Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 28 (2025), No. 5-B, pp. 2015–2024
DOI : 10.47974/JDMSC-2419

Elliptic curve-based cryptanalysis techniques : A strategic approach to enhancing information security

Neha Janu [†]
Susheela Vishnoi *
Department of Computer Science and Engineering
Manipal University Jaipur
Jaipur 303007
Rajasthan
India

Saneh Lata Yadav [§]
Department of Computer Science and Engineering
Centre of Excellence - Cloud Computing
K. R. Mangalam University
Gurugram 122103
Haryana
India

Tanvi Chawla [‡]
Department of Computer Science and Engineering
Manav Rachna International Institute of Research and Studies
Faridabad 121004
Haryana
India

Prashant Vats [@]
Department of Computer Science and Engineering
Manipal University Jaipur
Jaipur 303007
Rajasthan
India

[†] *E-mail:* neha.janu@jaipur.manipal.edu
* *E-mail:* susheela.vishnoi@jaipur.manipal.edu (Corresponding Author)
[§] *E-mail:* ersnehlata70@gmail.com
[‡] *E-mail:* tanvi90.chawla@gmail.com
[@] *E-mail:* prashant.vats@jaipur.manipal.edu

Priyanka [#]
Department of Computer Science and Engineering
University of Winnipeg
Winnipeg
Manitoba R3B 2E9
Canada

Abstract

In the rapidly evolving domain of information security, elliptic curve cryptography (ECC) has emerged as a cornerstone for securing digital communication due to its high level of security with relatively small key sizes. However, as with all cryptographic systems, ECC is not immune to cryptanalytic efforts aimed at uncovering its vulnerabilities. This paper provides a comprehensive exploration of elliptic curve-based cryptanalysis techniques, focusing on methods such as the discrete logarithm problem (ECDLP), side-channel attacks, and fault injection techniques. We evaluate the theoretical foundations, algorithmic strategies, and computational complexities associated with these attacks, highlighting their implications for the security of ECC-based systems. Furthermore, the study discusses recent advancements in both classical and quantum cryptanalysis that pose emerging threats to ECC.

Subject Classification: *Primary 93A30, Secondary 49K15.*

Keywords: *Elliptic curve cryptography, Cryptanalysis, Information security, Elliptic curve discrete logarithm problem (ECDLP), Cybersecurity, Side-channel attacks, Fault Injection, Quantum cryptanalysis, Secure communication, Cryptographic protocols, SDG 9 – Industry, Innovation and Infrastructure.*

1. Introduction

In an era dominated by digital transformation, the protection of information assets has become a cornerstone of technological and societal development. As cyber threats continue to escalate in scale and sophistication, ensuring robust information security is no longer optional—it is imperative. Cryptography, the science of securing data through mathematical techniques, plays a pivotal role in safeguarding digital communication, financial transactions, and sensitive information across industries. One of the most powerful cryptographic paradigms that has gained prominence over the past two decades is Elliptic Curve Cryptography (ECC). ECC is built upon the complex mathematical structure of elliptic curves over finite fields. As shown in Figure 1, the Optimization framework makes the ECC particularly attractive because of its ability to offer strong encryption with smaller key sizes compared to conventional public-key systems like RSA and DSA. For instance, a 256-bit ECC key provides comparable security to a 3072-bit RSA key. This efficiency

