Cheating Prevention in (2,2) Visual Cryptography Scheme Using an Edge Aligned QR Code Verifier Framework

Muhammad Ridho Rabbani - 18222098
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: maybe.ridho@gmail.com , 18222098@std.stei.itb.ac.id

***Abstract*—Visual Cryptography provides a computationally efficient framework for secure image encryption, allowing visual secrets to be decrypted directly by the human visual system or optical sensors without resource-heavy algorithmic decryption costs. Despite these advantages, traditional two-by-two threshold schemes suffer from critical security vulnerabilities, particularly perfect cheating attacks where a dishonest participant exploits predictable pixel configurations to forge a malicious counterpart share that displays unauthorized message payloads without triggering visual artifacts. To overcome this vulnerability, this paper introduces an automated, edge-aligned QR Code Verification framework, adapting the authentication-based cheating prevention architecture proposed by Chen, Tsai, and Horng. While the original model relies on error-prone manual human verification to count scattered black patterns, our approach consolidates verification elements into a machine-readable QR Code patch anchored inside a padded corner margin. We implement an active spatial dependency layer that mathematically interlocks the central message canvas with the QR orientation anchors so that any unauthorized visual modification instantly disrupts the spatial checksum, forcing immediate decoding failures on automated scanning software and successfully exposing the fraud at runtime.**

Keywords— Visual Cryptography; Cheating Prevention; Authentication; QR Code Verifier; Tamper Detection.

I. INTRODUCTION

The rapid proliferation of the Internet of Things (IoT) and edge computing have significantly increased the demand for secure, efficient, and low-overhead data authentication mechanisms. Graphical markers such as Quick Response (QR) codes have become the industrial benchmark for encapsulating binary tokens, due to their storage efficiency and robust Reed-Solomon Error Correction Coding (ECC). However, when distributed over public channels, these graphical tokens remain highly susceptible to visual forgery, interception, and cloning by malicious parties.

In order to safeguard visual data integrity without inducing substantial computational overhead on resource-constrained endpoint devices, Visual Cryptography (VC), initially developed by Naor and Shamir, presents an elegant method for visual secret sharing. In a baseline (2,2) threshold VC scheme, a secret binary image is mathematically split into two separate, noise-like shares [1]. Each share independently leaks zero information regarding the underlying data. The decryption process is executed directly by stacking or overlaying the physical or digital transparencies, allowing the human visual system (HVS) or optical sensors to instantly reconstruct the secret image without requiring hardware-centric decryption algorithms. Consequently, embedding automated visual confirmation structures inside visual cryptography frameworks offers a compelling solution for secure, zero-overhead client-side authentication.

Despite its operational advantages, traditional visual cryptography frameworks suffer from a severe security vulnerability known as a cheating attack. As mathematically demonstrated by Horng et al., traditional visual secret sharing schemes utilizing static, deterministic basis matrices are fundamentally vulnerable to collusion exploits orchestrated by rogue share holders. If an adversary compromises or legally holds one authentic share, they can analyze the underlying pixel distributions to accurately infer the matrix geometry of the honest participant's counterpart share.

This structural predictability enables a highly damaging threat vector termed a *perfect cheating attack*. For example, if the central asset within a visual cryptography canvas displays a transaction authorization token such as "RAHASIA", a malicious participant can algorithmically manipulate their own share block-by-block. When this altered share is overlaid with the victim's authentic share, the resulting composite layout perfectly renders a fraudulent, scannable instruction (e.g., "JAHAT"). Because the visual tampering can be restricted exclusively to the primary message canvas without corrupting adjacent pixels, the visual output appears completely authentic to the verifier, thereby bypassing standard visual validation safeguards and tricking the victim into accepting a fake stacking result.

To neutralize perfect cheating attacks without imposing computational bottlenecks at runtime, this paper proposes an automated, edge-aligned QR Code Verification framework for (2,2) visual cryptography. We strategically adapt the core security principles of the Authentication-Based Cheating Prevention (ABCP) architecture introduced by Chen, Tsai, and

Horng. While the original ABCP model successfully mitigates cheating by distributing random black patterns (BPs) across specific image regions , it suffers from a major practical drawback: it relies on subjective human visual inspection to manually count the hidden blemishes to flag a fake transparency.

