

SecureImgStego: A Keyed Shuffling-based Deep Learning Model for Secure Image Steganography

Trishna Chakraborty^{1*}, Imranur Rahman^{2*}, Hasan Murad^{3*},
Md Shohrab Hossain⁴, and Shagufta Mehnaz⁵

¹University of California, Irvine, ²North Carolina State University,
³Chittagong University of Engineering and Technology,
⁴Bangladesh University of Engineering and Technology,
⁵The Pennsylvania State University



* Work done while at Bangladesh University of Engineering and Technology

The New York Times

Hackers Breach U.S. Marshals System With Sensitive Personal Data

The compromised computer system includes information on both investigative targets and agency employees.

B B C

NEWS

**Serious data breaches across NI
government departments**

Sensitive information being left behind in a restaurant and the possible disclosure of a person's former identity are among serious government data

**The
Guardian**

**Tasmanian data breach:
schoolchildren's information among
16,000 documents leaked on dark web**

**Minister confirms education department documents breached
after third-party file transfer service was hacked**



INDEPENDENT

**Data breach potentially exposes details
of millions of booking.com and Expedia
customers**

'Anybody who has made a hotel booking with these major hotel reservation platforms since 2013 is potentially at risk,' says digital privacy expert

Background: Steganography

- Steganography conceals *confidential* data within *non-confidential* data to evade detection
- The process involves three objects:
 - *Secret object* (payload). Information to be hidden
 - *Cover object*. Embedding target
 - *Carrier object*. Combination of secret and cover objects
 - Publicly available, and visually resemble the cover object

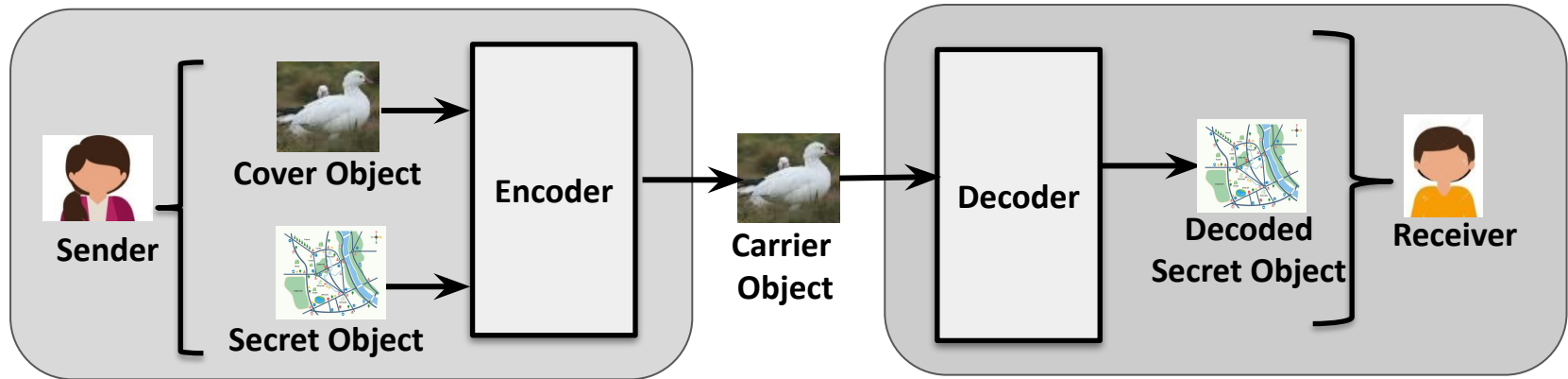


Figure: Steganography pipeline

Existing Approaches and Challenges [1/2]

- Traditional Image Steganography
 - Utilize image properties, e.g., histogram, pixel value difference, or least significant bits
 - Can be identified with statistical analysis
- Deep Image Steganography
 - Encoder and decoder are implemented with Deep Neural Networks

Existing Approaches and Challenges [1/2]

- Traditional Image Steganography
 - Utilize image properties, e.g., histogram, pixel value difference, or least significant bits
 - Can be identified with statistical analysis
- Deep Image Steganography
 - Encoder and decoder are implemented with Deep Neural Networks

RQ1: How **secure** are Deep Image Steganography models?

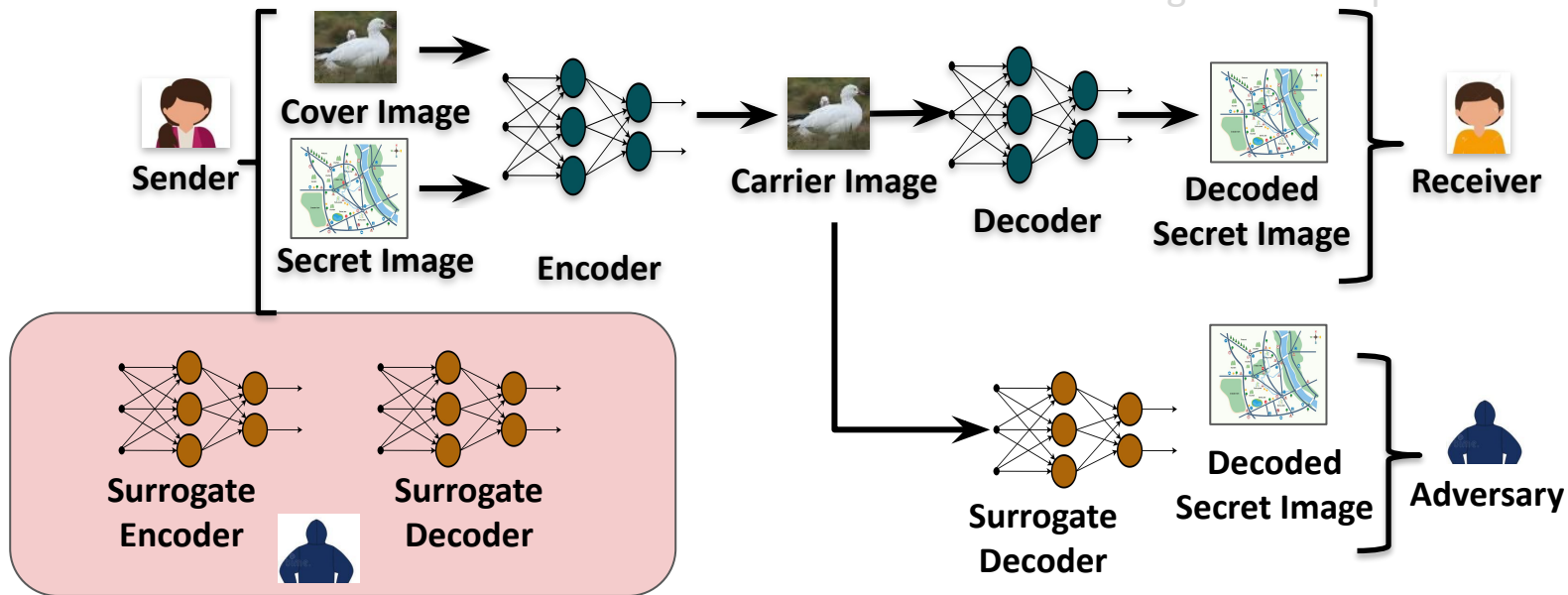
- We show that Deep Image Steganography models are not secure
 - Refer these as *Vanilla Deep Image Steganography*

