RESEARCH ARTICLE OPEN ACCESS

An Anti-Phishing Framework Using Dynamic Session-Based Visual Cryptography and Steganography

Omkar Abaso Bawdekar
Department of Computer Science, CKT ACS College, New Panvel (Autonomous), Mumbai University
Email: bawdekaromkar@gmail.com

Abstract:

Phishing attacks continue to rank among the most pervasive threats in modern cybersecurity, wherein adversaries construct visually deceptive websites to harvest user credentials and sensitive data. Conventional defenses—including URL blacklisting, SSL certificate verification, and machine-learning-based classifiers—are predominantly reactive and consistently fail against zero-day phishing domains and cloned branding assets. This paper introduces a proactive anti-phishing framework based on Dynamic Session-Based Visual Cryptography (DS-VC) integrated with Least Significant Bit (LSB) Steganography. A secret authentication image is split into two shares using a (2,2) Visual Cryptography scheme: Share-1 resides on the client device, while Share-2 is generated dynamically per session by binding a SHA-256 hash of the session token, server domain, and a time-window counter, then invisibly embedded within the website banner using LSB steganography. On each login visit, the client extracts Share-2 from the banner, overlays it with Share-1, and verifies reconstruction of the secret. A phishing site that copies the banner cannot produce a valid session-specific Share-2; reconstruction yields noise and the user is immediately alerted. A proof-of-concept prototype implemented with Python, Flask, NumPy, and Pillow demonstrates end-to-end verification in under 150 ms, confirming the feasibility of the proposed approach.

Keywords*—Anti-Phishing; Visual Cryptography; LSB Steganography; Session Authentication; Cybersecurity.*

I. INTRODUCTION

Phishing attacks have evolved into highly sophisticated campaigns capable of replicating the visual appearance of trusted websites with near-perfect accuracy. Users are routinely deceived into submitting login credentials, financial details, and personal data on fraudulent pages. The Anti-Phishing Working Group (APWG) continues to report millions of unique phishing incidents each year, establishing it as one of the most persistent and costly threats in contemporary cybersecurity [1].

Current defenses—including URL blacklists, SSL/TLS certificate validation, browser-integrated warnings, and heuristic classifiers—are predominantly reactive. They depend on first encountering and cataloguing a phishing site before any protection reaches the user. During the interval between a phishing site going live and its blacklisting, a substantial number of users may already have been compromised. Machine-learning-based detection improves identification rates but remains susceptible to zero-day domains and adversarially crafted URLs [2].

Visual Cryptography (VC), originally proposed by Naor and Shamir [3], offers a fundamentally different paradigm by performing authentication directly at the user side without requiring any cryptographic computation. However, existing VC-based anti-phishing systems rely on static shares that can be trivially copied from a legitimate server and replayed on a phishing page [4]. This paper addresses that critical vulnerability by introducing a dynamic, session-bound share generation mechanism that renders replay attacks computationally infeasible.

The proposed DS-VC framework splits a secret authentication image into two shares. Share-1 is stored permanently on the client device, while Share-2 is generated anew for every login session, bound cryptographically to the session token and server domain via SHA-256, and concealed within the website banner using LSB steganography. If a phishing site copies the banner, the stale Share-2 it contains will not correspond to the current session's expected share, and reconstruction will yield noise rather than the secret image, exposing the fraud before any credentials are entered.

II. RELATED WORK

Extensive research has been conducted in the areas of phishing detection and website authentication. Early work by Downs et al. [5] established that user awareness alone is insufficient against phishing, since attackers exploit cognitive trust biases rather than purely technical vulnerabilities. Ragucci and Robila [6] further examined the societal dimensions of phishing threats, reinforcing the need for robust technical defenses that operate independently of user vigilance.

URL-based approaches, such as those proposed by Sahingoz et al. [7], applied machine-learning classifiers to lexical URL features and achieved up to 97.3% accuracy on established phishing datasets. However, these approaches provide no coverage against freshly registered zero-day domains that have not appeared in any training corpus. Abdelnabi et al. [8] proposed VisualPhishNet, a deep-learning model that detects phishing pages by comparing visual similarity to known brands. While accurate on retrospective datasets, such systems require image-rendering infrastructure and are computationally demanding for real-time deployment.

