2023 IEEE Symposium on Security and Privacy

Discop: Provably Secure Steganography in Practice Based on "Distribution Copies"

Jinyang Ding[*] Kejiang Chen[*] Yaofei Wang[+] Na Zhao[*] Weiming Zhang[*] Nenghai Yu[*]
[*]University of Science and Technology of China [+]Hefei University of Technology

source@mail.ustc.edu.cn {chenkj, zhangwm}@ustc.edu.cn

中国科学技术大学 University of Science and Technology of China

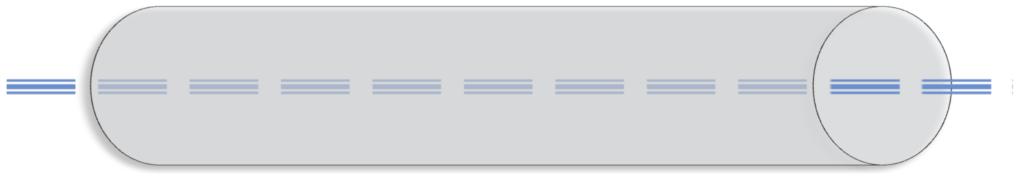
合肥工业大学 HEFEI UNIVERSITY OF TECHNOLOGY

Censorship is everywhere!

Combat Extreme Censorship

Encryption-based tools

Can be easily identified and blocked!



Tunnel-based covert channels

Protozoa [CCS '20], Balboa [USENIX Sec '21]

No guarantee of sustainability!

Censors can block the encrypted traffic where they don't have a suitable trapdoor.

Ideal Technique: Steganography

Steganography System

Steganography: Prisoners' Problem [Simmons. CRYPTO '83]

- Embed a secret message in a mundane-appearing object
- **Cover:** an object **without** a secret message embedded
- **Stego:** an object **with** a secret message embedded

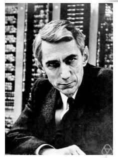
Current mainstream steganography methods

- Steganography by **cover modification**, e.g., LSB replacement, adaptive steganography
- **Limitation: their security cannot be formally proved!**

A more advanced pursuit: Provably Secure Steganography (PSS)

Steganographic Security

☐ Provably secure steganography (PSS)



C.E. Shannon

Communication Theory of Secrecy Systems*

By C. E. SHANNON

1 INTRODUCTION AND SUMMARY

The problems of cryptography and secrecy systems furnish an interesting application of communication theory[1]. In this paper a theory of secrecy systems is developed. The approach is on a theoretical level and is intended to complement the treatment found in standard works on cryptography[2]. There, a detailed study is made of the many standard types of codes and ciphers, and of the ways of breaking them. We will be more concerned with the general mathematical structure and properties of secrecy systems.

The treatment is limited in certain ways. First, there are three general types of secrecy system: (1) concealment systems, including such methods as invisible ink, concealing a message in an innocent text, or in a fake covering cryptogram, or other methods in which the existence of the message is concealed from the enemy; (2) privacy systems, for example speech inversion, in which special equipment is required to recover the message; (3) "true" secrecy systems where the meaning of the message is concealed by cipher, code, etc., although its existence is not hidden, and the enemy is assumed to have any special equipment necessary to intercept and record the transmitted signal. We consider only the third type—concealment system are primarily a psychological problem, and privacy systems a technological one.

Shannon. **Communication Theory of Secrecy Systems**. BSTJ '49

Rejection sampling-based PSS

TURING AWARD

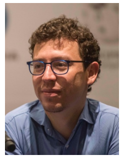
Manuel Blum

Provably Secure Steganography
(Extended Abstract)

Nicholas J. Hopper, John Langford, and Luis von Ahn

Computer Science Department, Carnegie Mellon University,
Pittsburgh PA 15213, USA
{hopper,jcl,biglou}@cs.cmu.edu

N.J. Hopper



Luis von Ahn

Hopper et al. **Provably Secure Steganography**. CRYPTO '02

Public-Key Steganography

Luis von Ahn and Nicholas J. Hopper

Computer Science Dept, Carnegie Mellon University, Pittsburgh PA 15213 USA

von Ahn and Hopper. **Public-Key Steganography**. EUROCRYPT '04

A necessary condition: sampleable distribution

Challenging to meet at that time ⇒ For a long time, cannot be put into practice

Steganographic Security

☐ Provably secure steganography (PSS)

- Put steganography on a solid theoretical foundation
- Theoretically ensure undetectability

☐ Information-theoretic security [Cachin. IH ’98]

- KL divergence between the cover and stego distributions

$$D_{\mathrm{KL}}(P_{\mathrm{c}} \parallel P_{\mathrm{s}}) = \sum_{\mathbf{x} \in \mathcal{C}} P_{\mathrm{c}}(\mathbf{x}) \log \frac{P_{\mathrm{c}}(\mathbf{x})}{P_{\mathrm{s}}(\mathbf{x})}$$

☐ Computational security [Hopper et al. CRYPTO ’02] [Katzenbeisser and Petitcolas. SWMC ’02]

- The adversary is playing a game of distinguishing covers and stegos
- Secure if all PPT adversaries have a negligible advantage in the game

$$|\Pr[\mathcal{A}_{\mathcal{D}}^{\mathsf{Encode}_{\mathcal{D}}(K,\cdot,\cdot)} = 1] - \Pr[\mathcal{A}_{\mathcal{D}}^{\mathcal{O}_{\mathcal{D}}(\cdot,\cdot)} = 1]| < \mathrm{negl}(\lambda)$$

AI Generative Models

ChatGPT reaches 100 million users two months after launch.

Stable Diffusion created millions of images in the first two months.

Gartner's report predicted that generative AI will account for 10% of all data produced by 2025.

The popularity of AI generative models provides a brand new camouflage environment for PSS!

