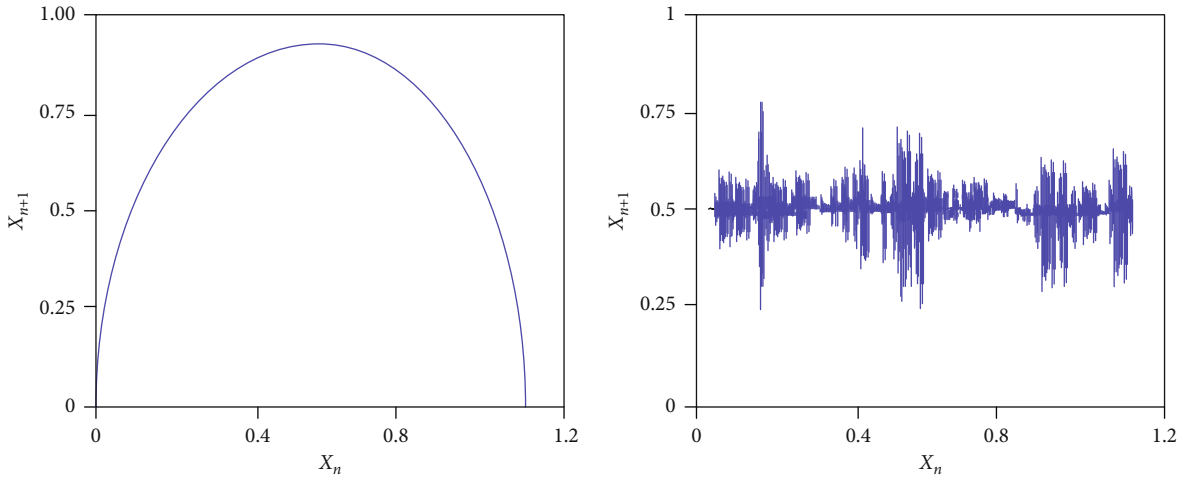


(d) Iterative convergence curve when F_4 is used as the benchmark function

Figure 6: Iterative curves of SSA and LCSSA under 4 benchmark functions.

decision forest (RDF) algorithm and LCSSA, and the results are shown in Figure 8.

From the relative error histogram in Figure 8, it is clear that none of the absolute values of the relative errors of the LCSSA proposed in this study exceed the 5% limit for each evaluation indicator. Only the relative errors of the individual evaluation indicators are problematic with values of 4.52% and -3.84%, while the relative errors of the rest of the indicators remain at a low level and their errors are much smaller than those of the RDF algorithm. In summary, it can be seen that the LCSSA proposed in the study has excellent results in targeting English education and assessment in universities and colleges, and the evaluation model has faster learning rate, higher prediction accuracy, and more stable performance.



(a) Logistic mapping (b) Improved algorithm

Figure 7: Phase space structure diagram of chaotic map.

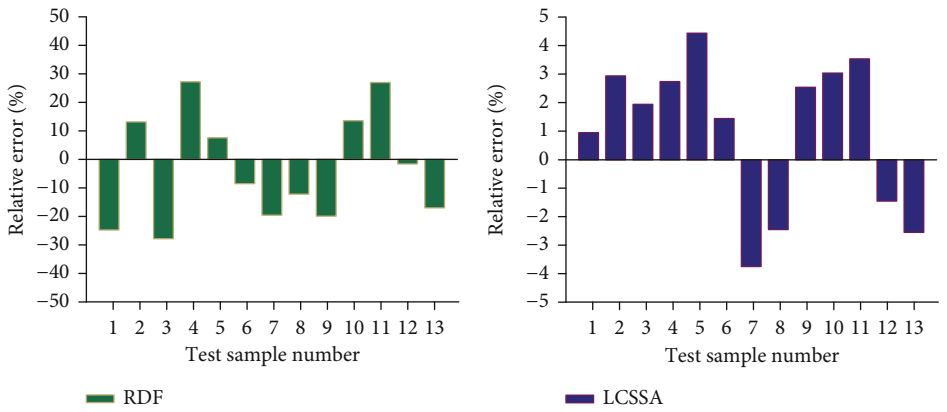


Figure 8: The average relative error of the two models.