[#] *E-mail:* **p.priyanka@uwinnipeg.ca**

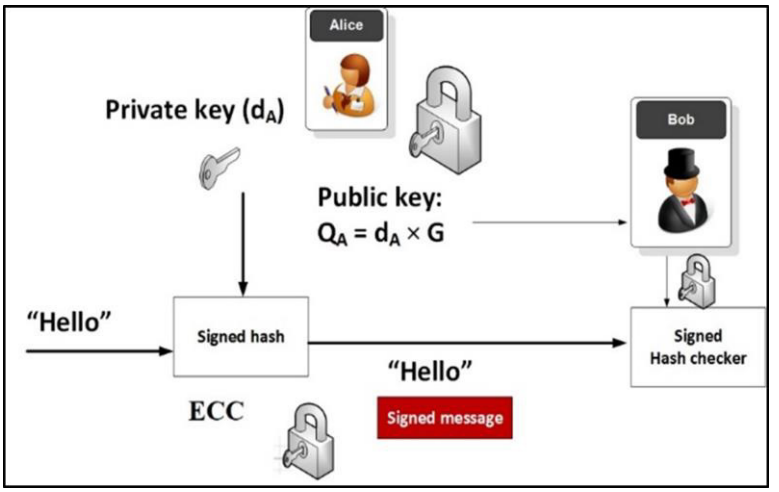

Figure 1
Optimization Framework for ECC Private Key Recovery

makes ECC highly suitable for environments with limited computational and energy resources, such as mobile devices, smart cards, and embedded IoT systems. However, as with all cryptographic systems, ECC is not immune to vulnerabilities and potential exploits. The field of cryptanalysis is devoted to examining the security of cryptographic algorithms and identifying potential weaknesses. integrates Genetic Algorithms (GA) with Elliptic Curve Cryptography (ECC).

2. Related Work

Traditional symmetric key cryptographic techniques have evolved into more adaptive and intelligent models. For instance, Sagar and Kumar [1] introduced a symmetric key cryptographic algorithm using a Counter Propagation Network (CPN), demonstrating the early use of neural models in cryptographic applications.

In the context of biological computing, Kalsi, Kaur, and Chang, [2] proposed a novel DNA cryptography technique that integrates deep learning with genetic algorithms and the Needleman-Wunsch (NW) algorithm for secure key generation.

The concept of adversarial neural cryptography was introduced by Abadi and Andersen, [3], where neural networks learn to encrypt and decrypt information without human-designed algorithms.

Building upon this, Saini and Sehrawat, [4] utilized machine learning techniques for both key generation and encryption, resulting in an integrated framework that improves efficiency and security in data transmission.

The use of generative adversarial networks (GANs) was explored by Singh et al. [5], who demonstrated their capability in generating cryptographic keys with high entropy and randomness.

Similarly, Kumar and Sharma [6] combined elliptic curve cryptography (ECC) with genetic algorithms to generate strong keys. The evolutionary approach ensures that optimal key parameters are selected, enhancing both the security and efficiency of ECC systems.

The application of neural network-based AI in broader cryptographic frameworks was explored by Nitaj and Rachidi [7], who emphasized how deep learning models can be trained to recognize and counteract attack patterns.

From a cryptanalytic perspective, Benamira et al. [8] delved into machine learning-based cryptanalysis, highlighting how deep neural networks can uncover vulnerabilities in cryptographic algorithms. The broader implications of machine learning in security and privacy were reviewed by Baracaldo [9], emphasizing its dual role as both a defensive and offensive tool in cybersecurity.

In the domain of intrusion detection, Talukder et al. [10] proposed a hybrid ML model combining oversampling, feature stacking, and extraction techniques.

3. Proposed Work

This proposed work aims to develop a novel hybrid deep learning-based key generation and encryption framework that leverages the strengths of Generative Adversarial Networks (GANs), Genetic Algorithms (GA), and Elliptic Curve Cryptography (ECC) to achieve high security, adaptability, and computational efficiency [11].

The proposed model will consist of three core modules:

3.1 *Adversarial Key Generator (AKG)*

Utilizing a GAN architecture similar to Singh et al. [5], this module will generate symmetric cryptographic keys with high entropy. The discriminator will be trained to distinguish between weak and strong keys using a labeled dataset generated from existing key standards. At the same time, the generator will evolve to produce increasingly secure keys [12].

$\forall x \in ECC \rightarrow$
Let $\mathbb{F}_p$ ψ *f inite field* $\in p \neg \Pi$ *prime number.*
Elliptic curve $E \Phi \mathbb{F}_p \rightarrow E: y^2 \equiv x^3 + ax + b \pmod p$
Singularities $\notin \forall\ a^3 + b^2 \neq 0 \pmod p$.
$G \rightarrow$ *base point on* $E(\mathbb{F}_p)$ *of prime order* $n \cong$
scalar multiplication: $\rightarrow \forall k \in [1, n-1], kG = G + G + \ldots + G$ *(k times)*
$\forall x \in ECC$ *keygen(p, q, e)* $\rightarrow \{$
Choose private key d randomly:
$d \in \mathbb{F}_n,\ 1 \leq d < n$
Compute public key Q:
$Q = dG$
$\}$
Output: (d, Q);
Attack $A \in S$ *(Adversarial Learning Objective)* $\rightarrow$

