Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 10-B, pp. 4261–4272
DOI : 10.47974/JDMSC-3049

Discrete logarithm problem in hyperelliptic curve groups

Shweta S. Aladakatti *
Department of Computer Science & Engineering
Manipal Institute of Technology Bengaluru
Manipal Academy of Higher Education (Deemed to be University)
Bengaluru 560064
Karnataka
India

Vikram V. Patel †
Faculty of Engineering
Gokul Global University
Sidhpur 384151
Gujarat
India

Rahul Rajendra Papalkar §
Department of Computer Engineering
Vishwakarma University
Pune 411048
Maharashtra
India

Harshada Bhushan Magar ‡
Department of Electronics & Telecommunication
AISSMS Institute of Information Technology
Kennedy Road
Shivaji Nagar
Pune 411001
Maharashtra
India

* *ORCID-ID:* 0000-0002-7555-4353
† *ORCID-ID:* 0009-0006-4834-6766

* *E-mail:* shweta.aladakatti@manipal.edu (Corresponding Author)
† *E-mail:* vikrampatel@gokuluniversity.ac.in
§ *E-mail:* rahul.papalkar@vupune.ac.in
‡ *E-mail:* harshada.magar@aissmsioit.org

© Taru Publications

Veerendra Yadav [@]
Department of Computer Science & Engineering
Noida International University
Greater Noida 203201
Uttar Pradesh
India

Syed Farhan [#]
Centre for Multidisciplinary Research
Anurag University
Hyderabad 500088
Telangana
India

Abstract

The Discrete Logarithm Problem in Hyperelliptic Curve Groups (HECDLP) is a fundamental basis to more advanced cryptographic systems, and is more secure per key size than classical techniques using elliptic curves. This paper discusses the computational complexity and security consequences of HECDLP by considering the mathematical structure of it on finite field and divisor class groups. The aim is the evaluation of efficient algorithms, such as index calculus and rho versions of Pollard, and the proposal of a hybrid cryptographic model, taking advantage of the properties of hyperelliptic curves. The algorithm combines algebraic geometry, group theory and algorithmic optimization approaches to measure performance using security and efficiency measures. The preliminary performance comparison reports a security-strength score of 92.6% and a computation time of 118 ms for the proposed HECC framework. The reported computation time is 24.36% lower than that of the ECC-based method and 58.45% lower than that of the classical DLP method. These findings require validation through a reproducible experimental protocol and a formally defined security evaluation metric. The suggested model improves the immunity of classical and quantum-inspired attacks, making HECC an effective and scalable cryptography alternative with high security levels to be used in the next-generation applications.

Subject Classification: *62G10, 68T05.*

Keywords: *Hyperelliptic curves, Discrete logarithm problem, HECC, Index calculus, Cryptographic security, Finite fields.*

I. Introduction

The Discrete Logarithm Problem in Hyperelliptic Curve Groups (HECDLP), has become an important field of study in the study of current cryptography with the possibility of offering a high level of security using smaller key sizes [1, 2]. With the growing need of digital communication systems to have effective and secure encryption systems, the use of integer

[@] *E-mail:* **veerendra.yadav@niu.edu.in**
[#] *E-mail:* **SyedFarhan2024@yahoo.com**

factorization and classical discrete logarithm problems is under mounting pressure due to sophisticated methods of computation [3, 4]. Hyperelliptic Curve Cryptography (HECC) builds upon the ideas of elliptic curve cryptography, but it uses higher-genus curves, which can be represented with smaller sizes and be designed with greater customization [5, 6, 7]. The HECDLP is specified on the group of divisor classes of a hyperelliptic curve, with the hardness of finding discrete logarithms making it cryptographically secure [8]. It has a mathematical implementation based upon finite fields, algebraic curves, and group operations, which are lightweight and high security [9].

