IAES Journal of Electrical and Electronics Engineering
Vol. 10, No. 1, June 2023, pp. 18~22
ISSN: 2338-7432, DOI: 10.11591/jeee.v10i1.pp18-22

High-security cryptographic systems with enhanced error correction and detection performance: a review of coding techniques, integrity verification, and fault-resilient design

Dodi Saputra[1], Sekar Sari[2]
[1]Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[2]Embedded Systems and Power Electronics Research Group, Yogyakarta, Indonesia

Article Info

Article history:

Received Nov 20, 2020
Revised Aug 24, 2021
Accepted Nov 2, 2021

Keywords:

Cryptographic security
Data integrity
Error detection
Error-correcting codes
Fault injection
Fuzzy extractors
Secure communication

ABSTRACT

High-security cryptographic systems must protect confidentiality and authenticity while remaining reliable when data, physical devices, or biometric inputs are imperfect. Error-correcting codes can recover corrupted information, whereas error-detecting mechanisms identify alterations or faults that should not be silently corrected. Their integration with cryptographic primitives is therefore useful in secure key reconstruction, code-based public-key cryptography, authenticated communication, biometric template protection, and hardware fault resistance. This review examines these complementary roles using literature published through 2022. It compares secure sketches and fuzzy extractors, code-based cryptosystems, error reconciliation in quantum key distribution, authenticated encryption, and parity- or code-based fault detection in cryptographic hardware. The analysis emphasizes that correction capability alone does not establish security because public helper data, decoding behavior, or residual errors may reveal information or enable attacks. Detection performance must likewise be reported through explicit measures, including fault coverage, undetected-error probability, false alarms, and decoding failure. A layered design framework is proposed to coordinate correction, integrity verification, key management, and fault response. The review identifies priorities in leakage-aware code design, adaptive decoding, independent integrity checks, realistic fault testing, and reproducible security evaluation.

This is an open access article under the CC BY-SA license.

1. INTRODUCTION

Modern digital systems rely on cryptographic mechanisms to protect sensitive information during storage, processing, and transmission. However, cryptographic security alone does not guarantee reliable operation when data are corrupted by communication noise, hardware faults, unstable power supplies, or imperfect biometric measurements. Such conditions can cause authentication failures, incorrect key reconstruction, or unexpected behavior in security-critical hardware. Consequently, combining cryptographic protection with effective error correction and detection has become an important research direction [1]-[4].

Error correction and error detection address different technical problems and should not be treated as interchangeable security mechanisms. Error-correcting codes introduce structured redundancy that enables a receiver to reconstruct data affected by a limited number of errors. Error-detecting codes identify inconsistencies, but they do not necessarily recover the original information or establish whether an alteration was intentional. Cryptographic authentication mechanisms, in contrast, verify whether data originate from an authorized source and have remained unmodified [5], [6].

Existing research has explored these functions in code-based public-key cryptography, biometric key generation, quantum key distribution, and protection against hardware fault attacks. Nevertheless, these research areas often use different evaluation criteria, making it difficult to compare their effectiveness within

a unified security architecture. This review therefore examines their complementary roles and identifies how correction, detection, and cryptographic verification can be coordinated. The objective is to establish a practical design framework that improves system reliability without weakening confidentiality, authenticity, or resistance to adversarial manipulation.

2. ERROR CORRECTION AND DETECTION IN CRYPTOGRAPHIC SYSTEMS

Error-correcting codes improve reliability by adding redundancy to information before it is stored or transmitted. The receiver uses this redundancy to identify and reconstruct corrupted symbols, provided that the error pattern remains within the code's correction capability. Common approaches include Hamming, BCH, Reed–Solomon, and low-density parity-check (LDPC) codes, each offering different trade-offs in correction capability, computational complexity, and redundancy overhead. Their application in cryptographic systems requires additional analysis because the redundancy or decoding information may expose information about the protected data [1], [2], [4].

Error detection is particularly important when a cryptographic operation is performed by hardware exposed to physical disturbances or deliberate fault injection. Parity checks, syndrome-based verification, redundant computation, and code-based signatures can identify inconsistencies in internal processing. These mechanisms are useful against natural faults and attacks that attempt to manipulate intermediate cryptographic states to reveal secret information. However, a detected fault must trigger an appropriate response, such as rejecting the output, restarting the operation, or disabling the affected component [7]-[10].