In 2018, WM group first proposed to use AI generative models to conduct PSS

Yang et al. **Provably Secure Generative Steganography Based on Autoregressive Model**. IWDW '18

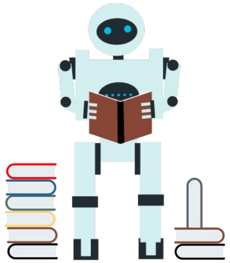
Existing PSS Methods

- Thanks to AI generative models, PSS is moving towards practicality

Theoretical w/o AI

Practical w/ AI

Category	Authors	Publication Venue	Abbr.	Message Expression Method
Rejection sampling-based	Hopper et al.	CRYPTO '02	RejSamp	$f(\text{stego})$
	von Ahn and Hopper	EUROCRYPT '04		
	Backes and Cachin	TCC '05		
Arithmetic coding-based	Le	IACR ePrint '03	AC	Token indexes
	Yang et al.	**IWDW '18**		
	Ziegler et al.	EMNLP '19		
	Chen et al.	**TDSC '21**		
	Kaptchuk et al.	CCS '21	Meteor	
Grouping-based	Zhang et al.	ACL Findings '21	ADG	Group indexes

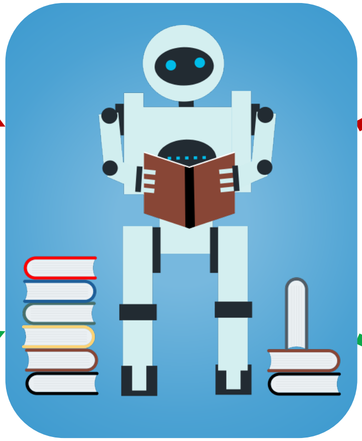


Taking Text Generation as an Example

☐ **Auto-regressive (AR) model** (e.g. GPT-2/3/4)

- Generates text in a **token-by-token** fashion
- Trained to predict the probability distribution of **the next token** $\Pr[x_t \mid x_{<t}]$
- Iteratively repeats **prediction** and **random sampling**

Context
Hello, my



Predict

Token	Probability
name	0.439
friend	0.350
dear	**0.211**

Random Sampling → **dear**

Append →

Context
Hello, my **dear**

Predict

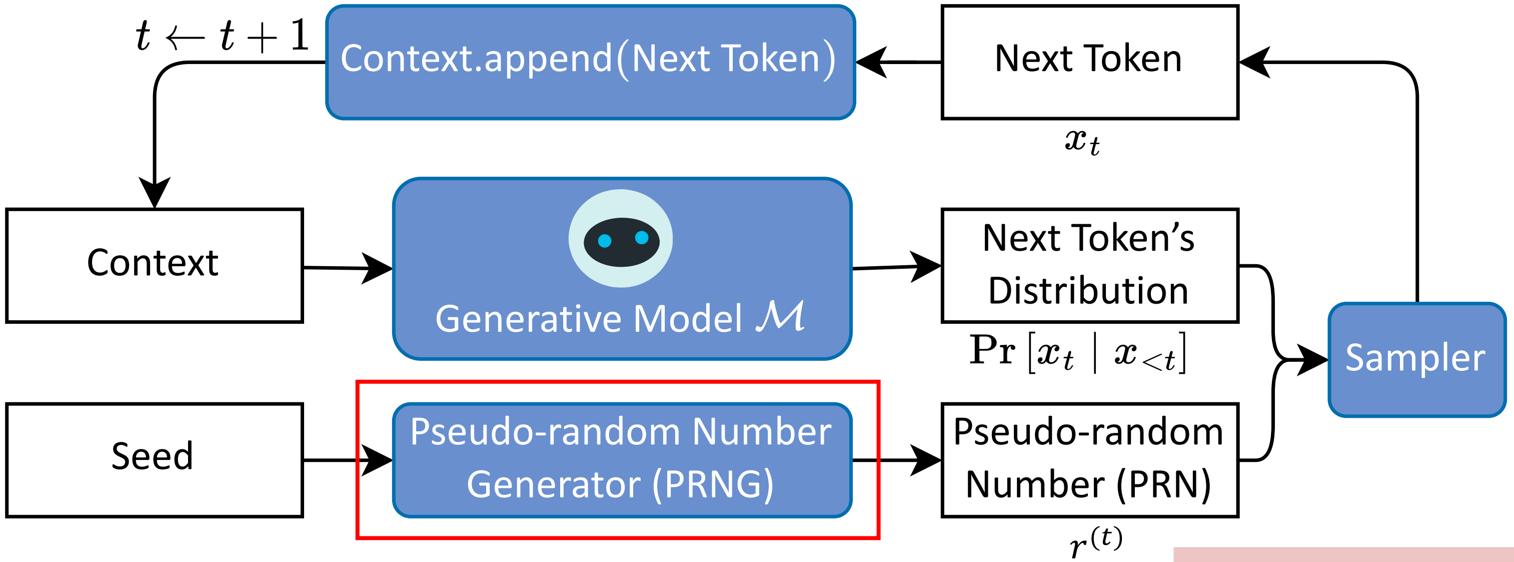
Token	Probability
friend	0.666
audience	**0.233**
reader	0.101

Random Sampling → **audience**

Taking Text Generation as an Example

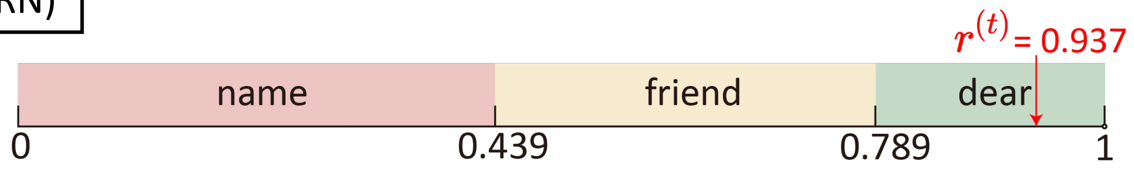
Random sampling

- **Assign** an interval in $[0,1)$ to each token according to $\Pr[x_t \mid x_{<t}]$
- **Consume** a pseudo-random number $r^{(t)} \sim U[0,1)$ from the PRNG
- **Select** the token corresponding to the interval $r^{(t)}$ falls into as the next token



