Defending Against Adversarial Attacks in Speaker Verification Systems

Li-Chi Chang, Zesheng Chen, Chao Chen, Guoping Wang, and Zhuming Bi

Purdue University Fort Wayne

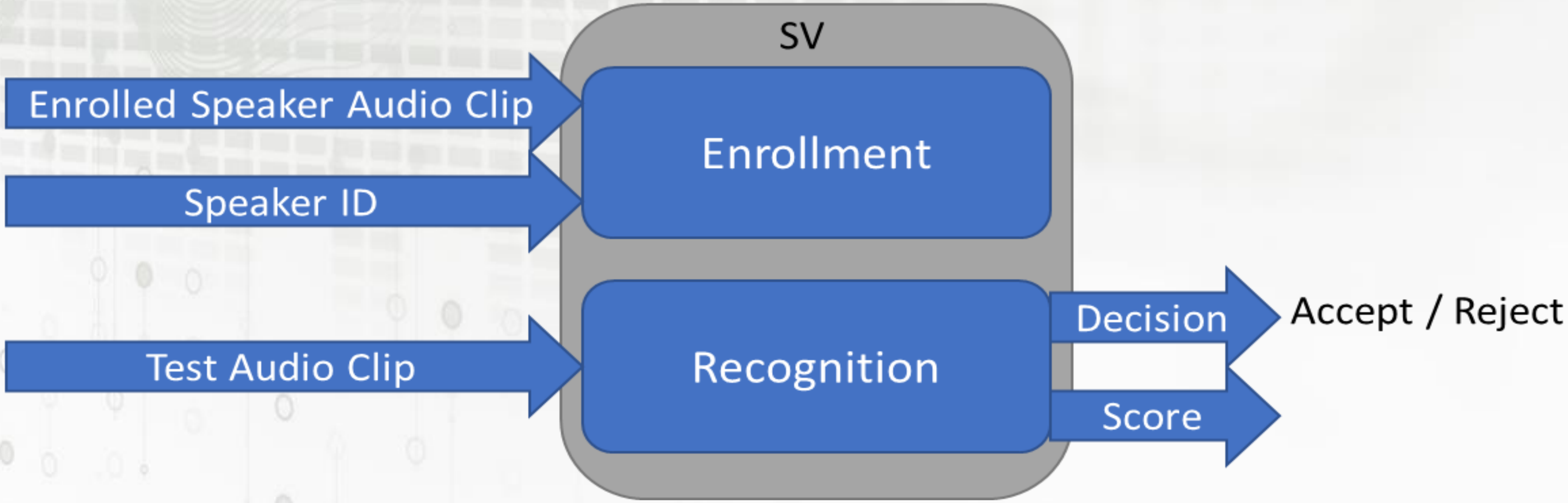
Outlines

- Motivation
- Our Proposed Defense System
- Experiments
- Conclusions and Future Works

Motivation
- Speaker Verification Systems

Speaker verification systems are important to apply human voice as biometrics
- Accurately identify a legitimate user
- Avoid illegal access

