Gsm Encryption Algorithm Emulation Using Neuro-Identifire

Mahmood Khalel Ibrahem Al Ubaidy
College of Information Engineering, Al Nahrain University.
E-Mail: mahmoodkhalel@coie.edu.iq.
mahmood_khalel@yahoo.com.

Abstract

The security model of the Global System for Mobile Communication (GSM) relies on security through obscurity (STO). The later is a controversial principal in security engineering based on the premise that security of an element or function can ensure security of the whole product. STO assumption breaks one of the basic requirements of cryptographic system (Kerckhoff's assumption), which states that the security of the cryptographic system should lies solely on the key, and the algorithm should be assumed as publicly available. In this research, security through obscurity, Kerckhoff's assumption and GSM encryption algorithm (A_5) will be discussed. The target unknown encryption algorithm (A_5) is constructed using a black-box approach with neuro-identifier and a set of input and output data only. The constructed algorithm then is tested and compared with actual unknown algorithm. The experimental results of the research demonstrate nearly complete similarities between both algorithms, which prove that a GSM security model cannot be secure through obscuring algorithms.

Keywords: Security through Obscurity, Neuro-identifier, System Identification, Encryption Algorithm, Emulation.

Introduction

The GSM security algorithm specifications were designed by the GSM Consortium in secrecy and were distributed only on a need-to-know basis to hardware and software manufacturers and to GSM network operators. The specifications were never exposed to the public, thus preventing the open science community around the world from studying the enclosed authentication and enciphering algorithms as well as the whole GSM security model. The GSM consortium relied on Security through Obscurity (STO), i.e. the algorithms would be harder to crack if they were not publicly available [1].

According to the open scientific community, one of the basic requirements for secure cryptographic algorithms is that the security of the crypto system lies solely on the key. This is known as Kerckhoff's' assumption. The algorithm in question should be publicly available, so that the algorithm is exposed to the scrutiny of the public. According to the general opinion no single entity can employ enough experts to compete with the open scientific community in cryptanalysing an algorithm. Thus, the algorithms designed and implemented in secrecy will probably be somehow cryptographically weak and contain design faults. Eventually, the GSM algorithms leaked out and have been studied extensively ever since by the open scientific community. Interesting facts have been discovered since then, during the cryptanalysis of the *A3*, *A5* and *A8* algorithms [1].

GSM faults result from a combination of designing algorithms in secret (security through obscurity) and deliberate weakening of the system. It prevents public security and eventually the algorithm will be exposed anyway [2].

System identification concern with inferring models from observation and studying of systems behavior and properties. System identification deals with the problem of building mathematical models of dynamical systems based on observed data from the system [3].

Artificial neural networks (ANNs) are simplified models of the central nervous system. They are networks of highly interconnected neural computing elements that have the ability to respond to input stimuli. Among the capabilities of ANN, are their ability to learn adaptively from dynamic environments to establish a generalized solution through approximation of the underlying mapping between input and output [4]. Neural networks can be regarded as Black-Box that transforms input vector of m-dimensional space to an output vector in n-dimensional space, which make them ideal tools for Black-Box system identification [5].

In this paper, STO and Kerckhoff's assumption is analyzed. Also we present a theoretical background on system identification and neuro-identifier which is used to identify the unknown GSM cryptographic algorithm and construct an equivalent system that emulates the unknown algorithm.

Security Through Obsecurity

The belief that code secrecy can make a system more secure is commonly known as security through obscurity. Certainly, vendors have the right to use trade secret protection for their products in order to extend ownership beyond the terms afforded under copyright and patent law. But some software systems must satisfy critical requirements under intensive challenges, and thus must be trustworthy. [6].