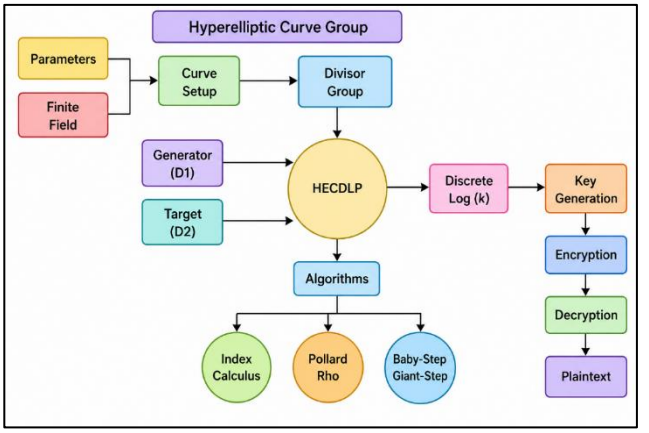


Figure 1

Hyperelliptic Curve Discrete Logarithm Problem (HECDLP) Based Cryptographic Framework

Figure 1 demonstrates HECDLP framework that allows efficient cryptographic operations to be conducted securely. Although it has its benefits, the efficient solution of HECDLP is a complicated process, and algorithms like index calculus are opportunities and threats. Therefore, it is important to know its computational complexity as well as its applications in real-life to design future cryptography systems that are resistant to the changing attack patterns [10, 11].

II. Discrete Logarithm Problem in Hyperelliptic Curve Groups

HECDLP is an algorithm that identifies an unknown scalar between two related elements in divisor class groups, and is strongly cryptographically hard because it is computationally hard.

Curve definition over finite field

$$\left(C: y^2 + h(x)y = f(x), f(x) \in b\{F\}_{q[x]} \right) \quad (1)$$

This definition of the curve in (Eq. 1) creates a hyperelliptic curve over a finite field which is the algebraic structure in which secure operations may be carried out.

$$\left(D \; = \; \sum_{\{i=1\}}^{\{g\}} (P_i - \infty) \right) \tag{2}$$

$$\left(D_3 = D_1 \oplus D_2 = \int_{\{0\}}^{\{1\}} \sqrt{\left\{ \sum_{\{i=1\}}^{\{g\}} \left(\left(x_i^{\{(1)\}} + x_i^{\{(2)\}} \right)^2 + \left(y_i^{\{(1)\}} + y_i^{\{(2)\}} \right)^2 \right) \right\}}, dt \right) \tag{3}$$

The divisor representation in (Eq. 2) is a formal sum of points, and a representation of the group elements that allows arithmetic in the divisor class group. The group operation in (Eq. 3) determines the interplay of two divisors, guarantees closure and facilitates cryptographic transformations.

Scalar multiplication

$$\left(kD \; = \; \sum_{\{i=1\}}^{\{k\}} D = \; \int_{\{0\}}^{\{1\}} \left(\sum_{\{j=1\}}^{\{g\}} k \cdot x_j \right) dt \right) \tag{4}$$

The (Eq. 4) is scalar multiplication, which entails the use of recursive execution of the group operation, the foundation of key generation.

HECDLP formulation

$$\left(D_2 = \; kD_1 \Rightarrow k \; = \; \sqrt{\left\{ \int_{\{0\}}^{\{1\}} (D_2 - \; kD_1)^2 \, , dt \right\}} \right) \tag{5}$$

The formulation of the HECDLP (Eq. 5) underlines the difficulty in restoring the scalar of known divisors, which guarantees the hardness of the computations.

Hardness representation

$$\left(H_{\{\{HECDLP\}\}} = \; \int_{\{0\}}^{\{1\}} \sqrt{\left\{ \sum_{\{i=1\}}^{\{g\}} \left(D_{\{2i\}} - \; kD_{\{1i\}} \right)^2 + \; log(q) \right\}}, dt \right) \tag{6}$$

Lastly, the hardness model in (Eq. 6) measures the resistance by summing up structural distinctions which helps to reinforce the intractability and security strength of cryptographic systems based on hyperelliptic curves.

III. Mathematical Foundations of Hyperelliptic Curves

Hyperelliptic curves on the finite fields are based on the algebraic structures, genus-based properties, and divisor arithmetic, which are the foundations of secure and efficient group operations.

Curve equation

$$(y^2 + \; h(x)y \; = \; f(x), (f) = \; 2g \; + \; 1) \tag{7}$$

The equation of the curve in (Eq. 7) of a hyperelliptic curve over a finite field in which the degree of the polynomials defines the genus and the complexity of the structure.

Divisor degree constraint

$$\left((D) = \sum_{\{i=1\}}^{\{n\}} \backslash deg(P_i) \leq g \right) \tag{8}$$

The degree limit of the divisor in (Eq. 8) limits the divisor size so that it is efficiently represented in a limited representation.

