

# 面向移动支付防盗刷的动态二维码水印算法

李红棒<sup>1,2</sup>, 陈家乐<sup>1,2</sup>, 董理<sup>1,2</sup>, 王让定<sup>1,2</sup>, 孙巍巍<sup>3</sup>, 张玉书<sup>4</sup>

<sup>1</sup> 宁波大学信息科学与工程学院 宁波 中国 315211

<sup>2</sup> 浙江省移动网络应用技术重点实验室 宁波 中国 315211

<sup>3</sup> 阿里巴巴集团-橙盾信息科技有限公司 杭州 中国 311121

<sup>4</sup> 南京航空航天大学计算机科学与技术学院 南京 中国 210016

**摘要** 二维码付款在小额免密支付中被广泛应用,但是它所面临的“隔空盗刷”问题却是一个重要的安全隐患。“隔空盗刷”是指不法分子趁消费者不备,通过盗摄非法获取消费者的付款码,进而对付款码进行扫描,盗取财产。针对这一问题,本文提出了一种面向移动支付防盗刷的动态二维码水印算法。首先,付款端根据从服务器获取的原始支付令牌生成原始水印序列和新的支付令牌,对原始水印序列使用 $(t,n)$ 门限秘密共享技术生成多份水印序列。然后,对每份水印序列进行校验和卷积编码,生成多份待嵌入的水印序列并将其以半鲁棒的方式嵌入到由新的支付令牌生成的二维码图像中。最后,多张带有水印的付款码图像以预设帧率连续循环播放在亮码设备上。扫描端需捕获若干张带有水印的付款码图像后,才可提取出原始水印序列从而恢复原始支付令牌,完成付款验证。本文对现场支付场景、现场盗刷场景和协同盗刷场景等三种场景下的支付效率和防盗刷效率等指标进行了实验测试。在正常支付场景下,支付的成功率为100%,支付用时为 $1000 \pm 250\text{ms}$ 。在现场盗刷场景和协同盗刷场景下,本文各进行100次盗刷实验,记一分钟内成功提取原始水印序列为盗刷成功,结果显示攻击者均无法在指定时间内成功提取原始水印通过验证实现盗刷。本文所提出的动态二维码水印防盗刷方案可有效抵御盗刷犯罪,是一种有效且无需用户额外操作的安全解决方案,将有助于更好地保护消费者小额移动支付安全。

**关键词** 移动支付; 动态二维码; 半鲁棒数字水印; 秘密共享

中图分类号 TP309.7 DOI号 10.19363/J.cnki.cn10-1380/tn.2025.05.10

## Dynamic QR Code Watermarking Algorithm for Anti-Theft Mobile Payment

LI Hongbang<sup>1,2</sup>, CHEN Jiale<sup>1,2</sup>, DONG Li<sup>1,2</sup>, WANG Rangding<sup>1,2</sup>, SUN Weiwei<sup>3</sup>, ZHANG Yushu<sup>4</sup>

<sup>1</sup> Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China

<sup>2</sup> The Key Lab of Mobile Network Application Technology of Zhejiang Province, Ningbo 315211, China

<sup>3</sup> Alibaba Group Orange Shield Information Technology Co, Hangzhou 311121, China

<sup>4</sup> College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

**Abstract** QR code payment is widely used in the small amount of confidential payment, but it faces the problem of “remote theft”, which is an important security risk. The problem of “remote theft” refers to the illegal elements taking advantage of the consumer’s unpreparedness, illegally obtaining the consumer’s payment code through take photographs without permission, and then scanning the payment code and stealing the property. To address this problem, this paper proposes a dynamic QR code watermarking algorithm for mobile payment anti-theft. First, the payment side generates the original watermark and a new payment token based on the original payment token obtained from the server, and generates multiple copies of the watermark sequence using  $(t,n)$  threshold secret sharing technique for the original watermark sequence. Then, checksum and convolutional coding are performed on each watermark sequence to generate multiple copies of the watermark to be embedded and embed them in a semi-robust manner into the QR code image generated from the new payment token. Finally, the plurality of payment code images with watermarks are continuously looped on the bright code device at a preset frame rate. The scanning side needs to capture several payment code images with watermarks before the original watermark information can be extracted to recover the original payment token and complete the payment verification. In this paper, the payment efficiency and anti-theft efficiency and other indexes under three scenarios, including on-site payment scenario, on-site theft scenario and coordinated theft scenario, are tested experimentally. In the normal payment scenario, the success rate of payment is 100%, and the payment time is  $1000 \pm 250\text{ ms}$ . In the on-site theft scenario and collaborative theft scenario, this paper conducts 100 theft experiments each, and the successful extraction of the original wa-

通讯作者: 董理, 博士, 副教授, Email: dongli@nbu.edu.cn。

本课题得到浙江省自然科学基金(No. LY23F020011), 国家自然科学基金(No. 62171244, No. 61901237), 宁波市自然科学基金-青年博士创新研究项目(No. 2022J080)以及阿里巴巴创新研究计划资助。

收稿日期: 2023-11-09; 修改日期: 2023-12-14; 定稿日期: 2025-03-04

term within one minute is recognized as the success of theft, and the results show that the attackers are unable to successfully extract the original watermark to realize the theft through the authentication in the specified time. The dynamic QR code watermarking anti-theft scheme proposed in this paper can effectively defend against “remote theft” crimes, and is an effective security solution that does not require any additional operation by the user, which will help to better protect the security of consumers’ small amount of mobile payment.

**Key words** mobile payment; dynamic QR codes; semi-robust watermarking; secret sharing

## 1 引言

随着科技的发展,手机支付已经成为越来越多人的选择,二维码付款越来越普及。从全球来看,各国正在以比以往更快的速度迈向“无现金时代”<sup>[1-2]</sup>。

在线下零售支付的场景中,主要有基于 NFC (Near Field Communication) 和基于条码这两种付款方式。其中, NFC 支付<sup>[3-4]</sup>属于移动支付中的近场支付范畴,指消费者在购买商品或服务时,采用 NFC 技术,通过手机等手持设备完成支付交易。这种付款方式成本较高,要求消费者必须有一个与商家相互支持(硬件标准和协议)的 NFC 设备,在欧洲和美国等地区使用较多。条码支付<sup>[5]</sup>指的是银行业金融机构、非银行支付机构应用条码技术,实现收付款人之间货币资金转移的业务活动。条码支付业务包括付款扫码和收款扫码,付款扫码是指付款人通过移动终端识读收款人展示的条码完成支付的行为,收款扫码是指收款人通过识读付款人移动终端展示的条码完成支付的行为。条码支付具有成本低、通用性强、普及度高的优势,在中国和世界范围内得到了大规模的使用,例如支付宝、微信支付、Venmo 等。

在基于条码的付款方式中,二维码以其数据容量大、具有抗损毁能力等特点,被广泛使用。其中,在基于二维码的支付方式中有一种支付方式称为“付款码小额免密支付”<sup>[6]</sup>。其形式是付款者向收款者出示付款码,收款者使用扫码设备扫描付款码,即可在付款者不用输入密码的情况下从付款者的账户扣除一笔额度范围内的款项。这是一种方便快捷的支付方式,并且得到了广泛的应用,主流支付软件如支付宝、微信、Paypal、TNG Wallet 等都支持此种付款方式。但是,通过调研本文发现这种付款方式存在一个尚无有效解决办法的漏洞亟待解决,称为“隔空盗刷”<sup>[7]</sup>。即犯罪分子利用人们的安全意识不强以及付款码小额免密支付的特点,在消费者提前打开付款码排队等待付款时,趁消费者不备,利用偷拍或偷录的手段,非法获取消费者的付款码,通过使用收银软件应用完成财产盗取的行为。

经过调研,目前主流的解决方案包括如下方式:(1)定时刷新付款码;(2)限制对付款码的截屏;(3)严

格甄选商户的闪付服务资格;(4)交易限额;(5)对每笔交易进行后台风险监测和特征识别等操作;(6)建立“风险全赔付”机制;(7)支付时输入密码。这些方法的使用大大提高了盗刷犯罪的门槛,虽然降低了盗刷犯罪的发生概率,但是,并没有从根本上消除此类犯罪的可能。现存的方法,有一个共同特点,就是从支付行为机制或者策略上来规避可能产生的问题级风险,并没有从本质技术上来解决,特别是存在的“隔空盗刷”问题,是无法防御的。因此,亟需一种方法,从技术层面出发,破坏“隔空盗刷”犯罪实施的所需条件,从而从根本上解决“隔空盗刷”问题。

本文针对目前付款码免密支付中存在的“隔空盗刷”安全隐患,提出了动态二维码水印算法,从技术层面上解决了“隔空盗刷”问题。可以有效防止消费者在使用付款码的过程中被他人盗刷的问题,为消费者的资金提供了安全保障。同时,本文所提方案可以推广应用到各大移动支付平台以及其他“亮码-扫码”的场景,例如电子检票等,可以有效增强各类场景下二维码使用的安全性。

## 2 相关工作

本文所提方案的实施需要对动态二维码技术、二维码水印技术、半鲁棒水印和屏摄鲁棒水印技术进行详细了解,充分利用到水印的半鲁棒性和二维码的动态变化来设计算法。

### 2.1 动态二维码

Qiao 等人<sup>[8]</sup>在传统二维码的基础上在一段时间内多帧二维码图像交替显示,其中每一帧二维码图像数据分别采用不同的密钥动态加密生成,不易伪造不易破解,每一帧二维码都有时效性,这样更能保证数据的安全。但是,此方法无法实现脱机验证。Zhou 等人<sup>[9-10]</sup>研究了支持 SM2、SM3、SM4 加密算法的动态二维码支付系统,可以实现二维码扫描和扫描交易、银联云 QuickPass 交易等,并在交易过程中实时生成动态二维码信息,一单一码。通过动态算法分布,保证了二维码生成的随机性和唯一性,适用于多场景应用事务。这种方法在密钥的分发管理和验证计算上,会产生较大的资源消耗,对于用户数以亿计的系统,此种消耗是难以承受的。Wei 等人<sup>[11]</sup>研发