Security Mindset in Deep Image Steganography [1/2]

An adversary can train surrogate model with varying access to the vanilla deep steganography model

Surrogate models can be employed to:

1. Retrieve hidden information
2. Differentiate carrier images from unperturbed ones



**Adversary trains
its own surrogate model**

Figure: Surrogate model attack

Security Mindset in Deep Image Steganography [2/2]

- Surrogate models can be employed to:
1. Retrieve hidden information
 2. Differentiate carrier images from unperturbed ones

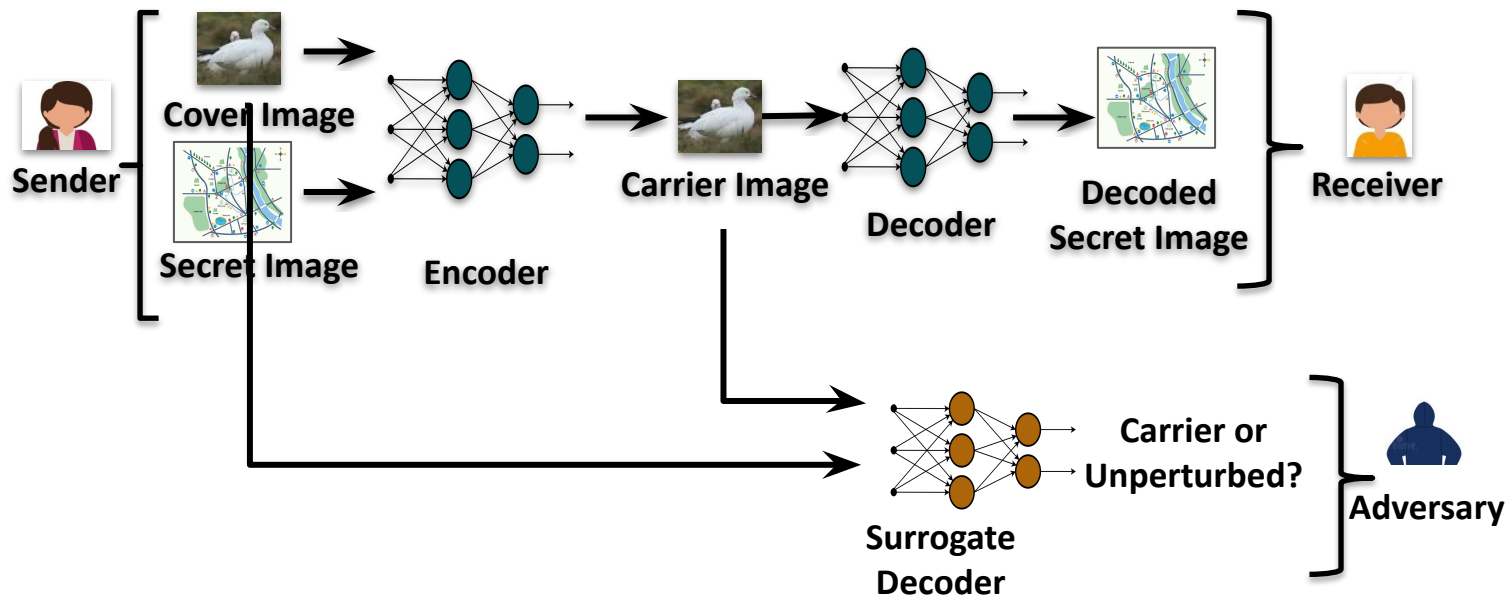


Figure: Surrogate model attack

Existing Approaches and Challenges [2/2]

- Encrypted Steganography
 - Deep image steganography lacks lossless transmission, making it unsuitable for advanced encryption
 - A keyed text steganography [2] incorporates key with concatenation operation
 - We demonstrate **concatenation approach fails for images**
 - [3] hides a scrambled version of secret image with a fixed order of shuffling
 - Adversary can retrieve all **encoded secret images with just one scrambling order**
 - [4] proposes a **complex** and **time-intensive keyed** steganography using asymmetric ECC

[2] Li et al., [DeepKeyStego: Protecting Communication by Key-Dependent Steganography with Deep Networks](#), HPCC'19

[3] Sharma et al., [Hiding Data in Images Using Cryptography and Deep Neural Network](#), Journal of Artificial Intelligence and Systems'19

[4] Duan et al., [A New High Capacity Image Steganography Method Combined with Image Elliptic Curve Cryptography and Deep Neural Network](#), IEEE Access'20

Existing Approaches and Challenges [2/2]

- Encrypted Steganography
 - Deep image steganography lacks lossless transmission, making it unsuitable for advanced encryption
 - A keyed text steganography [2] incorporates key with concatenation operation
 - We demonstrate concatenation approach fails for images

RQ2: Can we build *simple, fast, and effective* deep image steganography model capable of incorporating key for each *<sender, receiver> pair* to ensure defense-in-depth while also *preserving the utility* of the model?