Speaker Verification Systems
- GMM
- I-Vector
- D-Vector
- X-Vector

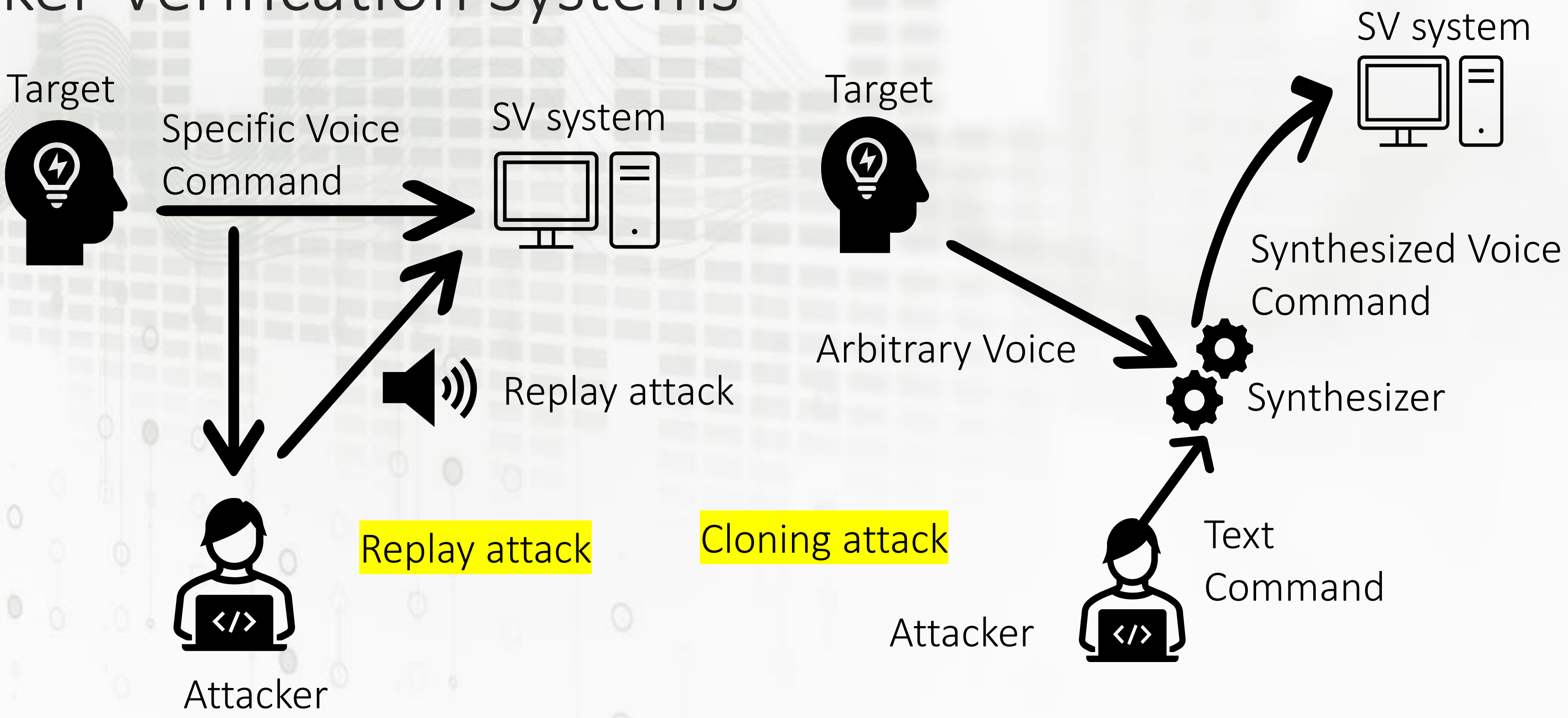


Motivation
- Attack Against SV Systems

There are many attacks targeted on the speaker verification systems.

Attacks against Speaker Verification Systems

- Replay attack
- Cloning attack
- Adversarial attack

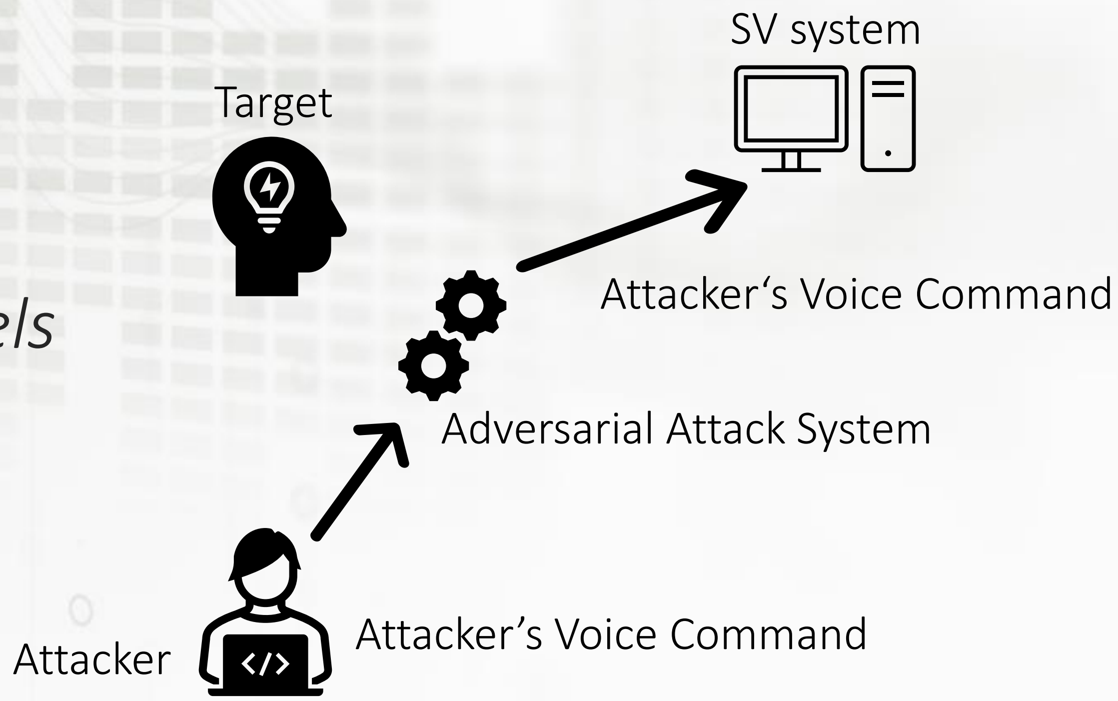


Motivation
- Adversarial Attack Against SV Systems

There are many attacks targeted on the speaker verification systems.

