Software Engineering and Applications 软件工程与应用, 2025, 14(4), 821-831
Published Online August 2025 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2025.144072

Hans汉斯

基于Kyber交换加密水印算法设计

蒋　雪，吕经浪，李子臣

北京印刷学院信息工程学院，北京

收稿日期：2025年6月24日；录用日期：2025年8月1日；发布日期：2025年8月13日

摘　要

随着量子计算的迅速发展，传统的同态加密水印算法面临着安全威胁。本文提出了一种基于Kyber密文域同态加密水印算法。通过将Kyber加密算法与Patchwork水印算法相结合，确保信息在传输时不被泄露，同时实现加密与水印嵌入运算先后顺序的交换。实验结果表明，该算法确保了水印嵌入与数据加密的顺序不影响密文数据的生成和水印提取，不仅在密文域可以提取水印，解密后也可以提取水印。

关键词

Kyber算法，同态加密，数字水印，交换加密水印

Design of Exchange Encryption Watermarking Algorithm Based on Kyber

Xue Jiang, Jinglang Lyu, Zichen Li

School of Information Engineering, Beijing Institute of Graphic Communication, Beijing

Received: Jun. 24th, 2025; accepted: Aug. 1st, 2025; published: Aug. 13th, 2025

Abstract

The rapid development of quantum computing, traditional homomorphic encryption watermark algorithms face security threats. This paper proposes a watermark algorithm based on Kyber ciphertext domain homomorphic exchange encryption. By combining Kyber encryption with Patchwork watermark algorithm, we ensure that information is not leaked during transmission, and at the same time, the order of encryption and watermark embedding operations is exchanged. Experimental results show that the algorithm ensures that the order of watermark embedding and data encryption does not affect the generation and watermark extraction of ciphertext data. Not only can

文章引用: 蒋雪, 吕经浪, 李子臣. 基于 Kyber 交换加密水印算法设计[J]. 软件工程与应用, 2025, 14(4): 821-831.
DOI: 10.12677/sea.2025.144072

the watermark be extracted in the ciphertext domain, but the watermark can also be extracted after decryption.

Keywords

Kyber Algorithm, Homomorphic Encryption, Digital Watermark, Exchange Encrypted Watermarks

1. 引言

在数字信息飞速传播的当下，多媒体数据在网络中的传输极为频繁，水印和加密技术的结合在多个关键领域有着重要的作用。在数字版权管理领域，内容创作者和发行商通过将版权信息以水印的形式嵌入数字作品，并利用加密技术保护水印和作品，防止盗版和未经授权的传播[1]。在医疗数据共享场景中，医院和科研机构需要在保证患者隐私的前提下，将医疗影像等数据分享给合作方进行研究分析，水印与加密技术结合可确保数据来源可追溯且内容不被非法获取[2]。在金融交易领域，电子票据、合同等重要文件需确保其完整性和真实性，水印和加密技术结合能够有效防止文件被篡改或伪造[3]。

同态加密是密码学领域的一项重要技术，对密文进行特定的计算操作，其结果与对明文进行相同计算操作后再加密的结果一致[4]。这一特性使得同态加密在云计算、数据外包等场景中具有巨大的应用潜力，能够在保证数据机密性的前提下实现数据的高效处理[5]。然而，传统的同态加密算法在面对复杂多变的安全威胁时，逐渐暴露出一些局限性[6]。

为了进一步保障信息安全，将同态加密与水印技术相结合，形成了同态加密水印算法。现有的基于公钥加密方案的水印算法，在一定程度上提高了水印嵌入与提取过程中的安全性[7]。但是，基于经典密码体制的算法在面对量子计算机强大的计算能力时，其安全性正面临严峻挑战[8]。量子计算机能够在短时间内破解传统加密算法，这对基于经典公钥加密方案的水印算法构成了巨大的安全威胁[9]。

CEW (Commutative Encryption Watermarking)交换加密水印算法是一种将加密和水印技术结合，实现二者操作顺序可交换的方案[10]。传统的水印加密方案将加密与水印固定顺序，这存在显著局限性，一旦改变可能导致密文异常或水印无法正确提取。而 CEW 交换加密水印方案打破了这一限制，无论先进行加密还是先进行水印嵌入，最终生成的密文数据和水印提取的结果都不受影响[11]。在军事通讯和医疗领域中具有重要作用。