Our approach: *SecureImgStego*

[2] Li et al., [DeepKeyStego: Protecting Communication by Key-Dependent Steganography with Deep Networks](#), HPCC'19

[3] Sharma et al., [Hiding Data in Images Using Cryptography and Deep Neural Network](#), Journal of Artificial Intelligence and Systems'19

[4] Duan et al., [A New High Capacity Image Steganography Method Combined with Image Elliptic Curve Cryptography and Deep Neural Network](#), IEEE Access'20

Our Approach [1/3]

- Introducing **SecureImgStego**, a keyed shuffling based deep steganography model

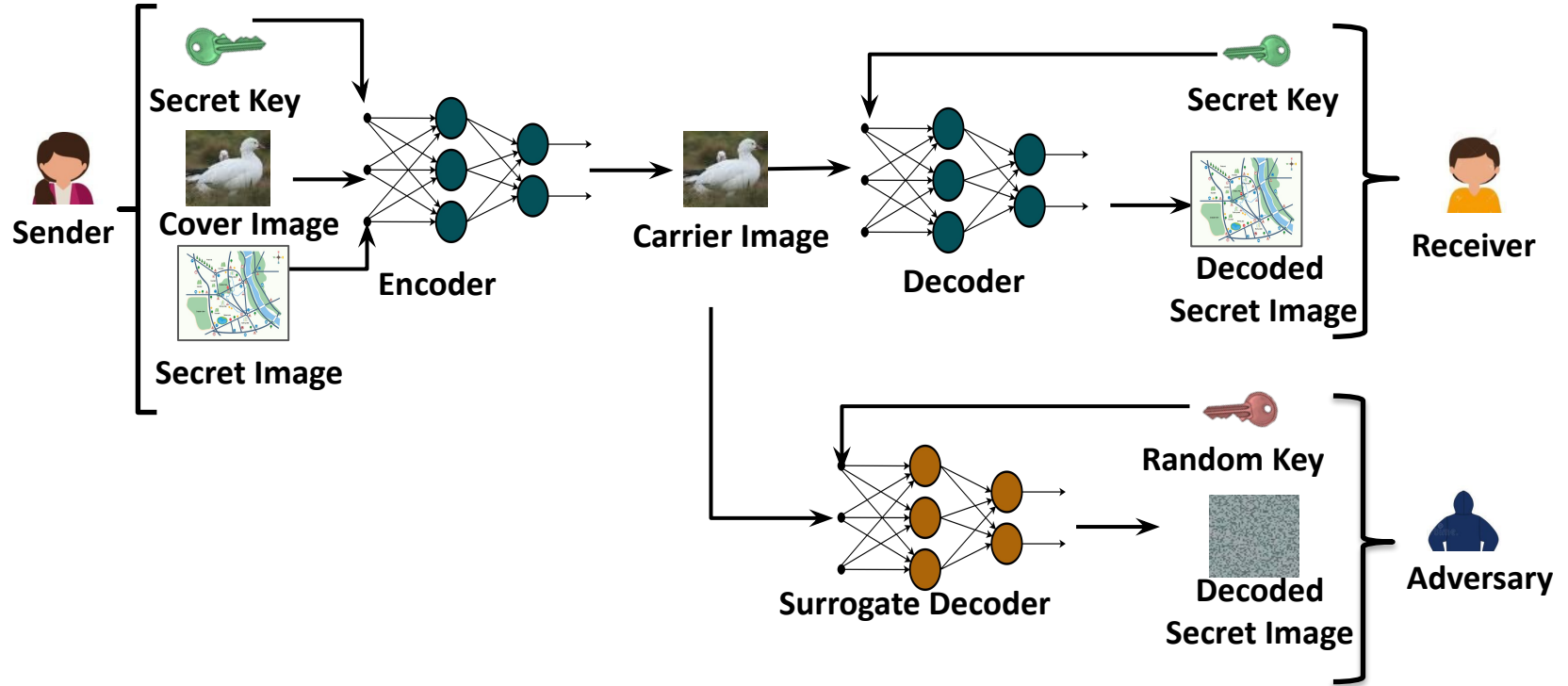


Figure: Sender and receiver share a secret key for encoding and decoding

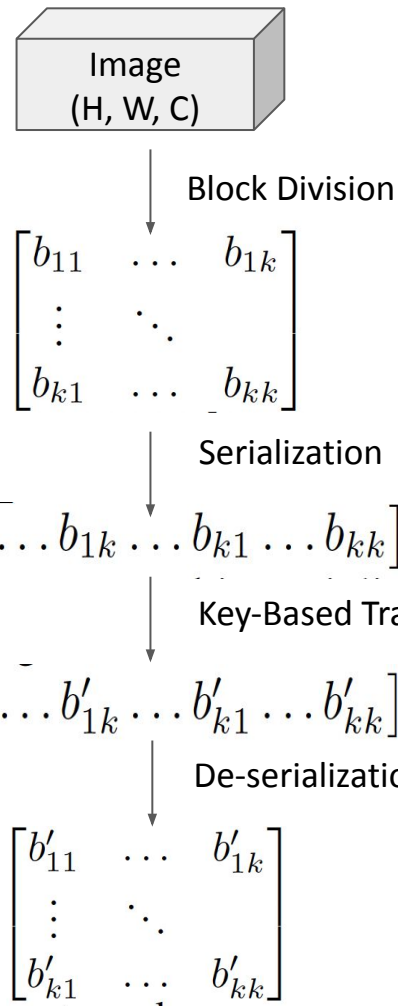
Our Approach [2/3]