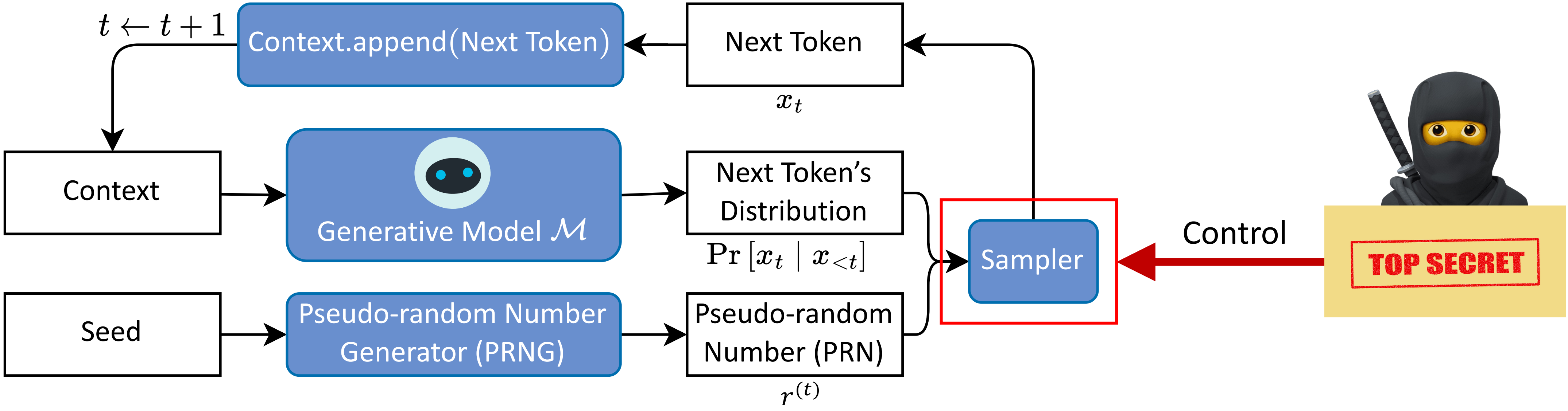
The seed is the number (or vector) used to initialize the PRNG

Token	Probability	Interval
name	0.439	[0.000, 0.439)
friend	0.350	[0.439, 0.789)
dear	**0.211**	**[0.789, 1.000)**



Taking Text Generation as an Example

- **How to achieve PSS in a generation process?**
- **PSS sampling** (sample under the control of the secret message)
  - **Indistinguishable** from normal sampling (random sampling)
  - **Reversible:** the receiver can recover the secret message from the sampled token



Existing PSS Methods

☐ We analyze their problems in practice

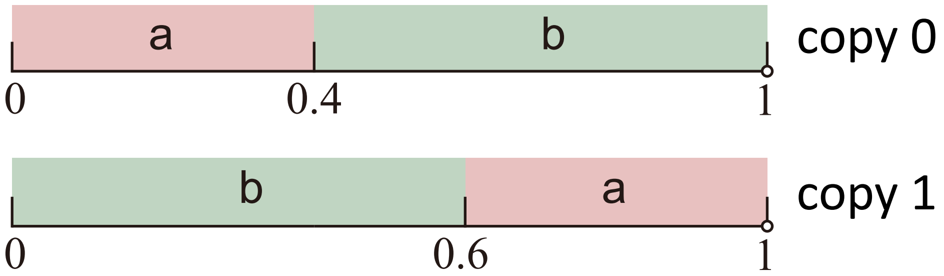
Category	Authors	Publication Venue	Abbr.	Message Expression Method	Problems in Practice
Rejection sampling-based	Hopper et al.	CRYPTO ’02	RejSamp	$f(\text{stego})$	Ineffcient
	von Ahn and Hopper	EUROCRYPT ’04			
	Backes and Cachin	TCC ’05			
Arithmetic coding-based	Le	IACR ePrint ’03	AC	Token indexes	Fail to achieve the expected security
	Yang et al.	**IWDW ’18**			
	Ziegler et al.	EMNLP ’19			
	Chen et al.	**TDSC ’21**			
	Kaptchuk et al.	CCS ’21	Meteor		
Grouping-based	Zhang et al.	ACL Findings ’21	ADG	Group indexes	

Our Basic Construction

Our insight

- The **interval assignment scheme** is not unique
- All schemes share identical distribution, hence called **"distribution copies"**

Token	Probability
a	0.4
b	0.6



Our idea

- If we want to embed n bits of information, we can construct 2^n "distribution copies" and use the **copy index** to express information!