A reliable security architecture must distinguish accidental transmission errors from malicious modifications. A communication channel may introduce random bit errors, whereas an attacker may deliberately alter ciphertext, authentication data, or cryptographic intermediate states. Automatically correcting every detected discrepancy can be dangerous because an attacker may exploit the decoder's behavior or induce an incorrect correction. Therefore, error correction should be followed by independent integrity verification before recovered information is accepted [8], [11].

3. SECURE KEY RECONSTRUCTION AND BIOMETRIC CRYPTOGRAPHY

Secure sketches and fuzzy extractors provide a practical way to derive cryptographic keys from information that cannot be reproduced exactly. Biometric measurements, physical identifiers, and other noisy sources may produce slightly different digital representations during repeated acquisition. A secure sketch stores auxiliary information that helps reconstruct the original representation from a sufficiently similar measurement, while a fuzzy extractor derives nearly uniform cryptographic key material from that representation. These techniques make error tolerance possible without storing the original biometric template directly [1], [2].

Fuzzy commitment is an established construction that binds a secret codeword to a biometric representation and uses error correction during authentication. Subsequent research has examined the security of helper data, the reusability of sketches, and the risk of information leakage when multiple templates are generated from related biometric measurements. Improved fingerprint encoding using BCH codes illustrates how code selection and feature representation influence both reconstruction reliability and template protection. These findings show that a high recognition rate does not automatically imply strong cryptographic security [3], [12], [13].

Applications have extended beyond fingerprint authentication to gait recognition and secure electronic health records. Machine-vision-based gait authentication has used BCH encoding and fuzzy commitment to reconstruct keys from variable biometric features. Other studies have combined biometric fuzzy commitment with elliptic-curve cryptography and blockchain-based data management to protect sensitive records. However, biometric entropy, feature correlation, helper-data exposure, and revocability remain essential considerations when evaluating the actual security of these systems [6], [14]-[16].

4. CODE-BASED CRYPTOGRAPHY AND SECURITY ENHANCEMENT

Code-based cryptography uses mathematical problems related to decoding error-correcting codes as the foundation of public-key security. The McEliece and Niederreiter families are prominent examples in which a legitimate receiver uses secret structural information to decode a ciphertext containing deliberately introduced errors. The difficulty of solving the corresponding decoding problem without the secret information provides the basis for security. This approach is particularly relevant to post-quantum cryptography because code-based constructions are among the established alternatives to public-key systems that rely on integer factorization or discrete logarithms [5], [17].

However, the presence of error-correcting codes does not mean that every code-based cryptosystem is automatically secure. Code parameters, public-key structure, decoding algorithms, and information about the error vector can influence resistance to cryptanalysis. Research on modified McEliece constructions and code-based signatures has examined how parameter selection and structural properties affect security. Consequently, the design must balance decoding performance, key size, computational cost, and resistance to known attacks [17], [18].

Error-correction coding can also be employed as an additional security mechanism for selected encryption schemes. Mihaljević and colleagues proposed an approach in which coding and simulated synchronization errors create different channel characteristics for legitimate users and potential attackers. Their analysis illustrates that coding can contribute to security enhancement when its structure and parameters are integrated into a carefully defined threat model. Nevertheless, such techniques should supplement rather than replace established encryption and authentication mechanisms [19].

5. ERROR RECONCILIATION AND SECURE COMMUNICATION

Quantum key distribution (QKD) provides an important example of the relationship between error correction and cryptographic security. Measurements performed by two legitimate parties may produce slightly different raw keys because of channel noise, imperfect devices, or environmental disturbances. Information reconciliation uses error-correcting procedures to make their keys consistent while limiting the information disclosed through public communication. Privacy amplification is subsequently applied to reduce any information that an adversary may have obtained [20]-[22].

LDPC codes are attractive for reconciliation because their sparse parity-check structures support iterative decoding and efficient processing. Syndrome-based approaches can reduce unnecessary communication and improve the use of information exchanged during reconciliation. Research on continuous-variable QKD has also investigated error-correction protocols designed to improve reconciliation performance under different signal conditions. These studies demonstrate that correction efficiency, communication overhead, and information leakage must be evaluated together rather than independently [20]-[22].