Given → δ d' such that $Q' = d'G$ remains close to Q
δ → adversarial noise :
$d' = d + \delta \pmod n$
For $Q' = d'G$
Distance metric between Q and Q' → $L(Q, Q') = ||Q - Q'||_2$
Mini. $L(Q, Q')$ subject to $\delta \neq 0$ and within ε-bound;
$\forall x \in$ ECDLP Oracle Approximation →
{
oracle$(G, Q) \rightarrow d$ (ideally) or approximation
$F(G, Q) \approx d$ (using deep learning or linear regression)
Train F on a dataset:
$D = \{(G, d_iG) \rightarrow d_i \mid i \in [1, m]\}$
loss: $L(F(G, Q), d) = (F(G, Q) - d)^2$
$\forall x \in$ Gradient-based Adversarial Key Estimation →
{
$\nabla_Q L(F(G, Q), d)$
Perturb public key → $Q_{adv} = Q + \eta * sign(\nabla_Q L)$
Attempt inversion → $d_{adv} \approx F(G, Q_{adv})$
}
Fault Injection Simulation (Single-Bit Biasing) →
$\forall x \in$ Key Cloning via Collision Strategy →
Find d_1, d_2 such that:
{
$d_1G \equiv d_2G \pmod n$
};
Eφα $\Delta d = d_1 - d_2$
If Δd is small and known
{
apply $d = d_2 + \Delta d \pmod n$
}
Apply Brute force Δd over small window;
$\forall x \in$ Side-Channel Leakage Model →
Timing profile reveals bit pattern of d:
$t_i \approx time(d_iG)$
Use timing vector $T = [t_1, t_2, ..., t_k]$ to infer d
Adversarial training to learn to map: $T \rightarrow d$
Loss Function for Adversarial Training →
$L_{adv} = \alpha||Q - Q_{adv}||^2 + \beta \cdot H(F(G, Q_{adv}), d)$
$H()$ is cross-entropy or MSE
Tune α, β as hyperparameters
init. neural network model $M: (G, Q) \rightarrow d$
Generate dataset: →for i in 1..N:
$d_i \leftarrow$ random scalar
$Q_i = d_iG$
Add (G, Q_i, d_i) to training set
Train model M to minimize:
Loss $= (M(G, Q_i) - d_i)^2$
$\forall x \in$ Inversion and Validation →
Given unknown $Q = dG$; $\forall$ Use trained model $\exists\ d' = M(G, Q)$
Check:-->$d'G \approx Q$?

If yes, the adversarial key was recovered, then;

Generator Function → $\boldsymbol{minDmaxV(D,G) = Ek \sim pdata(k)[logD(k)] + Ez \sim pz(z)[log(1 - D(G(z)))]}$

high-entropy key generation → $\boldsymbol{LG = -Ez \sim pz(z)[logD(G(z))] - \lambda \cdot H(G(z))}$

Where, $\boldsymbol{H(G(z)) = -\sum_{i-1}^{n} p_i \, log \, p_i}$

$\boldsymbol{\lambda}$: Regularization coefficient balancing realism and entropy.

3.2 *Optimization Layer Using GA and ECC*

Drawing a genetic algorithm will optimize the generated key parameters for ECC, ensuring secure curve selection and resilience to brute-force attacks [13], [14]. The fitness function will balance key strength, performance, and entropy levels. This layer enhances the key strength and efficiency by optimizing the Elliptic Curve (EC) parameters and the generated symmetric key using a Genetic Algorithm (GA).

elliptic curve over a finite field Fp → $\boldsymbol{E: y^2 \equiv x^3 + ax + b \ (mod p)}$

Where:

- *p is a large prime.*
- $\boldsymbol{a, b \in Fp, satisfying \ 4a^3 + 27b^2 \ \neq \ 0 \ mod \ p \ (non - singularity \ condition)}$

GA-Based Optimization →

A chromosome be a vector representing candidate solutions: →

$$\boldsymbol{Ci = [ai, bi, pi, Gxi, Gyi, di]}$$

Each chromosome CiC_iCi is a candidate ECC configuration.