Block Shuffling Procedure

- Block Division: An image of dimension (H, W, C) is divided into blocks (b)
- Serialization: These blocks are serialized in a row-wise fashion
- Key-Based Transformation: Using a key of the same length, we transform this serialization into a different mapping
- De-serialization: Finally, we de-serialize this mapping in a row-wise fashion to construct the shuffled image



Figure: Shuffling process illustration



Our Approach [3/3]

- Explore various forms of key shuffling techniques
 - Multi-level block shuffling
 - Blocks are shuffled in spatial dimension
 - 1B (Block size = 1)
 - 4B (Block size = 4)
 - 8B (Block size = 8)
 - Keyed pixel shuffling
 - Pixels are shuffled along the spatial and channel dimension
 - Key concatenation and block shuffling combination
 - Key concatenation + 8B Shuffling

Threat Model

- Attacker Classification
 - White-box: Full knowledge of model architecture and hyperparameters
 - Grey-box: Partial knowledge of model architecture
 - Black-box: No knowledge of model architecture
- Data Access
 - None of the adversaries have access to the original dataset
- Secure Key Transmission
 - Each <sender, receiver> pair shares a key
 - Keys are assumed to be securely transmitted before usage
 - Adversaries do not have access to these keys

Our System Architecture [1/2]

1) Encoder (3)

- Shuffling layer: Takes secret image (S) and shared key (K) to create Shuffled Secret Image
- Prep net: Extracts high-level features
- Hiding net: Embeds prepnet features into cover image (C) to create carrier image (C')

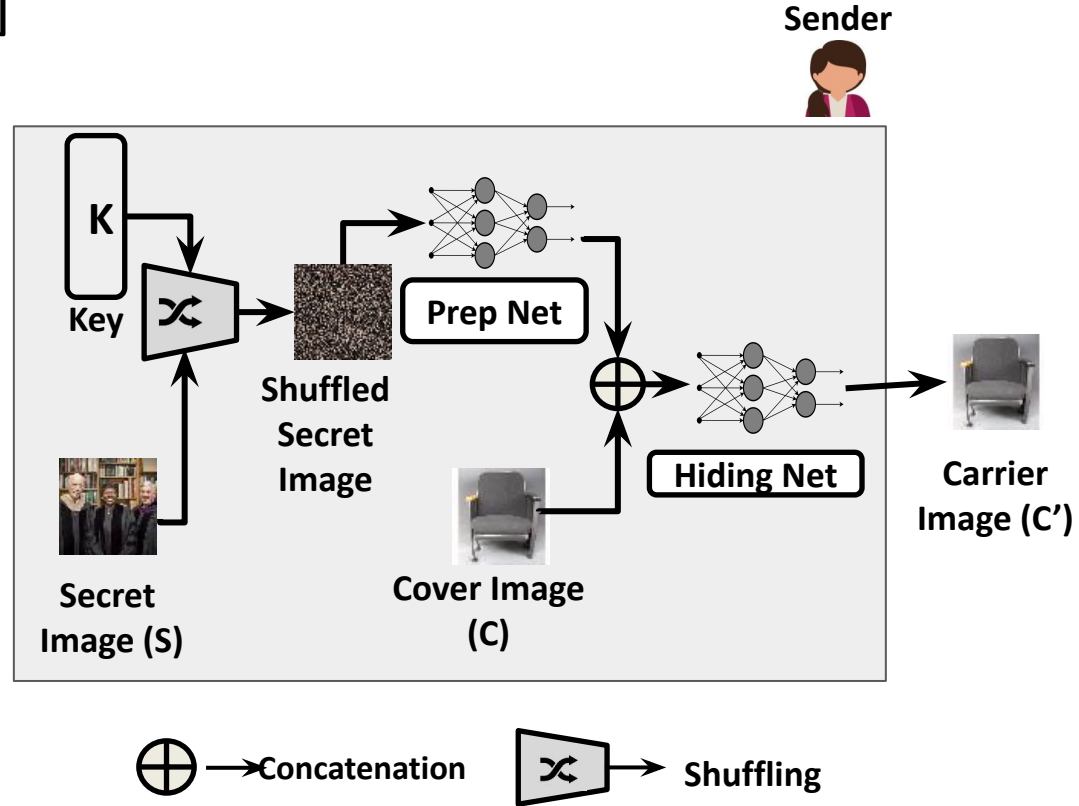


Figure: Sender's encoder includes shuffling, prep net, and hiding net.

Our System Architecture [2/2]



Receiver

2) Decoder (2)

- Reveal net: Extracts encoded secret image from carrier
- Deshuffling layer: Uses shared secret key for retrieval