本文创新性地提出了一种基于 Kyber 加密方案[12]交换加密水印算法。Kyber 是一种基于格的后量子密码算法，具备较强的抗量子攻击能力，能够有效保障加密数据在水印嵌入与提取过程中的安全性[13]。Kyber 加密与 Patchwork 水印算法[14]相结合的 CEW 方案，不仅确保了信息在传输过程中的安全性，防止信息被泄露，还实现了加密与水印嵌入操作的互换。这种算法设计确保了水印嵌入与数据加密的顺序不影响密文数据的生成和水印提取，显著提升了水印嵌入的机密性和多媒体数据的安全性，在量子环境下展现出了显著的抗攻击优势。

2. 理论基础

本节将对 CEW [10]，Kyber 密码体制[13]所具备的加法同态特性以及 Patchwork 水印算法[14]展开论述，并详细介绍本文提出的基于 Kyber 交换加密水印算法的基本原理。

2.1. 交换加密水印

交换加密水印方案是将数字水印和加密技术结合的一个重要方向，主要用于在加密数据中嵌入水印，并且保证水印提取能够在数据加密后进行，且不泄露数据内容。CEW 方案的核心思想是加密与水印嵌入的操作是可互换的，而且不影响水印的提取。CEW 方案能够有效防止数据篡改和水印攻击。下图 1 是交换加密水印算法的流程图。

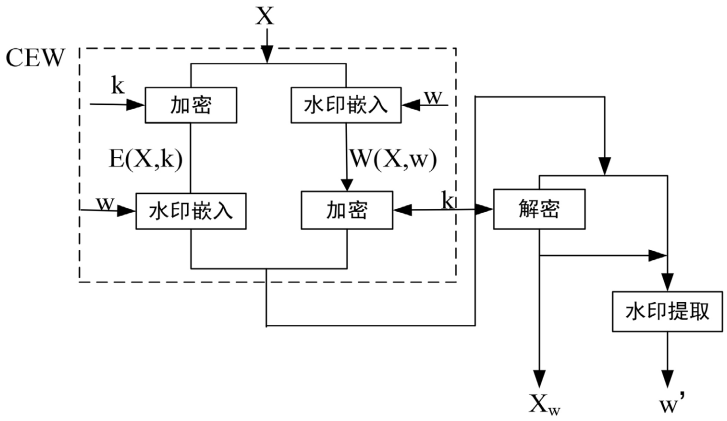


Figure 1. Flowchart of the exchange encryption watermarking algorithm
图 1. 交换加密水印算法流程图

图 1 中，X 为原始数据，E 是加密函数，W 是数字水印函数，k 是加密密钥，w 是嵌入的水印信息。一方面，使用加密密钥 k 对 x 进行加密操作，生成加密后的数据，之后将水印信息嵌入其中；另一方面，可先将水印信息嵌入得到含水印的图像，再使用加密密钥 k 进行加密。之后，对经过上述操作的数据进行解密得到含水印的明文载体 X_w 以恢复原始数据，同时也可以在密文域中提取水印信息 w'，用于验证版权归属等。

当前，多数 CEW 算法为防止加密与水印相互干扰，都通过操作的独立性的来实现。这种方式下，水印操作数对用户完全可见，由于缺乏加密保护，使得多媒体信息数据的安全性有所降低。针对这一问题，人们将正交分解应用于基于独立操作的 CEW (CEWod) [15]，试图提升多媒体信息的安全性。然而，CEWod 本质上仍属于基于独立操作的 CEW 算法，并没有从根本上解决 CEW 的安全问题。在本文中，运用同态加密技术，不仅弱化了对特定加密算法和水印算法的限制，还进一步增强了水印方案的安全性。

2.2. Kyber 加法同态加密算法

Kyber 加法同态加密算法基于整数编码技术，实现了整数加法同态，首先通过编码将整数映射到明文多项式上，然后进行后续的加密操作。

2.2.1. 整数编码

编码算法 $Encode\ (M)$：

令 M 表示整数，其二进制表示为 $bin\ (M)$；

将 $bin\ (M)$分组为长度为 $dp-1$ 的子串，并在每个子串前面添加一个 0，得到向量：

$$vec(M)=\underbrace{(0,bin_1,0,bin_2,\cdots,0,bin_l)}_{l\text{ 个分组}} \tag{1}$$

其中，bin_i 表示第 i 个子串，l 为分组的总数；

将每个向量元素转换为十进制整数：

$$vec(M)' = \underbrace{(0, dec_1, 0, dec_2, \cdots, 0, dec_l)}_{l\text{ 个分组}} \tag{2}$$

