

Design of a new curve based cipher

Addapalli VN Krishna

Department of Computer Science & Engineering
School of Engineering and Technology(SoET)
CHRIST University
Bengaluru 560074
Karnataka
India

Abstract

This work aims to develop a model with curve based cryptographic scheme which supports Confidentiality and Authentication at low computing resources and equal security.

Subject Classification: 11G20.

Keywords: Complexity, Cubic spline curve, Encryption & decryption, Linear cryptographic model.

1.1.1 Mathematical Model

A cubic Spline curve is generated by considering weights at the points to be connected. The weights being considered on the cubic polynomials are used for interpolating the data. Cubic spline has an advantage of balancing between computational speed and flexibility. They offer better memory utilization than higher order curves and better modeling than lower order curves.

To develop a cubic spline Interpolation model, a one dimensional steady state equation is being used.

By considering one dimensional steady state equation

We get $\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$ Where ‘ α ’ is a constant and Variable T varies with space(x) and time (t) as parameters.

Integrating on both sides

$$\begin{aligned} \int_{x_w}^{x_e} \left(\frac{\partial T}{\partial t} \right) dx &= \alpha \int_{x_w}^{x_e} \left(\frac{\partial^2 T}{\partial x^2} \right) dx \\ &= \alpha \left\{ \left[\frac{\partial T}{\partial x} \right]_{x_e} - \left[\frac{\partial T}{\partial x} \right]_{x_w} \right\} \\ \left(\frac{T_e^{t+1} - T_e^t}{\Delta t} \right) (x_e - x_w) &= \alpha \left\{ \left(\frac{\partial T}{\partial x} \right)_{x_e}^{t+1} - \left(\frac{\partial T}{\partial x} \right)_{x_w}^{t+1} \right\} \end{aligned}$$

E-mail: Adapalli.krishna@christuniversity.in

Where, $\left[\frac{\partial T}{\partial x}\right]_{x_e} = \left(\frac{T_{I+1}^{t+1} - T_I^{t+1}}{\Delta x_e}\right)$, $\left[\frac{\partial T}{\partial x}\right]_{x_w} = \left(\frac{T_I^{t+1} - T_{I-1}^{t+1}}{\Delta x_w}\right)$, $\Delta x_e = \Delta x_w$;
 $x_e - x_w = \Delta x$ are considered.

$$\left(\frac{T_I^{t+1} - T_I^t}{\Delta t}\right) \Delta x = \alpha \left\{ \left(\frac{T_{I+1}^{t+1} - T_I^{t+1}}{\Delta x_e}\right) - \left(\frac{T_I^{t+1} - T_{I-1}^{t+1}}{\Delta x_w}\right) \right\}$$

Since $\Delta x_e = \Delta x_w = \Delta x$

$$\begin{aligned} \left(\frac{T_I^{t+1} - T_I^t}{\Delta t}\right) \Delta x &= \alpha \left\{ \left(\frac{T_{I+1}^{t+1} - T_I^{t+1}}{\Delta x}\right) - \left(\frac{T_I^{t+1} - T_{I-1}^{t+1}}{\Delta x}\right) \right\} \\ \left(\frac{T_I^{t+1} - T_I^t}{\Delta t}\right) \Delta x &= \frac{\alpha}{\Delta x} \{ (T_{I+1}^{t+1} - T_I^{t+1}) - (T_I^{t+1} - T_{I-1}^{t+1}) \} \\ \left(\frac{T_I^{t+1} - T_I^t}{\Delta t}\right) &= \frac{\alpha}{(\Delta x)^2} \{ T_{I+1}^{t+1} - 2T_I^{t+1} + T_{I-1}^{t+1} \} \\ T_I^{t+1} - T_I^t &= \frac{\alpha \Delta t}{(\Delta x)^2} \{ T_{I+1}^{t+1} - 2T_I^{t+1} + T_{I-1}^{t+1} \} \end{aligned}$$

Simplify the above equation

$$\begin{aligned} T_I^{t+1} - T_I^t &= k \{ T_{I+1}^{t+1} - 2T_I^{t+1} + T_{I-1}^{t+1} \} \\ T_I^t &= T_I^{t+1} - k \{ T_{I+1}^{t+1} - 2T_I^{t+1} + T_{I-1}^{t+1} \} \\ &= T_I^{t+1} - k T_{I+1}^{t+1} + k 2T_I^{t+1} - k T_{I-1}^{t+1} \\ &= T_I^{t+1} (1 + 2k) - k T_{I+1}^{t+1} + k T_{I-1}^{t+1} \\ T_I^t &= a T_I^{t+1} + b T_{I+1}^{t+1} + c T_{I-1}^{t+1} = d \end{aligned}$$

