Leaking Sensitive Financial Accounting Data in Plain Sight using Deep Autoencoder Neural Networks

Marco Schreyer[1], **Christian Schulze**[2], **Damian Borth**[1]
[1]University of St. Gallen (HSG), St. Gallen, Switzerland
[2]German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany
{marco.schreyer, damian.borth}@unisg.ch, christian.schulze@dfki.de

Abstract

Nowadays, organizations collect vast quantities of sensitive information in 'Enterprise Resource Planning' (ERP) systems, such as accounting relevant transactions, customer master data, or strategic sales price information. The leakage of such information poses a severe threat for companies as the number of incidents and the reputational damage to those experiencing them continue to increase. At the same time, discoveries in deep learning research revealed that machine learning models could be maliciously misused to create new attack vectors. Understanding the nature of such attacks becomes increasingly important for the (internal) audit and fraud examination practice. The creation of such an awareness holds in particular for the fraudulent data leakage using deep learning-based steganographic techniques that might remain undetected by state-of-the-art 'Computer Assisted Audit Techniques' (CAATs). In this work, we introduce a real-world 'threat model' designed to leak sensitive accounting data. In addition, we show that a deep steganographic process, constituted by three neural networks, can be trained to hide such data in unobtrusive 'day-to-day' images. Finally, we provide qualitative and quantitative evaluations on two publicly available real-world payment datasets.

Introduction

In recent years data became one of the organizations most valuable asset ('the new oil' (The Economist 2017)). As a result, data protection, preventing it from being stolen or leaked to the outside world, is often considered of paramount importance. This observation holds in particular for data processed and stored in *Enterprise Resource Planning* (ERP) systems. Steadily, these systems collect vast quantities of business process and accounting data at a granular level. Often the data recorded by such system encompasses sensitive information such as (i) the master data of customers and vendors, (ii) the volumes of sales and revenue, and (iii) strategic information about purchase and sales prices. Nowadays, the leakage of such information poses a severe issue for companies as the number of incidents and the costs to those experiencing them continue to increase[1]. The potential damage and adverse consequences of a leakage incident may result in indirect losses, such as violations of (privacy) regulations or settlement and compensation fees. Furthermore, it can also result in indirect losses, such as damaging a company's goodwill and reputation or the exposure of the intellectual property. Formally, *data leakage* can be defined as the *'unauthorized transmission of information from inside an organization to an external recipient'* (Shabtai, Elovici, and Rokach 2012). The key characteristic of a data leak incident is that it is carried out by 'insiders' of an organization, e.g., employees or subcontractors. The detection of such insider attacks remains difficult since it often involves the misuse of legitimate credentials or authorizations to perform such an attack.

Recent breakthroughs in artificial intelligence, and in particular deep neural networks (LeCun, Bengio, and Hinton 2015), created advances across a diverse range of application domains such as image classification (Krizhevsky, Sutskever, and Hinton 2012), speech recognition (Saon et al. 2015), language translation (Sutskever, Vinyals, and Le 2014) and game-play (Silver et al. 2017). Due to their broad application, these developments also raised awareness of the unsafe aspects of deep learning (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014; Papernot et al. 2017), which become increasingly important for the (internal-) audit practice. The research of the potential impact of deep learning techniques in accounting and audit is still in an early stage (Sun 2019; Schreyer et al. 2017, 2019b). However, we believe that it is of vital relevance to understand how such techniques can be maliciously misused in this sphere to create new attack vectors (Ballet et al. 2019; Schreyer et al. 2019a). This holds in particular for the fraudulent[2] data leakage using deep learning-based steganographic techniques that might remain undetected by state-of-the-art *Computer Assisted Audit Techniques* (CAATs).

In general, the objective of *steganography* denotes a secret communication (Shabtai, Elovici, and Rokach 2012): a sender encodes a message into a cover medium, such as an image or audio file, such that the recipient can decode the message. At the same time, a potential (internal-) auditor or

[1]An overview of leakage incidents: https://selfkey.org/data-breaches-in-2019/.

[2]According to (Garner 2014) the term *fraud* refers to the use of one's occupation for personal enrichment through the deliberate misuse or misapplication of the using organization's resources or assets, e.g. the booking of fictitious sales, fraudulent invoices, wrong recording of expenses.

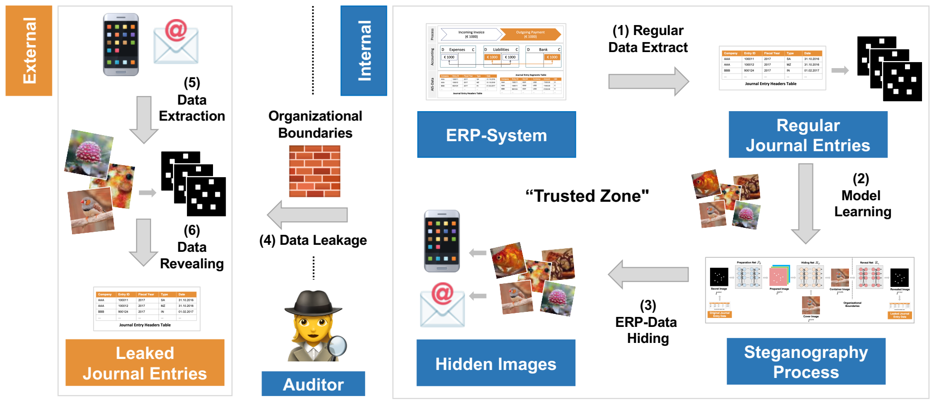

Figure 1: Exemplary data leakage *threat model* designed to leak sensitive accounting data such as (i) detailed journal entries or (ii) sensitive customer and vendor master data. The sensitive data is encoded into non-suspicious appearing 'container' images that transmitted to a location outside of the organization.

fraud examiner cannot judge whether the cover medium contains a secret message or not. In this paper, we present a deep learning-based steganographic process designed to leak sensitive accounting data. We regard this work to be an initial step towards the investigation of such future challenges of audits or fraud examinations. In summary, we present the following contributions: First, we describe a real-world 'threat model' designed to leak sensitive accounting data using deep steganographic techniques. Second, we show that deep neural networks are capable of learning to hide the accounting data in everyday data, such as simple 'cover' images. The created cover images are deliberately designed not to raise awareness and misguide auditors or fraud examiners. Third, to complete the attack scenario, we demonstrate how the hidden information can be revealed from the cover images upon the successful leak.

