INTELLIGENT SOFT COMPUTING BASED CRYPTOGRAPHIC TECHNIQUE USING CHAOS SYNCHRONIZATION FOR WIRELESS COMMUNICATION (CSCT)

Arindam Sarkar[1] and J. K. Mandal[2]

Department of Computer Science & Engineering, University of Kalyani, W.B, India

ABSTRACT

In this paper a novel intelligent soft computing based cryptographic technique based on synchronization of two chaotic systems (CSCT) between sender and receiver has been proposed to generate session key using Pecora and Caroll (PC) method. Chaotic system has some unique features like sensitive to initial conditions, topologically mixing; and dense periodic orbits. By nature, the Lorenz system is very sensitive to initial conditions meaning that the error between attacker and receiver is going to grow exponentially if there is a very slight difference between their initial conditions. All these features make chaotic system as good alternatives for session key generation. In the proposed CSCT few parameters (σ, b , r , x_1 ,y_2 and z_2) are being exchanged between sender and receiver. Some of the parameter which takes major roles to form the session key does not get transmitted via public channel, sender keeps these parameters secret. This way of handling parameter passing mechanism prevents any kind of attacks during exchange of parameters like sniffing, spoofing or phishing.

KEYWORDS

chaos synchronization, soft computing, cryptography, Wireless Communication.

1. INTRODUCTION

A variety of session key generation techniques are available to secure data and information from eavesdroppers [1, 2, 6, 10] with some merits and demerits. Most of the key generation algorithm needs large memory and energy [3, 4, 5, 7, 8, 9]. There are few applications where soft computing is used in key generation purpose. In recent days cryptographic protocols are also get deployed in wireless communication. Wireless devices have the problem of memory and energy constraints. In this paper, a novel soft computing based technique has been proposed for energy efficient session key generation in wireless communication to address this problem.

The organization of this paper is as follows. Section 2 of the paper deals with the proposed ACI based key generation technique. Encryption and Decryption Process has been discussed in section 3 and 4 respectively. Example of ACI based key stream generation is discussed in section 5. Results are described in section 6. Conclusions are drawn in section 7 and that of references at end.

2. THE TECHNIQUE

Chaotic system has no closed form solution even though they defined by following simple equation that makes unpredictability in chaos system.

DOI : 10.5121/ijasa.2014.2302

$$\dot{x} = \sigma(x - y)$$

$$\dot{y} = rx - y - xz$$

$$\dot{z} = xy - bz$$

The above differential equations describing the rate of changes of parameter x, y, z respectively.

Coordination of chaos refers to a method where two (or more) chaotic systems (either identical or non identical) regulate a given property of their motion to a similar performance owing to a pairing or to a forcing (periodical or noisy). Chaotic systems inherently resist to harmonization, because two identical systems having to some extent dissimilar preliminary conditions would progress in time in an unsynchronized way (the divergence in the systems' status would cultivate exponentially). A number of researches showed that synchronization of two chaotic systems is possible. There are a lot of synchronization methods in chaotic systems; one of them is Pecora and Caroll (PC) method. The PC method assume a dynamical system characterized by the state space equations

$$\dot{\overline{x}} = f\left(\overline{x}\right)$$

Where $\overline{x} = \left(x_1, x_2, \ldots, x_n\right)$ is the system vector and f is is an arbitrary mapping. Further system is decomposed into two following sub system

$$\left.\begin{array}{l} \dot{\overline{u}} = f\left(\overline{u}, \overline{v}\right) \\ \dot{\overline{v}} = g\left(\overline{u}, \overline{v}\right) \end{array}\right\} driver$$

$$\left.\dot{\overline{w}} = h\left(\overline{u}, \overline{w}\right)\right\} response$$

Driver signal $\overline{u}\left(t\right)$ is drives the response system.

Using Lyapunov exponents of the response system along with consideration that the actions of the driver are negative Chaotic Synchronization can be possible between two systems.