Where $k = \frac{\alpha \Delta t}{(\Delta x)^2}$, $a = 1 + 2k$, $c = b = -k$, $d = T_I^t$

The above equations are reduced to Tri Diagonal Matrix algorithm by back Substitution process

$$a_i = a_i - \left(\frac{c_i}{a_{i-1}}\right) a_{i-1}, d_i = d_i - \left(\frac{c_i}{a_{i-1}}\right) d_{i-1}, T_i = (d_i - b_i T_{i+1}) / a_i$$

Considering the model to generate data

Consider a set of 'N' Grid points (say G) of Data values on a curve. Consider it as 'y_i' and considering a case where 1st and last Data values on the curve as known i.e T₁, T_N as known.

For each control volume, for the grid point 2

$$\left(\frac{T_2^{t+1} - T_2^t}{\Delta t}\right) = \frac{\alpha}{(\Delta x)^2} \{ T_3^{t+1} - 2T_2^{t+1} + T_1^{t+1} \}$$

On simplifying it,

$$\begin{aligned} T_2^{t+1} - T_2^t &= \frac{\alpha \Delta t}{(\Delta x)^2} \{ T_3^{t+1} - 2T_2^{t+1} + T_1^{t+1} \} \\ T_2^{t+1} \left(1 + 2 \left(\frac{\alpha \Delta t}{(\Delta x)^2} \right) \right) - T_3^{t+1} \left(\frac{\alpha \Delta t}{(\Delta x)^2} \right) &= T_2^t + T_1^{t+1} \left(\frac{\alpha \Delta t}{(\Delta x)^2} \right) \end{aligned}$$

Introducing $b_2 = -\frac{\alpha \Delta t}{(\Delta x)^2}$, $a_2 = (1 + 2\alpha) \frac{\alpha \Delta t}{(\Delta x)^2}$, $d_2 = y_2 + \left(\frac{\alpha \Delta t}{(\Delta x)^2}\right) T_1$

By considering $T_2^t = y_2$, $T_1^{t+1} = T_1$;

For the points 3 to N-2,

$$\begin{aligned} b_i &= -\left(\frac{\alpha \Delta t}{(\Delta x)^2}\right) \\ c_i &= b_i \\ a_i &= (1 + 2\alpha) \frac{\alpha \Delta t}{(\Delta x)^2} \\ d_i &= y_i \end{aligned}$$

For the N-1 point

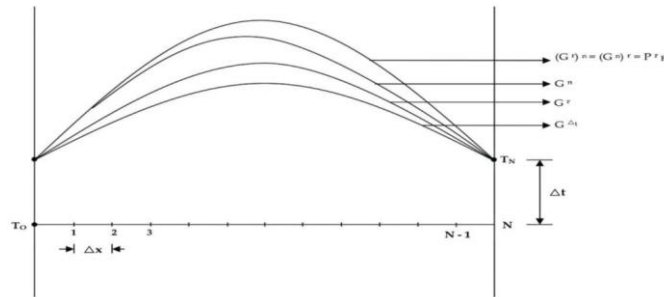
$$\begin{aligned} c_{N-1} &= -\left(\frac{\alpha \Delta t}{(\Delta x)^2}\right) \\ a_{N-1} &= (1 + 2\alpha) \frac{\alpha \Delta t}{(\Delta x)^2} \\ d_{N-1} &= y_{N-1} + \left(\frac{\alpha \Delta t}{(\Delta x)^2}\right) T_N \end{aligned}$$

The process is repeated for 'n' iterations (which is arbitrary), to generate a Cubic Spline curve CU (N).

The developed model may be used for cryptographic study. In cubic spline model, it generates a graph which is a smooth curve. But in cryptographic study, every point must satisfy the field, ie it will be considered as mod P and so a non smooth curve will be generated. For a cryptographic model, different curves will be generated depending on El Gamal model being used. These curves are represented as follows.

Graphical Analysis: Different cases are presented which are associated with functionality of the proposed Model. X- axis: Δx , Y-axis: Δt

$$N = n \cdot \Delta t, \quad r = r \cdot \Delta t$$



Case A: Initially assumed set of 'N' Grid points (say G) of Data values on a curve.

Case B: Cubic curve generated at first round at Δt , ie $G^{\Delta t}$.