Related Work

A wide variety of steganography settings and methods have been propsed in the literature; most relevant to our work are methods for blind image steganography, where the message is encoded in an image and the decoder does not have access to the original cover image.

Least Significant Bit (LSB) Methods: The Least Significant Bit (Chandramouli and Memon 2001; Fridrich, Goljan, and Du 2001; Mielikainen 2006; Viji and Balamurugan 2011; Qazanfari and Safabakhsh 2017) describes a classic steganographic algorithm. In general, each pixel of a digital image is comprised of three bytes (i.e., 8 binary bits) that represent the RGB chromatic values respectively. The $n$bit-LSB algorithm replaces the least n significant bits of the cover image by the n most significant bits of the secret image. For each byte, the significant bits dominate the colour values. This way, the chromatic variation of the container image (altered cover) is minimized. Revealing the concealed secret image can be accomplished by reading the n least significant bits and performing a bit shift despite that its distortion is often not visually observable. However, the LSB algorithm is, highly vulnerable to steganalysis. Often statistical analyses can easily detect the pattern of the altered pixels as shown in (Fridrich, Goljan, and Du 2001; Lerch-Hostalot and Megías 2016; Luo, Huang, and Huang 2010). Recent works have therefore focused on methods that preserve the original image statistics or the design of sophisticated distortion functions (Holub and Fridrich 2012; Holub, Fridrich, and Denemark 2014; Long and Li 2018; Pevný, Filler, and Bas 2010; Swain 2018; Tamimi, Abdalla, and Al-Allaf 2013).

Discrete Cosine Transform (DCT) Methods: To overcome the drawbacks of the LSB algorithm, the variant HRVSS (Eltahir et al. 2009) and (Muhammad et al. 2018) focus for example, on the unique biological trait of human eyes when hiding grey images in colour images. Other approaches utilize the bit-plane complexity segmentation in either the spatial or the transform domain (Kawaguchi and Eason 1999; Spaulding et al. 2002; Ramani et al. 2007). Other algorithms embed secret information in the Discrete Cosine Transformation 'DCT' frequency domain of an image. This is achieved by deliberately changing the DCT coefficients (Chae and Manjunath 1999; Kaur, Kaur, and Singh 2011; Nag et al. 2011; Zhang et al. 2018). As many of those coefficients are zero in general, they can be exploited to embed information in the images frequency domain rather than the spatial domain (El_Rahman 2018; Cheddad et al. 2010; Sadek, Khalifa, and Mostafa 2015).

Deep Learning Methods: Recently, deep learning-based steganography methods have been proposed to encode (binary) text messages in images (Baluja 2017; Zhu et al. 2018). Prior works (Husien and Badi 2015; Pibre et al. 2015) mostly focused on the decoding of the hidden information (in terms of determining which bits to extract from the container image) to increase the transmission accuracy. Besides, deep learning methods are also increasingly employed in the context of steganalysis to detect the information potentially hidden (Pibre et al. 2016; Qian et al. 2015; Tan and Li 2014; Xu, Wu, and Shi 2016). In this work, we build on the ideas of

(Baluja 2017; Hayes and Danezis 2017) that proposed an entire steganographic based on deep networks, encompassing a preparation, an encoding (hiding) and decoding (revealing) network. To the best of our knowledge, this work presents the first analysis of a deep neural network based data leakage attack targeting sensitive financial accounting data.

Deep Steganography Threat Model

To establish such a data leakage attack, we introduce a *threat model* depicted in Fig. 1, in which a person or group of people, referred to as the *perpetrator*, within an organization intends to transmit 'leak' sensitive data of the organization to an external destination or recipient. To initiate the attack, a perpetrator will query the ERP system to extract the sensitive accounting data to be leaked, such as detailed journal entry data or customer and vendor master data tables (1). Afterwards, the extracted entries are converted into binary images, referred to as *secret images*, exhibiting a 'one-hot' encoded representation of the data. One could think of converting the information into a Quick Response (QR) code representation using designated publicly available software[3].

Following, a population of images referred to as *cover images* is selected. Usually, the cover images contain unobtrusive content to leak the secret information covertly. Such images can, for example, be obtained from prominent and publicly available image databases such us (Thomee et al. 2016; Russakovsky et al. 2015). Following, a deep *steganographic model* is learned utilizing a deliberately designed training process. The process is comprised of a sequence of deep neural networks that process both the secret and the cover images. The model is trained to create a *container image* that efficiently encodes the information contained in the secret image into the cover image (2). Thereby the encoding is regularized to hide the secret information with minimal distortion of the visual appearance of the cover image. Simultaneously, the model is trained to also decode the encoded secret information from the container image.

Upon successful training, the model is exploited to facilitate the attack (3). Therefore, a single or a series of container images are created that conceal the sensitive data to be leaked. The created container images are then transmitted either (i) electronically (e.g., via email or files-haring) or (ii) physically (e.g., via mobile phone or USB flash drive) to a location outside of the organizational boundaries, usually referred to as 'safe zone'. (4). Visually the container images seem to correspond to unobtrusive 'everyday' images but encode the sensitive information to be leaked. As a result, they exhibit a high chance of being judged as non-suspicious throughout an internal audit. Once the container images are successfully transmitted, the secret binary images can be revealed using the steganographic model (5). Ultimately the sensitive data can be encoded at high accuracy (6).

Deep Steganography Process

Inspired by and building on the work of Baluja (Baluja 2017) we investigate if a deep-learning based steganography process can be misused to leak sensitive accounting data. The proposed process, is constituted of three neural networks: a *preparation* network P_θ, a *hiding* network H_ϕ, and a *reveal* network R_γ. The distinct networks are trained 'end-to-end' and θ, ϕ, and γ denote the trainable parameters of each network respectively.