Security through obscurity is a controversial principle in security engineering based on the premise that secrecy of an element or function can ensure security of the whole. Obscurity is not the only control in effect; instead it is viewed as complimentary to the confidentiality and integrity methodologies built into the cipher system. Security can be complimented by obscurity measures, and as long as it's not employed in complete isolation, it can be considered another powerful tool in the arsenal to provide defense in depth [7]. The obscurity element forces an adversary to begin from a position of distinct disadvantage. Analysis of the system must be made in a “*'black box*” environment with the mechanics of the system slowly evaluated and understood.

Kerckhoffs' Law

Kerckhoffs' law (also called Kerckhoffs' assumption or Kerckhoffs' principle) was stated by Auguste Kerckhoffs in the 19th Century:“**A cryptosystem should be designed to be secure if everything is known about it except the key information**” [8].

It was reformulated (perhaps independently) by Claude Shannon as “the enemy knows the system” [9]. In that form it is called Shannon's Maxim. Since the advent of open source software development, these principles have increasingly been used to ground arguments for it and against security through obscurity. This statement takes the position that cryptosystems should be secure even when an opponent has full working knowledge of the mechanisms employed for confidentiality, so STO by itself is insufficient protection [9].

Military cryptosystems are typically developed in accordance with Shannon's Maxim, but also employ a level of obscurity protection, where the underlying protocols will be maintained as a closely guarded secret; often even withheld from operational users. Kerckhoffs' principle was one of six design principles laid down by Kerckhoffs for military ciphers. Kerckhoffs' original six cipher design principles were [8]:

1. The system must be practically, if not mathematically, indecipherable;
2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;
3. Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents;
4. It must be applicable to telegraphic correspondence;
5. It must be portable, and its usage and function must not require the concourse of several people;
6. Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

Bruce Schneier argued that Kerckhoffs' principle applies beyond codes and ciphers to security systems. In general; every secret creates a potential failure point. Secrecy, in other words, is a prime cause of brittleness, and therefore something likely to make a system prone to catastrophic collapse. Conversely,“openness provides ductility” [10].

What Schneier means is that the things which are kept secret ought to be those which are least costly to change if inadvertently disclosed. A cryptographic algorithm may be implemented by hardware and software which is widely distributed among its users; if security depended on keeping that secret, then disclosure would lead to major logistic headaches in developing, testing and distributing implementations of a new algorithm. Whereas if the secrecy of the algorithm were not important, but only that of the keys used with the algorithm, then disclosure of the keys would require the much less hard process of generating and distributing new keys. In other words, the fewer and simpler the things one needs to keep secret in order to ensure the security of the system, the easier it is to maintain that security [10].

Eric Raymond extended this principle in support of software, saying; “Any security software design that doesn't assume the enemy possesses the source code is already untrustworthy; therefore, never trust closed source” [11]. The controversial idea; that open-source software is inherently more secure than closed-source is promoted by the concept of security through transparency.

Black-Box System Identification

System identification is an important issue in determining a dynamical model for an unknown plant as well as in monitoring and control of system states. Black-Box approach (input-output description); which is used when no information is available about the system except its input and output. Fig.-1 illustrates unknown system with x_n input signals and y_n output signals. The central concept in identification problems is identifiability [3]. The problem is whether the identification procedure will yield a unique value of the parameter (θ), and/or whether the resulting model (M) is equal to the true system. In other meaning, a model structure is globally identified at (θ^*) if:

$$M(\theta) = M(\theta^*) \quad \theta \in D_M \Rightarrow \theta = \theta^* \quad \cdots\cdots\cdots\cdots (1)$$

Where M is a model structure, θ is a parameter vector range over a set of values D_M.

The input-output description of a system gives a mathematical relation between the input and output of the system. In developing this description, the knowledge of the internal structure of a system may be assumed to be unavailable (unknown); the only access to the system is by means of the input and output terminals. Under this assumption, a system may be considered as a “Black Box” as shown in Fig.(1). Clearly what one can do to a black box is to apply all kinds of inputs and measure their corresponding outputs, and then try to abstract key properties of the system from these input-output pairs. An input-output model assumes that the new system output can be predicted by the past inputs and outputs of the system [12].