Physical-layer security provides another complementary approach by exploiting characteristics of the communication channel to improve confidentiality. Polar-code construction has been investigated as a way to adapt coding structures to channel-quality information and security requirements. However, physical-layer techniques do not eliminate the need for cryptographic authentication and integrity verification at higher protocol layers. A secure communication system should therefore combine channel coding, cryptographic protection, and post-transmission verification [23].

6. HARDWARE FAULT DETECTION AND INTEGRITY PROTECTION

Cryptographic hardware may be exposed to voltage glitches, clock disturbances, electromagnetic interference, temperature variation, or deliberate physical manipulation. Such disturbances can produce incorrect intermediate values that may be exploited through differential fault analysis. Error-detecting codes provide an additional protection layer by monitoring internal data and identifying unexpected computational results. Experimental studies using parity bits and Hamming-code-based signatures have demonstrated the potential of these methods for detecting faults in AES implementations [7], [8].

More advanced approaches combine error correction with fault detection to distinguish recoverable disturbances from potentially malicious events. Gay and colleagues proposed an error-control architecture that supports correction of selected low-multiplicity faults while improving detection of more complex disturbances. Other research has examined fine-grained parity checking to improve detection coverage against timing fault injection while controlling hardware overhead. These results indicate that protection effectiveness depends on the fault model, circuit architecture, and verification strategy [7], [9], [10].

Authenticated encryption provides a separate but essential mechanism for verifying data integrity and authenticity. Unlike ordinary error-detection codes, authenticated encryption is designed to reject unauthorized modifications through cryptographic verification. A systematic review of authenticated encryption schemes identified the continuing challenge of balancing security properties, implementation efficiency, and functional requirements. Integrating authenticated encryption with error-control mechanisms can improve reliability, provided that corrected data are not accepted before authentication succeeds [24].

7. COMPARATIVE ASSESSMENT OF TECHNIQUES

The reviewed methods address different sources of uncertainty and different security objectives. Fuzzy extractors primarily support reliable key reconstruction from noisy inputs, whereas code-based cryptography uses decoding complexity as a security foundation. QKD reconciliation corrects differences between generated keys, while hardware error-detection mechanisms identify faults during cryptographic processing. Table 1 summarizes these approaches and their principal design considerations.

Table 1. Comparison of error-control and security techniques in high-security cryptographic systems

Technique	Main purpose	Key advantage	Main limitation
BCH and Hamming codes	Error correction and detection	Structured decoding and hardware suitability	Limited correction capability or redundancy overhead
Reed–Solomon codes	Burst and symbol-error correction	Effective for symbol-oriented data	Decoding complexity and storage overhead
Secure sketches	Reconstruction of noisy inputs	Enables error-tolerant recovery	Helper-data leakage and reuse risks
Fuzzy extractors	Cryptographic key generation	Converts noisy inputs into stable key material	Entropy and reconstruction constraints
Code-based cryptography	Public-key encryption and signatures	Security based on hard decoding problems	Large keys and parameter sensitivity
LDPC reconciliation	Key agreement and communication reliability	Efficient iterative decoding	Processing and information-leakage overhead
Parity and syndrome checks	Hardware fault detection	Relatively low-cost monitoring	Some multi-bit or structured faults may escape detection
Authenticated encryption	Confidentiality and integrity	Cryptographic rejection of unauthorized modifications	Does not correct corrupted data

The comparison highlights that no single technique provides complete protection against all relevant threats. Error-correcting codes improve data recovery, but their correction capability must be bounded and evaluated under realistic error patterns. Cryptographic authentication is necessary to reject unauthorized changes, while hardware monitoring is needed to identify faults that occur during internal processing. An effective architecture therefore combines complementary mechanisms according to the threat model and operational requirements.