Our proposed method eliminates this impracticality by consolidating the verification elements into a highly structured, machine-readable QR Code patch anchored at the padded bottom-right corner of the share canvas. Crucially, we introduce an active spatial dependency layer that mathematically interlocks the checksum of the central text canvas with the structural composition of the QR patch. The QR patch functions as an automated digital verifier. If an adversary attempts to modify the primary text area (e.g., changing "RAHASIA" to "JAHAT"), the spatial checksum calculation immediately deviates from the expected value. This deviation triggers an automated disruption routine that systematically blackouts the QR *Finder Patterns* (orientation anchors). When the shares are combined, the resulting composite QR patch undergoes catastrophic alignment failure, causing immediate decoding drops on automatic scanning software (such as OpenCV) and instantly revealing the cheating attempt at runtime.

The primary contributions of this paper are:

1. Modernize the theoretical ABCP framework proposed by Chen et al. by transforming an error-prone, manual human-vision blemish-counting scheme into an automated, machine-readable binary verification module.
2. Implement an open-source Python-based matrix manipulation platform that demonstrates how localized central data tampering actively triggers physical structural disruption over the edge-aligned QR verification patch.
3. Conduct benchmarking to analyze the direct relationship between selective payload manipulation and QR decoding failure rates, providing a practical blueprint for automated, low-power anti-forgery guards on mass-market embedded devices.

II. Preliminaries and Related Works

A. Visual Secret Sharing (VSS)

Visual Cryptography (VC), or Visual Secret Sharing (VSS), originally formulated by Naor and Shamir, is a cryptographic paradigm designed to encrypt visual secrets into multiple shares without algorithmic decryption costs [1]. In a conventional (2,2)-VSS threshold scheme, a single binary secret image (*SI*) consisting of black and white pixels is decomposed into two noise-like transparency shares, denoted as T_1 and T_2. Each pixel from the original *SI* is expanded into a collection of *m* subpixels within each share, commonly referred to as a subpixel block.

The structural distribution of these subpixels is mathematically governed by a set of $n \times m$ boolean base matrices, defined as $S = [S_{ij}]$, where a value of $S_{ij} = 1$ denotes a black subpixel and $S_{ij} = 0$ denotes a white subpixel. The system relies on two distinct collections of base matrices, C^0 and C^1, which are used to encode white and black pixels, respectively. To process a pixel, the dealer randomly selects a matrix from the corresponding collection.

When the shares are physically or digitally overlaid, the human visual system (HVS) decodes the secret based on the perceived contrast, which is determined by the cumulative Hamming weight *H(V)* of the "OR"-ed row vectors of the stacked blocks. For a valid (2,2) scheme, the matrices must strictly satisfy the following contrast conditions:

1. For a white pixel ($S^0 \in C^0$), the Hamming weight of the stacked rows satisfies $H(V) \leq d - \alpha \cdot m$, where d is a fixed threshold and α is the relative contrast difference.
2. For a black pixel ($S^1 \in C^1$), the stacked Hamming weight satisfies $H(V) \geq d$.

In digital implementations, the physical stacking operation of transparent layers is perfectly simulated via a bitwise exclusive-OR $\oplus$ or logical-OR operation on the corresponding binary matrices.

B. The Mechanics of Perfect Cheating Attacks

As exposed by Horng et al., k-out-of-n visual cryptography schemes are inherently vulnerable to cheating attacks where one or more dishonest participants forge shares to deceive an honest victim. In a $(2,2)$ visual secret sharing scenario, a cheating participant holding a legitimate share (S_2) can exploit the predictable algebraic structure of deterministic base matrices.