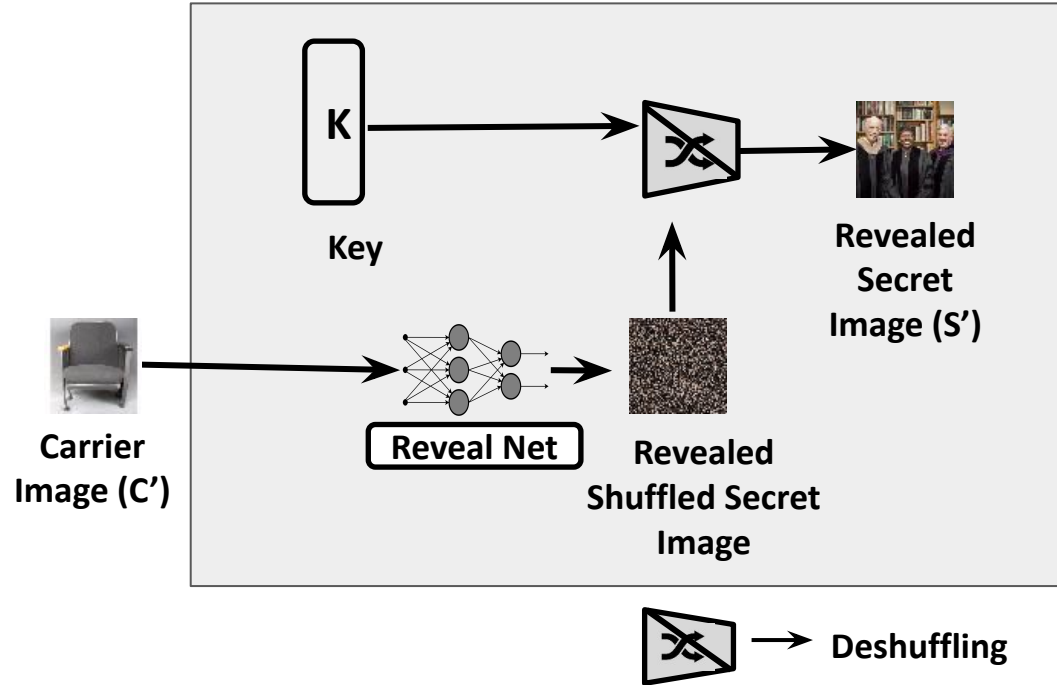

















Figure: Receiver's decoder has reveal net and deshuffling.

Attack on Vanilla Deep Steganography: Reveal Secret Information

- White-Box Adversary
 - Full knowledge of vanilla model architecture and hyperparameters
 - Differences arise from distinct training data
- Grey-Box Adversary
 - Partial knowledge of vanilla model architecture
 - Knows about the usage of *prep*, *reveal*, and *hiding* components
 - Adopts *InceptionV3* to develop *prep*, *reveal*, and *hiding* nets
- Black-Box Adversary
 - No details about vanilla model architecture
 - Uses *VGG16* as a basis for a single encoder-decoder

ATTACKS IN VANILLA DEEP STEGANOGRAPHY









Secret Image	Cover Image	Decoded Secret Image		
		White-box	Grey-box	Black-box
				
				
				

- Decoded secret images are ***conspicuous enough to reveal the secret information***

Attack on Vanilla Deep Steganography: Carrier Image Distinction

- *Steganalysis* identifies images as
 - carriers (with hidden information)
 - unperturbed (no hidden data)
- Adversary can use the surrogate decoder to serve the same purpose
 - **Unperturbed image** yields **meaningless decoded content**
 - **Carrier image** produces **meaningful content**, indicating hidden data

STEGANALYSIS WITH DEEP STEGANOGRAPHY

Input Type	Input Image	Decoded Secret Image		
		White-box	Grey-box	Black-box
Unperturbed				
Carrier				

Experimental Setup

- Dataset: We use the *Tiny-ImageNet* dataset
 - 110K color images
 - Dimensions: 64x64 pixels, RGB
 - Training Set: 100K images
 - Secret: 50K, Cover: 50K
 - Test Set: 10K images
 - Secret: 5K, Cover: 5K
 - Key Generation: *Random numbers as shuffling key* for each secret-cover image pair
- Extensively assessing ***SecureImgStego*** against existing methods:
 - Human perceptibility
 - Key sensitivity
 - Adaptivity
 - Cover image availability
 - Keyspace
 - Steganalysis robustness

Experimental Results: Performance Analysis

- Shuffling Methods Comparison

- 4B and 8B shuffling methods outperform vanilla deep steganography in image quality
 - *CNN benefits from block creation*
- 1B and pixel shuffling *perform poorly* due to pixel scattering

Encoding-Decoding Time

- Longer computational time with increased shuffling depth

COMPARISON AMONG DIFFERENT IMAGE STEGANOGRAPHY APPROACHES IN TERMS OF RMSE, SSIM, PSNR, AND COMPUTATION LATENCY










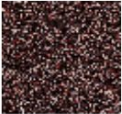


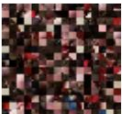


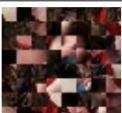


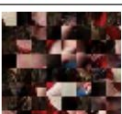
Method	RMSE ↓		SSIM ↑		PSNR ↑		Latency ↓	
	Secret	Cover	Secret	Cover	Secret	Cover	Encoding	Decoding
Vanilla Deep Steganography [5]	8.34	8.59	0.94	0.90	30.23	29.72	5.91 s	4.08 s
Key Concat [11]	5.73	7.28	0.97	0.92	33.39	31.17	6.22 s	4.30 s
Pixel Shuffling	50.71	14.05	0.31	0.89	14.45	25.98	6.83 s	5.04 s
1B Shuffling	10.39	10.87	0.92	0.86	28.76	27.70	6.4 s	4.48 s
4B Shuffling	5.72	8.89	0.96	0.89	33.51	29.37	6.29 s	4.75 s
8B Shuffling	7.07	8.46	0.95	0.90	31.59	29.85	6.16 s	4.30 s
Key Concat [11] + 8B Shuffling	8.33	11.34	0.95	0.90	30.00	27.90	6.40 s	4.48 s

Experimental Results:

Sensitivity to Random Key

- Key Concatenation
 - *Performs poorly for images*, unlike text data
 - In images, subtle changes in RGB values are imperceptible, unlike texts from 'A' to '@'
- Keyed Shuffling
 - *Significant differences between retrieved images with correct and random keys*
 - Pixel Shuffling performs poorly even with correct keys