了动态二维码的生成方法,通过对已编码的信息码字进行冗余编码生成动态二维码图像序列;研发了动态二维码的读取方法,只需读取足够数量的帧即可正确地解码得到所承载的信息;设计了彩色动态二维码防伪系统软件,具有防伪效果好、鉴伪方便等特点。此方法在实体防伪标签上的使用效果得到了验证,但在电子二维码上的效果还有待考究。

## 2.2 二维码水印

Vongpradhip 等人<sup>[12]</sup>提出通过带不可见水印的二维码图像,利用离散余弦变换或 DCT 嵌入不可见水印的二维码,用于组内的信息隐藏。编码过程使用 DCT,允许使用基于块 DCT 的方法将二维码图像分解为不同的频带;中频带系数之间的比较然后将不可见的水印信息嵌入到中频带中。水印提取系统通过逆向嵌入过程从不可见水印中确定二维码图像中的水印。Xun 等人<sup>[13]</sup>提出一种二维码信息加密和数字水印双重防伪方法。首先,使用基于 RSA 的加密算法对授权信息进行加密。然后,根据加密信息生成相应的水印图像。最后,介绍了一种结合离散小波变换(DWT)和奇异值分解(SVD)的反打印嵌入和提取图像水印方法。将提取的水印与解密信息进行对比验证,实现二维码的双重安全。这两种方法在频率域对二维码嵌入水印,取得的跨界质鲁棒性效果较差。Shan 等人<sup>[14]</sup>根据二维码的信息存放位置特点和二维码图像特征,将水印位置和链码相结合,设计了一种新的二维码水印加密和其解密算法。另外可以结合人体的多重生物特征对二维码进行加密,水印位置有限时采用模拟退火算法解决各项特征之间的位置竞争问题。理论和实验表明,该算法具有速度快,准确率高,加密不影响二维码基本信息的识别。但是这种方法中水印的鲁棒性难以调节。Tao 等人<sup>[15]</sup>提出了一种基于 QR 码隐写和多级加密的物流隐私保护系统。在不改变 QR 码扫描结果和功能的条件下,该系统利用 QR 码自身纠错功能实现了隐写加密数据到物流 QR 码上。此方法针对自身隐私保密要求高的企业组织快递物流的隐私保护问题提出了有效的解决方案,但是其加解密方法对于庞大的支付系统而言所需计算资源和通信资源消耗难以承受。

## 2.3 半鲁棒水印

鲁棒是 Robust 的音译,也就是健壮和强壮的意思。半鲁棒水印是指对水印的鲁棒性进行调节,使之在某些情况下比较脆弱,从而限制水印在某些场景下的使用。Fridrich 等人<sup>[16]</sup>使用类似于差分编码的原理,将原始图像的颜色深度减小的循环移位嵌入到原始图像中。随着篡改图像中噪声量的增加,重建图

像的质量逐渐下降。Tashk 等人<sup>[17]</sup>提出了一种改进的具有篡改检测和恢复能力的半鲁棒数字图像水印方法。Qiao 等人<sup>[18]</sup>研究了数字图像水印技术,主要工作有以下几方面:首先,对数字水印技术的原理、框架、特性、分类方法以及水印系统的评价标准做了归纳和阐述,同时详细介绍了鲁棒性水印和半脆弱水印的概念、特性、原理、分类、区别以及两种水印在空域和频域内的各种经典算法。其次,提出了一种改进的 DCT 域鲁棒水印算法。上述方法在自然图像中的效果较好,但对于色彩和纹理都较为简单的二维码图像,是否也能取得良好效果有待考究。

## 2.4 屏摄鲁棒水印

屏摄即“屏显-拍摄”过程,是一种跨媒体传输方式,它对信息的破坏力强,不但给屏摄图像带来严重的视觉失真,也会破坏其中的水印信息。屏摄过程随着屏读时代的到来日渐常见,屏摄鲁棒水印的应用场景也在近几年得到了极大关注。相比于打印拍照而言,屏摄过程的光照和采样失真更为严重。Fang 等人<sup>[19]</sup>总结了屏摄过程中最为特殊的三种失真—镜头失真、光源失真和摩尔纹失真,并结合已有的嵌入模板水印方案,提出了一种基于 SIFT 关键点强度定位和 DCT 系数相对应关系的屏摄鲁棒数字水印算法,该算法有效的实现了屏摄场景下水印的嵌入和提取。Cheng 等人<sup>[20]</sup>提出了制造摩尔纹实现屏幕内容溯源,以达到保护版权的作用。Fang 等人<sup>[21]</sup>提出了将神经网络引入了提取端,提出了一种深度模板水印算法。之后,利用人眼对高频信息的不敏感的性质,在有效保证鲁棒性的前提下,提升了屏摄鲁棒水印的透明性。Li 等人<sup>[22]</sup>提出了一种使用四元数傅里叶变换和张量分解的水印方法进行嵌入,并且总结了屏摄水印的主要的嵌入步骤。

上述工作都对各自领域的进步做出了贡献,但无法直接应用于解决“隔空盗刷”问题。本文所提算法要充分参考上述工作的成果和不足,以实现良好的防盗刷效果。

## 3 动态水印二维码防盗刷方案

通过调研本文发现,对于正常支付场景和盗刷场景,其主要的区别在于信道失真的不同,即两种场景下扫码设备获取的付款码图像通过不同的传输通道,受到了不同的传输损失。对于正常支付而言,付款码和扫码设备之间的距离较近且角度较正。对于盗刷犯罪而言,付款码和扫码设备之间的距离往往较远且很难保持正对的视角。这个特点直接决定了二者会经过不同的信道失真。本文根据这个特点,

设计了一套面向移动支付防盗刷的动态二维码水印算法。

### 3.1 动态二维码水印算法框架

本文提出的面向移动支付防盗刷的动态二维码水印算法包含三个主体, 分别是支付系统后台、付款者和收款者。整体的框架如图 1 所示。其中支付系

统后台负责支付令牌的生成、分发和验证。付款者使用的是付款软件应用, 其工作流程是根据从支付系统后台接收到的支付令牌生成带水印的付款码并以视频帧的形式动态播放。收款者使用的是收款软件应用, 其功能是扫描付款者出示的付款码, 实现收款操作。

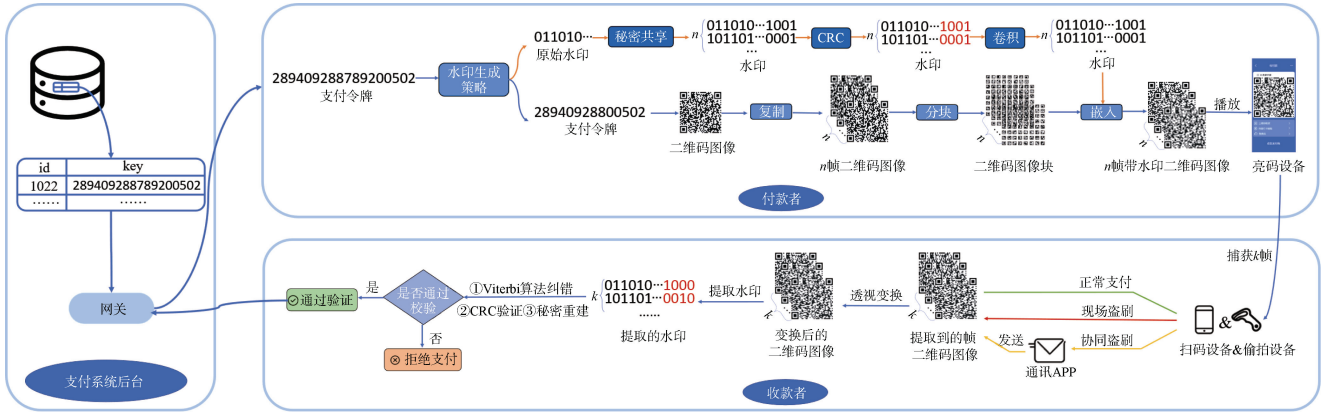


图 1 系统框架图

Figure 1 System Framework

设从服务器中接收到的支付令牌为  $\hat{K}$ , 根据水印生成算法(见 3.3 节), 得到原始水印序列  $w$ , 再通过秘密共享策略(见 3.4 节)得到  $n$  份水印序列。对每一份水印序列, 使用循环冗余校验(Cyclic Redundancy Check, CRC)为水印序列加上校验位, 得到带有校验位的水印序列集合  $W_{cr}$ , 然后再使用卷积码(Convolutional Code)对其进行卷积编码得到最终待嵌入的水印序列集合  $W_{cc}$ 。然后再将二维码图像复制  $n$  份, 其中  $I_k$  表示第  $k$  张二维码图像, 每一张二维码图像都会进行分块, 得到二维码图像块  $I_{kl}$ , 其中  $k$  和  $l$  表示第  $k$  个二维码图像中的第  $l$  个图像块。最后将水印嵌入以预设的嵌入策略(见 3.5 节)嵌入到每一个二维码图像块中, 会得到一张带有半鲁棒水印的二维码图像。将这个过程重复  $n$  次, 会得到  $n$  张带有水印的二维码图像  $I_k^w$ , 通过视频帧的方式将这些图像以一定的帧率显示出来。

带有水印的二维码图像会通过不同的支付信道被捕获, 若收款者成功捕获并提取出指定数量张二维码图像中的水印序列, 即可成功重建出原始水印序列  $w$ , 从而恢复出原始支付令牌, 在支付系统后台通过验证完成支付。接下来详细介绍不同捕获通道的失真。

### 3.2 不同支付信道失真分析

对于任何付款码防盗刷方案而言, 最基本的要

求是不影响用户的正常支付, 在这个基础上, 实现高效的防盗刷效果。为此, 有必要对正常支付场景及盗刷场景进行分析, 找出不同捕获通道的失真差异。

#### 3.2.1 现场合法支付场景

现场支付场景为消费者在付款时通过向收银员展示付款码, 收银员使用扫码设备扫描消费者的付款码进行收款的过程。这个过程中主要历经了屏摄失真<sup>[23]</sup>, 包含透视变换失真、光学失真和摩尔纹失真。接下来详细介绍这几种失真。