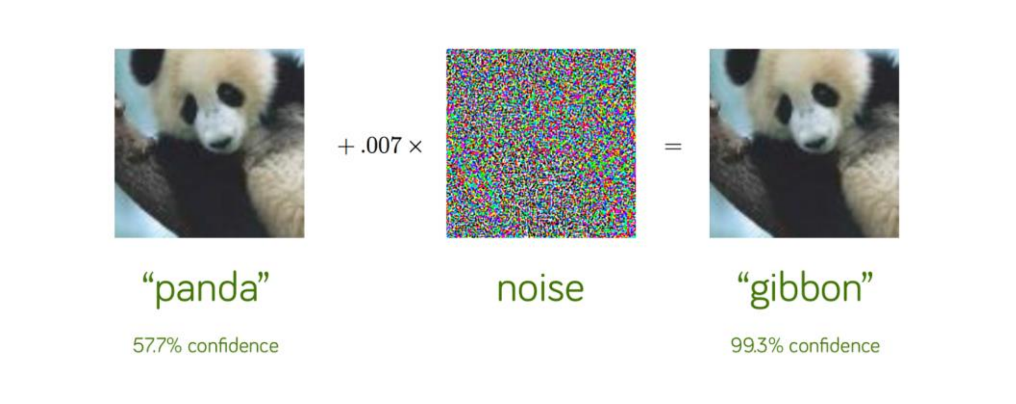
Attacks against Speaker Verification Systems

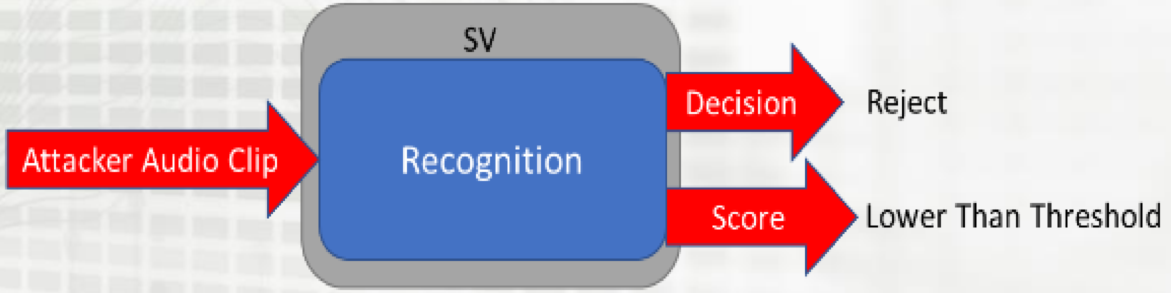
- Replay attack
- Cloning attack
- Adversarial attack
  - *Machine learning or deep learning models*
  - *Most dangerous*
  - *Very difficult to detect and defend*



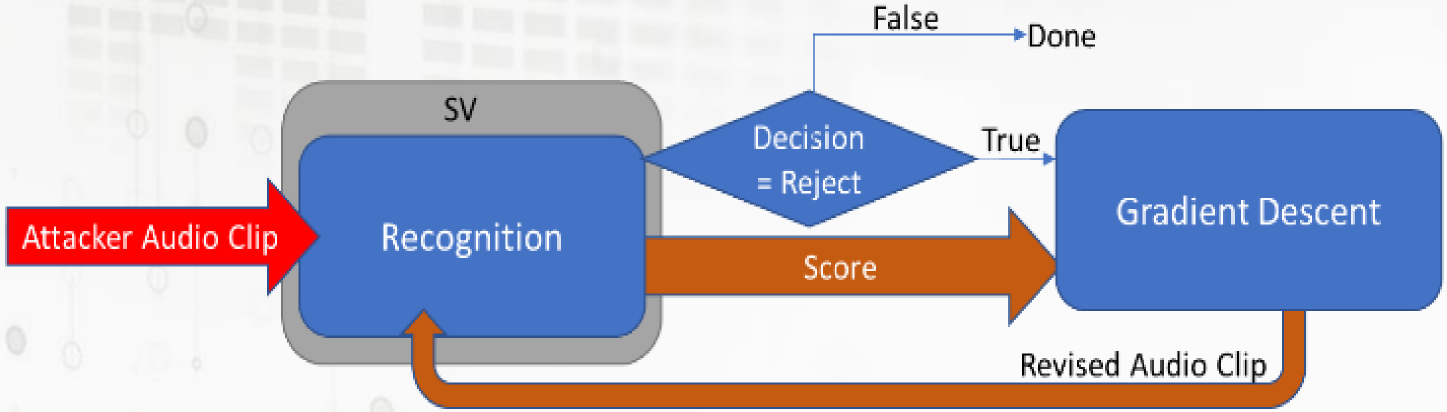
Motivation
- Adversarial Attacks

Attack the weakness of machine learning and deep learning models (Goodfellow, Shlens, and Szegedy ICLR 2015)