其中，dec_i 表示将二进制串 bin_i 转换为十进制的结果；

如果向量长度 $l < n$，则在向量前面补零，使其长度达到 n，即

$$encode(M) = \underbrace{(0, \cdots, 0, vec(M)')}_{n\text{ 个分组}} \tag{3}$$

解码算法 $decode(m)$：

令整数向量 $m = (m_0, m_1, \cdots, m_{n-1})$，取其逆序 $m_{reverse} = (m_{n-1}, m_{n-2}, \cdots, m_1, m_0)$，计算：

$$decode(m) = \sum_{i=0}^{n-1} m_{reverse}[i] \cdot \left(2^{dp-1}\right)^i \tag{4}$$

2.2.2. 算法步骤

算法主要包含四个部分：密钥生成，加密，解密和同态加法，参数集如表 1。

Table 1. Algorithm parameter set
表 1. 算法参数集

符号	定义
n	环 R_q 上多项式的次数
k	模格的维数
q	环 R_q 上多项式的系数模
η_1，η_2	噪声分布参数
du，dv，dp	压缩参数

输入安全参数 n (须满足为 2 的幂)，产生算法参数集 *parameters*：模格的维数 k；多项式模 q (须满足 $q = 1 \bmod 2n$)；多项式环 $R_q = Z_q[x]/x^n + 1$；噪声分布 β_η。

本方案的具体算法包含四个部分：密钥生成，加密，解密和同态加法。

a) 密钥生成 KeyGen(*parameters*)：

输入参数集 *parameters* 中的模维数 k，多项式模 q，以及噪声分布参数 η_1，η_2。

取样生成随机矩阵和向量：$\boldsymbol{A} \xleftarrow{R} R_q^{k \times k}$；$\boldsymbol{s} \xleftarrow{R} \beta_{\eta_1}^k$，$\boldsymbol{e} \xleftarrow{R} \beta_{\eta_1}^k$。

计算公钥：$\boldsymbol{t} := \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$。

输出公钥 $pk := (t, A)$，私钥 $sk := s$。

b) 加密算法 $Enc(pk, M)$：

输入公钥 pk，整数 $M \in \left[0, 2^{n*(dp-1)} - 1\right]$，以及参数集 *parameters* 中的压缩参数 du，dv，dp。取样产生随机向量和噪声向量：$\boldsymbol{r} \xleftarrow{R} \beta_{\eta_1}^k$，$\boldsymbol{e}_1 \xleftarrow{R} \beta_{\eta_1}^k$，$e_2 \xleftarrow{R} \beta_{\eta_1}$。

对 M 进行编码 $m = encode(M)$，计算：

$$\boldsymbol{u} := \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r} + \boldsymbol{e}_1, v := \boldsymbol{t}^{\mathrm{T}}\boldsymbol{r} + e_2 + Decompress_q(m, dp) \tag{5}$$

进行压缩：

$$c := (\boldsymbol{u}', v') = \left(Compress_q(\boldsymbol{u}, du), Compress_q(v, dv)\right) \tag{6}$$

输出密文 $c := (\boldsymbol{u}', v')$。

c) 解密算法 *Dec* (*sk*, *c*)：

输入私钥 *sk*，密文 *c*，以及参数集 *parameters* 中的压缩参数 *du*，*dv*。对密文进行解压缩：

$$\boldsymbol{u} = Decompress_q(\boldsymbol{u}', du), v = Decompress_q(v', dv) \tag{7}$$

计算：

$$m := Compress_q(v - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}, dp) \tag{8}$$

对 *m* 进行解码 $M = decode(m)$。输出解密结果 *M*。

d) 同态加法 *Add* (c_1, c_2)：

输入两组密文 c_1，c_2，以及参数集 *parameters* 中的压缩参数 *du*，*dv*。对两组密文求和：

$$c' = c_1 + c_2 = (\boldsymbol{u}_1, v_1) + (\boldsymbol{u}_2, v_2) = (\boldsymbol{u}_1 + \boldsymbol{u}_2, v_1 + v_2) \tag{9}$$

输出密文 *c*’。

2.3. Patchwork 水印算法

Patchwork 是一种基于图像的水印算法，通常用于图像数字水印的研究中。它主要通过改变图像的像素点来进行水印嵌入，目的是在保证水印信息不可见的情况下，提供足够的鲁棒性。它能够抵抗图像的常见攻击，如裁剪、压缩、噪声添加等。Patchwork 水印的核心是分别改变图像不同区域的像素值，通常这些区域具有一定关系，被改变的这些区域称为“patch”。Patchwork 水印算法具体流程为：

a) 图像分割：首先将图像分成两个区域，且两个区域的像素点接近。这两个区域中的数据分别是集合 $X = \{x_{i,j}\}$ 和 $Y = \{y_{i,j}\}$，$i \in [1, n]$，$j \in [1, n]$，改变每个区域的像素点用于水印的嵌入。

b) 水印嵌入：对于 *X* 区域，将所有像素点增加 *a*，对于 *Y* 区域，将所有像素点减少 *a*，使得这些区域的像素点发生微小的变化，这些变化是不可感知的。

c) 水印提取：通过计算均值来确定是否含有水印信息，计算均值的公式为：

$$s = \frac{1}{n^2}\sum_{i}^{n}\sum_{j}^{n}\left(x_{i,j}^{w} - y_{i,j}^{w}\right) \tag{10}$$