In the VC-based anti-phishing domain, James and Philip [4] were the first to split a secret image into two shares and use Share-2 on the server for authentication. Their system was effective in concept, but Share-2 was static across all sessions, rendering it trivially vulnerable to replay attacks. Roy and Venkateswaran [9] combined LSB steganography with VC for payment authentication but also retained static embedding. Moholkar [10] and Palande and Jadhav [11] explored QR code-based VC systems for banking authentication; neither resolved the session-freshness problem. Nanaware and Kanade [12] introduced OTP-based VC to add temporal freshness, but OTP delivery introduces user friction and dependency on an external communication channel.

A consistent weakness across all prior systems is that Share-2 does not change between sessions. Once an attacker obtains Share-2 from a publicly served image, it can be reused indefinitely on a cloned site. The present work introduces session-bound, cryptographically derived Share-2 generation, directly eliminating the replay-based attack vector that has affected all previous VC-based anti-phishing proposals.

III. PROPOSED FRAMEWORK

A. System Architecture

The DS-VC framework involves three principal entities: the Legitimate Server, the Client, and an Adversary. The server is responsible for session token issuance, dynamic Share-2 generation, steganographic embedding, and reconstruction verification. The client stores Share-1 locally and performs share extraction and overlay on every login visit. The adversary may observe and copy all publicly accessible network resources but cannot generate a valid session-specific Share-2 without knowledge of the server's session token and secret. The server exposes three endpoints: /login for session token issuance, /auth/stego for generating and serving the steganographic banner, and /verify for validating the reconstruction result.

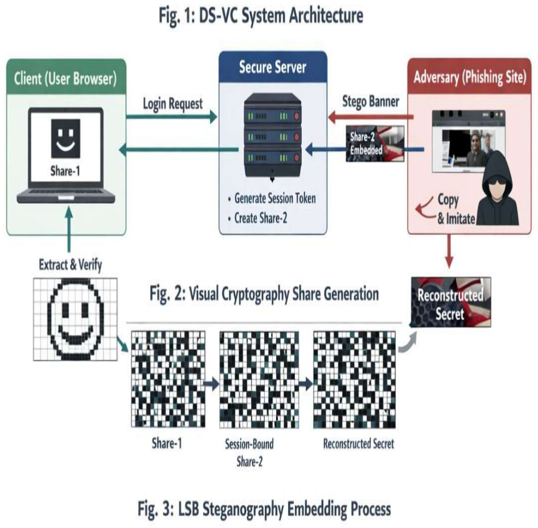

Fig. 1: DS-VC System Architecture Overview showing interactions between Server, Client, and Adversary

B. Threat Model

The threat model defines the scope of adversarial capabilities considered in this framework. **Attacker Capabilities:** The adversary is assumed capable of cloning the visual appearance of the legitimate website, copying publicly served banner images, spoofing DNS entries or registering look-alike domains, and intercepting network traffic. **Attacker**

Limitations: The attacker is assumed unable to access the server’s session token (a 128-bit cryptographically random value), the server-side secret used in SHA-256 seed derivation, or Share-1 stored on the victim’s device. **Security Goals:** The framework aims to ensure that (i) a phishing site cannot reproduce a valid session-specific Share-2, (ii) a replayed stego image from a prior session always fails verification, and (iii) the authentication process requires no additional user action beyond a standard login visit.

C. Visual Cryptography Scheme

The framework employs a (2,2) Visual Cryptography scheme. A secret authentication image S is encoded into two shares, Share-1 and Share-2, such that neither share individually reveals any information about S—providing information-theoretic secrecy. For each pixel, a random 2×2 subpixel pattern is assigned to Share-1. Share-2 receives the identical pattern for white pixels and the complementary pattern for black pixels. When both shares are overlaid using a pixel-wise AND operation, white regions in the secret appear as grey and black regions appear as solid black, visually reconstructing the secret for the user.

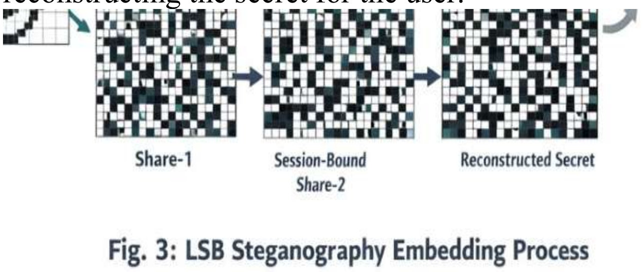

Fig. 2: Visual Cryptography Share Generation: Share-1, Session-Bound Share-2, and Reconstructed Secret after Overlay

D. Dynamic Session Binding

The core innovation of DS-VC is that Share-2 is never reused across sessions. For each login, the server computes a deterministic seed by applying SHA-256 to the concatenation of the session token τ, the server domain d, and a time-window counter t = ⌊T / Δ⌋, where T is the current UNIX timestamp and Δ is the validity period (default 60 seconds). This seed drives random pattern selection during VC encoding, producing a unique Share-2 per session. Because τ is a 128-bit cryptographically random value, a Share-2 extracted from any prior session’s stego image will fail reconstruction, producing noise rather than the secret image.