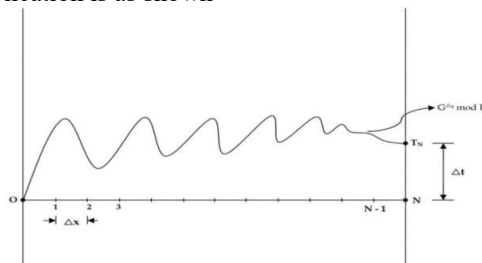
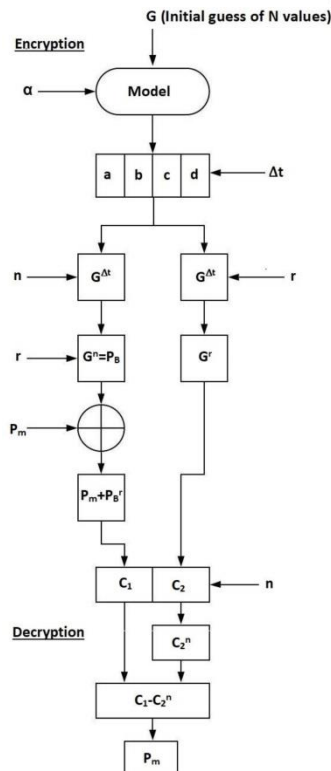
Case C: Cubic curve generated after r (random number) of rounds, ie G^r .

Case D: Cubic curve generated after n (predetermined value) of rounds, ie G^n .

Case E: Cubic curve generated after n rounds with G^r as initial assumption, ie $(G^r)^n$

1.1.2 Cryptographic Study

Several recent studies [1]-[5] says Cryptography requires modular arithmetic. In modular Arithmetic, the points on the curve do not make nice graphs as shown above, but the idea remains the same. We have defined a Cubic Spline curve group for cryptographic operation, but the operation on the coordinates of the point are over the $GF(p)$ field with $p > 3$. We use the same operations with the calculations done in $\text{mod } p$. We call the resulting Cubic Spline curve $CU_p(N)$ where p defines the Modulus and N refers the number of considered points on Curve. Note that the values of points on the curve ranges from 0 to p . A typical case of curve which is not a smooth curve in cryptographic application is as shown



Example

Consider the number of points on curve is, $N=5$.

Assuming values on the points on the curve say

$G = y_1 = y_2 = y_3 = y_4 = y_5 = 5$; $T_1 = 5$; $T_N = 7$. $\alpha = 7$. $P = 17$. $\Delta x = 2$, $\Delta t = 1$,

The values like N , T_1 , T_N , α , Δx , Δt to be randomly chosen and shared among the participants for proper functionality of the model.

The initial values of G are chosen based T_1

Considering the Mathematical model, T_2, T_3, T_4 will be calculated for one time step.

For the point '2',

$$b_2 = -\frac{\alpha \Delta t}{(\Delta x)^2}, \quad a_2 = (1 + 2\alpha) \frac{\alpha \Delta t}{(\Delta x)^2}, \quad d_2 = y_2 + \left(\frac{\alpha \Delta t}{(\Delta x)^2}\right) T_1$$

$$\text{So } b_2 = -\frac{7*1}{(2)^2} \bmod 17 = -7 * 4^{-1} \bmod 17 = 11.$$

$$a_2 = 8; \quad d_2 = (5 + \left(\frac{7*1}{(2)^2}\right) * 5) \bmod 17 = 1.$$

For the points 3

$$b_i = -\left(\frac{\alpha \Delta t}{(\Delta x)^2}\right)$$

$$c_i = b_i$$

$$a_i = (1 + 2\alpha) \frac{\alpha \Delta t}{(\Delta x)^2}$$

$$d_i = y_i$$

$$b_3 = 11, c_3 = 11, a_3 = 8; d_3 = 5$$

For the point 4

$$c_{N-1} = -\left(\frac{\alpha \Delta t}{(\Delta x)^2}\right)$$

$$a_{N-1} = (1 + 2\alpha) \frac{\alpha \Delta t}{(\Delta x)^2}$$

$$d_{N-1} = y_{N-1} + \left(\frac{\alpha \Delta t}{(\Delta x)^2}\right) T_N$$

$$c_4 = 11, a_4 = 8; d_4 = (5 + \left(\frac{7*1}{(2)^2}\right) * 7) \bmod 17 = 13.$$