其中 $x_{i,j}^{w}$ 和 $y_{i,j}^{w}$ 分别是嵌入 *x* 水印后的像素值：$x_{i,j}^{w} = x_{i,j} + a$，$y_{i,j}^{w} = y_{i,j} - a$。

假设 $a = 1$，要嵌入的载体图像为 8 * 8 的灰度图像，根据 Patchwork 水印算法规则，嵌入的水印算法为：

$$\begin{cases} x_{i,j}^{w} = x_{i,j} + 1 \\ y_{i,j}^{w} = y_{i,j} - 1 \end{cases} \tag{11}$$

嵌入水印后，计算均值提取水印：

$$s = \frac{1}{8^2}\sum_{i}^{n}\sum_{j}^{n}\left(x_{i,j}^{w} - y_{i,j}^{w}\right) \tag{12}$$

其中，*s* 的值将决定是否带有载波水印，若 $s \approx 2$ 含有水印载波，水印信息为 1，若 $s \approx 0$ 表示无水印，水印信息为 0。

3. 基于 Kyber 与 Patchwork 交换加密水印算法

Kyber 加密算法和 Patchwork 水印算法的结合可以形成一种安全且具有鲁棒性的加密水印方案。在此

方案中，Kyber 算法用于加密载体信息，可以实现在嵌入水印前加密，也可以实现先嵌入水印后加密；Patchwork 算法则用于将水印信息嵌入原始图像或加密图像中，在水印提取时实现在密文域提取水印和在解密后的明文域提取水印。算法总体结构如图 2 所示。

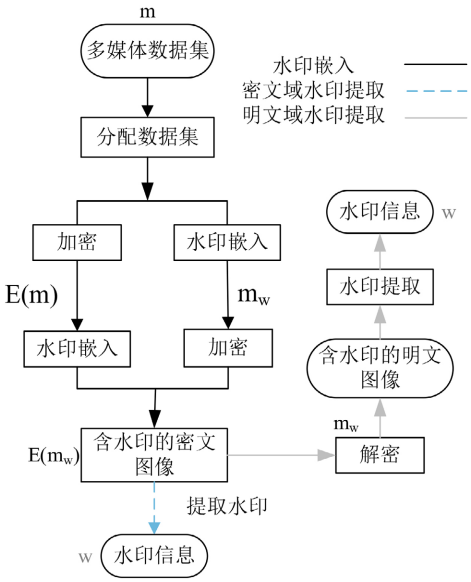


Figure 2. Algorithm overall structure diagram
图 2. 算法总体结构图

3.1. 交换加密水印算法基本原理

基于 Kyber 加密算法与 Patchwork 水印算法的 CEW 算法包括如下步骤：

a) 数据集合选取：从载体图像中筛选特定数据子集，依据像素点坐标之和的奇偶性，将所选数据重新划分为两个互斥集合，分别记为 $X=\{x_{i,j}\}$ 和 $Y=\{y_{i,j}\}$，其中 $i\in[1,n]$，$j\in[1,n]$，在实际图像的像素空间中，集合 X 内的像素值 $x_{i,j}$ 对应像素点的坐标之和为偶数，而集合 Y 内的像素值 $y_{i,j}$ 对应像素点的坐标索引之和为奇数。

b) 水印嵌入：对集合 X 中的所有像素点执行加法操作，增加值为 a；对集合 Y 中的所有像素点执行减法操作，减少值亦为 a。参数 a 的取值需通过严谨的实验分析与理论推导确定，以在保障水印提取准确率的同时，最大程度维持嵌入水印后载体图像的视觉质量与结构完整性，确保图像的各项质量评估指标，如峰值信噪比(*PSNR*)、误码率(*BER*)等，处于可接受范围。

c) 水印提取：过程同公式(10)。

3.2. 含水印密文载体的生成

首先在原始载体数据中选择一些数据，然后计算这些像素值下标的值，再按照下标值的奇偶性分配组成两个集合 $\{x_{i,j}\},\{y_{i,j}\},i\in[1,n],j\in[1,n]$。