From the Following Equations two secure sub systems i.e. initiator and responder respectively can be defined by applying the PC method on the Lorenz system.

$$\dot{x} = \sigma(x - y)$$

$$\dot{y} = rx - y - xz$$

$$\dot{z} = xy - bz$$

The initiator (x_l, z_l), can be defined by:

$$\dot{x}_1 = \sigma(x_1 - y)$$

$$\dot{z}_1 = x_1 y - bz_1$$

The responder (Y_2, z_2) can be defined by:

$$\dot{y}_2 = rx - y_2 - xz_2 \quad , \dot{z}_2 = xy_2 - bz_2$$

From the above two equations it can be observed that the Lyapunov exponents of the system are both negative. The initiator and responder response subsystems are driven by the signal y(t) and x(t). When t trends to infinity value of $|z_2 - z_1|$ trends to zero. After synchronization of both the system a common value of both the system is obtained.

2.1 ALGORITHM OF CHAOS SYNCHRONIZATION

CHAOSSKG Algorithm

Input : Input parameters σ, b, r
Output: Mutually tuned value of z_1 to form a session key.
Method:

Step 1. Sender initializes the value of σ and b, after that value of b is send to the receiver.

Step 2. Receiver initializes the value of r.

Step 3. Sender generates the point x_1 and z_1

Step 4. Receiver generates the point $\boldsymbol{y}_2$ and $\boldsymbol{z}_2$

Step 5. Sender sends $\boldsymbol{x}_1$ to receiver.

Step 6. Receiver calculates the new value of $\boldsymbol{y}_2$ and $\boldsymbol{z}_2$ with the help of $\boldsymbol{r}$ and $\boldsymbol{b}$ using the following equations and returns the value of $\boldsymbol{y}_2$ and $\boldsymbol{z}_2$ to the sender.

$$\dot{y}_2 = rx - y_2 - xz_2 \quad , \dot{z}_2 = xy_2 - bz_2$$

Step 7. Sender calculates the value of $\boldsymbol{x}_1$ and $\boldsymbol{z}_1$ with the help of $\boldsymbol{y}_2$, $\boldsymbol{\sigma}$ and $\boldsymbol{b}$ using following equations and sends the value of $\boldsymbol{x}_1$ to the receiver and so on.

$$\dot{x}_1 = \sigma(x_1 - y)$$

$$\dot{z}_1 = x_1 y - bz_1$$

Step 8. Sender generates a nonce. This nonce gets encrypted using a symmetric cipher with $\boldsymbol{Z}_1$ as the key and sends the results of the encryption using following equation.

$$En\_Nonce = Encrypt_{Z_1}(Nonce)$$

Step 9. The receiver decrypts $\boldsymbol{En\_Nonce}$ using $\boldsymbol{Z}_2$ as the key, performs a defined function on it using following equation.

$$De\_Nonce = Decrypt_{Z_2}(En\_Nonce)$$

$$Fn\_Nonce = f(De\_Nonce)$$

Step 10. The receiver encrypts the result of the previous step using $\boldsymbol{Z}_2$ as the key and sends the result to the sender.

$$En\_Fn\_Nonce = Encrypt_{Z_2}(Fn\_Nonce)$$

Step 11. The sender decrypts this message using $\boldsymbol{Z}_1$ as the key, performs the inverse of the pre-defined function and checks if the original nonce is obtained as shown in following equation

$$Nonce = f^{-1}\left(Decrypt_{Z_1}\left(En\_Fn\_Nonce\right)\right)$$

Step 12. If synchronization is not achieved, the process is repeated from step 5.

Step 13. If synchronization is achieved i.e. $\boldsymbol{Z}_1 = \boldsymbol{Z}_2$ then $\boldsymbol{Z}_1$ is used as a seed for a pseudo random number generator to generate the secret key between the two systems for a particular session.

In security engineering, a nonce is an arbitrary number used only once in a cryptographic communication. It is often a random or pseudo-random number issued in an authentication protocol to ensure that old communications cannot be reused in replay attacks.

2.2 Complexity Analysis of CHAOSSKG Algorithm

Step 1. Sender initialization of the value of σ and b takes needs $O(1)$ amount of computation.

Step 2. Receiver initialization of the value of r also takes $O(1)$ amount of computation.

Step 3. Generation of the point x_1 and z_1 takes unit amount of computation.

Step 4. Generation of the point y_2 and z_2 takes unit amount of computation.

Step 5. Sending the value of $\boldsymbol{x}_1$ to receiver needs unit amount of computation

Step 6. Receiver calculates the new value of $\boldsymbol{y}_2$ and $\boldsymbol{z}_2$ with the help of $\boldsymbol{r}$ and $\boldsymbol{b}$ and returns the value of $\boldsymbol{y}_2$ and $\boldsymbol{z}_2$ to the sender. This step also takes unit amount of computation.