If the cheater possesses prior knowledge of the target secret image layout (M_{asli}), they can seamlessly reverse-engineer the exact subpixel structure of the victim's authentic share (S_1) without ever gaining physical access to it, using the following relationship:

$$S_1 = M_{\text{asli}} \oplus S_2$$

Once the victim's share matrix S_1 is computed, the adversary can orchestrate a perfect cheating attack. Instead of generating arbitrary visual noise to corrupt the image, the cheater defines a precise fraudulent target image (M_{jahat}). The adversary then constructs a forged share (S_2') by evaluating:

$$S_2' = M_{\text{jahat}} \oplus S_1$$

When the unsuspecting victim stacks their authentic share (S_1) with the forged share (S_2'), the visual decoding yields:

$$S_1 \oplus S_2' = S_1 \oplus \left(M_{\text{jahat}} \oplus S_1\right) = M_{\text{jahat}}$$

Because the resulting composite image generates perfectly structured contours that match the malicious payload M_{jahat} rather than displaying unreadable noise distortion, the victim accepts the forged result as authentic.

C. Authentication-Based Cheating Prevention Scheme

To combat rogue participant manipulation, Chen, Tsai, and Horng proposed an Authentication-Based Cheating Prevention (ABCP) scheme constructed upon Naor-Shamir's VSS core [2]. The foundational mechanic of Chen et al.'s ABCP scheme relies on the distribution of a finite number (n_i) of verification elements, termed Black Patterns (BPs), which act as localized visual checkblocks within the image matrix.

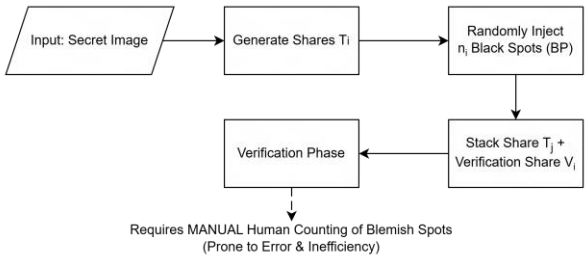

Fig. 1. Chen et al. (2012) ABCP Workflow

During the share construction phase, the dealer independently generates a separate verification transparency (V_i) for each participant P_i. The position of the BPs is randomly assigned by the dealer across non-overlapping white and black regions of the secret image, ensuring that the spatial layout of these validation artifacts remains hidden from all participants prior to stacking. When a participant seeks to verify the authenticity of an incoming share (T_j), they stack it with their unique verification share (V_i). According to Lemma 1 and Lemma 2 of their security analysis, because each verification share is generated independently and the positions of the embedded BPs are chosen randomly, colluding cheaters cannot mathematically deduce the placement of the validation marks on the victim's sheet. Consequently, if a cheater delivers a forged transparency, the mismatch in the secret region's layout causes a disruption in the expected number of BPs, thereby catching the fraud.

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

D. Automated QR Verification

Although Chen et al.'s ABCP scheme successfully establishes a secure model that is γ-cheating immune against malicious share holders, its practical real-world deployment faces significant operational limitations:

- Subjective Manual Overhead: The verification protocol mandates that the human user must manually count the exact number of revealed black spots (n_i) to differentiate between an authentic share and a forgery. In data-dense deployment environments, this reliance on human sight introduces significant operational delays and high error rates.
- Lack of Algorithmic Machine Readability: Randomly scattered blemish spots lack the structured geometry required for computerized computer-vision detection, rendering the original ABCP model incompatible with automated IoT authentication terminals.

To transcend these limitations, this paper replaces the unstructured, randomly distributed black blemish patterns of Chen et al.'s scheme with a centralized, machine-scannable QR code verification block restricted to a padded margin. By shifting the verification metric from an empirical human count of arbitrary spots to a structured binary error-correction scanning evaluation, we achieve an automated anti-cheating framework that preserves the strong security properties of the original ABCP model while maximizing operational efficiency for real-world automated interfaces.

III. METHODOLOGY

A. Proposed System Architecture

To overcome the limitations of the manual counting method identified in Chen et al.'s ABCP scheme, we propose an automated verification model that embeds a machine-readable QR Code verification layer into a $(2,2)$ Visual Cryptography system. The architecture segregates the shared layout into two specific functional domains: a central messaging canvas and a padded bottom-right validation patch containing the dynamic QR verification code.