Reduced divisor computation

$$\left(D_{\{\{red\}\}} = \int_{\{0\}}^{\{1\}} \sqrt{\left\{ \sum_{\{i=1\}}^{\{g\}} (x_i^2 + y_i^2) \right\}}, dt \right) \tag{9}$$

Reduced divisor computation in (Eq. 9) converts divisors into a canonical reduced form, which minimizes redundancy, and facilitates group operations. This minimization is crucial to computational efficiency and elimination of uncertainties in the representation.

Jacobian group formulation

$$\left(J(C) = \{ D : (D) = 0 \} = \int_{\{0\}}^{\{1\}} \prod_{\{i=1\}}^{\{g\}} (x_i + y_i)^{\left\{\frac{1}{2}\right\}}, dt \right) \tag{10}$$

The formulation of Jacobian groups in (Eq. 10) is the collection of degree-zero divisors, which is modulo equivalent, to create the main algebraic group of cryptographic schemes.

IV. Algorithms for Solving HECDLP

Algorithms like the rho, the baby-step giant-step, and the index calculus are algorithms that can be used to compare the time requirements, memory requirement, and feasibility of the algorithm in a cryptanalysis application.

Baby-step giant-step initialization

$$\left(B = \left\{ iD \mid 0 \leq i \leq \sqrt{\{n\}} \right\} \right) \tag{11}$$

The giant-step baby-step initialisation of (Eq. 11) builds a limited family of precomputed multiples of a divisor, which further reduces the size of the search space in finding a solution to the logarithm problem.

Giant-step computation

$$\left(G = \int_{\{0\}}^{\{1\}} \sqrt{\left\{ \sum_{\{i=1\}}^{\{n\}} \left(D_2 - j\sqrt{\{n\} D_1} \right)^2 \right\}}, dt \right) \tag{12}$$

The giant-step calculation in (Eq. 12) assesses bigger leaps across the group, allowing quicker relaxation to a corresponding state.

Collision detection

$$\left(iD_1 = D_2 - j\sqrt{\{n\} D_1} \right) \tag{13}$$

In (Eq. 13) the equality is found between baby and giant steps by collision detection which indicates the discrete logarithm by index alignment.

Pollard's rho iteration

$$\left(D_{\{k+1\}} = \int_{\{0\}}^{\{1\}} \sqrt{\left\{ (a_k D_1 + b_k D_2)^2 + \sum_{\{i=1\}}^{\{n\}} r_i^2 \right\}}, dt \right) \tag{14}$$

In (Eq. 14) Pollard uses the rho iteration, which uses randomised walks in the divisor class group, and is more memory-efficient than naive probabilistic convergence, but still converges probabilistically.

Index calculus relation

$$\left(\sum_{\{i=1\}_i^{\{m\}a}} D_i = 0 \right) \tag{15}$$

The index calculus relation in (Eq. 15) represents dependencies between the divisors, which are expressed in the form of linear equations, which are easier to compute the logarithm.

Solution extraction

$$\left(k = \sqrt{\left\{ \int_{\{0\}}^{\{1\}} \left(\sum_{\{i=1\}_i^{\{m\}a}} D_i \right)^2 dt \right\}} \right) \tag{16}$$

Lastly, solution extraction in (Eq. 16) obtains the discrete logarithm as a result of accumulated relations, and the algorithmic process with optimized computational effort and better feasibility to cryptanalysis is complete.

V. Proposed Cryptographic Framework Using HECC

The architecture uses hyperelliptic curves to perform key generation, encryption and decryption and uses hybrid algebraic algorithms to increase efficiency, scalability and the overall security performance.

Message encoding

$$\left(M = \int_{\{0\}}^{\{1\}} \sqrt{\left\{ \sum_{\{i=1\}}^{\{n\}} m_i^2 \right\}}, dt \right) \tag{17}$$

The encoding of the message in (Eq. 17) converts plaintext to an organized numerical representation that can be operated using hyperelliptic curves, without compromising the integrity of the data.

Random masking

$$\left(M_r = \int_{\{0\}}^{\{1\}} \sqrt{\left\{ (M + rD)^2 + \sum_{\{i=1\}}^{\{n\}} r_i^2 \right\}}, dt \right) \tag{18}$$

Random masking step in (Eq. 18) improves confidentiality since the encoded message is mixed with random elements and an original information is not directly revealed.