EFFECT OF ADVERSARY GENERATED RANDOM KEY

Secret Image	Model	Carrier Image	Revealed Secret Image	
			With Correct Key	With Random Key
	Key Concat [11]			
	Pixel Shuffling			
	1B Shuffling			
	4B Shuffling			
	8B Shuffling			
	Key Concat [11]+ 8B Shuffling			

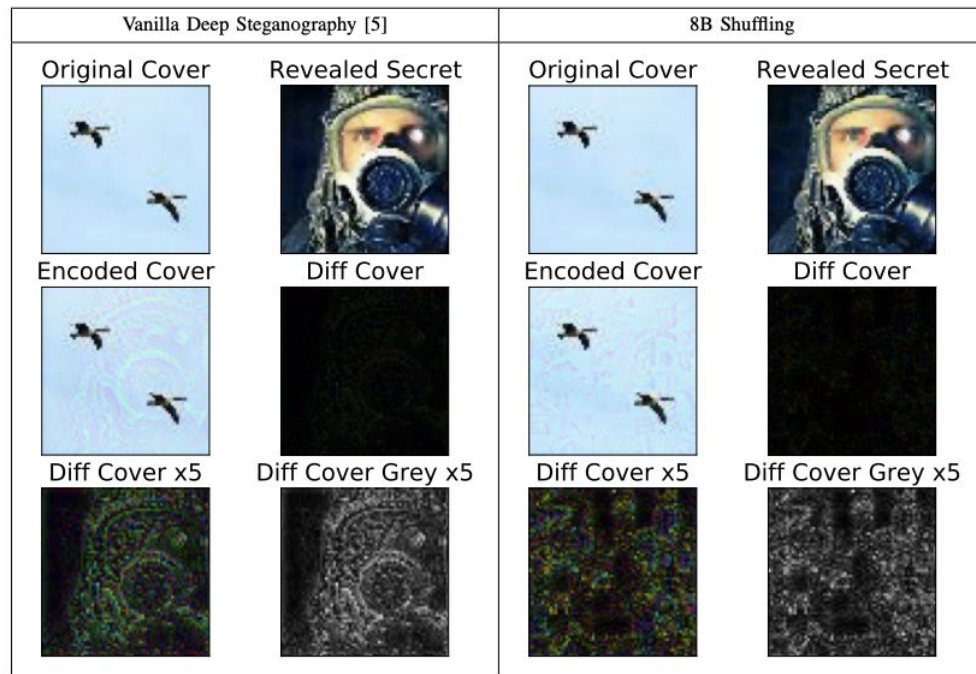
Experimental Results: Availability of Cover Image

In Vanilla Deep Steganography, *difference between cover and carrier images reveals part of the secret image*

Advantages of Keyed Shuffling

- *Keyed shuffling methods do not have this limitation*
- Even grayscale diff enhancement reveals no visual information

EFFECT OF ENCRYPTION



Conclusion & Future Work

- ***Vulnerabilities in Vanilla Deep Steganography***
 - Identified and demonstrated with *real attacks*
- ***Keyed Image Steganography***
 - Proposed ***SecureImgStego***, a secure keyed deep image steganography
 - Ensures secure communication between sender and receiver, even in the presence of a surrogate model
- ***Future Directions***
 - Exploring applications in audio and video data
 - Investigating hiding multiple secret images within a single cover image using SecureImgStego

Any Questions?

Thank you!

Scan the QR codes to reach us!



Our System Architecture [1/2]