3.2.1. 先嵌入水印后加密

在明文中嵌入水印：

$$\begin{cases} x_{i,j}^{w}=x_{i,j}+a \\ y_{i,j}^{w}=y_{i,j}-a \end{cases} \tag{13}$$

其中，$x_{i,j}^{w}, y_{i,j}^{w}$ 是带水印的明文。

然后对含水印的明文进行加密：

$$\begin{cases} c_{x'(i,j)} = Enc\left(pk, x_{i,j}^{w}\right) = \left(\boldsymbol{u}_{x'(i,j)}, v_{x'(i,j)}\right) \\ c_{y'(i,j)} = Enc\left(pk, y_{i,j}^{w}\right) = \left(\boldsymbol{u}_{y'(i,j)}, v_{y'(i,j)}\right) \end{cases} \tag{14}$$

其中，$m_{x'(i,j)} = encode\left(x_{i,j}^{w}\right)$，$\boldsymbol{u}_{x'(i,j)} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r} + \boldsymbol{e}_1$，$v_{x'(i,j)} = \boldsymbol{t}^{\mathrm{T}}\boldsymbol{r} + e_2 + Decompress_q\left(m_{x'(i,j)}, dp\right)$。

$y_{i,j}^{w}$ 的加密过程与 $x_{i,j}^{w}$ 相同。

3.2.2. 先加密后嵌入水印

对载体图像的每个像素点加密，具体过程与含水印的明文进行加密类似：

$$\begin{cases} c_{x(i,j)} = Enc\left(pk, x_{i,j}\right) = \left(\boldsymbol{u}_{x(i,j)}, v_{x(i,j)}\right) \\ c_{y(i,j)} = Enc\left(pk, y_{i,j}\right) = \left(\boldsymbol{u}_{y(i,j)}, v_{y(i,j)}\right) \end{cases} \tag{15}$$

其中，$m_{x(i,j)} = encode\left(x_{i,j}\right)$，$\boldsymbol{u}_{x(i,j)} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r} + \boldsymbol{e}_1$，$v_{x(i,j)} = \boldsymbol{t}^{\mathrm{T}}\boldsymbol{r} + e_2 + Decompress_q\left(m_{x(i,j)}, dp\right)$。

$y_{i,j}$ 的加密过程与 $x_{i,j}$ 相同。

同理，对水印信息 a 也进行加密：

$$\begin{cases} c_a = Enc\left(pk, a\right) = \left(\boldsymbol{u}_a, v_a\right) \\ c_{-a} = Enc\left(pk, -a\right) = \left(\boldsymbol{u}_{-a}, v_{-a}\right) \end{cases} \tag{16}$$

然后，在密文中嵌入水印：

$$\begin{cases} c_{x''(i,j)} = c_{x(i,j)} + c_a \\ c_{y''(i,j)} = c_{y(i,j)} + c_{-a} \end{cases} \tag{17}$$

得到的 $c_{x''(i,j)}, c_{y''(i,j)}$ 即含有水印的密文载体图像。

3.3. 水印信息的提取

3.3.1. 在解密后的明文提取水印

首先引入如下的结论。

定理：对利用上述先嵌入水印后加密与先加密后嵌入水印所得到的含水印的密文进行解密时，当密钥相同时，解密结果是相同的，即：

$$\begin{aligned} Dec\left(sk, c_{x'(i,j)}\right) &= Dec\left(sk, c_{x''(i,j)}\right) = x_{i,j} + a \\ Dec\left(sk, c_{y'(i,j)}\right) &= Dec\left(sk, c_{y''(i,j)}\right) = y_{i,j} - a \end{aligned} \tag{18}$$

证明：$c_{x'(i,j)}$ 为 $x_{i.j}^{w}$ 加密的结果，所以有

$$Dec\left(sk, c_{x'(i,j)}\right) = x_{i.j}^{w} = x_{i,j} + a \tag{19}$$

$c_{x''(i,j)}$ 为 $c_{x(i,j)}$ 与 c_a 进行加法同态加密的结果，所以此处证明算法加法同态的正确性：

令 $m = v - \mathbf{s}^{\mathrm{T}}\mathbf{u}$，对于两组由相同公钥加密得到的密文 $c_1 = \left(u_1, v_1\right)$，$c_2 = \left(u_2, v_2\right)$，做加法时，已经有 $m_1 + m_2 = m_3$，对等式左边，有

$$m_1 + m_2 = \left(v_1 - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}_1\right) + \left(v_2 - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}_2\right) = \left(v_1 + v_2\right) - \boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right) \tag{20}$$