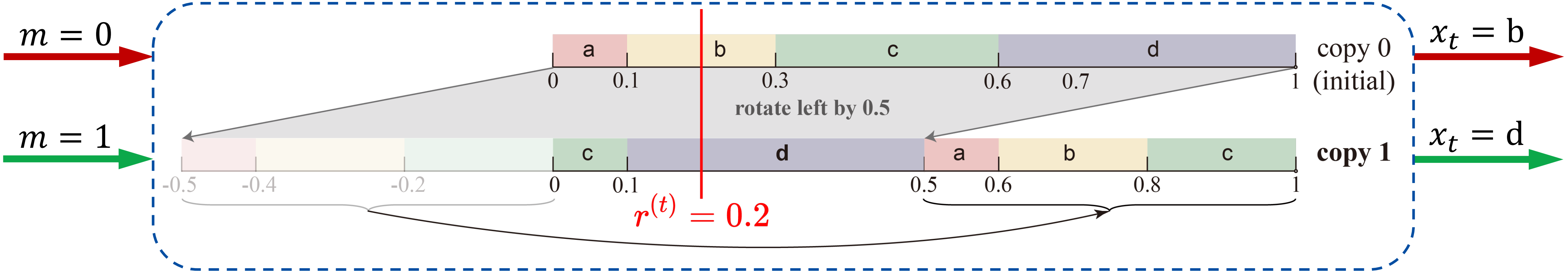
Our Basic Construction

Token	a	b	c	d
Probability	0.1	0.2	0.3	0.4

☐ How to construct multiple "distribution copies"? **Rotation!**

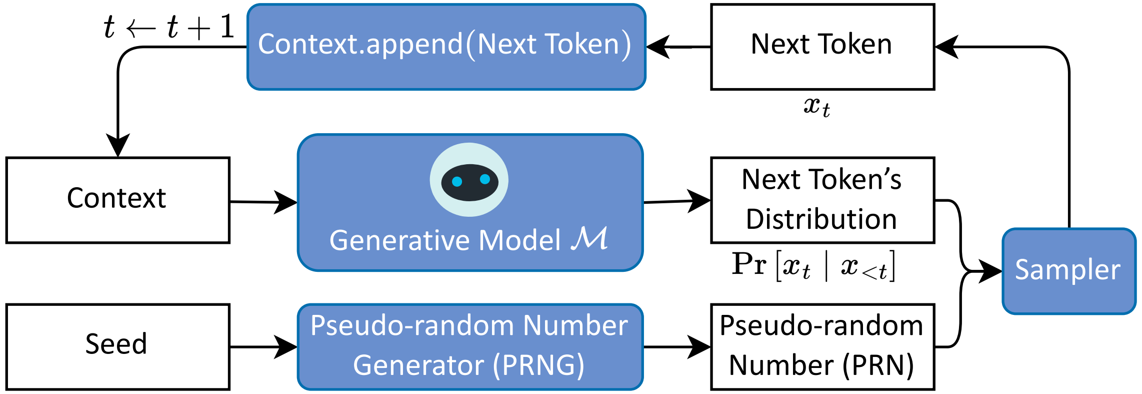
☐ A running example

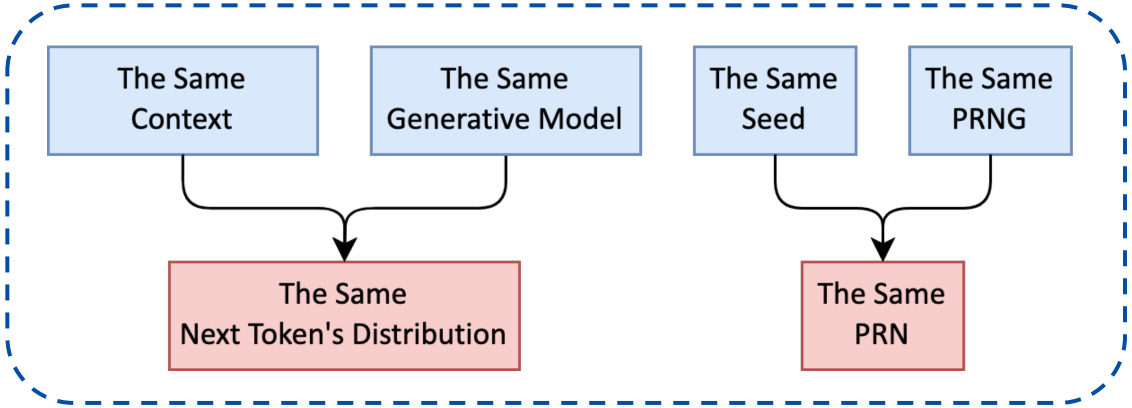
- Take "a-b-c-d" as **the initial interval assignment scheme** (i.e., **copy 0**)
- If we want to embed **1 bit** of information, we need to construct $2^1 =$ **2 copies**
- The corresponding **rotation step size** is $1/2 = \mathbf{0.5}$
- Rotate **copy 0** to the left by 0.5 to obtain **copy 1**
- Consume a **pseudo-random number** $r^{(t)} = 0.2$ from PRNG
- To embed message $m \in \{0, 1\}$, sample from copy m



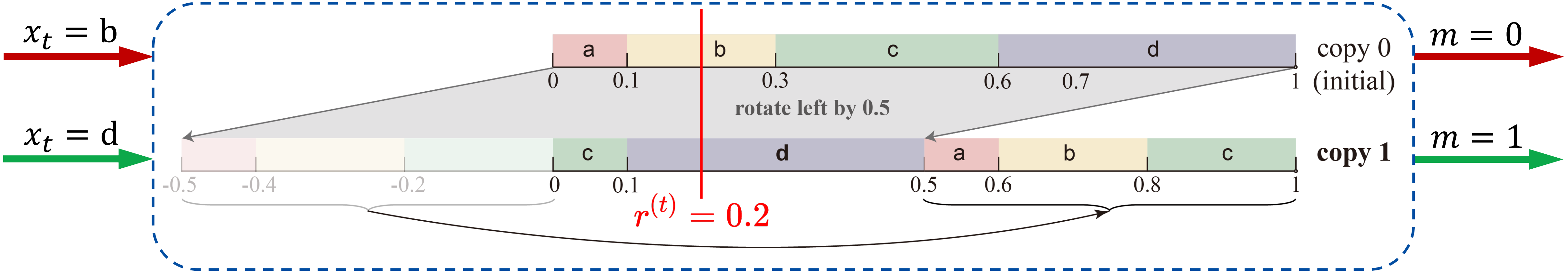
Our Basic Construction

From the receiver's perspective





- Bob can **synchronize all states** with Alice, so he can **recover the message** from the received token



Our Basic Construction

Condition for unique decoding & Embedding capacity

- If we want to embed **1 bit**, we need to construct **2 copies**, so the rotation step size is **0.5**

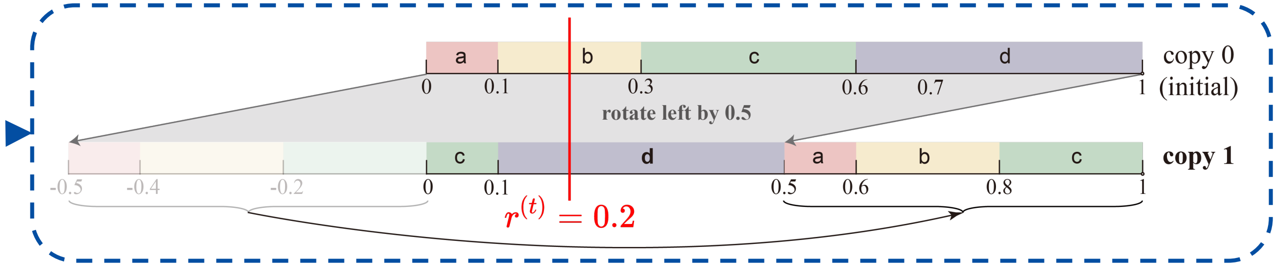


Fig. 1

- If we want to embed **2 bits**, we need to construct **4 copies**, so the rotation step size is **0.25**
- The parts covered by the gray masks indicate **the disputed ranges** (covered by grey)