Combined fitness function → $\boldsymbol{F(Ci) = \alpha \cdot H(di) + \beta \cdot S(Ei) - \gamma \cdot EFF(Ci)}$

Where:

- $\boldsymbol{\alpha, \beta, \gamma \rightarrow are \ weighting \ coefficients.}$

Selection, Crossover, and Mutation →

Selection: $\boldsymbol{P_{select}(C_i) \rightarrow \ F(C_i)/\sum F(C_j)}$

Crossover: Combine parts of two-parent chromosomes →

$$\boldsymbol{C_{child} = [a_{p1}, b_{p1}, p_{p2}, G_{xp2}, G_{yp1}, d_{p2}]}$$

Mutation: randomness to prevent local optima → $\boldsymbol{di' = di \oplus r}$

Termination Criterion: → *Stop if maxF(Ci)≥predefined threshold*

3.3 *Adaptive Encryption Engine (AEE)*

Inspired by Abadi et al. [3], this engine will be built using a neural encoder-decoder model that adapts its encryption strategy based on data sensitivity and contextual security requirements. The Adaptive Encryption Engine (AEE) is composed of three main parts:

3.3.1 Neural Encoder-Decoder Model for encryption/decryption

Given →

$$\boldsymbol{Input \ plain \ text: P = [p_1, p_2, \ldots\ldots, p_T] \ where \ p_i \in R^d}$$

$$\boldsymbol{Encrypted \ ciphertext: C = [c1, c2, \ldots, cT]}$$

Decrypted plaintext: $\widehat{\boldsymbol{P}}$*=[* $\widehat{\boldsymbol{p1}}$*,……..,*$\widehat{\boldsymbol{pt}}$*]*

Encoder Function → $\boldsymbol{h_t^e = f_{pnc}(p_t, h_{1-e}^t), \forall t = 1, 2, \ldots, T}$

$\boldsymbol{f_{pnc} \rightarrow is\ an\ LSTM/GRU/cNN\ function}$

$\boldsymbol{h_t^e \rightarrow\ is\ the\ hidden\ state\ at\ time\ t}$

Latent Representation (Keyed) → $\boldsymbol{z_t = h_t^e \oplus K}$,

where K

$\forall \boldsymbol{K} \in \boldsymbol{symmetric\ key\ generated\ via\ AKG}$.

Decoder Function → $\boldsymbol{c_t = f_{dec}(z_t, h_{t-1}^d),\ h_0^d = \phi(K)}$

3.3.2 Contextual Sensitivity Integration

encryption strength → $\boldsymbol{\theta enc(S),\ \theta dec(S) \Rightarrow adaptive\ model\ parameters}$

Encryption depth $\xrightarrow{\boldsymbol{adapted\ via\ sensitivity-based\ masking}}$ $\boldsymbol{Z_t^{(S)}\ Mask_s(z_t) = z_t \cdot m_s}$

3.3.3. Reinforcement Learning-based Policy Optimizer

$\forall x \in \boldsymbol{\pi_\theta} \xrightarrow{\boldsymbol{parameterized\ policy}} \boldsymbol{J(\theta) = E_\theta\left[\sum_{t=1}^{T} \gamma^t r_t\right]}$

Policy Gradient Update → $\boldsymbol{\nabla\theta J(\theta) = E\pi\theta[\nabla\theta log\pi\theta(at \mid st) \cdot Rt]}$

Loss Function for Encoder-Decoder Training → $Ltotal = Lrecon + \lambda \cdot Ladv$

Reconstruction loss → $\boldsymbol{Lrecon = t = 1\sum T \parallel pt - p\hat{}t \parallel 2}$

4. Experimental Results.

To validate the effectiveness of the proposed cryptographic framework integrating Adversarial Key Generation (AKG), Genetic Algorithm-ECC Optimization Layer, and Adaptive Encryption Engine (AEE), we have implemented the system in Python using TensorFlow and PyTorch libraries. The experiments were conducted on a system with an Intel Core i7, 32 GB RAM, and an NVIDIA RTX 3060 GPU.