对等式右边，有

$$m_3 = v_3 - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}_3 \tag{21}$$

由式(20) (21)可得

$$\left(v_1 + v_2\right) - \boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{u}_1 + \boldsymbol{u}_2\right) = v_3 - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}_3 \tag{22}$$

因此在密文上，$c_3 = \left(u_3, v_3\right) = \left(u_1 + u_2, v_1 + v_2\right)$，满足加法同态，即

$$Dec\left(sk, c_{x''(i,j)}\right) = Dec\left(sk, c_{x(i,j)}\right) + Dec\left(sk, c_a\right) = x_{i,j} + a \tag{23}$$

同理也可证明 $Dec\left(sk, c_{y'(i,j)}\right) = Dec\left(sk, c_{y''(i,j)}\right) = y_{i,j} - a$ 的正确性，定理成立。

水印提取过程同式(10)。

3.3.2. 在密文中提取水印

对于含有水印的密文载体图像：$c_{x''(i,j)}, c_{y''(i,j)}$

直接计算：

$$\begin{cases} c_{sumx} = \sum_{i}^{n}\sum_{j}^{n} c_{x''(i,j)} \\ c_{sumy} = \sum_{i}^{n}\sum_{j}^{n} c_{y''(i,j)} \end{cases} \tag{24}$$

由于

$$\begin{cases} n^2 s_x = Dec\left(sk, c_{sumx}\right) = Dec\left(sk, \sum_{i}^{n}\sum_{j}^{n} c_{x''(i,j)}\right) \\ n^2 s_y = Dec\left(sk, c_{sumy}\right) = Dec\left(sk, \sum_{i}^{n}\sum_{j}^{n} c_{y''(i,j)}\right) \end{cases} \tag{25}$$

所以当 $n^2 s_x \approx n^2 s_y$ 时，水印信息为 0；当 $n^2 s_x - n^2 s_y \approx 2n^2 a$ 时，水印信息为 1。

4. 实验结果分析

4.1. 实验设置

本次实验选用一幅 256 * 256 大小的 lena 灰度图作为原始载体，运用 Patchwork 水印算法和 Kyber 加密算法进行加密和嵌入水印，旨在探索并实现嵌入水印和加密的顺序可以交换，并能抵抗量子攻击。

峰值信噪比(*PSNR*)是衡量水印算法性能的重要参数，实验中通过该指标来评估含水印图像与原图的失真程度，*PSNR* 值越高，说明图像质量损失越小，水印隐蔽性越强。假设原始图像用 *r* 表示，解密后含水印的图像记为 *r*’，使用(*i*,*j*)表示图像的像素坐标，其中 *h* 代表图像的高度，*w* 表示图像的宽度，*M* 为嵌入的水印总比特数。

峰值信噪比(*PSNR*)的计算公式为：

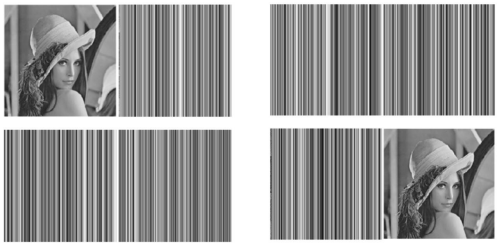
$$PSNR = 10 \times \lg \frac{h \times v \times 255^2}{\sum_{i=1}^{h}\sum_{j-1}^{w}\left[r\left(i,j\right) - r'\left(i,j\right)\right]^2} \tag{26}$$

在数字水印技术应用中，原始图像的像素值通常在 0 至 255 区间。但是，当采用加密算法对图像进行处理时，可能会导致部分像素值突破常规范围，出现负值或超出 255 上限的情况，但是本算法使用 Kyber 算法加密操作，不会出现像素值溢出的情况。Kyber 算法所采用的参数中 q = 3329 超出了像素值表示范

围，无法可视化地表示出来，本文采取了标准化的方法来解决该问题。在密文输出时，将 0~3329 等比例转化到 0~255 再输出为图像，当进行密文上的运算时，首先进行逆运算将 0~256 等比例转化到 0~3329，从而保证运算的正确性。