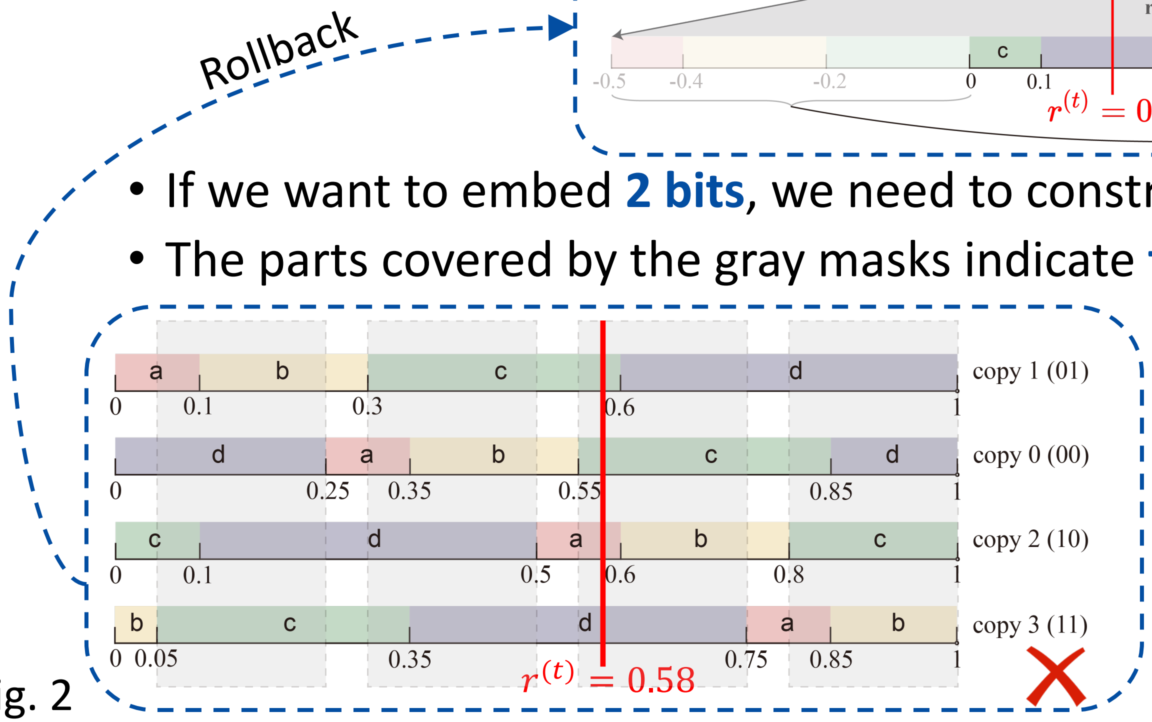


Fig. 2

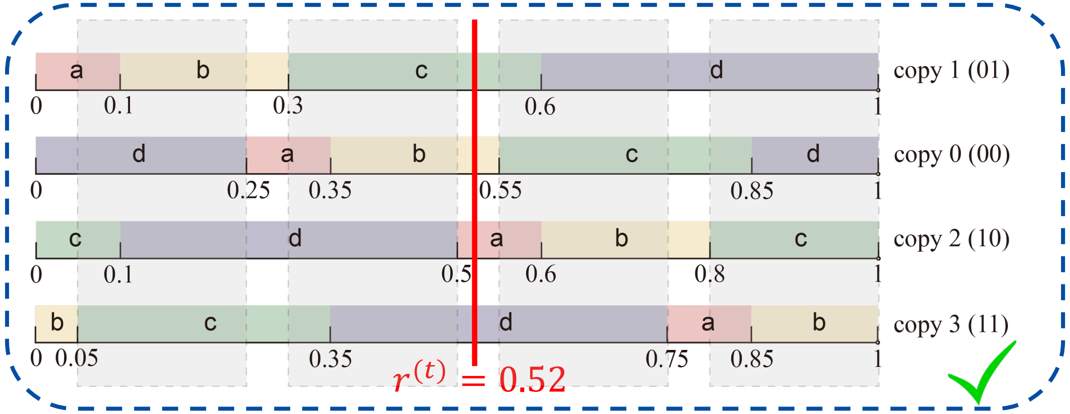


Fig. 3

- **Ave(Embedding rate) = The distribution's minimum entropy 😐 Its theoretical limit = The distribution's entropy**

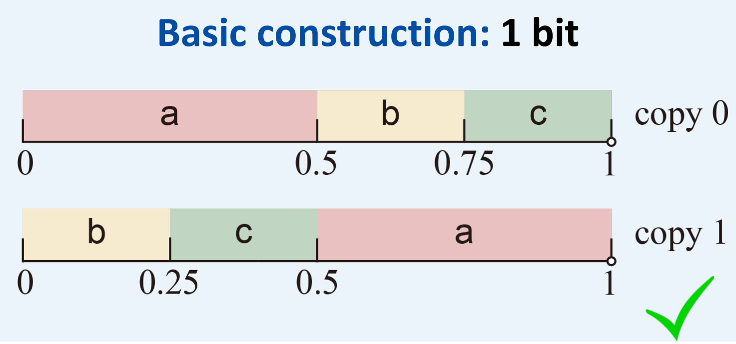
Our Improved Construction: Discop

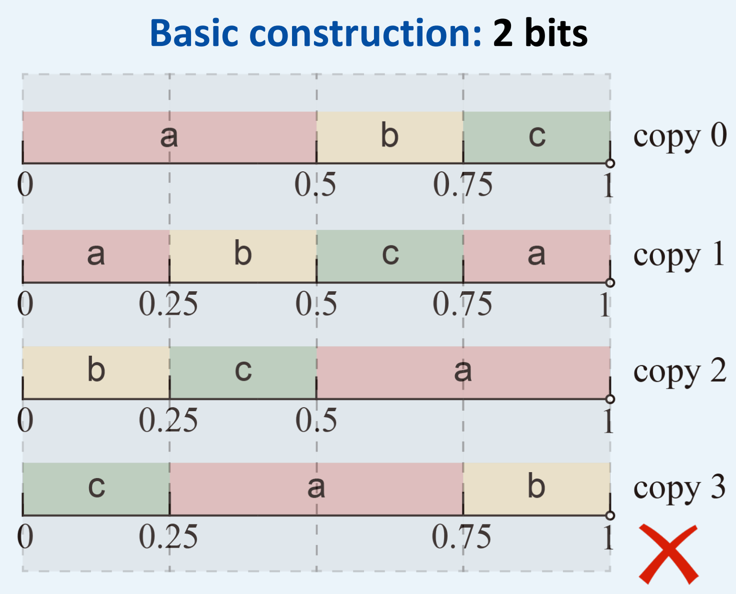
How to improve the embedding rate?

