Gwangju Institute of Science and Technology

Dissertation for Doctoral Degree

Error-Correction Code for Trust and Stealth

Haeung Choi

College of Information and Computing

Department of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

2025

Error-Correction Code for Trust and Stealth

신뢰성 및 은닉성을 위한 오류정정부호

Error-Correction Code for Trust and Stealth

Advisor: Professor Heung-No Lee

by

Haeung Choi

Collage of Information and Computing

Department of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

A dissertation submitted to the faculty of Gwangju Institute of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical Engineering and Computer Science

Gwangju, Republic of Korea

May 26, 2025

Approved by

Professor Heung-No Lee

Committee Chair

Error-Correction Coder for Trust and Stealth

Haeung Choi

Accepted in partial fulfillment of the requirements for

the degree of Doctor of Philosophy

May 26, 2025

Committee Chair

Professor Heung-No Lee

Committee Member

Professor Jong Won Shin

Committee Member

Professor Euiseok Hwang

Committee Member

Professor Yongwoo Lee

Committee Member

Dr. Sangjun Park

PhD/EC
20152070

Haeung Choi (최해웅), Error-Correction Code for Trust and Stealth (신뢰성 및 은닉성을 위한 오류정정부호). College of Information and Computing, Department of Electrical Engineering and Computer Science, 2025. 91 p. Prof. Heung-No Lee

Abstract

Error-correction codes are indispensable not only for reliable data transmission but also as hardness assumptions in modern cryptography such as public key encryption, digital signature/identification, hash function, zero-knowledge proof, etc. Since McEliece's pioneering work, code-based cryptography systems have progressed from theory to standardization. For instance, Hamming Quasi-Cyclic (HQC) has been selected for standardization in the NIST post-quantum cryptography program, and Classic McEliece remains under active consideration.

Beyond cryptography system, error-correction codes play an important role in steganography. Steganography aims to secure communication by embedding a secret message into a cover medium such as digital images. Error-correction code can be designed to embed as many secret symbols with minimal distortion, reducing the chance that a warden will detect the presence of hidden symbols.

A blockchain is a peer-maintained, distributed ledger that operates without central custody. Owing to its decentralized control and tamper-evident record, the technology now underpins diverse applications, including cryptocurrencies, digital identity, distributed storage, intellectual-property timestamping, governance voting, IoT, and energy-carbon trading. Decentralization is the foundation of these properties: it sustains immutability, censorship resistance, fault tolerance, and public auditability. The consensus algorithm is the mechanism that preserves trust of decentralization. In practice, however, specialized mining hardware, for example application-specific integrated circuits (ASICs), has undermined equal participation. ASICs deliver orders-of-magnitude performance gains, excluding users who lack such hardware and raising the economic barrier to meaningful participation, thereby breaking the trust of network decentralization.

Error-correction code can be applied to improve the trust of blockchain consensus. By using the one-wayness and design flexibility of error-correction decoder, one can design a blockchain consensus algorithm which is verifiable, robust, and fair.

In radio communication, security has been one main concern for a long time. The main stream study to pursue the security in communication has been protect the message from unauthorized users. Information-theoretic security and computational security are widely adapted security criterion. Encryption, physical layer securing methods and other many schemes are proposed to achieve such security criterions.

Covertness is another security criterion that has a slightly different objective from information-theoretic and computational security. This criterion seeks to hide the existence of the message rather than its contents. Steganography is a representative approach to achieving covertness. It alters a small part of the cover medium (e.g., images) to embed a secret message that cannot be detected by unauthorized users. However, covertness in radio communications is less studied because the cover medium is more difficult to control.

Error-correction codes can be applied to improve the stealth of radio communication. To achieve covertness, the secret message must be embedded while minimally distorting the radio signal. By designing an error-correction code that minimally distorts the channel, one can embed more information with a lower probability of detection by eavesdroppers.

This dissertation presents two error-correction code based contributions: one that achieves trust in blockchain networks and another that enables stealthy wireless communication in the presence of hostile jamming.

Chapter 2 of this dissertation presents error-correction code based method to achieve trust of blockchain. In this study, we propose Error-Correction-Code Verifiable Computation Consensus (ECCVCC), a proof-of-work variant in which each puzzle instance is a decoding problem. By randomizing code parameters on a per-block basis, ECCVCC neutralizes the economies of scale enjoyed by application-specific integrated circuits, thereby restoring competitive parity to general-purpose hardware. Analytical bounds and Monte-Carlo experiments confirm tightly controlled block-generation times, stable difficulty adjustment, and markedly slower centralization compared with conventional hash-based PoW.

Chapter 3 of this dissertation introduces error-correction code based method to achieve stealth wireless communication. In this study, we propose Gaussian-coded time-frequency modulation (G-TFM), whose coded symbol constellation is statistically indistinguishable from ambient additive white Gaussian noise. Two sparse-recovery receivers are developed to estimate and cancel narrow-band and wide-band jammers while simultaneously decoding the covert payload. Covertness is measured by the adversary's probability of detection, whereas robustness is quantified through bit-error-rate simulations under tone, sweep, and broadband jamming scenarios.

These contributions demonstrate that error-correction codes can serve as a unifying toolset for enhancing security in both decentralized ledgers and covert wireless links. The thesis delivers (i) a cryptographically sound, ASIC-resistant consensus protocol, and (ii) a unified framework that simultaneously achieves anti-jamming and covert objectives thereby advancing the state of the art in code-based security engineering.

Contents

List of Tables

List of Algorithms

List of Figures

Chapter I. Introduction

1.1. Error-Correction Code

Error-correction codes (ECCs) are mathematical techniques that add controlled redundancy into digital data so that a receiver can detect and correct errors introduced by noise, interference, or hardware faults. First formalized in Claude Shannon's 1948 work on information theory [1], ECC enables reliable communication even when the underlying channel is imperfect or the medium is degraded. Figure 1 illustrates the basic concept of ECC. With ECC, the message is encoded to a codeword in a higher-dimensional space and then transmitted. The error occurred by communication channel can be recovered by decoding process, thanks to the redundancy of the codeword.

A binary linear block code can be a simple example. An message vector, $\mathbf{m} \in \{0,1\}^k$, is encoded to yield an codeword vector, $\mathbf{c} \in \{0,1\}^n$. The encoding procedure is represented as a matrix-vector multiplication, $\mathbf{c} = \mathbf{mG}$, where $\mathbf{G} \in \{0,1\}^{k \times n}$ is called generator matrix. Here, the number of errors that the code can recover—often called error-correction capability—is closely related to the minimum distance among codewords. Decoding from erroneous codeword, $\mathbf{r} \in \{0,1\}^n$, to the original message can be done by various algorithms depending on the specific design of ECC. For example, message passing algorithm is generally adopted for decoding of low-density parity-check (LDPC) code [2]. For linear codes, sometime parity-check matrix (PCM) and can be used to transform the codeword decoding into its dual problem called syndrome decoding.

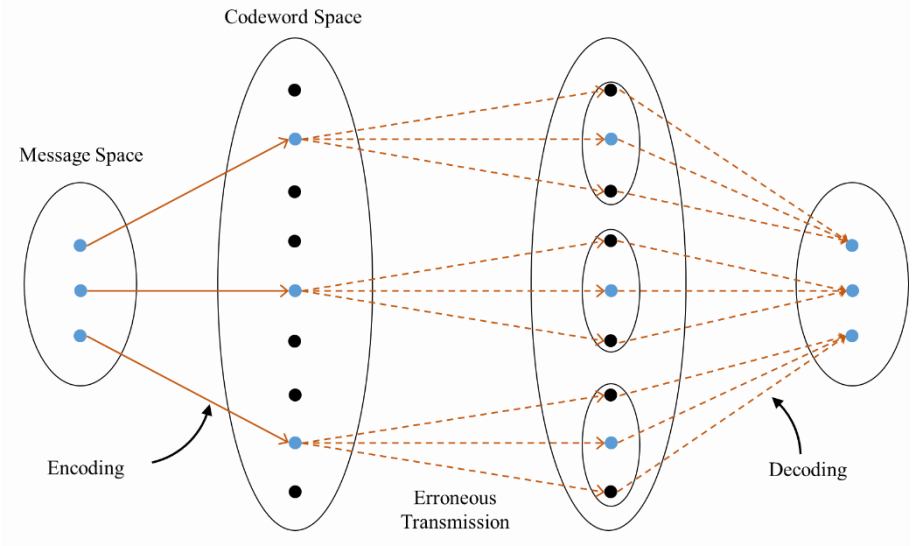


Figure 1. Concept of Error-Correction Code.

1.1.1. Error-Correction Code for Cryptography

Error-correction codes are now indispensable not only for correcting errors in communication systems but also in other areas of information technology, such as cryptography. The computational hardness of decoding a random linear code enables cryptographic constructions based on this problem. Since McEliece's pioneering work [3], code-based cryptography has attracted considerable attention and recently received more attentions as an post-quantum cryptography. Current public-key cryptosystems rely on the hardness of discrete logarithm or integer factorization problem are known to be vulnerable against the Shor's algorithm on quantum computing. Code-based cryptography is known to be able to provide post-quantum hard problems, for example decoding of a random linear code, which has no efficient algorithm even in quantum computers. Many code-based cryptography systems such as Classic McEliece [4] and Hamming Quasi-Cyclic (HQC) [5] are submitted for National Institute of Standards and Technology (NIST) post-quantum cryptography (PQC) standardization.

A. McEliece Cryptosystem

Classic McEliece is a key-encapsulation mechanism (KEM) based on McEliece and Niederreiter cryptosystems using binary Goppa codes. In the McEliece cryptosystem, a randomly constructed (n, k) binary Goppa code with a known efficient decoding algorithm is selected. The generator matrix $\mathbf{G} \in \{0,1\}^{k \times n}$ of this code is kept secret. Together with $\mathbf{G}$, a nonsingular randomizing matrix $\mathbf{S} \in \{0,1\}^{k \times k}$ and a permutation matrix $\mathbf{P} \in \{0,1\}^{n \times n}$ form the private key $K_{\text{private}} = (\mathbf{G}, \mathbf{S}, \mathbf{P})$. Public key is generated by $\mathbf{G}' = \mathbf{SGP}$. To encrypt a message $\mathbf{m} \in \{0,1\}^k$, the sender uses the public key and adds an error vector $\mathbf{e}$. The resulting ciphertext becomes

$$\mathbf{c} = \mathbf{m}\mathbf{G}' + \mathbf{e}, \tag{1}$$

where $\mathbf{e}$ is a binary error vector of length n and Hamming weight t. With appropriately chosen parameters (n, k, t), decoding the random linear code without the secret key is computationally infeasible, whereas a legitimate recipient can decrypt by computing

$$\mathbf{c}\mathbf{P}^{-1} = \mathbf{m}\mathbf{S}\mathbf{G} + \mathbf{e}\mathbf{P}^{-1}, \tag{2}$$

and applying the efficient decoder. The scheme can be stated equivalently in the Niederreiter form, which uses a parity-check matrix and syndrome decoding instead of a generator matrix and codeword decoding [6].

The most effective known attack against McEliece systems is information set decoding (ISD). The ISD repeatedly selects an *information set*, a subset of coordinates on which the restricted parity-check sub-matrix is invertible. Then, ISD aims to recover the error vector with the information set. Since Prange's original proposal

[7], many variants improved the attack efficiency by reducing the searching space of ISD [8]. However, the attack remains essentially combinatorial search; its complexity rapidly grows to an infeasible level by increasing the code length.

Although Classic McEliece has not been adopted in the current NIST POC standard, it is still regarded as secure and is expected to be standardized soon. NIST's fourth-round report notes that Classic McEliece was not selected because it is already under consideration by the International Organization for Standardization (ISO); concurrent standardization by both bodies could lead to incompatible specifications, not because of any security weakness [9]. Indeed, the McEliece cryptosystem has withstood cryptanalytic efforts since its introduction in the late 1970s.

B. Hamming Quasi-Cyclic Cryptosystem

Hamming Quasi-Cyclic (HQC) is another promising code-based KEM, which is selected as NIST standard in its fourth round challenge. As like McEliece cryptosystem, HQC cryptosystem also based on the hardness of decoding random codes. More specically, its security relies on the hardness of quasi-cyclic syndrome decoding (QCSD) problem [10]. The main difference of HQC cryptosystem from McEliece cryptosystem is that HQC cryptosystem uses two different codes. The first code is *public code*, an (n, k) code whose generator matrix and efficient decoding algorithm are publicly known. The second code is *random code*, a randomly selected ($2n$, n) code whose generator matrix is circulant in systematic form. The random code is used to construct an error vector with predictable Hamming weight, which can be decoded by the public code. The generator matrix of the random code is denoted by

$$\mathbf{H} = [\mathbf{I}_n \,|\, \mathbf{rot(h)}], \tag{3}$$

where $\mathbf{I}_n$ is the n-dimensional identity matrix and **rot**(**h**) is an right-wise circulant matrix whose last row is the reverse of the vector **h**. Here, **h** is a randomly selected *short* vector from a ring of polynomials,

$$\mathcal{R} = \mathbb{F}[X]/(X^n - 1), \tag{4}$$

over a finite field $\mathbb{F}$. The private key (**x**, **y**) is constructed by randomly select two elements, **x** and **y**, from $\mathcal{R}$. The public key becomes (**h**, **s**), where **s** is the syndrome such that

$$\mathbf{s} = [\mathbf{x} \,|\, \mathbf{y}]\mathbf{H}^T. \tag{5}$$

To encrypt a message **m**, the sender generates a ciphertext (**u**, **v**) as follows:

$$\mathbf{u} = [\mathbf{r}_1 \mid \mathbf{r}_2]\mathbf{H}^T, \tag{6}$$

$$\mathbf{v} = \mathbf{mG} + \mathbf{s} \cdot \mathbf{rot}(\mathbf{r}_2)^T + \mathbf{e}, \tag{7}$$

where $\mathbf{G}$ is the generator matrix of *public code*. With appropriately chosen Hamming weights for $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{e}$, a legitimate recipient can decrypt $\mathbf{m}$ by using the decoder of the public code for

$$\mathbf{v} - \mathbf{u} \cdot \mathbf{rot}(\mathbf{y})^T = \mathbf{mG} + \mathbf{r}_1\mathbf{rot}(\mathbf{y})^T + \mathbf{r}_2\mathbf{rot}(\mathbf{x})^T + \mathbf{e}. \tag{8}$$

The security of HQC cryptosystem relies on the difficulty of decoding random quasi-cyclic codes, which is well studied. HQC has advantages over Classic McEliece such as no hidden structure in the code, efficient decoding, smaller public key size, and accurate decryption failure rate.

1.1.2. Error-Correction code for Steganography

Steganography is an information-hiding method that concealing the existence of message, while cryptography aims to protect the context of the message [11]. In steganography systems, a message is embedded into redundant bits in cover medium such as image, without distorting the statistical properties of the medium [12]. The cover medium is called stego text, often called after type of medium. For example, stego image for image steganography.

A syndrome of error-correction code can be used for image steganography. Syndrome-based embedding is a steganography method that minimize the distortion of cover image. Basic principle underlying the syndrome-based embedding is that: an appropriately designed ECC maps each syndrome to corresponding error pattern with smallest weight (often called coset leader). In this method, an error vector is constructed by the syndrome decoding procedure regarding the secret message as a syndrome vector. The secret message can be embedded by adding the error vector to the cover medium, for example discrete cosine transform (DCT) coefficients of a digital image. Syndrome-Trellis Codes (STCs) [13], Steganographic Polar Codes (SPCs) [14], and Low-Density Generator-Matrix (LDGM) code based approaches [15] are known as near-optimal embedding methods with minimal distortion.

1.2. Blockchain Consensus

Distributed ledger is a ledger which is stored and managed by peer-to-peer and distributed manner without a centralized custody. Blockchain is a kind of distributed ledger, whose contents are divided and managed in small units called blocks. Blockchain became a key technology to guarantee the trust among decentralized

nodes without a centralized custody. The early blockchains such as Bitcoin [16] have only restricted usage for example cryptocurrency. However, recent generations of blockchains have widely applicable to a variety of fields by supporting smart contracts, which are computing codes carved in the blockchain.

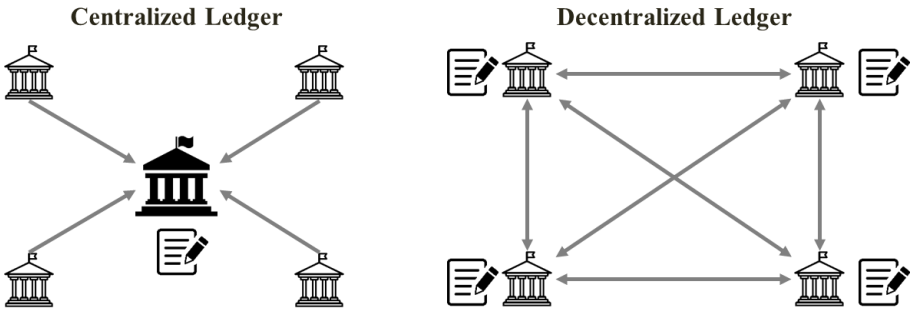


Figure 2. Concepts of Centralized and Decentralized Ledger.

If there are differences between contents in ledgers (or blocks) of each peer, consensus algorithm determines which contents are canonical. In blockchain, the consensus part plays a role in leading an agreement among trustless nodes without any communications. This part is the most innovative because it can prevent the double spending attack [17] in a peer-to-peer network in the absence of trusted parties. In Bitcoin [16], as an example, more than ten thousand of nodes randomly scattered across the world aim to reach a consensus in each block time. The Internet is the only way to connect them; communication packets are delayed and sometimes dropped though the Internet that is designed to provide the best effort service. Cyberattacks frequently happen, making transactions over the Internet insecure. Nevertheless, Bitcoin has shown secure peer-to-peer transactions over the past 10 years.

1.2.1. Proof-of-Work Consensus

Many blockchains, including Bitcoin, rely on a consensus mechanism known as proof-of-work (PoW). In PoW, nodes compete in a computational process called mining, to append the next block to the chain. Each miner must find a solution to a cryptographic puzzle and submit it as proof of its work. The first node to solve the puzzle earns the right to publish the new block and receives a coin reward. An adversary wishing to rewrite the ledger would need to re-solve the PoW puzzle for that block and for every subsequent block. Acquiring the requisite computational power is practically infeasible for any single node, so no participant can unilaterally modify published blocks. This resistance to alteration is what gives PoW its immutability.

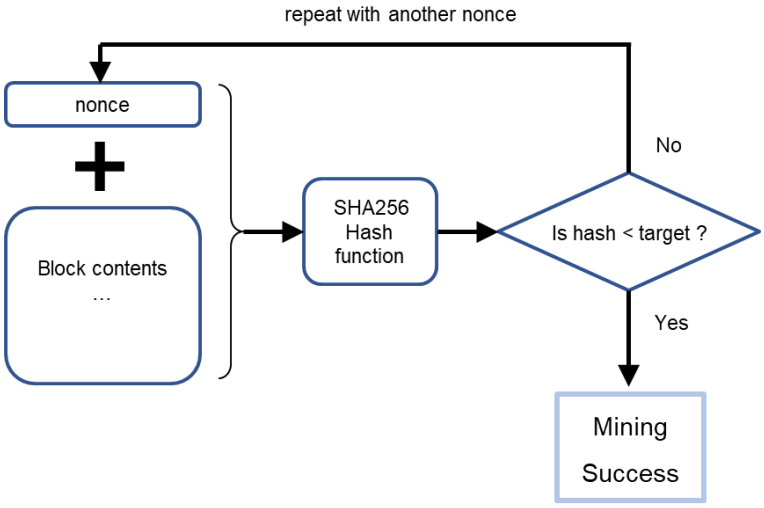


Figure 3. Blockdiagram of Bitcoin's Proof-of-Work.

1.2.2. Proof-of-Stake Consensus

Another class of consensus mechanism adopted by multiple blockchains is proof-of-stake (PoS), in which immutability is secured with a deposit of the block publishers as collateral. In this scheme, the block validators deposit their cryptocurrency (often called staking) on the blockchain protocol. At each block generation time slot, one validator is selected as a proposer, who has responsibility to generate a new block. Then other validator voting on whether the block is valid or not. If any proposer or validators does malicious action such as propose/vote on invalid block, the malicious user loss the collateral (often called "slashed"). In other words, the publishers bet their deposit, rather than computational effort, to prove the soundness of the block. If a block is revealed to have a flaw by other verifiers, the publisher loses the deposit. The PoS has advantages in terms of energy consumption and stability of block generation time than PoW. However, its security relies on two assumptions: 1) proposers are selected randomly and 2) verifiers work rationally.

1.2.3. Chain Selection Rule

In addition to the proof-of-X schemes, chain selection rule is another important component of consensus protocol. It provides a rule that determines which fork of a blockchain is valid one. Bitcoin's longest chain rule and Ethereum's heaviest chain rule are representative examples. The chain selection rule plays key role when a blockchain is forked because of clock asynchronization, network delay and selfish mining attack.

In Bitcoin, miners make rational decisions to maximize their profits by following a two stage process in which i) the miners select a blockchain whose length is the longest and ii) they extend this longest one by adding a newly mined block. Suppose there are two blockchains where one is longer than the other one in terms of the length. Since the longer chain has the more accumulated works, altering it is more difficult. This longer chain shall

be treated the more trustable and preferable by the miners. Thus, they select the longer chain. Making such a selection is rational for the sake of keeping the mining rewards. The mining reward is a delayed conditional payment. This time delay is measured in terms of the number of blocks, say 100 blocks. If this mined block was not a part of the longest chain at the future 100 block, the reward vanishes. Thus, rational miners select the longest chain.

1.3. Covert Communication

Since the advent of radio, security has been a central concern. Due to the open nature of wireless medium, any transmission can be intercepted by unintended receivers. Among these unintended recipients, eavesdroppers are malicious entities that seek to exploit intercepted signals for harmful purposes. Ensuring security in wireless communications therefore involves protecting legitimate users from such threats. Information-theoretic secrecy and computational security serve as two well-established criteria for defending against eavesdropper. Cryptography can achieve both criterions but most of modern cryptographic system aims computational security.

Covert communication [18], [19] is another security method to hide the radio transmission from an eavesdropper, whose aim is slightly different from cryptography. However, covert communication has received little attention compared to the other secrecy criterion since its tight restriction. In the remainder of section, we introduce and compare the three security criterions.

1.3.1. Perfect Secrecy

Information-Theoretic Secrecy is a type of security criterion models an adversary who has unlimited computational power. A scheme achieves information-theoretic secrecy when the probability that the adversary can recover the plaintext from the ciphertext is essentially the same as the probability of guessing the plaintext at random. When these two probabilities coincide exactly, the system provides perfect secrecy. The one-time pad is the classic example of a method that achieves the perfect secrecy.

Wyner [20] introduced the concept of secrecy capacity for the wiretap channel illustrated in Figure 4. The secrecy capacity is the maximum rate at which information can be transmitted error-free to the legitimate receiver while remaining confidential from the eavesdropper, i.e., transmission with any rate below this threshold is achievable with perfect secrecy. Formally, secrecy capacity is defined as follows:

$$C_{\text{secrecy}} = \max[I(X;Y) - I(X;Z)], \tag{9}$$

where $I(X;Y)$ denotes the mutual information between X and Y. The criterion implies there exist a perfectly secure transmission scheme where the eavesdropping channel is degraded than the legitimated channel.

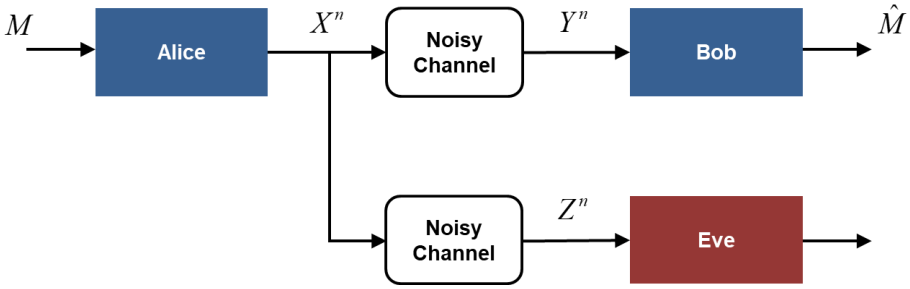


Figure 4. Wiretap Channel Model

Researchers have extended the study on secrecy capacity in several directions. Generalizing the derivation to a family of gamma fading channels [21], [22] and a MIMO channel [23] are the examples. Although secrecy capacity analysis proves that perfectly secure communication is theoretically possible, designing practical communication schemes that achieves perfect secrecy remains difficult. One promising approach is physical-layer based method with artificial-noise injection into non-degraded channels as studied in [24], [25], [26].

1.3.2. Computational Security

Computational security is a security criterion often used to evaluate the security of cryptography system. Although cryptography can achieve both computational security and information security, it is challenging to design a practical cryptography system. For example, one-time pad is proven as a perfect security achieving encryption scheme [1] but it requires secret information as long as transmitted message which have to be shared before communication; which is infeasible in most of applications. Instead, most of cryptography system targets computational security assuming adversaries with limited computing power.

Strictly speaking, a cryptography system is considered computationally secure if an attacker would need at least N operations to break the system, where N is chosen to be sufficiently large. In modern practice, $N > 2^{80}$ is considered sufficiently large. Because future advances in algorithms or hardware can lower this threshold, computational security is always evaluated with respect to the best known attacks at the time of analysis.

Encryption is a practice that provides computational security by concealing a message—called plaintext—from unauthorized recipients through its transformation into ciphertext [27]. Instead of transmitting the plaintext directly, the sender transmits a ciphertext. Unauthorized users without the appropriate key cannot recover

the original message, whereas an authorized receiver, who possesses that key, can decrypt the ciphertext and retrieve the plaintext. Thus, the security of an encryption scheme relies on the computational hardness of extracting the plaintext without the key.

Encryption algorithms fall into two broad classes: symmetric-key encryption and public-key encryption. The distinction lies in whether the same key is used for both encryption and decryption. In symmetric schemes (e.g., AES), a single secret key performs both functions, whereas public-key schemes employ a publicly known key for encryption and a private key for decryption. Symmetric encryption is far more efficient but suffers from the key-distribution problem. Modern systems address this by encrypting the bulk data with a high-speed symmetric cipher such as AES (Rijndael) [28] while exchanging the symmetric key with a key-encapsulation mechanism (KEM) based on public-key cryptography—for example Diffie-Hellman [29], or the post-quantum ML-KEM (CRYSTALS-Kyber) [30].

1.3.3. Security Criterion for Covert Communication

Covert communication [18], [19] or low-probability-of-detection (LPD) communication [31], [32] requires a different perspective on secrecy. It aims to hide the existence of a transmission from an eavesdropper. In scenario such as military operation in hostile region, the mere presence of a wireless signal can reveal sensitive activity. Because an adversary may detect that signal, covert systems target a low probability of detection (LPD) as a secrecy criterion. Covert communication is closely related to steganography, as both scheme aim to conceal communication from a warden. Indeed, a covert wireless link can be modelled as a form of steganography in which the cover medium is the wireless channel itself.

Table 1. Comparison of Three Security Criterions, Covert Communication, Perfect Secrecy, and Cryptography.

	Covert Communication	**Perfect Secrecy**	**Computation Security**
Main Operation Region	Physical Layer	Physical Layer	Application Layer
Objective	Concealing Signal Existence	Concealing Message Information	Concealing Message Information
Performance Metric	Probability of Detection Distortion	Secrecy Capacity	Computational Hardness

To measure the performance of covert communication, another security criterions is required rather than information-theoretic secrecy or computational security. Security formulation of steganography is studied for criterions, information-theoretic secrecy [33] and computational security [34]. However, in covert communication, It is shown that only $O(\sqrt{n})$ bits can be transmitted covertly while n symbol transmission in a binary symmetric

channel [18], [32], i.e., secrecy capacity is zero. Computational security formulation is also not easy to be applied to covert communication because of random nature of channel, which works as cover medium. Instead of such criterion, detection probability [35], [36] and distortion [13], [37] can be used to measure the performance of covert communication.

1.4. Contribution of this Dissertation

This dissertation investigates how ECC can serve as security primitives. Because maximum likelihood (ML) decoding of a random linear code is NP-hard [38], the decoding problem offers a natural source of computational hardness for primitives such as encryption modules, hash functions, and cryptographic puzzles. At the same time, certain well-structured code families admit efficient decoders, creating an asymmetry that enables practical ECC-based encryption.

Whereas most ECC-based primitives target on exponentially hard problems, some applications merely require a moderately difficult task—hard enough to impose delay or cost, but not so hard as to be infeasible. Cryptographic puzzle [39] they are designed to be solvable with a modest, typically polynomial, amount of work and have been proposed for spam deterrence [40], consensus and delayed information transmission.

The next chapter applies this concept to blockchain consensus. We design controllably difficult ECC puzzles and integrate them into a proof-of-work variant called ECCVCC. The ECC-based puzzle supplies the desired amount of work for miners while the design flexibility of ECCs allows ASIC-resistance, narrowing the performance gap between specialized hardware and general-purpose processors and thereby helping the network remain decentralization.

Chapter II. [Trust-Achieving ECC] Error-Correction Code Verifiable Computation Consensus

2.1. Motivation

Mining with application-specific integrated circuits (ASICs) has become a serious threat to the decentralization of PoW blockchains. Because block publishing is a competitive, reward-driven task, nodes equipped with ASICs earn far higher profits than those using general-purpose processors (GPPs) such as CPUs or GPUs. As competition intensifies, GPP-based nodes become unprofitable and eventually drop out, undermining the network's decentralized nature.

The performance gap is striking: a recent Bitcoin ASIC delivers roughly $\times 10^5$ better cost efficiency than a mainstream CPU. Consequently, ASIC-equipped miners now dominate Bitcoin's hash rate [41], concentrating block production in the hands of a few large operators that run expansive ASIC farms. This concentration reduces the set of independent participants, effectively re-centralizing the network and, in turn, eroding the trust that blockchains are meant to provide.

2.2. Introduction

PoW is a type of blockchain consensus protocol to protect the blockchain. In PoW-type blockchains (including Bitcoin), nodes must submit their own answers to cryptographic puzzles as PoWs to publish a new block. Attackers who are attempting to make a modification must solve the puzzle for all blocks after the target block. It is almost impossible for a single-node attacker to have such tremendous computational capabilities.

In PoW blockchain, the block-publishing process is a competitive task to win rewards. Therefore, the nodes equipped with ASIC devices are more profitable than nodes with General-purpose processors (GPPs), such as central processing units (CPUs) and graphics processing units (GPUs). The GPP nodes will be abandoned as the competition becomes fiercer due to their low profitability. Currently, nodes equipped with ASIC devices dominate the Bitcoin business [41].

The error-correction codes (ECCs) and their decoders can be utilized to reduce the performance gap. These codes have a rich history, there are numerous classes of codes available, and they have been applied to a variety of applications including McEliece cryptosystem [3] and ECC-based hash functions [42], [43], [44].

Error-Correction Codes Proof-of-Work (ECCPoW) was first proposed in [45] as an ASIC-resistant

PoW algorithm. ECCPoW applies the decoding of time-varying LDPC code as a crypto puzzle to achieve ASIC-resistance. Jung and Lee [46] implemented a software package for ECCPoW based on [45], and launched a new hardfork of Bitcoin by replacing its PoW algorithm with ECCPoW. Similarly, Kim *et al.* [47] developed a new hardfork of Ethereum based on the ECCPoW algorithm, proposing a new feature that allowed practical adjustment of block generation time. However, the simulation and analysis in [46] were insufficient to demonstrate that the block generation time of their implementation accorded with that predicted in the theory. The implementations in [46], [47] are unsustainable with respect to difficulty control, due to their reliance on a hard-coded look-up table. Moreover, [45], [46], [47] did not provide a quantitative analysis of ASIC resistance or discuss the relationship between ASIC-resistance and decentralization of blockchains.

2.3. Contribution

In this study, we propose a novel blockchain consensus algorithm called Error-Correction Code Verifiable Computation Consensus (ECCVCC) to improve the decentralization of a blockchain. The proposed ECCVCC achieves ASIC resistance through the time-varying structure of its crypto puzzle. The proposed ECCVCC includes practical and feasible difficulty control algorithm, which solves the limitations in [45], [46], [47]. We demonstrate the block generation time and difficulty control performance of proposed ECCVCC and compare to those of previous works in [45], [46], [47]. We provide a discussion on how ECCVCC can enhance decentralization of blockchain network compared to conventional ASIC-resistant PoWs. In addition, we provide a numerical simulation to illustrate the relationship between the ASIC resistance and the degree of decentralization in a blockchain network.