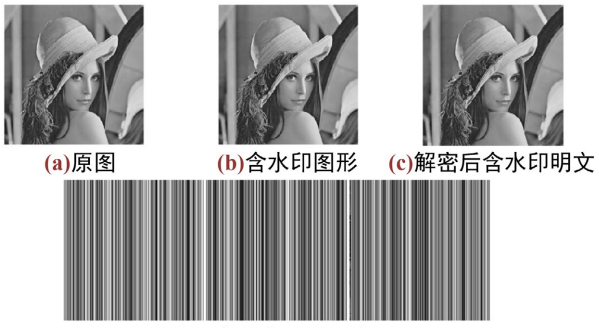
4.2. 实验结果分析

图 3(a)是原始载体图像和利用 Kyber 算法加密后得到的密文图像，图 3(b)是嵌入水印后的含水印密文图像和解密后含水印的明文图像，含水印的明文图像计算得到的 *PSNR* 值为 25.41 dB。其中，由于 Kyber 算法的密文膨胀原因，对 256 × 256 的图像加密将得到较多的密文数据，图 3(b)中的密文图像以及后续的密文图像仅为 256 × 1 部分像素所对应密文的图像，剩余 256 × 255 部分密文图像与之类似，在此不展示。图 4 分别是原始图像，嵌入水印信息之后的含水印图像和解密后的含水印图像以及密文含水印图像，图 4(b)的 *PSNR* 值为 48.13 dB，解密后的含水印图像 *PSNR* 值为 43.62 dB，实验结果表明，该算法在明文域和密文域均能完整提取水印内容并准确还原原始图像。经解密处理后的水印图像效果良好。



(a)原图与图像密文　(b)含水印密文和含水印明文

Figure 3. Experimental results in the encrypted domain
图 3. 加密域下的实验结果图



(a)原图　(b)含水印图形　(c)解密后含水印明文

(d)嵌入水印后加密

Figure 4. Experimental results in plaintext domain
图 4. 明文域下的实验结果图

上述实验为嵌入一个比特的水印信息的实验结果，此外还可嵌入字母和图片作为水印信息嵌入图像，嵌入后的水印信息在明文域和密文域都能准确的提取出来。

4.3. 水印算法的性能测试

为验证算法的适用性与抗鲁棒性，本研究另选取 lena、cameraman 两幅 256 × 256 标准测试图像进行实验。计算不同水印嵌入量下的 *PSNR* 值，其中表 2 记录密文域不同嵌入容量下的 *PSNR*，表 3 反映明文域不同嵌入容量下的 *PSNR* 值。测试发现：当嵌入容量减少时，明文域和密文域内的 *PSNR* 值均呈现提

升趋势。在嵌入容量较低时，含水印的图像解密后能较好的恢复原图像，在嵌入容量较高时，也能获得比较清晰的图像。

Table 2. *PSNR* (dB) under different embedding capacities in the ciphertext domain
表 2. 密文域不同嵌入容量下的 *PSNR* (dB)

原始图像	0.0312	0.0156	0.0078	0.0039
Lena	25.41	28.13	31.17	34.15
cameraman	24.20	27.33	31.60	33.16

Table 3. *PSNR* (dB) under different embedding capacities in plaintext domain
表 3. 明文域不同嵌入容量下的 *PSNR* (dB)

原始图像	0.0625	0.0312	0.0156	0.0078	0.0039
Lena	25.05	30.07	36.09	42.11	48.13
cameraman	24.20	30.07	36.09	42.11	48.13

与其他加密域的水印算法文献[10]、文献[16]和文献[17]进行比较结果见表 4。由表 4 可见，在无水印攻击时，以 256 × 256 的 Lena 灰度图像为例，与文献[10] [16]和[17]中的算法比较，本文水印算法在明文域的 *PSNR* 值为 48.13 dB，文献[10] [16]和[17]的 *PSNR* 值分别为 48.13 dB，42.81 dB 和 50.38 dB，本文算法求得的 *PSNR* 值，与其他算法基本上差不多，说明提取出的水印图像与原始图像差异较小，水印图像效果较好，水印图像质量得到了改善，同时还能实现密文域下水印的提取，并且本文算法还具有一定的抗量子攻击性能，由此可见，本文算法的整体性能得到了改善。

Table 4. Comparison of this algorithm with other literature algorithms
表 4. 本文算法与其他文献算法比较

水印算法	水印图像 *PSNR* (dB)	水印形式	鲁棒性测试
文献[10]	48.13	0/1	噪声|压缩
文献[16]	50.38	0/1	噪声|压缩|剪切
文献[17]	42.82	二值图像	无测试
本文算法	48.13	0/1	噪声

5. 结论与展望

本文针对传统同态加密水印算法在量子计算环境下的潜在安全风险，提出了一种基于 Kyber 后量子加密方案的加法同态交换加密水印算法。通过将 Kyber 加密算法与经典 Patchwork 水印嵌入技术相结合，实现了加密与水印操作顺序的可交换性，确保密文数据在传输和处理过程中的机密性。基于数字水印提供了数字版权保护的方案。因此，本方案为图像、视频、音频等数字文件的机密性、完整性、真实性，以及版权保护提供了较为完善的解决方案。

下一步的研究着重于在抗量子攻击的同时提高算法的效率，或增强算法的鲁棒性，来进一步提高算法的性能。

基金项目

国家自然科学基金(62472040)；中国版权保护中心版权研究课题(BQ2024017)。

参考文献

[1] 赵学军, 曹天宇, 李琳. 数字图像水印综述[J]. 工业控制计算机, 2025, 38(1): 108-110.