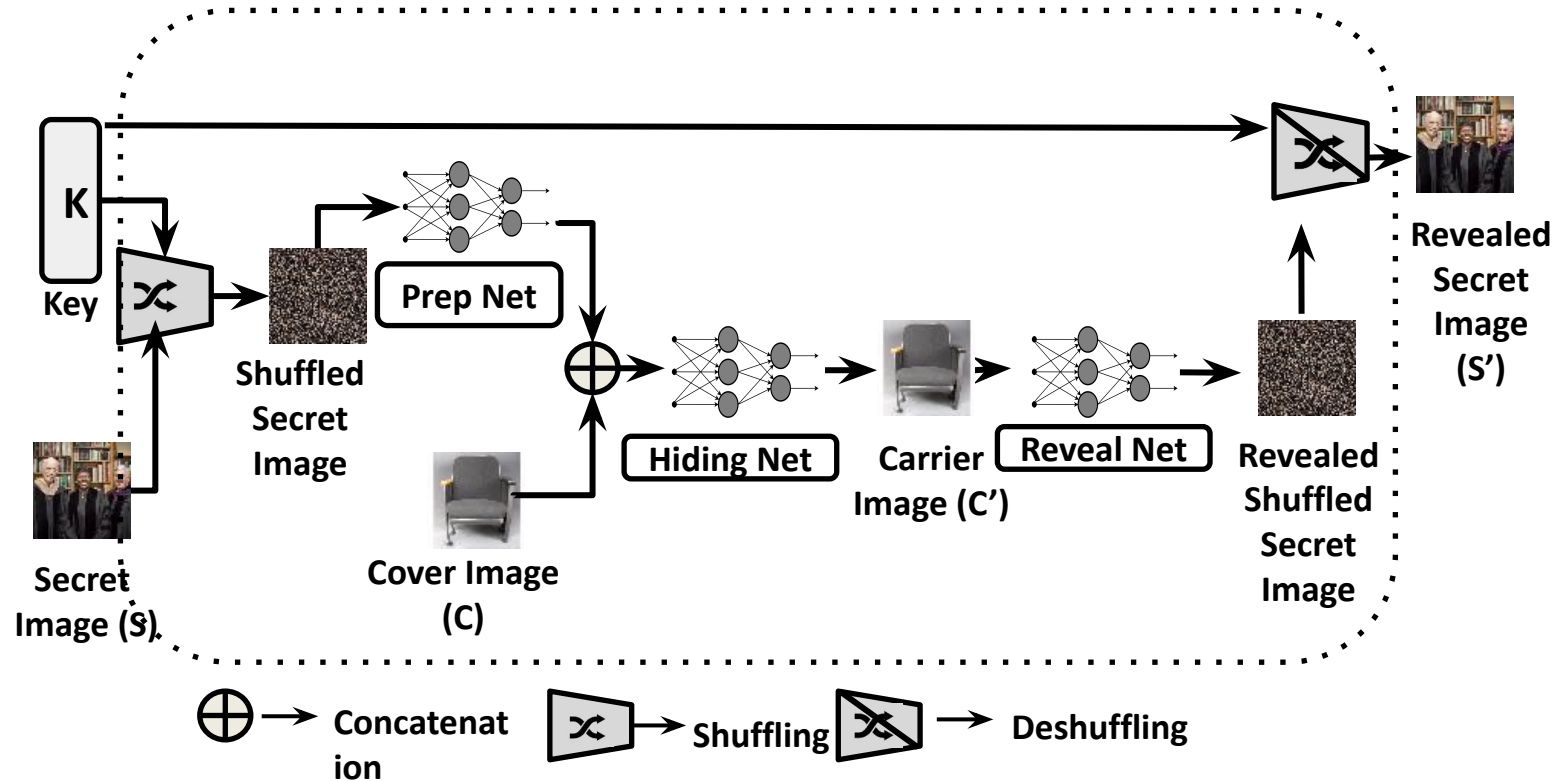












Figure: SecureImgStego architecture

Experimental Results: Adaptive Depth for Multi-level Block Shuffling

- Correlation with Secret Image Content
 - **Depth of block shuffling is correlated with secret image contents**
 - If the secret image retains significant information after d-depth block shuffling, a higher depth is advisable
- Effect on Security
 - 8B shuffling with a random key leaks crucial information of human face
 - 4B shuffling still reveals human eyes
 - 1B shuffling makes it impossible for the adversary to comprehend the content
- Trade-off Between Quality and Security
 - **Granularity of block shuffling impacts reconstructed image** quality and security





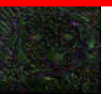
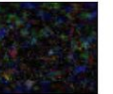




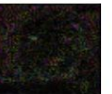
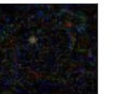




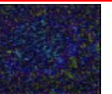
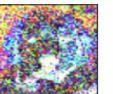





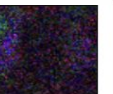




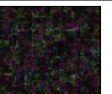
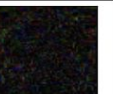




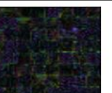
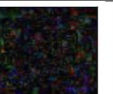






EFFECT OF DIFFERENT DEPTH VALUES IN MULTI-LEVEL BLOCK SHUFFLING

Secret Image	Model	Depth	Carrier Image	Revealed Secret Image	
				With Correct Key	With Random Key
	8B Shuffling	3			
	4B Shuffling	4			
	1B Shuffling	6			

Experimental Results: Visual Analysis

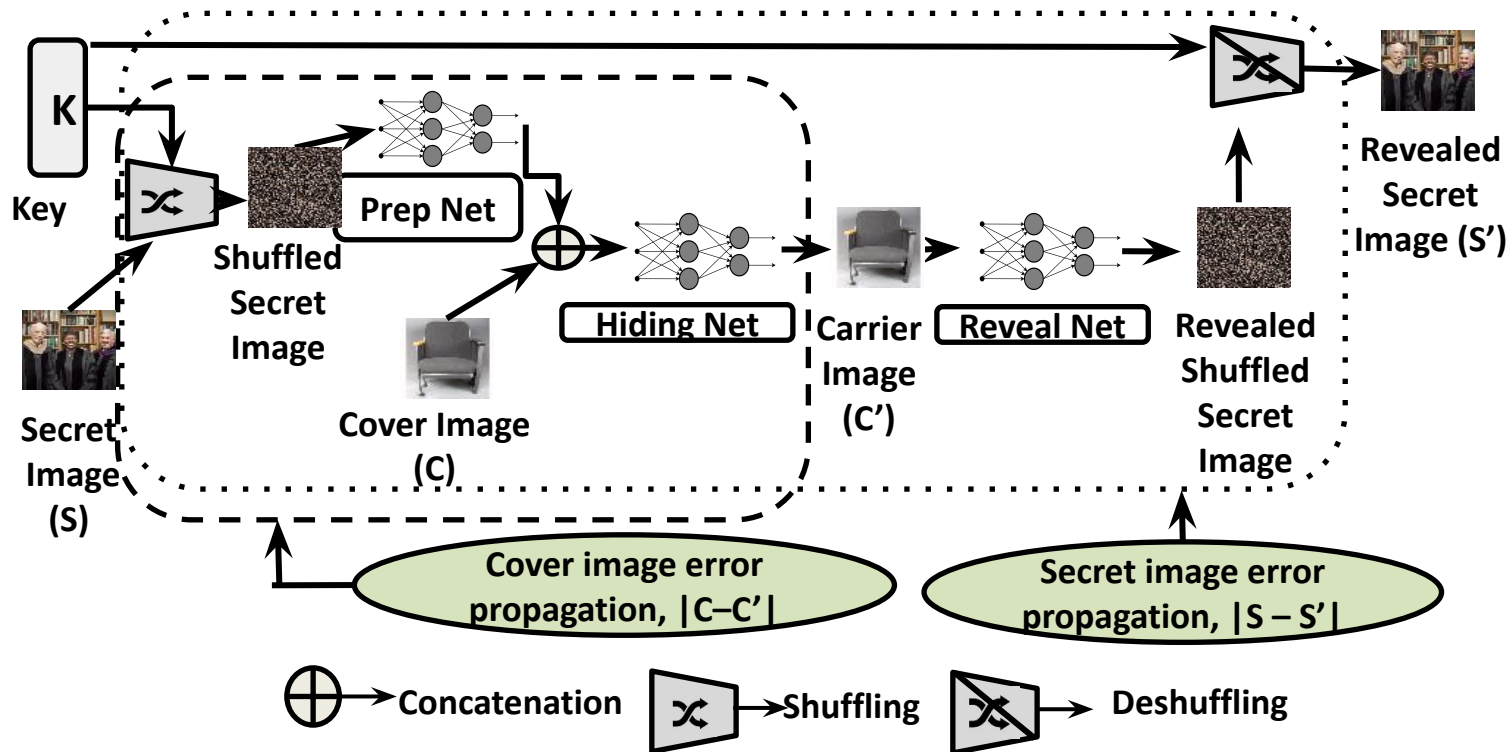
- Results for *Vanilla Deep Steganography* and *Key Concatenation*
 - *Differences reveal the shape of the secret image* (e.g., a dog's face)
 - Key Concat *is no better than* vanilla deep steganography
- Keyed Shuffling
 - *Addresses the issue of leaking secret image* details from carrier images
 - Pixel shuffling performs poorly in reconstructing the secret image due to spatial and channel shuffling