Our contributions are as follows:

- **Error-Correction Code Verifiable Computation Consensus.** We propose a new class of verifiable computation puzzle (VCP) based on error-correction code decoding problem. Using this puzzle, we construct error-correction code verifiable computation consensus (ECCVCC), a novel PoW-style blockchain consensus protocol. The proposed ECCVCC improves blockchain decentralization through inherent ASIC resistance, while maintaining the security of conventional PoW schemes.
- **Practical and feasible difficulty adjustment algorithm.** The proposed ECCVCC achieves more practical difficulty control than [45] and at the same time, more feasible difficulty control than [46], [47].

- **Characterization of the relationship between ASIC-resistance and decentralization.** We characterize how the ASIC-resistance of a PoW algorithm can affect the decentralization of the blockchain from two perspectives: the efficiency and the price of ASICs.

2.4. Prior Arts on ASIC-Resistant Proof-of-Work

2.4.1. SHA based PoW

The PoW of Bitcoin is based on the double secure hash algorithm, which is a function that hashes the input twice using SHA256 [48]. SHA256 is a hash function that satisfies the secure hash standard formulated by NIST [49]. Hash function is defined by a map from the set of messages to the set of the hash, which satisfies *one-way* and *collision resistance*, as defined in [44].

In Bitcoin, block publishers must find a proper input of the SHA256 hash function which yields the output to satisfy a certain condition. Because of the *one-wayness* of hash function, this crypto puzzle reduces to random guessing.

2.4.2. ASIC-resistant PoWs

The hash-based PoWs including Bitcoin's can successfully achieve security and trust between distributed nodes since their cryptographic puzzle based on hash function requires multiple computation of the hash function which is almost impossible to be computed by an attacker. Here, note that the most important of the cryptographic puzzle, i.e., hash function has a fixed computation routine. It implies that the puzzle is efficiently and quickly computed using an ASIC. To combat attackers and other competitors equipped with ASICs, block publisher has no choice but to use the ASIC to solve the hash puzzle.

The performance gap between GPP and ASIC can harm the decentralization of blockchain networks, which is one of the most important properties of blockchains. Moreover, a recent ASIC device for Bitcoin is 10^5 times more efficient than popular CPUs in terms of cost efficiency (hash/sec/$). While a blockchain network can be considered decentralized if its block-publishing can be profitable for everyone, a blockchain network is considered centralized if only a few users can be profitable. If the gap between the cost efficiency of a GPP and an ASIC is larger, GPP nodes' profits become zero more quickly. In other words, the blockchain network becomes centralized.

ASIC-resistance can be achieved through various principals. Especially in the ASIC-resistant PoW,

[50] categorized them into three classes, multi-hash PoW, Memory-hard PoW, and Programmatic PoW. In this study, we categorize the type of ASIC-resistant PoW into two classes, bandwidth-hard and optimization-hard, depending on the kind of operations that causes the bottleneck on ASIC devices.

Bandwidth-hard PoW is one kind of ASIC-resistant PoW. ASICs have far higher computational efficiency than GPPs, while their memory-accessing capability is similar to GPPs. Bandwidth-hard PoW exploits this property to achieve ASIC resistance. These algorithms frequently access random pieces of memory, meaning ASIC is not very efficient for these algorithms. As a result, ASIC devices designed for these algorithms have low area efficiency, which is generally proportional to their manufacturing cost of the ASIC [51]. Ethash is an example of a bandwidth-hard PoW. Ethash [52] was developed to prevent the incorporation of ASIC devices in Ethereum. In Ethash, there is a memory structure called directed acyclic graph (DAG), where the data are randomly regenerated every 30,000 blocks. The Ethash achieves ASIC resistance by requiring access to the DAG to solve a puzzle.

An optimization-hard PoW is another approach. Here, ASIC improves its computational efficiency by implementing only the functionalities used in the application. If the application has many operation sets that may or may not be used, the computational efficiency of ASIC for such applications will decrease significantly. An example of an optimization-hard PoW is X16r. X16r [53] is proposed as an extension of X11 [54] ASIC-resistant PoW. This algorithm achieves ASIC-resistance by concatenating 16 different hash functions in unpredictable order. This unpredictability impedes the development of ASIC devices for X16r.

Programmatic PoW (ProgPoW) [55] belongs to the Hybrid class. ProgPoW has been suggested as a new anti-ASIC PoW algorithm of Ethereum to replace Ethash. Compared to Ethash, ProgPoW has advanced ASIC resistance, which is achieved by increasing bandwidth-hardness and including a random sequence of math. Moreover, the design of ProgPoW renders it more suitable for the memory structure of GPUs rather than that of CPUs or ASICs [56]. Although the Ethereum community rejected applying the algorithm to Ethereum, several blockchains (such as SERO [57]) adopted a derivation of ProgPoW as their PoW algorithm. In addition, Ravencoin switched its PoW algorithm to KAWPoW [58], which is a derivation of ProgPoW.

2.5. Review of Error-Correction Code Proof-of-Work

In this section, we remind details on ECCPoW using several original words in [45]. ECCPoW is defined as a framework that follows a sequential process:

1. Generate a parity-check matrix (PCM) **H** by ECC puzzle generator function (ECCPGF).

2. Solve the ECC puzzle for the given PCM **H**.

3. Broadcast the solution, (**n**, **c**′) with the block.

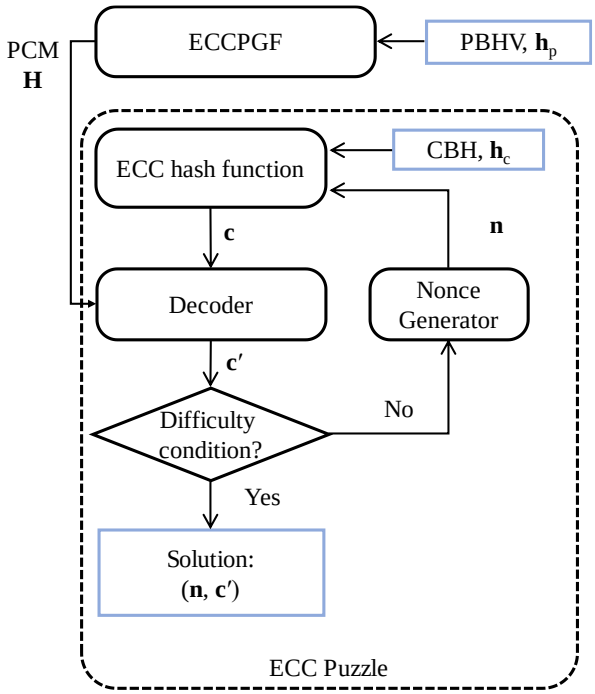


Figure 5. Scheme of ECCPoW.

The ECCPoW comprises two parts: *i*) ECC puzzle and *ii*) ECCPGF. The ECC puzzle part consists of two main modules, ECC hash function module and decoder module. The hash function module generates a hash $\mathbf{c} \in \{0,1\}^n$ with current block header (CBH), $\mathbf{h}_i$, and a nonce, $\mathbf{n} \in \{0,1\}^l$, from nonce generator. The decoder module outputs a binary word $\mathbf{c}' \in \{0,1\}^n$ from the output of hash module, **c**, using the message-passing algorithm [59]. In this process, a PCM $\mathbf{H} \in \{0,1\}^{m \times n}$ is inputted to determine the relationship between **c** and **c**′. After that, the output of the decoder, **c**′, is tested by the difficulty condition, which is designed to force nodes to run the decoder multiple times with a different nonce.

The puzzle generator part constructs a PCM from previous block hash values. Since the PCM determines the circuit of the decoder in the puzzle part, the puzzle also varies block by block. The puzzle generation is unpredictable but deterministic. **H** is generated based on the method of Gallager [2] with given parameters of code length n, row degree w_r, column degree w_c, and the previous block hash value (PBHV), $\mathbf{h}_p$. In other words, first generate a m/w_c-by-n binary submatrix **A** whose number of ones in a row is w_r, and then concatenate it with its column-permuted matrices to generate m-by-n PCM **H**. Note that the code parameters n, m, w_r, and w_c are chosen so that $n/w_r = m/w_c$, and the common value is an integer. In the remainder of this chapter, every division is treated as exact unless

an explicit rounding or flooring operator is specified. The order of column permutation is determined by a pseudo-random number generator (PRNG) whose seed is derived from the $\mathbf{h}_p$. The output of PRNG is deterministic given the seed and unpredictable without the seed. It should be noted that the PCM includes all information for decoding an ECC. This implies that if the decoding of an ECC is used as part of a puzzle for VCC, the construction procedure of PCM can be considered puzzle generation. Thus, the given PCM construction method can be defined as a puzzle generator, which generates time-varying cryptographic puzzles block by block.

2.6. Verifiable Computation Consensus

In this section, we aim to define the VCC using the concept of a verifiable computation puzzle (VCP). First, we define the VCP, and then we formally define VCC as a combination of solving routine for the VCP and its corresponding verifying routine.

2.6.1. Definition of Verifiable Computation Puzzle

The VCP is the most important component of the VCC protocol, which dominates the difficulty, block time distribution, and ASIC resistance of a blockchain. The potential block publishers spend their computational power to solve the VCP faster than others. The computational power secures the blockchain against modification of block contents. To fulfill this role, VCP for a blockchain consensus protocol must include both a solving routine and verifying routine to guarantee that a certain amount of computation is required to publish a block.

In this subsection, we aim to define VCP. First, a crypto puzzle can be defined as follows:

Definition 1. Crypto Puzzle. A crypto puzzle Π is defined by a pair consisting of the simplest solving routine and a simple verifying routine, i.e.,

$$\Pi := (\mathsf{Solve}_{\Pi}, \mathsf{Verify}_{\Pi}), \tag{10}$$

where $\mathsf{Solve}_{\Pi}: \mathbf{q} \mapsto \mathbf{a}$ is the simplest routine that outputs an answer $\mathbf{a}$ from an input query $\mathbf{q}$, and $\mathsf{Verify}_{\Pi}: (\mathbf{q}, \mathbf{a}) \mapsto \{\texttt{true}, \texttt{false}\}$ is a simple routine that verifies whether $\mathbf{a}$ is a legitimate solution for $\mathbf{q}$ or not. Each of $\mathbf{q}$ and $\mathbf{a}$ is a structured vector composed of feature elements, which will be exactly specified in Algorithm 4. Thus, the following must hold:

$$\mathsf{Verify}_{\Pi}(\mathbf{q}_i, \mathsf{Solve}_{\Pi}(\mathbf{q}_j)) = \begin{cases} \texttt{true}, & \text{if } i = j, \\ \texttt{false}, & \text{otherwise.} \end{cases} \tag{11}$$

Note that any crypto puzzle satisfies Definition 1 can be applied to PoW. However, a practical blockchain has additional requirements on crypto puzzle to guarantee robustness, fairness, and stability of block publishing. The crypto puzzle satisfying the following properties is defined as VCP. In the remainder of this section, we define three necessary properties of VCP, *i)* simple verification, *ii)* non-reusable answer, and *iii)* adjustable difficulty properties. The simple verification property in Definition 2 guarantees that the block is published as a result of a competition of multiple computation devices, while preventing denial-of-service (DoS) attacks with spamming invalid blocks. The non-reusable answer property in Definition 3 guarantees the fairness of the consensus. Without the property, a node who stores the previous blocks and the corresponding solutions can bypass the current PoW more easily. Finally, adjustable difficulty property in Definition 4 enables the block generation time to remain stable under the fluctuation of the total computation capacity of the network, which occurs through the improvement of the computing device and changes in the number of nodes.

Definition 2. Simple Verification Property. A crypto puzzle Π has simple verification property if average-case running time for computing Solve_{Π} is much larger than that for computing Verify_{Π}, i.e.,

$$T_{\mathsf{Solve}_{\Pi}} \gg T_{\mathsf{Verify}_{\Pi}}, \tag{12}$$

where T_{A} denotes the expected running time of algorithm A averaged over all possible inputs.

Definition 3. Non-reusable Answer Property. A crypto puzzle Π has non-reusable answer property if previous knowledge on $(\mathbf{q}_1, \mathbf{a}_1), (\mathbf{q}_2, \mathbf{a}_2), \ldots, (\mathbf{q}_{i-1}, \mathbf{a}_{i-1})$ is not helpful to find $\mathbf{a}_i$ without actually compute $\mathsf{Solve}_{\Pi}(\mathbf{q}_i)$. One can define an ϵ-non-reusable answer property by using mutual information, $I(\cdot;\cdot)$, as follows. A crypto puzzle Π is said to have ϵ-non-reusable answer property if, for a given k, there exist a small positive constant ϵ satisfying the following:

$$I(\mathbf{a}_k; \mathbf{a}_1, \ldots, \mathbf{a}_{k-1}, \mathbf{q}_1, \ldots \mathbf{q}_{k-1}) \leq \epsilon, \tag{13}$$

where $I(\cdot;\cdot)$ denotes mutual information.

Definition 4. Adjustable Difficulty Property. A VCP, Π, has adjustable difficulty property if there exist a set of parameters $\mathcal{D} = \{d_1, d_2, \ldots, d_{\max}\}$ which satisfies:

$$T_{\mathsf{Solve}_{\Pi}}(d_1) < T_{\mathsf{Solve}_{\Pi}}(d_2) < \cdots < T_{\mathsf{Solve}_{\Pi}}(d_{\max}), \tag{14}$$

where $d_1 < d_2 < \cdots < d_{\max}$. Such a parameter $d \in \mathcal{D}$ is called difficulty.

Using these properties, one can define the VCP as follows:

Definition 5. Verifiable Computation Puzzle. The VCP is defined as a crypto puzzle satisfies the three properties above, i.e., $\Pi = (\mathsf{Solve}_{\Pi}, \mathsf{Verify}_{\Pi}, \mathcal{D})$ is a VCP if Π satisfy (11)-(14).

One can check whether the crypto puzzles in existing PoWs satisfy the three properties or not. Hash puzzle of Bitcoin's PoW can be a simple example. Let the puzzle be Π_{B} and double SHA256 function be SHA256d. Then the solving and verifying routines for Π_{B} can be expressed as Algorithm 1 and 2, respectively. Note that the complexity of SHA256d dominates the total complexity of solving and verifying routines. Thus, by ignoring the complexity of any other operations except SHA256d, the time complexity of $\mathsf{Solve}_{\Pi_{\mathrm{B}}}$ becomes dT_{SHA256d}, while the time complexity of $\mathsf{Verify}_{\Pi_{\mathrm{B}}}$ becomes T_{SHA256}. Then the inequality (12) holds for $d \gg 1$. In other words, the simple verification property is satisfied. The non-reusable answer property is also satisfied thanks to collision resistance of SHA256d hash function. Since $\mathbf{h}_{\mathrm{c}}$ for each block contains unique values such as block height and timestamp, the $\mathbf{h}_{\mathrm{c}}$ is unique for each block, i.e., $\mathbf{h}_{\mathrm{c},i} \neq \mathbf{h}_{\mathrm{c},j}$ for all $i \neq j$. Then, by assuming independence of SHA256d, the input and output for i-th block, i.e.,

$\mathbf{w}_i$ = SHA256d($\mathbf{h}_{\mathrm{c},i}$, $\mathbf{n}_i$) would be non-informative to compute $\mathbf{w}_j$ = SHA256d($\mathbf{h}_{\mathrm{c},j}$, $\mathbf{n}_j$), for all $i \neq j$. In other words, (13) holds with small ϵ. Finally, the adjustable difficulty property is satisfied since $T_{\mathsf{Solve}_{\mathrm{B}}}(d) = dT_{\mathrm{SHA256d}}$ is a direct consequence of (14).

Algorithm 1. Pseudo codes of the solving routine of Π_{B}, $\mathsf{Solve}_{\Pi_{\mathrm{B}}}$.

Inputs:	$\mathbf{h}_{\mathrm{c}}$, $d \in \mathcal{D}$
1:	$\mathbf{n} \leftarrow 0$
2:	`do:`
3:	$\mathbf{w} \leftarrow$SHA256d($\mathbf{h}_{\mathrm{c}}$, $\mathbf{n}$)
4:	$\mathbf{n} \leftarrow \mathbf{n} + 1$
5:	`while` $\mathbf{w} > 2^{256}/d$
6:	`Return` ($\mathbf{w}$, $\mathbf{n}$)

Algorithm 2. Pseudo codes of the verifying routine of Π_{B}, $\mathsf{Verify}_{\Pi_{\mathrm{B}}}$.

Inputs:	$\mathbf{h}_{\mathrm{c}}$, $\mathbf{w}$, $\mathbf{n}$, $d \in \mathcal{D}$
1:	`if` $\mathbf{w}$ = SHA256d($\mathbf{h}_{\mathrm{c}}$, $\mathbf{n}$) `and` $\mathbf{w} \leq 2^{256}/d$:
2:	`return true`
3:	`return false`

2.6.2. Verifiable Computation Consensus

Blockchain consensus protocol is defined as a protocol that allows blockchain nodes to reach an agreement on which of potential blocks is the canonical block. VCC is a consensus protocol which can include

any of VCP. By inheriting the spirit of PoW, VCC aims to select the block includes the greatest computational cost. According to this criteria, the first block who contains the valid answer for an VCP will be selected. The validity of the answer is easily verifiable thanks to the simple verification property of the VCP.

Algorithm 3 describes the pseudo code of VCC. Let $\mathbf{b}_1$, $\mathbf{b}_2$, … be the potential blocks received by a node, and let $(\mathbf{q}_1, \mathbf{a}_1)$, $(\mathbf{q}_2, \mathbf{a}_2)$, … be the corresponding inputs to the Verify. For a given VCP and difficulty d, the VCC is defined as a protocol in which the node selects the canonical block $\mathbf{b}_C$ from the potential blocks by using Verify, and the chain-selection rule, denoted chainSelection. The chain selection rule enables nodes to determine the canonical chain when receiving different forks. Notable rules include Nakamoto's longest-chain rule [16] and heaviest-chain rule, such as the greedy heaviest-observed sub-tree (GHOST) [60].

In our Ethereum-based implementation [61], we have adopted the modified longest-chain rule used in Go-Ethereum version 1.11 [62]. The canonical chain is determined to be the chain with the highest cumulative difficulty among all forks. With a fixed difficulty, the rule is identical to Nakamoto's original longest-chain rule. Xia *et al.* [63] demonstrated that at the target BGT used in [61], the longest-chain rule resists block withholding attacks, including double-spending and selfish mining, almost as effectively as GHOST

Algorithm 3. Pseudo codes of Verifiable Computation Consensus

Inputs:	$\mathbf{b}_1, \ldots, \mathbf{b}_k, (\mathbf{q}_1, \mathbf{a}_1), \ldots, (\mathbf{q}_k, \mathbf{a}_k), d \in \mathcal{D}$
1:	$\mathcal{V} \leftarrow \{\}$
2:	`do`:
3:	`if` Verify$(\mathbf{q}_i, \mathbf{a}_i)$:
4:	$\mathcal{V} \leftarrow \mathcal{V} \cap \mathbf{b}(i)$
5:	`while` $i \leq k$
6:	$\mathbf{b}_C \leftarrow$ chainSelection$(\mathcal{V})$
7:	`return` $\mathbf{b}_C$

2.7. Error-Correction Code Verifiable Computation Consensus

In this section, we propose ECCVCC, a type of VCC in which the VCP is constructed by computation problems related to error-correction codes. ECCPoW [45] can be a special case of ECCVCC. It utilizes an iterative codeword decoding problem of LDPC codes as its VCP. Beyond this example, there are various error-correction code problems that can serve as VCPs, thanks to the extensive development within the field. By selecting different ECC-related VCPs, one can flexibly design a diverse range of ECCVCC variants. To demonstrate the flexibility of VCC design, we introduce a syndrome decoder based ECCVCC.

2.7.1. Error-Correction Code Verifiable Computation Consensus with Syndrome Decoder

The iterative syndrome decoding procedure can be utilized to construct an VCP. The syndrome decoding is one kind of ECC decoding method, which takes syndrome as an input and aim to find the error patterns. The objective of syndrome decoding for a given PCM $\mathbf{H} \in \{0,1\}^{m \times n}$ and a syndrome $\mathbf{s} \in \{0,1\}^{m}$ is finding an error pattern $\mathbf{e} \in \{0,1\}^{n}$ whose Hamming weight is smaller than a parameter t, i.e., finding $\mathbf{e}$ which satisfies:

$$\mathbf{He} = \mathbf{s} \text{ and } \|\mathbf{e}\|_H < t. \tag{15}$$

where $\|\cdot\|_H$ denotes the Hamming weight. The syndrome decoding problem of random linear code can be expressed as a short integer solution (SIS) problem in lattice-based cryptography, which enjoys one-wayness and worst-case hardness [64].

By using the syndrome decoding problem, one can construct a VCP, Π_{SD}, and the corresponding ECCVCC, named by ECCVCC-SD. The Π_{SD} is defined by a combination of solving routine, $\mathsf{Solve}_{\Pi_{SD}}$, verifying routine, $\mathsf{Verify}_{\Pi_{SD}}$, and the difficulty level $\mathcal{D}$. In $\mathsf{Solve}_{\Pi_{SD}}$ described in Algorithm 4, $\mathbf{q}$:= ($\mathbf{h}_c$, $\mathbf{h}_p$, n, w_r, w_c) and $\mathbf{a}$:= ($\mathbf{H}$, $\mathbf{s}$, $\mathbf{e}$, $\mathbf{n}$) are defined as the input query and the output answer, respectively. $\mathbf{h}_c$ and $\mathbf{n}$ are hashed using a hash function $\mathsf{Hash}(\cdot)$, which can be modeled as a random oracle, to generate a m-dimensional binary vector $\mathbf{s}$. Note that $\mathsf{Hash}(\cdot)$ denotes a cryptographically secure hash function chosen to suit implementation-specific requirements. For example, it may be instantiated as SHA256d or Keccak256. Then the iterative syndrome decoder tries to find a sparse error vector $\mathbf{e}$, considering $\mathbf{s}$ as a syndrome for an LDPC code with given $\mathbf{H}$. Here, $\mathbf{H}$ is generated by Gallager's regular LDPC parity-check matrix construction method [2], using the given parameters $w_r > w_c > 1$ and $\mathbf{h}_p$. Note that $\mathbf{h}_p$ is a hash value of the previous block, which includes information on transactions, chain, and its publisher. Because $\mathbf{h}_p$ is not fixed until the preceding $(i-1)$-th block is published, the resulting $\mathbf{H}$ remains unpredictable in advance. Thus, one can reasonably assume that GallagerPCM is a random oracle that samples uniformly form the set of all possible Gallager-structured PCMs [2]. The detailed procedure to generate $\mathbf{H}$ can be found in [45].

Focus on Step 5 in Algorithm 4. Note that $\mathbf{s}$ can either lie within or outside of the range of $\mathbf{H}$, as $\mathbf{s}$ is generated by the hash function. In general, there is no simple method to verify whether $\mathbf{s}$ is in the range of $\mathbf{H}$. However, due to the special structure of $\mathbf{H}$, there are some trivial cases where one can easily determine that $\mathbf{s}$ is not in the range of $\mathbf{H}$. Recall the structure of Gallager PCM. The $\mathbf{H}$ consists of w_c submatrices, each with dimensions of m/w_c-by-n. In each submatrix, the sum of rows forms an 1-vector. Therefore, if $\mathbf{s} = [s_0, s_1, \ldots, s_m]^T$

is obtained by **He**, then the sum from s_0 to $s_{m/wc-1}$, is equal to the sum from $s_{m/wc}$ to $s_{2m/wc-1}$. In other words, the following has to be satisfied:

$$\sum_{k=0}^{m/w_c-1} s_k = \sum_{k=w_c}^{2m/w_c-1} s_k = \ldots = \sum_{k=m-w_c+1}^{m-1} s_k \ . \tag{16}$$

Note that the probability that an uniform-randomly generated **s** satisfies (16) is 2^{-w_c+1}. Verifying whether a certain **s** satisfies this condition can be performed with small computations, fewer than m operations. This additional computation is far simpler than computing $(2^{wc-1}-1)$ more $\mathrm{SD}_{\mathrm{MP}}$ functions in step 7, which requires approximately $O(mn)$ operations. Thus, including the step 5 reduces the average number of operations required for computing $\mathsf{Solve}_{\Pi_{\mathrm{SD}}}$.

Algorithm 4. Pseudo codes of the solving routine of Π_{SD}, $\mathsf{Solve}_{\Pi_{\mathrm{SD}}}$.

Inputs:	$\mathbf{h}_c$, $\mathbf{h}_p$, n, w_r, w_c
1:	$\mathbf{n} \leftarrow 0$
2:	$\mathbf{H} \leftarrow$ GallagerPCM(n, w_r, w_c, $\mathbf{h}_p$)
3:	`do:`
4:	$\mathbf{s} \leftarrow$Hash($\mathbf{h}_c$, $\mathbf{n}$)
5:	`if` (16) does not hold:
6:	`go to` step 4
7:	$\mathbf{e} \leftarrow \mathrm{SD}_{\mathrm{MP}}(\mathbf{s}, \mathbf{H})$
8:	$\mathbf{n} \leftarrow \mathbf{n} + 1$
9:	`while` $\mathbf{He} \neq \mathbf{s}$
10:	`return` ($\mathbf{H}$, $\mathbf{s}$, $\mathbf{e}$, $\mathbf{n}$)

Algorithm 5. Pseudo codes of the verifying routine of Π_{SD}, $\mathsf{Verify}_{\Pi_{\mathrm{SD}}}$.

Inputs:	$\mathbf{h}_c$, $\mathbf{h}_p$, n, w_r, w_c, $\mathbf{H}$, $\mathbf{s}$, $\mathbf{e}$, $\mathbf{n}$
1:	$\mathbf{H'} \leftarrow$ GallagerPCM(n, w_r, w_c, $\mathbf{h}_p$)
2:	$\mathbf{s'} \leftarrow$Hash($\mathbf{h}_c$, $\mathbf{n}$)
3:	$\mathbf{e'} \leftarrow \mathrm{SD}_{\mathrm{MP}}(\mathbf{s}, \mathbf{H})$
4:	`if` (16) holds:
5:	`If` $\mathbf{He}=\mathbf{s}$ `and` ($\mathbf{H'}$, $\mathbf{s'}$, $\mathbf{e'}$) = ($\mathbf{H}$, $\mathbf{s}$, $\mathbf{e}$):
6:	`return true`

The iterative syndrome decoding function, $\mathrm{SD}_{\mathrm{MP}}$, is the most expensive computation in the routine. $\mathrm{SD}_{\mathrm{MP}}$ performs a message passing on factor graph consist of n variable nodes and $m = n \cdot w_r/w_c$ check nodes. For example, for sum-product message passing decoder, the check node (CN) message from i-th check node to j-th variable node, α_{ij} is computed by

$$\alpha_{ij} = -\zeta_i \tanh^{-1}\left(\frac{1}{2}\sum_{j'\in\mathcal{N}(i)\setminus j}\tanh^{-1}\left(\frac{|\beta_{j'i}|}{2}\right)\right)\prod_{j'\in\mathcal{N}(i)\setminus j}\text{sign}\left(\beta_{j'i}\right). \quad (17)$$

The variable node (VN) message from j-th variable node to i-th check node is computed by

$$\beta_{ji} = \gamma_j + \sum_{i'\in\mathcal{N}(j)\setminus\mathcal{I}\setminus i}\alpha_{i'j}, \quad (18)$$

where $\mathcal{N}(i)$ and $\mathcal{N}(j)$ be the set of node indices connected to i-check node and j-th variable node, γ_j be the pre-defined initial message for j-th element of **e**, and $\zeta_i \in \{-1, 1\}$ is the sign of i-th element of **s**. The decoder iteratively updates the message until convergence. Note that **H** is generated block-by-block, based on the previous block. Since **H** determines the connectivity between check and variable nodes, $\mathcal{N}(i)$ and $\mathcal{N}(j)$, the decoder routine (17)-(18) is hard to be optimized by hard-wired ASIC. An ASIC manufacturer shall contain more flexible components on the ASIC, which degrades the performance and efficiency of the ASIC.

2.7.2. Compliance of Syndrome Decoding Puzzle with VCP Property

In this subsection, we aim to check whether this puzzle is a VCP or not. Recall the definition of the VCP in **Definition 5**. According to the definition, Π_{SD} be a VCP if it has three properties, 1) simple verification property, 2) non-reusable answer property, and 3) adjustable difficulty property.

A. Simple Verification Property

The puzzle Π_{SD} has the simple verification property. One can prove it by comparing T_{Solve} and T_{Verify}. The average-case running time for the solving routine is lower-bounded as follows:

$$T_{\text{Solve}} \geq T_{\text{GP}} + 2^{n(w_c/w_r - H(\delta/2))}T_{\text{Loop}}, \quad (19)$$

where T_{GP} and T_{Loop} are the average-case running time for computing GallagerPCM and a main loop in Steps 3-9 in Algorithm 4, respectively, δ is the relative minimum distance of the code and $H(\cdot)$ denotes information theoretic entropy. The value $2^{n(w_c/w_r - H(\delta/2))}$ is derived from how many times that the main loop has to be repeated. The number of repetition is an inverse of decoding success (DS) probability, p. Note that a bound for p is given in [45], as follows:

$$2^{-nw_c/w_r} \leq p \leq 2^{-n\left(w_c/w_r - H(\delta/2)\right)}. \quad (20)$$

Similarly, T_{Verify} is found as follows:

$$T_{\text{Verify}} = T_{\text{GP}} + T_{\text{Loop}}. \quad (21)$$

Equations (19) and (21) imply that Π_{SD} satisfies (12), i.e., the Π_{SD} has simple verification property.

B. Non-Reusable Answer Property

The puzzle Π_{SD} has the non-reusable answer property. The property can be shown by finding the mutual information given in (13) is smaller than ϵ. It implies that the previous query-answer set is not informative to find current answer, i.e., the answer is non-reusable.

The mutual information in (13) is represented as difference between two entropies, as follows:

$$H(\mathbf{a}_k) - H(\mathbf{a}_k \mid \mathbf{a}_1, ..., \mathbf{a}_{k-1}, \mathbf{q}_1, ...\mathbf{q}_{k-1}) \leq \epsilon \,. \tag{22}$$

Recall that the answer of Π_{SD} is given as $\mathbf{a}$=($\mathbf{H}$, $\mathbf{s}$, $\mathbf{e}$, $\mathbf{n}$) and the query is $\mathbf{q}$ =($\mathbf{h}_c$, $\mathbf{h}_p$, n, w_r, w_c) in Algorithm 4. Let $\mathbf{h}_c$ is unique and $\mathbf{h}_p$ is also unique. Assume hash function is a random oracle.

