Distribution-Preserving Steganography Based on Text-to-Speech Generative Models

Kejiang Chen, Hang Zhou, Hanqing Zhao, Dongdong Chen, Weiming Zhang, Nenghai Yu

***Abstract*—Steganography is the art and science of hiding secret messages in public communication so that the presence of the secret messages cannot be detected. There are two distribution-preserving steganographic frameworks, one is sampling-based and the other is compression-based. The former requires a perfect sampler which yields data following the same distribution, and the latter needs explicit distribution of generative objects. However, these two conditions are too strict even unrealistic in the traditional data environment, e.g. the distribution of natural images is hard to seize. Fortunately, generative models bring new vitality to distribution-preserving steganography, which can serve as the perfect sampler or provide the explicit distribution of generative media. Take text-to-speech generation task as an example, we propose distribution-preserving steganography based on WaveGlow and WaveNet, which corresponds to the former two categories. Steganalysis experiments and theoretical analysis are conducted to demonstrate that the proposed methods can preserve the distribution.**

***Index Terms*—Steganography, generative media, arithmetic coding, provable security**

I. Introduction

STEGANOGRAPHY is the art and science of communicating in such a way that the presence of a message cannot be detected, which can be applied in various applications, such as anonymous communication, covert communication, etc. Cachin [1] firstly formalized an information-theoretic model for steganography in 1998, where a relative entropy function is employed as a basic measure of steganographic security for the concealment system. The security of a steganographic system can be quantified in terms of the relative entropy $D(P_c \parallel P_s)$ between the distributions of cover P_c and stego P_s, which yields bound on the detection capability of any adversary. $D(P_c \parallel P_s) = 0$ means perfectly secure, which is also regarded as distribution-preserving. From another perspective, Hopper *et al.* [2] formalized a perfectly secure system based on the computational complexity, which is immigrated from cryptography. The perfect security is defined as a polynomial-time distinguisher that cannot distinguish the cover and stego. Namely, the distribution of the cover and stego is indistinguishable.

With the definition of steganographic security, many schemes related to distribution-preserving steganography are proposed and can be divided into two : compression-based and sampling-based. As for the compression-based stegosystem, Anderson *et al.* [3] observed that cover can be compressed to generate the secret message, and for message embedding, we can decompress it into stego. Le *et al.* [4] constructed the distribution-preserving steganography called $\mathcal{P}$-code based on the arithmetic coding and it assumes that both of the sender and receiver know the distribution of cover. Sallee [5] designed a compression-based stegosystem for JPEG images that assumes the AC coefficients in JPEG images following generalized Cauthy distribution, and the receiver can estimate the distribution as well.

As for sampling-based stegosystem, Cachin [1] proposed using rejection-sampling to generate the stego that looks like the cover. In detail, the stegosystem samples a document from the cover distribution until the sampled document whose hash value equals to the XOR between the message and k, where k is a session secret key shared by both of the two parties. Hopper [2] improved Cachin's method and generalized it to be applicable to any distribution, which assumed it has sufficient capacity (entropy) and can be sampled perfectly based on the prior history. Von Ahn *et al.* [6] created public-key provably secure stegosystem and chosen-stegotext attacks. Lysyanskaya *et al.* [7] analyzed the problem of imperfect sampling by weakening the assumption that the cover distribution is modeled as a stateful Markov process. Zhu *et al.* [8] provided a more general construction of secure steganography with one-way permutation and unbiased sampler.

However, neither the compression-based stegosystem nor the sampling-based stegosystem, is effective and even feasible in the non-synthetic data environment. Compression-based schemes need to know the exact explicit distribution of cover, the constraint of which is too strict. The complexity and dimensionality of covers formed by digital media objects, such as natural audios, images and videos, will prevent us from determining a complete distribution P_c of cover. As for sampling-based systems, the difficulty is that the perfect sampler is hard to obtain, and the embedding capacity of the existing scheme is rather low [2].

Fortunately, generative models bring new vitality to distribution-preserving steganography. Generative models describe how generative media are generated, in terms of generating data whose distribution is approaching that of the training data. Prominent models include variational auto-encoders (VAE) [9], [10], generative adversarial networks (GAN) [11], auto-regressive models [12]–[14] and normalizing flows [15]. VAEs maximize a variational lower bound on the log-likelihood of the data. GANs employ an adversarial framework to train a generative model that mimics the true transition model, auto-regressive models and normalizing flows

This work was supported in part by the Natural Science Foundation of China under Grant U1636201, 62002334 and 62072421, and by Anhui Science Foundation of China under Grant 2008085QF296.

All the authors are with CAS Key Laboratory of Electro-magnetic Space Information, University of Science and Technology of China, Hefei 230026, China.

Corresponding author: Weiming Zhang (Email: zhangwm@ustc.edu.cn)

train with maximum likelihood, and avoid approximations by choosing a tractable parameterization of probability density. VAE, GAN, and flow-based generative models can generate vivid objects from latent variables, which follow a prior distribution, e.g. Gaussian distribution, and auto-regressive models can give the explicit distribution of generative objects. The generative models have been utilized for steganography [16]–[21], but most of them did not focus on distribution-preserving steganography.

In this paper, we design two distribution-preserving steganographic methods based on generative models, which own the advantages of efficiency, practicality and high embedding capacity. Inspired by the sampling-based stegosystem, we introduce a reversible flow-based generative model as a black-box sampler. In detail, the encrypted message is mapped into the latent codes that follows the Gaussian distribution, and then the latent codes is fed into the generative model, yielding the generative data. Once the generative model is well-trained and fixed, the generative model can be seen generating data following the same distribution, which meets the requirements of the black-box sampler. The receiver shares the same generative models with the sender, and the reversibility of generative models guarantees that the message can be extracted correctly from the generated data. Here, the text-to-speech generative model *WaveGlow* serves as the instance to verify the practical security.

For the compression-based stegosystem, auto-regressive models are adopted for their explicit distributions of the generative data. According to the duality between data compression and message embedding, the adaptive arithmetic coding is integrated into the data generation process, which decompresses the encrypted message into the generated samples following the given distribution predicted by the auto-regressive models. With the same well-trained generative model, the receivers can compress the generated samples to extract the message using the process of compression. Here, the text-to-speech generative model *WaveNet* is adopted as the auto-regressive model.

It is worth mentioning that choosing text-to-speech generative models rather than image generative models for validating our distribution-preserving stegosystem is carefully thought out. Generally, the quality of the generative speech is acceptable, while the generative image may be blurred and unstable. What's more, the semantic of speech is robust, which means that the receiver can get the text of speech through speech recognition. That is to say, there is no need to send the text as the side information every time for generating the corresponding audio. Additionally, the codes and the generated samples are available[1].

The main contributions of this work are three-fold:

- Based on the reversible generative model, WaveGlow, we design the sampling-based distribution-preserving steganography with a high capacity and efficiency of message extraction.
- Based on the auto-regressive generative model, WaveNet, we design a compression-based distribution-preserving stegosystem using the adaptive arithmetic coding. The theoretical analysis is given to prove our proposed method is distribution-preserving.
- Experiments validated the practical security of the proposed two stegosystems by defending against the state-of-the-art staganalysis methods.