8. PERFORMANCE EVALUATION AND RESEARCH CHALLENGES

The phrase "improved detection rate" requires a clearly defined measurement because different applications identify different types of errors. Hardware fault detection may be evaluated through fault coverage and the probability of undetected faults, while biometric systems commonly report false acceptance and false rejection rates. Communication systems may instead emphasize bit-error rate, frame-error rate, decoding failure probability, and reconciliation efficiency. These measures should not be combined into a single performance score without explaining their definitions and experimental conditions [8], [9], [12], [20].

A major research challenge is the interaction between error correction and information leakage. Public helper data, parity information, and reconciliation messages may reveal partial information about the protected input or key. Repeated use of the same helper data can also weaken security under certain biometric and key-reconstruction models. Future designs should therefore evaluate both the probability of successful reconstruction and the amount of information exposed to an adversary [3], [4], [19].

Another challenge concerns the limited comparability of experimental results across different cryptographic applications. Studies frequently use different datasets, error models, hardware platforms, and attack assumptions, making reported detection rates difficult to compare directly. Future investigations should adopt reproducible benchmarks that include random errors, burst errors, timing faults, fault injection, and deliberate data modification. The evaluation should also report processing time, memory requirements, energy consumption, false alarms, and undetected-error probability [7]-[10].

9. CONCLUSION

High-security cryptographic systems require both strong cryptographic protection and reliable mechanisms for handling errors and faults. Error-correcting codes, secure sketches, fuzzy extractors, code-based cryptography, and LDPC reconciliation provide complementary capabilities for recovering information and establishing reliable cryptographic keys. Parity checks, syndrome verification, and authenticated encryption further strengthen the system by identifying faults and rejecting unauthorized modifications. Their

integration must be guided by a clear threat model because correction alone cannot guarantee confidentiality, authenticity, or resistance to deliberate attacks.

The reviewed literature indicates that a layered architecture offers a practical foundation for improving security and reliability. Such an architecture should combine bounded error correction, independent integrity verification, secure key management, and fault-responsive hardware protection. Future research should prioritize leakage-aware coding, adaptive error-control mechanisms, realistic fault-injection experiments, and standardized performance measurements. These developments can support cryptographic systems that are more resilient, measurable, and dependable in increasingly demanding digital environments.

REFERENCES

[1] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM Journal on Computing*, vol. 38, no. 1, pp. 97–139, Jan. 2008, doi: 10.1137/060651380.

[2] A. Juels and M. Wattenberg, "Fuzzy commitment scheme," in *Proceedings of the 6th ACM conference on Computer and communications security*, New York, NY, USA: ACM, Nov. 1999, pp. 28–36. doi: 10.1145/319709.319714.

[3] M. Blanton and M. Aliasgari, "Analysis of reusability of secure sketches and fuzzy extractors," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1433–1445, Sep. 2013, doi: 10.1109/TIFS.2013.2272786.

[4] B. Fuller, X. Meng, and L. Reyzin, "Computational fuzzy extractors," *Information and Computation*, vol. 275, p. 104602, Dec. 2020, doi: 10.1016/j.ic.2020.104602.

[5] N. Sendrier, "Code-Based Cryptography: State of the Art and Perspectives," *IEEE Security and Privacy*, vol. 15, no. 4, pp. 44–50, 2017, doi: 10.1109/MSP.2017.3151345.

[6] S. Barman, S. Chattopadhyay, D. Samanta, and S. Barman, " A blockchain-based approach to secure electronic health records using fuzzy commitment scheme ," *Security and Privacy*, vol. 5, no. 4, Jul. 2022, doi: 10.1002/spy2.231.

[7] M. Gay, B. Karp, O. Keren, and I. Polian, "Error control scheme for malicious and natural faults in cryptographic modules," *Journal of Cryptographic Engineering*, vol. 10, no. 4, pp. 321–336, Nov. 2020, doi: 10.1007/s13389-020-00234-7.

[8] F. E. Potestad-Ordonez, E. Tena-Sanchez, A. J. Acosta-Jimenez, C. J. Jimenez-Fernandez, and R. Chaves, "Design and evaluation of countermeasures against fault injection attacks and power side-channel leakage exploration for AES Block Cipher," *IEEE Access*, vol. 10, pp. 65548–65561, 2022, doi: 10.1109/ACCESS.2022.3183764.