Step 7. Sender calculates the value of $\boldsymbol{x}_1$ and $\boldsymbol{z}_1$ with the help of $\boldsymbol{y}_2$, $\boldsymbol{\sigma}$ and $\boldsymbol{b}$ and sends the value of $\boldsymbol{x}_1$ to the receiver and so on. This step also takes unit amount of computation.

Step 8. Sender generates a nonce. This nonce gets encrypted using a symmetric cipher with Z_1 as the key and sends the results of the encryption. This step needs $O(nonce\ length)$ amount of computation.

Step 9. The receiver decrypts $\boldsymbol{En\_Nonce}$ using $\boldsymbol{Z}_2$ as the key. It also takes $\boldsymbol{O(nonce\ length)}$ amount of computation.

Step 10. The receiver encrypts the result of the previous step using $\boldsymbol{Z}_2$ as the key and sends the result to the sender. It takes $\boldsymbol{O(message\ length)}$ amount of computation.

Step 11. The sender decrypts this message using $\boldsymbol{Z}_1$ as the key, performs the inverse of the pre-defined function and checks if the original nonce is or not. It takes $\boldsymbol{O(encrypted\ message\ length)}$ amount of computation.

Step 12. If synchronization is not achieved, the process is repeated from step 5. If the loop executed for $\boldsymbol{n}$ times then algorithm needs total $\boldsymbol{O(n)}$ amount of computation.

Step 13. Synchronization is achieved when $\boldsymbol{Z}_1 = \boldsymbol{Z}_2$

From the above complexity analysis it has been seen that CHAOSSKG algorithm need to perform at most total $\boldsymbol{O(n)}$ amount of computation.

2.3 Hopfield Neural Networks (HNN) guided Pseudo Random Number Generation

Chaos synchronized Z_1 values used as a seed of a Hopfield Neural Networks (HNN) guided pseudo random number generation. Hopfield Neural Networks (HNN) possesses function approximation and generalization capabilities that collectively with unsteadiness and non-convergence possessions can be revealed to be beneficial when dealing with the production of random numbers. The learning in a Hopfield network is done by means of a weight adjustment mechanism that directly relates to minimization of an energy function that decreases over time in each iteration and finally stabilizes in some point of the state space representing the problem. The basic idea of Hopfield Neural Networks (HNN) is to memorize some patterns as stable points by associating them with specific inputs to the network. That is, after some that relates to the pattern that is memorized. Stability, therefore, is the most important features of Hopfield neural networks. The ability to converge in Hopfield networks is strongly related to network architecture, network initial condition, and updating rule mode. The convergence of the network occurs when the weight matrix is symmetric. Thus, there might be some alternatives which cause the network not to converge, e.g., by (i) applying an initial asymmetric weight matrix consists of large positive numbers in diagonal, (ii) letting two or more neurons active simultaneously, and (iii) using large network and training it with orthogonal and uncorrelated patterns.

In this paper, a HNN has been used with following conditions to guarantee non-convergence:

- A nonlinear function, $tanh(x)$ where x is summation of all inputs of a neuron, as the activation function for neurons,
- Weights matrix of HNN is asymmetric where the upper triangle of weight matrix contains positive numbers and lower triangle of matrix contains negative numbers, The diagonal of weight matrix contains large positive numbers,
- Uses of large number (100) of neurons.
- In selecting the weights if the output of one neuron is close to or more than 1 then in the next iteration that neuron amplifies itself by the weight of the corresponding branch regardless of other inputs of the neuron. This is also valid for output close to or less than -1 with decreasing impact on the neuron. In other word, this makes the neuron to always fire with ±1 or bigger than ±1 in all iterations and accordingly amplifying itself. To avoid this, we set a condition that the summation of all the weights for inputs to each neuron must be less than 1 and greater than -1.

- The output of each neuron in each iteration is calculated by following equation.

$$X_i^{new} = sgn\left(\sum_{j=1}^{n} W_{j,i} X_j - \theta_i + I_i\right)$$

Where $I, X, W, \theta,$ and n signify input, output, weight, threshold and number of neurons respectively.