[1]http://home.ustc.edu.cn/~chenkj/audio/audio.html

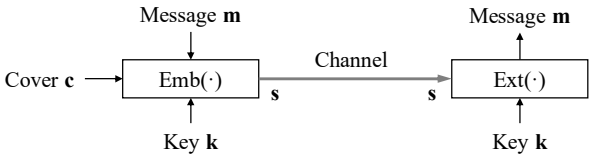


Fig. 1. The diagram of steganographic communication.

The rest of this paper is organized as follows. We review the related work in Section II, and the proposed sampling-based stegosystem and compression-based stegosystem on generative models are elaborated in Section III and Section IV, respectively. The experiments are presented in Section V. Conclusion and future work are given in Section VI.

II. Related Work

A. The Prisoners' Problem

In order to improve the readability of this paper, the prisoners' problem [22] formulated by Simmons is introduced first, which is a popular formulation of the steganography problem. Alice and Bob are imprisoned in separate cells and want to hatch an escape plan. They are allowed to communicate but their communication is monitored by warden Eve. If Eve finds out that the prisoners are secretly exchanging the messages, she will cut the communication channel and throw them into solitary confinement.

B. The Diagram of Steganography

According to the prisoners' problem, the diagram of the steganography is depicted in Fig. 1. A steganographic scheme can be regarded as a pair of embedding and extraction functions Emb() and Ext() for Alice and Bob, respectively [23].

$$\mathrm{Emb}(\mathbf{c}, \mathbf{k}, \mathbf{m}) = \mathbf{s}, \tag{1}$$

$$\mathrm{Ext}(\mathbf{s}) = \mathbf{m}, \tag{2}$$

where $\mathbf{c}, \mathbf{k}, \mathbf{m}, \mathbf{s}$ are cover object, secret key, message and stego object, respectively. Eve judges the object $\mathbf{s}$ is innocent or not by all the possible knowledge except secret key according to Kerckhoffs' principle [24].

C. The Theoretical Definition of Steganographic Security

The information-theoretic definition of steganographic security is given by Cathin [1]. Assuming the cover is drawn from C with probability distribution P_c and the steganographic method will generate stego object which has the distribution P_s. The distance of two distributions can be measured using relative entropy:

$$D(P_\mathrm{c} \parallel P_\mathrm{s}) = \sum_{\mathbf{x} \in C} P_\mathrm{c}(x) \log \frac{P_\mathrm{c}(x)}{P_\mathrm{s}(x)}. \tag{3}$$

When $D(P_\mathrm{c} \parallel P_\mathrm{s}) = 0$, the stegosystem is distribution-preserving.

D. Existing Distribution-Preserving Methods

The sampling-based distribution-preserving methods can be briefly described in **Algorithm 1**. Given a mapping function $f_k(\cdot) : \mathcal{K} \times C \to \mathcal{R}$ with the secret key k and $\mathcal{R} = \{0,1\}^e$, the message embedding in the stegosystem is based on rejection sampling algorithm $\text{Sample}_f^C(k,b)$. The object is generated by sampling according to the given distribution C by oracle $\mathcal{O}^C$, such that an e-bit symbol b will be embedded in it. The hiding algorithm randomly chooses a sample s from them that satisfies $f_k(s) = b$. However, the perfect sampler for generating multimedia objects is hard to obtain in the traditional data environment, and the capacity of existing schemes that adopt documents, network protocol [25] as the carrier, is rather low, e.g. one document only carries one-bit message.

Algorithm 1 $Sample_f^C(k,b)$

Require: a key k, a value $b \in \{0,1\}^e$.
1: **repeat**
2: $s \leftarrow_R \mathcal{O}^C$.
3: **until** $f_k(s) = b$.
4: Return s.

Another category of distribution-preserving steganography is compression-based steganography. With a perfect compressor, the medium can be compressed into the bitstream following the uniform distribution whose information entropy is maximum. The data decompression process is translating the bitstream into the original medium, which is similar to message embedding. With the distribution of the medium data, the encrypted message can be decompressed into a medium through a perfect compressor, resulting from the duality between data compression and message embedding. Message can be extracted by compressing the generated media. Generally, source encoding can serve as the compressor. Based on arithmetic coding, Le [4] proposed $\mathcal{P}$-code for distribution-preserving steganography, assuming that both sides know the distribution of cover exactly. Sallee [5] designed the compression-based stegosystem for JPEG images by assuming that AC coefficients follow Generalized Cathin distribution, whose parameters can be calculated by both sides. However, Generalized Cathin distribution is not the true distribution of AC coefficients. The complexity and dimensionality of covers formed by digital media objects, such as natural audios, images and videos, will prevent us from determining a complete distribution P_c of cover, which implies Sallee's method cannot achieve perfect distribution-preserving steganography.

In summary, compression-based schemes need to know the exact distribution of cover, which is too difficult to capture the distribution of digital media objects. As for the sampling-based system, the perfect samplers are hard to obtain, and the capacities of the existing schemes are rather low. To this end, we introduce generative models into distribution-preserving steganography.

E. Text-to-Speech Generative Models

The generative model describes how media are generated. The generation process can be seen as random sampling from the probability distribution learned in the stage of training, as shown in Fig. 2. The generative models can be divided into two categories, implicit density probability distribution, and explicit density probability distribution. Specifically, VAEs, GANs and flow-based models belong to the first category, and auto-regressive models attribute to the second category. The former meets the requirement of the sampling-based stegosystem, and the latter can be adopted to develop compression-based stegosystems.

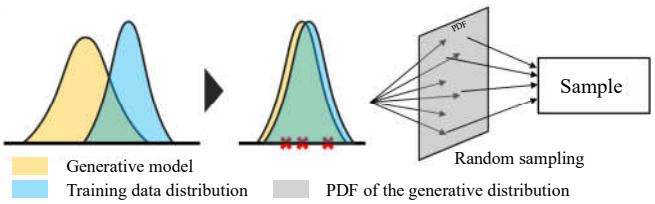


Fig. 2. The pipeline of sample generation using the generative model.

Text-To-Speech (TTS) has attracted a lot of attention in recent years and deep neural network based systems have become more and more popular for TTS, such as Tacotron 2 [26], Deep Voice 3 [27] and ClariNet [28], resulting from their satisfactory speech quality. These models usually first generate the mel-spectrogram from the input text and then synthesize speech from the mel-spectrogram using vocoders such as Griffin-Lim [29], WaveNet [14], WaveGlow [30].

In this paper, we propose two distribution-preserving steganographic methods based on text-to-speech generative models, WaveNet [14] and WaveGlow [30].

III. Distribution-preserving steganography based on WaveGlow

In this section, the distribution-preserving steganographic scheme based on the implicit generative model is proposed. VAEs and GANs have been adopted for steganography in previous works, however, an extra message extractor is needed for message embedding and the message cannot be 100% correctly extracted. Error codes can be utilized to alleviate the problem, but the capacity becomes lower due to the parity information. Consequently, we design a stegosystem on the flow-based generative model, WaveGlow, whose reversibility does favor to message extraction.

Fig. 3 presents the pipeline of the distribution-preserving method based on WaveGlow. In the upper part, we show the normal generation of audios by WaveGlow. The input text is first transferred into mel-spectrogram by spectrogram generation model (SPN). Here, Fastspeech [31] is adopted for its certainty in transformation, so that the receivers can reproduce the same mel-spectrogram. The generator randomly takes samples from a zero-mean spherical Gaussian, and then feeds the Gaussian sample and mel-spectrogram to WaveGlow to generate the cover audio.