To establish a potential data leak, a set of X of N journal entries $x^1, x^2, ..., x^n$ (or other tabular data record) is extracted from the ERP system to be attacked. Each journal entry x^i consists of M accounting specific attributes $x_1^i, x_2^i, ..., x_j^i, ..., x_m^i$. The individual attributes x_j describe the journal entries details, e.g., the entries' fiscal year, posting type, posting date, amount, general-ledger. Afterwards, each entry is sequentially processed by the distinct networks as described in the following.

Prepare Step: Each entry (or multiple entries) is converted into a *secret image*. The secret image $I^{sec} \in \{0,1\}^{H \times W}$ of shape $H \times W$ contains a binary representation of the information to be leaked. The preparation network P_θ receives I^{sec} and converts it into a RGB *prepared image* $I^{pre} \in \{0, 255\}^{C \times H \times W}$ that exhibits the desired characteristics to be hidden.

Hide Step: The hiding process is then initiated using the prepared image and a cover image. The RGB cover image $I^{cov} \in \{0, 255\}^{C \times H \times W}$ of shape $C \times H \times W$ corresponds to an arbitrary 'everyday' image and is used to hide the prepared image. The hiding network H_ϕ receives a cover image I^{cov} and the prepared image I^{pre}. Using both input images, H_ϕ produces a RGB *container image* I_{con} of the same shape as I^{cov}.

Reveal Step: Subsequently, the container image is passed to the reveal network R_γ to produce a binary reveal image $I^{rev} \in \{0,1\}^{H \times W}$ that exhibits a high similarity to the original prepared image I^{pre}. Ultimately, I^{rev} is supposed to reveal the secret information of I^{sec} hidden in I^{con}. The training of the architecture follows a dual training objective, namely (1) the minimization of the cover distortion (*cover loss*) and (2) the preservation of the secret information (*secret loss*) when optimizing the network parameters.

Cover loss: The first training objective requires that the container image I^{con} exhibits a high visual similarity to the cover image I^{cov}. Formally defined as the mean-squared difference $\mathcal{L}_{\theta,\phi}^{cov}$ between both images, given by:

$$\mathcal{L}_{\theta,\phi}^{cov} = \sum_{x=0}^{C-1} \sum_{y=0}^{H-1} \sum_{z=0}^{W-1} (I_{x,y,z}^{cov} - I_{x,y,z}^{con})^2, \qquad (1)$$

where C, H, and W denote the shape of both RGB images and θ and ϕ the network parameters.

Secret loss: The second training objective requires that the revealed image I^{rev} should contain the same binary information as encoded in the original secret image I^{sec}. Formally defined as the binary cross-entropy difference $\mathcal{L}_{\theta,\phi,\gamma}^{sec}$ of between images, given by:

$$\begin{aligned}\mathcal{L}_{\theta,\phi,\gamma}^{sec} = & \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} I_{x,y}^{sec} \log I_{x,y}^{rev} \\ & + (1 - I_{x,y}^{sec}) \log(1 - I_{x,y}^{rev}),\end{aligned} \qquad (2)$$

[3]GitHub: https://github.com/Bacon/BaconQrCode

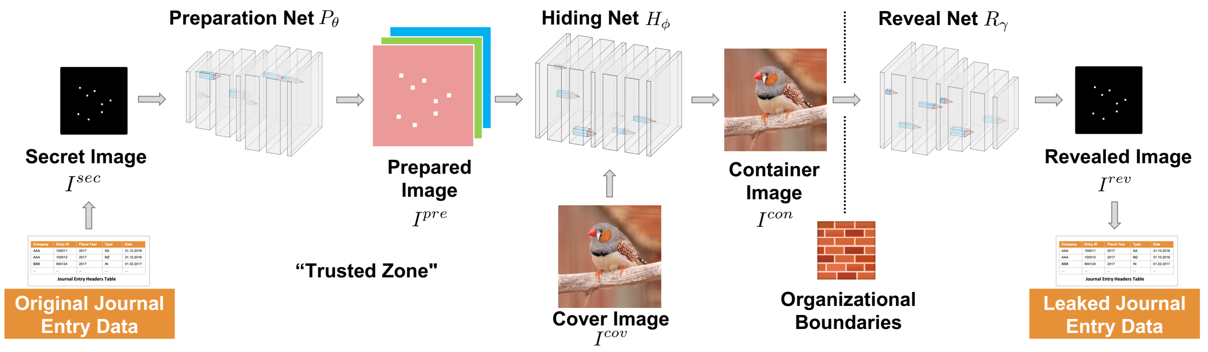

Figure 2: The data leakage process as introduced in (Baluja 2017), applied to learn a steganographic model of real-world accounting data. The process is designed to encode and decode sensitive information into unobtrusive 'day-to-day' cover images.

where H, and W denote the shape of both binary images and θ, ϕ and γ the network parameters. We train the network parameters θ, ϕ, and γ batch-wise using stochastic gradient descent to minimize a combined steganographic loss function $\mathcal{L}^{ALL}_{\theta,\phi,\gamma}$ over the distributions of secret and cover images, as defined by:

$$\mathcal{L}^{ALL}_{\theta,\phi,\gamma} = \alpha \cdot \frac{1}{N} \sum_{i=0}^{N-1} \mathcal{L}^{cov}_{\theta,\phi} + \beta \cdot \frac{1}{N} \sum_{i=0}^{N-1} \mathcal{L}^{sec}_{\theta,\phi,\gamma}, \quad (3)$$

where N denotes the size of the training batch and the factors α and β balance both losses. In summary, the approach builds upon the idea of autoencoder neural networks originally proposed in (Hinton and Salakhutdinov 2006). We encoded two input images such that the learned intermediate representation I^{con} comprises an image that appears as similar as possible I^{cov} and thereby hides the accounting data information contained in I^{sec}.

Datasets and Experimental Setup

Due to the high confidentiality of real-world accounting data and to allow for reproducibility of our results, we evaluate the proposed methodology based on two publicly available datasets of real-world city payments. The payment datasets serve as *secret data* to be hidden. We use a prominent dataset of everyday images that serves as *cover data* to hide the payment information.