##### 1) 透视变换失真

透视变换失真是当图像在屏幕显示时, 相机从不同角度对屏幕上的图像拍摄造成的失真。比如对于图像中的四个点  $P_1(x_1, y_1)$ 、 $P_2(x_2, y_2)$ 、 $P_3(x_3, y_3)$  和  $P_4(x_4, y_4)$  以及目标变换点  $P'_1(x'_1, y'_1)$ 、 $P'_2(x'_2, y'_2)$ 、 $P'_3(x'_3, y'_3)$  和  $P'_4(x'_4, y'_4)$ 。其可以通过下式进行模拟。

$$\begin{cases} x' = \frac{a_1x + b_1y + c_1}{a_0x + b_0y + 1} \\ y' = \frac{a_2x + b_2y + c_2}{a_0x + b_0y + 1} \end{cases} \quad (1)$$

对二维码图像  $I_k$  加上透视变换失真的失真图像  $I_k^{WD}$  可以根据上式得出。

##### 2) 光学失真

在对屏幕的拍摄过程中, 由于环境照明和屏幕照明的影响, 不同的拍摄条件会导致拍摄图像中的照明分布不同。这主要分为两种照明条件: 点光源和

线光源。对于这两种不同的光照条件, 可以利用点扩散分布和线扩散分布来近似。具体而言, 对于点光源失真, 失真矩阵服从如下分布权重:

$$IW_{\text{point}}(i, j) = \frac{\sqrt{(i - p_x)^2 + (j - p_y)^2}}{\max\text{dis}(p_x, p_y)} \cdot (l_{\min} - l_{\max}) + l_{\max} \quad (2)$$

其中,  $(p_x, p_y)$  为模拟的点光源的坐标, 这是直接在图像中随机选择的结果。而  $\max\text{dis}(p_x, p_y)$  表示距离点光源  $(p_x, p_y)$  到图像四个角的最大距离。  $l_{\min}$  和  $l_{\max}$  分别表示最小和最大光源变化率。对于线光源来说, 失真矩阵可以被表示为:

$$IW_{\text{line}} \sim \mathbb{U}\{T^0, T^{90}, T^{180}, T^{270}\} \quad (3)$$

其中  $\mathbb{U}\{\cdot\}$  表示等概率选择,

$$T^0(i, j) = \frac{(i - H) \cdot (l_{\min} - l_{\max})}{H} + l_{\min} \quad (4)$$

$T^{90}, T^{180}, T^{270}$  表示对  $T^0$  旋转不同的角度, 而线光源失真矩阵  $IW_{\text{line}}$  表示从集合  $\{T^0, T^{90}, T^{180}, T^{270}\}$  中随机选择。最后, 光源失真  $D$  可以从  $\{IW_{\text{line}}, IW_{\text{point}}\}$  随机选择。

### 3) 摩尔纹失真

摩尔纹失真  $M$  是屏幕拍摄过程中出现的最特殊的失真。由于屏幕和相机之间的采样频率差异, 可能会在捕获的图像中出现一些不规则纹理, 可以通过下式进行模拟。

$$\begin{cases} Z_1(i, j) = 0.5 + 0.5\cos\left(2\pi\sqrt{(i - z_x)^2 + (j - z_y)^2}\right) \\ Z_2(i, j) = 0.5 + 0.5\cos\left(\cos\left(\frac{\gamma}{\pi}\right) \cdot j + \sin\left(\frac{\gamma}{\pi}\right) \cdot i\right) \\ Z(i, j) = \min(Z_1(i, j), Z_2(i, j)) \\ M(i, j) = (Z(i, j) + 1) / 2 \end{cases} \quad (5)$$

其中  $\gamma \sim \mathbb{U}[0, \pi]$ ,  $(z_x, z_y)$  表示在整个图像中随机采样的一个选定点的坐标。

结合上述的三种失真, 最终带有屏摄失真的二维码图像可以表示为下式:

$$I_k^{\text{sc}} = \delta_1 \cdot D \cdot I_k^{\text{WD}} + \delta_2 \cdot M + G_N \quad (6)$$

其中  $\delta_1$ 、 $\delta_2$  为失真率的参数,  $G_N$  为高斯噪声。为了方便起见, 本文使用  $I_k^{\text{sc}} \triangleq \text{SC}_{\text{sp}}(I_k)$  表示。

## 3.2.2 现场盗刷场景

现场盗刷信道为盗刷者趁消费者不备, 使用收

银软件现场扫描消费者提前出示的付款码完成财产盗取的这一过程。对于现场盗刷失真通道而言, 其主要历经的失真和现场支付信道基本一致。主要的区别在于现场盗刷中, 盗刷者为了隐蔽起见, 亮码设备和扫码设备之间相隔的距离相对较远且呈现较大的夹角。而现场支付场景一般距离较近, 且捕获角度比较正。这主要体现在透视变换失真的参数上, 本文使用  $I_k^{\text{ss}} = \text{SC}_{\text{ss}}(I_k)$  表示这一过程。

### 3.2.3 协同盗刷场景

协同盗刷信道为盗刷者趁消费者不备, 使用偷拍设备偷拍或偷录消费者的付款码, 之后通过通讯软件发送给远方的共谋者, 由远方的共谋者使用收银软件扫描盗拍的付款码从而完成财产盗取的这一过程。协同盗刷信道主要历经了三个失真过程。首先, 盗刷者先通过手机对消费者手机中的付款码进行拍摄或录制, 这一过程中有屏幕拍摄失真。其二盗刷者将捕获的支付码发送给远方的共谋者, 这一过程包含复杂的传输信道失真。其三共谋者在接收到盗刷者发送的支付码后, 通过扫码设备提取消费者中的支付信息, 这一过程包含屏摄失真。可用下式表示协同盗刷的失真信道:

$$I_k^{\text{cs}} = \text{SC}_{\text{sp}}((\text{OSN}(\text{SC}_{\text{ss}}(I_k))) \quad (7)$$

其中  $\text{OSN}(\cdot)$  表示社交网络传输的失真, 由于社交网络本身是一个黑盒的过程, 不同的通信软件厂商采用了不同的传输和处理方式。事实上, 也有一些工作对 OSN 传输图片的失真进行了分析建模<sup>[24]</sup>。但是客观上 OSN 失真相对屏幕拍摄失真比较小, 并且传输的时候可能采用发送文件或者原图的形式。因此为简便起见本文考虑在处理时将其忽略不计。  $I_k^{\text{cs}}$  即为最终共谋者得到的付款码图像。

综上分析, 付款码历经的信道主要分为现场支付信道、现场盗刷信道以及协同盗刷信道。在这三者中, 现场支付信道和现场盗刷信道的差别主要在于不同的拍摄角度以及距离。而协同盗刷信道中包含了现场盗刷信道, 因此只要解决了现场盗刷就可以解决协同盗刷的问题。如果能够找到一种对距离和角度失真脆弱, 但是又对屏摄失真鲁棒的水印, 就可以解决移动支付中存在的盗刷犯罪问题。

## 3.3 水印及二维码生成

水印及付款二维码是根据从支付系统后台接收的支付令牌生成的, 假设从支付系统主机接收的支付令牌为  $K$ , 生成方法为  $G(K)$ , 原始水印序列为  $w$ , 处理后的支付令牌为  $\hat{K}$ 。水印的生成过程和支付令



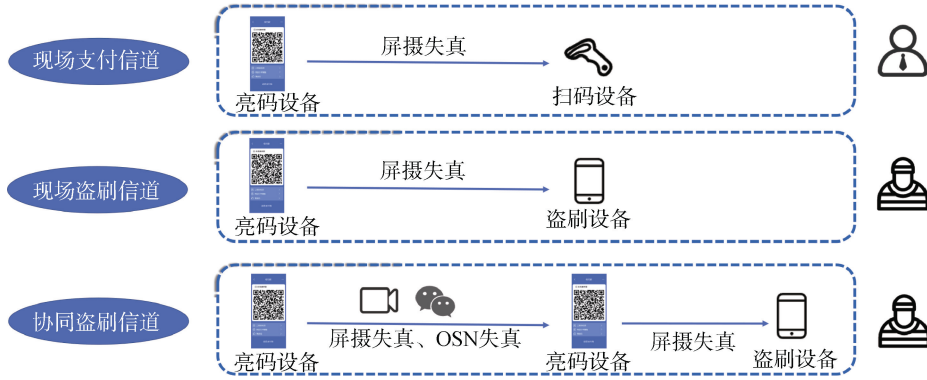


图 2 三种场景的失真信道

Figure 2 Distortion channels for three scenarios

牌的生成过程可以表示为:

$$\{\mathbf{w}, \hat{\mathbf{K}}\} = G(\mathbf{K}) \quad (8)$$

其中,  $G(\mathbf{K})$  为: 根据从支付系统后台接收到的 18 位数字支付令牌  $\mathbf{K}$ , 从中随机选取连续的三位, 将这三数字的值和位置分别编码成 10 比特和 4 比特的二进制数。最后, 采用奇偶校验法, 对末尾添加 1 比特的校验位, 得到总长度为 15 比特的原始水印序列  $\mathbf{w}$  和长度为 15 位数字的支付令牌  $\hat{\mathbf{K}}$ 。根据支付令牌  $\hat{\mathbf{K}}$  生成二维码, 在其中嵌入水印即得到带水印的付款码。

但是, 仅将上述生成的水印序列嵌入到单张二维码中不足以有效防止盗刷。这是因为对于单张的二维码图像, 盗刷者往往可以通过高分辨率的手机进行拍摄, 可能会导致盗刷成功。因此, 还要使水印算法满足单帧二维码图像不可完整获取原始水印序列的条件。为此, 本文使用了秘密共享的策略, 它将原始的水印序列分割成了  $n$  份并嵌入到多张二维码中, 使得单独捕获的一帧二维码图像无法获取原始水印序列。

### 3.4 秘密共享策略

秘密共享策略的基本思想是将秘密以适当的方式拆分, 拆分后的每一个部分由不同的参与者管理, 只有超过指定数目的参与者一同协作才能恢复秘密消息。

本文使用 Shamir 的  $(t, n)$  门限方案<sup>[25]</sup>, 它是一种有效的秘密共享算法, 用于在一个组中分发私密信息, 这样秘密就不会被泄露, 除非收集到的秘密份额数达到指定数量。为实现这一点, 秘密在数学上被分成多个部分, 只有当足够数量的部分组合在一起时才能重建原始秘密。