5. Conclusion

With the increasing popularity of embedded systems, the research of embedded systems has also received increasing attention, and the security of embedded systems is a new and important aspect. The study proposes an improved Sparrow Search Algorithm (SSA) based on chaos algorithm for SQLite encryption of embedded database, i.e., LCSSA; constructs a suitable cross-compilation environment; implements chaos encryption through software programming in the crosscompilation environment; and performs debugging analysis to optimize the encryption algorithm. The study shows that SQLite is more secure than regular Access in web development. The improved sparrow algorithm based on logistic chaos is then compared with the original sparrow algorithm. Setting the algorithm discovery rate for both groups to 20%, warning rate to 20%, alert value to 0.6, population size to 50, and iteration number to 100, the results show that the chaotic mapping of logistic can improve the quality of the original solution and make it have better global optimisation seeking performance. The linear decreasing weighting method is used to improve the application of LCSSA on population precociousness and global coordination and local mining ability, and the improved LCSSA has fast convergence and high optimisation seeking accuracy compared to the SSA, while still maintaining better stability. The improved algorithm has a larger key space and mapping space. The research gives a recursive algorithm for network weight calculation, which avoids repeated calculation after obtaining new learning data, and saves computing resources and computing time. The wavelet neural network designed according to the algorithm has better network structure and operation efficiency, which has great significance for the real-time application of wavelet neural network.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The author states that there is no conflict of interest.

Authors' Contributions

This research mainly focuses on the encryption technology of SQLite embedded database. Junlong Shi proposed an improved sparrow algorithm (Logistic Chaos Sparrow Search Algorithm, LCSSA) based on Logistic Chaos Map. It shows that the security level of SQLite in Web development is higher than that of conventional Access. Junlong Shi analyzed the results and wrote the manuscript.

Acknowledgments

The research is supported by Major Projects in Anhui Province "Anhui Province Higher Education Blockchain Technology Promotion Sub-Center," Project No. 2020qkl26.

References

[1] X. Miao and Y. Ge, "Energy management for energy harvesting-based embedded systems: a systematic mapping study," *Journal of Electrical and Computer Engineering*, vol. 2020, Article ID 5801850, 19 pages, 2020.

[2] J. Song, Z. G. Zhang, Y. J. Dong et al., "Detection of tuberculosis genes associated with drug-resistance in paraffin-embedded tissue specimens using next generation sequencing technology," *Chinese Journal of Tuberculosis and Respiratory Diseases*, vol. 43, no. 3, pp. 234–241, 2020.

[3] S. Kent and J. Bouvy, "PP 385 using common data models and data networks for evidence generation in health technology assessment," *International Journal of Technology Assessment in Health Care*, vol. 36, no. S1, pp. 32–32, 2020.

[4] Z. Zhang, J. Ren, H. Zhang, Z. Zhang, G. Liu, and S. Yan, "DLRF-Net: a progressive deep latent low-rank fusion network for hierarchical subspace discovery," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 17, no. 1s, pp. 1–24, 2021.

[5] K. Wang and A. Kumar, "Periocular-assisted multi-feature collaboration for dynamic iris recognition," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 866–879, 2021.

[6] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, "IoT-based data logger for weather monitoring using Arduino-based wireless sensor networks with remote graphical application and alerts," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.

[7] M. Yang, W. Wen, X. Wang, L. Shen, and G. Gao, "Adaptive convolution local and global learning for class-level joint representation of facial recognition with a single sample per data subject," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2469–2484, 2020.

[8] H. Rezaei, M. Motovali-Bashi, and S. J. Mowla, "MicroRNA and hemophilia-A disease: bioinformatics prediction and experimental analysis," *Cell Journal*, vol. 23, no. 3, pp. 341–348, 2021.

[9] R. Mama and M. Machkour, "Fuzzy querying with SQL: fuzzy view-based approach," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 5, pp. 9937–9948, 2021.