- In this HNN θ is zero and in the first iteration I is 1. We also need nonlinear activation function so we use $tanh(x)$ function instead of $sgn(x)$ function. Convergence in a HNN is achieved when the corresponding outputs of all neurons reach a stable state or oscillate between a limited numbers of states. The definition of state stability, however, directly relates to the degree of accuracy of our calculation. For example, if a state is stable with an accuracy of *n* digits after the decimal point, the very same state might not be stable for an accuracy of *m*>*n* digits after decimal point. While digits with higher order of significance have converged, the other digits with lower order of significance have not converged yet, and may converge in the following iterations. Therefore, any stable neuron in a HNN may be viewed as unstable if the accuracy of the calculation is increased. This holds true for both cases of stability where there is (i) on stable point, or (ii) a limited number of stable points.

3. SECURITY ANALYSIS OF CSCT

In this section some of the attacks are considered to check the immunity power of the Proposed CSCT against the attack. In key exchange protocol the major threat is the attacker who resides in the middle of the sender and receiver has access to all the messages exchanged by both synchronizing parties, also he/she knows all about the protocol details. Now, some of the following question to be raised to analyse the immunity capability of the proposed technique.

"Is the attacker able to know the secret information z?"

Or

“Is attacker able to synchronize with the system regarding all the information available?”

Or

"Is the attacker able to guess the key bits by any means of analytical and/or a numerical method?"

To answer these attacks on the system are divided in the following categories:

Attacks by synchronization attempts: In this type of attack, the attacker tries to synchronize with the system by eavesdropping on all the messages exchanged by sender and receiver. This type of attack will not work as the attacker does not know the initial conditions of any of the z components of any of the systems, and also the parameters a and r are hidden too. By nature, the Lorenz system is very sensitive to initial conditions meaning that the error between attacker and receiver is going to grow exponentially if there is a very slight difference between their initial conditions. The main difference between receiver and attacker is that the output of receiver $\boldsymbol{Z}_2$ influences the system A and hence affects its output $\boldsymbol{Z}_1$ resulting in a lack of synchronization between sender and attacker.

Attacks by solving the system differential equations: As the nature of chaotic systems, the problem of solving the system of differential equations representing the system is proven to be very hard. Numerical solution is of no use due to the approximation nature of the numerical methods and the butter fly effect of chaotic systems. Even if the Lorenz system could be solved, other more complex chaotic systems can be used.

4. RESULTS

A total of eight statistical tests of The NIST Test Suite have been performed to evaluate randomness of the key stream. The 8 tests are performed for the proposed CSCT and existing TPM [2] (Tree Parity Machine) scheme and results of these tests get compared and analyzed. The 8 tests are following:

Statistical Test 1: Frequency (Monobits) Test

Table 1 Status for Proportion of Passing and Uniformity of distribution

Technique	Expected Proportion	Observed Proportion	Status for Proportion of passing	P-value of P-values	Status for Uniform/ Non-uniform distribution
TPM [2]	0.972766	0.973333	Success	2.781309e-08	Non-uniform
CSCT	0.972766	0.979437	Success	3.122711e-10	Non-uniform

From the above table it has been observed that proposed CSCT technique passed the Frequency (Monobits) Test with higher proportion value along with non-uniform distribution than existing TPM method. The outcomes of the test confirm the well distribution of proportion of zeroes and ones for the entire sequence.

Statistical Test 2: Runs Test

Table 2 Status for Proportion of Passing and Uniformity of distribution

Technique	Expected Proportion	Observed Proportion	Status for Proportion of passing	P-value of P-values	Status for Uniform/ Non-uniform distribution
TPM [2]	0.972766	0.974275	Success	1.093862e-02	Uniform
CSCT	0.972766	0.971263	Unsuccess	0.831790e-01	Uniform

Proposed CSCT does not able to pass the test. Only TPM method has passed the test. From the above table it has been observed that proposed CSCT has uniform distribution.

Statistical Test 3: Binary Matrix Rank Test

Table 3 Status for Proportion of Passing and Uniformity of distribution

Technique	Expected Proportion	Observed Proportion	Status for Proportion of passing	P-value of P-values	Status for Uniform/ Non-uniform distribution
TPM [2]	0.972766	0.970173	Unsuccess	7.186328e-03	Uniform
CSCT	0.972766	0.992619	Success	7.571843e-02	Uniform

From the above table it has been observed that proposed CSCT technique passed the Binary Matrix Rank Test with higher proportion value than TPM along with uniform distribution. Whereas TPM methods does not able to pass the test.