Shamir 的  $(t, n)$  门限方案利用拉格朗日插值定理,

特别是  $t$  多项式上的点唯一确定次数小于或等于的多项式  $t-1$ 。假设原始秘密(即原始水印序列)为  $\mathbf{w}$ , 参与方个数为  $N$ , 秘密重建阈值为  $t$ , 即不少于  $t$  个参与方能够联合恢复秘密  $\mathbf{w}$ 。本文所提基于秘密共享策略的水印二维码生成流程如图 3 所示。将原始水印序列分成  $n$  份并嵌入到  $n$  张相同的二维码图像中, 成功提取其中任意  $t$  张中的水印信息, 即可恢复出原始水印序列。

秘密分发阶段: 首先, 将原始的二进制水印序列  $\mathbf{w}$  转换为十进制数  $s$ , 令  $a_0 = s$ 。然后, 生成  $t$  个随机数  $\{a_1, \dots, a_t\}$ , 并构造多项式(9)。

$$f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \dots + a_t \cdot x^{t-1} \quad (9)$$

接着, 任取  $n$  个在有限域  $F_p$  中不相同的数  $\{x_1, x_2, \dots, x_n\}$ , 根据随机生成的数在有限域  $F_p$  中计算  $\{f(x_1), f(x_2), \dots, f(x_n)\}$ 。随后, 将这  $n$  对  $(x_i, y_i)$  转换成二进制数, 即  $s_i = B(x_i) + B(y_i)$ , 其中  $B(x)$  表示将十进制转换为二进制,  $x$  表示十进制数。得到的  $\{s_1, \dots, s_n\}$  即为秘密共享后的  $n$  份水印序列, 再进行后续 CRC 和卷积操作, 得到最终  $n$  份待嵌入水印序列, 将它们分别嵌入到  $n$  张二维码图像中即得到  $n$  张带水印的二维码图像。

秘密恢复阶段: 基于拉格朗日公式,  $f(x)$  的表示方式如下:

$$f(x) = \sum_{i=1}^n \left( y_i \prod_{1 \leq j \leq n, j \neq i} \frac{x - x_j}{x_i - x_j} \right) \bmod P \quad (10)$$

假设成功捕获提取出  $t$  张不同的二维码图像中的水印信息。第一步: 将它们分别进行 CRC 和卷积操作的逆过程后, 得到  $t$  份二进制的水印序列。第二步: 将这  $t$  份水印序列分别转换成十进制形式, 可以得到  $t$  对形如  $(x_i, y_i)$  的十进制数对。最后, 将这  $t$  个

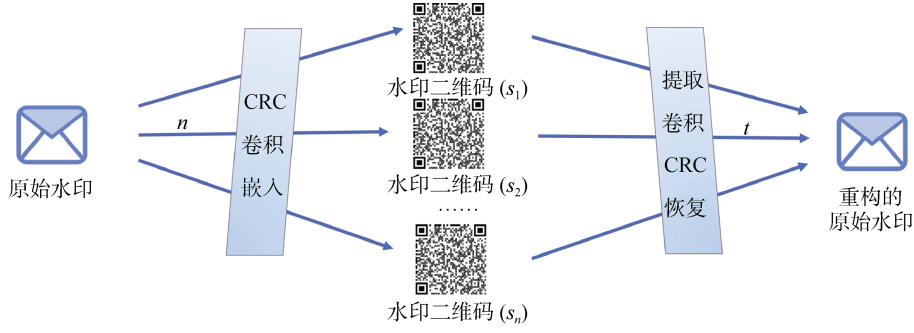


图3 基于秘密共享策略的水印二维码生成

Figure 3 Watermarked QR code generation based on secret sharing strategy

数对代入式(10), 取  $x=0$  时的值, 即为秘密  $s$ , 将其转换成二进制数, 即得原始水印序列  $w$ 。

### 3.5 水印嵌入

二维码水印采取分块嵌入的方法。假设每张二维码图像经过分块得到  $m$  个二维码图像块, 每个图像块按照如下嵌入方式嵌入一个比特。

#### 3.5.1 水印嵌入模型

该水印算法将每一个比特嵌入一个二维码图像块中。假设第  $k$  个二维码中第  $l$  个图像块为  $I_{kl}$ , 且大小为  $T \times T$ 。应用二维离散余弦变换(2D-DCT, 是一种常用的图像和视频处理方法, 可以用于图像增强、图像和视频压缩、特征提取等领域), 可以获得  $T \times T$  DCT 系数矩阵  $D_{I_{kl}}$ 。选择一对系数  $c_w, c_{\bar{w}}$ , 例如  $c_w = D_{I_{kl}}(u, v)$  和  $c_{\bar{w}} = D_{I_{kl}}(v, u)$ , 其中  $u, v > 0$ 。令  $\varepsilon = \max(c_w, c_{\bar{w}})$ ,  $\epsilon = \min(c_w, c_{\bar{w}})$ ,  $w$  和  $\bar{w}$  表示要嵌入的水印位, 其中  $\bar{w} = 1 - w$ 。嵌入过程可以表示为:

$$\begin{cases} \hat{c}_w = \varepsilon + \Delta, \hat{c}_{\bar{w}} = \epsilon - \Delta, & \text{if } w = 1 \\ \hat{c}_w = \epsilon - \Delta, \hat{c}_{\bar{w}} = \varepsilon + \Delta, & \text{if } w = 0 \end{cases} \quad (11)$$

其中  $\hat{c}_w$  和  $\hat{c}_{\bar{w}}$  是嵌入后的 DCT 系数,  $\Delta$  是嵌入强度参数, 旨在扩大  $\hat{c}_w$  和  $\hat{c}_{\bar{w}}$  之间的差异。可以通过逆 DCT 变换获得中间嵌入图像  $\tilde{I}^w$ , 其中  $\tilde{I}^w$  的每个像素是实值。也就是说,  $\tilde{I}_{kl}^w(i, j) \in \mathbb{R}$ , 其中,  $(i, j)$  是第  $i$  行第  $j$  列像素的索引。这种嵌入方案对自然图像有很好的效果, 并且具有很好的屏摄鲁棒性, 这一点已经在多篇工作中得到验证<sup>[23]</sup>。然而, 对于二维码这种纹理比较简单的图像, 这种水印算法无法直接应用。为了解决这个问题, 本文通过数学推导详细的剖析了该水印算法的原理, 提出了一种鲁棒性可控的水印方案。

根据 DCT 变换公式, 可以知道  $c_w$  和  $c_{\bar{w}}$  分别根据如下公式计算:

$$c_w = D_{I_{kl}}(u, v) = \frac{2}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I_{kl}(i, j) \cdot \cos \frac{(2i+1)u\pi}{2N} \cdot \cos \frac{(2j+1)v\pi}{2N} \quad (12)$$

$$c_{\bar{w}} = D_{I_{kl}}(v, u) = \frac{2}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I_{kl}(i, j) \cdot \cos \frac{(2i+1)v\pi}{2N} \cdot \cos \frac{(2j+1)u\pi}{2N} \quad (13)$$

根据式(11)可知, 由于嵌入过程是通过调制图中两个 DCT 系数之间的关系来嵌入一个比特, 那么对于提取过程来说, 就是通过判断两个系数之间的关系来提取一个比特。过程如下: 根据嵌入公式可以得出, 对于提取过程需要知道两个系数之间的差值, 令  $\Delta D_{I_{kl}}^w = c_w - c_{\bar{w}}$ , 那么提取水印的过程可以表示如下:

$$\hat{w} = \begin{cases} 1 & \text{if } \Delta D_{I_{kl}}^w \geq 0 \\ 0 & \text{if } \Delta D_{I_{kl}}^w < 0 \end{cases} \quad (14)$$

#### 算法1 水印嵌入算法

输入: 原始二维码图像  $I$ , 原始水印序列  $w$ , 嵌入数值  $p_w$  和  $p_{\bar{w}}$ , 系数对  $u$  和  $v$

输出: 水印二维码图像集合  $M$

1. 初始化水印二维码图像集合  $M$
2. 通过秘密共享得到嵌入水印序列集合  $W$
3. FOR  $k \leftarrow 0, n-1$  DO
4.   复制二维码图像  $I_k = I$
5.   对水印序列  $W^k$  处理得到  $W_{cc}^k$
6.   FOR  $l \leftarrow 0, m-1$  DO
7.     分块得到二维码图像块  $I_{kl}$
8.     获取水印比特  $w = W_{cc}^{kl}$
9.     计算嵌入集合  $P_{\bar{w}}$  和  $P_w$
10.      $I_{kl}(i, j) = p_{\bar{w}}$  其中  $(i, j) \in P_{\bar{w}}$
11.      $I_{kl}(i, j) = p_w$  其中  $(i, j) \in P_w$

- 
12. END FOR
  13. 将图像  $I_k$  放入到集合  $M$  中
  14. END FOR
  15. RETURN 水印二维码图像集合  $M$
- 

根据 DCT 的公式可以得知,  $\Delta D_{I_{kl}^w}$  的值可以使用公式表示如下:

$$\begin{aligned} \Delta D_{I_{kl}^w} &= D_{I_{kl}^w}(u,v) - D_{I_{kl}^w}(v,u) \\ &= \frac{2}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I_{kl}^w(i,j) \cdot \theta(i,j) \end{aligned} \quad (15)$$

$$\begin{aligned} \theta(i,j) &= \cos \frac{(2i+1)u\pi}{2N} \cdot \cos \frac{(2j+1)v\pi}{2N} \\ &\quad - \cos \frac{(2i+1)v\pi}{2N} \cdot \cos \frac{(2j+1)u\pi}{2N} \end{aligned} \quad (16)$$