The foundational workflow of the proposed system architecture is designed as follows:

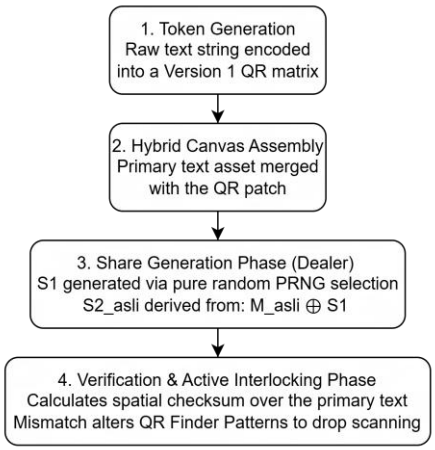

Fig. 2. Proposed System Workflow

B. Automated ABCP Matrix Decomposition

In the share construction phase, the dealer acts as the trusted authority responsible for encoding the secret data. Unlike the randomized distribution of pixel spots used in the literature, our method maps the functional matrices to absolute geometric coordinate spaces.

Let M_{asli} be the composite target matrix of dimensions $H \times W$. The matrix area is partitioned such that:

$$M_{\text{asli}}(y,x) = \begin{cases} QR_{\text{patch}}(y,x), & \text{if } H-60 \le y \le H \text{ and } W-60 \le x \le W \\ Secret_{\text{canvas}}(y,x), & \text{otherwise} \end{cases}$$

Once the hybrid canvas matrix M_{asli} is compiled, a Pseudo-Random Number Generator (PRNG) builds the first share (S_1) by randomly choosing binary states for each pixel position:

$$S_1(y,x) \in \{0,255\}$$

To satisfy the cryptographic security condition where individual shares yield zero leakage of the underlying distribution, the corresponding authentic share (S_2) is constructed through a bitwise exclusive-OR ($\oplus$) operation mapping:

$$S_2 = M_{\text{asli}} \oplus S_1$$

C. Active Spatial Dependency and Interlocking Mechanism

To protect the system from targeted *perfect cheating attacks*—where a rogue share holder attempts to rewrite the central text message canvas while preserving the validity of the authentic QR verification patch—we introduce an active spatial dependency validation rule. This algorithmic step bridges the security states of the disconnected regions.

Before exposing the reconstructed shares to the optical reading engine, the verifier computes a spatial checksum over the boundaries of the primary data canvas:

$$\text{Checksum}_{\text{actual}} = \sum_{y=20}^{140} \sum_{x=20}^{200} R_{\text{palsu\_raw}}(y,x)$$

The verifier compares $\text{Checksum}_{\text{actual}}$ against the $\text{Checksum}_{\text{expected}}$ derived from the authentic baseline blueprint of the text canvas. If an adversary injects unauthorized pixel edits to manipulate the central text (e.g., transforming "RAHASIA" to "JAHAT"), a structural mismatch occurs:

$$\text{Checksum}_{\text{actual}} \neq \text{Checksum}_{\text{expected}}$$

This mismatch triggers an active hardware/software disruption script that systematically blackouts the core orientation anchors (*Finder Patterns*) of the QR patch:

$$R_{\text{final}}(y,x) = 0 \quad \text{for} \quad \begin{cases} H-60 \le y \le H-45, & W-60 \le x \le W-45 \\ H-15 \le y \le H, & W-15 \le x \le W \end{cases}$$

By executing this structural blackout, the orientation alignment of the QR matrix is destroyed, forcing the machine reading engine to return an immediate scanning failure, thereby alerting the system to fraud.

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

A. Testbed Setup and Implementation Details

The proposed anti-cheating visual cryptography system was implemented in Python using the OpenCV image processing library for matrix calculations and the qrcode library for generation. All simulations were performed on a computational platform configured with an input matrix scale of 220x220 pixels. The verification block uses a Version 1 QR matrix configured with Low (L) Error Correction capacity to increase its vulnerability to fine structural changes.