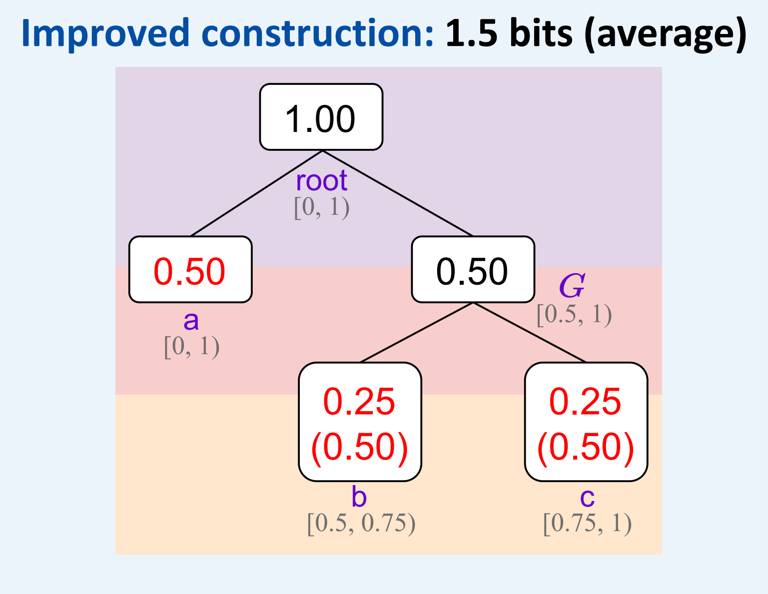
A toy example

- **Basic construction:** only 1 bit
- **Improved construction:** 1.5 bits (50% 1 bit, 50% 2 bits)

Token	a	b	c
Probability	0.50	0.25	0.25







Our Improved Construction: Discop

Our idea: two steps

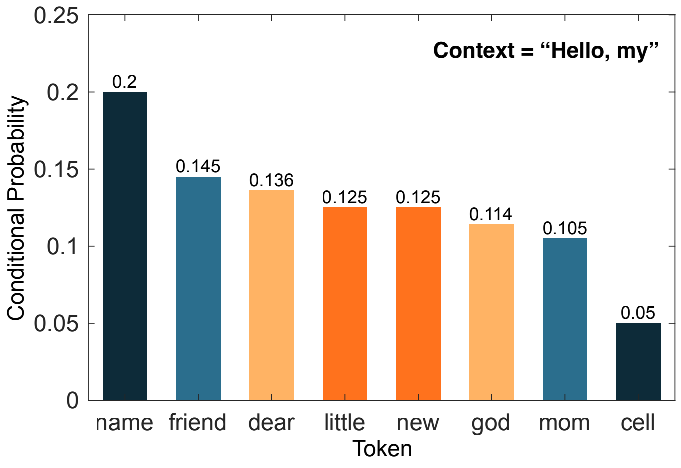
- Construct a **Huffman tree** by recursive grouping
- Embed the message bits in **child node selections**