To show the non-reusable answer property, we consider an attacker who aims to guess $\mathbf{a}$ without executing SD_{MP}. For simplicity, start with k = 2. Let an attacker compute $\mathbf{H}_2$ and $\mathbf{s}_2$ with the regular routine in Algorithm 4, which is deterministic. For this attacker, $\mathbf{e}_2$ will be a uniform random variable over the set of all possible $\mathbf{e}$. The distribution will not change by conditioning on $\mathbf{H}_2$, $\mathbf{H}_1$, $\mathbf{s}_2$, and $\mathbf{s}_1$, unless there is a Collision event. The Collision event is defined by an event that $\mathbf{H}_2 = \mathbf{H}_1$ and $\mathbf{s}_2 = \mathbf{s}_1$. Thus, the following holds:

$$H(\mathbf{a}_2) - H(\mathbf{a}_2 \mid \mathbf{a}_1, \mathbf{q}_1) = \Pr\{\mathsf{Collision}\} \log_2 |\mathcal{E}| \,, \tag{23}$$

where $\mathcal{E}$ is the set of all possible $\mathbf{e}$ and $\Pr\{\cdot\}$ represents probability that the event happens. For fixed n, w_r and w_c, the probability that randomly selected $\mathbf{H}_2$ equals $\mathbf{H}_1$ is

$$\Pr\{\mathbf{H}_2 = \mathbf{H}_1\} = \left(\frac{(w_r!)^{n/w_r}}{n!} \right)^{n/w_r - 1} , \tag{24}$$

and the probability that randomly selected $\mathbf{s}_2$ equals $\mathbf{s}_1$ is $2^{-nw_c/w_r}$. Note that $\mathbf{H}$ and $\mathbf{s}$ are independently generated. Thus,

$$\Pr\{\mathbf{H}_2 = \mathbf{H}_1 \; and \; \mathbf{s}_2 = \mathbf{s}_1\} = \Pr\{\mathbf{H}_2 = \mathbf{H}_1\} \Pr\{\mathbf{s}_2 = \mathbf{s}_1\} \,. \tag{25}$$

Combining these two probabilities, one can derive the probability that the event Collision occurs as follows:

$$\Pr\{\mathsf{Collision}\} = 2^{-nw_c/w_r} \left(\frac{(w_r!)^{n/w_r}}{n!} \right)^{n/w_r - 1} . \tag{26}$$

Because

$$\Pr\{\text{Collision}\} \le 2^{-nw_c/w_r}\left(n/w_r\right)^{-n+w_r}, \tag{27}$$

and $2^{-nw_c/w_r}\left(n/w_r\right)^{-n+w_r} = O(e^{-n\log n})$, the upper-bound is negligible for large $n > w_r$. Hence

$$\lim_{n\to\infty}\Pr\{\text{Collision}\} = 0\,. \tag{28}$$

Now, we generalize the Collision probability for $k > 2$. Without loss of generality, k-Collision can be defined by an event that at least one of the previous $(k-1)$-pairs is identical to the current pair. The k-Collision probability for $k > 2$ become:

$$\Pr\{k\text{-Collision}\} = 1-\left(1-\Pr\{\text{Collision}\}\right)^{k-1}, \tag{29}$$

which is negligibly small for a finite constant k and for a sufficiently large n. Finally, one can conclude that Π_{SD} has ϵ-non-reusable answer property with

$$\epsilon \ge \Pr\{\text{Collision}\}\log|\mathcal{E}|\,. \tag{30}$$

C. Adjustable Difficulty Property

The puzzle Π_{SD} has the adjustable difficulty property. Assuming that w_r and w_c are constant, (19) is a direct consequence of (14), by letting $d = n$.

2.7.3. Security Comparison

A codeword decoder is another type of decoder that solves the dual problem to syndrome decoding. Aforementioned syndrome decoder takes syndrome as input and find the error pattern. In contrast, the codeword decoder takes a codeword contaminated by errors as input and aims to recover the error-free codeword. The codeword decoder was used to construct a VCP in previous works [45], [46], [47]. For the remainder of this chapter, we refer to this VCP as Π_{CD}.

In this subsection, we compare the VCP of ECCVCC-SD, Π_{SD}, and the Π_{CD} in [45], [46], [47] in terms of simple verification property. By comparing the degree of simplicity of verification, one can determine how much the VCP secure against DoS attack. For this objective, we define verification efficiency of a puzzle Π, VE_{Π}, as the ratio between the average-case running times of its solving routine and verifying routine. i.e,

A. Verification Efficiency of Syndrome Decoding Puzzle

The complexity of $\text{Solve}_{\Pi_{SD}}$ is represented as follows:

$$T_{\mathsf{Solve}_{\Pi_{\mathsf{SD}}}} = T_{\mathsf{GP}} + \frac{1}{p}\left(2^{w_c-1}T_{\mathsf{Hash}} + T_{\mathsf{Decoder}} + 2^{w_c-1}m + m^2 n\right), \tag{31}$$

where T_{Hash} and T_{Decoder} are average-case running time for computing $\mathsf{Hash}(\mathbf{h}_i, \mathbf{n})$ and $\mathsf{SD}_{\mathsf{MP}}(\mathbf{s}, \mathbf{H})$, respectively. The complexities for computing **He** and checking (20) are approximated as $m^2 n$ and m, respectively. On the other hand, the complexity of $\mathsf{Verify}_{\Pi_{\mathsf{SD}}}$ becomes:

$$T_{\mathsf{Verify}_{\Pi_{\mathsf{SD}}}} \le T_{\mathsf{GP}} + \left(T_{\mathsf{Hash}} + T_{\mathsf{Decoder}} + m + m^2 n\right). \tag{32}$$

Finally, the verification efficiency of Π_{SD} is obtained as follows:

$$VE_{\Pi_{\mathsf{SD}}} = \frac{1}{p}\frac{pT_{\mathsf{GP}} + 2^{w_c-1}T_{\mathsf{Hash}} + T_{\mathsf{Decoder}} + 2^{w_c-1}m + m^2 n}{T_{\mathsf{GP}} + T_{\mathsf{Hash}} + T_{\mathsf{Decoder}} + m + m^2 n}. \tag{33}$$

B. Comparison of Verification Efficiency

For fair comparison, we assume the decoder in [45], [46], [47] is an iterative codeword decoder, denoted as $\mathrm{CD}_{\mathrm{MP}}(\mathbf{c}, \mathbf{H})$. Both $\mathrm{CD}_{\mathrm{MP}}(\mathbf{c}, \mathbf{H})$ and $\mathrm{SD}_{\mathrm{MP}}(\mathbf{s}, \mathbf{H})$ perform the same VN and CN updates as described in (18) and (17). The only difference between the two decoders is that $\mathrm{SD}_{\mathrm{MP}}$ initializes its CNs based on **s**, while $\mathrm{CD}_{\mathrm{MP}}$ initializes its VNs based on the given **c**. Under this assumption, we can reasonably equate the complexities and the corresponding DS probabilities of the two decoders. As a result, the verification efficiency of Π_{CD} is represented as follows:

$$VE_{\Pi_{\mathrm{CD}}} = \frac{1}{p}\frac{pT_{\mathsf{GP}} + T_{\mathsf{Hash}} + T_{\mathsf{Decoder}} + m^2 n}{T_{\mathsf{GP}} + T_{\mathsf{Hash}} + T_{\mathsf{Decoder}} + m^2 n}. \tag{34}$$

Then, the following inequality holds:

$$\begin{aligned} VE_{\Pi_{\mathsf{SD}}} &\ge \frac{1}{p}\frac{pT_{\mathsf{GP}} + T_{\mathsf{Hash}} + T_{\mathsf{Decoder}} + m + m^2 n}{T_{\mathsf{GP}} + T_{\mathsf{Hash}} + T_{\mathsf{Decoder}} + m + m^2 n} \\ &\ge \frac{1}{p}\frac{pT_{\mathsf{GP}} + T_{\mathsf{Hash}} + T_{\mathsf{Decoder}} + m^2 n}{T_{\mathsf{GP}} + T_{\mathsf{Hash}} + T_{\mathsf{Decoder}} + m^2 n} \\ &= VE_{\Pi_{\mathrm{CD}}}. \end{aligned} \tag{35}$$

Therefore, we conclude that Π_{SD} is more efficient than Π_{CD} in terms of verification efficiency. It implies that the proposed ECCVCC-SD is more secure against DoS attacks than the conventional ECCPoW.

2.8. Difficulty Adjustment for Error-Correction Code Verifiable Computation Consensus

In this section, we propose a novel difficulty adjustment algorithm for ECCVCC. For a VCP, there exist a parameter called difficulty which controls the block generation time of blockchain. In the proposed ECCVCC, the puzzle parameter n is directly related to the difficulty. Recall (20). The parameter n dominates the DS probability p and $1/p$ is the required number of decoding attempts for block publishing.

The previous works on ECC based VCPs has limitations on the difficulty adjustment. To compute p, [45] used an asymptotic upper-bound of δ. However, this result cannot provide an average-sense prediction of p, which is more important than the bound in practice. In other implementation works [46], [47] a difficulty table derived from a Monte-Carlo simulation of p is hard-coded for difficulty control. The difficulty table makes the block generation time (BGT) more predictable than the bounding method in [45]. However, the approach is not feasible for large-scale practical blockchain, which requires an extremely small value of p.

To overcome this problem, we introduce a novel difficulty adjustment algorithm for ECCVCC. The proposed algorithm is a hybrid of asymptotic approach in [45] and the difficulty table approach in [46], [47]. In the proposed algorithm, p is empirically measured for a feasibly small region by a Monte-Carlo simulation. Since (20) implies that p exponentially decreases over n, the exponential slope of p can be obtained by least square curve fitting on the measured values. Finally, p for the large n region is asymptotically obtained by assuming the exponential slope is constant over n. The resulting ECCVCC algorithm achieves a stable BGT distribution by using the proposed difficulty adjustment algorithm. The implementation including the difficulty adjustment algorithm can be found in [61].

2.8.1. Block Generation Time Model

Let us start from BGT model for a general PoW blockchain. Assume that the total hash rate of all nodes is given as r, and the difficulty is given as d. It is well known that the BGT for given r and d can be modeled as an exponential random variable whose rate parameter is r/d [65]. The following Definition 5 is a formal derivation of the BGT model.

Definition 6. Block Generation Time Model. Let r_i be the number of guesses for a cryptographic puzzle that the i-th node can compute in unit second. Let d be the expected number of guesses required to solve the puzzle. For $r = r_1 + r_2 + \dots + r_M$, the block generation time X follows an exponential distribution whose probability density function (pdf) is as follows:

$$f_X(x \mid r, d) = \lambda e^{-\lambda x}, \tag{36}$$

where

$$\lambda = \frac{r}{d}. \tag{37}$$

This model is derived by considering the block generation procedure as a random guessing, assuming the one-wayness of the puzzle. Let the *k*-th node trying to solve a PoW puzzle alone. Since the node makes r_i guesses in unit second, the required time to solve the puzzle, X_i, would be an exponential random variable with rate parameter $\lambda_i = r_i/d$. The BGT would be the minimum solving time among all *M* nodes. It is well known that the minimum of exponential random variables,

$$X = \min\{X_1, X_2, ..., X_M\}, \tag{38}$$

is also an exponential random variable with rate parameter $\lambda = \lambda_1 + \lambda_2 + \ldots + \lambda_M = r/d$.

Note that this model omits block-processing time and propagation delay. The block-processing time includes the time required to execute and verify transactions, while the propagation delay includes latency for transferring a block to another node. In addition, BGT may fluctuate in a special circumstance such that adversaries launch block withholding attacks. As such attacks are rare, our observation shows that the model remains valid. For example, this model is consistent with practice in Bitcoin blockchain. To show this, we first derived the cumulative distribution function (CDF) of the BGT, *X*, as follows:

$$F_X(x) = \begin{cases} 1 - e^{-\lambda x}, & \text{if } x \geq 0, \\ 0, & \text{elsewhere.} \end{cases} \tag{39}$$

In Bitcoin, the target BGT is 10 minute, i.e., $1 / \lambda = 600$. The observed behavior of Bitcoin after 2020 [66] indicates that 18%, 64%, and 87% of blocks are generated within 2, 10, and 20 minutes, respectively. This observation is consistent with the analysis in (39).

2.8.2. Relationship between Block Generation Time and Decoding Success Probability in ECCVCC

In ECCVCC, DS probability *p* dominates the difficulty of the VCP. Note that *p* can be expressed as the inverse of difficulty, *d*, assuming the time required for a decoding is constant. Then the average BGT, □(*X*) = $1/\lambda$, can be derived by substituting $d = 1/p$ into (37). For this, we first represent DS probability as a function of the puzzle parameters, *n*, w_r, and w_c

A. *Decoding Success Probability*

Analytic evaluation of p is challenging. A numerical evaluation is also not feasible, as the probability is too small to evaluate, especially in large n. Instead, we aim to derive asymptotic bounds to approximate the DS probability.

One can approximate p using equation (20). (20) tell us two key points. *i)* for fixed constants w_c and w_r, increasing the code length n causes the DS probability p to decrease. *ii)* the exponential slope of p is in a range $[-w_c/w_r, H(\delta/2)-w_c/w_r]$. Note that the lower-bound, derived from the poorest decoder, actually cannot correct any error. The actual exponent value of an iterative decoder will likely be closer to the upper-bound rather than the lower-bound. Thus, we can approximate $p \approx 2^{-n(wc/wr-H(\delta/2))}$, if decoder is properly designed.

In [45], δ_1, a theoretical upper-bound of δ, is given. However, substituting δ_1 into δ to approximate p will yield a value significantly different from the actual result, i.e.,

$$p_{\delta_1} = 2^{-n(w_c/w_r - H(\delta_1/2))}, \tag{40}$$

is much different with the actual value of p. To demonstrate this difference, we conducted a Monte-Carlo simulation for p.

The Figure 6 compares the actual value of p obtain by the Monte-Carlo simulation (x marked), p_{δ_1} (solid line), and p_{ref} (dashed line). The simulation is conducted at 7 uniform-interval points according to n, from n = 64 to n = 112. At each simulation point, the average value of p is estimated from a million times of decoding using randomly generated **s** and **H**. The exponential slope of p_{δ_1} is $w_c/w_r - H(\delta_1/2)$, but the actual p follows a smaller slope, $w_c/w_r - H(\delta_1/2) - c$. Here, we can find a reference DS probability, p_{ref}, defined in Definition 7 through a simple least square curve fitting on log-scaled observations.

Definition 7. Reference DS Probability. A reference DS probability, p_{ref}, is found by least square curve fitting with observations. i.e.,

$$p_{ref}(n, w_r, w_c) = 2^{an+b}, \tag{41}$$

where constants a and b are determined by a Monte-Carlo simulation for small n. For example, the constants determined under parameter (w_r = 8, w_c = 4) are a = −0.1076 and b = −1.6676. i.e., every time n increases by 8, the probability becomes approximately 0.55 times its previous value. By using the p_{ref}, one can successfully approximate p even for very large n which can be engaged in practice.

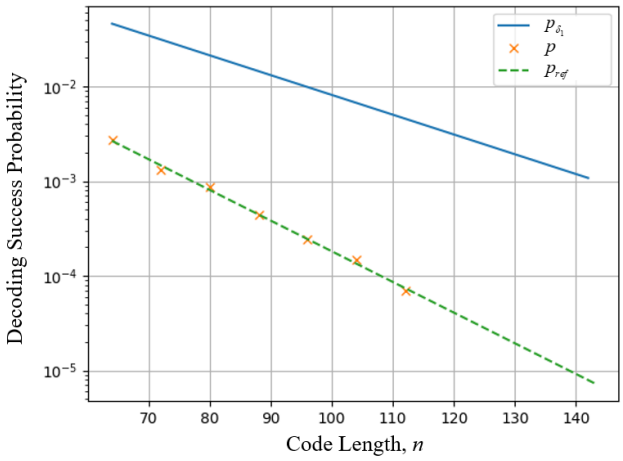


Figure 6. Theoretical DS probability, p_{δ_1}, actual DS probability, p, and p_{ref} for a certain parameter (w_r=8, w_c=4, δ_1=0.1765). The exponential slope of p_{δ_1} is −0.0694, while that of p and p_{ref} is −0.1076.

B. Block Generation Time

Now one can derive the expected block generation time, $\mathbb{E}[X]$, as a function of parameters (n, w_r, w_c) by replacing the d in (37) with $1/p_{ref}$, i.e.,

$$\mathbb{E}[X] = \frac{1}{r p_{ref}(n, w_r, w_c)}. \tag{42}$$

Equation (42) implies that difficulty of a puzzle can be controlled by adjusting n, w_r, and w_c to achieve a target BGT for a given r. However, r is generally unknown. Most PoW-based blockchains measure r from the BGTs of previous few blocks and then adjust the difficulty of the next blocks to achieve the target BGT.

2.8.3. Difficulty Adjustment Algorithm for ECCVCC

To maintain the target BGT under varying hash rate, one needs to know r and adjust d. Since the exact value r is challenging to determine publicly, most blockchains estimate r from the BGTs of the previous N blocks. In the remainder of this section, we compared two representative style of difficulty adjustment; Bitcoin and Ethereum. Then, we introduce the difficulty adjustment algorithm for ECCVCC.

A. Difficulty Adjustment of Bitcoin

In Bitcoin, difficulty of the puzzle is adjusted every N = 2016 blocks. The average BGT for the N blocks is measured and compared with the target BGT. Then, at every iN-th block, the difficulty of the next N blocks is determined by multiplying current difficulty by the ratio between the target BGT, τ_{target}, and average BGT, i.e., the difficulty of iN-th block, d_{iN}, is defined by

$$d_{iN} = d_{(i-1)N} \cdot \frac{\tau_{\text{target}} N}{\sum_{k=(i-1)N}^{iN-1} \tau_k}, \tag{43}$$

where τ_k is the realized BGT of k-th block. This algorithm is known to maintain the BGT according to the exponential model given in Definition 1 for a stable r. However, multiple studies [65], [67], [68] have pointed out that the algorithm cannot handle a steady increase or decrease of r over N blocks. Especially, these studies showed that the target BGT is not maintained when r grows exponentially.

B. *Difficulty Adjustment of Ethereum*

Ethash adjust its difficulty block-by-block, i.e., $N = 1$. At each block, the difficulty is determined by the BGT and the difficulty of the previous block. However, directly applying $N = 1$ to (43) results in unstable BGTs, since just one realization of BGT is not sufficient to estimate r accurately. To improve the stability of BGT, Ethash uses two parameters in its difficulty adjustment algorithm: sensitivity and threshold BGT. The following (44) expresses the difficulty adjustment of Ethash. The i-th block's difficulty, d_i is determined by:

$$d_i = d_{i-1}\left\{1 + S\left(U + 1 - \left\lfloor \frac{X_{i-1}}{\tau_{th}} \right\rfloor\right)\right\}, \tag{44}$$

where $\lfloor \cdot \rfloor$ is the floor function. Here $S > 0$, $\tau_{th} > 0$, and $U \in \{0,1\}$ are sensitivity, threshold BGT, and uncle indicator, respectively. S determines how sensitively the algorithm reacts on the difference between the target and actual BGT. Ethereum has used $S = 1/2048$ since its Homestead version [69]. τ_{th} affects the average BGT by determine a typical interval for BGT. Namely, d_i will not change from d_{i-1} if X_{i-1} falls into the interval $[\tau_{th}, 2\tau_{th})$, assuming $U = 0$. Ethereum used $\tau_{th} = 10$ in its Homestead version to target a BGT of 14 seconds. The parameter was updated by $\tau_{th} = 9$ to achieve a modified target BGT of 12 seconds after its Byzantium version [70]. U is a parameter introduced since Byzantium version to address hash power wasted mining uncle blocks (blocks legitimately mined but not included in canonical chain due to the chain selection rule). $U = 0$ if there is no uncle block and $U = 1$ if there are uncle blocks. Compared to the Bitcoin's, the difficulty adjustment algorithm of Ethash is robust against steady increase or decrease of r, thanks to its block-by-block adjusting mechanism.

C. *Difficulty Adjustment of ECCVCC*

In ECCVCC, we applied an Ethash-style difficulty adjustment algorithm (44) to achieve robustness against steady variation of r. The value of S is selected by considering the trade-off between stability and responsibility of BGT.

Unlike the puzzles of Bitcoin and Ethereum, the expected number of guesses required to solve Π_{SD}, $1/p$, is itself a random variable that depends on **H**, even when the difficulty parameters (n, w_r, w_c) are fixed. Thus, the effective difficulty and the corresponding BGT can fluctuate more sharply than in conventional PoW schemes. To maintain a stable BGT comparable to that of other blockchains, S may need to be set smaller to compensate for this intrinsic randomness in $1/p$.

Network size should also be taken into account when choosing S. In a blockchain network with many active participant and a high hash rate, a sudden change in hash rate hardly occurs. In this case, smaller S is sufficient. In contrast, small network requires larger S to pursue potential sharp swings in hash rate.

Finally, one can set the target BGT by modifying τ_{th}. Here, we derive the τ_{th} as a function of the target BGT. Consider a scenario where r is constant. In this scenario, the difficulty adjustment algorithm aims to maintain the current difficulty. Here, the ratio between current block difficulty, d_i, and previous block difficulty, d_{i-1} is modeled as a random variable since the BGT of previous block, τ_{i-1}, is a random variable. Then,

$$\mathbb{E}\left[d_i / d_{i-1}\right] = 1. \tag{45}$$

From (44) and (45), one can derive (46) by assuming $U = 0$.

$$\mathbb{E}\left[1 + S\left(1 - \left\lfloor \frac{X_{i-1}}{\tau_{th}} \right\rfloor\right)\right] = 1. \tag{46}$$

Then, one can conclude that

$$\mathbb{E}\left[\left\lfloor \frac{X_{i-1}}{\tau_{th}} \right\rfloor\right] = 1. \tag{47}$$

This can be expressed using the CDF of τ_{i-1}, $F(\tau)$, as follows:

$$\sum_{n=1}^{\infty} n\left\{F\left((n+1)\tau_{th} - 1\right) - F\left(n\tau_{th} - 1\right)\right\} = 1. \tag{48}$$

Recall that τ_{i-1} is an exponential random variable with mean $1/\lambda$. One can derive (49) by substituting (39) to (48) as follows:

$$e^{\lambda}\left(1 - e^{-\lambda\tau_{th}}\right)\sum_{n=1}^{\infty} n e^{-n\lambda\tau_{th}} = 1. \tag{49}$$

Finally, τ_{th} can be derived as a function of λ, as follows.

$$\tau_{th} = \frac{1}{\lambda}\ln(1 + e^{\lambda}). \tag{50}$$

For example, let the target BGT is 10. In this case, $\lambda = 0.1$. Then one can determine the corresponding $\tau_{th} = 7.444$.

Table 2 compares the difference between difficulty adjustment algorithm among ECCVCC and ECCPoW implementations. Park *et al*. [45] proposed an asymptotic upper-bound method to approximate p. Previous works [46], [47] implemented their difficulty adjustment algorithm based on hard-coded difficulty table with Bitcoin- and Ethereum-style adjustment algorithms, respectively.

Table 2. Comparison of Difficulty Adjustment Algorithm Among ECCPoW and ECCVCC Implementations.

	ECCVCC (Proposed)	Park *et al*. [45]	Jung and Lee [46]	Kim *et al.* [47]
Implementation Style	Ethereum	N/A	Bitcoin	Ethereum
Approximation Method of p	Hybrid	Asymptotic	Numerical	Numerical

D. Evaluation Results for Difficulty Adjustment

In this section, we visualize the behavior of BGT and difficulty according to various values of sensitivity parameter. We assume the BGT is a geometric random variable, which is discrete version of the exponential random variable, and there is no uncle block. We simulated the difficulty and BGT for 20,000 blocks, with difference scenarios for network computing power.

i. Constant Computing Power Scenario

First, we consider the network computing power is a constant. Under this assumption, we simulated the BGT and difficulty using sensitivity values of {1/8, 1/256, 1/1024, 1/2048}. We used $U = 0$ and $T_{th} = 10$ for this simulation. Here are the resulting difficulty flow and the histograms for BGT. In Figure 7(b), we can see that high sensitivities cause high volatility even though actual network computing power remains unchanged. Especially, sensitivity of 1/8 sometimes yields clipping to the minimum difficulty which can cause multiple consequent 1-second-blocks. We can see the effect of those blocks in Figure 8. The histogram for sensitivity = 1/8 is extremely distorted from the geometric distribution. The mean and variance of BGT in Table 3 also shows the distortion

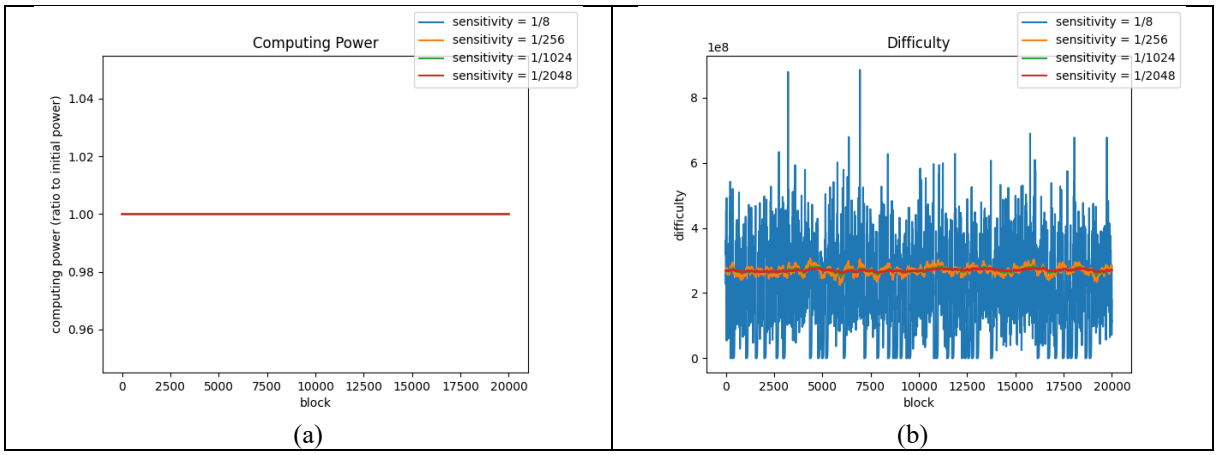


Figure 7. (a) Constant computing power scenario and (b) the corresponding difficulty.

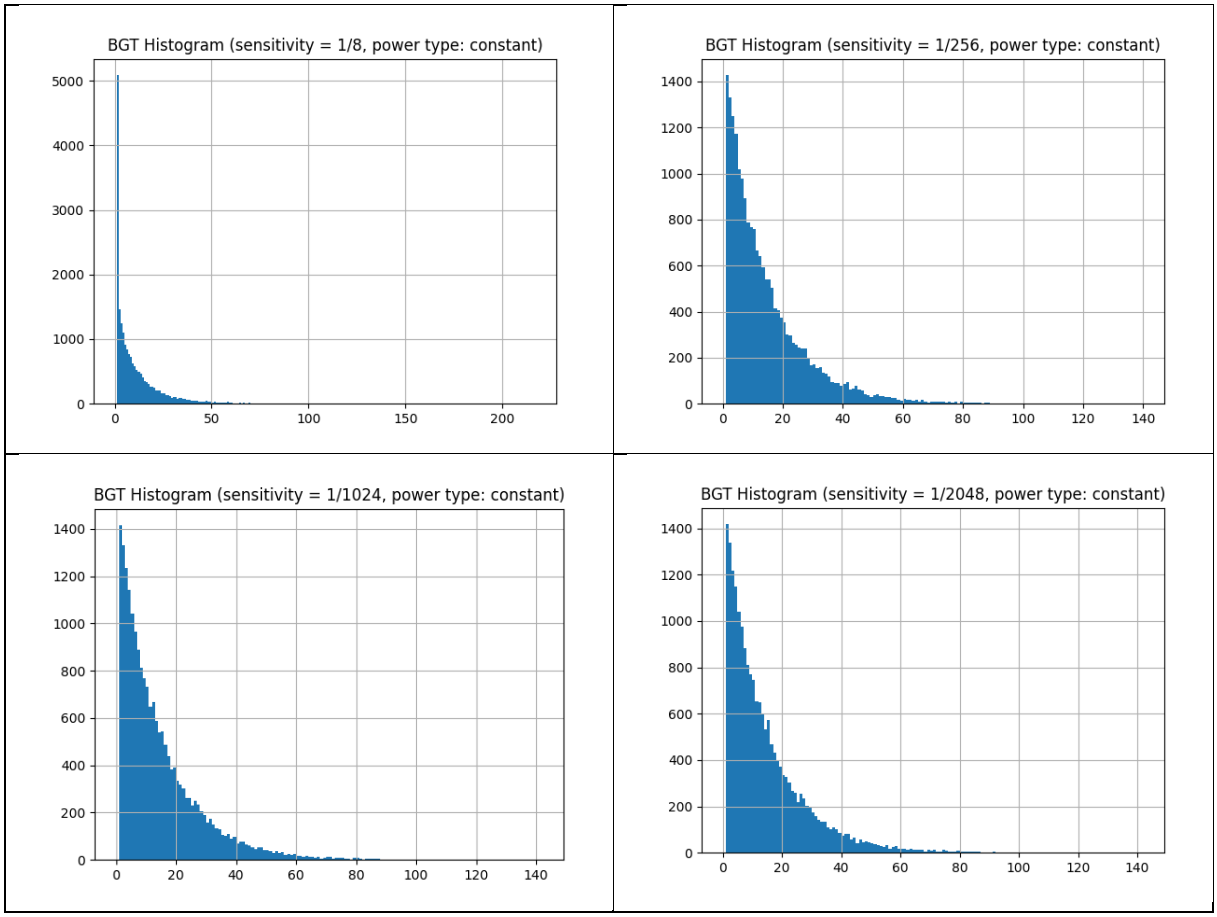


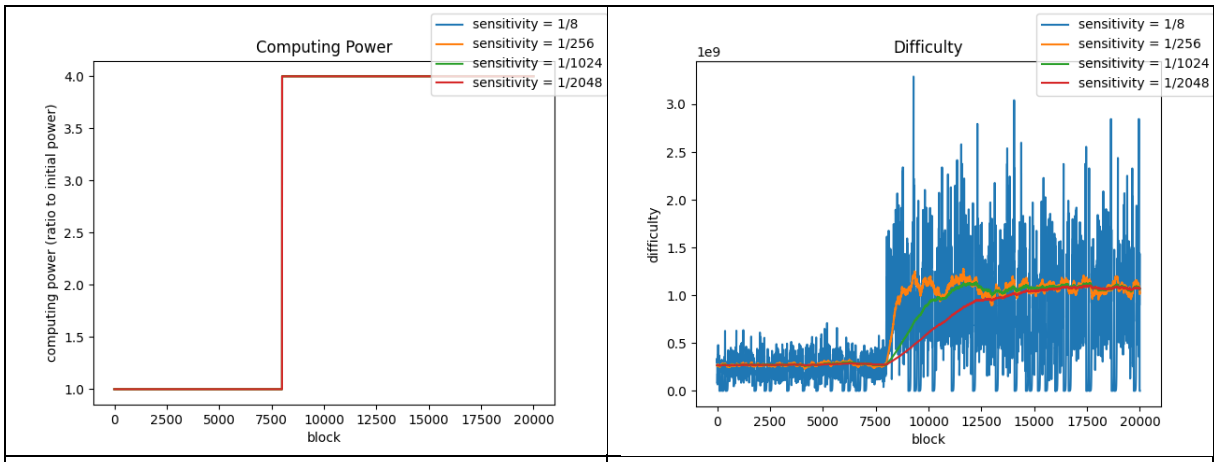
Figure 8. BGT Histograms under constant computing power scenario.

Table 3. BGT statistics according to sensitivity (constant).

Sensitivity	Mean	Variance	Maximum BGT
1/8	**10.41**	**174.11**	**218**
1/256	14.12	184.86	141
1/1024	14.18	185.86	143
1/2048	14.18	185.57	143

ii. Steep Increase of Computing Power

In this time, we consider a scenario that the network computing power is steeply increased. This scenario can happen when a heavy miner newly joins in the blockchain. We modeled that the network computing power is increased to x4 times of initial value at the 8000th block. We can see in Figure 9(b), the difficulty quickly tracks the network computing power with high sensitivity. We can also see the corresponding BGT is well maintained with high sensitivities, while BGT is severely decreased after transition point with low sensitivities. The Table *4* below shows the response time required to the difficulty catch up the change of network computing power (i.e., the first block with difficulty > 4 * initial difficulty.)



(a) (b)

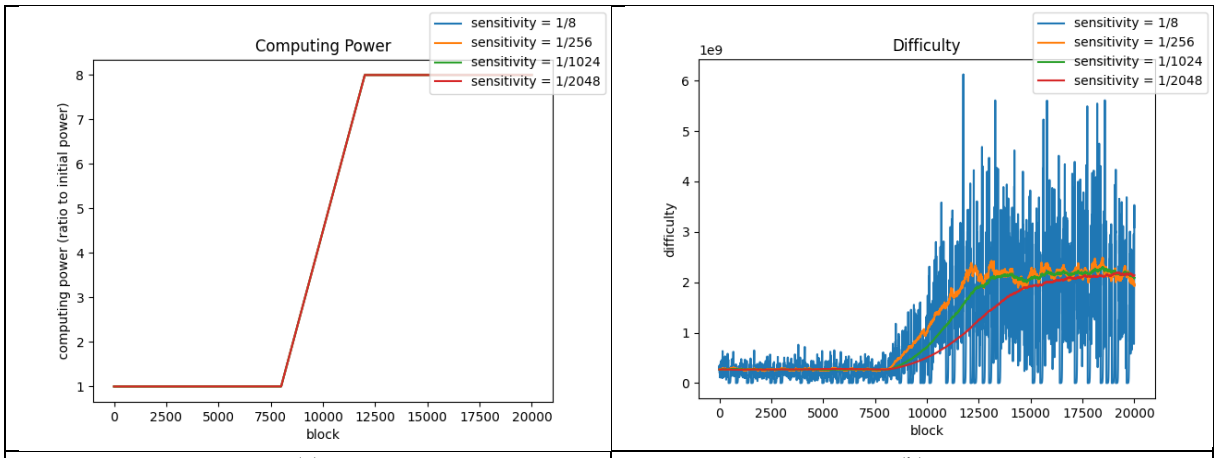
Figure 9. (a) Computing power scenario (steep increase) and (b) the corresponding difficulty.