从式(15)可知, 由于  $I_{kl}^w \in [0, 255]$ ,  $\Delta D_{I_{kl}^w}$  的值只与  $\theta(i,j) \neq 0$  的所有像素有关, 也就是说, 在提取的时候对于  $\theta(i,j) > 0$  的所有像素, 可以使图像嵌入  $w$  比特, 此区域的集合用  $P_w$  表示。而对于  $\theta(i,j) < 0$  的所有像素, 可以使图像嵌入  $\bar{w}$  比特, 此区域的集合用  $P_{\bar{w}}$  表示。只要保持嵌入像素后满足提取水印的条件即可嵌入一个比特。首先, 在图像的空域确定水印的嵌入位置, 然后再通过在空域控制嵌入像素数值, 以达到嵌入一个比特的目的。

由于所有的水印都是嵌入在黑色像素区域, 因此, 只需要在二维码的黑色区域中修改像素值使其达到提取水印的条件即可嵌入一个比特的水印。从数学上来说, 也就是找到一个这样的嵌入二维码图像块  $I_{kl}^w$ , 使其满足如下式子:

$$\begin{cases} \alpha - \beta \geq \Delta, & \text{if } w = 1 \\ \beta - \alpha > \Delta, & \text{if } w = 0 \end{cases} \quad (17)$$

其中,

$$\alpha \triangleq \sum_{(i,j) \in P_w} I_{kl}^w(i,j) \cdot |\theta(i,j)| \quad (18)$$

$$\beta \triangleq \sum_{(i,j) \in P_{\bar{w}}} I_{kl}^w(i,j) \cdot |\theta(i,j)| \quad (19)$$

为了方便嵌入操作, 不妨将  $I_{kl}^w(i,j) = p_w$ , 其中  $(i,j) \in P_w$ ,  $I_{kl}^w(i,j) = p_{\bar{w}}$ , 其中  $(i,j) \in P_{\bar{w}}$ ,  $p_w$  和  $p_{\bar{w}}$  分别表示不同比特嵌入区域所嵌入的像素值。因此可以通过调节  $p_w$  和  $p_{\bar{w}}$  之间的大小嵌入一个比特信息。具体嵌入公式如下:

$$\gamma \triangleq \sum_{(i,j) \in P_w} p_w \cdot |\theta(i,j)| \quad (20)$$

$$\delta \triangleq \sum_{(i,j) \in P_{\bar{w}}} p_{\bar{w}} \cdot |\theta(i,j)| \quad (21)$$

式(20)和(21)即为水印的嵌入公式。其原理是不同比特在二维码图像块上的区域嵌入不同的数值, 使其满足提取水印的条件(17)。

综上所述, 可以将该水印的嵌入算法总结为: 对于水印序列, 首先对原始水印序列  $w$  通过秘密共享得到嵌入水印序列集合  $W$ , 对每一份水印序列  $w^k$  进行 CRC 校验和卷积编码, 得到最终待嵌入的水印序列集合  $W_{cc}$ 。对于二维码图像, 首先将二维码图像复制  $n$  份, 再对每张二维码图像进行分块, 对找到二维码图像块中的所有水印嵌入像素坐标集合  $P_w$  和  $P_{\bar{w}}$ , 然后按照设置的嵌入参数  $p_w$  和  $p_{\bar{w}}$  将水印嵌入到二维码图像块上。其具体伪代码如算法 1 所示。

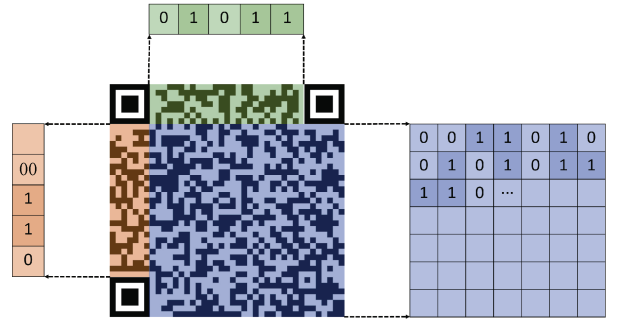


图 4 水印嵌入的分布区域

Figure 4 Embedding regions of watermarks

### 3.5.2 水印嵌入区域

二维码中水印的嵌入区域划分为三个部分, 如图 4 所示。其中橙色和绿色区域由 5 个嵌入块组成,

#### 算法 2 水印提取算法

输入: 捕获的水印二维码图像集合  $C$ , 系数对  $u$  和  $v$

输出: 重建的水印序列  $w$

1. 初始化水印集合  $W$
  2. FOR  $k \leftarrow 0, t-1$  DO
  3. 从集合  $C$  中获取二维码图像  $I_k$
  4. FOR  $l \leftarrow 0, m-1$  DO
  5. 从  $I_k$  中分块得到  $I_{kl}$
  6. 计算  $\Delta D_{I_{kl}^w}$
  7. IF  $\Delta D_{I_{kl}^w} \geq 0$  DO
  8. 将 1 比特放入水印序列  $W_k$  中
  9. ELSE
  10. 将 0 比特放入水印序列  $W_k$  中
  11. END FOR
  12. END FOR
-



13. 利用收集的 $t$ 份水印序列重建水印序列 $w$
14. RETURN 水印序列 $w$

蓝色区域由 $7\times 7$ 的嵌入块组成, 嵌入的信息总容量是 $59bits$ 。

3.6 二维码水印提取

水印提取是通过从循环播放的 $n$ 张带水印的二维码图像中捕获并提取 $t$ 张二维码图像中的水印信息, 从而通过秘密共享策略重建出原始的水印序列。

水印的提取过程主要分为三个步骤。首先, 对捕获的二维码透视变换得到矫正后的二维码图像, 然后再对二维码分块进行提取水印, 得到一段水印序列, 再使用 Viterbi 算法对其进行纠错, 最后再使用 CRC 校验判断水印序列是否发生错误, 如果验证通过, 则暂时保存这份水印序列。将这个过程重复, 直到收集到 $t$ 份互不相同的水印序列, 将这 $t$ 份水印序列通过秘密共享策略重建秘密, 即可得到原始的水印序列。

具体来讲, 可以将水印的提取部分描述如下。设捕获的第 $k$ 张矫正后的二维码图像为 $I_k$ , 而 $I_{kl}$ 表示第 $k$ 张二维码中第 $l$ 个二维码块, 由于水印被分块嵌入到二维码中, 对于当前二维码图像块中所嵌水印信息而言, 所嵌入水印可以通过式(12)、式(13)计算。根据式(14)即可提取到第 $k$ 幅图像的第 $l$ 个二维码图像块的水印信息, 然后对每一个二维码图像块做上述计算, 即可得到所有的水印位, 然后再对提取的水印序列使用 Viterbi 算法纠错, 接着使用 CRC 对水印序列进行校验, 如果校验有误则丢弃本次提取结果, 否则, 通过验证, 加入已提取水印序列集合。最后, 将收集的 $t$ 份水印序列通过秘密共享策略重建秘密, 得到原始水印序列。其具体伪代码如算法 2 所示。

4 实验与分析

4.1 测试环境及设置

实验设置如表 1 所示, 本文考虑到在实际应用场景下, 消费者及盗刷者使用的设备多种多样, 设置了低端设备组和高端设备组。对于亮码端, 使用了两台手机作为实验设备, 其中在低端设备组使用 Xiaomi Mi 13 作为付款码显示设备, 在高端设备组使用 iPhone 15 Pro, 其充当了消费者在支付场景下的支付设备。对于扫码端, 在低端设备组使用 Redmi Note11T Pro 充当了收银员使用的扫码设备, 而在高端设备组使用 Xiaomi 14 Pro。此外, 使用了多种设备作为协同盗刷偷录设备对消费者的付款码进行偷摄, 包括三台手机和一台全画幅数码相机(搭配 SIGMA

28-70mm F2.8 变焦镜头)。二维码图像大小为 $492\times 492$ , 嵌入块的大小为 $60\times 60$ , 二维码帧数为 5, 每一个二维码图像的显示帧率设置为 50 毫秒每张, 系数对的选择为 $u=6$ ,  $v=12$ , 扫码设备捕获的每一帧大小为 $1920\times 1080$ , 每一份水印序列共包含 59 比特。实验场景分为三类, 分别测试该方案在现场支付、现场盗刷和协同盗刷场景下的正常支付效率及防盗刷效率。

表 1 实验参数及设备设置  
Table 1 Experimental parameters and equipment settings

参数名称/设备名称	参数数值/设备型号
支付码显示设备	Xiaomi Mi 13、iPhone 15 Pro
现场盗刷扫码设备	Redmi Note11T Pro、Xiaomi 14 Pro
协同盗刷扫码设备	Redmi Note11T Pro、Xiaomi 14 Pro
偷录协同偷录设备	iPhone XR、iPhone 15 Pro、Xiaomi 14 Pro、SONY ILCE-7M3
偷录协同亮码设备	Xiaomi Mi 13、iPhone 15 Pro
支付码显示帧率 $f$	毫秒/张
捕获预览帧大小	$1920\times 1080$
二维码帧数 $n$	5
秘密共享阈值 $t$	2
$u$	6
$v$	12
$p_w$	100
$p_{\bar{w}}$	50
纠错码	卷积码(2,1,2)
CRC 校验	CRC
嵌入块大小	$60\times 60$
二维码版本	6

4.2 现场支付场景测试

在现场支付场景中, 主要测试本文所提算法在正常支付环境下的支付效率, 模拟测试了消费者在使用本文所提方案时, 付款码中的水印是否会影响正常的支付。这主要分为不同距离、角度的实验。支付的效率主要通过四个角度和一个距离进行实验测试。主要为垂直方向、水平方向、正对角方向和斜对角方向。衡量的指标为提取水印 100 次下的成功次数和平均单次提取时长。

4.2.1 距离实验

距离实验场景如图 5 所示。为不影响用户体验, 本文认为对于单次提取时长大于两秒的, 视为提取失败。在本实验中, 距离从 10 厘米到 50 厘米, 间隔为 5 厘米。



图 5 距离实验场景

Figure 5 Distance experiment scenario

实验结果显示, 在给定的距离范围内, 所有尝试均能完全提取成功, 平均正确提取时间结果图 6 所示。可以看出, 动态水印二维码方案在正常的扫码距离上不会影响支付码的正常支付效率。此外, 在恰当的距离下, 从平均提取的时间来看, 距离越小, 水印能够完全提取的速度就越快。这是可以理解的, 因为在正常的支付场景下, 消费者往往将支付码凑近扫码设备进行支付操作。在距离过近的情况下, 受到摩尔纹的干扰, 水印提取时间会急剧增加, 但仍能在给定的时间范围内完成水印的提取。综上, 本文认为, 所提二维码水印方案不会影响用户的正常支付。