For message embedding, the encrypted message is mapped into a Gaussian vector, and then fed to the WaveGlow along with the mel-spectrogram to yield the stego audio. For ensuring the receiver extracting the message correctly, the

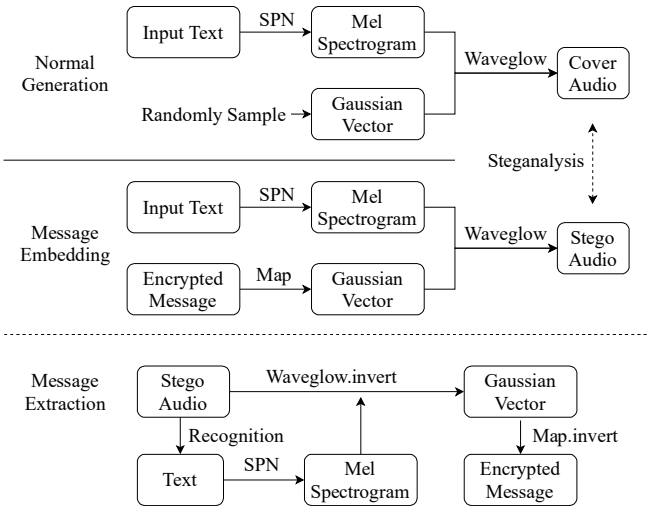


Fig. 3. The distribution-preserving steganographic scheme based on WaveGlow. The upper part presents the normal generation of audios, and the generated audios are regarded as cover audios. The bottom part shows that the encrypted message is mapped into the Gaussian vector and then the Gaussian vector and mel-spectrogram are fed into WaveGlow to generate the stego audio. As for message extraction, the stego audio is first recognized as text, which is exactly the same as the input text on the sender-side. Since the distribution mapping and WaveGlow are invertible, their inverse processes can be used to extract the message.

stego audio is required to be recognized the same as the input text. With the recognized text and SPN, the same mel-spectrogram can be obtained by the receiver. The message can be extracted by the inverse processes of WaveGlow generation and message mapping. Each module will be addressed in the next subsections.

A. Message Mapping

Generally, the message $\mathbf{m}$ is encrypted into the random binary bitstream by XOR with the secret key sequence $\mathbf{k}$:

$$\mathbf{m}' = \mathbf{m} \oplus \mathbf{k}. \tag{4}$$

It is known that if the secret key is random and the key is as long as the message, then the encrypted message is a uniform variable. Cryptographically secure pseudo-random number generator (CSPRNG) is used to generate the secret key sequence, which is computationally indistinguishable from true randomness. Therefore, the encrypted message $\mathbf{m}'$ can be regraded as following the uniform distribution. Then, the encrypted message $\mathbf{m}'$ is transferred into Gaussian latent codes $\boldsymbol{z}$ by the mapping module.

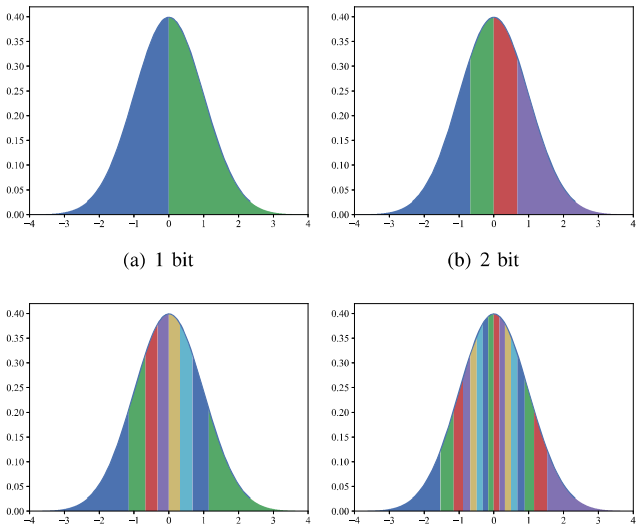
The mapping module maps variables from the uniform distribution to the Gaussian distribution, which can be well carried out by rejection sampling. Here, we define the payload p as the information that each dimension of latent codes carry. Given p-bit binary message, we can map it into the Gaussian latent codes using mapping function $\mathcal{M}(m, p)$:

$$z = \mathcal{M}(m, p) = RS\left(F^{-1}\left(\frac{m}{2^{p-1}}\right), F^{-1}\left(\frac{m+1}{2^{p-1}}\right)\right), \tag{5}$$

where $RS(x, y)$ is a rejection sampling function that will keep randomly sampling a value z from $(-\infty, \infty)$ until z drops into the interval (x, y). F^{-1} is the inverse of cumulative distribution function (CDF) F of Gaussian distribution. m is the encrypted message in p-ary form transferred from p-bit encrypted binary message $\mathbf{m}'$:

$$m = \sum_{i=0}^{p-1} 2^i \cdot \mathbf{m}'(i). \tag{6}$$

Fig. 4 gives the examples of the interval division of $p = 1, 2, 3, 4$, respectively. After the division is determined, rejection sampling is utilized to map the binary stream into the corresponding interval. Details of the mapping process are given in Algorithm 2.



(a) 1 bit (b) 2 bit (c) 3 bit (d) 4 bit

Fig. 4. The examples of division in the module of mapping.

Algorithm 2 Mapping function $\mathcal{M}$

Require: Payload p, encrypted message $\mathbf{m}'$ of length n.
Ensure: latent codes $\boldsymbol{z}$.

```
Divide (-∞,∞) into 2^p intervals according to the CDF of Gaussian distribution.
Make every p-bit message as one group and compose n/p message groups G.
for each item i in G do
    repeat
        Sample s from the Gaussian distribution.
    until s drops into the corresponding interval of item i.
    Append s to the latent codes z.
end for
```

B. Audio Generation

WaveGlow is a reversible generative model, constructed on affine coupling layers. It takes samples from a zero-mean spherical Gaussian with the same number of dimensions as the desired output, and puts those samples as well as mel-spectrogram through a series of invertible layers that transform the Gaussian distribution to one which has the desired

distribution. Specifically, WaveGlow models the distribution of audio samples conditioned on a mel-spectrogram.

$$z \sim \mathcal{N}(z; 0, \boldsymbol{I}) \tag{7}$$

$$\boldsymbol{x} = \boldsymbol{f}_0 \circ \boldsymbol{f}_1 \circ \ldots \boldsymbol{f}_k(\boldsymbol{z}, F_{mel}), \tag{8}$$

where $\boldsymbol{f}_i, i = 0, 1, ..., k$ is invertible affine coupling layer, and F_{mel} is the mel-spectrogram of the text. From audio to latent codes can be expressed as:

$$\boldsymbol{z} = \boldsymbol{f}_k^{-1} \circ \boldsymbol{f}_{k-1}^{-1} \circ \ldots \boldsymbol{f}_0^{-1}(\boldsymbol{x}, F_{mel}). \tag{9}$$

After mapping the message into the Gaussian latent codes $\boldsymbol{z}$, the mel-spectrogram F_{mel} and the latent codes $\boldsymbol{z}$ are fed into WaveGlow to generate the stego audio $\boldsymbol{y}$.