High Attack Success Rate (ASR)



Motivation
- FakeBob Attack

G. Chen, S. Chen, L. Fan, X. Du, Z. Zhao, F. Song and Y. Liu, "Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems," in IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 2021

One of adversarial attacks on SV systems

~99% ASR

Attacker Original Audio

Attacker Adversarial Audio

Algorithm 3.3 *FakeBob Attacks*

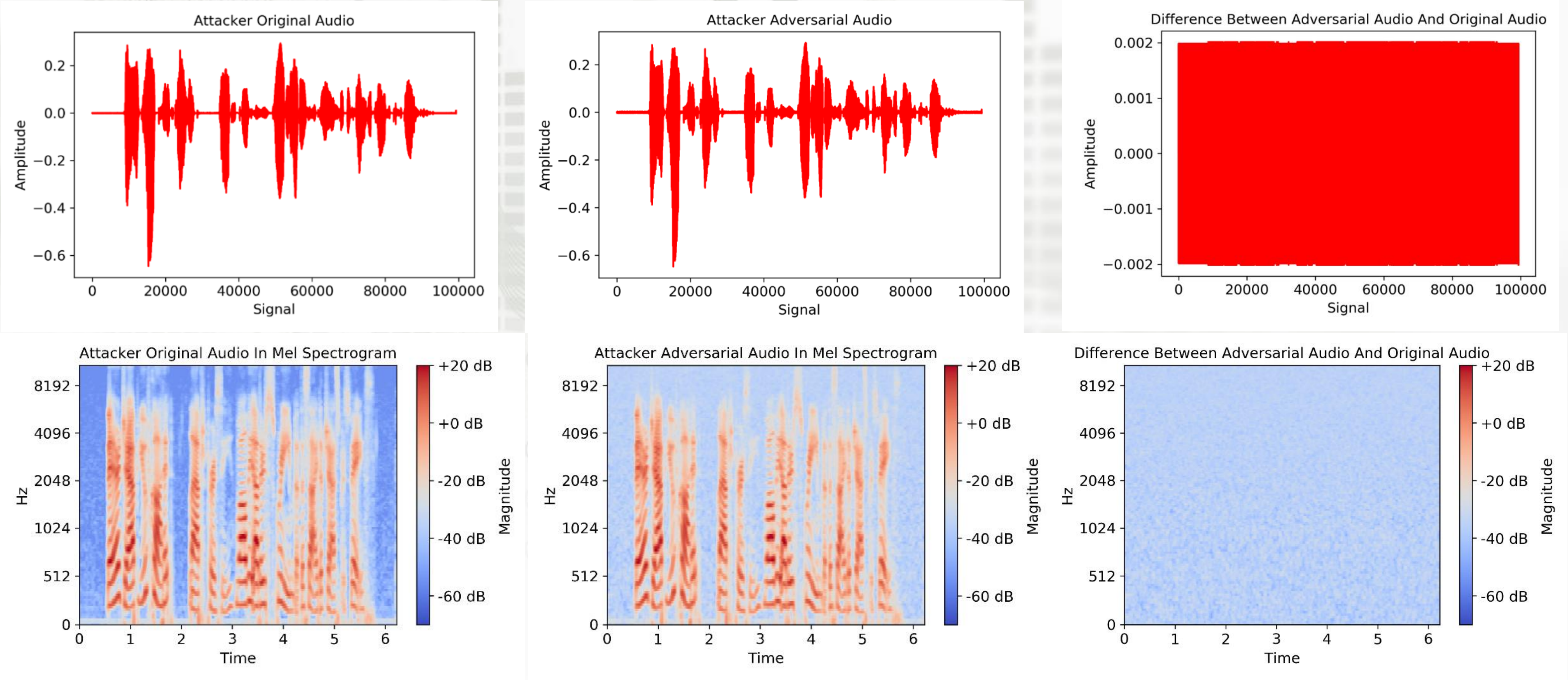
Input: *an audio signal array, threshold of target SV system*