Table 4. Difficulty response time according to sensitivity (steep increase).

Sensitivity	90% Response Time (blocks)	70.71% Response Time (blocks)	Theoretical Response Time (blocks)
1/8	12	7	12
1/256	669	365	356
1/1024	2410	1445	1421
1/2048	5174	2958	2840

iii. Hard Increase

In this time, we assumed the network computing power linearly grows over 4000 blocks and finally reached x8 times of the initial value. Note that the increase ratio per block, $10^{\frac{\log 8}{4000}}$, exceeds the maximum increase ratio for $S = 1/2048$. In Figure 10, we can see that the difficulties for S = 1/2048 and S = 1/1024 cannot reach to the hash power until the transition finished; while the difficulties for S = 1/8 and S = 1/256 immediately pursue the hash power.



(a) (b)

Figure 10. (a) Computing power scenario (hard increase) and (b) the corresponding difficulty.

Table 5. Mean BGT comparison between before/after transition (hard increase).

Sensitivity	Mean (before transition start)	Mean (after transition end)
1/8	10.40	10.10 (-2.89%)
1/256	14.09	14.13 (+0.28%)
1/1024	14.07	13.83 (-1.72%)
1/2048	14.07	**12.25 (-12.94%)**

2.9. ASIC-Resistance of ECCVCC

LDPC decoders based on ASIC (ASIC-LDPC decoders) are implemented to achieve low power consumption and fast processing. In the decoders, the check and variable nodes must be physically connected using logic gates for a given parity check matrix. Most of ASIC-LDPC decoders only support either a set of pre-defined parity check matrices or structured parity check matrices. The surveys related to existing ASIC-LDPC decoders are discussed in the following paragraph.

First, the ASIC-LDPC decoder in [71] supported quasi-cyclic parity check matrices decomposed into cyclic-shifted identity and zero matrices. These matrices have the same structure as used when implementing the decoder. Second, the ASIC-LDPC decoder in [72] supported the parity check matrices included in the IEEE 802.16e system. These matrices were fixed and did not change. Third, Hanzo *et al*. [73] reviewed state-of-the-art ASIC-LDPC decoders and stated that these decoders must take a bank of hardware to support many random parity check matrices. In other words, additional components such as memories, controllers, and switchable interconnections were required, resulting in these components occupying the largest chip area in the decoders. They supported their statement by providing an example [74], in which the ASIC-LDPC decoder supported approximately 100 parity check matrices. However, the previously mentioned additional components occupied 75% of the total area of the decoder. These examples demonstrate that there are no practical implementations of the

ASIC-LDPC decoder that support an infinite number of random parity check matrices.

There are FPGA-LDPC decoders in which the LDPC decoders are implemented on FPGA chips. Although FPGA-LDPC decoders consume more power than ASIC-LPDC decoders, they are much easier to reprogram, implying there is more flexibility in terms of designs. The FPGA-LDPC decoder presented in [75] supported parity check matrices up to n = 65,000. However, it is necessary to load a parity check matrix onto this decoder when it has to be changed, requiring additional time. In [76], Hanzo *et al*. stated that FPGA-LDPC decoders require additional routing and processing devices to support many parity check matrices. They also highlighted that the use of these additional devices could result in complex designs, incurring increased costs.

The computational efficiency of an ASIC is improved by only implementing the functionalities that are actually used in the application. To make a PoW algorithm resistant to ASICs, the performance gap between GPPs and ASICs must be reduced, i.e., the ASIC resistant PoWs must include a certain operation that ASICs cannot conduct much faster than GPPs.

Bandwidth-hard applications achieve ASIC resistance by using the bottleneck that occurs when fetching data from external memory. This type of application includes a routine for fetching data from a large dataset. Here, the dataset should be sufficiently large so that it cannot be carved in the ASICs, and the location of data in the dataset should not be predictable prior to actual computation. The hardness of bandwidth-hard applications can be measured by a function of the dataset size (fetch data size) and how often fetching occurs. For example, the dataset of the current Ethereum (Ethash) is 4.8 GB and the fetch data size is 128 bytes, and such fetching occurs 63 times per hash cycle.

Optimization-hard applications make it difficult for manufacturers to implement ASICs, although the degree of difficulty when implementing an ASIC optimized for an application is not well-formulated. Nevertheless, it is clear that as the number of variable parts in an application increases, optimization of the ASIC will become more difficult. To make ASICs for applications that have programmable parts, they must include programmable blocks, such as programmable logic devices or FPGA. However, such programmable blocks are not fast enough, and occupy more space than the non-programmable parts. As a result, the performance efficiency of the ASIC decreases. Accordingly, the hardness of optimization-hard algorithms can be measured via the size of the variable parts in the algorithms.

In ECCVCC, PCM requires variable parts. It is obvious that efficiency of ASIC is degraded as possible number of PCMs increases. In [73], ASIC-LDPC decoder which supports only one PCM can achieve very high area efficiency by using hard-wired structure without additional components; but the ASIC-LDPC decoder which

supports a hundred kinds of PCMs has limited area efficiency (almost 1/100 of single PCM decoder) because of the additional components required for the flexibility. As shown in Fig. 11 of [73], area efficiency (scaled to 65nm) of ASIC LDPC decoder that supports only one PCM is on the order of 10^5 Mbps/mm^2, while that supports hundreds PCMs is decreased to order of 10^3 Mbps/mm^2. The energy efficiency of ASIC LDPC decoder also decreases from order of 10^2 bit/nJ to order of 10^0 bit/nJ, where the number of supporting PCMs increases from one to a hundred.

For example, the variable parts in the X16r algorithm are of the order of 16 hash functions. This information can be measured as 64 bits. The proposed ECCVCC is also a type of optimization-hard algorithm. Thus the ASIC resistance of ECCVCC can be measured via the size of the variable parts. The following Corollary 1 shows how ECCVCC is ASIC-resistant compared to the X16r algorithm.

Corollary 1: ECCVCC is ASIC-resistant.

Proof: The size of variable parts in length-n X16r puzzles are $n + 64$ bits ($O(n)$). In contrast, variable part of length-n ECC puzzles is PCM of size $n \times nw_c/w_r$. This can be compressed by storing only the position of the 1s. Then the variables are $nw_c\log n$ bits ($O(n\log_2 n)$), where w_c/w_r is fixed. Thus, ECCVCC is stronger than X16r in terms of ASIC resistance.

This shows that ECCVCC is more ASIC-resistant than X16r for $n > 8$ bits. The problem size n in most of commercial PoWs is fixed by $n = 256$. In this setting, the size of variable component in ECCVCC is 19.2 times more resistant against ASIC than X16r (where $w_c = 3$ is used). Note that this performance gap will be larger since n in ECCVCC grows as total computational capacity of network increases, while that of X16r is fixed. Table 6 compares the ASIC-resistance of ECCVCC, X16r, and Bitcoin PoW. ECCVCC achieves O($n$log$n$) resistance, whereas X16r and Bitcoin provide $O(n)$ and $O(1)$, respectively.

Table 6. Comparison of ASIC-Resistance among ECCVCC, X16r and Bitcoin.

	ECCVCC (Proposed)	X16r [53]	Bitcoin [16]
ASIC Resistance	$O(n\log n)$	$O(n)$	$O(1)$

2.10. ASIC-Resistance and Decentralization of Blockchains

This section presents a discussion on the relationship between ASIC resistance and the decentralization of the blockchain. The ASIC resistance is affected by the expense and efficiency of the ASIC. The more ASIC-resistant a consensus algorithm is, ASIC for the consensus algorithm will be expensive and inefficient.

Moreover, as briefly mentioned in Section 2.1, ASIC-friendly PoW harms decentralization much more than an ASIC-resistant PoW. To demonstrate this, we conducted simplified numerical modeling and simulation

2.10.1. Simulation Model

We modeled the ASIC devices using two major parameters: initial cost and cost efficiency. The initial cost (or price) of an ASIC device, I_{ASIC}, is the cost to equip an ASIC device. The computational capacity that a single ASIC device can obtain in unit time is modeled as C_{ASIC}. The energy consumption of an ASIC is defined as E_{ASIC}. The block-publishing competition is simplified by randomly picking one of the competitors, where the probability of someone being picked is proportional to their computational capacity. The competition repeats every unit of time. The amount of rewards that the winner of the competition obtains is modeled as ρ.

A. Assumption of User Behavior

We assume the behavior of users who compete to publish a block as follows:

- A1. All users have their own initial budget.
- A2. Each user decides whether they will participate in the competition, based on their expected profit. The decision period for all user can be different.
- A3. Users readily have one GPP device, whose computational capacity $C_{GPP} = 1$ and energy consumption $E_{GPP} = 1$.
- A4. Several users whose budget is larger than I_{ASIC} start the competition with as many ASIC devices as their budget will allow. For example, a user with a budget of $2I_{ASIC}$ will start the competition with two ASIC chips.
- A5. After winning a reward, the users could equip themselves with more ASIC chips to extend their computational capacity, provided their budget allows and provided it is profitable.

B. Assumption of Non-Technical Factors

The amount of reward, ρ, is a constant measured in unit of the native cryptocurrency of blockchain. Note that this model omitted various non-technical and irrational factors, such as the market value of the native cryptocurrency, which are beyond the scope of our study

C. Modelling of User Behavior

Here, we define the set of all potential users be $\mathcal{U}$, where $|\mathcal{U}|=K$. $\mathcal{U}$ has a subset $\mathcal{P}$, which is the set of users actually participated in the consensus, i.e., the k-th user u_k participated in the consensus if $u_k \in \mathcal{P}$.

Conversely, $u_k \in \mathcal{P}^c$ means that the user doesn't participate in the consensus.

The total computational capacity (CC) of network, C_T, is defined as the sum of computation capacities of each user who participates in the consensus, i.e.,

$$C_T = \sum_{\forall k,\ \text{s.t. } u_k \in \mathcal{P}} C_k, \tag{51}$$

where

$$C_k = \begin{cases} 1, & \text{for GPP users,} \\ N_k C_{ASIC}, & \text{for ASIC users,} \end{cases} \tag{52}$$

and N_k is the number of ASIC devices that u_k has. Assume all profitable users continue the mining until a block is published by one of them. In this scenario, the probability that a user $u_k \in \mathcal{P}$ published a block is proportional to C_k, i.e., p_k , the probability that u_k published a block, is as follows:

$$p_k = \begin{cases} 0, & \text{if } u_k \in \mathcal{P}, \\ \dfrac{C_k}{C_T}, & \text{otherwise.} \end{cases} \tag{53}$$

The expected profit of k-th user, ψ_k , is computed as follows:

$$\psi_k = \max\left(\rho \frac{C_k}{C_T} - \varepsilon_k E_k, 0 \right), \tag{54}$$

where E_k is the energy consumption of the k-th user and max(A, B) returns the maximum of A and B. It should be noted that the first term of (54), $\rho C_k / C_T$, is expected reward of the k-th user, and the second term, $\varepsilon_k E_k$, is the cost of the k-th user, which will be discussed subsequently.

The expected profits for an ASIC user and GPP user can be separated by using the number of ASIC devices that the k-th user has, N_k , as follows:

$$\psi_{k,ASIC} := \max\left(\rho \frac{N_k C_{ASIC}}{C_T} - \varepsilon_k N_k E_{ASIC}, 0 \right) \text{ for ASIC user,}$$

$$\psi_{k,GPP} := \max\left(\rho \frac{C_{GPP}}{C_T} - \varepsilon_k E_{GPP}, 0 \right) \text{ for GPP user.}$$

D. Modelling of Economy-of-Scale

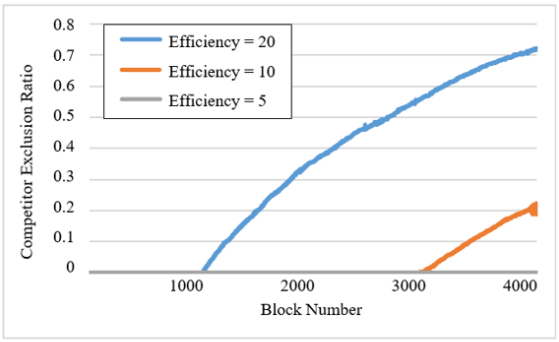
The economy of scale (EOS) is considered in terms of energy price. For example, a user who consumes a lot of energy will enjoy a discounted energy price. The EOS for ASIC users is modeled as follows:

$$\varepsilon_k = \frac{\varepsilon_0}{1 + \gamma \log(N_k E_{ASIC})}, \tag{55}$$

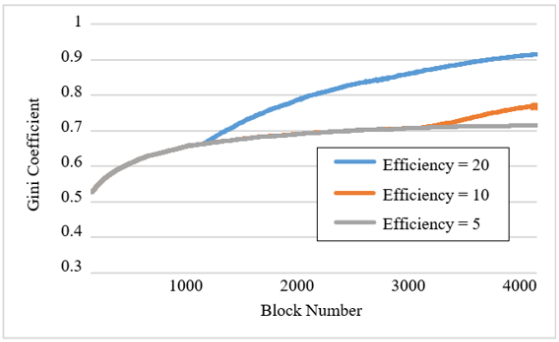
where ε_0 is the base energy price that GPP users experience and γ is a small positive constant. It should be noted that this EOS can capture the energy price and other operating costs.

2.10.2. Simulation Result

Under these scenarios, we simulated the block-publishing competition for 4,000 blocks and measured the competitor exclusion ratio (CER), and Gini coefficient of the blockchain network. CER is defined as the ratio of potential participants excluded from the competition. This ratio implies the fraction of users that are removed from the competition due to low profitability. The Gini coefficient is a metric for wealth monopoly. In this works, we applied the Gini coefficient to measure how the computation capacity is capitalized by heavy users. Figure 11 and Figure 12 demonstrate how those metrics change with different ASIC parameters.

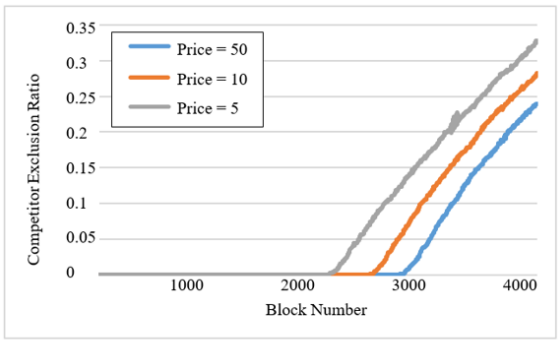


(a) Competitor Exclusion Ratio

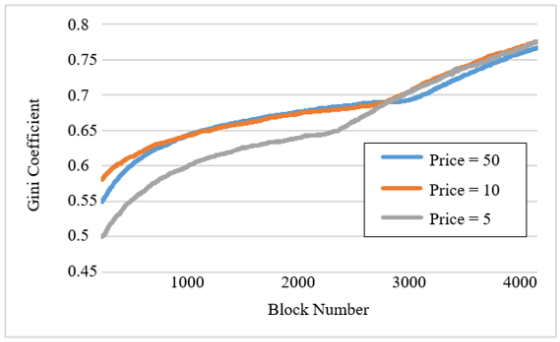


(b) Gini Coefficient

Figure 11. Network Centralization according to the cost efficiency of ASIC. The degree of centralization is measured by (a) competitor exclusion ratio and (b) Gini coefficient.



(a) Competitor Exclusion Ratio



(b) Gini coefficient

Figure 12. Network Centralization according to the price of ASIC. The degree of centralization is measured by (a) competitor exclusion ratio and (b) Gini coefficient.

The simulation results in Figure 11 and Figure 12 indicate that a network with cheap and highly efficient ASIC device (i.e., ASIC-friendly network) becomes centralized more quickly than a network with expensive and inefficient ASIC device (i.e., ASIC-resistant network). However, it should be noted that the CER values and Gini coefficient in Figure *12* indicate opposite result. The CER result in Figure 12(a) implies that high-

price ASIC devices accelerate the centralization by driving GPP users out of competition. Conversely, the Gini coefficient result in Figure 12(b) reveals that the high price of ASIC devices can be helpful for decentralization by slowing down high budget users' monopolization via reinvestment to ASIC devices. Thus, we can derive the following two conclusions:

- C1. Efficient ASIC devices accelerate the centralization of the blockchain network.
- C2. Cheap ASIC devices harm the decentralization by promoting wealth monopolies, but they can also enhance decentralization by increasing the number of participants.

Based on these conclusions, we can outline the characteristics of an ideal ASIC-resistant PoW. The performance gap between GPP and ASIC for the PoW must be minimal. At the same time, ASICs should be affordable enough to be accessible to low-budget users. Such a PoW can maintain decentralization over time, ensuring fair opportunities for low-budget users while still allowing high-budget users to benefit from EOS.

2.10.3. Discussion on Algorithm-Induced Efficiency Gaps

In addition to hardware advances, several studies report software-level accelerations for the LDPC syndrome-decoding problem in (15). Vaz and Gerald [77] shorten message-passing convergence by exploiting code-length constraints. Replacing the sum-product message-passing with lower-complexity decoders can further accelerate, with come loss of DS probability. Representative examples include min-sum decoder [78], [79] and decoders based on the Alternating Direction Method of Multipliers (ADMM) [80], [81]. Other work tunes implementations for specific GPP. A series of GPU-oriented implementations [82], [83] demonstrates throughput improvements for LDPC decoding by exploiting massive parallelism.

However, those software-level accelerations have limitations. The algorithm optimizations such as reducing the number of iterations [77] or replacing certain multiplication operations with logical operations [78]–[81] hardly achieve order-of-magnitude gains. Furthermore, they introduce a modest loss in decoding performance. Consequently, the net effect on throughput, balancing the lower per-decoding complexity against the increased number of decoding attempts, remains uncertain relative to the original decoder. Optimizations that exploit particular constraints such as quasi-cyclic structure of PCM [82], [83] are difficult to apply to the flexible structure of ECCVCC. As a result, they provide only constant-factor improvements, which remain modest compared with the order-of-magnitude advantage achieved by custom ASICs. In addition, the algorithm-induced efficiency gaps can be mitigated at relatively low cost, whereas compensating hardware-induced disparities is considerably more expensive.

2.11. Discussions on Proof-of-Stake Blockchains

In this study, we mainly focused on blockchains based on PoW consensus. Through its operations over the past decade, Bitcoin has demonstrated that these classes of consensus algorithms are quite secure.

Another class of consensus mechanism adopted by multiple blockchains is PoS, in which immutability is secured with a deposit of the block publishers as collateral. In other words, the publishers bet their deposit, rather than energy, to prove the soundness of the block. If a block is revealed to have a flaw by other verifiers, the publisher loses the deposit. In the remainder of this section, we discuss the decentralization and energy consumption of the PoS blockchains comparing with PoW blockchains.

2.11.1. Decentralization of Proof-of-Work and Proof-of-Stake Blockchain

An official document of Ethereum claimed that "the transition to PoS will make the Ethereum significantly more secure and decentralized by comparing to PoW [84]." In the document, two reasons why PoW promotes centralization are suggested: the large initial cost of setting powerful devices and the energy consumption of PoW results in economies of scale. However, a conflicting result can be obtained using our simulation model presented previously. In PoS blockchains, only the verifiers who lock up a deposit can participate in the consensus and be rewarded. The other candidates without a deposit never succeed in publishing. This procedure can be modeled as a block publishing of VCC blockchains, using ASIC with high I_{ASIC} and C_{ASIC} / E_{ASIC}. In other words, the minimum deposit of PoS can be modeled as the initial cost for an ASIC device, I_{ASIC}, while the cost efficiency, C_{ASIC} / E_{ASIC}, approaches infinity. The result in the previous Section VI indicates that such a blockchain would be centralized quickly.

The simulation model was proposed to compare the decentralization of PoW blockchains, rather than to compare PoW and PoS. Thus, the previous result cannot directly imply that PoW-based blockchains are more decentralized than PoS blockchains. Further research on this would be interesting future work.

2.11.2. Energy Consumption of Proof-of-Work and Proof-of-Stake Blockchain

The computational challenge of PoW secures a blockchain through energy spending without relying on trust, whereas PoS depends on the assumption that validators who stake their cryptocurrency will operate the network honestly. Because nodes in PoS do not compete in computational capacity, the energy required by PoS is far less than that of PoW, in which heavy energy use is unavoidable.

Nevertheless, several design variants demonstrate that equivalent security, can be achieved far more efficiently. GreenPoW [85] lowers energy consumption by granting the exclusive right to mine the next block to

the producer of the most recent uncle block, eliminating redundant hashing. Sprint [86] combines proof-of-work with verifiable delay functions (VDFs), forcing miners into intermittent activity and thus curbing power demand. EpoW [87] awards the block to the second solver of the PoW puzzle, which discourages continuous hash-rate escalation across the network. Incorporating one or more of these techniques into ECCVCC would bring its energy consumption to a more acceptable level

2.12. Conclusions

In this study, we proposed ECCVCC, a novel ASIC-resistant blockchain consensus algorithm whose design can be tailored by selecting a VCP derived from error-correction code problems. As illustrative cases, we compared ECCVCC-CD and ECCVCC-SD, which employ codeword and syndrome decoding puzzles, respectively. In that comparison, ECCVCC-SD outperforms ECCVCC-CD and the previous works [45], [46], [47] with respect to difficulty-adjustment flexibility and robustness against DoS attacks.

We also explored the relationship between ASIC-resistance and blockchain decentralization. The simulation results in Section VII demonstrated that a blockchain network with an ASIC-resistant algorithm remained decentralized for a longer time compared to one using an ASIC-friendly algorithm. Furthermore, as discussed in Section VIII, this finding implies that a blockchain utilizing PoS, which can be modeled as an extremely ASIC-friendly algorithm, tends to become centralized more quickly.

Chapter III. [Stealth-Achieving ECC] Covert Anti-Jamming Communication based on Gaussian Coded Modulation

3.1. Motivation

In wireless communication systems (WCSs), both covertness and anti-jamming capability are often required simultaneously. In a hostile environment, an adversary may attempt to eavesdrop on or jam the link. Concealing the signal from eavesdroppers also helps counter jamming, because modern “smart” jammers rely on estimated signal parameters to maximize their effectiveness. If the signal is hidden, those parameters cannot be estimated. Moreover, several components, such as error-correction coding, spread-spectrum techniques, and beamforming, serve both covert and anti-jamming objectives. Accordingly, it is possible to design a unified wireless framework that provides covertness and jamming resilience at the same time.

3.2. Introduction

Wireless communication systems (WCSs) have become an essential part of the infrastructure for exchanging information. However, WCSs are exposed to various threats due to the open nature of wireless media. One of these threats is eavesdropping. Eavesdroppers means malicious receivers that harm innocent users using information on the radio transmission. Since the advent of wireless communication, security to prevent damage caused by eavesdroppers has been considered an important concern.

Covert communication [18], [19] has received little attention, relative to its importance. Without covert communications, the physical location of wireless transmitters can be detected by an eavesdropper, owing to the wireless transmission of signals. In such scenarios, the presence of wireless communications should not be easily detectable by hostile eavesdroppers.

Intentional electromagnetic radiation, also called jamming, is another serious threat to the robustness of WCSs. Jamming attacks can disturb ongoing wireless communications by causing additional errors. Jamming attacks can become more destructive if the jammers cooperate with eavesdroppers. A follower jammer [88] is an example of such co-operation. It first measures the time-frequency band of the target signal and then concentrates its jamming energy onto that band. Covert communication decreases damage by jammers using the eavesdropping-and-jamming strategy by preventing the leakage of system parameters. Thus, to protect the security and robustness of WCSs, covertness and anti-jamming performances are required simultaneously.

A spread spectrum (SS) scheme is one common solution for overcoming jamming at-tacks in the physical layer. Through this scheme, the bandwidth of band-limited jamming becomes much smaller than that of

the communication signals. Furthermore, the scheme makes communication difficult to detect by hiding communication signals under the noise level, thus reducing the threat of eavesdroppers and eavesdropper-aided smart jammers such as follower jammers.

The coded modulation method using Gaussian-distributed code can be a complement to the SS scheme. If the modulated symbols follow a Gaussian distribution, it is difficult for the eavesdropper to distinguish whether there is a communication signal or not without an exact codebook. Furthermore, the legitimate receiver can use the error-correction property of the coding scheme. The idea of real-valued channel coding has been used, and is known as analog coding [89]. The code can correct sparse errors (e.g., narrowband jamming). Reference [90] have studied a generalization of this coding scheme: a complex-field coding using orthogonal frequency-division multiplexing (OFDM). Candes and Tao [91] proposed a sparse error-correction method for an arbitrary generator matrix, which is a polynomial–time linear programming method unified within a compressive sensing (CS) recovery framework [92]. Studies motivated by analog coding and many other CS-based anti-jamming approaches have exploited the sparse characteristic of jamming in both the time-frequency [93], [94] and spatial domains [95]. Thus, analog coding can be applied to anti-jamming communication.

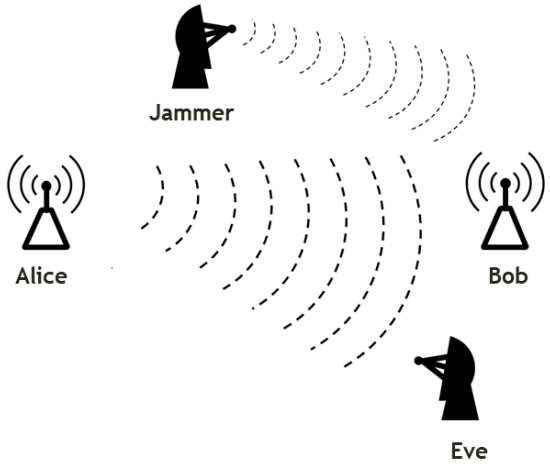


Figure 13. Tactical wireless communication system under an electronic warfare scenario.

In this study, we consider wireless communications under a jammer and eavesdropper scenario, for example, the electronic warfare scenario depicted in Figure 13. The proposed system transmits information messages on a coded Gaussian signal through a wireless channel that is attacked by an eavesdropper, called Eve, and a jammer. We consider two types of jammers. The first type is a blind jammer, which radiates a partial-band or pulse jamming signal without knowing when or where a target communication signal exists. The second type is a follower jammer, which detects an ongoing communication signal and then radiates a jamming signal to interfere with this ongoing signal. We assume that Eve is not aware of the exact codebook and time-frequency

band of the target system. Suppose that there is no secret information shared between the transmitter and the receiver. Then the achievability of covert communication depends entirely on the channel quality difference between the receiver and Eve, which is difficult for communicators to control. Thus, many studies assume that the transmitter and the receiver share some secret information, such as a codebook [18], [96] or spreading sequence [35], [36]. This secret information is often time-varying to prevent information leakage [97]. One practical example is a frequency-hopping radio that exploits the exact time-frequency band of a signal as secret information.

3.3. Contribution

In this study, we propose a novel covert anti-jamming communication system based on a noise-like Gaussian-coded time-frequency modulation, as illustrated in Figure 14. To provide covertness to a communication signal, we propose a time-frequency modulation scheme. It is difficult to distinguish a signal generated using the proposed scheme from Gaussian noise. To provide robust anti-jamming performance, we propose two receiver algorithms that estimate and remove the jamming signal from the received signal by exploiting the sparse nature of the jamming signal.

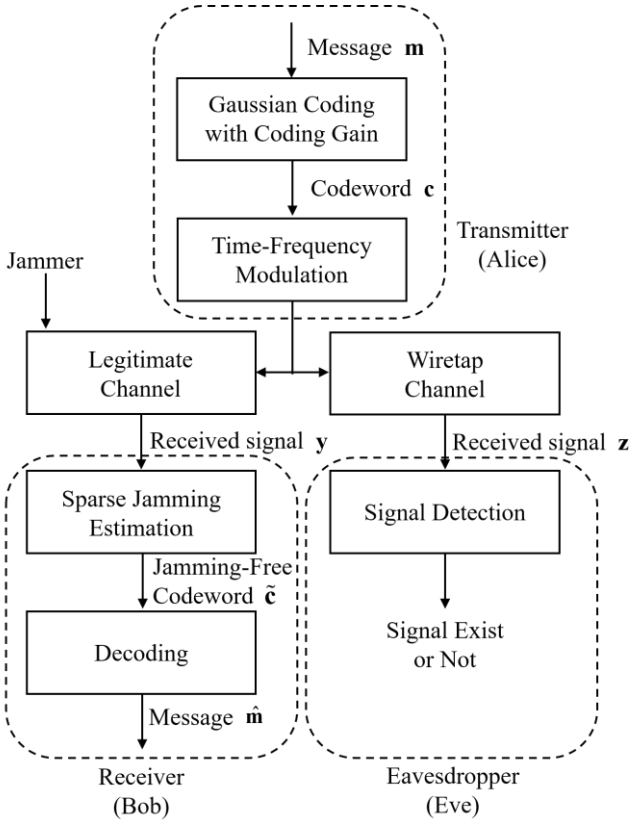


Figure 14. Proposed covert anti-jamming communication system with Gaussian-coded time-frequency modulation.

The main contributions of this study are as follows:

- We propose a covert anti-jamming communication system based on Gaussian-coded time-frequency modulation. We consider the communication system that is exposed to the threat of the jammer and Eve. Previous studies on analog coding [89], [90], [91], [93], [94], [95] have not considered the threat of jammers and eavesdroppers simultaneously. We designed a coding and modulation method for a transmitter to achieve anti-jamming and covert communication simultaneously. For a receiver, to estimate and remove jamming, we propose two novel sparse jamming estimation (SJE) algorithms, i.e., greedy SJE (GSJE) and Bayesian SJE (BSJE), which are described as Algorithm 1 and Algorithm 2, respectively.
- For the proposed Gaussian coding scheme, we aimed to develop a real-valued coding scheme in which a coding gain is provided. Previous studies on analog coding [89], [91] have considered analog messages. In this study, we show that the coding method cannot provide a coding gain for finite-field messages. As a countermeasure, we introduce two methods to achieve coding gain even when the finite-field messages are encoded. The first method is a two-stage coding approach, applying finite-field coding and linear Gaussian coding sequentially. In this method, the overall coding gain is equivalent to that of the finite-field coding used in the first stage. Another coding method is a codebook method that uses a pregenerated codebook rather than a linear operation on message vectors. We demonstrate the existence of a coding gain through numerical results that compare the bit error rate (BER) performance of the proposed system under jamming and the uncoded direct-sequence spread spectrum (DSSS) without jamming.
- We show the undetectability of the signal transmitted from the proposed system. The signal of the proposed system is compared to that of a conventional binary phase-shift keying (BPSK)-DSSS scheme. For this comparison, we considered Eve to be an energy-detecting eavesdropper that can distinguish the communication signal from noise using the maximum likelihood (ML) method. The detection capability of Eve increases as the signal-to-noise ratio (SNR) and false alarm probability increases. Our results demonstrate that the proposed system offers significantly better undetectability than the BPSK-DSSS scheme in such critical cases. Thus, the proposed method is superior when Eve is the most threatening.

3.4. Signal Design

In this section, we describe the proposed signal design. In the proposed system, the codeword alphabet consists of Gaussian-distributed real numbers rather than a finite-field alphabet. The transmitted Gaussian codeword appears as additive white Gaussian noise (AWGN) to Eve. In contrast, the receiver who knows the exact

codebook can detect the transmitted codeword even with a significantly low SNR setting. Figure 15(a) illustrates the time-domain signal waveform of the proposed system at the 0 dB SNR setting. The wave-form with noise in Figure 15(c) is hardly distinguishable from the AWGN in Figure 15(b). A detailed analysis of this undetectability is presented in Section 3.6. In this section, we con-sider two encoding methods used to construct such codewords. We first introduce the simplest method, linear block coding (LBC), as proposed in previous studies [89], [91]. Then, we present the limitations of that method and propose two methods to address these limitations.