[9] M. Zhang, H. Li, P. Wang, and Q. Liu, "Parity check based fault detection against timing fault injection attacks," *Electronics*, vol. 11, no. 24, p. 4082, Dec. 2022, doi: 10.3390/electronics11244082.

[10] M. J. Mihaljević, L. Wang, and S. Xu, "An approach for security enhancement of certain encryption schemes employing error correction coding and simulated synchronization errors," *Entropy*, vol. 24, no. 3, p. 406, Mar. 2022, doi: 10.3390/e24030406.

[11] M. A. Jimale *et al.*, "Authenticated encryption schemes: a systematic review," *IEEE Access*, vol. 10, pp. 14739–14766, 2022, doi: 10.1109/ACCESS.2022.3147201.

[12] S. Shukla and S. J. Patel, "Securing fingerprint templates by enhanced minutiae-based encoding scheme in Fuzzy Commitment," *IET Information Security*, vol. 15, no. 3, pp. 256–266, May 2021, doi: 10.1049/ise2.12024.

[13] T. Van Hamme, E. A. Rua, D. Preuveneers, and W. Joosen, "On the security of biometrics and fuzzy commitment cryptosystems: a study on gait authentication," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5211–5224, 2021, doi: 10.1109/TIFS.2021.3124735.

[14] L. A. Elrefaei and A. M. Al-Mohammadi, "Machine vision gait-based biometric cryptosystem using a fuzzy commitment scheme," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 2, pp. 204–217, Feb. 2022, doi: 10.1016/j.jksuci.2019.10.011.

[15] T. S. C. Lau and C. H. Tan, "On the design and security of Lee metric McEliece cryptosystems," *Designs, Codes, and Cryptography*, vol. 90, no. 3, pp. 695–717, Mar. 2022, doi: 10.1007/s10623-021-01002-2.

[16] K. Gümüş *et al.*, "A novel error correction protocol for continuous variable quantum key distribution," *Scientific Reports*, vol. 11, no. 1, p. 10465, May 2021, doi: 10.1038/s41598-021-90055-3.

[17] Y. Zeng, Y. Tang, and L. Xiang, "Physical layer security design for polar code construction," *Cryptography*, vol. 6, no. 3, p. 35, Jul. 2022, doi: 10.3390/cryptography6030035.

[18] H. K. Mao, Y. C. Qiao, and Q. Li, "High-efficient syndrome-based ldpc reconciliation for quantum key distribution," *Entropy*, vol. 23, no. 11, p. 1440, Oct. 2021, doi: 10.3390/e23111440.

[19] T. Tsurumaru, "Equivalence of three classical algorithms with quantum side information: privacy amplification, error correction, and data compression," *IEEE Transactions on Information Theory*, vol. 68, no. 2, pp. 1016–1031, Feb. 2022, doi: 10.1109/TIT.2021.3126160.

[20] Y. Huang, X. Zhang, and X. Ma, "Stream privacy amplification for quantum cryptography," *PRX Quantum*, vol. 3, no. 2, p. 020353, Jun. 2022, doi: 10.1103/PRXQuantum.3.020353.

[21] G. Passante, O. Moussa, and R. Laflamme, "Measuring geometric quantum discord using one bit of quantum information," *Physical Review A*, vol. 85, no. 3, p. 032325, Mar. 2012, doi: 10.1103/PhysRevA.85.032325.

[22] M. Lucamarini, Z. L. Yuan, J. F. Dynes, and A. J. Shields, "Overcoming the rate–distance limit of quantum key distribution without quantum repeaters," *Nature*, vol. 557, no. 7705, pp. 400–403, May 2018, doi: 10.1038/s41586-018-0066-6.

[23] D. A. McGrew and J. Viega, "The Security and Performance of the Galois/Counter Mode (GCM) of Operation," in *International Conference on Cryptology in India*, 2004, pp. 343–355. doi: 10.1007/978-3-540-30556-9_27.

[24] I. Salam, T. H. Ooi, L. Xue, W.-C. Yau, J. Pieprzyk, and R. C.-W. Phan, "Random differential fault attacks on the lightweight authenticated encryption stream cipher grain-128AEAD," *IEEE Access*, vol. 9, pp. 72568–72586, 2021, doi: 10.1109/ACCESS.2022.3147201.