The multi-variate distribution

decompose

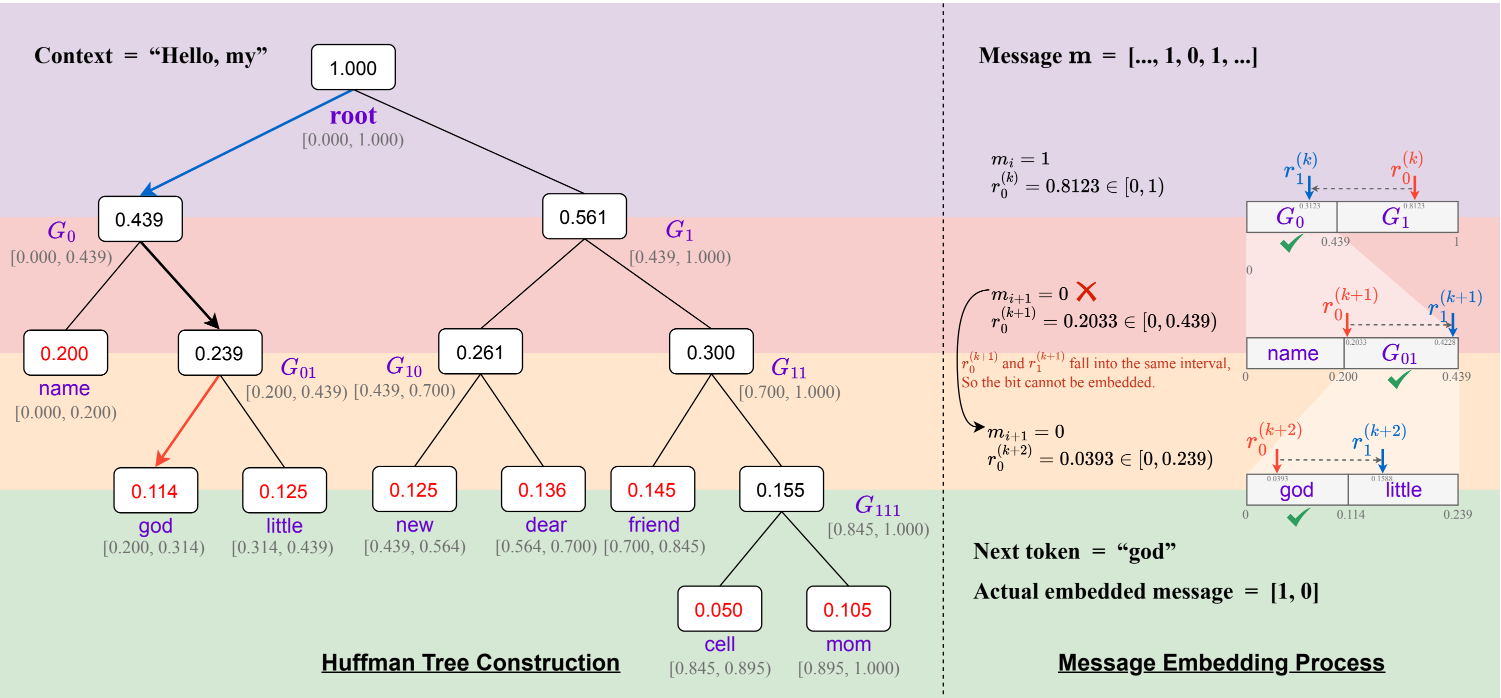
Multiple bi-variate distributions

A running example



Example of probability distribution ▲

Example of Discop's embedding algorithm ▶



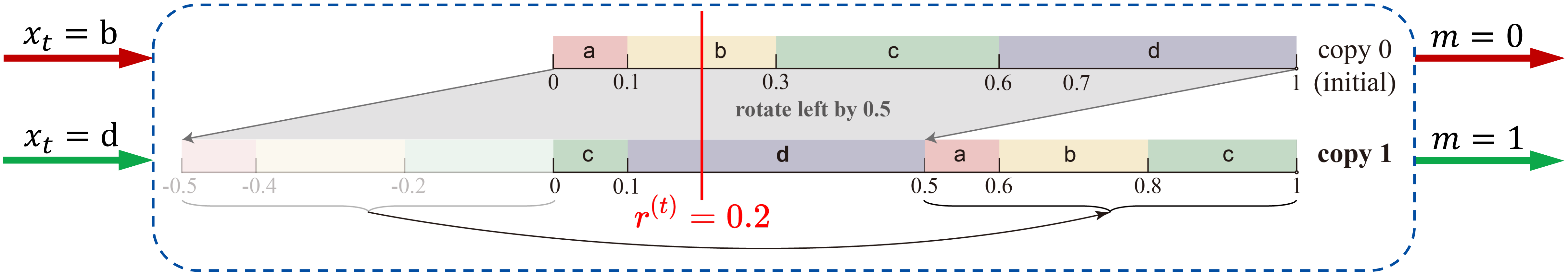
Proof of Security

Intuitive proof of security

- Random sampling from one of the "distribution copies"
- The distribution of all copies is identical
- The steganographic behavior **DOES NOT** damage the original distribution

More rigorous proof of security

- Preserve the probability density of arbitrary pseudo-random number
- For more details, please refer to our paper



Deployments

☐ Deploy Discop on three typical generation tasks

Text Generation
GPT-2

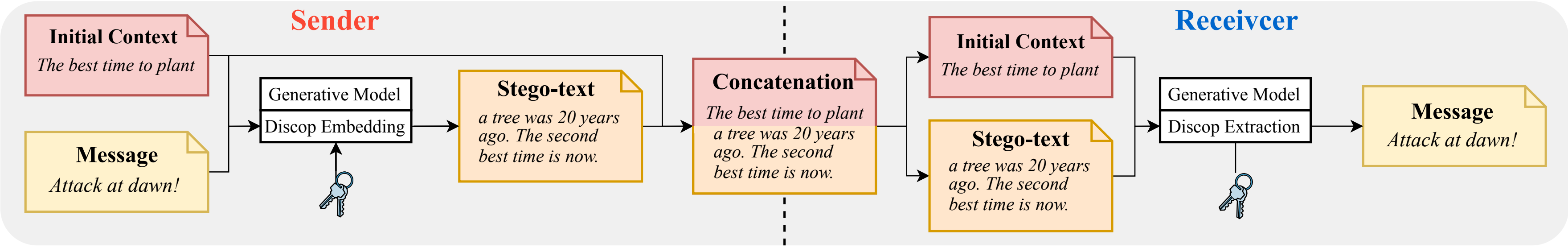
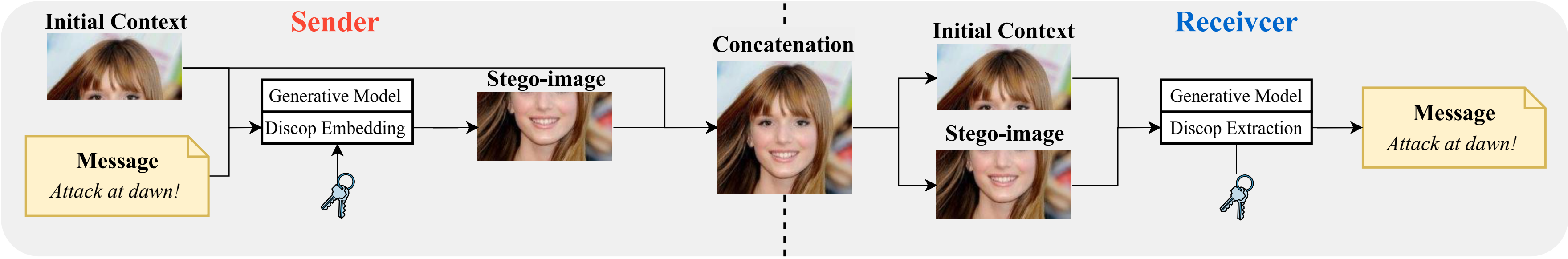
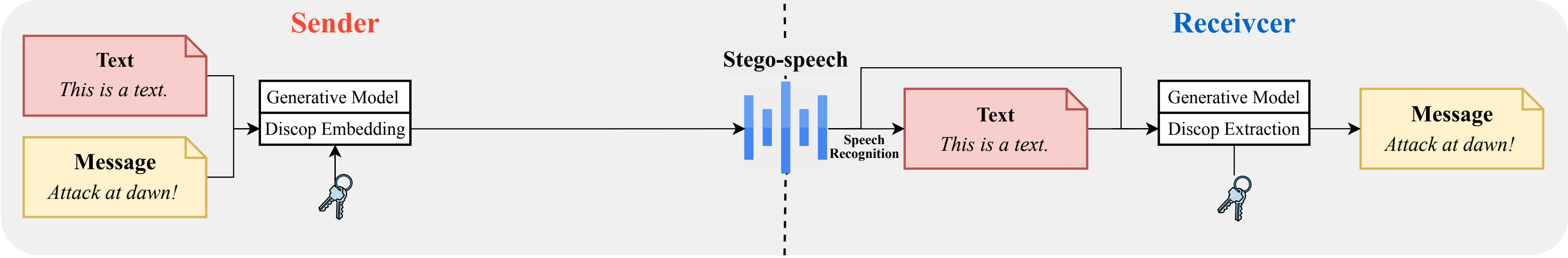