C. Discussion of the Security

It can be seen from Fig. 3, the difference between normal generation and message embedding is only the latent codes. The former randomly takes samples from the Gaussian distribution, while the latter take samples from the Gaussian distribution according to the encrypted message by rejection sampling. As analyzed before, the encrypted message can be regarded computationally indistinguishable random thanks to the CSPRNG. In this way, the latter can be also seen as randomly taking samples from the Gaussian distribution. Since the subsequent processes are the same, the distribution of cover audios and stego audios can be deemed as the same. In conclusion, the stegosystem based on WaveGlow is distribution-preserving.

IV. Distribution-preserving steganography based on WaveNet

Data hiding is the dual process of compression, and we present a comparison between data hiding and data compression in Fig. 5. The upper part presents the data compression and the second row presents the data hiding. Data decompression corresponds to message embedding and data compression corresponds to message extraction. Generally, decompressing bitstream into medium requires prior knowledge, such as the distribution of the medium. However, it is hard to obtain the distribution of natural media. Fortunately, the development of deep generative models brings us an opportunity, where the auto-regressive models present the explicit density probability distribution of the generated media, which may solve the aforementioned problems.

In our previous work [32], we built a secure system on auto-regressive model, PixelCNN [33]. However, the quality of the generated image is unsatisfying. Furthermore, the images always belong to the same category, which means the semantic of the image is single, and this behavior is suspicious.

To solve this, we design a stegosystem based on text-to-speech auto-regressive models, WaveNet, which can generate audios with abundant semantics. Besides, the semantics are naturally embedded in audios, which means the same text can be obtained by the receiver as that in the sender-side, so that they can reimplement the same generation process with the identical WaveNet model.

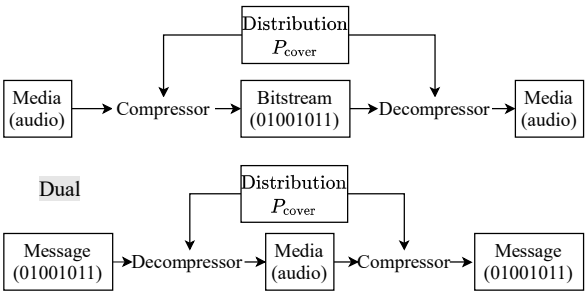


Fig. 5. The duality between data compression and data hiding. Data decompression corresponds to message embedding and data compression corresponds to message extraction.

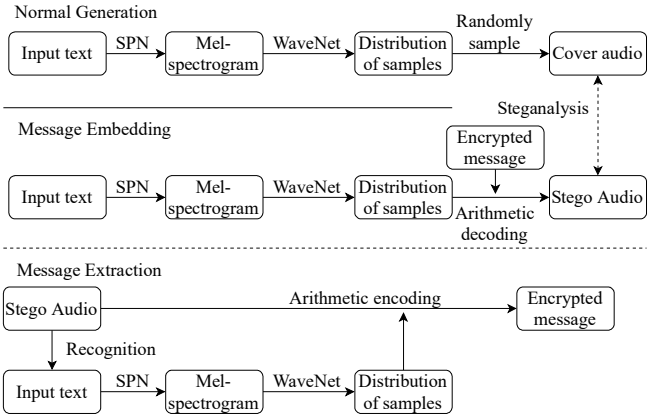


Fig. 6. The diagram of distribution-preserving steganography based on WaveNet and adaptive arithmetic coding.

WaveNet describes the joint distribution of an audio $\boldsymbol{x} = (x_1, \ldots, x_n)$ as a product of conditionals:

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_1, \ldots, x_{i-1}, \boldsymbol{h}), \tag{10}$$

where $\boldsymbol{h}$ can be additional global information, such as speaker ID and audio semantic. That is to say, we can obtain the exact probability distribution of every sample of the generated audio, which supports us to design the compression-based stegosystem.

The diagram of the compression-based stegosystem using WaveNet is shown in Fig. 6. The input text is transferred into the mel-spectrogram, and the mel-spectrogram as well as samples already generated are fed into the WaveNet to obtain the distribution of the next sample. The normal generation will randomly choose the sample value from the distribution. As for message embedding, with the distribution, the stegosystem embeds the encrypted message into samples using adaptive arithmetic decoding (AAD). The embedding process responds to the data decompression. Sharing the same SPN and WaveNet, the receiver can first recognize the stego audio into text and then generate the same distribution. With the same distribution and stego audio, the receiver can extract the message using adaptive arithmetic encoding (AAE), which responds to data compression. The details of message embedding and extraction using AAD and AAE will be elaborated below.

A. Message Embedding and Extraction

Given the distribution P_c of the generated audio by WaveNet, the process of embedding message corresponds to adaptive arithmetic decoding, and extraction corresponds to adaptive arithmetic encoding. $\mathcal{A} = \{a_1, a_2, ..., a_m\}$ is the alphabet of the generated audio sample values in a certain order with the probability $\mathcal{P} = \{P(a_1), P(a_2), ..., P(a_m)\}$. The cumulative probability of a symbol can be defined as

$$F(a_i) = \sum_{k=1}^{i} P(a_k). \tag{11}$$

Owning these notations, we start to introduce the process of message embedding and extraction.

1) Message embedding: Given the encrypted message $\mathbf{m}' = [m_1 m_2 m_3 ... m_L]$, it can be interpreted as a fraction q in the range $[0, 1)$ by prepending "0." to it:

$$m_1 m_2 m_3 ... m_L \rightarrow q = 0.m_1 m_2 m_3 ... m_L = \sum_{i=1}^{L} m_i \cdot 2^{-i}. \tag{12}$$

Following the adaptive arithmetic decoding algorithm, we start from the interval $[0, 1)$ and subdivide it into the subinterval $[l, h)$ according to the probability $\mathcal{P}$ of the symbols in $\mathcal{A}$, and then append the symbol a_j corresponding to the subinterval in which the dyadic fraction q lies into the stego $\mathbf{y}$:

$$\boldsymbol{y} = \boldsymbol{y} :: a_j, \tag{13}$$

where :: represents appending the subsequent symbol into the previous vector. Regularly, the probability $\mathcal{P}$ of symbols will be updated. Then calculate the subinterval $[l, h)$ according to the updated probability $\mathcal{P}$ by

$$h_k = h_{k-1} + (h_{k-1} - l_{k-1}) * F(a_j), \tag{14}$$

$$l_k = l_{k-1} + (h_{k-1} - l_{k-1}) * F(a_{j-1}), \tag{15}$$

where h_k and l_k are the bound of subinterval in the k-th step. Repeat the process until the fraction q satisfies the constraint:

$$\begin{cases} q + (0.5)^L \notin [l_k, h_k) \\ q - (0.5)^L \notin [l_k, h_k) \end{cases} \tag{16}$$

The constraint guarantees that the dyadic fraction q is the unique fraction of length L in the interval $[l_k, h_k)$, such that the message can be extracted correctly. The message length L and the probabilities $\mathcal{P}$ of symbol are shared with the receiver.

2) Message extraction: On the receiver-end, the interval $[l, h)$ starts from $[0, 1)$, and will be subdivided into subintervals of length proportional to the probabilities of the symbols. if k-th element y_j corresponds to the symbol a_j, update the subinterval as follows:

$$h_k = h_{k-1} + (h_{k-1} - l_{k-1}) * F(a_j), \tag{17}$$

$$l_k = l_{k-1} + (h_{k-1} - l_{k-1}) * F(a_{j-1}). \tag{18}$$

Repeat the process until the number of steps reaches the length of $\mathbf{y}$. Finally, find the fraction $q = \sum_{i=1}^{n} m_i 2^{-i}$ satisfying $l_n \leq q \leq h_n$, where $m_i \in \{0, 1\}$ is the message bit and n is the length of message.