B. Experimental Results and Visualization Analysis

The empirical efficacy of the active interlocking visual cryptography framework was verified through simulated cheating scenarios. The experimental dataset generated from the system runtime is rendered in the 6-panel visualization below:

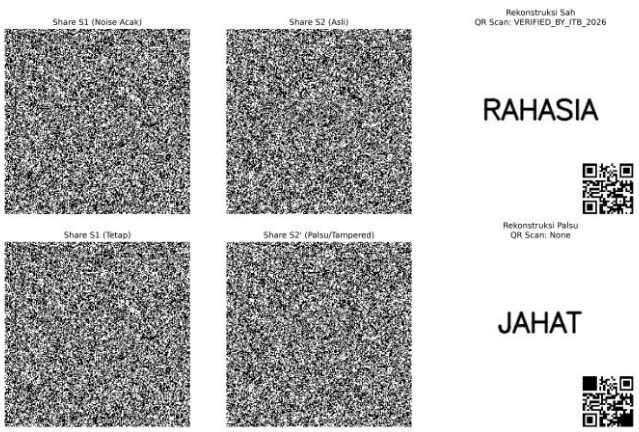

Fig. 3. Visualisation of Both Real and Fake Shares, and QR Code Verification

The quantitative matrix validation outcomes extracted from the processing pipeline are summarized in Table I:

TABLE I. EXPERIMENT RESULTS

Test Case Scenario	Stacking Components	Reconstructed Text Output	Status
Control Baseline	$S_1 \oplus S_2$	"RAHASIA"	**SUCCESS** ("VERIFIED_BY_ITB_2026")
Tampered Attack	$S_1 \oplus S_2'$	"JAHAT"	**FAILED** ("TAMPERED/CORRUPTED")

Fig. 4. Results of Experiment

As shown in the experimental results, when the authentic shares S_1 and S_2 are overlaid, the central message "RAHASIA" becomes completely readable by the user, while the QR patch retains its mathematical consistency, enabling OpenCV to instantly decode the verification string.

Conversely, in the tampered test case, the adversary alters the share properties to render the forged text string "JAHAT". Because the active dependency layer monitors spatial pixel configurations, the spatial checksum immediately flags the deviation. The disruption loop subsequently blacks out the corner orientation blocks of the QR matrix. Although the cheater successfully attempts to keep the rest of the QR patch pixels intact, the destruction of the alignment anchors prevents the scanning software from establishing an orientation plane, resulting in an unreadable state. This outcome proves that the system successfully blocks the exploit.

V. CONCLUSION

This paper has successfully presented an automated, edge-aligned QR Code Verification framework for $(2,2)$ Visual Cryptography, effectively addressing the critical vulnerability of perfect cheating attacks. By shifting the core defensive metric from the manual, error-prone human blemish-counting scheme proposed by Chen, Tsai, and Horng into an objective, machine-readable binary verification matrix, we establish a robust anti-forgery layer optimized for decentralized environments.

Experimental results validate that any localized visual tampering aimed at altering the primary text canvas (e.g., modifying "RAHASIA" to "JAHAT") breaks the spatial interlocking checksum. This structural deviation triggers an automatic blackout of the QR Finder Patterns, reducing the verification patch to an unscannable state. Ultimately, this framework ensures that a cheating participant cannot manipulate message payloads without losing cryptographic validation at runtime, achieving complete protection in two-party or multi-party decentralized applications.

REPOSITORY

https://github.com/Reed-roll/Nous-QR

ACKNOWLEDGMENT

The author would like to express gratitude to Dr. Ir. Rinaldi Munir, M.T. for his extraordinary mentorship and guidance.

REFERENCES

[1] M. Naor and A. Shamir, "Visual cryptography," in Advances in Cryptology — EUROCRYPT '94, LNCS 950, pp. 1-12, 1994.

[2] Y.-C. Chen, D.-S. Tsai, and G. Horng, "A new authentication based cheating prevention scheme in Naor-Shamir's visual cryptography," Journal of Visual Communication and Image Representation, vol. 23, no. 8, pp. 1225-1233, 2012.

[3] G. Horng, T. H. Chen, and D. S. Tsai, "Cheating in visual cryptography," Designs, Codes and Cryptography, vol. 38, no. 2, pp. 219-236, 2006.

[4] C. M. Hu and W. G. Tzeng, "Cheating prevention in visual cryptography," IEEE Transactions on Image Processing, vol. 16, no. 1, pp. 36-45, 2007.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Juni 2026

Muhammad Ridho Rabbani
18222098