Output: *an adversarial audio*

Require: *Threshold of target SV system* θ, *Audio signal array* A, *Maximum iteration* m, *Score function* S, *Gradient decent function* f_G, *Clip function* f_c, *Learning rate* lr, *Sign function* f_{sign}

$begin$
    $adver \leftarrow A$
    $for\ i = 0;\ i < m;\ i++:$
        $score \leftarrow S(adver)$
        $if\ score \geq \theta:$
            $return\ adver$
        $end\ if$
        $adver \leftarrow f_c(adver - lr \times f_{sign}(f_G(adver)))$
    $end\ for$
end

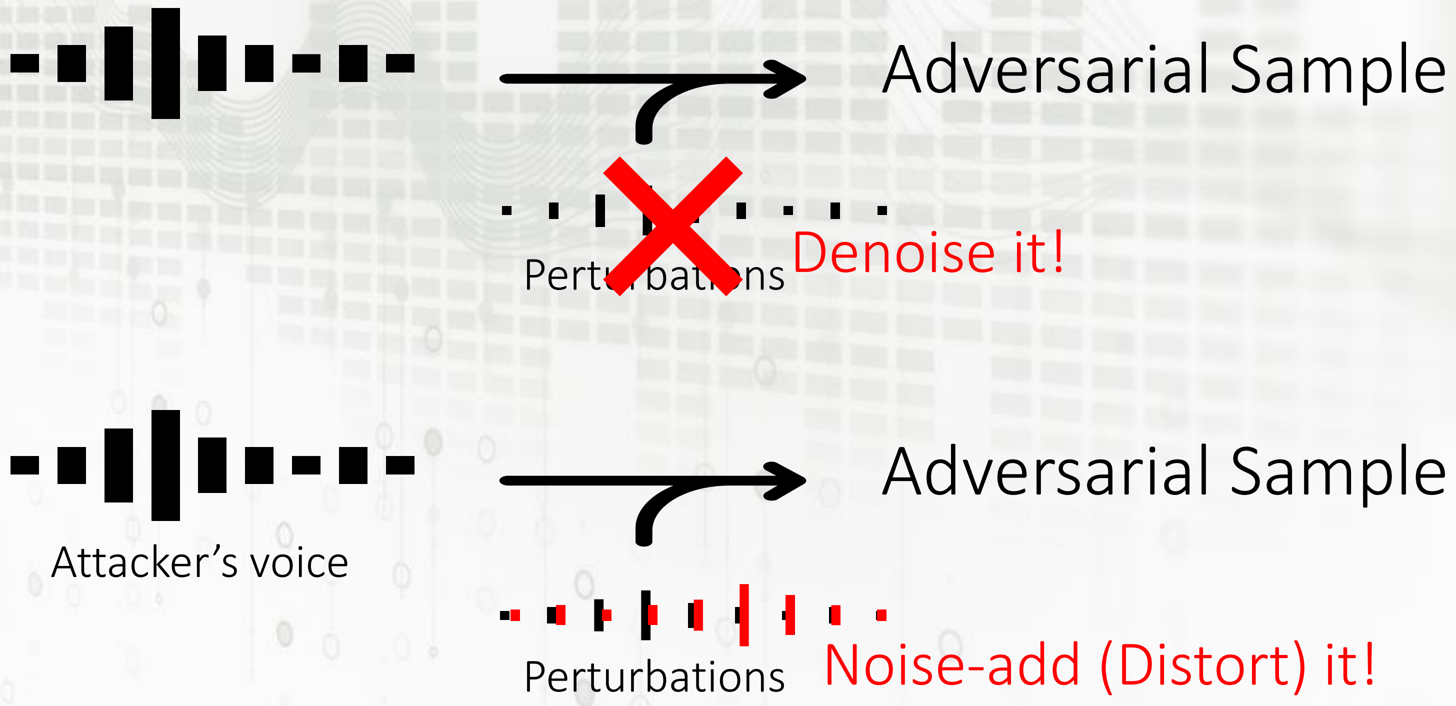
Motivation
- Perturbations



Motivation
- Intuition

Noise-like

Unique



Our Proposed Defense System - Goal of Our Approaches

Simple

Easy to implement