To further clarify the scheme of message embedding using arithmetic decoding, we provide a simple example in Fig. 7 and Fig. 8 .

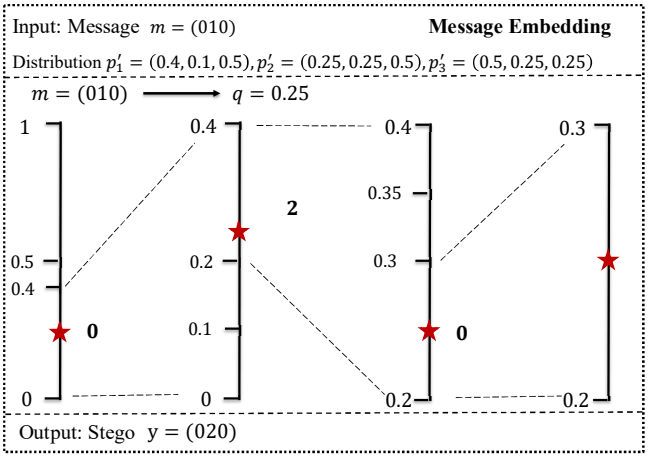


Fig. 7. An example of message embedding using adaptive arithmetic decoding.

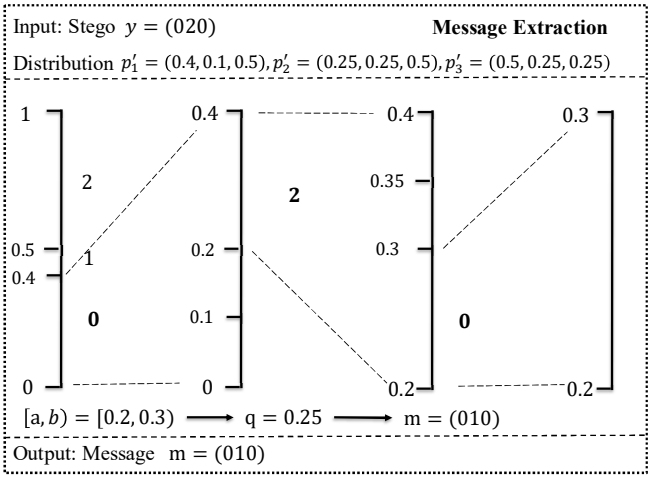


Fig. 8. An example of message extraction using adaptive arithmetic encoding.

B. Proof of distribution-preserving

The secure proof of using adaptive arithmetic coding is discussed in the subsection. The arithmetic code is prefix free, and by taking the binary representation of q and truncating it to $l(c) = \lceil \log \frac{1}{P(c)} \rceil + 1$ bits [34], we obtain a uniquely decodable code. When it comes to encoding the entire sequence $\mathbf{c}$, the number of bits $l(\mathbf{c})$ required to represent $F(\mathbf{c})$ with enough accuracy such that the code for different values of $\mathbf{c}$ are distinct is

$$l(\mathbf{c}) = \lceil \log \frac{1}{P(\mathbf{c})} \rceil + 1. \tag{19}$$

Note that $l(\mathbf{c})$ is the number of bits required to encode the entire sequence $\mathbf{c}$. Therefore, the average length of an arithmetic code for a sequence of length n is given by

$$
\begin{aligned}
l_{A^n} &= \sum P(\mathbf{c}) l(\mathbf{c}) \\
&= \sum P(\mathbf{c}) \left[l(c) = \lceil \log \frac{1}{P(c)} \rceil + 1 \right] \\
&< \sum P(\mathbf{c}) \left[l(c) = \log \frac{1}{P(c)} + 1 + 1 \right] \\
&= -\sum P(\mathbf{c}) \log P(\mathbf{c}) + 2 \sum P(\mathbf{c}) \\
&= H(C^n) + 2.
\end{aligned} \tag{20}
$$

Since the average length is always greater than the entropy, the bounds on l_{A^n} are

$$
H(C^n) \leq l_{A^n} < H(C^n) + 2. \tag{21}
$$

The length per symbol l_A, or rate of the arithmetic code is $\frac{l_A(n)}{n}$. Therefore, the bounds on l_A are

$$
\frac{H(C^n)}{n} \leq l_A < \frac{H(C^n)}{n} + \frac{2}{n}. \tag{22}
$$

Also we know that the entropy of the sequence is nothing but the length of the sequence times the average entropy of every symbol [35]:

$$
H(C^n) = nH(C). \tag{23}
$$

Therefore,

$$
H(C) \leq l_A < H(C) + \frac{2}{n}. \tag{24}
$$

In our framework, P_{s}^n is the real distribution of n samples generated by the process of message embedding using AAD, and P_{c} is the target distribution which we desire to approximate. According to [36, Theorem 5.4.3], using the wrong distribution P_{s}^n for encoding when the true distribution is P_{c} incurs a penalty of $D(P_{\mathrm{c}} \parallel P_{\mathrm{s}}^n)$. In other words, the increase in the expected description length due to the approximate distribution P_{s}^n rather than the true distribution P_{c} is the relative entropy $D(P_{\mathrm{c}} \parallel P_{\mathrm{s}}^n)$. Directly extended from Eq. (24), $D(P_{\mathrm{c}} \parallel P_{\mathrm{s}}^n)$ has a upper bound:

$$
D(P_{\mathrm{c}} \parallel P_{\mathrm{s}}^n) < \frac{2}{n}, \tag{25}
$$

and if $n \to \infty$, then

$$
D(P_{\mathrm{c}} \parallel P_{\mathrm{s}}^n) \to 0. \tag{26}
$$

By increasing the length of the sequence, the relative entropy between P_{c} and P_{s}^n turns to be 0, meaning that the proposed steganographic scheme is distribution-preserving when the sequence is long enough.

V. Experiments

In this section, experimental results and analysis are presented to demonstrate the feasibility and effectiveness of the proposed schemes.

A. WaveGlow System

At first, we introduce the setting of every component of our stegosystem. Here, Fastspeech [31] is adopted as the SPN, due to its certainty in transferring text to mel-spectrogram, and the dictionary and the pretrained model are available at Google Drive[2]. The WaveGlow is trained on the LJ speech [37] data, which consists of 13,100 short audio clips of a single speaker reading passages from 7 non-fiction books. The dataset consists of roughly 24 hours of speech data recorded on a MacBook Pro using its built-in microphone in a home environment. The sampling rate of audios is set as 22.05 kHz. The pretrained model is available at PytorchHub[3].

Sentences with different semantics are fed to the FastSpeech to generate mel-spectrograms. Then the normal cover audios are generated with the Gaussian latent codes, and the stego audios are generated using the mapped latent codes from the encrypted message. All the audio clips are stored in the uncompressed WAV format.

The state-of-the-art audio steganalysis feature, the combined feature of Time-Markov and Mel-Frequency (abbreviated as CTM) [38] is selected to evaluate the security performance. The detectors are trained as binary classifiers implemented using the FLD ensemble with default settings [39]. A separate classifier is trained for each embedding algorithm and payload. The ensemble by default minimizes the total classification error probability under equal priors:

$$
P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2} (P_{\mathrm{FA}} + P_{\mathrm{MD}}). \tag{27}
$$

The ultimate security is qualified by average error rate $\overline{P}_{\mathrm{E}}$ averaged over ten 50/50 database splits, and larger $\overline{P}_{\mathrm{E}}$ means stronger security.