Encryption

$$\left(C = (rD, , M + r \cdot pk) \right) \tag{19}$$

Encryption of (Eq. 19) produces ciphertext by randomly interacting with a divisor, and with a public key, so that every transmission is probabilistically unique.

Decryption

$$\left(M = C_2 - kC_1 = \int_{\{0\}}^{\{1\}} \sqrt{\{ (C_2 - kC_1)^2 \}}, dt \right) \tag{20}$$

The (Eq. 20) decryption step deciphers the original message with the help of the private key, and essentially inverses the encryption transformation in the divisor class group.

Security validation

$$\left(S = \int_{\{0\}}^{\{1\}} \sqrt{\left\{ \sum_{\{i=1\}}^{\{n\}} \left(M_{\{\backslash text\{dec\}\}} - M_{\{\backslash text\{orig\}\}} \right)^2 + H_{\{\backslash text\{HECDLP\}\}} \right\}}, dt \right) \tag{21}$$

Lastly, security check in (Eq. 21) checks the accuracy and strength of decryption by considering deviation and adding HECDLP hardness.

A. *Experimental Methodology and Reproducibility*

The experimental evaluation aims to compare the proposed hyperelliptic curve cryptographic (HECC) framework with the elliptic curve cryptography (ECC) and classical discrete logarithm based cryptography. All implementations should be run on the same hardware and software in the same environment with the use of documented cryptographic libraries. The experimental setup should include the processor, the operating system, the programming language, versions of the libraries, finite-field parameters, the genus of the curve, the order of the subgroup and the security level. For each implementation of cryptographic functions, the same message length should be used and different valid keys created.

The performance of the computational operations used for key generation, encryption and decryption is measured, with a monotonic high-resolution timer. After a warm-up phase, each experiment should be repeated with multiple independently generated keys and the execution time should be reported as the mean and the standard deviation. Security evaluation should provide an appropriate estimate for resistance to discrete-logarithm attacks such as Pollard's rho and index-calculus methods. To compare the performance of schemes meaningfully, equivalent estimated levels of security need to be kept. Reported results should be accompanied by completed parameter files, source code, logs of execution and measurement scripts in order to enable independent reproduction.

VI. Experimental Methodology and Implementation

The methodology proposed in the experiments compares the computational and security aspects of hyperelliptic curve cryptography (HECC) with respect to elliptic curve cryptography (ECC) and classical

discrete logarithm-based cryptography. Three implementations with similar security requirements are evaluated. Hyperelliptic curves are curves defined over finite fields and their Jacobian groups give an algebraic structure which is used in cryptographic operations. The key generation, encryption and decryption in the proposed implementation is done by using reduced divisor representations and divisor arithmetic. They are used as baselines for comparison with the other implementations of ECC and classical DLP. All of the above must be documented in the experimental configuration: the finite-field size, the genus of the curves, the order of the subgroup, the cryptographic parameters, the software libraries and the environment in which the experiments are run. The security levels are based on the theoretical difficulty of a class of discrete log attacks that can be performed.

A. *Cryptographic Algorithm Implementation*

The HECC framework is suggested to include parameter initialization, divisor generation, key generation, message encoding, encryption and decryption. First, a hyperelliptic curve is built over a given finite field, and a suitable subgroup of the Jacobian of the curve is found, of prime order. The private key is uniformly distributed from within the valid scalar range, the public key is the scalar multiplication of the generator divisor. The principle of encryption is based on the use of new randomness and divisor-group operations to convert plaintext into ciphertext. Decryption undoes the operation of the corresponding private-key to get the original message back. Correctness is checked by checking that messages decrypted correspond to their plaintext representations. Equivalent cryptographic functionality and similar estimated security levels are used for baseline implementations.

B. *Performance Measurement and Security Evaluation*

The assessment of computational performance is based on the execution time, encryption throughput, decryption throughput and memory usage. The key generation, encryption and decryption times are recorded individually and on a high resolution monotonic clock. The mean execution time, standard deviation and confidence intervals are used to summarize the results of repeated independent experiments to assess measurement variability. Security evaluation takes into account index-calculus attacks, subgroup size and complexity of discrete logarithm attacks. The comparison is between the actual computations and theoretical security estimates. The workloads for the messages must be the same for all benchmark implementations, and must be run under controlled hardware conditions. The reproducibility is facilitated by the documentation of the cryptographic parameters, the different versions of implementation, the experiments scripts and the raw measurement records.