Secret Data: The secret datasets to be leaked, encompass two publicly available payment datasets of the city of Philadelphia and the city of Chicago. The majority of attributes recorded in both datasets (similar to real-world ERP data) correspond to categorical (discrete) variables, e.g. posting date, department, vendor name, document type. For each dataset we pre-process the original payment line-item attributes to (i) remove semantically redundant attributes and (ii) obtain a binary ('one-hot') representation of each payment record. The following descriptive statistics summarise both datasets upon successful data pre-processing:

The **'City of Philadelphia'** dataset[4] (*dataset A*) encompasses the city's payments of the fiscal year 2017. It represents nearly \$4.2 billion in payments obtained from almost 60 city offices, departments, boards and committees. The dataset encompasses $n = 238,894$ payments comprised of 10 categorical and one numerical attribute. The encoding resulted in a total of $\mathcal{D} = 8,565$ one-hot encoded dimensions for each vendor payment record $x^i \in \mathcal{R}^{8,565}$.

The **'City of Chicago'** dataset[5] (*dataset B*) encompasses the city's vendor payments ranging from 1996 to 2020. The data is collected from the city's 'Vendor, Contract and Payment Search' and encompasses the procurement of goods and services. The dataset encompasses $n = 72,814$ payments comprised of 7 categorical and one numerical attribute. The encoding resulted in a total of $\mathcal{D} = 2,354$ one-hot encoded dimensions for eachvendor payment record $x^i \in \mathcal{R}^{2,354}$.

Cover Data: The cover dataset used to hide the secret data is derived from the 'ImageNet 2012' dataset. The 'ImageNet 2012' (Russakovsky et al. 2015) dataset[6] (*ImageNet*) is a publicly available datasets commonly used to evaluate machine learning models. The original dataset contains a total of 1.2 million RGB images organized in non-overlapping 1,000 subcategories. Each category corresponds to a real-world concept, such as 'llama', 'guitar', or 'pillow'. We use a subset of the entire dataset referred to as the 'Tiny ImageNet' dataset[7]. The dataset consists of 200 ImageNet subcategories. Thereby, each category encompasses 500 images resulting in a total of 100k RGB images. To demonstrate the generalization of the proposed method, we crop from each image a random image patch of size 224 $\times$ 224 pixels per channel and resize it to 256 $\times$ 256 pixels.

Architectural Setup: The steganographic process as shown in Fig. 2 and described in (Baluja 2017) is trained end-to-end comprised of the *preparation* network P_θ, the *hiding* network H_ϕ, and the *reveal* network R_γ. Each network applies a series 2D convolutions (LeCun, Bengio et al. 1995) of different filter sizes applied onto the input images (architectural details are described in the appendix). The convolutions are followed by non-linear $ReLU$ activation

[4]https://www.phila.gov

[5]https://data.cityofchicago.org

[6]http://image-net.org/download-images

[7]https://tiny-imagenet.herokuapp.com

Table 1: Network training losses ($\mathcal{L}^{ALL}_{\theta,\phi,\gamma}$, $\mathcal{L}^{cov}_{\theta,\phi}$, $\mathcal{L}^{sec}_{\theta,\phi,\gamma}$), PSNR, SSIM, and BACC obtained on both city payment datasets when using three channel $C = 3$ **RGB** ($H \times W \times 3$) cover images.

Dataset	BPP	α	β	$\mathcal{L}^{ALL}_{\theta,\phi,\gamma}$	$\mathcal{L}^{cov}_{\theta,\phi}$	$\mathcal{L}^{sec}_{\theta,\phi,\gamma}$	PSNR	SSIM	BACC
A	0.0436	0.2	1.0	0.63 ± 0.01	0.43 ± 0.06	0.54 ± 0.01	43.91 ± 0.64	0.997 ± 0.001	0.999 ± 0.001
		0.5	1.0	0.74 ± 0.03	0.40 ± 0.04	0.54 ± 0.04	44.23 ± 0.50	0.997 ± 0.001	0.998 ± 0.003
		0.8	1.0	0.81 ± 0.03	0.34 ± 0.03	0.55 ± 0.01	44.97 ± 0.43	0.998 ± 0.001	0.999 ± 0.001
		1.0	1.0	0.92 ± 0.06	0.38 ± 0.05	0.54 ± 0.02	44.42 ± 0.65	0.997 ± 0.001	0.998 ± 0.002
B	0.0120	0.2	1.0	0.67 ± 0.07	0.79 ± 0.03	0.51 ± 0.00	41.38 ± 0.32	0.998 ± 0.002	0.999 ± 0.001
		0.5	1.0	0.77 ± 0.01	0.52 ± 0.02	0.51 ± 0.01	42.94 ± 0.23	0.996 ± 0.001	0.998 ± 0.001
		0.8	1.0	0.87 ± 0.05	0.46 ± 0.07	0.51 ± 0.00	43.51 ± 0.65	0.995 ± 0.003	0.998 ± 0.002
		1.0	1.0	0.99 ± 0.04	0.47 ± 0.04	0.52 ± 0.01	43.35 ± 0.37	0.997 ± 0.001	0.997 ± 0.003

Values of the distinct loss functions $\mathcal{L}$ are multiplied by 10^4, variances originate from parameter initialization using four distinct random seeds.

Table 2: Network training losses ($\mathcal{L}^{ALL}_{\theta,\phi,\gamma}$, $\mathcal{L}^{cov}_{\theta,\phi}$, $\mathcal{L}^{sec}_{\theta,\phi,\gamma}$), PSNR, SSIM, and BACC obtained on both city payment datasets when using single channel $C = 1$ **grayscale** ($H \times W \times 1$) cover images.

Dataset	BPP	α	β	$\mathcal{L}^{ALL}_{\theta,\phi,\gamma}$	$\mathcal{L}^{cov}_{\theta,\phi}$	$\mathcal{L}^{sec}_{\theta,\phi,\gamma}$	PSNR	SSIM	BACC
A	0.1307	0.5	1.0	0.29 ± 0.00	0.01 ± 0.00	0.54 ± 0.01	60.93 ± 1.58	0.999 ± 0.001	0.997 ± 0.002
B	0.0359	0.5	1.0	0.87 ± 0.04	0.64 ± 0.07	0.55 ± 0.02	42.27 ± 0.49	0.989 ± 0.003	0.977 ± 0.002

Values of the distinct loss functions $\mathcal{L}$ are multiplied by 10^4, variances originate from parameter initialization using four distinct random seeds.

functions (Nair and Hinton 2010).