1) Visualization: Both cover audios and corresponding stego audios are available[4], where the readers can evaluate the quality of the audio. Moreover, The audio waveforms as well as their mel-spectrograms are shown in Fig. 9. For the same text, we generate three audios, where the first two audios are generated by the normal process and the third audio is generated by the WaveGlow stegosystem. It can be observed that the waveforms and mel-spectrograms of three audios are similar, but the details are different from each other, showing no special property within one category.

2) Security Performance: The first-order distributions of the cover audios and the stego audio are presented in Fig. 10. Since the KL-divergence will output unstable results, we use energy distance [40] and Wasserstein distance [41] for measurement. The energy distance between two distributions u and v, whose respective CDFs are U and V, equals to:

$$
D(u, v) = \left(2\mathbb{E}|X - Y| - \mathbb{E}\left|X - X'\right| - \mathbb{E}\left|Y - Y'\right| \right)^{1/2}, \tag{28}
$$

where X and X' (resp. Y and Y') are independent random variables whose probability distribution is u (resp. v). $\mathbb{E}$ is the expected value, and $|\cdot|$ denotes the length of a vector.

[2]https://drive.google.com/open?id=1P9I4qag8wAcJiTCPawt6WCKBqUfJFtFp
[3]https://pytorch.org/hub/
[4]https://home.ustc.edu.cn/~chenkj/audio/audio.html

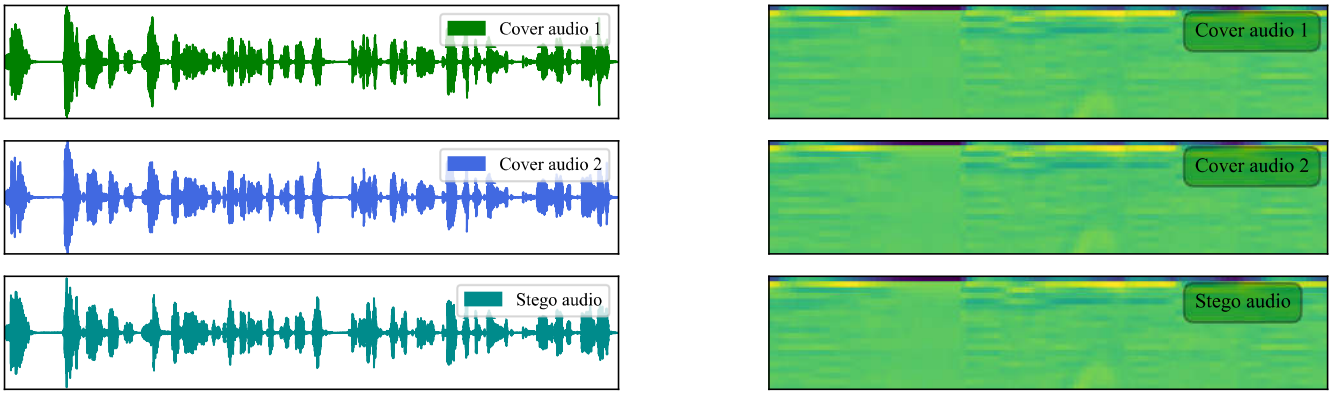


Fig. 9. The visualization of the cover audios triggered by different random seeds and the stego audio generated by WaveGlow stegosystem, including waveforms and mel-spectrograms. The waveforms and the mel-spectrograms are similar between the covers and the stego. The details are different from each other, indicating the stego audio has no special deviation with respect to the cover audios.

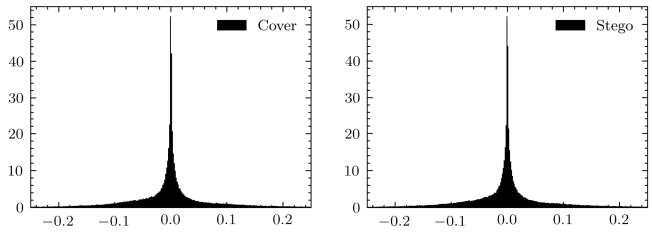


Fig. 10. The first-order distribution of cover audio and stego audio generated by WaveGlow stegosystem with respect to the same text.

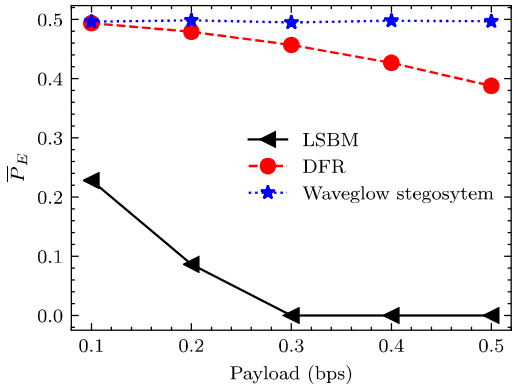


Fig. 11. The average detection error rate $\overline{P}_{\mathrm{E}}$ as a function of payload in bits per sample (bps) for steganographic algorithm payloads ranging from 0.1-0.5 bps against CTM on generated audios by the WaveGlow stegosystem.

The Wasserstein distance between the distributions u and v is:

$$D(u,v) = \inf_{\pi \in \Gamma(u,v)} \int_{\mathbb{R}\times\mathbb{R}} |x-y| \mathrm{d}\pi(x,y), \tag{29}$$

where $\Gamma(u,v)$ is the set of (probability) distributions on $\mathbb{R}\times\mathbb{R}$ whose marginals are u and v on the first and second factors respectively.

The energy distance and the Wasserstein distance of the first-order distributions of the cover audio and the stego audio are presented in Table I. The distance between the first-order distributions of the cover audios and the stego audio is near to 0, and is also near to the distance between the first-order distributions of the cover audios triggered by different random seeds, meaning that the first-order distribution is well preserved by the WaveGlow stegosystem. The performance of the high-order distribution will be checked through steganalysis later.

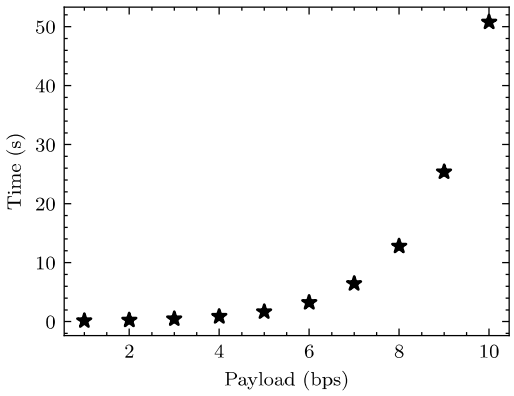


Fig. 12. The running time of message mapping in the WaveGlow stegosystem with respect to different payloads.

TABLE I
THE ENERGY DISTANCE AND WASSERSTEIN DISTANCE OF THE DISTRIBUTION OF THE COVER AUDIOS AND THE STEGO AUDIO GENERATED BY THE WAVEGLOW STEGOSYSTEM.