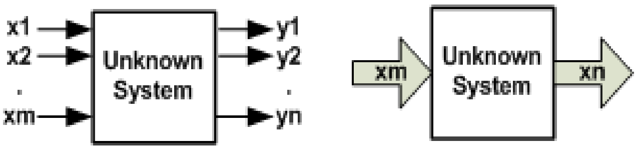

Fig.(1) A Black-Box system with m input and n output [3].

Moreover, black-box model allows finite-dimensional identification techniques to be applied, which is an effective rule in nonlinear system identification. In developing the input-output description, before an input is applied, the system must be assumed to be relaxed or at rest [12], and that the output is excited solely and uniquely by the input applied thereafter and described as follows:

$$y = H\,x \quad \cdots\cdots\cdots\cdots (2)$$

Where H is some function that specifies uniquely the output y in terms of the input x of the system.

For detailed mathematical description of input-output identification, refer to [13]. Fig.(2) illustrates unknown system identification using artificial neural networks as an adaptive system.

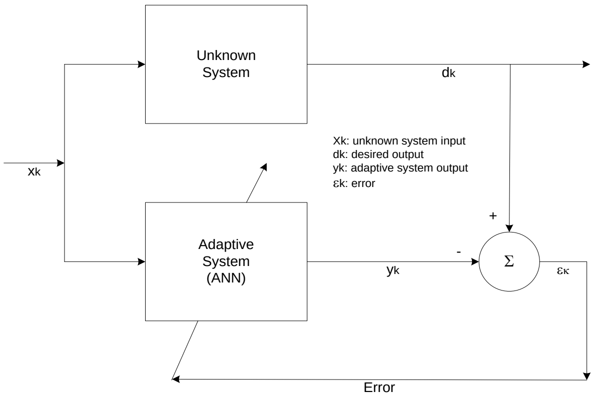

Fig. (2) System Identification using Neuro-identifier.

Neuro-Identifier (Nid)

Neural networks have been widely employed for system identification as they can learn complex mappings from a set of examples. The mapping property, the adaptive nature, and the ability of neural networks to deal with uncertainties make them viable choices for identification and state estimation of nonlinear systems. Neuro-Identifier (NID) are basically a Multi-Layer Feed-Forward artificial neural networks (MLFF) with an input layer (buffer layer), a single or multiple nonlinear hidden layer with biases, and a linear/or nonlinear output layer [14].

Neural network cannot match a nonlinear system exactly; the modeling error depends on the structure of the network. For some nonlinear processes when the operation conditions are changed or its operation environment is complex, one model for these processes is not enough to follow the whole plant; multiple models can give better identification accuracy. Although the single neural network can identify any nonlinear process (black-box), the identification error is high if the network structure is not good. In general we cannot find the optimal network structure, but we can use several possible networks and select the best one by a proper switching algorithm [15]. In this paper we concentrate on the black-box approach because we assume no prior knowledge about the system.

The procedure of identification begins with the choice of neural model which is defined by its architecture and an associated learning algorithm. This choice can be made through a trial and error base. Once the neural model is chosen, and system input-output data are available, learning can begin. Different structures are trained and compared using learning set and simulation set of data, and a criterion (error goal). The optimal structure then, is the one having the fewest units (neurons) for which the criterion is minimum. There is no mathematical formulation to calculate the optimal size of such networks. However, too many free units will learn faster, avoid local minima, and exhibit a better generalization performance [13].