[10] M. Espinosa, F. Lizárraga, K. Vázquez-Santillán et al., "Coexpression of Smac/DIABLO and estrogen receptor in breast cancer," *Disease Markers*, vol. 30, no. 4, pp. 429–446, 2021.

[11] C. Zhao, L. Jiang, and K. L. Teo, "Bi-objective integrated supply chain design with transportation choices: a multi-objective particle swarm optimization," *Journal of Industrial and Management Optimization*, vol. 13, no. 5, pp. 1–26, 2017.

[12] W. Tuerxun, X. Chang, G. Hongyu, J. Zhijie, and Z. Huajian, "Fault diagnosis of wind turbines based on a support vector machine optimized by the sparrow search algorithm," *IEEE Access*, vol. 9, pp. 69307–69315, 2021.

[13] P. Wang, Y. Zhang, and H. Yang, "Research on economic optimization of microgrid cluster based on chaos sparrow search algorithm," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 5556780, 18 pages, 2021.

[14] T. Liu, Z. Yuan, L. Wu, and B. Badami, "An optimal brain tumor detection by convolutional neural network and enhanced sparrow search algorithm," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 235, no. 4, pp. 459–469, 2021.

[15] J. Yuan, Z. Zhao, Y. Liu et al., "DMPPT control of photovoltaic microgrid based on improved sparrow search algorithm," *IEEE Access*, vol. 9, pp. 16623–16629, 2021.

[16] C. Ouyang, D. Zhu, and Y. Qiu, "Calibration and analysis of mechanical modeling for traction wire rope of mountainous orchard carrier," *Mathematical Problems in Engineering*, vol. 2021, Article ID 7391524, 15 pages, 2021.

[17] B. Alshammari, "Cryptanalysis of a bilateral-diffusion image encryption algorithm based on dynamical compound chaos," *Przeglad Elektrotechniczny*, vol. 1, no. 1, pp. 130–133, 2021.

[18] S. Skaria, V. Jacob, I. Jinchu, and S. K. Savithryamma, "A new chaos based cipher texting technique for secure data transfer," *International Journal of Applied Engineering Research*, vol. 15, no. 5, pp. 524–531, 2020.

[19] H. Xiang and L. Liu, "An improved digital logistic map and its application in image encryption," *Multimedia Tools and Applications*, vol. 79, no. 41-42, pp. 30329–30355, 2020.

[20] M. Gafsi, M. A. Hajjaji, J. Malek, and A. Mtibaa, "Efficient encryption system for numerical image safe transmission," *Journal of Electrical and Computer Engineering*, vol. 2020, Article ID 8937676, 12 pages, 2020.

[21] L. Shi, X. Ding, M. Li, and Y. Liu, "Research on the capability maturity evaluation of intelligent manufacturing based on firefly algorithm, sparrow search algorithm, and BP Neural network," *Complexity*, vol. 2021, Article ID 5554215, pp. 1–26, 2021.

[22] M. Ilba, "Parallel algorithm for improving the performance of spatial queries in SQL: the use cases of SQLite/SpatiaLite and PostgreSQL/PostGIS databases," *Computers & Geosciences*, vol. 155, article 104840, 2021.

[23] T. S. Kondo, "Investigating the efficiency of secret key encryption algorithms with similar key length and block size," *International Journal of Digital Information and Wireless Communications*, vol. 10, no. 2, pp. 29–34, 2020.

[24] A. Altigani, S. Hasan, and B. Barry, "The need for polymorphic encryption algorithms: a review paper," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 3, pp. 360–377, 2020.

[25] S. Gopinathan and M. Suganthi, "A study of image compression and SHA 256 encryption algorithms for secure transmission," *International Journal of Computer Applications*, vol. 181, no. 37, pp. 9–12, 2019.

[26] S. M. Dudhani and S. S. Lomte, "Performance Analysis of Data Encryption Algorithms for Secure EHR Transmission," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 2, pp. 363–366, 2019.