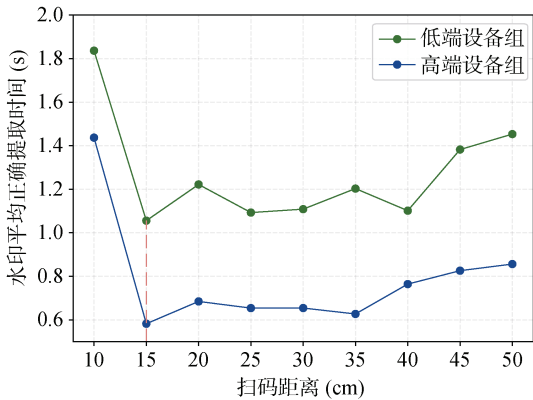


图 6 距离实验结果

Figure 6 Distance experiment results

4.2.2 角度实验

角度实验场景如图 7 所示, 主要测试所提方案在不同角度下水印提取的效率。实验按照设置的四个方向上的角度变化将水印的提取角度范围设置为从 $-45^{\circ}$ 到 $+45^{\circ}$ , 间隔为  $15^{\circ}$ , 扫码的距离, 从相机镜头到二维码中心的距离固定为 30 厘米。

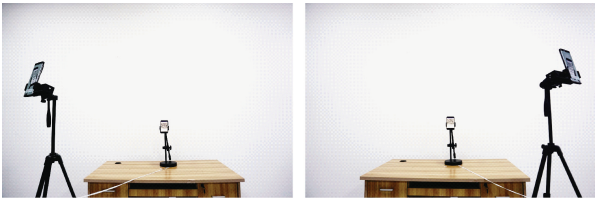


图 7 角度实验场景

Figure 7 Angle experiment scenario

实验结果如表 2、表 3 所示。可以看出, 在角度小于  $15^{\circ}$  时, 所有的支付均能在指定时间内完成, 随着角度的增大, 水印提取的平均时间不断增加。在对角  $45^{\circ}$  时, 已经无法正确提取出水印, 而此时扫码设备与亮码设备的距离仅为 30 厘米, 在实际盗刷场景下盗刷者很难达到这个距离, 这进一步证明了所提防盗刷方案的安全性。此外, 高端设备组的水印正确提取次数及平均提取时间优于低端设备组, 可以看出水印的解码速度依赖于强大的计算速度。

表 2 不同角度完全正确提取水印次数

Table 2 The number of times the watermark is extracted completely correctly from different angles

角度	垂直	水平	正对角	斜对角
	低端设备组/高端设备组			
$+45^{\circ}$	28/56	27/57	0/0	0/0
$+30^{\circ}$	66/78	59/80	48/68	73/78
$+15^{\circ}$	100/100	100/100	100/100	100/100
0	100/100	100/100	100/100	100/100
$-15^{\circ}$	100/100	100/100	100/100	100/100
$-30^{\circ}$	73/86	60/82	48/67	81/72
$-45^{\circ}$	27/62	27/66	0/0	0/0

表 3 不同角度完全正确提取水印平均时间

Table 3 Average time to fully and correctly extract the watermark from different angles

角度	垂直	水平	正对角	斜对角
	低端设备组/高端设备组			
$+45^{\circ}$	7.14/2.65	7.41/2.73	$\infty/\infty$	$\infty/\infty$
$+30^{\circ}$	3.03/0.87	1.97/0.88	4.17/1.82	2.74/2.06
$+15^{\circ}$	1.22/0.74	1.37/0.73	1.47/0.84	1.62/0.86
0	0.88/0.63	0.95/0.63	1.05/0.65	1.04/0.64
$-15^{\circ}$	1.26/0.74	1.22/0.74	1.57/0.91	1.51/0.87
$-30^{\circ}$	2.32/0.90	3.33/0.90	4.17/2.56	2.47/2.34
$-45^{\circ}$	7.41/2.74	7.41/2.79	$\infty/\infty$	$\infty/\infty$

4.3 现场盗刷场景测试

在现场盗刷场景中, 主要测试本文所提算法解决现场盗刷问题的效率, 模拟了盗刷者通过使用收款软件对消费者的付款码进行现场扫描从而完成财产盗取的场景。

现场盗刷实验场景如图 8 所示。本文将实验场景的设置分为从不同角度盗刷消费者的付款码, 角度主要为对角的盗刷方式。为了模拟更真实的盗刷场景, 本实验测试了多组不同角度的盗刷方式, 盗刷距离固定为 50 厘米, 盗刷者可以双指放大屏幕进行精确捕获。由于盗刷者一般是在消费者背后, 因此在盗刷过程中, 不可能正视捕获消费者的付款码。亮

码设备和扫码设备之间一般呈现俯视的视角关系。此外, 一般情况下, 盗刷者为了隐蔽起见, 不可能在消费者近前进行盗刷活动。为此, 本实验需要考虑人眼的视觉范围。一般情况下人眼的视觉范围如图 9 所示, 有效视野大约跨越  $120^\circ$  的弧线<sup>[27]</sup>。

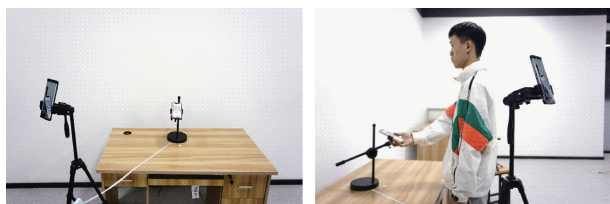


图 8 现场盗刷实验场景

Figure 8 Experimental scenario of on-site theft

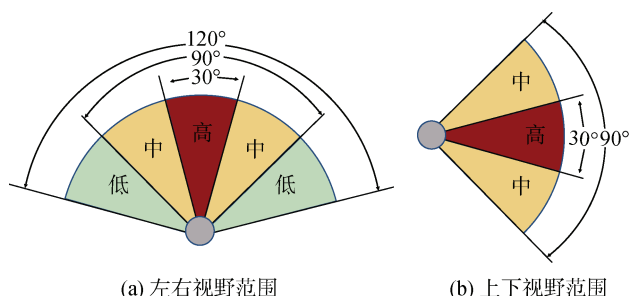


图 9 人眼视觉范围

Figure 9 Human visual range

((a) left and right field of view range; (b) top and bottom field of view range)

结合人眼的视觉范围, 本实验将角度设置为消费者的左(右)后方, 如图 10 所示。具体如下, 假设以消费者身体中心在手机高度作为原点, 消费者的朝向和右方以及上方作为三个坐标轴的正方向。而主要的盗刷的角度范围为以  $x$  和  $y$  轴平面的右侧  $[-45^\circ, 0^\circ]$  和左侧  $[0^\circ, +45^\circ]$ , 以及  $z$  轴和  $x$  轴构成平面的  $[15^\circ, 45^\circ]$ 。

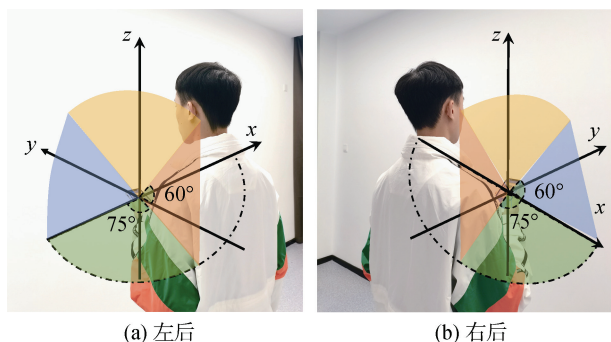


图 10 捕获视角范围

Figure 10 Capture angle range

((a) left rear; (b) right rear)

为了更充分的模拟现场盗刷, 本文假定盗刷者

的作案时间为 1 分钟, 在作案时间内, 盗刷者可以在受害者的背后任意角度移动或缩放盗刷设备, 水印提取成功视为盗刷成功。本文将此过程重复 100 次, 得到最终实验结果。

实验结果显示, 所有现场盗刷尝试全部失败, 都无法成功提取原始水印序列完成验证。也就是说, 本算法中的水印对于现场盗刷场景是脆弱的, 不能抵抗盗刷过程中的失真。这里需要特别注意的是, 本实验环境在完全静止下进行的, 也就是说忽略了消费者在排队过程中发生的抖动。这可以进一步表明, 本文提出的水印算法可以有效防止盗刷行为。

#### 4.4 协同盗刷场景测试

在协同盗刷场景中, 主要测试本文所提算法解决协同盗刷问题的效率, 模拟了盗刷者通过偷录和偷拍的手段对消费者的付款码进行非法获取, 之后发送给远方的共谋者进行扫描从而完成财产盗取的场景。这主要分为偷录协同和偷拍协同的实验。

协同盗刷的流程如图 11 所示, 协同盗刷场景是盗刷者与其共谋者的协同作案。这里需要注意的是, 盗刷者有两种途径对消费者的二维码非法捕获。其一为偷录, 这种盗刷行为考虑了动态二维码的性质, 盗刷者在现场将消费者的二维码偷录下来, 发送给其共谋者。其二为偷拍, 这种场景主要为盗刷者在现场将偷拍的付款码发送给共谋者然后再进行盗刷, 此场景主要考虑盗刷者对解码过程不了解。下面详细对这两种协同盗刷场景进行实验。

##### 4.4.1 偷录协同盗刷

首先, 偷录协同盗刷场景的实验设置应当包含现场盗刷的设置。在实验设置中, 本文将盗刷者的实验角度和距离设置为现场盗刷的角度和距离, 盗刷者和共谋者的通信软件为微信。

实验结果表明, 所有偷录的付款码视频都无法成功提取水印序列通过验证完成支付。这是因为, 二维码历经了两次屏摄失真和一次社交网络的失真。其历经的三个信道的结果如图 12 所示, 可以看出, 屏摄失真对水印信息影响较大。