Although backpropagation algorithm (BP) is commonly used in multilayer-Feedforward neural networks training, it failed in complex nonlinear systems identification such as cipher systems. It suffers from local minima, unstable performance surface, and lake of convergence [13, 16]. A modified BP algorithm can be used to update the weights of neural networks in designing stable identification scheme for general nonlinear system with no prior knowledge about their system dynamics [17]. Another promising algorithm Levenberg-Marquardt has been used instead. Levenberg-Marquardt (LM) is a training algorithm based on nonlinear optimization technique by minimizing the sum of squares of error (SSE). LM algorithm is regarded as an intermediate method between the Steepest Descent (SD) and the Gauss-Newoton (GN) methods, it has better convergence properties than the other two methods, and well known that it is the best choice in many off-line training of neural nets. The reason is that, the neural nets minimization problem is often ill-conditioned and LM algorithm disregards nuisance directions in the parameter space which influence the criterion marginally. Levenberg-Marquardt Algorithm (LM) has been used to train the neuro-identifier. LM training algorithm has been proved experimentally to be more effective in multilayer feed forward networks (MLFF) training especially for large degree of accuracy. Moreover it can converge even with less hidden neurons than the optimal number, but with much more epochs [13].

Gsm Encryption Algorithm : A_5

The encryption algorithm used in the GSM system is a stream cipher known as the A_5 algorithm. Multiple versions of the A_5

algorithm exist which implement various levels of encryption [18].

- $A_5/0$ utilizes no encryption.
- $A_5/1$ is the original A_5 algorithm used in Europe and other countries.
- $A_5/2$ is a previous encryption algorithm.
- $A_5/3$ is the latest encryption algorithm created as part of the third Generation Partnership Project (3GPP).

A_5 encryption algorithm scrambles the user's voice and data traffic between the handset and the base station to provide privacy. An A_5 algorithm is implemented in both the handset and the Base Station Subsystem (BSS). Encrypted communication is initiated by encryption mode request command from the GSM network. Upon receipt of this command, the mobile station begins encryption and decryption of data. Each frame in the over-the-air traffic is encrypted with a different key-stream [1].

A_5 consists of three Linear Shift Feedback Registers (LSFR). The three LSFRs are initialized with the session key (K_c) and the frame number (F_n). The 64-bit K_c is first loaded into the register bit by bit. The LSB of the key is XORed into each of the LSFRs. The registers are then all clocked (the majority clocking rule is disabled). All 64 bits of the key are loaded into the registers the same way. The 22-bit frame number is also loaded into the register in the same way except that the majority clocking rule applies from now on. After the registers have been initialized with the K_c and the current frame number, they are clocked one hundred times and the generated keystream bits are discarded. This is done in order to mix the frame number and keying material together. Now 228 bits of keystream output are generated. The first 114 bits are used to encrypt the frame from Mobile Station (MS) to Base Transceiver Station (BTS) and the next 114 bits are used to encrypt the frame from BTS to mobile station. After this, the A_5 algorithm is initialized again with the same K_c and the number of the next frame [1].

Fig.(3) illustrates A_5 encryption and decryption processes. The stream cipher algorithm is initialized with the Session Key (K_c) and the number of each frame (F_n). The same K_c is used throughout the call, but the 22-bit frame number changes during the call, thus generating a unique key stream for every. For detail specifications of A_5, please refer to [18].

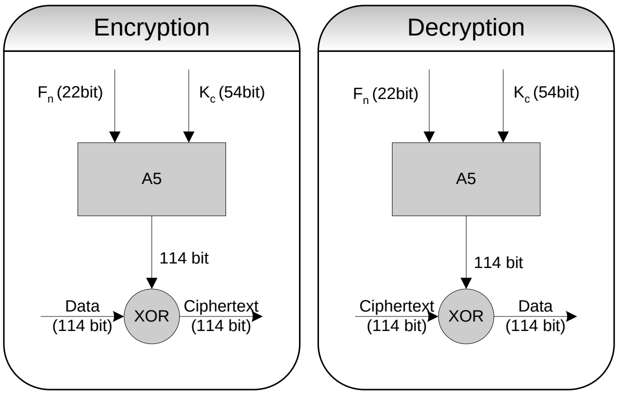

Fig.(3) $A_5/3$ Encryption and Decryption Processes.

A_5 Algorithm Emulation

A_5 emulation using the neuro-identifier based on black-box system identification explained above and performed in three stages.