Image Completion
Image GPT



Text-to-speech
WaveRNN



WM group

Evaluations

☐ Setup

- **Top-p sampling:** $p = 0.80, 0.92, 0.95, 0.98, 1.00$
- **Text:** **100** texts from IMDb, the first 3 sentences as the context, generate **100** tokens
- **Image:** **100** images from CelebA, the upper part as the context, complete the image (**512** pixels)
- **Speech:** **100** texts from IMDb, synthesis the speech of the first sentence
- **Baselines:** Meteor and ADG (only on the text generation task)
- **Hardware:** CPU 3.00GHz, 128GB RAM, and NVIDIA RTX 3090

☐ Evaluation axes

- **Security:** Ave KLD, Max KLD (bits / token)
- **Time Efficiency:** Ave Time (seconds / bit)
- **Capacity Efficiency:** Utilization New!

$$\text{Utilization} = \frac{\text{Embedding capacity (total length of the embedded message)}}{\text{Embedding capacity's theoretical limit (entropy sum over all time steps)}}$$

Evaluations

Comparison to Meteor and ADG (using GPT-2)

Method	p	Total Time (seconds)	Ave Time ↓ (seconds/bit)	Ave KLD ↓ (bits/token)	Max KLD ↓ (bits/token)	Capacity (bits/token)	Entropy (bits/token)	Utilization ↑
ADG	0.80	96.71	3.16E-03	7.93E-03	6.76E-02	3.07	3.95	0.78
	0.92	104.48	2.57E-03	1.02E-02	4.75E-02	4.06	4.93	0.82
	0.95	114.72	2.62E-03	1.09E-02	4.73E-02	4.38	5.34	0.82
	0.98	150.68	3.08E-03	1.20E-02	4.54E-02	4.89	5.83	0.84
	1.00	846.27	1.57E-02	1.31E-02	4.99E-02	5.39	6.26	0.86
Meteor w/o sort	0.80	95.58	3.73E-03	5.13E-02	8.28E+00	2.56	3.83	0.67
	0.92	96.16	2.79E-03	8.17E-03	5.62E+00	3.44	4.82	0.71
	0.95	98.57	2.61E-03	3.40E-03	1.30E+00	3.78	5.15	0.73
	0.98	105.37	2.51E-03	6.59E-04	1.74E+00	4.20	5.61	0.75
	1.00	251.48	**5.56E-03**	1.05E-06	1.68E-05	4.52	5.96	0.76
Meteor	0.80	282.18	9.71E-03	5.57E-02	9.01E+00	2.91	3.76	0.77
	0.92	1359.87	3.33E-02	9.34E-03	4.63E+00	4.09	4.87	0.84
	0.95	2334.54	5.21E-02	2.77E-03	6.98E-01	4.48	5.23	0.86
	0.98	5559.88	1.16E-01	5.57E-04	8.23E-01	4.79	5.60	0.86
	1.00	47301.20	9.11E-01	1.06E-06	1.68E-05	5.19	5.98	0.87
Discop w/o recursion (Proposed)	0.80	101.33	5.52E-03	**0**	**0**	1.84	3.84	0.48
	0.92	102.78	5.00E-03	**0**	**0**	2.06	4.83	0.43
	0.95	103.11	4.74E-03	**0**	**0**	2.17	5.29	0.41
	0.98	105.81	4.70E-03	**0**	**0**	2.25	5.68	0.40
	1.00	145.81	6.38E-03	**0**	**0**	2.29	6.03	0.38
Discop (Proposed)	0.80	104.30	**2.99E-03**	**0**	**0**	3.48	3.79	**0.92**
	0.92	104.36	**2.29E-03**	**0**	**0**	4.55	4.86	**0.94**
	0.95	107.07	**2.21E-03**	**0**	**0**	4.84	5.18	**0.94**
	0.98	115.13	**2.17E-03**	**0**	**0**	5.29	5.59	**0.95**
	1.00	362.63	6.29E-03	**0**	**0**	5.76	6.08	**0.95**

Additional time consumption

Task/Model	p	Random Sampling Time (seconds)	Discop Time (seconds)	Ratio
Text Generation GPT-2	0.80	91.21	104.30	1.14
	0.92	90.89	104.36	1.15
	0.95	92.39	107.07	1.16
	0.98	95.20	115.13	1.21
	1.00	174.09	362.63	2.08
Image Completion Image GPT	0.80	739.82	741.35	1.00
	0.92	740.57	750.96	1.01
	0.95	742.79	832.52	1.12
	0.98	738.57	763.67	1.03
	1.00	752.24	759.24	1.01
Text-to-speech WaveRNN	0.80	2500.56	2679.60	1.07
	0.92	2522.93	2599.81	1.03
	0.95	2520.88	2649.25	1.05
	0.98	2537.15	2700.39	1.06
	1.00	2744.50	3582.87	1.31

Evaluation axes

Security: Ave KLD, Max KLD (bits/token)

Time Efficiency: Ave Time (seconds/bit)

Capacity Efficiency: Utilization

Discop: Provably Secure Steganography in Practice Based on "Distribution Copies"

Jinyang Ding[*] Kejiang Chen[*] Yaofei Wang[+] Na Zhao[*] Weiming Zhang[*] Nenghai Yu[*]

[*]University of Science and Technology of China [+]Hefei University of Technology

source@mail.ustc.edu.cn {chenkj, zhangwm}@ustc.edu.cn

Summary

- **Analyzed** the practical issues of existing PSS methods
- **Introduced** a novel PSS method based on "distribution copies"
- **Improved** the embedding capacity to ~0.95 of its theoretical limit
- **Conducted** deployments, benchmarking and comparison

THANK YOU