Compatible with any existing SV system

Modalized

Light weight

Low computation load

Real-time task

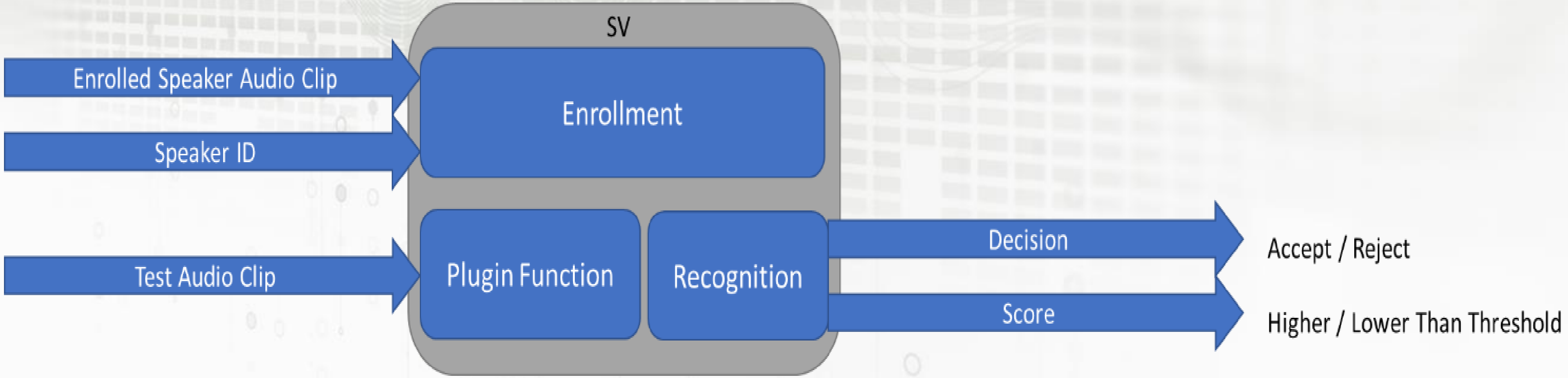
Effective

Greatly increase the adversarial processing time

Reduce the attack success rate

Our Proposed Defense System
- Defense Systems

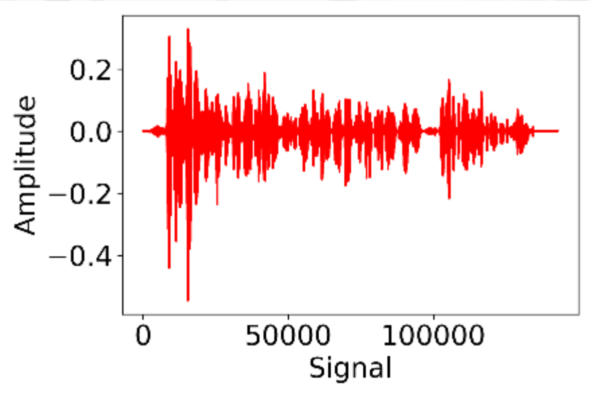
Plugin functions
- Denoising
- Noise-Adding



Our Proposed Defense System - Denoising Plugin Effect

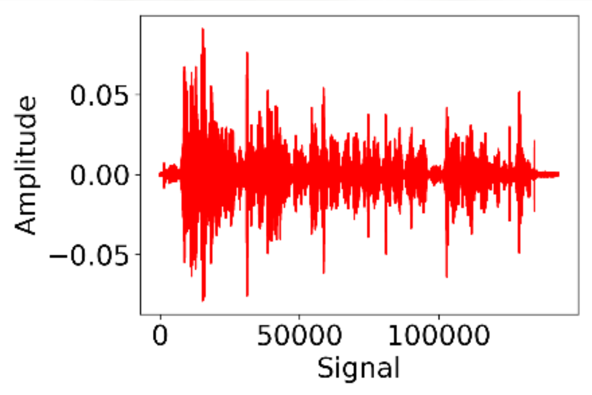
Referring from: T. Sainburg, "timsainb/noisereduce: v1.0," Zenodo, 2019. [Online]. Available: https://github.com/timsainb/noisereduce

Denoised Audio



(a) $\sigma = 0.001$

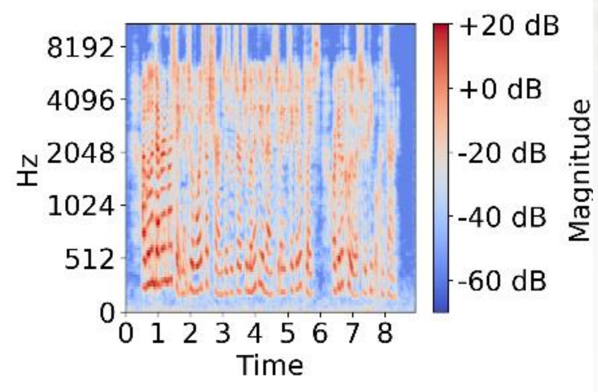
Difference Between Denoised Audio and Original Audio