VISUAL ANALYSIS OF DIFFERENT STEGANOGRAPHY METHODS' PERFORMANCES

Model	Cover	Secret	Encoded Cover	Decoded Secret	Diff Cover×5	Diff Secret×5
Vanilla Deep Steganography [5]						
Key Concat [11]						
Pixel Shuffling						
1B Shuffling						
4B Shuffling						
8B Shuffling						
Key Concat [11] +8B Shuffling						

Our System Architecture [3/3]

The Overall Loss: $L(C, C', S, S') = || C - C' || + || S - S' ||$



Background: Importance of Steganography

- Privacy Protection
 - Can be simply used to save data on a location
 - e.g., private banking & medical information
- National Security
 - Government usage in military communication
- Superior than Cryptography
 - Hides both the **content** and the **existence** of secret message

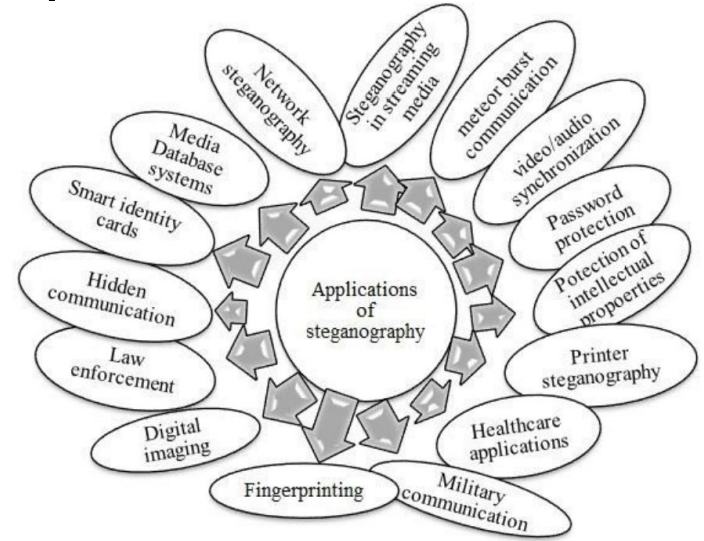
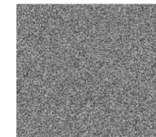


Figure: Applications of Steganography [1]



Carrier Object



Crypto Object

Existing Approaches and Challenges [1/2]

- Traditional Image Steganography
 - Utilize image properties, e.g., histogram, pixel value difference, or least significant bits
 - Can be identified with statistical analysis
- Deep Image Steganography
 - Encoder and decoder are implemented with Deep Neural Networks

RQ1: How **secure** are Deep Image Steganography models?

- We show that Deep Image Steganography models are not secure
 - Refer these as *Vanilla Deep Image Steganography*

Existing Approaches and Challenges [2/2]

- Encrypted Steganography
 - Deep image steganography lacks lossless transmission, making it unsuitable for advanced encryption
 - A keyed text steganography [2] incorporates key with concatenation operation
 - We demonstrate **concatenation approach fails for images**
 - [3] hides a scrambled version of secret image with a fixed order of shuffling
 - Adversary can retrieve all **encoded secret images with just one scrambling order**
 - [4] proposes a **complex** and **time-intensive keyed** steganography using asymmetric ECC

[2] Li et al., [DeepKeyStego: Protecting Communication by Key-Dependent Steganography with Deep Networks](#), HPCC'19

[3] Sharma et al., [Hiding Data in Images Using Cryptography and Deep Neural Network](#), Journal of Artificial Intelligence and Systems'19

[4] Duan et al., [A New High Capacity Image Steganography Method Combined with Image Elliptic Curve Cryptography and Deep Neural Network](#), IEEE Access'20

Existing Approaches and Challenges [2/2]

- Encrypted Steganography
 - Deep image steganography lacks lossless transmission, making it unsuitable for advanced encryption
 - A keyed text steganography [2] incorporates key with concatenation operation
 - We demonstrate concatenation approach fails for images

RQ2: Can we build *simple, fast, and effective* deep image steganography model capable of incorporating key for each *<sender, receiver> pair* to ensure defense-in-depth while also *preserving the utility* of the model?

[2] Li et al., [DeepKeyStego: Protecting Communication by Key-Dependent Steganography with Deep Networks](#), HPCC'19

[3] Sharma et al., [Hiding Data in Images Using Cryptography and Deep Neural Network](#), Journal of Artificial Intelligence and Systems'19

[4] Duan et al., [A New High Capacity Image Steganography Method Combined with Image Elliptic Curve Cryptography and Deep Neural Network](#), IEEE Access'20