Training Setup: The networks are trained on secret images I^{sec} derived from the entire population of payments in each dataset. Throughout the training, the secret images are paired with randomly drawn cover images I^{cov} of the ImageNet dataset. We train with a mini-batch size of $m = 6$ secret-cover image pairs for a max. $\tau = 800$k iterations and apply early stopping once the combined loss defined in Eq. 3 converges. Thereby, the network parameters θ, ϕ, and γ are optimized with a constant learning rate of $\eta = 10^{-5}$ using Adam optimization (Kingma and Ba 2014), setting $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We sweep the weight factor of the container image loss $\mathcal{L}^{cov}_{\theta,\phi}$ through $\alpha \in [0.2, 1.0]$ to determine an optimal hyperparameter setup.

Experimental Results

In this section, we quantitatively and qualitatively assess the data leakage capability of the introduced steganographic process. The quantitative evaluation is conducted according to: (i) model *secrecy*, the difficulty of detecting the sensitive accounting data hidden in each image; (ii) model *accuracy*, the extent to which the secret information can be revealed correctly from the container image.

Quantitative Evaluation: We evaluate the model secrecy using the *Peak Signal-to-Noise Ratio* (PSNR) ratio as a proxy. The PSNR measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation, as defined by:

$$\text{PSNR} = 10 \cdot \log_{10} \frac{\max |I^{cov}|}{\mathcal{L}^{cov}_{\theta,\phi}(I^{cov}, I^{con})}, \tag{4}$$

where $\max |I^{cov}|$ denotes the maximum absolute possible value of the cover image. In the absence of noise, the cover image I^{cov} and the container image I^{con} are identical, and thus the $\mathcal{L}^{cov}_{\theta,\phi}$ is zero. In this case, the PSNR is infinite (Salomon 2004). It is assumed that acceptable values for wireless transmission quality loss are considered to be about 20 dB to 25 dB (Li and Cai 2007). In addition, we measure the *Structural Similarity Index Measure (SSIM)* (Wang et al. 2004). The SSIM defines a perception-based model considering image degradation as a perceived change in structural information. We quantify the visibility of differences between the distorted container image I^{con} and the cover image I^{cov}, as defined by:

$$\text{SSIM} = \frac{(2\mu_{cov}\mu_{con} + c_1)(2\sigma_{cov,con} + c_2)}{(\mu^2_{cov} + \mu^2_{con} + c_1)(\sigma^2_{cov} + \sigma^2_{con} + c_2)}, \tag{5}$$

where μ_{cov} and μ_{con} denotes the average pixel value of I^{cov} and I^{con} respectively, σ_{cov} and σ_{con} their variance, and $\sigma_{cov,con}$ the co-variance of the pixel values in both images. The SSIM maximum possible value of 1 indicates that the two images are structurally similar, while a value of 0 indicates no structural similarity.
We measure the model accuracy using *Bit Accuracy (ACC)*, which denotes the fraction of identical active bits between the secret image and revealed image, as defined by:

$$\text{BACC} = 1 - \frac{\sum_{x=0}^{H-1} \sum_{y=0}^{W-1} \mathbb{1}_{[\| I^{sec}_{x,y} - M \circ I^{rev}_{x,y} \| \leq \delta]}}{\sum_{x=0}^{H-1} \sum_{y=0}^{W-1} \mathbb{1}_{[I^{sec}_{x,y} > 0]}}, \tag{6}$$

where $\mathbb{1}$ denotes the indicator function and $M = \max(I^{rev}, k)$ denotes a binary pixel mask corresponding to the k most active pixel values in I^{rev}. We set k to the number of active binary pixels in I^{sec}. The operator $\circ$ denotes the element-wise multiplication, and δ denotes the dissimilarity threshold of the most active pixel values. We set to $\delta = 0.001$ in all our experiments.

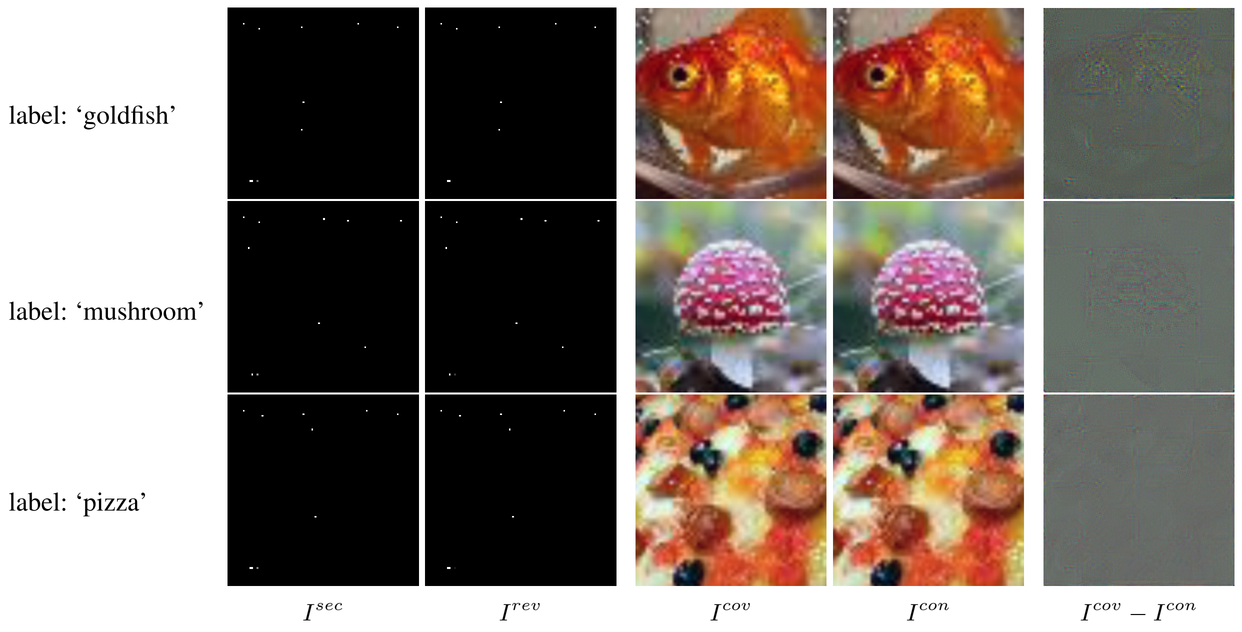

Figure 3: Exemplary data hiding results of a deep steganographic process model trained for $\tau = 800k$ iterations using RGB cover images (left to right): original secret, revealed secret, original cover, and created container image, as well as the pixel-wise residual errors of the cover and the container image.