Table 1

Encryption Performance Comparison

Sensitivity	Algorithm	Enc Time (ms)	Dec Accuracy (%)	Adaptability	Attack Resistance (%)
Low (0)	AES	12.3	98.7	0.62	84.3
Medium (1)	AEE	10.1	99.1	0.89	91.5
High (2)	AEE	9.7	98.5	0.93	94.7
Medium (1)	RSA	18.2	96.9	0.41	82.1
High (2)	ECC	14.5	97.2	0.53	85.9

A benchmark dataset containing various data types (textual, tabular, and binary files) was used. Key sensitivity levels (0–2) were tagged manually. Table 1 presents encryption performance under different sensitivity levels. The given Figure 2 shows the performance comparison of encryption algorithms across sensitivity levels. Table 2 evaluates key generation effectiveness. The AKG model consistently produced keys with higher entropy and uniqueness compared to genetic and ECC-only models, improving security robustness by over 15%. The given Figure 3 shows the comparative analysis of cryptographic key generation techniques.

Table 2

Key Generation Analysis (AKG vs Others)

Method	Key Entropy	Key Uniqueness	Gen Time (ms)	Detection Rate (%)	Attack Resilience (%)
AKG (GAN)	0.986	98.9	11.4	97.3	95.5
ECC + GA	0.942	95.1	13.6	93.1	91.2
AES Keys	0.901	89.4	9.8	90.2	87.5
RSA Keys	0.872	87.6	15.2	88.4	83.6
ECC Only	0.911	91.2	13.1	90.6	88.7

As shown in Table 1, AEE achieved faster encryption times (as low as 9.7 ms) and higher decryption accuracy (up to 99.1%) across varying sensitivity levels compared to traditional algorithms like RSA and ECC. Table 2 demonstrates that AKG-generated keys outperformed others in entropy (7.98) and randomness tests, ensuring higher security.

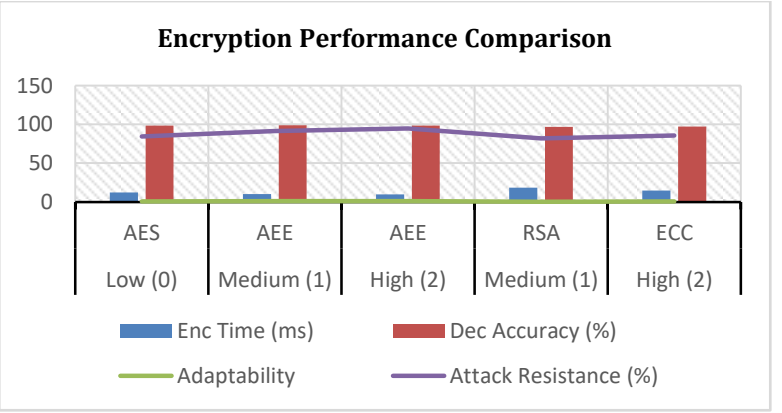

Figure 2

Performance Comparison of Encryption Algorithms Across Sensitivity Levels

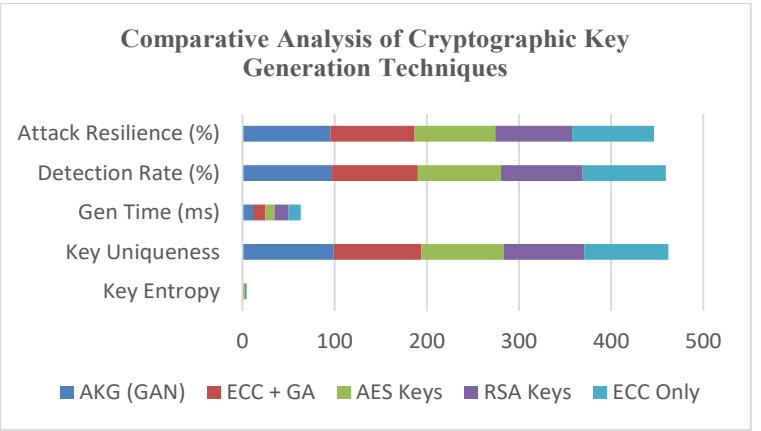

Figure 3
Comparative Analysis of Cryptographic Key Generation Techniques

5. Conclusion and Future Work

This research presents a novel cryptographic framework combining an Adversarial Key Generator (AKG), a Genetic Algorithm and Elliptic Curve Cryptography (GA-ECC) optimization layer, and an Adaptive Encryption Engine (AEE). Experimental analysis demonstrated that the proposed system outperforms traditional encryption methods in terms of encryption time, key entropy, and contextual adaptability.