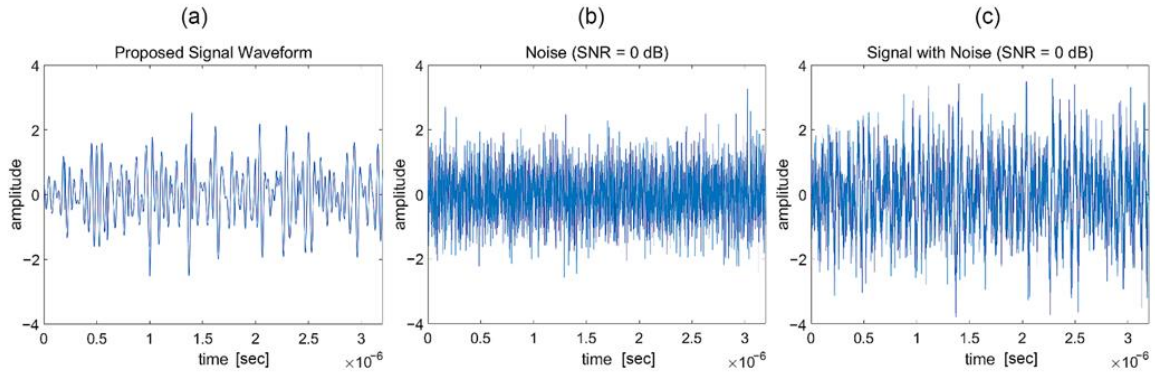


Figure 15. Time-domain waveform of (a) proposed signal, (b) noise, and (c) signal with noise under additive white Gaussian noise (AWGN) channel with a signal-to-noise ratio (SNR) of 0 dB

3.4.1. Codeword Generation

A binary message vector $\mathbf{m} \in \{-1,1\}^L$ is encoded into a real-valued codeword vector $\mathbf{c} \in \mathbb{R}^{2N}$, where $\mathbb{R}$ is the set of real numbers. Each element of the codeword, i.e., each time-frequency symbol, follows a Gaussian distribution. Here, the encoding $\mathrm{Enc}: \{-1,1\}^L \mapsto \mathbb{R}^{2N}$ can be any function that includes a redundancy to the codeword. The redundancy can be exploited to estimate the jamming and correct the channel-induced errors.

In the previous work of Candes and Tao [91], LBC-like encoding of the real-valued message was studied. That is, $\mathrm{Enc}_{\mathrm{LBC}}: \mathbb{R}^L \mapsto \mathbb{R}^{2N}$ was previously investigated. As it is a simple method, we can use it as a codeword generation method. We call this method Gaussian-LBC. Let us define the probability density function (PDF) for a Gaussian random variable with a mean μ and variance σ^2 as $\mathcal{N}(\mu, \sigma^2)$. In the Gaussian-LBC, a codeword vector becomes the product of a generator matrix and a message vector:

$$\mathbf{c} = \mathbf{G}\mathbf{m}, \tag{56}$$

where $\mathbf{m}$ is the L-dimensional message vector, $\mathbf{G} \in \mathbb{R}^{2N \times L}$ is the generator matrix of which the entries follow $\mathcal{N}(0,1/N)$, and $\mathbf{c} \in \mathbb{R}^{2N}$ is the codeword vector. Because the method is similar to LBC in finite fields, one can define concepts and notations corresponding to the parity-check matrix, syndromes, and error

correction. However, error correction using syndromes in real-valued code is a nonlinear pattern recognition problem that has no simple solution for an arbitrary $\mathbf{G}$ [89]. A linear programming solution to this real-valued syndrome decoding problem was described in [91]. The solution has been widely accepted for several other sparse signal recovery (SSR) problems, under the name of CS [92]. Numerous algorithms have been proposed to achieve this solution.

However, we require another approach with Gaussain-LBC. This is because $\mathrm{Enc}_{\mathrm{LBC}}$ cannot fully exploit the characteristics of the digital message; thus, one cannot obtain an adequate coding gain. In the following section, we discuss an alternate design of $\mathrm{Enc}:\{-1,1\}^L \mapsto \mathbb{R}^{2N}$ to obtain the coding gain

3.4.2. Gaussian Coding with Coding Gain

The Gaussian-LBC method has a limitation in that the method considers only the real-valued message case. However, for modern communication systems that transmit a variety of finite-field messages, the encoding of a finite-field message must be considered. Unfortunately, the direct application of the Gaussian-LBC method to finite-field messages is inefficient because there is no actual coding gain. In finite-field coding, the coding gain is defined by how much the minimum Hamming distance between two codeword vectors increases when compared to that between message vectors. This is because the finite-field decoder maps the input vectors to the closest codeword in terms of the Hamming distance. In contrast, the coding gain of a real-valued code cannot be defined by the Hamming distance because the closeness of real-valued codewords to non-sparse noise is meaningless. Instead, the Euclidean distance has to be used for real-valued code. It can be noted that the minimum Euclidean distance is constant before and after the Gaussian-LBC, i.e.,

$$\begin{aligned}(\mathbf{c}-\tilde{\mathbf{c}})^T(\mathbf{c}-\tilde{\mathbf{c}}) &= (\mathbf{G}\mathbf{m}-\mathbf{G}\tilde{\mathbf{m}})^T(\mathbf{G}\mathbf{m}-\mathbf{G}\tilde{\mathbf{m}}) \\ &= (\mathbf{m}-\tilde{\mathbf{m}})^T\mathbf{G}^T\mathbf{G}(\mathbf{m}-\tilde{\mathbf{m}}) \\ &= (\mathbf{m}-\tilde{\mathbf{m}})^T(\mathbf{m}-\tilde{\mathbf{m}}),\end{aligned} \tag{57}$$

where $(\cdot)^T$ is the transpose operator for the vector/matrix and $\mathbf{G}^T\mathbf{G}=\mathbf{I}$ for large N. To benefit from coding gain, we considered two approaches: concatenation coding and the codebook method.

A. Concatenation Coding

To provide an encoding method with coding gain, concatenation coding was considered. Namely, the encoding of a digital message $\mathrm{Enc}:\{-1,1\}^L \mapsto \mathbb{R}^{2N}$ becomes a serial combination of two different coding methods, as depicted in Figure 16. The message is encoded twice, first by an outer code

$\mathrm{Enc}_{out}:\{-1,1\}^{L}\mapsto\{-1,1\}^{N_{out}}$ and then by an inner code $\mathrm{Enc}_{in}:\{-1,1\}^{N_{out}}\mapsto\mathbb{R}^{2N}$. We used a finite-field coding method, which yields a coding gain, as the outer code, and then used the Gaussian-LBC as the inner code so that the system could obtain the advantages of both the error-correcting coding and Gaussianity of the signal.

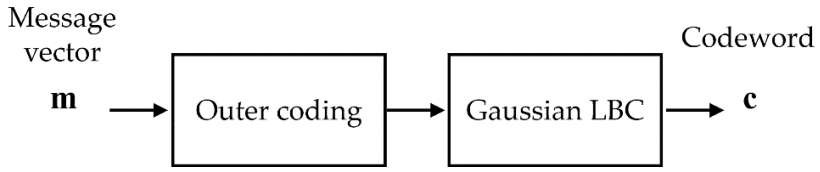


Figure 16. Concatenation coding scheme Gaussian signaling.

The decoding procedure is also two-fold. Given a jamming-mitigated codeword $\tilde{\mathbf{c}}$, the Gaussian-LBC codeword is decoded first by $\mathrm{Dec}_{in}:\mathbb{R}^{2N}\mapsto\{-1,1\}^{N_{out}}$; then, the outer codeword is decoded by $\mathrm{Dec}_{out}:\{-1,1\}^{N_{out}}\mapsto\{-1,1\}^{L}$. Here, we focus on the decoding of the Gaussian-LBC Dec_{in}. Because the decoding of the Gaussian-LBC is equivalent to an overdetermined linear inverse problem (LIP), ML decoding can be performed using the least-squares method with $O(NN_{out})$ computations, where N_{out} is the length of the outer codeword. If the code rate of the outer code is fixed (i.e., $O(N_{out})=O(L)$), the complexity is equivalent to $O(NL)$. Based on the output of the Gaussian-LBC decoder Dec_{in}, the original message is decoded using the outer code decoder Dec_{out}, i.e., the recovered message vector $\hat{\mathbf{m}}:=\mathrm{Dec}_{out}(\mathrm{Dec}_{in}(\tilde{\mathbf{c}}))$.

B. Gaussian Codebook

The Gaussian codebook method is another coding method with coding gain. If the length of the binary message vector is *L*, then 2^{L} codeword vectors of length 2*N* are pregenerated. Then, the one-by-one mapping encoding function $\mathrm{Enc}:\{-1,1\}^{L}\mapsto\mathbb{R}^{2N}$ can be constructed between 2^{L} message vectors and codeword vectors. The codeword can be decoded using the ML method. $O(N2^{L})$ computations are required to decode the original message from the proposed codeword generated by Enc. Here, we construct a codebook $\mathbf{G}$ with a Toeplitz matrix of which the columns are cyclic shifts of a Gaussian random vector $\mathbf{g}=[g_1,g_2,...,g_{2N}]^T$, rather than generating every element of $\mathbf{G}$ with an independent and identically distributed (i.i.d.) Gaussian. There are two benefits to constructing the codebook as a Toeplitz matrix. First, the Toeplitz structure accelerates the computation of the decoding method by using the fast Fourier transform (FFT). Second, such a codebook can be stored in a more memory-efficient manner than a random codebook because the Toeplitz codebook requires as much memory space as $O(N)$, whereas the i.i.d. codebook requires as much memory space as $O(N2^{L})$. If the computational

costs of the FFT and inverse FFT (IFFT) are considered, the overall computational complexity becomes $O(N \log N)$.

The concatenation method is attractive in practice because well-studied finite-field coding methods can be directly exploited as the outer code. A proper outer code can be selected according to the purpose and requirements of the system. We used the codebook method in the following simulation section because the Toeplitz codebook method provides a good restricted isometric property (RIP) condition and is independent of the outer code design.

3.5. System Model

The codewords are transmitted by spreading over a wide block of time-frequency OFDM slots. After demodulation and demultiplexing, the receiver obtains the codewords that are distorted by noise and jamming. The jamming portion is removed using the SSR technique [92]. The receiver decodes the original messages based on the jamming-free codewords.

3.5.1. Time-Frequency Modulation

The real-valued codeword **c** of dimension $2N$ can be rewritten as a complex-valued codeword vector of dimension N, i.e., $\mathbf{x} \in \mathbb{C}^N$, where $\mathbb{C}$ is a set of complex numbers. The real part of **x** is the first half of the vector **c** and the imaginary part of **x** is the latter half of **c**. That is, for the real-valued codeword $\mathbf{c} = [c_1, c_2, ..., c_{2N}]^T$, the corresponding complex expression is $\mathbf{x} = [c_1 + jc_{N+1}, ..., c_N + jc_{2N}]^T$. Each element of **x** is assigned to the time-frequency grid for transmission by OFDM. Let the number of OFDM subbands be N_F and the number of time slots be N_T, where $N = N_F N_T$. Figure 17(a) illustrates the proposed signal in the time-frequency domain, where $N_F = N_T = 16$. The signal is transmitted through all 16 subbands, whereas the frequency-hopping spread spectrum (FHSS) signal in Figure 17(b) uses only a single subband in a time slot. The energy of the proposed signal remains under the noise floor at the plane where the energy = 1, whereas the energy of the FHSS signal with the same power has peaks above the noise floor.

The complex codeword vector **x** is divided into N_T column vectors, i.e., $\mathbf{x}_k \quad (k = 1, 2, ..., N_T)$, where $\mathbf{x} = [\mathbf{x}_1^T \quad \mathbf{x}_2^T \quad \cdots \quad \mathbf{x}_{N_T}^T]^T$. The k-th OFDM symbol is determined by an IFFT of $\mathbf{x}_k$. Here, we omit the cyclic prefix (CP) for simplicity, as channel estimation and the inter-symbol interference problem are beyond the scope of this study. If one assumes perfect channel estimation, then the CP being jammed cannot be a problem as

the CP is discarded on the receiver side. However, note that jamming attacks on the CP can distort the OFDM system efficiently [102], [103], thus degrading the performance of the system. Future research should study the effect of CP jamming.

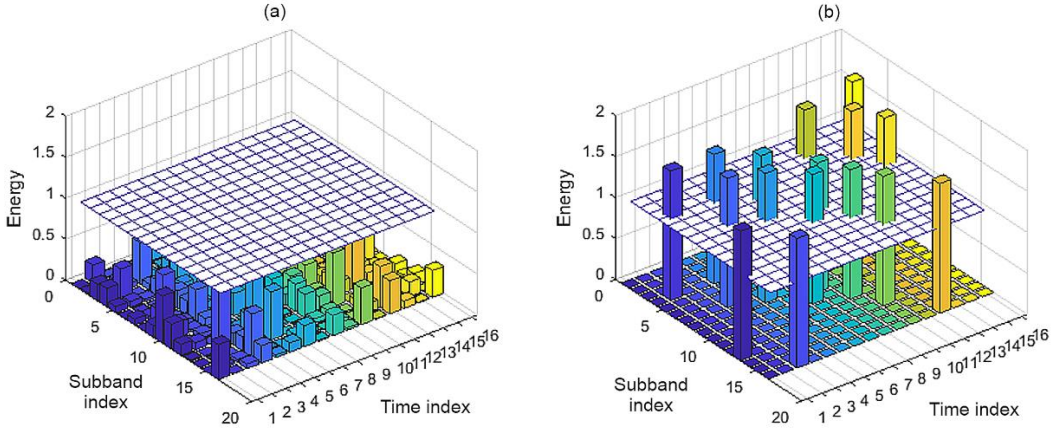


Figure 17. Comparison between proposed signal and frequency-hopping signal in the time-frequency domain: (a) proposed signal with orthogonal frequency-division multiplexing and (b) frequency-hopping signal.

3.5.2. Channel Model

There are two separate channels in the proposed system. One is the legitimate channel from the transmitter, Alice, to the legitimate receiver, Bob. The other is a wiretap channel from Alice to the eavesdropper Eve, who is not a legitimate user. We consider that the legitimate channel suffers from hostile jamming, whereas the wiretap channel does not. Figure 18 illustrates the channel model. Bob obtains the signal through the legitimate channel, which is contaminated by jamming, whereas Eve receives the signal through the wiretap channel. This modeling encompasses the following two scenarios within a single logically equivalent scheme. The first scenario is that Eve and the jammer are combined into a single actor. The follower jammer is an example. The second scenario is that the two are located apart. For example, the jammer can be located close to the receiver, whereas the eavesdropper can be placed near the transmitter for better performance. In addition, Alice and Bob share a codebook $\mathcal{C} \in \mathbb{C}^N$ which is the set of possible codewords. To focus on covert communication, we assume that Eve does not have the codebook.

Let $\mathbf{h}_k$ be the channel impulse response vector of which the length is N_F. The k-th received vector after OFDM demultiplexing and CP removal is

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{e}_k + \mathbf{w}_k, \tag{58}$$

where $\mathbf{e}_k \in \mathbb{C}^{N_F}$ is a jamming vector, $\mathbf{w}_k \in \mathbb{C}^{N_F}$ is additive circularly symmetric complex Gaussian (CSCG) noise, and $\mathbf{H}_k \in \mathbb{C}^{N_F \times N_F}$ is a diagonal matrix of which the diagonal elements are the FFT of $\mathbf{h}_k$. Assuming that

the fading channel has no null, the matrix $\mathbf{H}_k$ is invertible. The N_T OFDM symbols can be expressed as a single-vector expression through concatenation; as is the case with $\mathbf{x}=[\mathbf{x}_1^T \quad \mathbf{x}_2^T \quad \cdots \quad \mathbf{x}_{N_T}^T]^T$, the single-vector expression for the entire received vector $\mathbf{y}=[\mathbf{y}_1^T \quad \mathbf{y}_2^T \quad \cdots \quad \mathbf{y}_{N_T}^T]^T$ becomes

$$\mathbf{y}=\mathbf{H}\mathbf{x}+\mathbf{e}+\mathbf{w}, \tag{59}$$

where the combined channel matrix $\mathbf{H}$ is the block-diagonal matrix consisting of $\mathbf{H}_1,\mathbf{H}_2,\ldots,\mathbf{H}_{N_T}$, $\mathbf{e}=[\mathbf{e}_1^T \quad \mathbf{e}_2^T \quad \cdots \quad \mathbf{e}_{N_T}^T]^T$ is the jamming vector, and $\mathbf{w}=[\mathbf{w}_1^T \quad \mathbf{w}_2^T \quad \cdots \quad \mathbf{w}_{N_T}^T]^T$ is the CSCG noise vector.

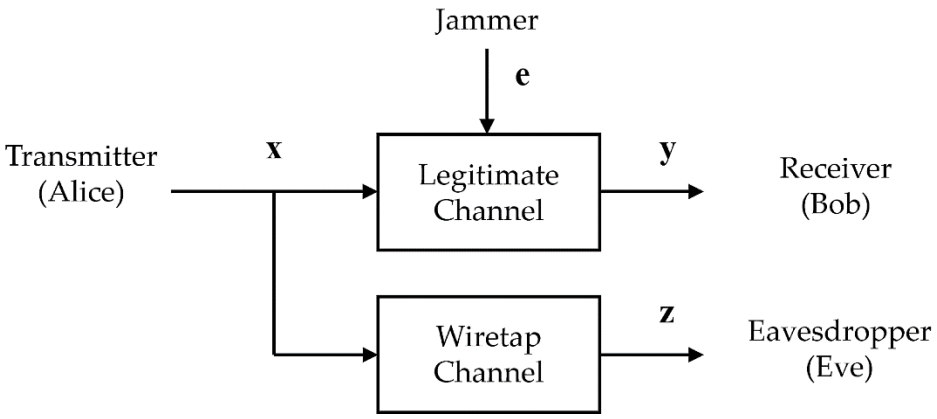


Figure 18. Wiretap channel model under jamming

3.5.3. Sparse Jamming Model

In our scenario, the jamming vector $\mathbf{e}$ is assumed as a spare vector because practical jammers are modeled as band-limited or time-limited jammers. In fact, a jamming method that interferes with every time-frequency symbol simultaneously would be a powerful jamming method. However, in the proposed covert communication system, the jammer cannot be sure of the exact time-frequency band of the target signal because the proposed system is intended to provide undetectable characteristics of the transmitted waveform, as shown in Section 4. Several studies on the energy optimization of jammers, including [104], [105], have indicated that a jamming policy without a precise signal power estimation is not optimal. To achieve energy efficiency, the jammer must acquire the time-frequency band of the target communication.

Without knowing the codebook and exact time-frequency band of target signals, the jammer can have two options—blind barrage jamming and sparse jamming. On the one hand, we consider a barrage jammer that aims to attack a target signal with its signal bandwidth when the time duration is unknown. The jammer cannot launch an attack with a wider bandwidth and longer time duration than those of the target communication signal.

Consider B as the bandwidth of the target signal and D as the time duration of the target signal. Obviously, the bandwidth and time duration of the attack signal become $B+\Delta B$ and $D+\Delta D$ for $\Delta B \geq 0$ and $\Delta D \geq 0$, respectively. Here, it can be observed that the jammer would expend an excessive amount of energy due to the amount of excessive time and bandwidth $O(\Delta B \Delta D)$. Nevertheless, the attack by the jammer may still remain ineffective, provided that the time and frequency band of the target signal is unknown. The only option that the barrage jammer has is to spread jamming power over a wide range of time-frequency regions with large ΔB and ΔD. However, the power budget of a practical jammer is not infinite. Thus, to launch a barrage jamming attack, the energy density per unit time-frequency region shall remain low. In such a case, the effective jamming energy, i.e., the actual jamming energy spent on the target time-frequency region, shall remain small, and the jamming becomes unsuccessful.

On the other hand, the jammer can choose a sparse jamming approach, aiming to reduce the inefficiency of barrage jamming and concentrate the jamming energy onto a specific time-frequency band. Many jammers therefore aim to identify the time-frequency band of the ongoing communications. Once this is identified, an attack can be launched in the form of partial-band noise jamming, such as pulse jamming, as discussed in [93], and reactive jamming, which is discussed in [94]. Thus, the jammers can concentrate their jamming power on a small, selected part of the entire time-frequency band. What does this mean to the proposed covert anti-jamming communication system? The proposed system is supposed to provide undetectable characteristics of the transmitted waveform, which results in two scenarios. On the one hand, jammers cannot detect the presence of the signal or its time-frequency band at all. This is the case when Eve is physically located far from the transmitter. In this case, the jammers do not launch any jamming attacks. On the other hand, in the case in which Eve is close to the transmitter unit, the signal is partially detectable. It can be recalled that our signals are wideband Gaussian signals. Thus, there are a few time-frequency bands of which the energy levels pierce through the noise floor (refer to the illustration in Figure 17). Eve can be deceived that the ongoing communication exists only on those detected time-frequency bands; consequently, Eve will aim to launch precise jamming attacks at the detected time-frequency bands. The previous discussion explains why we have modeled the jamming attacks to be sparse in the time-frequency domain. By exploiting the sparse nature of jamming, Bob can successfully estimate and remove jamming from the received vector.

3.5.4. Sparse Jamming Estimation

Let us recall the received vector specified in (59). There always exists a nonzero annihilating matrix $\mathbf{A}$ such that $\mathbf{AHx}=\mathbf{0}$, unless $\mathbf{Hx}$ spans $\mathbb{C}^N$. Let a matrix $\mathbf{G}$ be a generator matrix or codebook matrix of

which the columns span the range space of the codewords. Here, note that the generator or codebook matrix $\mathbf{G}$ is constructed by a random Gaussian distribution.

Because the codeword space is a subspace of $\mathbb{C}^N$, singular-value decomposition (SVD) can be applied to determine the annihilating matrix $\mathbf{A}$. Consider an SVD of $(\mathbf{HG})^H$, i.e.,

$$(\mathbf{HG})^H = \mathbf{U}\begin{bmatrix}\mathbf{D} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^H \\ \mathbf{A}\end{bmatrix}. \tag{60}$$

It can be determined that $\mathbf{A}$ satisfies $\mathbf{AHx} = \mathbf{0}$ for all possible values of $\mathbf{x}$. By multiplying $\mathbf{A}$ to both sides of (59), Bob obtains the measurement vector $\mathbf{b}$,

$$\begin{aligned}\mathbf{b} &= \mathbf{A}(\mathbf{e}+\mathbf{w}) \\ &= \mathbf{A}\tilde{\mathbf{e}},\end{aligned} \tag{61}$$

where $\tilde{\mathbf{e}} := \mathbf{e}+\mathbf{w}$ is the jamming-plus-noise vector. Because $\mathbf{b}$ and $\mathbf{A}$ are known to Bob, $\mathbf{e}$ can be obtained by solving for the SJE problem of (61), which is in the form of a LIP. Unfortunately, the SJE problem is underdetermined because the annihilating matrix $\mathbf{A}$ is the null space of $\mathbf{HG}$. However, let us recall the assumption that $\mathbf{e}$ is a sparse vector. It is known that a sparse vector can be recovered using SSR techniques even if the LIP is underdetermined.

There are two conditions related to the successful reconstruction of the jamming vector $\mathbf{e}$ from (61) using SSR techniques. The first condition is that the LIP should have good properties. The goodness of LIP is often measured using a condition referred to as RIP [91] of the sensing matrix. In the proposed SJE problem, the sensing matrix corresponds to the annihilating matrix **A**. Several studies have shown that a sensing matrix has a good RIP if the matrix is created from an i.i.d. zero-mean Gaussian. However, the annihilating matrix **A** in the SJE problem is not created from such a distribution, but is created from the null space of **HG**.

Several studies have discussed the RIP of the null-space matrix. Candes *et al.* [91] and Stojnic *et al.* [106] argued that the null-space matrix of an i.i.d. Gaussian random matrix demonstrates a good property because the null space can consist of an i.i.d. Gaussian random basis. Xu *et al.* [107] showed that the property of the sensing matrix that is created from the null space of a Toeplitz matrix is good for SSR. Note that $\mathbf{HG}$ is a Gaussian matrix because the OFDM channel matrix $\mathbf{H}$ is a Gaussian matrix and $\mathbf{G}$ is also a Gaussian matrix (refer to Section 3.4 for more details). Further, $\mathbf{HG}$ can be an i.i.d. Gaussian matrix or Gaussian Toeplitz matrix when the channel is a frequency nonselective channel. Thus, these results suggest that the SJE problem can be solved successfully, depending on the channel.

The second condition for successful SSR is that the jamming with noise vector $\tilde{\mathbf{e}}$ should be close to the sparse vector. If the energy of a sparse vector is much larger than that of a non-sparse vector, the sum of the two vectors is said to satisfy soft sparsity. Raskutti *et al.* [108] discussed the convergence of the SSR under the soft sparsity model. Arias-Castro and Eldar [109] studied the premeasurement noise model, which is similar to the soft sparsity model. In their study, it was shown that the premeasurement noise model can be reformulated into a general SSR model (i.e., measurement noise model), with a small cost on RIP. Based on these studies, we can conclude that $\tilde{\mathbf{e}}$ can be successfully recovered, as $\tilde{\mathbf{e}}$ also satisfies the soft sparsity model. To derive the SJE algorithms, we explored two representative SSR algorithms: greedy algorithms and sparse Bayesian learning (SBL) algorithms. In the following subsections, we propose the GSJE algorithm and the BSJE algorithm.

A. *Greedy Sparse Jamming Estimation Algorithm*

To estimate the sparse jamming vector $\mathbf{e}$ from the measurement $\mathbf{b}$, one can define a support set $\mathbb{S} \subset \{1,2,...,N\}$, which is a set of indices k, such that the k-th element of $\mathbf{e}$ is nonzero. Thus, $\mathbb{S} = \{k \mid \hat{e}_k \neq 0, k = 1,2,...,N\}$. Let us define $\mathbf{A}_{\mathbb{S}}$ and $\mathbf{e}_{\mathbb{S}}$ as submatrices/subvectors of which the columns/elements contain atom $\mathbf{a}_k$ and coefficient $\hat{e}_k$ for $k \in \mathbb{S}$, respectively. Then, $\hat{\mathbf{e}}$ can be determined by solving the following optimization:

$$\hat{\mathbf{e}} = \arg\min_{\mathbf{e}} \|\mathbf{b} - \mathbf{A}\mathbf{e}\|^2, \text{ subject to } \|\mathbf{e}\|_0 \leq K, \tag{62}$$

where K is the maximum number of nonzero coefficients in $\mathbf{e}$. Equation (7) is an optimization problem that determines the solution $\mathbf{e}$ that minimizes the approximation error with the l_0 norm constraint. To solve this problem, several greedy algorithms have been proposed. The greedy algorithms iteratively pursue one column of the annihilating matrix $\mathbf{A}$ at a time, which significantly reduces the approximation error. Thus, the estimate of the support set $\mathbb{S}$ grows one by one as the iteration progresses, starting from the empty set in the beginning. The matching pursuit (MP) [29], orthogonal MP (OMP) [18 19] and their variants [30 31] are most relevant to the derivation presented in this Section. The MP algorithm for SSR was first proposed by Mallat and Zhang [29]. By applying the MP algorithm to our problem, we can determine one column of the annihilating matrix $\mathbf{A}$ that maximizes the inner product with a residual vector at each iteration, and then calculate the corresponding nonzero coefficient that minimizes the estimation error. Conversely, the OMP algorithm has an additional procedure that updates all the coefficients in the subvector $\mathbf{e}_{\mathbb{S}}$ at each iteration using the orthogonal projection of the measurement vector onto the subspace $\mathbf{A}_{\mathbb{S}}$.

In this section, we propose the GSJE algorithm described as Algorithm 6. The algorithm aims to estimate the sparse jamming vector from the received signal vector $\mathbf{y}$ in (59). Given $\mathbf{H}$, $\mathbf{G}$ and $\mathbf{y}$, the algorithm calculates $\mathbf{A}$ using (60). Then, the jamming vector $\mathbf{e}$ is estimated by approximating the solution of (62) using the greedy iteration. We assume that the channel matrix $\mathbf{H}$ and codeword space $\mathbf{G}$ are known to the receiver. The annihilating matrix $\mathbf{A}$ can then be determined from the SVD of $(\mathbf{HG})^H$, as given (60), and used to construct the SJE problem. We assume here that the magnitudes of the nonzero coefficients of the jamming vector $\mathbf{e}$ are significantly larger than those of the noise vector $\mathbf{w}$, i.e., $\|\mathbf{e}\|^2 \gg \|\mathbf{w}\|^2$. This assumption is called the high jamming-to-noise ratio (JNR) assumption. Under this assumption, each iteration of the GSJE will identify a member of the true support set with high probability. In contrast, if the JNR is significantly low (for example, under −10 dB), the accuracy of SJE under the high JNR assumption might be degraded. However, note that the overall system performance, i.e., BER, is no longer affected by the accuracy of the SJE. However, if JNR is in the moderate region, the failure of the GSJE may cause an overall performance degradation. Thus, the model mismatch becomes more significant in the moderate-JNR region. Another challenge is that the GSJE requires certain prior knowledge about jamming. As discussed above, the parameters δ and K that determine the termination condition depend on prior information such as sparsity and energy of jamming. We limited the number of maximum iterations by K, where the cardinality of the support set is known as K. At the same time, we set a lower-bound on $\|\boldsymbol{\rho}\|$ by δ. If the prior knowledge is inexact for determining a proper termination condition, the GSJE might not be able to estimate the exact jamming. The assumption that the receiver has exact prior information a priori might be unrealistic in practice as the jammer and the receiver are adversarial to each other. These effects are of interest in the discussion of our simulation results (refer to Section 3.8.1).