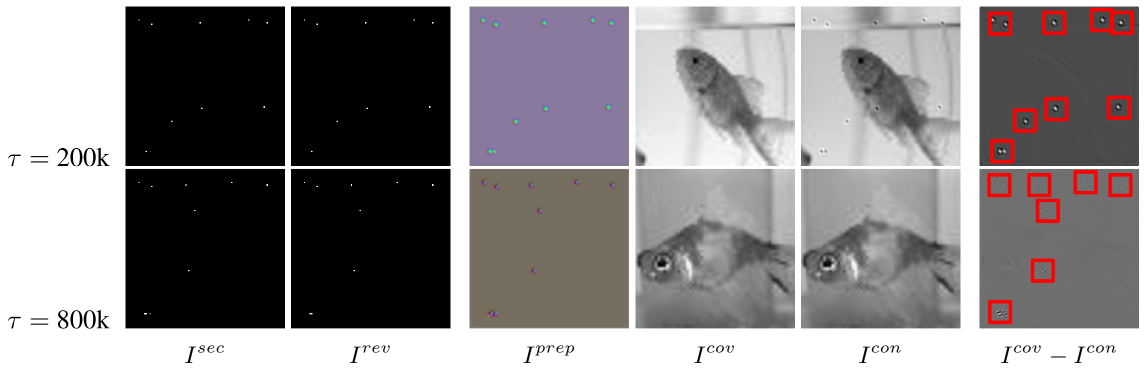

Figure 4: Exemplary data hiding results using grayscale cover images (left to right): original secret, revealed secret, prepared secret, original cover, and created container image, as well as the pixel-wise residual errors of the cover and the container image.

We analyze the performance of the introduced process in two setups: In the first setup, we use the original three-channel RGB images of the ImageNet dataset to hide the secret information. In the second setup, we reduced the capacity of the cover images by converting the dataset of cover images to single-channel grayscale images. We measure capacity as *Bits Per Pixel (BPP)* (Zhu et al. 2018), which denotes the number of encoded secret bits in the encoded image, defined by $\mathcal{D}/(H \times W \times C)$. The trained models are evaluated on randomly drawn combinations of 5k secret-cover image pairs. Tables 1 and 4 show the quantitative results of both setups. It can be observed that in the RGB cover image setup different α parameterizations yield high a secrecy and accuracy of the secret information. Similar results are obtained for the grayscale cover image setup, indicating that the grayscale images provide sufficient capacity to hide the data to the unaided human eye.

Qualitative Evaluation: Figure 3 shows three exemplary data hiding results using RGB cover images. The reconstructed container images I^{con} appear almost identical to the original cover images I^{cov} and encode all the information necessary to reconstruct the original payment information. The rightmost column shows the pixel-wise residual error between I^{cov} and I^{con}. It can be observed that the secret information is encoded across the pixel intensities of I^{con} and cannot easily be obtained. This learned obfuscation is of high relevance in a scenario when the original cover image becomes accessible to the auditor or fraud examiner. Figure 4 shows two exemplary data hiding results of the grayscale cover image setup. It can be observed that, due to the limited capacity of the cover, the secret information remains partially visible in the pixel-wise residual error even when training the model for $\tau = 800k$ iterations.

Conclusion

In this work, we conducted an analysis of deep steganography as a future challenge to the (internal) audit and fraud examination practice. We introduced a 'threat model' to leak sensitive information recorded in ERP systems. We also provided initial evidence that deep steganographic techniques can be maliciously misused to hide such information in unobtrusive 'everyday' images. In summary, we believe that such an attack vector pose a substantial challenge for CAATs used by auditors nowadays and the near future.

References

Ballet, V.; Renard, X.; Aigrain, J.; Laugel, T.; Frossard, P.; and Detyniecki, M. 2019. Imperceptible Adversarial Attacks on Tabular Data. *arXiv preprint:1911.03274* .

Baluja, S. 2017. Hiding Images in Plain Sight: Deep Steganography. In *Advances in Neural Information Processing Systems*, 2069–2079.

Chae, J. J.; and Manjunath, B. 1999. Data Hiding in Video. In *Proceedings 1999 International Conference on Image Processing*, volume 1, 311–315. IEEE.

Chandramouli, R.; and Memon, N. 2001. Analysis of LSB Based Image Steganography Techniques. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, volume 3, 1019–1022. IEEE.

Cheddad, A.; Condell, J.; Curran, K.; and Mc Kevitt, P. 2010. Digital Image Steganography: Survey and Analysis of Current Methods. *Signal Processing* 90(3): 727–752.

El_Rahman, S. A. 2018. A Comparative Analysis of Image Steganography based on DCT Algorithm and Steganography Tool to Hide Nuclear Reactors Confidential Information. *Computers & Electrical Engineering* 70: 380–399.

Eltahir, M. E.; Kiah, L. M.; Zaidan, B. B.; and Zaidan, A. A. 2009. High Rate Video Streaming Steganography. In *2009 International Conference on Information Management and Engineering*, 550–553. IEEE.

Fridrich, J.; Goljan, M.; and Du, R. 2001. Detecting LSB Steganography in Color, and Grayscale Images. *IEEE Multimedia* 8(4): 22–28.

Garner, B. 2014. *Black's Law Dictionary*. Thomson Reuters.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and Harnessing Adversarial Examples. *arXiv preprint:1412.6572* .

Hayes, J.; and Danezis, G. 2017. Generating Steganographic Images via Adversarial Training. In *Advances in Neural Information Processing Systems*, 1954–1963.

Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313(5786): 504–507.