Distance	Energy distance	Wasserstein distance
Cover1-Cover2	0.0190	0.02815
Cover1-Stego	0.0204	0.02543

To validate the distribution-preserving ability of steganalytic methods on the synthesized audio, we introduce modification-based steganographic methods for comparison, including least significant bit matching (LSBM) and derivative filter residual (DFR) [42]. During the embedding process of LSBM, if the LSB of the cover element matches the secret data bit, this element will be kept intact. Otherwise, the element will be altered by $+1$ or -1 at random. DFR is an adaptive steganographic method, which assigns different modification

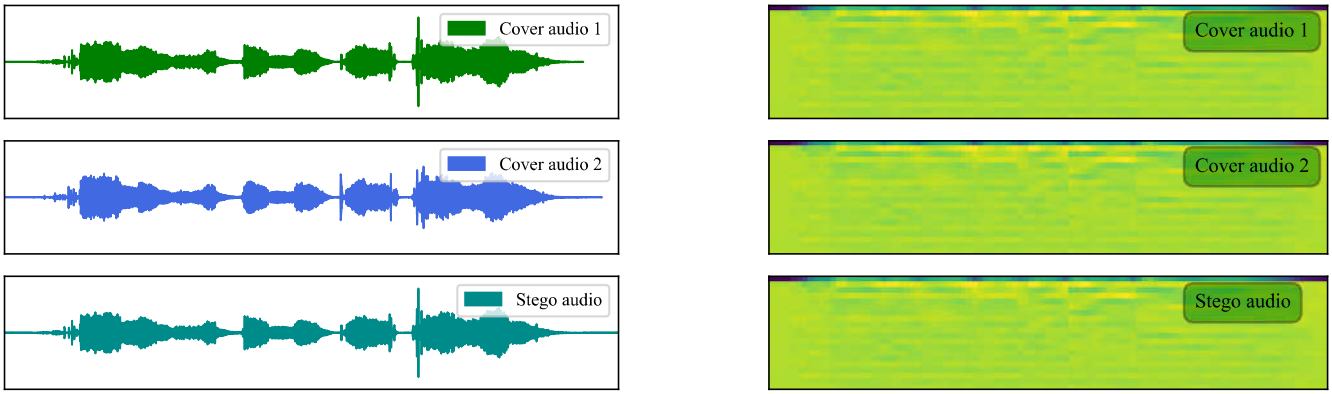


Fig. 13. The visualization of the cover audios triggered by different random seeds and the stego audio generated by the WaveNet stegosystem, including waveforms and mel-spectrograms. The waveforms and the mel-spectrograms are similar between the cover audios and the stego audio. The details are different from each other, indicating the stego audio has no special deviation with respect to the cover audios.

distortion to different elements and then embeds message using minimizing distortion codes, such as syndrome trellis codes (STCs) [43]. Here, the DFR algorithm is simulated at its payload-distortion bound. The payload ranges from 0.1 to 0.5 bit per sample (bps).

Fig. 11 shows the steganalysis performance of different embedding methods. The $\overline{P}_{\mathrm{E}}$ of LSBM and DFR is lower than 50%. With the increment of payload, the $\overline{P}_{\mathrm{E}}$ decreases, showing that the CTM steganalytic feature is effective on this dataset. The $\overline{P}_{\mathrm{E}}$ of our proposed WaveGlow stegosystem is around 50%, meaning that the steganalyzer nearly randomly judges synthesized audio is cover or stego. That is to say, the high-order distribution of cover audios and stego audios cannot be distinguished by the steganalyzer as well, indicating the WaveGlow system is distribution-preserving.

3) Efficiency and Capacity: To evaluate the efficiency of the WaveGlow stegosystem, we measure the running time of the system in terms of different payloads in Fig. 12. The running time increases exponentially with the increase of the payload. As for the k-division situation, the probability p of rejection sample dropping into the interval of the given message is $\frac{1}{2^k}$. Then the expectation E of the sampling time is:

$$E = \sum_{i=1}^{n} i \cdot p \cdot (1-p)^{i-1} = \frac{1}{p} = 2^k. \tag{30}$$

It can be seen that the expectation E grows exponentially as k grows, which is consistent to the experimental results in Fig. 12. The capacity of the WaveGlow stegosystem is determined by the message mapping module. The upper bound of the capacity is 32 or 64 bps, depending on which 32-bit system or 64-bit system the current system is.

B. WaveNet Stegosystem

We randomly collect 1,000 short text sentences and transfer them into mel-spectrograms using the SPN[5] in Tacotron-2 [26]. Then WaveNet vocoder is used for audio waveform generation[6]. The WaveNet vocoder is trained on *CMU ARCTIC* dataset [44] with 100,000 steps. All the audio clips are stored in the uncompressed WAV format. The audio length ranges from 0.5s to 3s, and the sample rate is 16kHz.

1) Visualization: Similarly, both cover audios and corresponding stego audios are also available[7]. The audio waveforms and their mel-spectrograms are shown in Fig. 13. The phenomenon of these audios are similar to Section V-A1.

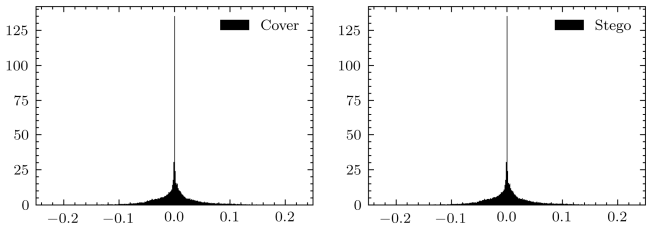


Fig. 14. The first-order distribution of cover audio and stego audio generated by the WaveNet stegosystem.

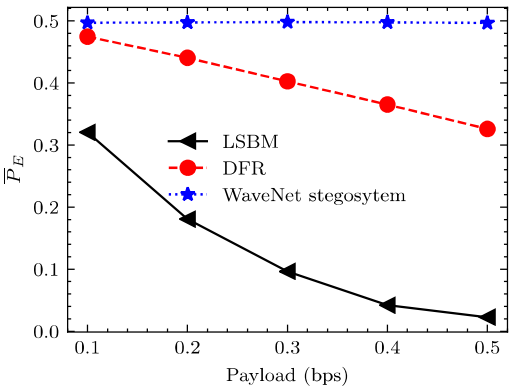


Fig. 15. The average detection error rate $\overline{P}_{\mathrm{E}}$ as a function of payload in bits per sample (bps) for steganographic algorithm payloads ranging from 0.1-0.5 bps against CTM on generated audios by the WaveNet stegosystem.

[5]The architecture of the spectrogram prediction network can be downloaded at https://github.com/Rayhane-mamah/Tacotron-2.

[6]The architecture of WaveNet vocoder can be downloaded at https://github.com/kan-bayashi/PytorchWaveNetVocoder

[7]https://home.ustc.edu.cn/~chenkj/audio/audio.html

TABLE II
THE ENERGY DISTANCE AND WASSERSTEIN DISTANCE BETWEEN THE DISTRIBUTION OF THE COVER AUDIOS AND THE STEGO AUDIO GENERATED BY THE WAVEGLOW STEGOSYSTEM.

Distance	Energy distance	Wasserstein distance
Cover1-Cover2	0.0337	0.0571
Cover1-Stego	0.0291	0.0681

2) Security Performance: The first-order distributions of the cover audios by different random seeds and the stego audio are presented in Fig. 14. The energy distance and Wasserstein distance of the first-order distributions of the cover audios triggered and the stego are listed in Table II. The distance between the first-order distributions of the cover audios and the stego audio is near to 0 and is also near to the distance between the first-order distributions of the cover audios triggered by different random seeds, meaning that the first-order distribution of cover is well preserved.