[2] Li, H., Yang, Y., Dai, Y., Yu, S. and Xiang, Y. (2020) Achieving Secure and Efficient Dynamic Searchable Symmetric Encryption over Medical Cloud Data. *IEEE Transactions on Cloud Computing*, **8**, 484-494. https://doi.org/10.1109/tcc.2017.2769645

[3] Olaiya, O.P., Adesoga, T.O., Adebayo, A.A., *et al*. (2024) Encryption Techniques for Financial Data Security in Fintech Applications. *International Journal of Science and Research Archive*, **12**, 2942-2949. https://doi.org/10.30574/ijsra.2024.12.1.1210

[4] Marcolla, C., Sucasas, V., Manzano, M., *et al*. (2022) Survey on Fully Homomorphic Encryption, Theory, and Applications. *Proceedings of the IEEE*, **110**, 1572-1609.

[5] Mahato, G.K. and Chakraborty, S.K. (2021) A Comparative Review on Homomorphic Encryption for Cloud Security. *IETE Journal of Research*, **69**, 5124-5133. https://doi.org/10.1080/03772063.2021.1965918

[6] Bernstein, D.J. and Lange, T. (2017) Post-Quantum Cryptography. *Nature*, **549**, 188-194. https://doi.org/10.1038/nature23461

[7] Liang, X., Xiang, S., Yang, L. and Li, J. (2021) Robust and Reversible Image Watermarking in Homomorphic Encrypted Domain. *Signal Processing*: *Image Communication*, **99**, Article ID: 116462. https://doi.org/10.1016/j.image.2021.116462

[8] Pillai, S.E.V.S. and Polimetla, K. (2024) Analyzing the Impact of Quantum Cryptography on Network Security. 2024 *International Conference on Integrated Circuits and Communication Systems* (*ICICACS*), Raichur, 23-24 February 2024, 1-6. https://doi.org/10.1109/icicacs60521.2024.10498417

[9] Azhari, R. and Salsabila, A.N. (2024) Analyzing the Impact of Quantum Computing on Current Encryption Techniques. *IAIC Transactions on Sustainable Digital Innovation* (*ITSDI*), **5**, 148-157. https://doi.org/10.34306/itsdi.v5i2.662

[10] Jiang, L. (2018) The Identical Operands Commutative Encryption and Watermarking Based on Homomorphism. *Multimedia Tools and Applications*, **77**, 30575-30594. https://doi.org/10.1007/s11042-018-6142-y

[11] Ren, N., Tong, D., Cui, H., Zhu, C. and Zhou, Q. (2022) Congruence and Geometric Feature-Based Commutative Encryption-Watermarking Method for Vector Maps. *Computers & Geosciences*, **159**, Article ID: 105009. https://doi.org/10.1016/j.cageo.2021.105009

[12] 吕经浪, 李子臣. 基于 Kyber 的小数加法同态密码体制的研究[J]. 计算机技术与发展, 2024, 34(11): 101-108.

[13] Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., *et al*. (2018). CRYSTALS-Kyber: A CCA-Secure Module-Lattice-Based KEM. 2018 *IEEE European Symposium on Security and Privacy* (*EuroS&P*), London, 24-26 April 2018, 353-367. https://doi.org/10.1109/eurosp.2018.00032

[14] Bender, W., Gruhl, D., Morimoto, N. and Lu, A. (1996) Techniques for Data Hiding. *IBM Systems Journal*, **35**, 313-336. https://doi.org/10.1147/sj.353.0313

[15] Jiang, L., Xu, Z. and Xu, Y. (2012) Commutative Encryption and Watermarking Based on Orthogonal Decomposition. *Multimedia Tools and Applications*, **70**, 1617-1635. https://doi.org/10.1007/s11042-012-1181-2

[16] 方立娇, 李子臣, 丁海洋. 基于 ElGamal 的同态交换加密水印算法[J]. 计算机系统应用, 2021, 30(5): 234-240.

[17] 项世军, 罗欣荣, 石书协. 一种同态加密域图像可逆水印算法[J]. 计算机学报, 2016, 39(3): 571-581.