##### 4.4.2 偷拍协同盗刷

对于偷拍协同盗刷, 本实验设置与偷录盗刷相同, 区别在于该实验盗刷者使用手机对消费者的付款码进行偷拍, 而非偷录。实验结果显示, 不管盗刷者从任意角度偷拍, 都无法成功获取原始水印。根据本水印算法的原理, 在水印处理过程中, 使用了秘密共享的策略, 理论上单独捕获并成功提取出一张二维码中的水印信息, 不可能恢复出原始水印序列, 本实验证明了秘密共享策略对消除偷拍协同盗刷的有效性。



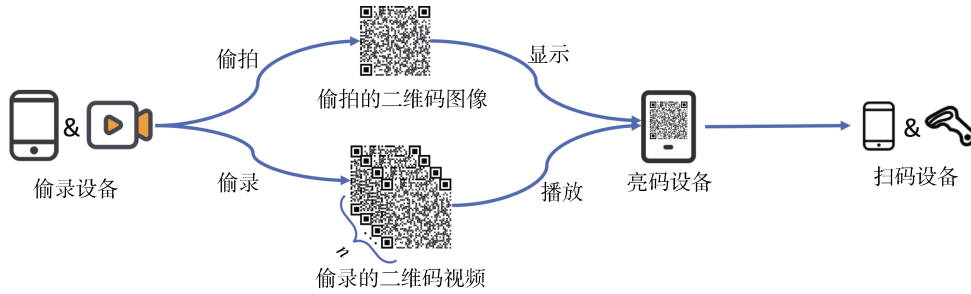


图 11 协同盗刷流程图

Figure 11 Collusive theft process diagram

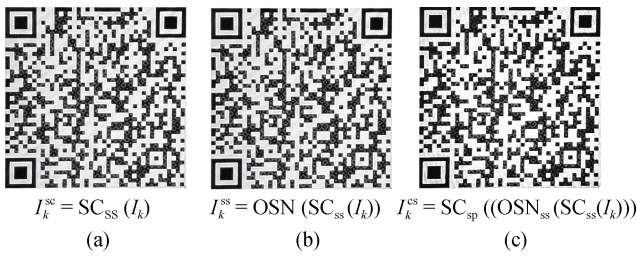


图 12 协同盗刷场景中不同信道失真的二维码图像  
(a)第一次屏摄失真后图像 (b)经过社交网络后的失真图像 (c)第二次屏摄失真后图像

Figure 12 QR code images with different channel distortions in collusive theft scenarios.

((a) QR code image after distortion during the first screen capture; (b) QR code Image after OSN distortion; (c) QR code image after distortion during the second screen capture; )

#### 4.5 水印参数的选取

本小节主要探讨水印算法中的参数的选取。主要分为系数对  $u$ 、 $v$ 、二维码个数  $n$ 、秘密共享阈值  $t$ 、嵌入数值  $p_w$  和  $p_{\bar{w}}$  以及二维码显示帧率  $f$ 。接下来, 详细介绍各个参数选取的实验方案。

##### 4.5.1 系数对 $u$ 、 $v$ 的选取

不同的系数对选取会影响水印的屏摄鲁棒性。一般来说, 在低频嵌入水印一般会比较鲁棒, 而高频嵌入水印会比较脆弱。为了找到介于脆弱和鲁棒之间的半鲁棒水印算法, 本文使用已有结论中<sup>[26]</sup>的屏摄失真模型对不同的系数对的选择进行了模拟, 如图 13 所示。前面提到, 半鲁棒水印算法旨在对角度脆弱, 对屏摄失真鲁棒。因此, 本文对所有的 DCT 系数对进行了实验。

在实验中, 本文将每个系数对都生成相同数量的二维码水印图像进行屏摄失真模拟。在实验中, 倾斜的角度为  $15^\circ \sim 45^\circ$ , 每段原始水印序列被嵌入到 5 张二维码图像中, 并且这所有的 5 张二维码图像中都会使用相同的透视变换角度对其失真变换, 然后

添加不同的光源失真和摩尔纹失真。这是因为不同帧二维码捕获条件可能不相同, 但是角度大致相似。本文将不同系数对的提取错误比特数作为评价指标, 也就是水印提取错误越多表示对角度失真越敏感, 实验结果如图 14 所示。可以看出, 相对于水平变化的频率和垂直变化的频率系数, 对角度带来的失真比较敏感, 而其他区域则比较鲁棒。因此, 为了使水印对角度比较敏感, 本文选择(6,12)作为嵌入系数对。

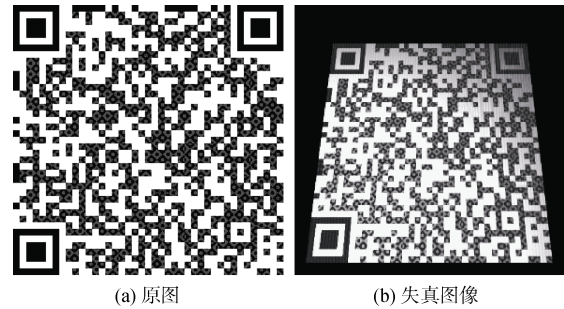


图 13 屏摄失真的模拟

Figure 13 Simulation of screen capture distortion  
(a) original image; (b) distorted images)

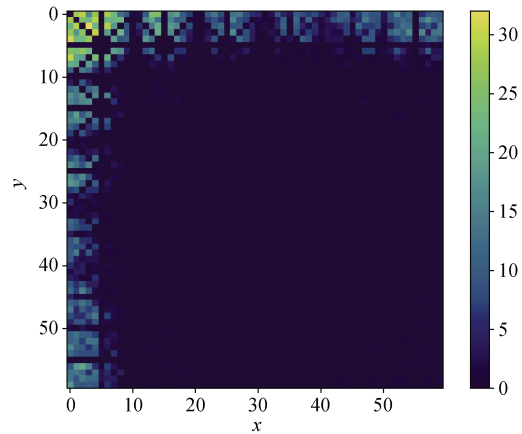


图 14 不同系数下生成的水印二维码在不同角度的屏摄失真下的错误比特数

Figure 14 Error bits of watermarked QR codes generated with different coefficients under screen capture distortion at different angles

### 4.5.2 嵌入数值 $p_w$ 和 $p_{\bar{w}}$

根据嵌入算法可知  $p_w$  和  $p_{\bar{w}}$  主要决定水印的嵌入强度, 当  $p_w$  和  $p_{\bar{w}}$  的差值越大, 说明两个水印比特之间的差别也就越大。当嵌入数值较大时会在二维码的黑色码元中呈现纹理状的条纹, 这些条纹相对来说比较明显。相反, 当嵌入数值较小时, 二维码中水印的不可感知性就越大。如图 15 所示。为了平衡水印的半鲁棒性和不可感知性, 本文选取了  $p_w = 100$  和  $p_{\bar{w}} = 80$  作为实验参数。

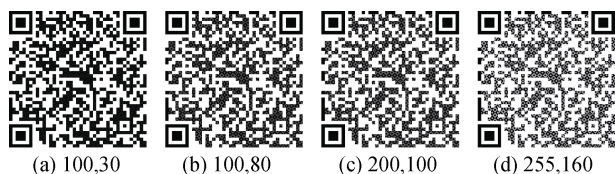


图 15 不同  $p_w, p_{\bar{w}}$  值下水印二维码实例图

Figure 15 Example images of watermarked QR codes with different and values

- (a)  $p_w = 100$ ,  $p_{\bar{w}} = 30$ ; (b)  $p_w = 100$ ,  $p_{\bar{w}} = 80$ ;  
(c)  $p_w = 200$ ,  $p_{\bar{w}} = 100$ ; (d)  $p_w = 255$ ,  $p_{\bar{w}} = 160$

### 4.5.3 显示帧率 $f$

对于显示帧率而言, 过快的帧率会导致捕获设备无法捕获到连续的二维码图像且部分设备无法正常显示, 这就会造成捕获的二维码图像无法正确恢复原始水印, 从而无法通过验证完成支付。在本实验中, 假设二维码分解个数  $n=5$ , 秘密共享阈值  $t=2$ , 亮码设备和扫码设备的距离固定为 30 厘米的情况下, 本文将二维码的显示频率范围从 25 毫秒每张到 80 毫秒每张进行了水印提取实验。本文认为在 2 秒内若能正确提取水印视为提取成功, 每个显示帧率各提取 100 次。将 100 次提取中水印正确提取次数和平均正确提取时间作为评价指标。

实验结果显示所有的显示帧率均能正确提取出水印, 平均正确提取时间结果如图 16 所示。可以看出, 在显示帧率达到 50 毫秒/张时, 随着帧率  $f$  的增加和减少, 水印提取所花费的时间也越来越大。此外, 当捕获相机的预览帧率和显示帧率超过奈奎斯特采样定律时也可能无法提取水印。最终选择最优的显示帧率  $f = 50$  毫秒/张。

### 4.5.4 二维码帧数 $n$ 和秘密共享阈值 $t$

秘密共享策略中,  $n$  和  $t$  的选择会影响水印的提取速率和防盗刷的效果。本文将 100 次实验中正确提取次数和水印的平均正确提取时间作为评价指标, 其中正确提取次数越高、平均正确提取时间越少表

示该  $n$ 、 $t$  系数的水印提取效果越好。受所提方案水印嵌入容量的限制的影响, 本文中的  $t$  值小于等于 3。

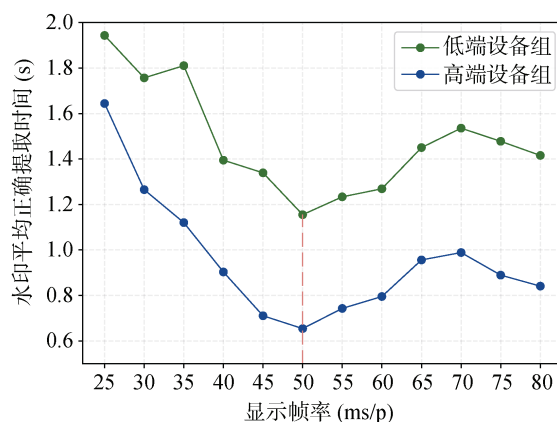


图 16 显示帧率 与水印平均正确提取时间的关系

Figure 16 Relationship between the displayed frame rate and the average correct extraction time of watermark

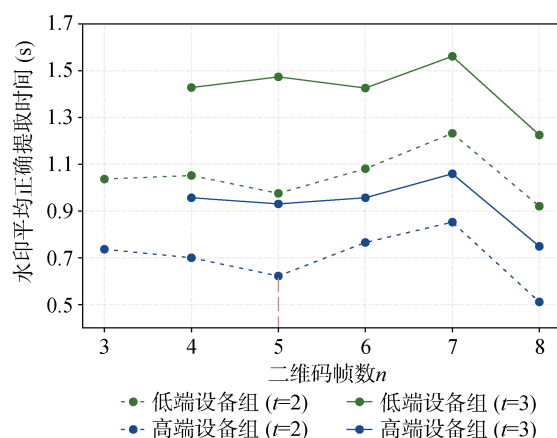


图 17 二维码帧数  $n$  和秘密共享阈值  $t$  的取值与水印平均正确提取时间的关系

Figure 17 Relationship between the number of QR Code frames  $n$  and the value of the secret sharing threshold  $t$  and the average correct extraction time of the watermark.