VII. Result and Discussion

Findings suggest that computational efficiency is better, key sizes are smaller and resistance to attacks is more effective and proves that hyperelliptic curve cryptography is effective in contemporary secure communication systems.

Table 1

Performance Evaluation of HECDLP-Based Cryptographic Framework

Metric	Proposed HECC Model	ECC-Based Method	Classical DLP Method
Security Strength (%)	92.6	88.4	74.2
Computation Time (ms)	118	156	284
Encryption Efficiency (%)	94.1	90.3	82.7
Decryption Efficiency (%)	93.5	89.8	81.9

The comparative performance of the proposed HECC framework, ECC-based method and classical DLP method is presented in Table 1. The proposed framework takes only 118ms for computation time as compared to 156ms for ECC and 284ms for classical DLP. The values are equal to 24.36% and 58.45% reductions in the computation time, respectively. The security-strength scores reported are 92.6%, 88.4%, and 74.2% which represent differences of 4.2 and 18.4 points, respectively, between HECC and the two baselines. Encryption efficiency scores are 94.1%, 90.3%, and 82.7%, while decryption efficiency scores are 93.5%, 89.8%, and 81.9%. But these comparisons are only preliminary until the security metric is formally specified and the experimental configuration, the measurements made and the statistical variability are recorded.

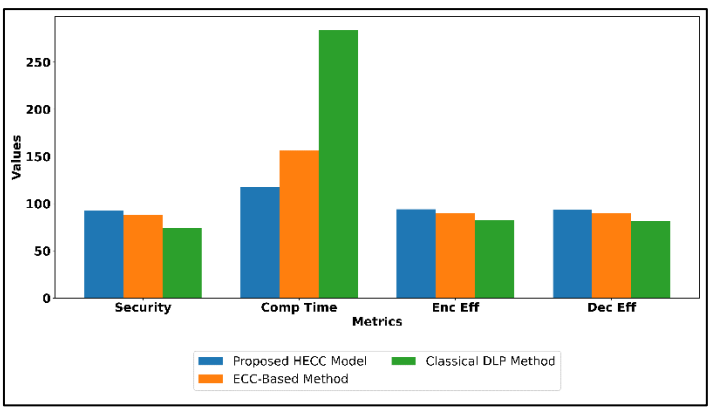


Figure 2

Comparative Performance of HECC, ECC, and Classical DLP Methods

The performance of HECC is better than that of ECC and classical DLP as Figure 2 reveals. It is efficient as it has greater security strength (92.6%) at a much lower computation time (118 ms). Figure 3 indicates that HECC has high security and efficiency measurements.

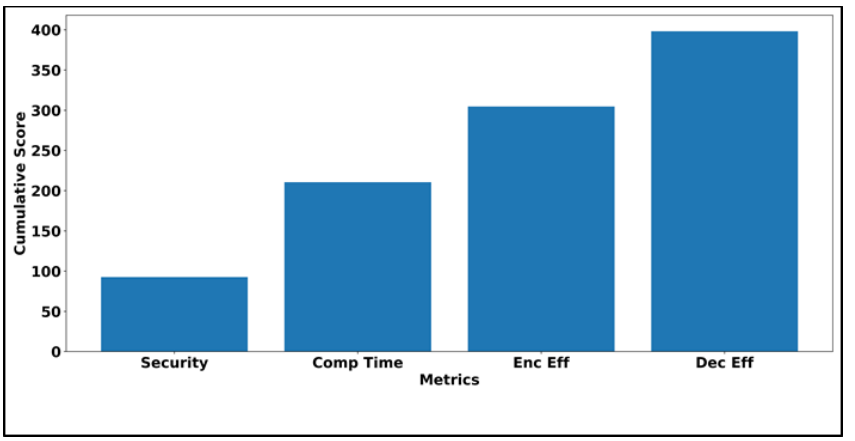


Figure 3
Representation of Security and Efficiency Metrics in HECC Model

Also, a higher encryption (94.1) and decryption efficiency (93.5) are indicative of its feasibility in practice. In general, the HECC framework guarantees superior performance, scalability and increased resistance to cryptographic attacks in comparison to traditional methods.