Algorithm 6. Greedy sparse jamming estimation (GSJE) algorithm

Inputs:	$\mathbf{H}$, $\mathbf{G}$, $\mathbf{y}$, δ, K
1:	**Initialize:** $\mathbf{A} \leftarrow \mathrm{svd}(\mathbf{G}^H\mathbf{H}^H)$, $\boldsymbol{\rho} \leftarrow \mathbf{A}\mathbf{y}$, $S \leftarrow \{\}$
2:	**do:**
3:	Find the maximum correlated column index, $k \leftarrow \arg\max_{k \in \mathbb{S}} < \mathbf{a}_k, \boldsymbol{\rho} >$
4:	Update the support set by storing the index in the support set, $\mathbb{S} \leftarrow \mathbb{S} \cup \{k\}$
5:	Estimate the corresponding coefficient subvector by $\hat{\mathbf{e}}_{\mathbb{S}} \leftarrow (\mathbf{A}_{\mathbb{S}}^H \mathbf{A}_{\mathbb{S}})^{-1} \mathbf{A}_{\mathbb{S}}^H \mathbf{A}\mathbf{y}$
6:	Update residual by subtracting the contribution of current support set, $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} - \mathbf{A}\hat{\mathbf{e}}$
7:	**while** $\|\mathbb{S}\| \leq K$ and $\|\boldsymbol{\rho}\| \geq \delta$
8:	**return:** $\hat{\mathbf{e}}$

B. Bayesian Sparse Jamming Estimation Algorithm

As an alternative to GSJE, we propose a novel BSJE algorithm. In the previous section, we discussed that GSJE requires prior knowledge of jamming, such as the sparsity or power of jamming. Instead, we aim to develop the BSJE method in which the jamming is modeled as a Gaussian mixture and Bayesian inference is sought to estimate the jamming vector. From (61), BSJE determines the posterior estimation of $\mathbf{e}$ by exploiting the assumption that $\mathbf{e}$ is a realization of a zero-mean i.i.d. Gaussian-distributed random vector of which the covariance matrix is $\mathbf{\Gamma} = diag[\gamma_1 \quad \gamma_2 \quad ...\cdots \quad \gamma_N]$ (i.e., $\mathbf{e} \sim \mathcal{CN}(\mathbf{0}, \mathbf{\Gamma})$). A simple suboptimal solution is to determine the maximum a posteriori probability (MAP) estimation without exploiting $\mathbf{\Gamma}$, i.e.,

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} p(\mathbf{e} \mid \mathbf{b}), \tag{63}$$

where $p(\mathbf{e} \mid \mathbf{b})$ is a conditional PDF. In contrast, the BSJE algorithm exploits the information about $\mathbf{\Gamma}$ by applying SBL [24]. Let the ML estimation of $\mathbf{\Gamma}$ be $\hat{\mathbf{\Gamma}}$. The algorithm first determines $\hat{\mathbf{\Gamma}}$ and then evaluates the MAP estimator for a given $\hat{\mathbf{\Gamma}}$. That is,

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} p(\mathbf{e} \mid \hat{\mathbf{\Gamma}}) p(\hat{\mathbf{\Gamma}}) p(\mathbf{b} \mid \mathbf{e}), \tag{64}$$

where

$$\begin{aligned} \hat{\mathbf{\Gamma}} &= \arg\max_{\mathbf{\Gamma}} p(\mathbf{\Gamma} \mid \mathbf{b}) \\ &= \arg\max_{\mathbf{\Gamma}} p(\mathbf{\Gamma}) p(\mathbf{b} \mid \mathbf{\Gamma}) \\ &= \arg\max_{\mathbf{\Gamma}} p(\mathbf{\Gamma}) \int p(\mathbf{b} \mid \mathbf{e}) p(\mathbf{e} \mid \mathbf{\Gamma}) d\mathbf{e}. \end{aligned} \tag{65}$$

To solve the optimization problem, the spirit of expectation-maximization (EM) is exploited; rather than directly maximizing the log-likelihood, the algorithm maximizes the *Q* function iteratively. The *Q* function is defined by

$$Q(\mathbf{\Gamma}) = E_{\mathbf{e}|\mathbf{b};\gamma}\left[\log p(\mathbf{b} \mid \mathbf{e}) + \log p(\mathbf{e} \mid \mathbf{\Gamma}) + \log p(\mathbf{\Gamma})\right], \tag{66}$$

and the solution to (65) is equivalent to the maximizer of (66),

$$\hat{\mathbf{\Gamma}} = \arg\max_{\mathbf{\Gamma}} Q(\mathbf{\Gamma}). \tag{67}$$

Let us recall the SJE problem in (61). The given problem is a form of the so-called input noise model:

$$\mathbf{b} = \mathbf{A}(\mathbf{e} + \mathbf{w}). \tag{68}$$

The closed-form solution to (67) is well known as

$$\hat{\gamma}_k = E_{\mathbf{e}|\mathbf{b},\gamma_k,\sigma^2} \mid e_k \mid^2, \quad (k = 1, 2, ..., N), \tag{69}$$

which is a solution to $\partial Q(\mathbf{\Gamma}) / \partial \gamma_k = 0$ by assuming $p(\gamma_k) = 1$ and approximating $Q(\mathbf{\Gamma})$ as

$$Q(\boldsymbol{\Gamma}) \approx E_{\mathbf{e}|\mathbf{b};\gamma,\sigma^2}\left[\sum_k -\frac{1}{2}\log\gamma_k - \frac{|e_k|^2}{\gamma_k} + \log p(\gamma_k)\right]. \tag{70}$$

If we assume a CSCG jamming $\mathbf{e} \sim \mathcal{CN}(\mathbf{0}, \boldsymbol{\Gamma})$ and a CSCG noise $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ with given noise variance σ^2, the conditional PDF $p(\mathbf{e}|\mathbf{b};\boldsymbol{\Gamma},\sigma^2)$ becomes CSCG, with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, where

$$\boldsymbol{\mu} = \boldsymbol{\Gamma}\mathbf{A}^H(\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^H + \sigma^2\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{b}, \tag{71}$$

$$\boldsymbol{\Sigma} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\mathbf{A}^H(\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^H + \sigma^2\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{A}\boldsymbol{\Gamma}. \tag{72}$$

Thus, the ML estimator in (21) becomes

$$\hat{\gamma}_k = \Sigma_{kk} + |\mu_k|^2, \tag{73}$$

where μ_k is the k-th element of $\boldsymbol{\mu}$ and Σ_{kk} is the k-th diagonal element of $\boldsymbol{\Sigma}$. According to (71)-(73), $\hat{\boldsymbol{\Gamma}}$ can be determined by iteratively updating $\boldsymbol{\Gamma}^{(t)}$ at the t-th iteration, and by repeating the iteration until convergence. A little is known about the convergence of EM iteration, including the monotonicity property [110]. This property states that the solution converges to at least a saddle point or local optimum. After $\hat{\boldsymbol{\Gamma}}$ is converged, the MAP estimation is obtained by

$$\begin{aligned}\hat{\mathbf{e}} &= \arg\max_{\mathbf{e}} p(\mathbf{e}|\mathbf{b};\hat{\boldsymbol{\Gamma}},\sigma^2) \\ &= \boldsymbol{\mu}_{\mathbf{e}|\mathbf{b}}(\hat{\boldsymbol{\Gamma}}) \\ &= \hat{\boldsymbol{\Gamma}}\mathbf{A}^H(\mathbf{A}\hat{\boldsymbol{\Gamma}}\mathbf{A}^H + \sigma^2\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{b}.\end{aligned} \tag{74}$$

It can be recalled that our goal is to determine the complex-valued codeword $\mathbf{x}$, not $\mathbf{e}$ itself, which is obtained by the iteration above. To estimate, we must determine not only $\hat{\mathbf{e}}$ but also $\hat{\mathbf{w}}$; thus, $\hat{\mathbf{x}}$ can be calculated by

$$\hat{\mathbf{x}} = \mathbf{H}^{-1}(\hat{\mathbf{e}} + \hat{\mathbf{w}}). \tag{75}$$

Here, $\hat{\mathbf{w}}$ can be obtained during the BSJE iteration using the same method used for determining $\hat{\mathbf{e}}$, as $\hat{\mathbf{w}}$ also follows a CSCG distribution such that $\mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$. The MAP estimation of $\mathbf{w}$ can be calculated within the same iteration for determining $\hat{\mathbf{e}}$ in (71):

$$\hat{\mathbf{w}} = \sigma^2\mathbf{A}^H(\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^H + \sigma^2\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{b}. \tag{76}$$

Note that (71)–(72) are equivalent to the equations presented by Giri and Rao (numbered as (36) and (37), respectively, in their paper) [111] if $\mathbf{A}^H$ is orthonormal. This implies that if all codewords are orthonormal with each other and the channel matrix $\mathbf{H}$ is an identity matrix (i.e., a frequency-nonselective channel), the proposed BSJE algorithm is equivalent to the SBL algorithm for the general measurement noise model.

The BSJE algorithm for the proposed system is described as Algorithm 7. To guarantee the completeness of the algorithm, two parameters, $t_{\max}$ and δ, can be specified. $t_{\max}$ is defined by the maximum number of iterations and δ is the lower-bound on the minimum required update, $\mathrm{tr}(|\mathbf{\Gamma}^{(t)}-\mathbf{\Gamma}^{(t-1)}|)$, where $\mathrm{tr}()$ is the trace operator. These two parameters can be set according to the time budget.

The BSJE algorithm does not require prior knowledge about jamming. The BSJE algorithm only requires the channel, codebook, and noise variance as inputs, which can be easily obtained by a legitimate receiver. This mild requirement on prior information is a potential benefit of BSJE, making it more attractive than GSJE in practical implementation. Furthermore, the BSJE algorithm has no model-mismatch problem in the arbitrary JNR condition region, whereas GSJE suffers from estimation performance degradation in the moderate-JNR region.

Algorithm 7. Bayesian sparse jamming estimation (BSJE) algorithm

Inputs:	$\mathbf{H}$, $\mathbf{G}$, σ^2, $t_{\max}$, δ
1:	**Initialize:** $\mathbf{A} \leftarrow \mathrm{svd}(\mathbf{G}^H\mathbf{H}^H)$, $\mathbf{b} \leftarrow \mathbf{A}\mathbf{y}$, $\mathbf{\Gamma}^{(0)} \leftarrow \mathbf{I}$
2:	**for** $t=0$ to $t_{\max}$ **do:**
3:	Compute $\mathbf{T}$ matrix by $\mathbf{T} \leftarrow \mathbf{A}^H(\mathbf{A}\mathbf{\Gamma}^{(t)}\mathbf{A}^H+\sigma^2\mathbf{A}\mathbf{A}^H)^{-1}$
4:	Update the mean vector $\boldsymbol{\mu} \leftarrow \mathbf{\Gamma}^{(t)}\mathbf{T}\mathbf{b}$
5:	Update the noise vector $\mathbf{w} \leftarrow \sigma^2\mathbf{T}\mathbf{b}$
6:	Update the covariance matrix $\mathbf{\Sigma} \leftarrow \mathbf{\Gamma}^{(t)}-\mathbf{\Gamma}^{(t)}\mathbf{T}\mathbf{A}\mathbf{\Gamma}^{(t)}$
7:	Update the hyperparameter $\mathbf{\Gamma}^{(t+1)} \leftarrow diag(\lvert \mu_k \rvert^2+\Sigma_{kk})\rvert_{k=1,2,\ldots,N}$
8:	**while** $t \le t_{\max}$ and $\mathrm{tr}(\lvert \mathbf{\Gamma}^{(t)}-\mathbf{\Gamma}^{(t-1)} \rvert) \ge \delta$
9:	**return:** $\tilde{\mathbf{e}} \leftarrow \boldsymbol{\mu}+\mathbf{w}$

3.5.5. Decoding

Using GSJE and BSJE, we can successfully estimate the jamming with a noise vector $\tilde{\mathbf{e}}$. Then, Bob can obtain the jamming-mitigated complex-valued codeword $\tilde{\mathbf{x}}$ by subtracting the estimation of $\tilde{\mathbf{e}}$ from $\mathbf{y}$. The corresponding real-valued codeword $\tilde{\mathbf{c}}$ is obtained by cascading the real and imaginary parts of $\tilde{\mathbf{x}}$. Decoding the original codeword $\mathbf{m}$ from $\tilde{\mathbf{c}}$ is typically performed using MAP estimation, which is a procedure to obtain the most likely input for a given output, i.e.,

$$\hat{\mathbf{m}} = \arg\max_{\mathbf{m}} p(\mathbf{m};\tilde{\mathbf{c}}) . \tag{77}$$

Assuming a message $\mathbf{m}$ has a uniform distribution, the MAP estimator is equivalent to the ML estimator [112]. The ML estimator of $\mathbf{m}$ is defined by $\hat{\mathbf{m}}$, which maximizes the likelihood function. This relationship is represented as follows:

$$\hat{\mathbf{m}} = \arg\max_{\mathbf{m}} p(\hat{\mathbf{c}}';\mathbf{m}) . \quad (78)$$

3.6. Undetectability Analysis

In this section, we present a comparison of the undetectability of the Gaussian-coded time-frequency modulation and binary modulation methods in terms of detection probability for Eve. To measure undetectability, a privacy rate was proposed in [18]. The privacy rate is defined by the number of bits that can be transmitted covertly through the use of N channels. According to [18], only $O\left(\sqrt{N}\right)$ bits can be transmitted covertly; thus, the privacy rate cannot be a constant. However, [19], [112] determined that a constant privacy rate is achievable if Eve is uncertain of the channel noise level. Conversely, detection probability is a simple and useful metric of undetectability. For example, the detection probability of chaotic-sequence DSSS signals was studied in [35], [36]. The authors assumed that Eve knows most of the protocols used by target communication systems, such as carrier frequency, modulation, length of the spread sequence, and symbol duration, but not the exact spreading sequence. Then, she can attempt matched filtering with all possible spreading sequences for the received samples to determine if a signal exists. The studies determined the detection probability of chaotic DSSS signals for several types of chaotic sequences and compared the probability with that of conventional binary sequence DSSS signals.

To measure undetectability, we considered the detection probability as a metric. The eavesdropping problem was formulated as a hypothesis test. We explored the problem by dividing it into three cases according to uncertainty of noise for Eve. In the first case, Eve is assumed to have no information on the noise level. That is, Eve has an infinitely large uncertainty in terms of the noise level. In such extreme cases, the only option for Eve is to test whether the received samples are from a white Gaussian distribution or from other random distributions, i.e., Eve aims to determine the existence of a signal using a normality test. The second case is the other extreme scenario on the opposite side. In the second case, Eve has no uncertainty regarding the noise level. Thus, Eve knows the exact noise energy. In such a case, Eve can apply a hypothesis test based on a threshold of the symbol energy to distinguish whether the signal exists or not. However, the two extreme cases rarely occur in practice. Thus, in the third case, we define a parameter ρ to quantify the amount of uncertainty Eve has on the noise level. In the following subsections, we measure and compare the undetectability of the proposed signal and conventional signals.

3.6.1. Undetectability under Unknown Noise Level

In the first case, Eve is assumed to have no information on the noise level (Case 1). In the following section, we provide a simple analysis and example showing that Eve cannot distinguish the proposed signal, although she has a chance of detecting conventional signals. Let us recall the channel model depicted in Figure 18 to determine the eavesdropping problem. Using the same vectorizing process as the legitimate channel model as presented in (58) and (59), Eve obtains

$$\mathbf{z} = \mathbf{H}'\mathbf{x} + \mathbf{w}', \tag{79}$$

where $\mathbf{H}'$ is a combined wiretap channel matrix and $\mathbf{w}'$ is a CSCG noise vector, which are derived using the same method as those of a legitimate channel, as discussed in Section 3.6.2.

Consider that Eve monitors the communication channel and obtains a measurement vector assuming an exact carrier frequency, phase, and symbol duration. Moreover, consider a perfect channel state information and flat fading at Eve, which is a threat scenario for a WCS. Then, the receiving model is simplified to

$$\mathbf{z} = h'\mathbf{x} + \mathbf{w}', \tag{80}$$

where h' is the channel coefficient satisfying $\mathbf{H}' = h'\mathbf{I}$. In this scenario, Eve has to determine whether there is a signal using the hypothesis test:

$$\begin{cases} H_0 : \mathbf{d} = \mathbf{n} \\ H_1 : \mathbf{d} = \mathbf{c} + \mathbf{n}, \end{cases} \tag{81}$$

where $\mathbf{c} = [real(\mathbf{x})^T \quad imag(\mathbf{x})^T]^T$ is the real-valued codeword of which the entries are zero-mean Gaussian random variables with variance σ_c^2, $\mathbf{n} = [real(\mathbf{w}'/h')^T \quad imag(\mathbf{w}'/h')^T]^T$ is the AWGN whose entries are zero-mean Gaussian random variables with variance σ_n^2, and $\mathbf{d} = [real(\mathbf{z}/h')^T \quad imag(\mathbf{z}/h')^T]^T$ is the channel-compensated measurement vector. In Case 1, Eve has to determine which hypothesis is true based on the PDF of $\mathbf{d}$. Let the variance of $\mathbf{d}$ be $\sigma_{\mathbf{d}}^2$. To test the two hypotheses, the only thing Eve can do is compare how close the sample distribution is to the distribution of Gaussian noise. This test is well known as the normality test. First, assume that Eve demonstrates perfection while calculating the sample distribution. Perfect here implies that Eve can collect an infinite number of samples each with infinite resolution; thus, the histogram distribution of the perfect Eve is identical to the distribution according to the true hypothesis.

Lemma 1. *Assume that Eve is perfect while calculating the sample distribution. Let Eve not know the noise level. Then, she cannot detect the existence of the proposed signal, but she can detect finite-field modulated signals.*

Proof of Lemma 1. Eve does not know the noise level. The addition of the Gaussian signal sample to the Gaussian noise sample produces a Gaussian sample. Thus, what Eve observes are Gaussian samples; Eve can only treat it

as Gaussian noise. When Eve observes finite-level signal samples, a perfect Eve will notice that the observed signal deviates from the Gaussian samples. Thus, Eve can detect the presence of an ongoing communication signal.

It is worth noting that Lemma 1 is satisfied in the ideal case. Namely, it holds when the degree of freedom of the codebook is infinite and Eve can correct an infinite number of samples. However, both conditions cannot be perfect in practice. The degree of freedom of a fixed codebook is limited by the size of the codebook. It causes a distortion between the distributions of the actual symbol and desired Gaussian symbols. In addition, Eve must perform the normality test with a finite number of samples and a finite level of precision. This means that the error probability of the normality test is not zero even for the Gaussian signal. However, the distortion of the distribution can be compensated for by the limited resolvability [113] of the channel between the transmitter and Eve. Since the channel is noisy, small distortions of the input distribution cannot affect the test output of Eve. Moreover, despite all these limitations, it is obvious that the probability of being detected for the proposed signal is far lower than the probability of any conventional non-Gaussian signal.

3.6.2. Undetectability under Exact Noise Level

Now, let us focus on Case 2. In this case, Eve is assumed to have perfect information on the noise level. The simplest solution for this testing strategy is to use an energy detector of which the goal is to determine whether the signal is present by comparing the energy of the received symbol.

In the energy detection, the test statistic of (81) becomes the squared sum of symbols. As we assumed, Eve does not know the exact time-frequency band of the target signals. Then, Eve has two strategies for testing. The first is obtaining a single test statistic by integrating the whole energy over a suspected time-frequency band. It is obvious that the test statistic depends on both the SNR and the suspected region, regardless of the coding and modulation method of the target signal. Then the test result depends only on how accurate the suspected region is. Rather than this trivial case, we focus on the other strategy: dividing the suspected region with as fine a grid as Eve can produce, and applying per-symbol energy detection to determine the signal existence according to each time-frequency slot. By this strategy, Eve can evaluate the slot-by-slot possibility of signal presence. Based on this possibility, Eve can jointly estimate the suspected region and probability of signal presence. The performance of this joint detection of possibility and region depends on the per-symbol test. From this aspect, we derive the true positive of the per-symbol hypothesis test (i.e., probability of detection). We compare the probability for the proposed method and that for the conventional binary method, which follows Lemma 2.

Lemma 2. *Let Eve, using an energy-detecting strategy, know the exact noise level. If Eve has a false alarm*

probability P_F, *then the per-symbol detection probabilities of the proposed signal* $P_{D,Gaussian}$ *and BPSK-modulated signal* $P_{D,Binary}$ *are*

$$\begin{aligned} P_{D,Binary} &= Q\left(\alpha - \sqrt{\beta}\right) + Q\left(\alpha + \sqrt{\beta}\right), \\ P_{D,Gaussian} &= 2Q\left(\frac{\alpha}{\sqrt{1+\beta}}\right), \end{aligned} \tag{82}$$

where $\alpha = Q^{-1}(P_F/2)/\sigma_n$, $\beta = \sigma_c^2/\sigma_n^2$, *and the Q-function is defined by*

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt \,. \tag{83}$$

Proof of Lemma 2. In Case 2, Eve has information about the wiretap channel noise level. Without the loss of generality, the hypothesis test in (81) is simplified into a per-symbol hypothesis test, as shown below:

$$\begin{cases} H_0' : d = n \\ H_1' : d = c + n, \end{cases} \tag{84}$$

where d, c, and n are the corresponding single samples of $\mathbf{d}$, $\mathbf{c}$, and $\mathbf{n}$. We determine a threshold α and use the following likelihood ratio test to determine which hypothesis is true:

$$|d| \underset{H_0}{\overset{H_1}{\gtrless}} \alpha\sigma_n \,. \tag{85}$$

The threshold α is selected according to the desired false alarm probability P_F. Because the noise is Gaussian, α can be calculated using the Q-function:

$$\alpha = Q^{-1}\left(\frac{P_F}{2}\right). \tag{86}$$

We can now compare the probability of detection for the traditional binary modulation and proposed modulation methods. The probability of detection can be calculated as the $\alpha\sigma_n$-tail probability of the distribution of d given that H_1' is true, through a process similar to that used in [35], [36]. For a binary modulation scheme of which the constellation is distributed uniformly at the points $\pm\sigma_c$, the distribution of d when H_1' is true becomes

$$f_b(x)=\frac{1}{2\sqrt{2\pi\sigma_n^2}}e^{-\frac{(x-\sigma_c)^2}{2\sigma_n^2}}+\frac{1}{2\sqrt{2\pi\sigma_n^2}}e^{-\frac{(x+\sigma_c)^2}{2\sigma_n^2}}. \tag{87}$$

The probability of detection for a uniformly distributed binary signal is

$$P_{D,Binary}=Q\left(\frac{\alpha\sigma_n-\sigma_c}{\sigma_n}\right)+Q\left(\frac{\alpha\sigma_n+\sigma_c}{\sigma_n}\right). \tag{88}$$

In contrast, consider the proposed signal of which the sample value obeys a Gaussian random distribution with zero mean and variance σ_c^2. Then, the PDF of $(d\,|\,H_1')$ is also a Gaussian PDF with zero mean and $\sigma_c^2+\sigma_n^2$ variance. The α-tail probability is calculated directly as

$$P_{D,Gaussian}=2Q\left(\frac{\alpha\sigma_n}{\sqrt{\sigma_n^2+\sigma_c^2}}\right). \tag{89}$$

Letting $\beta=\sigma_c^2/\sigma_n^2$, the two probabilities obtained using (87)-(88) are equivalent to (82). ■

Lemma 2 shows that if the normalized SNR of Eve is $\beta\ll1$, the two probabilities are nearly identical. In contrast, if $\beta\to\infty$, the detection probability of a Gaussian signal is lower than that of a binary signal. Thus, if the SNR of the channel between Alice and Eve is higher, a Gaussian signal is superior to a binary signal in terms of undetectability. At a moderate SNR, the false alarm probability of Eve (parameter α) determines the detection probability. As the false alarm probability increases (i.e., Eve is more sensitive), a Gaussian symbol becomes more undetectable than a binary symbol.

3.6.3. Undetectability under Uncertain Noise Level

In practice, Eve must estimate the noise level, σ_n^2, to determine the proper threshold, α. The estimation error affects the tradeoff between the detection probability and the false alarm probability. Let the estimated noise variance be $\sigma_{est}^2:=\rho\sigma_n^2$ using an uncertainty parameter, $\rho\in[\rho_{\min},\rho_{\max}]$. Under this definition, the uncertainty increases when the interval $[\rho_{\min},\rho_{\max}]$ grows. In contrast, the uncertainty becomes smaller when the interval becomes narrower, and finally becomes zero when $\rho_{\min}=\rho_{\max}=1$. In this scenario, Eve sets the parameter α by using σ_{est}^2 instead of σ_n^2 to set the decision threshold, $\alpha\sigma_{est}$. Then, the detection and false alarm probabilities in Lemma 2 change according to ρ, as shown below.

Lemma 3. *Let Eve, using an energy-detecting strategy, estimate the noise level to be* $\sigma_{est}^2 := \rho\sigma_n^2$ *with the uncertainty parameter* ρ. *Then, for the target false alarm probability of Eve,* $P_{F,\text{target}} = 2Q(\alpha)$, *the per-symbol detection probabilities of the proposed signal* $P_{D,Gaussian}$ *and BPSK-modulated signal* $P_{D,Binary}$ *are*

$$\begin{aligned} P_{D,Binary} &= Q\left(\alpha\sqrt{\rho} - \sqrt{\beta}\right) + Q\left(\alpha\sqrt{\rho} + \sqrt{\beta}\right), \\ P_{D,Gaussian} &= 2Q\left(\frac{\alpha\sqrt{\rho}}{\sqrt{1+\beta}}\right), \end{aligned} \tag{90}$$

and the actual false alarm probability becomes $P_F = 2Q(\alpha\sqrt{\rho})$.

One can easily prove Lemma 3 by calculating the $\alpha\sigma_{est}$-tail probability using the same procedure as used in (87)-(89). Lemma 3 shows that the uncertainty of the noise level of Eve only causes a shift in the tradeoff between the detection probability and the false alarm probability. This implies that P_F increases and P_D decreases if Eve underestimates ($\rho<1$) the noise level; in contrast, P_F decreases and P_D increases if Eve overestimates ($\rho>1$) the noise level. The results for different ρ are illustrated in Figure 19.

In conclusion, Lemma 1 shows that the proposed method is more undetectable than the finite-field modulation method, regardless of the eavesdropping scenario. Lemmas 2 and 3 show that the proposed method is more undetectable than the traditional binary modulation method, especially when the eavesdropping scenario is more threatening.

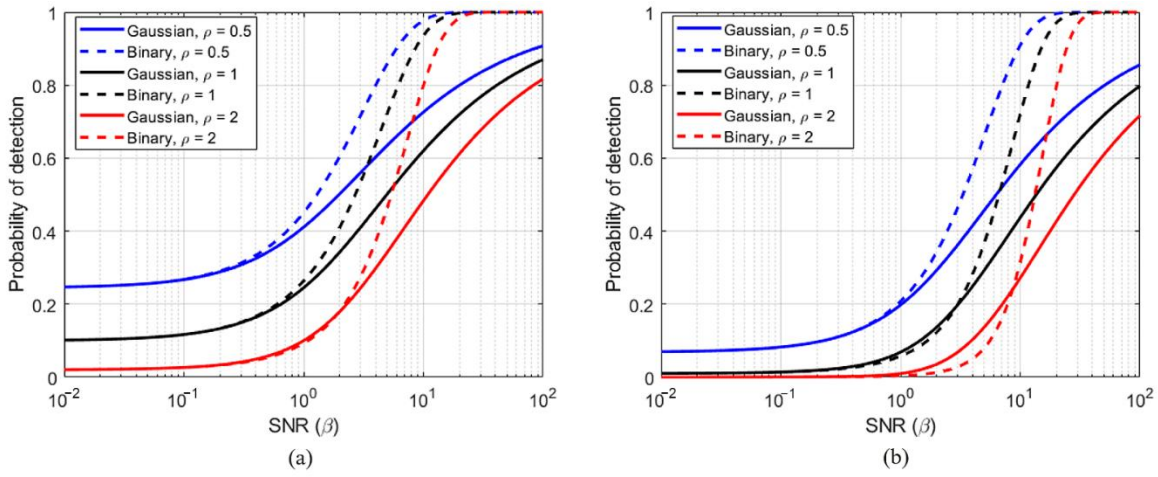


Figure 19. Per-symbol detection probability of proposed Gaussian and conventional binary symbols: (a) false alarm probability of 0.1 (α=1.645) and (b) false alarm probability of 0.01 (α=2.576)

3.7. Implementation Problems

Showing how the proposed system is implemented in real time is important. In this section, we aim to present two problems in real-time implementation and address solutions for each. The first is modulation. The proposed system achieves undetectability from Gaussian-distributed codeword samples. However, digital signal processing units do not usually support Gaussian samples. Therefore, in Section 3.8.1, we discuss how the Gaussian samples can be represented with finite precision, and determine the number of quantization levels required for sufficient precision. The possible adoption of commercially applicable solutions such as analog-to-digital converters and digital signal processing units are discussed as well in order to tackle this finite precision problem. The second problem is computational complexity. The SSR algorithms include the matrix inversion operation in each iteration. In Section 3.8.1, we aim to show that hardware/software solutions such as parallel processing, field-programmable gate array (FPGAs), and application-specific integrated circuits (ASICs) can achieve high throughput and satisfy power constraints.

3.7.1. Modulation Method

In our proposed system, we used Gaussian samples. This has a noise-mimicking property that is highly likely to remain undetected by the watchful Eve. This undetectable characteristic of the Gaussian code is bolstered with the time-frequency modulation, which is not available with traditional digital modulation constellations, such as quadrature phase-shift keying (QPSK) and quadrature amplitude modulation (QAM). We can recall that in Section 4, the proposed method showed higher undetectability than the traditional method, especially in the scenario in which Eve is the most threatening.

The proposed system has to be implemented in signal processing processors that have a finite dynamic range and a finite number of precision levels. Hence, truncation and quantization must be considered when implementing the Gaussian constellation. For several decades, optimal quantization methods have been studied in signal processing and machine learning communities. Well-known quantization methods include Lloyd's algorithm I [114]. It determines a partition with respect to k mean points or quantization points and then updates these points by calculating the center of each partition. Generalized vector quantization [115] is a relaxation of Lloyd's algorithm. It combines Lloyd's algorithm with a global optimization method to avoid local minimums. There are several other heuristic algorithms for determining optimal quantization, such as competitive learning vector quantization [116].

The most important parameter for these methods is the number of quantization points, which determines the truncation limits and quantization precision. If the number of quantization points is large, the

undetectability performance of the quantized Gaussian signal becomes closer to that of an ideal Gaussian signal. As the number of quantization points increases, a more precise analog-to-digital converter is required.

The quantized Gaussian constellation methods allow us to transmit and receive an approximation of the proposed Gaussian-coded signal by bounding the maximum magnitude with finite values. For example, Pages and Printems [116] designed a 500-point quantized constellation that closely approximates the ideal Gaussian signal. For this constellation, the maximum magnitude of a sample is bounded by $3\sqrt{E_s}$, where E_s is the average energy of the signal constellation. This result shows that the proposed method with quantization can be achieved in modern radios. For example, the commercial mobile radio platform of Qualcomm, called Snapdragon 855, includes a long-term evolution modem that supports 256-QAM [117]. When 256-QAM is used, the maximum magnitude of a sample is bounded by $1.6\sqrt{E_s}$. Thus, the dynamic range required for a quantized Gaussian constellation is not significantly far from the dynamic range of the specifications of commercial hardware.

If the quantized Gaussian signal is used instead of the true Gaussian signal, performance degradation might occur due to the gap between the two distributions. However, the difference will be negligible if the quantization precision is sufficiently high. In Table 4, the results of the Anderson–Darling test for samples from a quantized and truncated Gaussian distribution are listed. If the quantized Gaussian distribution is close to the real Gaussian distribution, the test returns 0; otherwise, it returns 1. In the numerical simulation, 1024 realizations of 8-, 10-, 12-, 14-, and 16-bit quantized distributions were considered; moreover, a [–3, 3]-truncated standard normal distribution was tested. The results of the tests were averaged from 10,000 simulations per simulation point. The results show that with a significance level of 5%, the probability of distinguishing between the samples from 12-bit quantization and the samples from the true Gaussian distribution is as small as 6.24%. It can be noted that a $1000 software-defined radio device includes a 12-bit digital-to-analog converter [118]. Thus, we can conclude that the quantized Gaussian signal is sufficiently close to the real Gaussian signal despite the hardware limitation.

Table 7. Average results of the Anderson-Darling test for quantized Gaussian distribution

Quantization Level [bits]	16	14	12	10	8
Average Return	0.0557	0.0583	0.0624	0.0661	0.1969

3.7.2. Computational Complexity and Power Consumption

The computational complexity is another important consideration for real-time implementation. The receivers of the proposed system include the SSR algorithm for solving the SJE problem and removing it from the

received vector. The additional computation of the SSR algorithm requires a certain amount of processing time and power consumption. On the one hand, for the GSJE algorithm, the dominant complexity term is that of the least-squares method for obtaining nonzero coefficients. The computational complexity of the least-squares or the equivalent matrix inversion method is $O(NK)$ per iteration, because the Cholesky decomposition of $\mathbf{A}_{\mathbb{S}}^{H}\mathbf{A}_{\mathbb{S}}$ can be precalculated for all possible support sets, $\mathbb{S}$. The amount of overall computations for K number of iterations is $O(NK^2)$. Conversely, the BSJE algorithm has to calculate an inversion of the matrix $(\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{H}+\sigma^2\mathbf{A}\mathbf{A}^{H})^{-1}$ for each iteration, resulting in $O(MN)$ computations, where $\mathbf{A}\in\mathbb{C}^{M\times N}$ and $M<N$. We now discuss the complexity of two other conventional covert anti-jamming methods. The simplest conventional method is the DSSS, which is compared with the proposed method in Section 3.7. The receiver for DSSS requires $O(N)$ computations to perform a matched filtering of length $2N$. The other conventional method, FHSS-type cognitive radio [119], [120], also requires $O(N)$ computations for spectrum sensing of N_F subbands within $O(N_T)$ hops. The results show that the complexity of the SSR algorithm is marginally higher than that of these conventional methods. However, owing to the recent advances in software and hardware solutions, this complexity increase does not pose any limitation to real-time implementation. We aim to show this in the following section.