Holub, V.; and Fridrich, J. 2012. Designing Steganographic Distortion using Directional Filters. In *2012 IEEE International workshop on information forensics and security (WIFS)*, 234–239. IEEE.

Holub, V.; Fridrich, J.; and Denemark, T. 2014. Universal Distortion Function for Steganography in an Arbitrary Domain. *EURASIP J. on Information Security* 2014(1): 1.

Husien, S.; and Badi, H. 2015. Artificial Neural Network for Steganography. *Neural Computing and Applications* 26(1): 111–116.

Kaur, B.; Kaur, A.; and Singh, J. 2011. Steganographic Approach for Hiding Image in DCT Domain. *International Journal of Advances in Engineering & Technology* 1(3): 72.

Kawaguchi, E.; and Eason, R. O. 1999. Principles and applications of BPCS steganography. In *Multimedia Systems and Applications*, volume 3528, 464–473. International Society for Optics and Photonics.

Kingma, D. P.; and Ba, J. 2014. ADAM: A Method for Stochastic Optimization. *arXiv preprint:1412.6980* .

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, 1097–1105.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep Learning. *Nature* 521(7553): 436.

LeCun, Y.; Bengio, Y.; et al. 1995. Convolutional Networks for Images, Speech, and Time Series. *The handbook of brain theory and neural networks* 3361(10): 1995.

Lerch-Hostalot, D.; and Megías, D. 2016. Unsupervised Steganalysis Based on Artificial Training Sets. *Engineering Applications of Artificial Intelligence* 50: 45–59.

Li, X.; and Cai, J. 2007. Robust Transmission of JPEG2000 Encoded Images over Packet Loss Channels. In *2007 IEEE International Conference on Multimedia and Expo*, 947–950. IEEE.

Long, M.; and Li, F. 2018. A Formula Adaptive Pixel Pair Matching Steganography Algorithm. *Advances in Multimedia: 2018* .

Luo, W.; Huang, F.; and Huang, J. 2010. Edge Adaptive Image Steganography Based on LSB Matching Revisited. *IEEE Transactions on Information Forensics and Security* 5(2): 201–214.

Mielikainen, J. 2006. LSB Matching Revisited. *IEEE Signal Processing Letters* 13(5): 285–287.

Muhammad, K.; Sajjad, M.; Mehmood, I.; Rho, S.; and Baik, S. W. 2018. Image Steganography using Uncorrelated Color Space and its Application for Security of Visual Contents in Online Social Networks. *Future Generation Computer Systems* 86: 951–960.

Nag, A.; Biswas, S.; Sarkar, D.; and Sarkar, P. P. 2011. A Novel Technique for Image Steganography Based on DWT and Huffman Encoding. *International Journal of Computer Science and Security,(IJCSS)* 4(6): 497–610.

Nair, V.; and Hinton, G. E. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 807–814.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical Black-Box Attacks Against Machine Learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 506–519. ACM.

Pevnỳ, T.; Filler, T.; and Bas, P. 2010. Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. In *International Workshop on Information Hiding*, 161–177. Springer.

Pibre, L.; Jérôme, P.; Ienco, D.; and Chaumont, M. 2015. Deep Learning for Steganalysis is Better than a Rich Model with an Ensemble Classifier, and is Natively Robust to the Cover Source-Mismatch. *arXiv preprint:1511.04855* .

Pibre, L.; Pasquet, J.; Ienco, D.; and Chaumont, M. 2016. Deep Learning is a good Steganalysis Tool when Embedding Key is Reused for Different Images, even if there is a Cover Source Mismatch. *Electronic Imaging* 2016(8): 1–11.

Qazanfari, K.; and Safabakhsh, R. 2017. An Improvement on LSB Matching and LSB Matching Revisited Steganography Methods. *arXiv preprint:1709.06727* .

Qian, Y.; Dong, J.; Wang, W.; and Tan, T. 2015. Deep Learning for Steganalysis via Convolutional Neural Networks. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, 94090J. Int. Society for Optics and Photonics.

Ramani, G.; Prasad, E.; Varadarajan, S.; SVUCE, T.; and Jntuce, K. 2007. Steganography using BPCS to the Integer Wavelet Transformed Image. *IJCSNS* 7(7): 293–302.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3): 211–252.

Sadek, M. M.; Khalifa, A. S.; and Mostafa, M. G. 2015. Video Steganography: A Comprehensive Review. *Multimedia Tools and Applications* 74(17): 7063–7094.

Salomon, D. 2004. *Data Compression: The Complete Reference*. Springer Science & Business Media.

Saon, G.; Kuo, H.-K. J.; Rennie, S.; and Picheny, M. 2015. The IBM 2015 English Conversational Telephone Speech Recognition System. *arXiv preprint:1505.05899* .

Schreyer, M.; Sattarov, T.; Borth, D.; Dengel, A.; and Reimer, B. 2017. Detection of Anomalies in Large Scale Accounting Data using Deep Autoencoder Networks. *arXiv preprint:1709.05254* .

Schreyer, M.; Sattarov, T.; Reimer, B.; and Borth, D. 2019a. Adversarial Learning of Deepfakes in Accounting. *NeurIPS'19 Workshop on Robust AI in Financial Services* .

Schreyer, M.; Sattarov, T.; Schulze, C.; Reimer, B.; and Borth, D. 2019b. Detection of accounting anomalies in the latent space using adversarial autoencoder neural networks. *KDD'19 Workshop on Anomaly Detection in Finance* .

Shabtai, A.; Elovici, Y.; and Rokach, L. 2012. *A Survey of Data Leakage Detection and Prevention Solutions*. Springer Science & Business Media.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the Game of Go Without Human Knowledge. *Nature* 550(7676): 354.

Spaulding, J.; Noda, H.; Shirazi, M. N.; and Kawaguchi, E. 2002. BPCS Steganography using EZW Lossy Compressed Images. *Pattern Recognition Letters* 23(13): 1579–1587.

Sun, T. 2019. Applying Deep Learning to Audit Procedures: An Illustrative Framework. *Accounting Horizons* 33(3): 89–109.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, 3104–3112.