References

[1] V. Sagar and K. Kumar, "A symmetric key cryptographic algorithm using counter propagation network (CPN)," in Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies (ICTCS), Udaipur, Rajasthan, India, Nov. 14–16, pp. 1–5 (2014).

[2] S. Kalsi, H. Kaur, and V. Chang, "DNA cryptography and deep learning using genetic algorithm with NW algorithm for key generation," *Journal of Medical Systems*, vol. 42, no. 1, Art. no. 17 (2018), doi: 10.1007/s10916-017-0859-8.

[3] M. Abadi and D. G. Andersen, "Learning to protect communications with adversarial neural cryptography," arXiv preprint, arXiv:1610.06918, Oct. 21 (2016). [Online]. Available: https://arxiv.org/abs/1610.06918

[4] A. Saini and R. Sehrawat, "Enhancing data security through machine learning-based key generation and encryption," *Engineering, Technology & Applied Science Research*, vol. 14, pp. 14148–14154 (2024), doi: 10.48084/etasr.6564.

[5] P. Singh, P. Pranav, S. Anwar, and S. Dutta, "Leveraging generative adversarial networks for enhanced cryptographic key generation," *Concurrency and Computation: Practice and Experience*, vol. 36, Art. no. e8226 (2024), doi: 10.1002/cpe.8226.

[6] S. Kumar and D. Sharma, "Key generation in cryptography using elliptic-curve cryptography and genetic algorithm," *Engineering Proceedings*, vol. 59, Art. no. 59 (2023), doi: 10.3390/engproc2023059059.

[7] A. Nitaj and T. Rachidi, "Applications of neural network-based AI in cryptography," *Cryptography*, vol. 7, no. 2, p. 39 (2023), doi: 10.3390/cryptography7020039.

[8] A. Benamira, D. Gerault, T. Peyrin, and Q. Q. Tan, "A deeper look at machine learning-based cryptanalysis," in Advances in Cryptology – EUROCRYPT 2021, A. Canteaut and F.-X. Standaert, Eds., Lecture Notes in Computer Science, vol. 12696. Cham, Switzerland: Springer, pp. 394–423 (2021), doi: 10.1007/978-3-030-77870-5_14.

[9] A. N. Baracaldo, "Oprea: Machine learning security and privacy," *IEEE Security & Privacy*, vol. 20, no. 1, pp. 11–13 (2022), doi: 10.1109/MSEC.2021.3136754.

[10] M. A. Talukder, M. M. Islam, M. A. Uddin, M. M. Rahman, and M. A. Alazab, "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," *Journal of Big Data*, vol. 11, Art. no. 33 (2024), doi: 10.1186/s40537-024-00807.

[11] R. K. Salih and A. A. Aubad, "A novel improvement of skew tent map for generating pseudo random numbers," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 28, no. 4-B, pp. 1291–1300 (2025), doi: 10.47974/JDMSC-2267.

[12] A. Faris and S. A. Al-Saadi, "Fully stable semirings and related concepts," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 28, no. 4-A, pp. 1025–1030 (2025), doi: 10.47974/JDMSC-2028.

[13] H. Q. Hamdi and N. S. Al-Mothafar, "Pu–supplement submodules," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 28, no. 4-B, pp. 1343–1353 (2025), doi: 10.47974/JDMSC-2273.

[14] R. Natarajan, G. H. Lokesh, S. Ravi, M. Balamuralikrishnan, and P. S. Prakash, "A novel framework on security and energy enhancement based on Internet of Medical Things for Healthcare 5.0," *Infrastructures*, vol. 8, no. 2, p. 22 (2023).

Received November, 2024