Thus by using tri diagonal matrix algorithm

Refer to TDMA algorithm

Thus for $i = 3, 4$

$$R_3 = 12; a_3 = 12; d_3 = 10$$

$$R_4 = 8; a_4 = 6; d_4 = 1;$$

$$d_4 = (1 * 6^{-1}) \bmod 17 = 3;$$

$$J = 4 - 3 + 2 = 3$$

$$\text{So } d_3 = (D(3) - B(3) * D(4)) / A(3)$$

$$= 10 - (11 * 3) / 12 = (10 - 33 * 12^{-1}) \bmod 17 = 3$$

$$d_2 = 16.$$

After the first iteration, the points on the curve will be

$$T_1 = 5; T_2 = 16; T_3 = 3; T_4 = 3; T_5 = 7;$$

1.1.3 Encryption & Decryption Models:

Using the Proposed model, Public Key is generated for a identified private key and using ELGAMAL model Encryption and decryption models are generated. The model uses a random number for a specified number of plain text and cipher text pairs to avoid any side channel attacks. The Proposed model is shown by the following flow chart.

Encryption and decryption using Linear Curve based model

Generating Public and private keys

1. Bob Chooses G (Initial guess of points on curve), N (no. of points on the curve), T_1 & T_N (Initial and End points of the curve).
2. Bob & Alice agree for ' α ' as Constant which may be valid for a session or set of sessions.
3. Bob chooses an integer ' n ', which is the chosen private key.
4. Bob calculates G^n using the model to generate public key ' P_B '.
5. Bob announces G , N , ' P_B ', T_1 & T_N as public key parameters. He keeps ' n ' as his private key.

Encryption:

Alice selects Plain text of length $N = 'P_m'$. She considers a random number ' r ', generates P_B^r , G^r

$$\text{Calculate the cipher text } C_1 = (P_m + P_B^r) \quad C_2 = (G^r)$$

Decryption:

Bob after receiving C_1 & C_2 , Calculate the Plain text P_m

$$= C_1 - C_2^n = C_1 - (G^r)^n = C_1 - (G^n)^r = C_1 - (P_B)^r = P_m$$

Enhancing security by the Proposed Model:

In each round, each point on the curve is represented by 4 coefficients which are considered modulo p . Thus to generate a point on curve, the process goes by 4 modulus operations in the forward direction and by back substitution by 4 more operations. For ' n ' rounds, this process will be repeated by n times. Since the process is following a discrete logarithm problem like $G^n \text{ mod } p$ which is a hard problem, provides sufficient strength against crypto analysis.

1.1.4 Computational Complexity of proposed work

The proposed work will be studied along with standard Public key algorithm like ECC and compared for its performance in terms of its complexity

and key length. The complexity of ECC of order n is $O(n^3)$ and the complexity of Proposed work of order m that interpolates n data is $O(m^2 \cdot n)$. Since the Proposed work is based on Asymmetric mode for generating Public Keys and Private keys and Symmetric mode for Encryption and Decryption process, the length of key may be considered between 80 bits to 163 bits to provide sufficient security. By properly simulating the results of the proposed model, the values may be verified.

Conclusion

Studying the proposed work for data Protection and security at low computing resources and Comparing the work with Standard Algorithm like Elliptic Curve Cryptography.

Future work

Non Linear Encryption Model: The Model may also be considered for a non linear case to make it more effective. Thus by making α as a combination of Variables which are non linear in nature, the Model may be tried for its cryptographic purpose. A data base has to be maintained which supports these values while going for encryption and decryption process.

References

- [1] A. Cilardo, L. Coppolino, N. Mazzocca, and L. Romano, "Elliptic curve cryptography engineering," *Proc. IEEE*, vol. 94, no. 2, pp. 395-406 (2006).
- [2] J. Gauthier, Q. V. Wu, and T. A. Gooley, "Cubic splines to model relationships between continuous variables and outcomes: a guide for clinicians," *Bone Marrow Transplantation*, vol. 55, pp. 675-680 (2020).
- [3] K. Toraichi, K. Katagishi, I. Sekita, and R. Mori, Computational complexity of spline interpolation. *International Journal of Systems Science*, vol. 18 Issue 5, pp. 945-954. (1987).
- [4] S. Mittal, A. Bansal, D. Gupta, S. Juneja, H. Turabieh, M. M. Elarabawy, A. Sharma, and Z. K. Bitsue, "Using identity-based cryptography as a foundation for an effective and secure cloud model for e-health," *Computational Intelligence and Neuro Science: Exploration of Human Cognition using Artificial Intelligence in Healthcare*, Special Issue, vol. (2022).
- [5] V. P. Suhas, Numerical Heat Transfer and Fluid Flow, pp. 11-75 (1991).

Received June, 2021

Revised January, 2023