Swain, G. 2018. Digital Image Steganography using Eight-directional PVD against RS Analysis and PDH Analysis. *Advances in Multimedia* 2018.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing Properties of Neural Networks. *arXiv preprint:1312.6199* .

Tamimi, A. A.; Abdalla, A. M.; and Al-Allaf, O. 2013. Hiding an Image inside another Image using Variable-Rate Steganography. *International Journal of Advanced Computer Science and Applications (IJACSA)* 4(10).

Tan, S.; and Li, B. 2014. Stacked Convolutional Autoencoders for Steganalysis of Digital Images. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, 1–4. IEEE.

The Economist. 2017. The World's Most Valuable Resource is No Longer Oil, but Data. *New York, NY, USA* .

Thomee, B.; Shamma, D. A.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; and Li, L.-J. 2016. YFCC100M: The New Data in Multimedia Research. *Communications of the ACM* 59(2): 64–73.

Viji, G.; and Balamurugan, J. 2011. LSB Steganography in Color and Grayscale Images without using the Transformation. *Bonfring International Journal of Advances in Image Processing* 1: 11–14.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13(4): 600–612.

Xu, G.; Wu, H.-Z.; and Shi, Y.-Q. 2016. Structural Design of Convolutional Neural Networks for Steganalysis. *IEEE Signal Processing Letters* 23(5): 708–712.

Zhang, S.; Yang, L.; Xu, X.; and Gao, T. 2018. Secure Steganography in JPEG Images Based on Histogram Modification and Hyper Chaotic System. *International Journal of Digital Crime and Forensics (IJDCF)* 10(1): 40–53.

Zhu, J.; Kaplan, R.; Johnson, J.; and Fei-Fei, L. 2018. Hidden: Hiding Data with Deep Networks. In *Proc. of the European Conference on Computer Vision (ECCV)*, 657–672.

Appendix A: Architectural Details

Table 3: Architectural details of the distinct neural networks that constitute the steganography process, namely preparation network P_θ, hiding network H_ϕ, and reveal network R_γ.

$I^{sec} \in \mathbb{R}^{H \times W \times 1}$		
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Cat^{150}_{H\times W}$		
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
	$Conv^{1}_{4\times4}$ $ReLU$	
$Cat^{3}_{H\times W}$		
$I^{pre} \in \mathbb{R}^{H \times W \times 3}$		

Preparation Net P_θ

$I^{pre}, I^{cov} \in \mathbb{R}^{H \times W \times 3}$		
$Cat^{6}_{H\times W}$		
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Cat^{150}_{H\times W}$		
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Conv^{1}_{3\times3}$ $ReLU$	$Conv^{1}_{4\times4}$ $ReLU$	$Conv^{1}_{5\times5}$ $ReLU$
$Conv^{3}_{1\times1}(Cat^{150}_{H\times W})$		
$I^{con} \in \mathbb{R}^{H \times W \times 3}$		

Hiding Net H_ϕ

$I^{con} \in \mathbb{R}^{H \times W \times 3}$		
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
$Cat^{150}_{H\times W}$		
$Conv^{50}_{3\times3}$ $ReLU$	$Conv^{50}_{4\times4}$ $ReLU$	$Conv^{50}_{5\times5}$ $ReLU$
	$Conv^{1}_{4\times4}$ $ReLU$	
$Conv^{1}_{2\times2}(Cat^{3}_{H\times W})$		
$I^{rev} \in \mathbb{R}^{H \times W \times 1}$		

Reveal Net R_γ

Each network applies a series of convolutions and non-linear activation's onto their respective input images. Each network applies a series of 2D convolutions $Conv^{c}_{k\times k}$ (LeCun, Bengio et al. 1995) of different filter sizes applied onto the input images, where c denotes the number of output channels and $k \times k$ denotes the filter kernel size. The convolutions are followed by non-linear $ReLU$ activation's (Nair and Hinton 2010).

Appendix B: Additional Experimental Results

Table 4: Network training losses ($\mathcal{L}^{ALL}_{\theta,\phi,\gamma}$, $\mathcal{L}^{cov}_{\theta,\phi}$, $\mathcal{L}^{sec}_{\theta,\phi,\gamma}$), PSNR, SSIM, and BACC obtained on both city payment datasets when using single channel $C = 1$ **grayscale** ($H \times W \times 1$) cover images.

Dataset	BPP	α	β	$\mathcal{L}^{ALL}_{\theta,\phi,\gamma}$	$\mathcal{L}^{cov}_{\theta,\phi}$	$\mathcal{L}^{sec}_{\theta,\phi,\gamma}$	PSNR	SSIM	BACC
A	0.1307	0.2	1.0	0.11 ± 0.01	0.01 ± 0.00	0.54 ± 0.01	64.84 ± 0.49	0.999 ± 0.001	0.997 ± 0.003
		0.5	1.0	0.29 ± 0.00	0.01 ± 0.00	0.54 ± 0.01	60.93 ± 1.58	0.999 ± 0.001	0.997 ± 0.002
		0.8	1.0	0.44 ± 0.00	0.01 ± 0.00	0.54 ± 0.00	60.98 ± 0.13	0.999 ± 0.001	0.999 ± 0.006
		1.0	1.0	0.55 ± 0.01	0.01 ± 0.00	0.54 ± 0.01	60.15 ± 1.79	0.999 ± 0.001	0.999 ± 0.004
B	0.0359	0.2	1.0	0.75 ± 0.02	0.82 ± 0.08	0.53 ± 0.01	39.98 ± 0.33	0.985 ± 0.001	0.983 ± 0.003
		0.5	1.0	0.87 ± 0.04	0.64 ± 0.07	0.55 ± 0.02	42.27 ± 0.49	0.989 ± 0.001	0.977 ± 0.002
		0.8	1.0	0.98 ± 0.05	0.52 ± 0.06	0.58 ± 0.01	43.19 ± 0.53	0.990 ± 0.001	0.967 ± 0.006
		1.0	1.0	0.99 ± 0.13	0.41 ± 0.14	0.58 ± 0.01	44.28 ± 1.34	0.991 ± 0.001	0.966 ± 0.005

Values of the distinct loss functions $\mathcal{L}$ are multiplied by 10^4, variances originate from parameter initialization using four distinct random seeds.