Statistical Test 4: Non-overlapping (Aperiodic) Template Matching Test

Table 4 Status for Proportion of Passing and Uniformity of distribution

Technique	Expected Proportion	Observed Proportion	Status for Proportion of passing	P-value of P-values	Status for Uniform/ Non-uniform distribution
TPM [2]	0.972766	0.986667	Success	0.000000e+00	Non-uniform
CSCT	0.972766	0.992275	Success	0.000000e+00	Non-uniform

From the above table it has been observed that proposed CSCT technique passed the Non-overlapping (Aperiodic) Template Matching Test with higher proportion value along with non-uniform distribution than existing TPM methods.

Statistical Test 5: Maurer's "Universal Statistical" Test

Table 5 Status for Proportion of Passing and Uniformity of distribution

Technique	Expected Proportion	Observed Proportion	Status for Proportion of passing	P-value of P-values	Status for Uniform/ Non-uniform distribution
TPM [2]	0.972766	1.000000	Success	0.000000e+00	Non-uniform
CSCT	0.972766	1.000000	Success	0.000000e+00	Non-uniform

Like existing TPM technique, CSCT also passed the Maurer's "Universal Statistical" Test along with same observed proportion and non-uniform distribution like others.

Statistical Test 6: Serial Test

Table 6 Status for Proportion of Passing and Uniformity of distribution

Technique	Expected Proportion	Observed Proportion	Status for Proportion of passing	P-value of P-values	Status for Uniform/ Non-uniform distribution
TPM [2]	0.977814	0.478333	Unsuccess	0.000000e+00	Non-uniform
CSCT	0.977814	0.488874	Unsuccess	0.000000e+00	Non-uniform

From the above table it has been seen that CSCT not able to pass the Serial Test. Even existing TPM also not able to pass the technique.

Statistical Test 7: Cumulative Sums Test

Table 7 Status for Proportion of Passing and Uniformity of distribution

Technique	Expected Proportion	Observed Proportion	Status for Proportion of passing	P-value of P-values	Status for Uniform/ Non-uniform distribution
TPM [2]	0.977814	0.953762	Unsuccess	0.000000e+00	Non-uniform
CSCT	0.977814	0.980000	Success	1.915241e-06	Non-uniform

Proposed CSCT passes the Cumulative Sums Test with non-uniform distribution. TPM is not able to pass the test.

Statistical Test 8: Random Excursions Variant Test

Table 8 Status for Proportion of Passing and Uniformity of distribution

Technique	Expected Proportion	Observed Proportion	Status for Proportion of passing	P-value of P-values	Status for Uniform/ Non-uniform distribution
TPM [2]	0.985938	0.972593	Unsuccess	0.000000e+00	Non-uniform
CSCT	0.985938	0.981893	Unsuccess	0.000000e+00	Non-uniform

From the above table it has been seen that like existing TPM and CSCT not able to pass the Random Excursions Variant Test.

5. CONCLUSION

Proposed technique is very simple and easy to implement in various high level language. The test results also show that the performance and security provided by the proposed technique is good and comparable to standard technique. The security provided by the proposed technique is comparable with other techniques. To enhance the security of the technique, proposed technique offers changes of some parameters randomly in each session. To generate the secret session key index mask get exchanged between sender and receiver. This technique has a unique ability to construct the secret key at both sides using this exchanged information. Since the encryption and decryption times are much lower, so processing speed is very high. Proposed method takes minimum amount of resources which is greatly handle the resource constraints criteria of wireless communication. This method generates a large number of keys which is the same number of neurons in the map. For ensuring the randomness in every session, some of the parameters get change randomly at each session. No platform specific optimizations were done in the actual implementation, thus performance should be similar over varied implementation platform. The whole procedure is randomized, thus resulting in a unique process for a unique session, which makes it harder for a cryptanalyst to find a base to start with. This technique is applicable to ensure security in message transmission in any form and in any size in wireless communication. Some of the salient features of proposed technique can be summarized as follows:

a) *Session key generation and exchange – Identical session key can be generate after the tuning of Chaos system in both sender and receiver side. So, no need to transfer the whole session key via vulnerable public channel.*
b) *Degree of security – Proposed technique does not suffers from cipher text only Attack, known plaintext attack, chosen plaintext attack, Chosen cipher text only attack, brute force attack.*
c) *Variable size key –128/192/256 bit session key with high key space can be used in different session. Since the session key is used only once for each transmission, so there is a minimum time stamp which expires automatically at the end of each transmission of information. Thus the cryptanalyst will not be able guess the session key for that particular session.*
d) *Complexity – Proposed technique has the flexibility to adopt the complexity based on infrastructure, resource and energy available for computing in a node or mesh through wireless communication. So, the proposed technique is very much suitable in wireless communication.*
e) *Key sensitivity – Proposed method generates an entirely different cipher stream with a small change in the key and technique totally fails to decrypt the cipher stream with a slightly different secret session key.*
f) *Trade-off between security and performance – The proposed technique may be ideal for trade-off between security and performance of light weight devices having very low processing capabilities or limited computing power in wireless communication.*

In future, some other soft computing based approach can be used to generate the session key.

ACKNOWLEDGEMENT

The author expresses deep sense of gratitude to the DST, Govt. of India, for financial assistance through INSPIRE Fellowship leading for a PhD work under which this work has been carried out.

REFERENCES

[1] Atul Kahate, Cryptography and Network Security, 2003, Tata McGraw-Hill publishing Company Limited, Eighth reprint 2006.

[2] R. Mislovaty, Y. Perchenok, I. Kanter, and W. Kinzel. Secure key-exchange protocol with an absence of injective functions. Phys. Rev. E, 66:066102, 2002.

[3] A. Ruttor, W. Kinzel, R. Naeh, and I. Kanter. Genetic attack on neural cryptography. Phys. Rev. E, 73(3):036121, 2006.

[4] Wolfgang Kinzel and ldo Kanter, "Neural cryptography" proceedings of the 9th international conference on Neural Information processing(ICONIP 03).

[5] Charles Pfleeger, Shari Lawrence Pfleeger, Security in computing, Third Edition 2003, pp 48, Prentice Hall of India Pvt Ltd, New Delhi.

[6] Biham, E. and Seberry, J."Py (Roo): A Fast and Secure Stream Cipher". EUROCRYPT'05 Rump Session, at the Symmetric Key Encryption Workshop (SKEW 2005), 26-27 May 2005.

[7] Chung-Ping Wu, C.C. Jay Kuo, "Design of Integrated Multimedia Compression and Encryption Systems", IEEE Transactions on Multimedia, Volume 7, Issue 5, Oct. 2005 Page(s): 828 – 839.

[8] HongGeun Kim, JungKyu Han and Seongje Cho."An efficient implementation of RC4 cipher for encrypting multimedia files on mobile devices". SAC '07 Proceedings of the ACM symposium on Applied computing, 2007, pp 1171--1175, NewYork, USA.

[9] Mantin and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4", Lecture Notes in Computer Science, Vol. 2259, Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography, pp: 1 - 24, 2007.

[10] Sarkar Arindam, Mandal J. K., "Artificial Neural Network Guided Secured Communication Techniques: A Practical Approach", Paperback: 128 pages, Publisher: LAP LAMBERT Academic Publishing (June 4, 2012), Language: English, ISBN-10: 3659119911, ISBN-13: 978-3659119910.

Authors

Arindam Sarkar

INSPIRE FELLOW (DST, Govt. of India), MCA (VISVA BHARATI, Santiniketan, University First Class First Rank Holder), M.Tech (CSE, K.U, University First Class First Rank Holder).

Jyotsna Kumar Mandal

M. Tech.(Computer Science, University of Calcutta), Ph.D.(Engg., Jadavpur University) in the field of Data Compression and Error Correction Techniques, Professor in Computer Science and Engineering, University of Kalyani, India. Life Member of Computer Society of India since 1992 and life member of cryptology Research Society of India. Dean Faculty of Engineering, Technology & Management, working in the field of Network Security, Steganography, Remote Sensing & GIS Application, Image Processing. 25 years of teaching and research experiences. Eight Scholars awarded Ph.D. and 8 are pursuing.