The first stage is collecting data from the terminals of the unknown system, which means collecting the input data and its corresponding output data. Two sets of data are collected in this stage. The first set is used to train the neuro-identifier and construct an equivalent system to the unknown system. The second set of data is used to test the constructed system.

Training data can be collected by presenting known messages to a known subscriber GSM mobile handset (input) and collecting encrypted messages over the air (output). In this research A_5 algorithm has been simulated using MATLAB to collect the input and output data.

The second stage is the neuro-identifier training. The strategy of black-box approach relies on analyzing and approximating the mapping between the input data and the output data, and hence constructs the internal transformation or mapping function of the targeted unknown system

Before starting training stage, we have to decide on the number of neurons in the input, hidden and output layers of the neuro-identifier. The suggested configuration of the neuro-identifier that has been used in this research was as follows:

- Input layer: 1 neuron (input plain message) + 1 bias neuron (B_1).

- Hidden layer: 2 layers x 16 neurons (Hexadecimal data) + 2 bias neurons (B_2 and B_3).
- Output layer: 1 neuron (output encrypted message).
- Error goal: 10^{-6} (termination condition).
- Training algorithm: Levenberg-Marquardt (LM).

The above configuration has been chosen among different combinations and proven to give optimal solution.

Fig.(4) illustrates the architecture of the neuro-identifier.

In the training stage; input and output data of the training set has been presented to the neuro-identifier and on termination (satisfying the error goal), the weights and biases has been saved for the next stage. The collected weights and biases of the neuro-identifier represent the internal function of the unknown system (A_5 algorithm).

The third stage is the emulation stage which is performed to test the trained neuro-identifier of the second stage (constructed system) using the test data. Input plain message is presented to the constructed system and the output encrypted data is collected and will be called the output of the simulated function. The resulted output will be compared with the collected output from the actual unknown system. They should be identical.

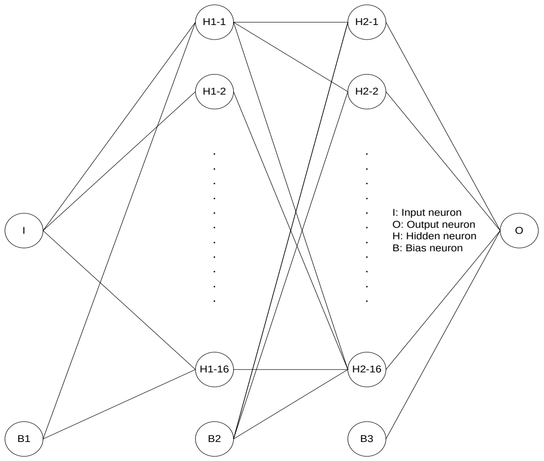

Fig.(4) The Architecture of the Neuro-Identifier.

Experimental Work

Experimental work has been conducted using MATLAB v 7.1 R14 SP3 installed on a PC with 2.2 GHz CPU and 2GB of Ram. Several trials have been conducted to reach the optimal configuration of the neuro-identifier. The selected configuration mentioned in the previous article, took 1.95 x 10^5 seconds (≅ 54.167 hours) in the training stage. The experiment is conducted in three stages;

1. In the first stage, A5 algorithm has been programmed and used to collect two sets of data. The first set is the training data and the second set is the test data to be compared with the output of the constructed system. Both sets are collected by presenting input plain messages and collecting the output encrypted messages of the unknown system. The training set is used to train the neuro-identifier. The second set is used to test the constructed algorithm against the actual unknown system.
2. In the second stage, the proposed neuro-identifier described above has been trained off-line using the training data collected from the first stage to construct the unknown system. When the training reaches the error goal (10^{-6}), the process is terminated and the weights and biases of the neuron-identifier have been saved. Fig.(5) illustrates the training of the neuro-identifier using Levenberg-Marquardt Algorithm (LM). It shows that the error goal has been satisfied in (745) epochs (One cycle through the entire set of training vectors).
3. In the third stage, the second set of data (test data) has been used to run the actual unknown system and the emulated system (weights and biases of the trained neuro-identifier) which are saved from the second stage. The results of the two systems are compared to. Fig.(6) illustrates the behavior of the actual unknown system and the behavior (output) of the constructed system when presented with the same input test data. It shows complete similarities, which means that the constructed algorithm is identical to the unknown algorithm. In another meaning we have constructed an algorithm which is equivalent to the unknown system by collecting its input and output data.
4. It is worth mentioning that the obtained results of the emulated system are not