实验结果如图 17 所示。可以看出,  $t$  的取值增大使得水印提取的平均时间增加, 在  $t$  相同的情况下, 理论上随着  $n$  值的增加水印提取的平均时间会逐渐减小, 在实验中, 其时间呈现波动下降的趋势。理论上随着  $n$ 、 $t$  取值增大且二者数值越接近会取得越好的防盗刷效果, 但也会使得提取时间增加从而降低用户体验。为了平衡在实际使用中不影响用户体验同时取得较好的防盗刷效果, 在本实验条件下, 综合考虑选择二维码帧数  $n=5$ , 秘密共享阈值  $t=2$  作为最终的系数。

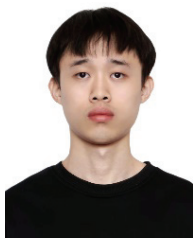


## 5 结论

本文提出了一种动态二维码水印算法来解决付款码免密支付中的盗刷问题。在该水印算法中, 付款方将原本以明码形式展示在付款码中的支付令牌信息进行了信息隐藏。将支付令牌的部分信息以水印的形式通过秘密共享策略分成多份并隐藏在多张相同的二维码图像中。收款方需要在恰当的距离和角度范围内扫描提取到指定数量的水印信息, 进而重建原始水印序列。从而根据二维码扫描结果和水印提取结果, 联合重建支付令牌, 从而完成验证实现付款。实验表明, 本文提出的动态二维码水印算法不会影响付款码的正常支付速率, 且可以有效抵抗付款码免密支付过程中存在的盗刷问题。通过对这一项目的研究, 本文发现还有一些需要解决的问题, 比如水印嵌入容量有待增加的问题。未来工作将继续完善水印算法以及相关软件应用的功能。

## 参考文献

- [1] Fabris N. Cashless Society - the Future of Money or a Utopia?[J]. *Journal of Central Banking Theory and Practice*, 2019, 8(1): 53-66.
- [2] China Academy of Information and Communication Research. White paper on the development of China's digital economy (2021) [R]. 2021, 2-6.
- [3] Pasquet M, Reynaud J, Rosenberger C. Secure Payment with NFC Mobile Phone in the SmartTouch Project[C]. *2008 International Symposium on Collaborative Technologies and Systems*, 2008: 121-126.
- [4] Li Haibo. Research on mobile payment based on NFC technology[J]. *Information Recording Materials*, 2019, 20(03):53-54. DOI: 10.16009/j.cnki.cn13-1295/tq.2019.03.037.
- [5] Yan L Y, Tan G W, Loh X M, et al. QR Code and Mobile Payment: The Disruptive Forces in Retail[J]. *Journal of Retailing and Consumer Services*, 2021, 58: 102300.
- [6] Tang Xizhuo. The current situation and solution of flash payment in the consumer support domain[J]. *Computer and Information Technology*, 2016, 24(2): 48-51.
- [7] Wan Chen. Alert to the "black phone" behind the code payment[J]. *Police station work*, 2018(11): 74.
- [8] Qiao Liangshu. Dynamic QR code generation verification method[J]. *Electronic Technology and Software Engineering*, 2017(24): 163.
- [9] Zhou Y K, Hu B D, Zhang Y T, et al. Implementation of Cryptographic Algorithm in Dynamic QR Code Payment System and Its Performance[J]. *IEEE Access*, 2021, 9: 122362-122372.
- [10] Zhou M X, Ruan S H, Liu J W, et al. VTPM-SM: An Application Scheme of SM2/SM3/SM4 Algorithms Based on Trusted Computing in Cloud Environment[C]. *2022 IEEE 15th International Conference on Cloud Computing*, 2022: 351-356.
- [11] Wei Naixu. Colorful dynamic two-dimensional code anti-counterfeiting label and its anti-counterfeiting system. Zhejiang Province, Cangan Ante Anti-Counterfeiting Science and Technology Co, Ltd, 2016-11-18.
- [12] Vongpradhip S, Rungrangsilp S. QR Code Using Invisible Watermarking in Frequency Domain[C]. *2011 Ninth International Conference on ICT and Knowledge Engineering*, 2012: 47-52.
- [13] Xun Y, Li Z, Zhong X, et al. Dual anti-counterfeiting of QR code based on information encryption and digital watermarking[C]. *Advances in Graphic Communication, Printing and Packaging: Proceedings of 2018 9th China Academic Conference on Printing and Packaging*, 2019: 187-196.
- [14] Shan Lian. Research on QR two-dimensional code watermark encryption and decryption algorithm[J]. *Wireless Internet Technology*, 2013(10): 122-123.
- [15] Tao Jing, Luo Zhenhao, Wang Baosheng et al. A system for logistics privacy protection based on QR code steganography[J]. *Journal of Cyber Security*, 2023, 8(02):1-10.
- [16] Fridrich J, Goljan M. Images with self-correcting capabilities[C]. *Proceedings 1999 International Conference on Image Processing*, 1999, 3: 792-796.
- [17] Tashk A, Danyali H, Ali Alavianmehr M. A Modified Dual Watermarking Scheme for Digital Images with Tamper Localization/Detection and Recovery Capabilities[C]. *2012 9th International ISC Conference on Information Security and Cryptology*, 2012: 60-65.
- [18] Qiaojun. Research on robust and semi-fragile image watermarking algorithm based on DCT[D]. Northwest Normal University, 2014.
- [19] Fang H, Chen D D, Wang F, et al. TERA: Screen-to-Camera Image Code with Transparency, Efficiency, Robustness and Adaptability[J]. *IEEE Transactions on Multimedia*, 2022, 24: 955-967.
- [20] Cheng Y, Ji X, Wang L, et al. {mID}: Tracing screen photos via {Moiré} patterns[C]. *30th USENIX Security Symposium*, 2021: 2969-2986.
- [21] Fang Han. Research on Robust Watermarking Methods for Screen Shot [D]. University of Science and Technology of China, 2021. DOI: 10.27517/d.cnki.gzkju.2021.000591.
- [22] Li L, Bai R, Lu J F, et al. A Watermarking Scheme for Color Image Using Quaternion Discrete Fourier Transform and Tensor Decomposition[J]. *Applied Sciences*, 2021, 11(11): 5006.
- [23] Fang H, Jia Z Y, Ma Z H, et al. PIMoG: An Effective Screen-Shooting Noise-Layer Simulation for Deep-Learning-Based Watermarking Network[C]. *The 30th ACM International Conference on Multimedia*, 2022: 2267-2275.
- [24] Wu H W, Zhou J T, Tian J Y, et al. Robust Image Forgery Detection Against Transmission Over Online Social Networks[J]. *IEEE Transactions on Information Forensics and Security*, 2022, 17: 443-456.
- [25] Shamir A. How to Share a Secret[J]. *Communications of the ACM*, 1979, 22(11): 612-613.
- [26] Hammoud R I. Passive eye monitoring: algorithms, applications and experiments[M]. Springer Science & Business Media, 2008.
- [27] Fang H, Chen D D, Huang Q D, et al. Deep Template-Based Watermarking[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, 31(4): 1436-1451.



**李红棒** 于 2022 年在河南工业大学软件工程专业获得学士学位。现在宁波大学计算机技术专业攻读硕士学位, CCF 会员。研究领域为图像安全。研究兴趣包括: 图像水印、图像哈希。Email: leechb2001@gmail.com



**陈家乐** 于 2024 年在宁波大学计算机科学与技术专业获得硕士学位。现在北京理工大学计算机科学与技术专业攻读博士学位。研究领域为多媒体内容安全。研究兴趣包括: AIGC 安全, 鲁棒图像水印, 计算机视觉。Email: chenoly@foxmail.com



**董理** 于 2018 年在澳门大学计算机专业获得博士学位。现任宁波大学计算机系副教授, CCF 会员。研究领域为多媒体安全。研究兴趣包括: 统计图像建模和处理、多媒体取证和安全以及计算摄影学。Email: dongli@nbu.edu.cn



**王让定** 于 2004 年在同济大学模式识别与智能系统专业获得博士学位。现任宁波大学计算机系教授, CCF 会员。研究领域为信息隐藏/隐写分析/数字取证。研究兴趣包括: 大数据(多媒体)信息隐藏、隐写分析, 数字取证, 数字水印。Email: wangrangding@nbu.edu.cn



**孙巍巍** 于 2018 年在澳门大学计算机专业获得博士学位。现就职于阿里巴巴集团控股有限公司。研究领域为多媒体安全。研究兴趣包括: 统计图像建模和处理、多媒体取证和安全。Email: sunweiwei.sww@alibaba-inc.com



**张玉书** 于 2014 年在重庆大学计算机科学与技术专业获得博士学位。现任南京航空航天大学计算机科学与技术学院教授, CCF 会员。研究领域为多媒体安全与人工智能、区块链及其应用。研究兴趣包括: 多媒体安全与人工智能、区块链与物联网安全、云计算与大数据安全。Email: yushu@nuaa.edu.cn