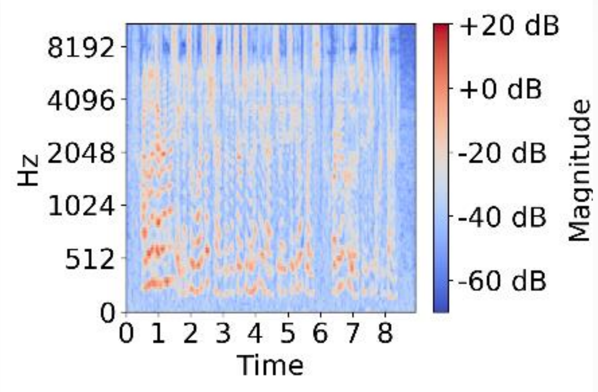


(a) $\sigma = 0.001$

Original Audio

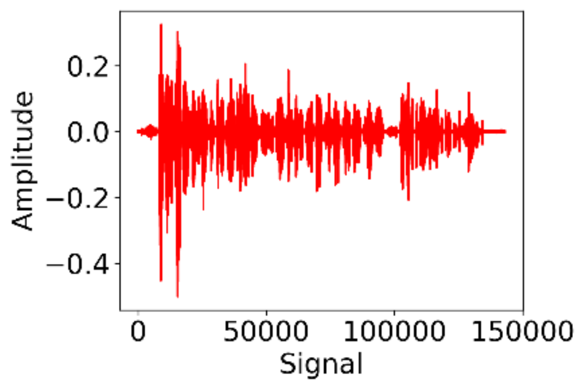
Denoised Audio





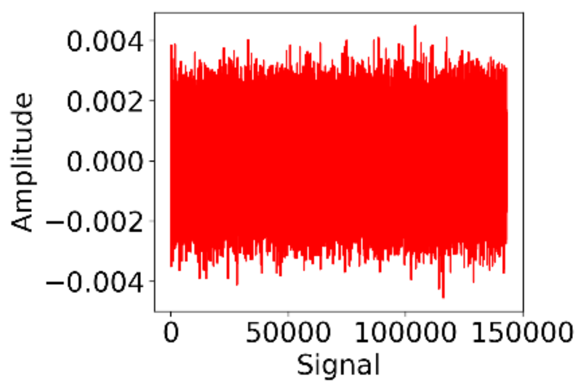
Our Proposed Defense System - Noise-Adding Plugin Effect

Noise-added Audio



(a) $\sigma = 0.001$

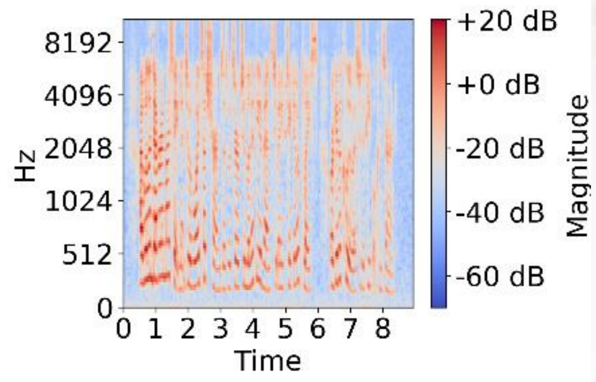
Difference Between Noise-added Audio and Original Audio

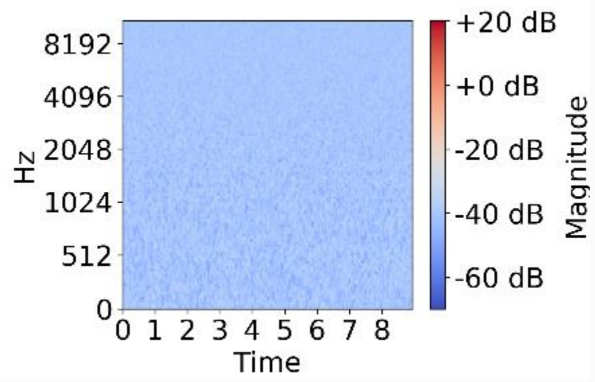


(a) $\sigma = 0.001$

Original Audio

Noise-added Audio





Experiments - Setup

- Environment
  - *Google Cloud Platform*
  - *Local GPU server*
- SV systems
  - *GMM*
  - *i-Vector*
- Tools
  - *Kaldi speech recognition toolkit*
  - *Pre-trained models from VoxCeleb 1*
- Adversarial Attack
  - *FakeBob*
- Audio dataset
  - *LibriSpeech*