completely identical with the actual system, because the error goal has been set to 10^{-6} and never reaches zero. Hence the approximated function differs slightly from the actual one. We obtained identical systems by rounding the results of the emulated system to nearest integer value. Theoretically, we can obtain completely equivalent system if we set the error goal to zero and increase the size of the hidden layer, but practically, the time of the training process will increased dramatically.

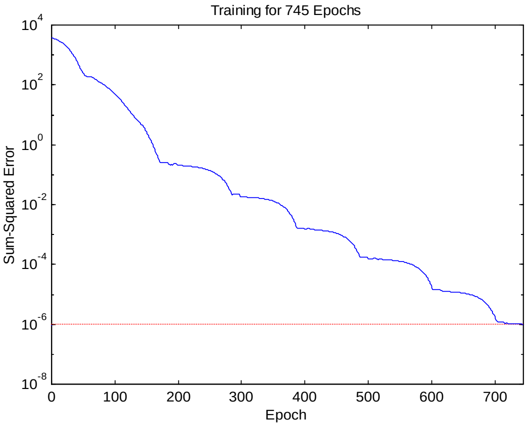

Fig. (5) LM Training Algorithm Learning Curve for A_5 Algorithm with 16x16 Nodes.

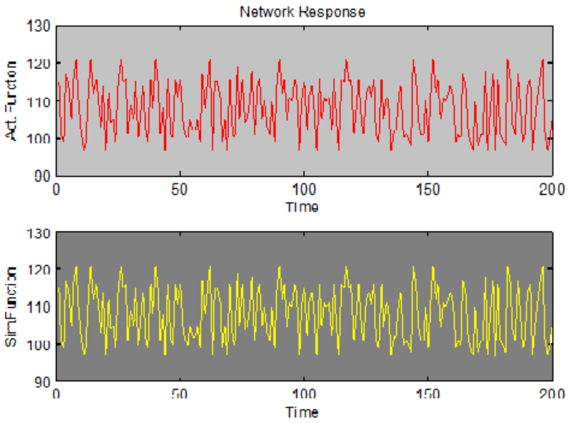

Fig. (6) Network Response for Actual and Simulated Algorithm.

Conclusions

1. Security through obscurity breaks one of the important rules in designing encryption algorithms; that is Kerckhoffs' principle and Shannon Maxim. These principles have increasingly been used to ground arguments against security through obscurity. Eric Raymond extends this principle in support of open source software. The controversial idea that open-source software is inherently more secure than closed source is promoted by the concept of security through transparency.
2. In this research an equivalent system was constructed to emulate the unknown algorithm using neuro-identifier by approximating the transfer function of the unknown system.
3. The results of the experimental work show that the emulated system was nearly identical to the unknown system. More accurate results could be obtained by decreasing the error goal but with more training time. This problem could be solved by decreasing the error goal to a reasonable limit and rounding the results.

References

[1]. Lauri Pesonen, “**GSM Interception**”, GSM Security, Department of Computer Science and Engineering, Helsinki University of Technology http://www.gsm-security.net/gsm-security-papers.shtml, cited: 2/11/2010.

[2] Jeremy Quirke, “**Security in the GSM System**”, AusMobile, © 2004.

[3] Ljung, Lennart, “**System Identification, Theory for the User”**, 2nd Ed. Prentice-Hall, Upper Saddle River, N J, 2002.