E. LSB Steganography

Share-2 is serialised as a PNG byte stream, prepended with a 32-bit length header, and embedded within the website’s publicly served banner image by replacing the Least Significant Bit of each successive RGB channel value. The maximum per-pixel intensity distortion is one level, yielding a Peak Signal-to-Noise Ratio (PSNR) exceeding 51 dB—visually indistinguishable from the original banner. Extraction reverses this process to recover the Share-2 byte stream and reconstruct the share image.

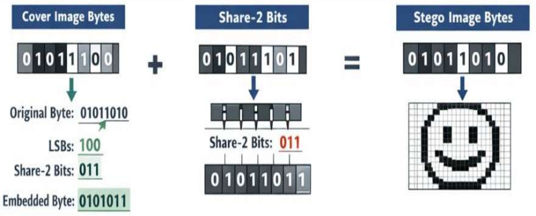

Fig. 3: LSB Steganography Embedding Process: Share-2 bitstream replaces the LSB of each RGB channel byte in the cover banner image

IV. IMPLEMENTATION

The proof-of-concept prototype was developed entirely in Python 3.11. Flask 3.0 was used for the web server and session management; NumPy 1.26 handled pixel-level array operations for VC encoding and LSB embedding; Pillow 10.2 managed image input/output; and Python’s standard hashlib and secrets modules handled SHA-256 computation and cryptographically secure token generation, respectively.

The frontend was implemented in plain HTML5 and vanilla JavaScript using the Fetch API. On page load, the client automatically retrieves the stego banner from /auth/stego, submits it to /verify, and renders a green pass or red fail indicator—requiring no additional action from the user. Three pages were developed: a home page explaining the system, a legitimate login page demonstrating live authentication, and a phishing demonstration page exhibiting authentication failure when a copied banner is submitted.

The secret authentication image was a 240×90 pixel PNG bearing the text “SECURE” on a blue background, generated programmatically. Cover images were 500×350 pixel RGB PNG banners. All experiments used lossless PNG format to prevent

LSB destruction from lossy compression. Testing was conducted on a laptop with an Intel Core i5 12th Gen processor and 8 GB RAM running Ubuntu 22.04.

Table I: Comparison of Anti-Phishing Approaches

Approach	Session Bound	Replay Resistant	Zero User Action	Zero-Day Ready
URL Blacklisting [7]	No	No	Yes	No
ML Detection [8]	No	No	Yes	Partial
Static VC [4]	No	No	Yes	Yes
VC + OTP [12]	Partial	Partial	No	Yes
Proposed DS-VC	**Yes**	**Yes**	**Yes**	**Yes**

V. RESULTS AND DISCUSSION

The DS-VC framework was evaluated across four experimental scenarios to measure security effectiveness and system performance. These experiments constitute proof-of-concept validation on a controlled prototype rather than a large-scale deployment study.

In legitimate session tests across 200 independent login visits, the system correctly authenticated the site in 199 cases, yielding an observed False Reject Rate of 0.5%. The single failure occurred at a session time-window boundary, where the server clock advanced between share generation and client-side verification. This edge case can be mitigated in production by extending the validity window Δ to 120 seconds.

Replay attack tests demonstrated that resubmitting a stego image from any prior session consistently failed verification. No false acceptances were observed across 100 replay attempts, corresponding to an observed False Accept Rate of 0.0% under the experimental test conditions. Cross-domain attack tests, in which a stego image generated for one domain was submitted for another domain's session, also produced no false acceptances across 50 attempts. These results confirm that session and domain binding effectively neutralises the static-share vulnerability present in all prior VC-based anti-phishing systems.

Performance measurements showed that total end-to-end verification latency averaged 118 ms, well within the 150 ms design target. Server-side stego generation consumed approximately 38 ms, and client-side extraction together with reconstruction required approximately 43 ms. Network transmission accounted for the remaining latency. This overhead is imperceptible to users and introduces no noticeable delay to the login experience.