First, we verify the effectiveness of the steganalysis on the generated audios. Fig. 15 shows the average detection error rate $\overline{P}_{\mathrm{E}}$ as a function of payload in bps for steganographic algorithm payloads ranging from 0.1-0.5 bps against CTM. It can be observed in Fig. 15 that the $\overline{P}_{\mathrm{E}}$ of DFR and LSBM decreases with the increment of payload and, showing that the steganalysis is effective with respect to the generated audios. The $\overline{P}_{\mathrm{E}}$ of the proposed WaveNet stegosystem is nearly 50%, which means the stego generated by the proposed system cannot be detected. In other words, the strong steganalyzer judges the stego nearly by random guess. The experimental results verify that our proposed WaveNet stegosystem is distribution-preserving, as proved in Section IV-B.

3) Efficiency and Capacity: Since WaveNet is an auto-regressive generative model, the process of generating audios is time-consuming. The running time of arithmetic decoding with respect to the time of generating audios is negligible. Numerically, the speed of audio generation on NVIDIA GeForce 1080Ti is about 0.435 sec/sample. The capacity of a generated audio is equivalent to the entropy of the distribution of audio samples.

VI. CONCLUSIONS

In this paper, we review the distribution-preserving steganography, which includes: sampling-based stegosystem and compression-based stegosystem, and conclude the limitation of current works: lack of efficient perfect sampler and cannot give the explicit distribution of natural media.

The fast development of generative models brings a great opportunity to the distribution-preserving steganography. Based on WaveGlow, we design a sampling-based stegosystem, where the generator serves as the sampler. A message mapping module transfers the message into the latent codes, which is then fed to the sampler and produce the stego. The inverse functions of the mapping module and the generator are used to extract the message. Besides, according to the duality between message embedding and data compression, we design a compression-based method based on the auto-regressive generative model, WaveNet. With the probability distribution of every sample, the message can be decompressed into the generated audios. The security performances of the two systems are verified by state-of-the-art steganalysis methods. Additionally, the theoretical proof is given for the WaveNet stegosystem using adaptive arithmetic coding.

In our future work, we will explore other effective source encoding schemes and transfer them to generative steganographic codes. Furthermore, other generative media, such as videos and 3D meshes, will be utilized under the proposed framework.

ACKNOWLEDGMENT

The authors would like to thank Prof. Weiqi Luo from Sun Yat-sen University for providing us the source codes of audio steganalysis.

REFERENCES

[1] C. Cachin, "An information-theoretic model for steganography," in *International Workshop on Information Hiding*, 1998, pp. 306–318.
[2] N. J. Hopper, J. Langford, and L. Von Ahn, "Provably secure steganography," in *Annual International Cryptology Conference*, 2002, pp. 77–92.
[3] R. J. Anderson and F. A. Petitcolas, "On the limits of steganography," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 474–481, 1998.
[4] T. Van Le, "Efficient provably secure public key steganography." *IACR Cryptology ePrint Archive*, vol. 2003, p. 156, 2003.
[5] P. Sallee, "Model-based steganography," in *International Workshop on Digital Watermarking*, 2003, pp. 154–167.
[6] L. Von Ahn and N. J. Hopper, "Public-key steganography," in *International Conference on the Theory and Applications of Cryptographic Techniques*, 2004, pp. 323–341.
[7] A. Lysyanskaya and M. Meyerovich, "Provably secure steganography with imperfect sampling," in *International Workshop on Public Key Cryptography*, 2006, pp. 123–139.
[8] Y. Zhu, M. Yu, H. Hu, G.-J. Ahn, and H. Zhao, "Efficient construction of provably secure steganography under ordinary covert channels," *Science China Information Sciences*, vol. 55, no. 7, pp. 1639–1649, 2012.
[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[10] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.
[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
[12] M. Germain, K. Gregor, I. Murray, and H. Larochelle, "Made: Masked autoencoder for distribution estimation," in *International Conference on Machine Learning*, 2015, pp. 881–889.
[13] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, "Conditional image generation with pixelcnn decoders," in *Advances in Neural Information Processing Systems*, 2016, pp. 4790–4798.
[14] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *ISCA Speech Synthesis Workshop*, 2016, p. 125.
[15] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 215–10 224.
[16] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," in *Advances in Neural Information Processing Systems*, 2017, pp. 1954–1963.
[17] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A novel image steganography method via deep convolutional generative adversarial networks," *IEEE Access*, vol. 6, pp. 38 303–38 314, 2018.
[18] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *European Conference on Computer Vision*, 2018, pp. 657–672.

[19] Z. Zhang, J. Liu, Y. Ke, Y. Lei, J. Li, M. Zhang, and X. Yang, "Generative steganography by sampling," *IEEE Access*, vol. 7, pp. 118 586–118 597, 2019.

[20] M. Yedroudj, F. Comby, and M. Chaumont, "Steganography using a 3 player game," *arXiv preprint arXiv:1907.06956*, 2019.

[21] N. Zhong, Z. Qian, Z. Wang, X. Zhang, and X. Li, "Batch steganography via generative network," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[22] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptology*, 1984, pp. 51–67.

[23] J. Fridrich, *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009.

[24] A. Kerckhoffs, "A. kerckhoffs, la cryptographie militaire," *Journal Des Sciences Militaires*, vol. 9, p. 38, 1883.

[25] N. Hopper, L. von Ahn, and J. Langford, "Provably secure steganography," *IEEE Transactions on Computers*, vol. 58, no. 5, pp. 662–676, 2009.

[26] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4779–4783.

[27] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," *Proc. ICLR*, pp. 214–217, 2018.

[28] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," in *International Conference on Learning Representations*, 2019.

[29] Y. Masuyama, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Deep griffin-lim iteration," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 61–65.

[30] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 3617–3621.

[31] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, 2019, pp. 3165–3174.

[32] K. Yang, K. Chen, W. Zhang, and N. Yu, "Provably secure generative steganography based on autoregressive model," in *International Workshop on Digital Watermarking*, 2018, pp. 55–68.

[33] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *arXiv preprint arXiv:1601.06759*, 2016.

[34] K. Sayood, *Introduction to data compression*. Morgan Kaufmann, 2017.

[35] A. Said, "Introduction to arithmetic coding-theory and practice," *Hewlett Packard Laboratories Report*, pp. 1057–7149, 2004.

[36] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[37] K. Ito and L. Johnson, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[38] W. Luo, H. Li, Q. Yan, R. Yang, and J. Huang, "Improved audio steganalytic feature and its applications in audio forensics," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 2, p. 43, 2018.

[39] J. Kodovskỳ, J. J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media." *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[40] M. L. Rizzo and G. J. Székely, "Energy distance," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 8, no. 1, pp. 27–38, 2016.

[41] R. L. Dobrushin, "Asymptotic behavior of gibbsian distributions for lattice systems and its dependence on the form of the volume," *Theoretical and Mathematical Physics*, vol. 12, no. 1, pp. 699–711, 1972.

[42] K. Chen, H. Zhou, W. Li, K. Yang, W. Zhang, and N. Yu, "Derivative-based steganographic distortion and its non-additive extensions for audio," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2027–2032, 2020.

[43] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[44] J. Kominek and A. W. Black, "The CMU ARCTIC speech databases," in *ISCA Workshop on Speech Synthesis*, 2004.