Let the block size be $N=2^9$ for transmitting a message of length $l_m=16$. Then, the computational complexity of the GSJE algorithm is approximately $2^{23}\approx 8\times 10^6$ floating-point operations per second (FLOPS), assuming the worst case where $K=N/4$. We can use serial processing with modern digital signal processing chips to compute the algorithm. For example, a single core of the TMS320C6678 digital signal processor offers a few tens of giga-FLOPS. When we employ these chips, we can process a few thousand blocks in real time, while consuming 1 W [121]. This implies that the throughput that the proposed transceiver offers can easily support communication speeds on the order of kilobits per second (kbps), while offering covert communication with strong anti-jamming protection.

We can increase the speed of the SSR algorithm. To achieve this, we can utilize recent advances in research on parallel computation for matrix inversion and multithread processing on GPUs. Examples include Gauss–Jordan-based implementations [122] and squared Givens rotation-based implementations [123]. Sharma *et al.* [122] claimed that the time complexity of the matrix inversion decreased up to $O(N)$ and Yu *et al.* [123] showed that their CPU/GPU-combined implementations achieved more than 20 times higher throughput than CPU-only implementations. These examples suggest the possibility of a real-time implementation on the order of

megabits per second based on a GPU or a cluster of GPUs. However, the challenge of power consumption remains. For example, the GPU cores used in [123] consume a maximum of 49 W. These additional power requirements might not be bearable for certain power-critical applications, such as a mobile radio device.

To satisfy the power constraints of real-time applications, hardware solutions such as FPGAs and ASICs can be utilized. FPGA implementations [124], [125] can process the SSR algorithm sufficiently fast, within the order of 10 μs. The implementations in [124], [125] run the SSR algorithm for $N=128$ within 18.3 μs and $27.0\mu s$, respectively. If we assume $N=128$ and $L=4$, the throughputs of the implementations become 218.6 and 148.1 kbps, respectively. ASICs can be used to further increase the throughput. A 65-nm complementary metal-oxide-semiconductor ASIC design [126] can process the GSJE algorithm for $N=256$ within 591.36 ns. By assuming $N=256$ and $L=8$, a throughput of 13.53 Mbps can be achieved.

It may appear that adding an ASIC for the sole purpose of eliminating jamming is too expensive. However, it must be noted that hardware costs decrease rapidly each year. Furthermore, it should be noted that achieving covert communication capability with anti-jamming protection in any battlefield situation is highly desired. Imagine a situation in which the entire communication link between allied forces is disabled and destroyed by hostile jamming. Multiple studies emphasize how detrimental it is to have even a minor information loss in mission-critical military communications [127], [128].

3.8. Results

In this section, we present numerical results to demonstrate the anti-jamming performance of the proposed system. First, we compare the SJE qualities of the two algorithms, GSJE and BSJE, in terms of the mean squared error (MSE). Second, we demonstrate the BER at the receiver under a jamming attack using several channels (AWGN and frequency-selective fading), coding methods (linear Gaussian and codebook), and jamming estimation algorithms (GSJE and BSJE).

3.8.1. Sparse Jamming Estimation Error

We evaluated the SJE performance of the GSJE and BSJE algorithms for the proposed system. In general, the BSJE algorithm shows better SJE performance than the GSJE. However, as the two algorithms have advantages and disadvantages, we must select an algorithm for practical applications.

- Limitations in BSJE implementation

In practical implementations, the appropriate maximum number of iterations should be chosen to ensure the completeness of the iterative algorithm. As the maximum number of iterations for BSJE is empirically

selected, the output of the algorithm may not sufficiently converge to the solution. In contrast, GSJE has a fixed maximum number of iterations, which is N in the worst case. Owing to the aforementioned implementation limitations of BSJE, there are several regions in which the BSJE algorithm performs equivalently to, and sometimes worse than, GSJE.

- Time complexity

The BSJE algorithm calculates several matrix products and matrix inversions in a single iteration, whereas the GSJE algorithm requires only a single matrix product and the inversion of a smaller submatrix. In general, GSJE is significantly faster than BSJE, especially when the size of the system becomes larger.

- Level of prior knowledge in the GSJE algorithm

The above two problems discourage the use of the BSJE algorithm. However, the GSJE algorithm requires prior knowledge about the signal to define its termination conditions, and this assumption about prior knowledge might be unrealistic. For example, if the maximum cardinality of the support set is given as the termination condition, a receiver executing the GSJE algorithm must know the support set of the jammer. This assumption that the receiver will know the information a priori is impractical. If this prior information is unknown, the optimum K is the maximum possible number of jamming coefficients for GSJE. Unfortunately, this value depends on the RIP of the annihilating matrix $\mathbf{A}$. Since $\mathbf{A}$ is constructed from the random matrix $\mathbf{G}$, the K must be calculated every time the codebook changes. However, evaluating the RIP takes considerable time. The overhead further increases since the codebook must vary over time to guarantee security. Thus calculating the optimum K for each codebook change is not practicable. Instead, we heuristically evaluated the average RIP for multiple realizations of $\mathbf{A}$, and roughly set $K = N/4$ by applying Wakin's bound [129].

Accordingly, we performed a numerical simulation to compare the GSJE and BSJE algorithms. The two algorithms were compared in terms of jamming estimation performance (as measured by MSE) and time complexity (as measured by running time) at various JNRs and levels of prior knowledge for the GSJE algorithm. We set the level of prior knowledge as how much Bob precisely knows about the cardinality of jamming (e.g., sparsity, K). The measure of prior knowledge is classified into two levels as a function of the true value of K, as listed in Table 5.

Table 8. Definition for the level of prior knowledge

Level of Prior Knowledge	Perfect	Unknown
Maximum iteration of GSJE	$\hat{K} = K$	$\hat{K} = N/4$

Using the values in Table 5, we compared the normalized MSE of the jamming estimation and the corresponding running time. The block length of the complex-valued codeword was set to 1024, and the number of possible message vectors was eight. Thus, the code rate was 8/2048 = 1/256. Among the 1024 complex samples, 128 samples were contaminated by jamming; thus, the jamming rate was 128/1024 = 1/8. We compared the results according to three different JNRs: 10, 5, and 0 dB. The results are listed in Table 6.

Table 9. Mean square error (MSE) comparison results between BSJE and GSJE algorithms, according to jamming-to-noise ratio (JNR).

JNR	10 dB		5 dB		0 dB	
Prior Knowledge	**Perfect**	**Unknown**	**Perfect**	**Unknown**	**Perfect**	**Unknown**
MSE (BSJE)	0.0525	0.0523	0.166	0.166	0.523	0.523
MSE (GSJE)	0.0446	0.0880	0.193	0.295	0.754	0.992
Run time (BSJE)	11300	11500	10900	11800	12600	11600
Run time (GSJE)	977	2530	900	2510	896	2440

- MSE performance according to prior knowledge

The results reveal the jamming estimation performance of the GSJE and BSJE algorithms. The MSE of the GSJE algorithm increases as the uncertainty of prior knowledge increases. For example, the MSE of the GSJE with perfect knowledge is 0.0446 at a JNR of 10 dB, which is higher than that of BSJE; however, the MSE decreases to 0.0880 under the same conditions, except for the condition in which prior knowledge is unknown. Its MSE value with unknown prior knowledge is lower than that of BSJE.

- MSE performance according to JNR

The performance of both algorithms degrades as the JNR decreases. However, the degree of performance degradation is greater for the GSJE algorithm. For example, the MSEs of the GSJE with perfect knowledge are 0.193 at a JNR of 5 dB and 0.754 at a JNR of 0 dB, which are lower than those of BSJE. It can be recalled that the MSE of GSJE is higher than that of BSJE at a JNR of 10 dB. Thus, BSJE is more suitable for environments with large JNR ranges than the GSJE. This agrees with the theoretical expectation that the output noise model of GSJE becomes more distinguishable with the true signal model as JNR decreases from 10 dB to 0 dB.

- Running time

The results indicate that the BSJE algorithm is far slower than the GSJE algorithm. However, the running time of GSJE increases when it suffers from a lack of prior knowledge. For example, GSJE is more than ten times faster when it has perfect knowledge and approximately four times faster when it does not have any prior knowledge. In contrast, the running time of BSJE is almost constant regardless of the prior knowledge since BSJE does not explicitly exploit the information.

3.8.2. Bit Error Rate under Jamming

We evaluated the BER at the receiver under jamming attacks in multiple scenarios. The results presented in Figures 20 and 21 illustrate the BER performance of the proposed system in the AWGN channel under a jamming attack with a JNR of 0 dB. The block length of the complex-valued codeword was set to 512 and the number of possible message vectors was four. Thus, the code rate was 4/1024 = 1/256. At the same time, we evaluated the BER of an uncoded DSSS that had same rate in order to use this as a baseline. Among the 512 complex-valued samples, 64 samples were contaminated by jamming; thus, the jamming rate was 64/512 = 1/8. We evaluated the BER under jamming over the E_b / N_0 region from 0 dB to 16 dB, corresponding to SNRs from −24 dB to −8 dB, as SNR is calculated by $(E_b / N_0) \cdot (\text{code rate}) \approx (E_b / N_0) - 24\ \text{dB}$. Note that the SNR is far below the noise floor. As illustrated in Figure 20, the BER of the proposed method using the BSJE and GSJE algorithms with perfect prior knowledge approaches the BER of jamming-free QPSK as E_b / N_0 grows. In contrast, the BER of the GSJE without any prior knowledge cannot achieve anti-jamming performance. Furthermore, if the proposed codeword is drawn using the codebook method, the BER of the proposed method with the BSJE algorithm outperforms that of the jamming-free QPSK, as illustrated in Figure 21. Thus, the proposed method achieves a coding gain while simultaneously having an anti-jamming property.

Figure 22 and 23 illustrate the BER performance of the proposed system over a frequency-selective fading channel. The simulation was performed over a frequency-selective fading channel with frequency-domain channel coefficients following an i.i.d. CSCG random distribution $\mathcal{CN}(\mathbf{0}, \mathbf{I})$, with the other parameters being identical to those in the AWGN channel simulation. Because several subbands have significantly low gain, the BER of the uncoded QPSK signal is significantly more degraded than that in the AWGN channel simulation. In contrast, the BER degradation of the proposed method is tolerable when compared to that of the uncoded QPSK. The method using BSJE demonstrates the highest performance, and the method using GSJE shows the lowest

performance among the proposed methods. Consistent with the AWGN channel simulation, the performance of the codebook method illustrated in Figure 22 is superior to that of the linear Gaussian method illustrated in Figure 23.

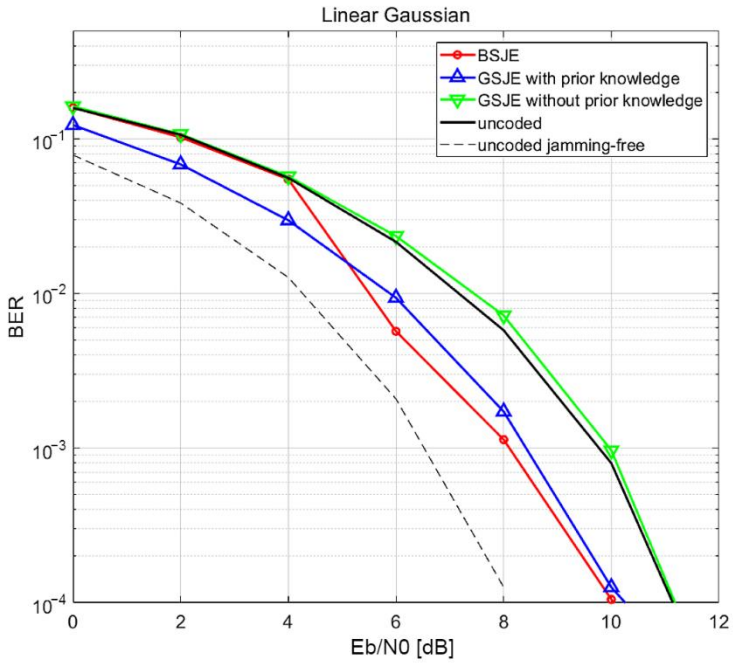


Figure 20. Bit error rate (BER) performance of the proposed system in an AWGN channel at JNR = 0 dB for the linear Gaussian method

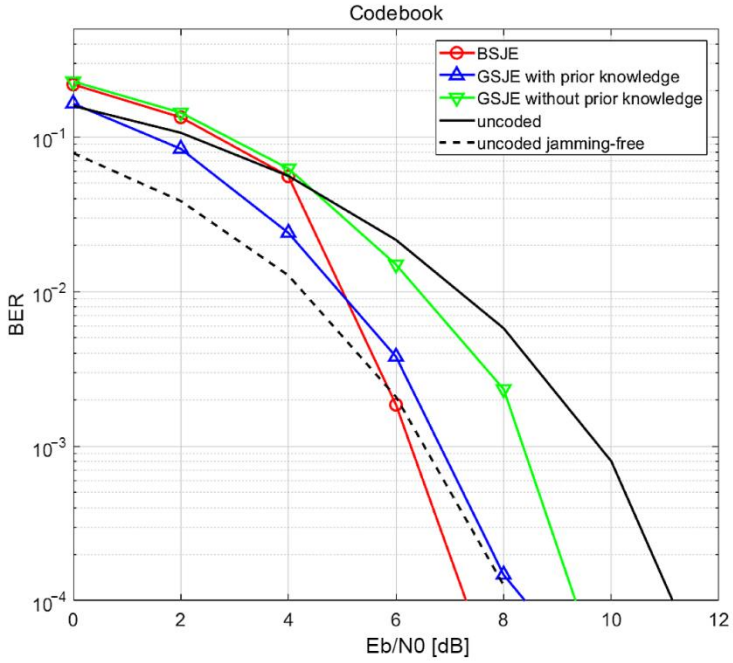


Figure 21. BER performance of the proposed system in an AWGN channel at JNR = 0 dB for the codebook method

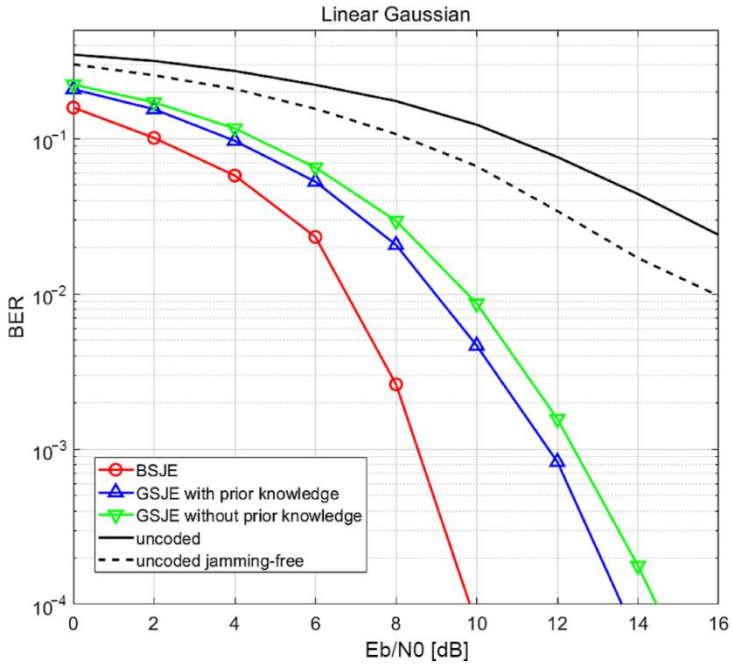


Figure 22. BER performance of the proposed system over a frequency-selective channel at JNR = 0 dB for the linear Gaussian method

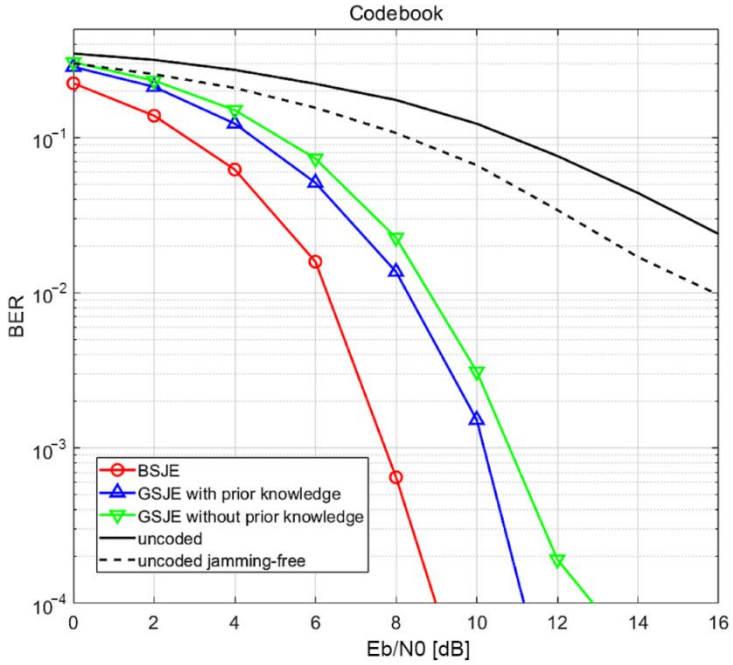


Figure 23. BER performance of the proposed system over a frequency-selective channel at JNR = 0 dB for the

codebook method

3.9. Conclusions

In this study, a novel covert anti-jamming communication system was proposed. The proposed system exploited the Gaussian-coded time-frequency modulation method to obtain covertness. As the signal samples of the proposed system followed Gaussian distributions, the signal was less detectable than the traditional binary modulated signals. We proposed the SSR-algorithm-based SJE algorithms to estimate and remove the effects of hostile jamming. In contrast to previous studies, the proposed system used the Gaussian codebook method to achieve a coding gain when the finite-field messages were encoded. We analyzed the undetectability of the proposed system as a measurement of the covertness of communications, and compared the undetectability with that of the conventional binary modulated system. We also demonstrated its BER performance through numerical simulations. The proposed system can be applied in eavesdropping- and jamming-critical applications, such as military communications in electronic warfare scenarios.

Chapter IV. Summary and Future Research Direction

4.1. Summary

In this dissertation, it is shown that ECC can be a powerful security primitive beyond its original purpose. Computational hardness of decoding, one-wayness, design flexibility, and extremely well-studied theoretical bounds make ECC as an attractive toolkit for various security criterions. This dissertation focus on addressing two challenges, one is securing trust of decentralization in PoW blockchains whose mining is dominated by specialized hardware such as ASICs. The other challenge is securing stealth wireless communication under active adversaries who can both eavesdrop and jam the target communication. This dissertation shows that suitably chosen ECC problems can contribute to (i) an ASIC-resistant consensus protocol that maintains blockchain decentralization and (ii) a noise-like modulation that remains undetectable while withstanding smart jammer.

In the first research described in Chapter 2, Error-correction code verifiable computation consensus (ECCVCC) algorithm is proposed to secure decentralization of a blockchain. This research generalized traditional hash-based PoW consensus to a verifiable computation consensus, where its core part called verifiable computation puzzle can be designed based on decoding problem of an ECC. The representative variant, ECCVCC-SD, embeds syndrome-decoding of an LDPC code problem in every block publishing puzzle. The parity-check matrix required for the puzzle is generated each block, whereas the puzzle generation maintaining unpredictable but verifiable. This flexibility makes designing an ASIC inefficient, therefore sustaining mining process fair and decentralized.

In the research, solving-verifying gap of ECCVCC is analyzed, which is a security metric against DoS attack. The result shows that ECCVCC-SD is more secure than prior arts such as ECCPoW. A difficulty adjusting algorithm for ECCVCC is studied to more precisely retarget the puzzle difficulty and sustain block generation time. Relationship between ASIC-resistance and decentralization is numerical evaluated. The evaluation shows that ASIC-friendly consensus algorithm accelerates re-centralization, while ASIC-resistant algorithm such as ECCVCC can delay it. In this model, PoS consensus is equivalent to an extremely ASIC-friendly consensus, which significantly threat the decentralization.

In the second research described in Chapter 3, covert anti-jamming wireless communication scheme based on Gaussian coded time-frequency modulation is proposed. The proposed scheme maps finite-field payload bits to Gaussian codeword symbols spread across wide time-frequency grid. Ensemble of the symbols is

statistically hard to distinguish from channel noise under AWGN channel. Because of spread nature of symbols, blind jamming with limited time or band is naturally modeled by sparse error which can be cancelled out using the proposed greedy and Bayesian sparse jamming estimation (GSJE and BSJE) algorithms.

Covertness of proposed scheme is analyzed by symbol-level detection probability against passive eavesdroppers with energy-detection strategy. The analysis shows proposed scheme is more covert than general binary signaling scheme, especially under the threating scenario that eavesdropper is powerful and aggressive. Anti-jamming performance is evaluated by two metrics, MSE of jamming estimation algorithm and BER of decoded information bits. The results show that BER approached jamming-free scheme, which implies sparse jamming can be successfully suppressed. GSJE algorithm shows faster running time than BSJE but requires prior knowledge. Possible implementation issues are discussed. The polynomial complexity of SJE algorithm is feasible and can be even more reduced by specialized hardware. Practice with commercial 12-bits analog-to-digital converter (ADC) for symbol quantization does not degrade the undetectable performance.

4.2. Future Research Direction

4.2.1. Open Problems in ECCVCC

Our research in Chapter 2 proposed ECCVCC to achieve trust of decentralization of PoW-based blockchain. ECCVCC inherits the reliability and security of PoW while achieving ASIC-resistance. However, ECCVCC still faces two open challenges.

A. Energy Efficiency

Since ECCVCC secures the integrity of blockchains by embedding verifiable computational effort into each block, its energy consumption remains high compared with other consensus mechanism such as PoS. Existing PoW variants are proposed as countermeasures for the energy consumption problem but they have their own risks. GreenPoW [85] grants the producer of the latest uncle block an exclusive privilege to mine the next block. This method reduces the computing power wasted for mining the uncle block. This approach, however, introduce liveness risks. Sprint [86] inserts mandatory pauses enforced by a verifiable delay function (VDF). This approach limits the working time of mining device, resulting reduction of energy consumption. However, the energy cost of evaluating the VDF is excluded from the authors' analysis, so its net benefit is unclear. EpoW [87] shares the reward between the first and second block proposers, discouraging hash-rate growth. This design, however, demands finely tuned reward sharing and difficulty adjustment rules to avoid new security vulnerabilities.

We will study the combination of ECCVCC and a verifiable random function (VRF). The VRF randomly, and verifiably selects only a small subset of nodes to execute the ECCVCC puzzle in each block-publishing round. By controlling the size of the subset, total network energy consumption should fall. A critical research question is whether the resulting protocol can still secure for double-spending and Sybil attacks under realistic adversarial models.

B. Improving ASIC-Resistance

The second challenge is additional improvement of ASIC-resistance of ECCVCC. The current design of ECCVCC derives ASIC-resistance solely from optimization-hardness: the algorithm is difficult to be efficiently optimized with ASIC. However, hybrid schemes such as ProgPoW [55] and KAWPoW [58] go further by combining optimization-hardness with bandwidth-hardness. They randomize the order of diverse arithmetic operations and require frequent, unpredictable memory access to a large directed acyclic graph (DAG), thereby introducing optimization- and bandwidth-hardness.

We aim to incorporate memory-intensive, randomly indexed data structures into ECCVCC so that a solver must satisfy both compute and bandwidth constraints. Designing such structures without undermining the verifiability, latency, and security constitutes our second major future work.

4.2.2. Steganographic Code for Large Language Model

Steganography is another promising research direction. Our covert wireless communication research can be generalized to steganography, especially language steganography which attracts many interests because of recent advance in large language model (LLM). A variety of steganography system based on LLM [130], [131] has been proposed to achieve covert communication through LLM. One can design a steganographic ECC aims embed as many secret bit with minimal distortion in LLM-generated stegotext.

References

[1] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948, doi: 10.1002/j.1538-7305.1948.tb01338.x.

[2] R. G. Gallager, *Low-Density Parity-Check Codes*. MIT Press, 1963. doi: 10.7551/mitpress/4347.001.0001.

[3] R. J. McEliece, “A Public-Key Cryptosystem Based On Algebraic Coding Theory,” *Deep Space Network Progress Report*, vol. 44, pp. 114–116, Jan. 1978.

[4] “Classic McEliece: Intro.” Accessed: May 20, 2025. [Online]. Available: https://classic.mceliece.org/

[5] “HQC.” Accessed: May 20, 2025. [Online]. Available: https://pqc-hqc.org/index.html

[6] Y. X. Li, R. H. Deng, and X. M. Wang, “On the equivalence of McEliece’s and Niederreiter’s public-key cryptosystems,” *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 271–273, Jan. 1994, doi: 10.1109/18.272496.

[7] E. Prange, “The use of information sets in decoding cyclic codes,” *IRE Transactions on Information Theory*, vol. 8, no. 5, pp. 5–9, Sep. 1962, doi: 10.1109/TIT.1962.1057777.

[8] Q. Guo, T. Johansson, and V. Nguyen, “A New Sieving-Style Information-Set Decoding Algorithm,” *IEEE Transactions on Information Theory*, vol. 70, no. 11, pp. 8303–8319, Jan. 2024, doi: 10.1109/TIT.2024.3457150.

[9] G. Alagic *et al.*, “Status Report on the Fourth Round of the NIST Post-Quantum Cryptography Standardization Process,” National Institute of Standards and Technology, NIST Internal or Interagency Report (NISTIR) 8545, Mar. 2025. doi: 10.6028/NIST.IR.8545.

[10] C. Aguilar-Melchor, O. Blazy, J.-C. Deneuville, P. Gaborit, and G. Zémor, “Efficient Encryption From Random Quasi-Cyclic Codes,” *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3927–3943, May 2018, doi: 10.1109/TIT.2018.2804444.

[11] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Information hiding-a survey,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999, doi: 10.1109/5.771065.

[12] N. Provos and P. Honeyman, “Hide and seek: an introduction to steganography,” *IEEE Security & Privacy*, vol. 1, no. 3, pp. 32–44, May 2003, doi: 10.1109/MSECP.2003.1203220.

[13] T. Filler, J. Judas, and J. Fridrich, “Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, Sep. 2011, doi: 10.1109/TIFS.2011.2134094.

[14] W. Li, W. Zhang, L. Li, H. Zhou, and N. Yu, “Designing Near-Optimal Steganographic Codes in Practice Based on Polar Codes,” *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 3948–3962, Jul. 2020, doi: 10.1109/TCOMM.2020.2982624.

[15] Q. Yao, W. Zhang, K. Chen, and N. Yu, “LDGM Codes-Based Near-Optimal Coding for Adaptive Steganography,” *IEEE Transactions on Communications*, vol. 72, no. 4, pp. 2138–2151, Apr. 2024, doi: 10.1109/TCOMM.2023.3342247.

[16] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, [Online]. Available: https://bitcoin.org/bitcoin.pdf

[17] U. Chohan, “The Double Spending Problem and Cryptocurrencies,” *SSRN Electronic Journal*, Jan. 2017, doi: 10.2139/ssrn.3090174.

[18] B. A. Bash, D. Goeckel, D. Towsley, and S. Guha, “Hiding information in noise: fundamental limits of covert wireless communication,” *IEEE Communications Magazine*, vol. 53, no. 12, pp. 26–31, Feb. 2015, doi: 10.1109/MCOM.2015.7355562.

[19] B. He, S. Yan, X. Zhou, and V. K. N. Lau, “On Covert Communication With Noise Uncertainty,” *IEEE Communications Letters*, vol. 21, no. 4, pp. 941–944, Apr. 2017, doi: 10.1109/LCOMM.2016.2647716.

[20] A. D. Wyner, “The wire-tap channel,” *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, Oct. 1975, doi: 10.1002/j.1538-7305.1975.tb02040.x.

[21] Y. M. Eldokmak, M. H. Ismail, and M. S. Hassan, “Secrecy Analysis Over Correlated Generalized Gamma Fading Channels,” in *2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Feb. 2022, pp. 1–6. doi: 10.1109/ICCSPA55860.2022.10019241.

[22] A.-A. A. Boulogeorgos and A. Alexiou, “On the Ergodic Secrecy Capacity of Reconfigurable Intelligent Surface Aided Wireless Systems Under Mixture Gamma Fading,” in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2023, pp. 1–6. doi: 10.1109/WCNC55385.2023.10118925.

[23] A. Mukherjee, B. Ottersten, and L.-N. Tran, “On the Secrecy Capacity of MIMO Wiretap Channels: Convex Reformulation and Efficient Numerical Methods,” *IEEE Transactions on Communications*, vol. 69, no. 10, pp. 6865–6878, Oct. 2021, doi: 10.1109/TCOMM.2021.3098700.

[24] C. Song, W. Guo, H. Zhao, and S. Shao, “Artificial Noise Sheltered FH Broadcasting With Frequency Mismatch: Secrecy Analysis and Power Allocation,” *IEEE Transactions on Broadcasting*, vol. 68, no. 1, pp. 279–285, Mar. 2022, doi: 10.1109/TBC.2021.3110410.

[25] P. Xu, W. Tan, K. Niu, Z. Li, and C. Peng, “Secrecy Performance Analysis of Artificial Noise Assisted Secure

Transmission for MISO systems with Multiple Eavesdroppers," in *2023 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Jan. 2023, pp. 1–6. doi: 10.1109/ICSPCC59353.2023.10400224.
[26] S. Liu, X. Zhu, H. Chen, and Z. Han, "Secure Communication for Integrated Satellite–Terrestrial Backhaul Networks: Focus on Up-Link Secrecy Capacity Based on Artificial Noise," *IEEE Wireless Communications Letters*, vol. 12, no. 8, pp. 1369–1373, Aug. 2023, doi: 10.1109/LWC.2023.3274761.
[27] N. Smart, *Cryptography: An Introduction*. London: McGraw-Hill College, 2003.
[28] J. Daemen and V. Rijmen, "AES proposal: rijndael," Sep. 1999. Accessed: May 22, 2025. [Online]. Available: https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf
[29] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976, doi: 10.1109/TIT.1976.1055638.
[30] N. I. of S. and Technology, "Module-Lattice-Based Key-Encapsulation Mechanism Standard," U.S. Department of Commerce, Federal Information Processing Standard (FIPS) 203, Aug. 2024. doi: 10.6028/NIST.FIPS.203.
[31] L. Wang, G. W. Wornell, and L. Zheng, "Limits of low-probability-of-detection communication over a discrete memoryless channel," in *2015 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2015, pp. 2525–2529. doi: 10.1109/ISIT.2015.7282911.
[32] B. A. Bash, D. Goeckel, and D. Towsley, "Limits of Reliable Communication with Low Probability of Detection on AWGN Channels," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 1921–1930, Sep. 2013, doi: 10.1109/JSAC.2013.130923.
[33] C. Cachin, "An Information-Theoretic Model for Steganography," 2000, 2000/028. Accessed: May 22, 2025. [Online]. Available: https://eprint.iacr.org/2000/028
[34]N. Hopper, L. von Ahn, and J. Langford, "Provably Secure Steganography," *IEEE Transactions on Computers*, vol. 58, no. 5, pp. 662–676, May 2009, doi: 10.1109/TC.2008.199.
[35] J. Yu and Y.-D. Yao, "Detection performance of chaotic spreading LPI waveforms," *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, pp. 390–396, Mar. 2005, doi: 10.1109/TWC.2004.842948.
[36] S. Sedaghatnejad and M. Farhang, "Detectability of Chaotic Direct-Sequence Spread-Spectrum Signals," *IEEE Wireless Communications Letters*, vol. 4, no. 6, pp. 589–592, Feb. 2015, doi: 10.1109/LWC.2015.2469776.
[37] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, and D. Schipani, "Using LDGM Codes and Sparse Syndromes to Achieve Digital Signatures," in *Post-Quantum Cryptography*, P. Gaborit, Ed., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, pp. 1–15. doi: 10.1007/978-3-642-38616-9_1.
[38] V. Guruswami and A. Vardy, "Maximum-likelihood decoding of Reed-Solomon codes is NP-hard," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2249–2256, Jul. 2005, doi: 10.1109/TIT.2005.850102.
[39] I. M. Ali, M. Caprolu, and R. D. Pietro, "Foundations, Properties, and Security Applications of Puzzles: A Survey," *ACM Comput. Surv.*, vol. 53, no. 4, p. 72:1-72:38, Aug. 2020, doi: 10.1145/3396374.
[40] A. Back, "Hashcash - A denial of service counter-measure." [Online]. Available: https://nakamotoinstitute.org/library/hashcash/
[41] A. Sajjad, Q. Zhang, F. Ciampi, F. Pascucci, and B. Gupta, "What Motivates Bitcoin Miners to Practice Bitcoin Mining: An Assessment Based on Behavioral Reasoning Theory," *IEEE Transactions on Engineering Management*, vol. 71, pp. 8209–8222, Nov. 2023, doi: 10.1109/TEM.2023.3335662.
[42] L. Knudsen and B. Preneel, "Construction of secure and fast hash functions using nonbinary error-correcting codes," *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2524–2539, Sep. 2002, doi: 10.1109/TIT.2002.801402.
[43] S. Ermon, C. Gomes, A. Sabharwal, and B. Selman, "Low-density Parity Constraints for Hashing-Based Discrete Integration," in *Proceedings of the 31st International Conference on Machine Learning*, PMLR, Jan. 2014, pp. 271–279. Accessed: Jan. 30, 2023. [Online]. Available: https://proceedings.mlr.press/v32/ermon14.html
[44] S. P. Vadhan, "Pseudorandomness," *FNT in Theoretical Computer Science*, vol. 7, no. 1–3, pp. 1–336, Dec. 2012, doi: 10.1561/0400000010.
[45] S. Park, H. Choi, and H.-N. Lee, "Time-Variant Proof-of-Work Using Error-Correction Codes," Jun. 22, 2020, *arXiv*: arXiv:2006.12306. doi: 10.48550/arXiv.2006.12306.
[46] H. Jung and H.-N. Lee, "ECCPoW: Error-Correction Code based Proof-of-Work for ASIC Resistance," *Symmetry*, vol. 12, no. 6, Art. no. 6, Jun. 2020, doi: 10.3390/sym12060988.
[47] H. Kim, J. Jang, S. Park, and H.-N. Lee, "Error-Correction Code Proof-of-Work on Ethereum," *IEEE Access*, vol. 9, pp. 135942–135952, 2021, doi: 10.1109/ACCESS.2021.3113522.
[48] A. Koutra and V. Tenentes, "Multi-Vt-Based Energy Efficiency Optimization for ASIC Designs of the Double Secure Hash Algorithm Toward a Sustainable Bitcoin Network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 71, no. 3, pp. 1596–1600, Mar. 2024, doi: 10.1109/TCSII.2024.3350035.