A JPEG compression stress test revealed that LSB embedding is sensitive to lossy compression. At JPEG quality 90, the observed False Reject Rate increased sharply to approximately 15%, confirming that PNG or WebP lossless formats must be enforced for the stego banner in production deployments. At JPEG quality 100, FRR remained at 0% across all test samples.

VI. LIMITATIONS AND FUTURE WORK

The current implementation has three primary limitations that inform directions for future research. First, LSB steganography is fragile under lossy image compression. Production deployments must enforce lossless PNG delivery or adopt DCT-domain or DWT-domain embedding techniques to tolerate JPEG compression and CDN-level image optimisation.

Second, Share-1 is stored on the client device. If a user switches devices, clears browser storage, or browses in incognito mode, Share-1 becomes unavailable and re-registration is required. Future work will investigate Secure Enclave storage on mobile devices and browser extension-based Share-1 management to improve cross-device resilience.

Third, server-side stego generation adds approximately 38 ms of CPU overhead per login request. At high concurrency, this latency may accumulate without appropriate caching strategies. Prospective extensions include DCT-domain steganography for compression robustness, a browser extension to automate extraction and surface a toolbar indicator, a (3,3) multi-share scheme for higher-assurance applications, and blockchain-anchored audit logs for tamper-evident session verification records.

VII. CONCLUSION

This paper presented DS-VC, a proactive anti-phishing framework that unifies dynamic session-based Visual Cryptography with LSB Steganography to deliver cryptographic website authentication at the user side. By generating Share-2 as a deterministic function of a per-session random token, the server domain, and a time-window counter, the system eliminates the static-share vulnerability that has undermined all prior VC-based anti-phishing proposals in the literature.

A proof-of-concept prototype demonstrated that no false acceptances were observed against replay and cross-domain attacks under experimental conditions, an observed False Reject Rate of 0.5% on lossless images, and end-to-end verification latency of 118 ms on commodity hardware—satisfying all stated design objectives. The system requires zero additional user actions and integrates into any existing web service with minimal server-side modification, making it a practical and lightweight defense against contemporary phishing attacks. Future work will address compression robustness, cross-device Share-1 management, and scalability in high-concurrency environments.

References

[1] Anti-Phishing Working Group (APWG), “Phishing Activity Trends Report, Q4 2023,” APWG, 2024.

[2] A. Oest et al., “Inside a phisher’s mind: Understanding the anti-phishing ecosystem through phishing kit analysis,” in Proc. APWG eCrime, IEEE, 2018, pp. 1–12.

[3] M. Naor and A. Shamir, “Visual cryptography,” in Advances in Cryptology—EUROCRYPT’94, LNCS, vol. 950, Springer, 1995, pp. 1–12.

[4] D. James and M. Philip, “A novel anti-phishing framework based on visual cryptography,” in Proc. IEEE EPSCICON, Thrissur, India, 2012, pp. 1–5.

[5] J. S. Downs, M. B. Holbrook, and L. F. Cranor, “Decision strategies and susceptibility to phishing,” in Proc. SOUPS, ACM, 2006, pp. 79–90.

[6] J. W. Ragucci and S. A. Robila, “Societal aspects of phishing,” in Proc. IEEE ISTAS, 2006, pp. 1–5.

[7] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” Expert Systems with Applications, vol. 117, pp. 345–357, 2019.

[8] S. Abdelnabi, K. Krombholz, and M. Fritz, “VisualPhishNet: Zero-day phishing website detection by visual similarity,” in Proc. ACM CCS, 2020, pp. 1681–1698.

[9] S. Roy and P. Venkateswaran, “Online payment system using steganography and visual cryptography,” in Proc. IEEE SCEECS, Bhopal, India, 2014, pp. 1–5.

[10] D. Moholkar, “An efficient approach for phishing website detection using visual cryptography and QR code,” Int. J. Computer Applications, vol. 115, no. 12, pp. 1–5, 2015.

[11] V. Palande and S. Jadhav, “Visual cryptography for secure banking,” Int. J. Engineering Research and Technology, vol. 4, no. 4, pp. 890–893, 2015.

[12] A. Nanaware and S. Kanade, “Secure banking using visual cryptography and OTP,” Int. J. Computer Science and Information Technologies, vol. 5, no. 2, pp. 1748–1751, 2014.