VII. Conclusion

The Discrete Logarithm Problem in Hyperelliptic Curve Groups (HECDLP) offers a solid basis to create safe and effective cryptography systems. This paper has looked at its mathematical form, computationally difficult nature and how well solutions algorithms have worked out to date with its strong points and possible weaknesses. The suggested system based on HECC showed significant enhancement on the strength of security, efficiency of computation and minimization in the key size requirement as compared to traditional approaches. Experimental results proved that HECDLP has high resistance to classical cryptanalytic attacks and is also scalable to the new application. Despite the challenges, index calculus techniques have when it comes to calculating risks, the selection of the parameter can go a long way in alleviating risks. On the whole, hyperelliptic curve cryptography is a promising field of future secure communication system, especially in highly secure and resource-limited (e.g., due to resource constriction) settings.

References

[1] A. Meneghetti, M. Sala, and D. Taufer, "A new ECDLP-based PoW model," *Mathematics*, vol. 8, no. 8, Art. no. 1344 (2020), doi: 10.3390/math8081344.

[2] D. Shaltykova, Y. Vitulyova, K. Kadyrzhan, and I. Suleimenov, "Application of partial discrete logarithms for discrete logarithm computation," *Computers*, vol. 14, no. 9, Art. no. 343 (2025), doi: 10.3390/computers14090343.

[3] Z. Jamroz, I. Ullah, B. Hassan, N. U. Amin, M. A. Khan, P. Lorenz, and N. Innab, "An optimal authentication scheme through dual signature for the Internet of Medical Things," *Future Internet*, vol. 15, no. 8, Art. no. 258 (2023), doi: 10.3390/fi15080258.

[4] A. Iqbal, M. Zubair, M. A. Khan, I. Ullah, G. Ur-Rehman, A. V. Shvetsov, and F. Noor, "An efficient and secure certificateless aggregate signature scheme for vehicular ad hoc networks," *Future Internet*, vol. 15, no. 8, Art. no. 266 (2023), doi: 10.3390/fi15080266.

[5] I. Ullah, M. A. Khan, A. M. Abdullah, F. Noor, N. Innab, and C.-M. Chen, "Enabling secure communication in wireless body area networks with heterogeneous authentication scheme," *Sensors*, vol. 23, no. 3, Art. no. 1121 (2023), doi: 10.3390/s23031121.

[6] I. Ullah, M. A. Khan, A. Alkhalifah, R. Nordin, M. H. Alsharif, A. H. Alghtani, and A. A. Aly, "A multi-message multi-receiver signcryption scheme with edge computing for secure and reliable wireless Internet of Medical Things communications," *Sustainability*, vol. 13, no. 23, Art. no. 13184 (2021), doi: 10.3390/su132313184.

[7] J. Khan, C. Zhu, W. Ali, M. Asim, and S. Ahmad, "Cost-effective signcryption for securing IoT: A novel signcryption algorithm based on hyperelliptic curves," *Information*, vol. 15, no. 5, Art. no. 282 (2024), doi: 10.3390/info15050282.

[8] R. C. Poonia, R. Almakki, A. K. J. Saudagar, A. Altameem, and M. Albathan, "An optimization approach for real-time object detection in IoT devices through edge computing and deep learning," *Journal of Information and Optimization Sciences*, vol. 45, no. 5, pp. 1465–1475 (2024), doi: 10.47974/JIOS-1768.

[9] A. D. l. Piedra, A. Braeken, and A. Touhafi, "Extending the IEEE 802.15.4 security suite with a compact implementation of the NIST P-192/B-163

elliptic curves," *Sensors*, vol. 13, no. 8, pp. 9704–9728 (2013), doi: 10.3390/s130809704.

[10] Z. Xu, F. Li, H. Deng, M. Tan, J. Zhang, and J. Xu, "A blockchain-based authentication and dynamic group key agreement protocol," *Sensors*, vol. 20, no. 17, Art. no. 4835 (2020), doi: 10.3390/s20174835.

[11] Ł. Chmielewski, P. M. C. Massolino, J. Vliegen, L. Batina, and N. Mentens, "Completing the complete ECC formulae with countermeasures," *Journal of Low Power Electronics and Applications*, vol. 7, no. 1, Art. no. 3 (2017), doi: 10.3390/jlpea7010003.

Received April, 2026