[49] Q. H. Dang, “Secure Hash Standard,” National Institute of Standards and Technology, NIST FIPS 180-4, Jul. 2015. doi: 10.6028/NIST.FIPS.180-4.
[50] H. Cho, “ASIC-Resistance of Multi-Hash Proof-of-Work Mechanisms for Blockchain Consensus Protocols,” *IEEE Access*, vol. 6, pp. 66210–66222, Oct. 2018, doi: 10.1109/ACCESS.2018.2878895.
[51] L. Ren and S. Devadas, “Bandwidth Hard Functions for ASIC Resistance,” in *Theory of Cryptography*, Y. Kalai and L. Reyzin, Eds., Cham: Springer International Publishing, Nov. 2017, pp. 466–492. doi: 10.1007/978-3-319-70500-2_16.
[52] V. Buterin, “Dagger: A Memory-Hard to Compute, Memory-Easy to Verify Scrypt Alternative.” Accessed: Jan. 30, 2023. [Online]. Available: http://www.hashcash.org/papers/dagger.html
[53] T. Black and J. Weight, “X16R ASIC Resistant by Design.” Accessed: Jan. 30, 2023. [Online]. Available: https://ravencoin.org/assets/documents/X16R-Whitepaper.pdf
[54] E. Duffield and D. Diaz, “Dash: A Payments-Focused Cryptocurrency,” GitHub. Accessed: Jan. 30, 2023. [Online]. Available: https://github.com/dashpay/dash/wiki/Whitepaper
[55] G. Colvin, A. Lanfranchi, M. Carter, and IfDefElse, “EIP-1057: ProgPoW, a Programmatic Proof-of-Work,” Ethereum Improvement Proposals. Accessed: Jan. 30, 2023. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1057
[56] J. Orender, R. Mukkamala, and M. Zubair, “Is Ethereum’s ProgPoW ASIC Resistant?:,” in *Proceedings of the 6th International Conference on Information Systems Security and Privacy*, Valletta, Malta: SCITEPRESS - Science and Technology Publications, Feb. 2020, pp. 310–316. doi: 10.5220/0008909203100316.
[57] “SERO_ENG_V2.0.pdf,” Super Zero (SERO) Technical White Paper. Accessed: Jan. 30, 2023. [Online]. Available: https://sero.cash/SERO_ENG_V2.0.pdf
[58] T. Black, “Ravencoin — ASIC Thoughts Round Two,” Medium. Accessed: Jan. 30, 2023. [Online]. Available: https://tronblack.medium.com/ravencoin-asic-thoughts-round-two-f4f743942656
[59] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge University Press, 2009.
[60] Y. Sompolinsky and A. Zohar, “Secure High-Rate Transaction Processing in Bitcoin,” in *Financial Cryptography and Data Security*, R. Böhme and T. Okamoto, Eds., Berlin, Heidelberg: Springer, Jan. 2015, pp. 507–527. doi: 10.1007/978-3-662-47854-7_32.
[61] *cryptoecc/WorldLand*. (May 17, 2024). Go. DeSecure Blockchain. Accessed: Jun. 19, 2024. [Online]. Available: https://github.com/cryptoecc/WorldLand
[62] “ethereum/go-ethereum at release/1.11.” Accessed: Jun. 04, 2025. [Online]. Available: https://github.com/ethereum/go-ethereum/tree/release/1.11
[63] Q. Xia, W. Dou, F. Zhang, and G. Liang, “The Performance of Selfish Mining in GHOST,” in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Oct. 2021, pp. 26–33. doi: 10.1109/TrustCom53373.2021.00022.
[64] D. Micciancio and C. Peikert, “Hardness of SIS and LWE with Small Parameters,” in *Advances in Cryptology – CRYPTO 2013*, R. Canetti and J. A. Garay, Eds., Berlin, Heidelberg: Springer, Jan. 2013, pp. 21–39. doi: 10.1007/978-3-642-40041-4_2.
[65] R. Bowden, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, “Block arrivals in the Bitcoin blockchain,” Jan. 23, 2018, *arXiv*: arXiv:1801.07447. doi: 10.48550/arXiv.1801.07447.
[66] M. Waldvogel, “Bitcoin Block Timing Statistics,” Netfuture: The future is networked. Accessed: Apr. 09, 2024. [Online]. Available: https://netfuture.ch/2022/03/bitcoin-block-timing-statistics/
[67] D. Kraft, “Difficulty control for blockchain-based consensus systems,” *Peer-to-Peer Netw. Appl.*, vol. 9, no. 2, pp. 397–413, Mar. 2016, doi: 10.1007/s12083-015-0347-x.
[68] D. Fullmer and A. S. Morse, “Analysis of Difficulty Control in Bitcoin and Proof-of-Work Blockchains,” in *2018 IEEE Conference on Decision and Control (CDC)*, Feb. 2018, pp. 5988–5992. doi: 10.1109/CDC.2018.8619082.
[69] V. Buterin, “EIP-2: Homestead Hard-fork Changes,” Ethereum Improvement Proposals. Accessed: Dec. 02, 2024. [Online]. Available: https://eips.ethereum.org/EIPS/eip-2
[70] V. Buterin, “EIP-100: Change difficulty adjustment to target mean block time including uncles,” Ethereum Improvement Proposals. Accessed: Dec. 02, 2024. [Online]. Available: https://eips.ethereum.org/EIPS/eip-100
[71] M. Awais and C. Condo, “Flexible LDPC Decoder Architectures,” *VLSI Design*, vol. 2012, p. e730835, Jun. 2012, doi: 10.1155/2012/730835.
[72] C.-H. Liu *et al.*, “An LDPC Decoder Chip Based on Self-Routing Network for IEEE 802.16e Applications,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 684–694, Mar. 2008, doi: 10.1109/JSSC.2007.916610.
[73] S. Shao *et al.*, “Survey of Turbo, LDPC, and Polar Decoder ASIC Implementations,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2309–2333, Jan. 2019, doi: 10.1109/COMST.2019.2893851.
[74] Y.-L. Ueng, B.-J. Yang, C.-J. Yang, H.-C. Lee, and J.-D. Yang, “An Efficient Multi-Standard LDPC Decoder Design Using Hardware-Friendly Shuffled Decoding,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 3, pp. 743–756, Mar. 2013, doi: 10.1109/TCSI.2012.2215746.

[75] C. Beuschel and H.-J. Pfleiderer, “FPGA implementation of a flexible decoder for long LDPC codes,” in *2008 International Conference on Field Programmable Logic and Applications*, Sep. 2008, pp. 185–190. doi: 10.1109/FPL.2008.4629929.
[76] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, “A Survey of FPGA-Based LDPC Decoders,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1098–1122, Dec. 2015, doi: 10.1109/COMST.2015.2510381.
[77] J. M. R. Vaz and J. A. B. Gerald, “A fast LDPC encoder/decoder for small/medium codes,” *Microelectronics Journal*, vol. 44, no. 10, pp. 888–896, Oct. 2013, doi: 10.1016/j.mejo.2013.02.023.
[78] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, “Reduced-complexity decoding of LDPC codes,” *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005, doi: 10.1109/TCOMM.2005.852852.
[79] H.-Y. Kwak, J.-W. Kim, Y. Kim, S.-H. Kim, and J.-S. No, “Neural Min-Sum Decoding for Generalized LDPC Codes,” *IEEE Communications Letters*, vol. 26, no. 12, pp. 2841–2845, Feb. 2022, doi: 10.1109/LCOMM.2022.3208834.
[80] J. Bai, Y. Chi, and C. Yuen, “Efficient MP Decoding via Fast G-BADMM Approach for Binary LDPC Codes,” *IEEE Communications Letters*, vol. 27, no. 3, pp. 782–786, Mar. 2023, doi: 10.1109/LCOMM.2022.3232703.
[81] Y. Chen, R. Wang, J. Zhu, and Z. Wen, “Decoding LDPC Codes by Using Negative Proximal Regularization,” *IEEE Transactions on Communications*, vol. 71, no. 7, pp. 3835–3846, Jul. 2023, doi: 10.1109/TCOMM.2023.3274150.
[82] J. Ling and P. Cautereels, “Fast LDPC GPU Decoder for Cloud RAN,” *IEEE Embedded Systems Letters*, vol. 13, no. 4, pp. 170–173, Feb. 2021, doi: 10.1109/LES.2021.3052714.
[83] C. Tarver, M. Tonnemacher, H. Chen, J. Zhang, and J. R. Cavallaro, “GPU-Based, LDPC Decoding for 5G and Beyond,” *IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 278–290, 2021, doi: 10.1109/OJCAS.2020.3042448.
[84] “The Beacon Chain,” ethereum.org. Accessed: Dec. 02, 2024. [Online]. Available: https://ethereum.org/en/roadmap/beacon-chain/
[85] N. Lasla, L. Al-Sahan, M. Abdallah, and M. Younis, “Green-PoW: An energy-efficient blockchain Proof-of-Work consensus algorithm,” *Computer Networks*, vol. 214, p. 109118, Sep. 2022, doi: 10.1016/j.comnet.2022.109118.
[86] M. Mirkin, L. Zhou, I. Eyal, and F. Zhang, “Sprints: Intermittent Blockchain PoW Mining,” in *33rd USENIX Security Symposium*, Aug. 2024, pp. 6273–6289. Accessed: Dec. 12, 2024. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/mirkin
[87] S. Yu, Y. Qiao, J. Bo, F. Yang, and S. Wang, “EPoW: Energy-Efficient Proof-of-Work,” *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 6, pp. 6285–6297, Jan. 2024, doi: 10.1109/TNSE.2024.3418558.
[88] B. Sklar, *Digital Communications: Fundamentals and Applications*. Upper Saddle River, NJ: Prentice Hall, 2016.
[89] T. Marshall, “Coding of Real-Number Sequences for Error Correction: A Digital Signal Processing Problem,” *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 2, pp. 381–392, Mar. 1984, doi: 10.1109/JSAC.1984.1146063.
[90] Z. Wang and G. B. Giannakis, “Complex-field coding for OFDM over fading wireless channels,” *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 707–720, Mar. 2003, doi: 10.1109/TIT.2002.808101.
[91] E. J. Candes and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Feb. 2005, doi: 10.1109/TIT.2005.858979.
[92] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006, doi: 10.1109/TIT.2006.871582.
[93] C. Chen and Y. Zhuo, “A research on anti-jamming method based on compressive sensing for OFDM analogous system,” in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, Oct. 2017, pp. 655–659. doi: 10.1109/ICCT.2017.8359718.
[94] S. Huan, G. Dai, G. Luo, and S. Ai, “Bayesian Compress Sensing Based Countermeasure Scheme Against the Interrupted Sampling Repeater Jamming,” *Sensors*, vol. 19, no. 15, Art. no. 15, Jan. 2019, doi: 10.3390/s19153279.
[95] B. Liu, G. Gui, S.-Y. Matsushita, and L. Xu, “Compressive Sensing Based Direction-of-Arrival Estimation in MIMO Radars in Presence of Strong Jamming via Blocking Matrix,” in *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Jul. 2018, pp. 292–296. doi: 10.1109/SAM.2018.8448914.
[96] M. Forouzesh, P. Azmi, A. Kuhestani, and P. L. Yeoh, “Covert Communication and Secure Transmission Over Untrusted Relaying Networks in the Presence of Multiple Wardens,” *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3737–3749, Jun. 2020, doi: 10.1109/TCOMM.2020.2978206.
[97] K. Shahzad, X. Zhou, S. Yan, J. Hu, F. Shu, and J. Li, “Achieving Covert Wireless Communications Using a Full-Duplex Receiver,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8517–8530, Feb.

2018, doi: 10.1109/TWC.2018.2878014.
[98] Amir. Beck and Marc. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009, doi: 10.1137/080716542.
[99] Junfeng. Yang and Yin. Zhang, "Alternating Direction Algorithms for ℓ_1-Problems in Compressive Sensing," *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 250–278, Jan. 2011, doi: 10.1137/090777761.
[100] J. A. Tropp and A. C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007, doi: 10.1109/TIT.2007.909108.
[101] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004, doi: 10.1109/TIT.2004.834793.
[102] A. Scott, "Effects of Cyclic Prefix Jamming Versus Noise Jamming in OFDM Signals," 2011. [Online]. Available: https://scholar.afit.edu/etd/1426
[103] C. Shahriar *et al.*, "PHY-Layer Resiliency in OFDM Communications: A Tutorial," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 292–314, 2015, doi: 10.1109/COMST.2014.2349883.
[104] L. Xiao, T. Chen, J. Liu, and H. Dai, "Anti-Jamming Transmission Stackelberg Game With Observation Errors," *IEEE Communications Letters*, vol. 19, no. 6, pp. 949–952, Jun. 2015, doi: 10.1109/LCOMM.2015.2418776.
[105] D. Yang, G. Xue, J. Zhang, A. Richa, and X. Fang, "Coping with a Smart Jammer in Wireless Networks: A Stackelberg Game Approach," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 4038–4047, Aug. 2013, doi: 10.1109/TWC.2013.071913121570.
[106] M. Stojnic, W. Xu, and B. Hassibi, "Compressed sensing - probabilistic analysis of a null-space characterization," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2008, pp. 3377–3380. doi: 10.1109/ICASSP.2008.4518375.
[107] W. Xu, E.-W. Bai, and M. Cho, "Toeplitz matrix based sparse error correction in system identification: Outliers and random noises," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6640–6644. doi: 10.1109/ICASSP.2013.6638946.
[108] G. Raskutti, M. J. Wainwright, and B. Yu, "Minimax Rates of Estimation for High-Dimensional Linear Regression Over \ell_q -Balls," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6976–6994, Oct. 2011, doi: 10.1109/TIT.2011.2165799.
[109] E. Arias-Castro and Y. C. Eldar, "Noise Folding in Compressed Sensing," *IEEE Signal Processing Letters*, vol. 18, no. 8, pp. 478–481, Aug. 2011, doi: 10.1109/LSP.2011.2159837.
[110] S. Chrétien and A. O. Hero, "On EM algorithms and their proximal generalizations," *ESAIM: PS*, vol. 12, pp. 308–326, Apr. 2008, doi: 10.1051/ps:2007041.
[111] R. Giri and B. Rao, "Type I and Type II Bayesian Methods for Sparse Signal Recovery Using Scale Mixtures," *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3418–3428, Jul. 2016, doi: 10.1109/TSP.2016.2546231.
[112] S. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Englewood Cliffs, N.J: Pearson, 1993.
[113] T. S. Han and S. Verdu, "Approximation theory of output statistics," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 752–772, May 1993, doi: 10.1109/18.256486.
[114] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982, doi: 10.1109/TIT.1982.1056489.
[115] U. Moller, M. Galicki, E. Baresova, and H. Witte, "An efficient vector quantizer providing globally optimal solutions," *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2515–2529, Sep. 1998, doi: 10.1109/78.709539.
[116] G. Pages and J. Printems, "Optimal quadratic quantization for numerics: the Gaussian case," *Monte Carlo Methods and Applications*, vol. 9, no. 2, pp. 135–165, Apr. 2003, doi: https://doi.org/10.1515/156939603322663321.
[117] "Snapdragon 855 Mobile Platform," Qualcomm. Accessed: Feb. 25, 2020. [Online]. Available: https://www.qualcomm.com/products/snapdragon-855-mobile-platform
[118] "USRP-2900 Specifications - National Instruments," p. 6.
[119] M. A. Aref, S. K. Jayaweera, and S. Machuzak, "Multi-Agent Reinforcement Learning Based Cognitive Anti-Jamming," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2017, pp. 1–6. doi: 10.1109/WCNC.2017.7925694.
[120] Y. Wu, B. Wang, K. J. R. Liu, and T. C. Clancy, "Anti-Jamming Games in Multi-Channel Cognitive Radio Networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 4–15, Jan. 2012, doi: 10.1109/JSAC.2012.120102.
[121] "C6000 Power-Optimized DSP |Floating point DSP Processor | Overview | DSP | TI.com." Accessed: Mar. 16, 2020. [Online]. Available: http://www.ti.com/processors/digital-signal-processors/c6000-floating-point-dsp/overview.html

[122] G. Sharma, A. Agarwala, and B. Bhattacharya, "A fast parallel Gauss Jordan algorithm for matrix inversion using CUDA," *Computers & Structures*, vol. 128, pp. 31–37, Nov. 2013, doi: 10.1016/j.compstruc.2013.06.015.

[123] D. Yu, S. He, Y. Huang, G. Yu, and L. Yang, "A fast parallel matrix inversion algorithm based on heterogeneous multicore architectures," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2015, pp. 903–907. doi: 10.1109/GlobalSIP.2015.7418328.

[124] Y. Quan, Y. Li, X. Gao, and M. Xing, "FPGA Implementation of Real-Time Compressive Sensing with Partial Fourier Dictionary," *International Journal of Antennas and Propagation*, vol. 2016, no. 1671687, p. e1671687, Jan. 2016, doi: https://doi.org/10.1155/2016/1671687.

[125] S. Kim *et al.*, "Reduced Computational Complexity Orthogonal Matching Pursuit Using a Novel Partitioned Inversion Technique for Compressive Sensing," *Electronics*, vol. 7, no. 9, Art. no. 9, Sep. 2018, doi: 10.3390/electronics7090206.

[126] V. Q. Nguyen and S. Y. Park, "High-performance ASIC realization of orthogonal matching pursuit algorithm," *IEICE Electron. Express*, vol. 15, no. 7, pp. 20180075–20180075, 2018, doi: 10.1587/elex.15.20180075.

[127] S. H. Cha, M. Shin, J.-H. Ham, and M. Y. Chung, "Robust Mobility Management Scheme in Tactical Communication Networks," *IEEE Access*, vol. 6, pp. 15468–15479, 2018, doi: 10.1109/ACCESS.2018.2812178.

[128] G. Chen, P. Sun, and J. Zhang, "Repair Strategy of Military Communication Network Based on Discrete Artificial Bee Colony Algorithm," *IEEE Access*, vol. 8, pp. 73051–73060, 2020, doi: 10.1109/ACCESS.2020.2987860.

[129] M. A. Davenport and M. B. Wakin, "Analysis of Orthogonal Matching Pursuit Using the Restricted Isometry Property," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4395–4401, Sep. 2010, doi: 10.1109/TIT.2010.2054653.

[130] J. Wu, Z. Wu, Y. Xue, J. Wen, and W. Peng, "Generative Text Steganography with Large Language Model," in *Proceedings of the 32nd ACM International Conference on Multimedia*, Oct. 2024, pp. 10345–10353. doi: 10.1145/3664647.3680562.

[131] N. Perry, S. Gupte, N. Pitta, and L. Rotem, "Robust Steganography from Large Language Models," Apr. 11, 2025, *arXiv*: arXiv:2504.08977. doi: 10.48550/arXiv.2504.08977.

Curriculum Vitae

Name	:	Haeung Choi
Birth Date	:	Dec. 30. 1990.
Birth Place	:	Republic of Korea
Permanent Address	:	123 Cheomdangwagi-ro (Oryong-dong), Buk-gu, Gwangju
E-mail	:	haeung@gist.ac.kr , choihaeung@gmail.com

Research Interests

- Blockchain
- User Authentication System
- Covert Communication, Steganography
- Large Language Model

Education

2015 – 2025 School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (Ph.D.)

2013 – 2015 School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (M.S.)

2009 – 2013 School of Electronic Engineering, Kyungpook National University (B.S.)

Work Experience

2023 – 2025 리버밴스 주식회사, 책임 연구원

Professional Activities

2024. 11. 01. 목포대학교 초청 강연, “블록체인과 스마트 컨트랙트 기초 실습”

2023. 11. 24. 전남대학교 초청 강연, “블록체인과 스마트 컨트랙트 기초 실습”

Patent Publications

1. 이흥노, 박상준, **최해웅**, 이웅비, “부호-암호 화폐 시스템”, 국내특허등록, 출원번호: 2018-0097677, 출원일: 2018-08-21, 등록번호: 10-2231257, 등록일: 2021-03-17.
2. **Haeung Choi**, Heung-No Lee, Jusung Kang, and Sumin Lee, “Transaction apparatus of cryptocurrency and management method for the transaction apparatus of cryptocurrency”, US patent applied, application number: 19/080664, application date: 2025-03-14.
3. Rahman, S M Wahidur, 이흥노, 김수민, **최해웅**, “인공지능서비스장치, 및 인공지능서비스방법”, 국내특허출원, 출원번호: 2025-0046007, 출원일: 2025-04-09.
4. **최해웅**, 이흥노, 강주성, 김수민, “암호화폐 거래장치 및 암호화폐 거래장치의 운용방법”, 국내특허출원, 출원번호: 2024-0045254, 출원일: 2024-04-03

Journal Publications

1. **H. Choi**, S. Kim, and H.-N. Lee, “Error-Correction Code Verifiable Computation Consensus,” accepted for publication in *IEEE Transactions on Information Forensics and Security*, 2025.
2. D. S. Bhatti, M. A. Haroon, **H. Choi**, S. Kim, and H.-N. Lee, “Outage reduction in 5G uplink cooperative NOMA using PSO-optimized relay selection and successive interference cancellation,” accepted for publication in *EURASIP J. Wirel. Commun. Netw*., 2025.
3. R. S. M. Wahidur, S. Kim, **H. Choi**, D. S. Bhatti, and H.-N. Lee, “Legal Query RAG,” *IEEE Access*, vol. 13, pp. 36978–36994, 2025.
4. **H. Choi**, S. Park, and H.-N. Lee, “Covert Anti-Jamming Communication Based on Gaussian Coded Modulation,” *Applied Sciences*, vol. 11, no. 9, Art. no. 9, Jan. 2021.
5. S. Park, **H. Choi**, and H.-N. Lee, “Time-Variant Proof-of-Work Using Error-Correction Codes,” Jun. 22, 2020, *arXiv*: arXiv:2006.12306.

Conference Publications

1. Seungmin Kim, **Haeung Choi** and Heung-No Lee, “검증 가능한 무작위 함수를 사용한 에너지 효율적인 VRF-작업증명 합의 알고리즘”, *2024 년도 한국통신학회 동계종합학술발표회*, Feb. 1st, 2024.
2. Sumin Kim, **Haeung Choi**, Jusung Kang, Minho Yoon and Heung-No Lee, 탈중앙거래소의 비영구적 손실 제거를 위한 딥러닝 기반 Dynamic AMM. *대한전자공학회 학술대회*, 2023, 2411-2414.
3. Ha-young Park, **Haeung Choi** and Heung-No Lee, “플래시론”, *2021 년도 한국정보통신설비학회*, Nov 25~26th, 2021.
4. Ha-young Park, **Haeung Choi** and Heung-No Lee, “아토믹스왑 기술 및 동향 연구”, 2021 년도 대한전자공학회, Nov 26~27th, 2021.

5. **Haeung Choi** and Heung-No Lee, “공격 신호 구분을 위한 무선 신호 패턴 학습 및 분류 방법”, 2020 년도 한국전자파학회 하계종합학술대회, Aug. 19~22nd, 2020.
6. Sangjun Park, **Haeung Choi**, Woong-Bi Lee, Cheolsun Kim, Jusung Kang and Heung-No Lee, “다중 측정 벡터 모델에 관한 최신 분석 결과 소개”, 2017 년도 하계종합학술대회, June 29 -July 1, 2017.
7. Jaewon Bang, **Haeung Choi** and Heung-No Lee, “희소한 부분 대역 잡음 재밍 환경에서 OMP 를 이용한 항재밍 통신시스템과 대역확산 방식을 이용한 BCH 코드의 성능 비교”, 2017 년도 한국통신학회 하계종합학술발표회, pp.1074-1075, Jun. 2017.
8. Jaewon Bang, **Haeung Choi**, Cheolsun Kim and Heung-No Lee, “압축센싱 신호처리 기법을 적용한 항재밍/저피탐 통신 시스템”, 2016 년도 한국통신학회 하계종합학술발표회, pp.1289-1290, Jun. 2016.
9. **Haeung Choi** and Heung-No Lee, “희소한 가중 그래프 상에서의 압축센싱 복구방법”, 제 26 회 통신정보 합동학술대회 (JCCI2016), April 27-29th, 2016.
10. **Haeung Choi**, Jaegun Choi, Jongmok Shin, Seungyun Choe, and Heung-No Lee, “압축 센싱 기술과 신호간의 상관관계를 이용하여 전력 소모를 줄인 무선 센서 네트워크 시스템의 설계”, 대한전자공학회 2014 년도 하계종합학술발표회, pp.483-485, Jun 25-27, 2014
11. Muhammad A. Raza, Zafar Iqbal, Seungchan Lee, **Haeung Choi**, and Heung-No Lee, “Issues and Resolution Efforts Pertaining to TV Whitespace Usage”, 한국통신학회 2014 년도 동계종합학술발표회, pp. 77-79, Jan 22-24

Project Experience

1. LiberVance Layer 2 네트워크 기술 개발 — Layer 2 플랫폼
   - TIPS 프로그램, 중소벤처기업진흥공단, 5 억원, 2022. 07 ~ 2024. 06
2. AI-DEX 서비스 사업화
   - AI(시)제품·서비스 상용화 지원사업, 인공지능산업융합사업단 (AICA), 1.2 억원, 2023.06. ~ 2023. 11.
3. AI 기반 AMM 알고리즘 및 IL 제어기술을 통한 차세대 DEX 서비스 개발
   - AI (시)제품·서비스 제작 지원사업, 인공지능산업융합사업단 (AICA), 0.9 억원, 2022. 06. ~ 2022. 11.
4. 영지식 센싱, 암호인증, 블록체인 기반 클라우드 서비스 융합 기술 개발
   - 대학 ICT 연구센터육성지원사업(ITRC), 정보통신기획평가원 (IITP), 65 억원, 2021. 07. ~ 2028. 12.
5. 대규모 노드를 위한 탈중앙화 합의체 구성 기술 개발

- 정보통신기획평가원 (IITP), 1 억원, 2021.04 ~ 2025.12.

6. EW33: 잡음통신기술을 적용한 적응형 다중모드 항재밍/저피탐 통신 기법 연구
   - 국방특화연구센터사업, 국방과학연구소, 4.1 억, 2019. 07. ~ 2021. 12.
7. 확장가능한 탈중앙화 보안성 ECCPoW 블록체인
   - 정보통신기획평가원, 3 억원, 2019.04. ~ 2019.12.
8. 레이다, 분광기, 뇌-컴퓨터 인터페이스 시스템에 특화된 지능형 신호 복원 및 분류 시스템 연구
   - 한국연구재단, 8.4 억원, 2018.03 ~ 2021.02
9. 부호이론적 다중 압축센싱 시스템 개발, 중견연구자 후속연구지원사업(도전),
   - 한국연구재단, 2010.05 ~ 2018.04,

Acknowledgements

광주과학기술원 (GIST)에서 2015 년 시작하여 2025 년 마치게 된 박사학위과정동안 따뜻한 격려와 아낌없는 지도를 보내주신 모든 분들께 진심으로 감사의 마음을 전합니다. 이 기간 동안 다양한 배경, 문화, 행동 및 사고방식을 가진 여러 사람들을 만나 함께 생활하는 귀중한 시간을 통해 서로 이해하고 소통하는 방법에 대해 많이 배우고 여러 각도로 생각해보는 기회를 얻을 수 있었습니다.

먼저, 연구의 큰 방향을 제시해 주셨을 뿐 만 아니라 세세한 부분까지 지도해주신 지도교수 이흥노 교수님께 깊은 존경과 감사를 드립니다. 교수님께서 강조하신 늘 읽고, 생각하고, 쓰는 원칙이 연구가 막힐 때 마다 되새길 수 있는 표지석이 되었습니다. 학문적 엄격함의 중요성을 일깨워 주셨고, 어려움 속에서도 포기하지 않도록 격려를 아끼지 않으셨습니다. 또한 학문 외적으로도 이루 다 말할 수 없는 가르침을 얻었습니다.

학위논문 심사위원으로 참여해주신 신종원 교수님, 황의석 교수님, 인하대학교 이용우 교수님, 한국전자통신연구원 박상준 박사님께도 감사의 말씀을 올립니다. 심사위원분들의 귀중한 조언과 질문은 본 연구의 완성도를 한층 높이는 데 큰 도움이 되었습니다. 특히 논문 작성 단계 에서부터 많은 도움을 주신 박상준 박사님께 다시 한번 깊은 감사의 말씀을 전합니다.

또한 INFONET 연구실의 모든 동료 연구원들과 선·후배님들께 감사드립니다. 공동작업 및 강평에서 나눴던 토론은 저의 큰 자산이 되었으며, 때로는 밤 늦게까지 함께 연구실을 지키며 보내던 시간들이 있었기에 오늘의 결과가 가능했습니다. 졸업 후에도 꾸준히 관심을 기울여 주신 성진택 선배님, 신영학 선배님, 이승찬 선배님, 박상준 선배님, 강주성 선배님께 정말 감사합니다. 최충재 선배님, 정효영 선배님께도 감사의 마음을 전합니다. 스스로의 연구에 바빠 잘 돌봐주지 못했음에도 불구하고 학위 과정을 잘 해쳐 나가고, 오히려 저에게 많은 가르침을 준 모든 후배들께도 감사합니다. 특히 정신 없었던 마지막 한 해에 많이 도와준 수민, 승민, 영인 후배님들께 고마운 마음을 전하고 싶습니다.

무엇보다도, 많은 관심을 기울이지 못했음에도 불구하고 변함없는 사랑과 믿음으로 저를 지지해 준 가족에게 깊은 감사를 전합니다. 속 마음을 잘 털어놓지 못하는 제가 답답하셨을 텐데도 내색하시지 않고 언제나 지지해주신 아버지와 어머니, 동생의 끝없는 헌신이 긴 학위과정을 해쳐내 가는데 든든한 버팀목이 되어주었습니다. 정말 말로 이루 다 표현할 수 없을 만큼 감사합니다.

끝으로, 이 글에 다 담지 못한 감사의 마음을 모든 분께 전합니다. 여러분의 격려와 도움이 없었다면 오늘의 결과를 얻을 수 없었을 것입니다. 앞으로도 연구자의 길에서 쌓은 배움을 사회와 학문 발전에 환원할 수 있도록 최선을 다하겠습니다.

2025 년 5 월 최해웅 올림.