[4] Sarle, W. S., “**Artificial Neural Networks and Their Biological Motivation**”, www.csa.ru, 1999, cited 15/12/2010.

[5] Zbikowski R. & Andrzej D., “**Neural Approximation: a Control Perspective**”, Advances in Industrial Control, Springer, 1996.

[6] Rebecca T. Mercuri, Peter G. Neumann, "**Inside Risk, Security by Obscurity**", Communications of the ACM, Volume 46, Number 11, pp 160, 2003.

[7]Ross Patel, "**Security through Obscurity**", BCS, Computing, 20, 2005.

[8] Auguste Kerckhoffs, “**La Cryptographie Militaire**”, Journal des sciences militaires, vol. IX, pp. 5-83, 1883.

[9] Shannon, C. E., “**A Mathematical Theory of Communication**”, Bell System Technical Journal, pp. 623, 1948.

[10]Mann, Charles C., "**Homeland Insecurity**", the Atlantic Monthly 290 (2), 2002.

[11] Raymond, Eric, "**If Cisco Ignored Kerckhoffs's Law, Users Will Pay The Price**", LWN.net.com, cited 5/1/2011.

[12] Chen, Chi-Tsong, "**Linear System Theory and Design**", Oxford University Press, 3rd edition, 2009.

[13] Al-Ubaidy, Mahmood K. "**Black-Box Attack using Neuro-Identifier**", Cryptologia, Vol. 28, No. 4, 2004.

[14] Pham, Duc Truong & Liu Xing, "**Neural Networks for Identification, Prediction and Control**", Springer-Verlag ltd, 1999.

[15] Wen Yu, "**Multiple Recurrent Neural Networks for Stable Adaptive Control**", Neurocomputing Vol. 70, pp. 430–444, 2006.

[16] Jose´ de Jesu´ s Rubioa, Wen Yub, "**Nonlinear System Identification with Recurrent Neural Networks and Dead-Zone Kalman Filter Algorithm**", Neurocomputing Vol 70, 2460–2466, 2007.

[17] Farzaneh Abdollahi, H. Ali Talebi, and Rajnikant V. Patel, "**Stable Identification of Nonlinear Systems Using Neural Networks: Theory and Experiments**", IEEE/ASME Transactions On Mechatronics, Vol. 11, no. 4, 2006.

[18] Joint GSMA TSG SA WG3 Working party, "**Requirements Specification for the GSM A5/3 Encryption Algorithm (Version 0.5)**", 2000.

الخلاصة

تعتمد خوارزمية التشفير في نظم الاتصالات للاجهزة المحمولة على مبدأ سرية الغموض عن طريق حجب الخوارزمية عن المستخدمين، وهذا المبدأ مناقض لفرضية (كيرشوف) في مبادئ نظم التشفير الذي ينص على ان تعتمد سرية التشفير على المفتاح بشكل أساسي وان تكون الخوارزمية معلنة للعموم لاتاحة الفرصة للباحثين في تطوير الخوارزمية وكشف عيوبها. في هذا البحث تم استعراض فرضية (كيرشوف) وخوارزمية نظم الاتصالات المتنقلة (A_5) والمميز العصبي الاصطناعي المستخدم في البحث لاثبات امكانية بناء خوارزمية مماثلة للخوارزمية المجهولة. يفترض البحث ان خوارزمية نظم الاتصالات المتنقلة هي خوارزمية مجهولة لاعتمادها على مبدأ سرية الغموض ولدينا مجموعة من المدخلات والمخرجات فقط تم جمعها من منظومة الاتصالات. تم استخدام المميز العصبي الاصطناعي لبناء خوارزمية مشابهة يمكنها ان تحل محل الخوارزمية المجهولة، وبذلك تم اثبات بان مبدأ سرية الغموض هو مبدأ لايصلح استخدامه في منظومات التشفير.