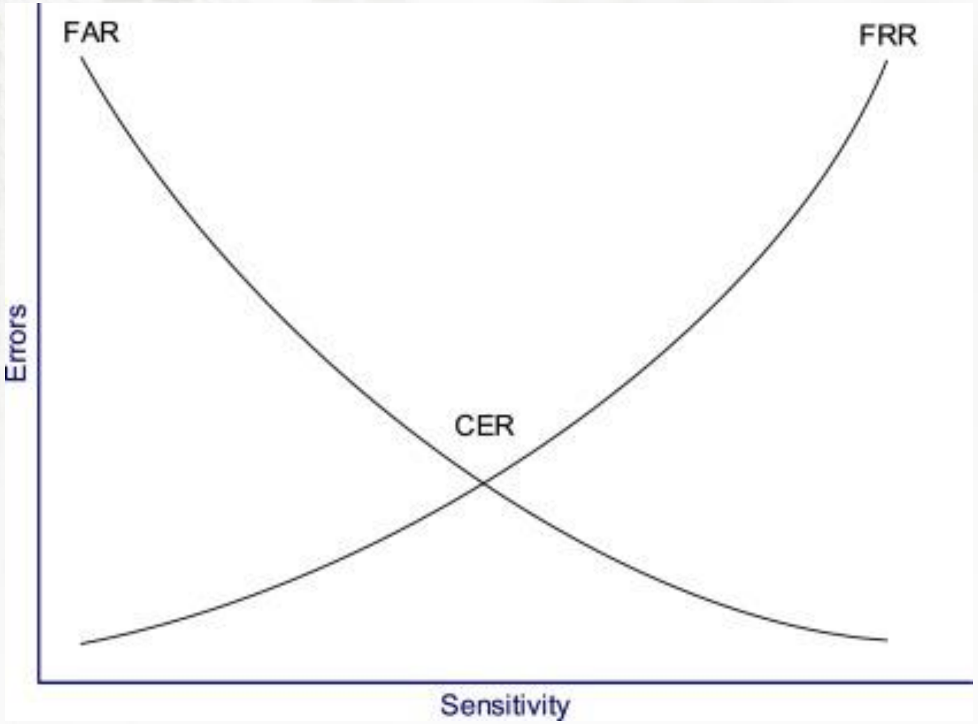
Experiments
- Efficiency Evaluation (Equal Error Rate)

$$\text{EER} = \text{CER} = FAR_i = FRR_j, where\ Threshold(FAR_i) = Threshold(FRR_j)$$

Crossover Error Rate

False Acceptance Rate

False Rejection Rate



Good Performance, low EER

Bad Performance, high EER

Experiments
- Normal Operations in GMM

Plugin	σ	EER (%)	Processing Time (sec)
Original	0	1.05	18.44
Denoising	0.001	1.61	30.67
Denoising	0.002	2.95	30.41
Denoising	0.005	3.36	30.79
Noise-Adding	0.001	1.21	19.34
Noise-Adding	0.002	1.92	19.78
Noise-Adding	0.005	3.94	20.31

Experiments
- Normal Operations in I-Vector

Plugin	σ	EER (%)	Processing Time (sec)
Original	0	0	433.45
Denoising	0.001	0.15	447.37
Denoising	0.002	0.05	447.82
Denoising	0.005	0.49	446.20
Noise-Adding	0.001	0.44	435.35
Noise-Adding	0.002	0.39	435.89
Noise-Adding	0.005	1.14	435.51

Experiments
- Against FakeBob Attacks in GMM

Plugin	σ	Avg Iterations	Avg Time (sec)	Avg ASR (%)
Original	0	23.00	158.68	100.00
Denoising	0.001	18.90	192.02	77.20
Denoising	0.002	22.85	235.96	56.05
Denoising	0.005	22.30	235.78	51.00
Noise-Adding	0.001	92.6	614.92	24.35
Noise-Adding	0.002	604.95	3992.88	5.20
Noise-Adding	0.005	694.95	4350.35	4.10

Max iterations = 1000

Experiments
- Against FakeBob Attacks in I-Vector

Plugin	σ	Avg Iterations	Avg Time (sec)	Avg ASR (%)
Original	0	168.88	6080.47	95.00
Denoising	0.001	97.40	3702.36	55.68
Denoising	0.002	100.58	3825.02	38.63
Denoising	0.005	344.53	13130.24	17.73
Noise-Adding	0.001	556.33	20041.00	8.98
Noise-Adding	0.002	918.23	33017.30	0.50
Noise-Adding	0.005	921.48	33103.39	1.03

Max iterations = 1000

Conclusions

Simple

Modalized as a small plugin

Does not need to change the internal structure of an existing SV system

Light weight

Low computation load

Minor effect on EER

Effective

Reduce the targeted ASR from 100% to 5.2% in GMM and 0.5% in i-vector

Slow down the adversarial attack processing speed 25 times in GMM and 5.43 times in i-vector

Future Works

- Future works
  - X-vector
  - D-vector
- Other type of noise like rustle noise

Thank You