SADIO
(Secure Audio)

A solution for secure audio encoding and storage, with the goal of minimising piracy through illegal access/downloads.

"I don't think the music business is dying. I think we're just experiencing technology and we just have to pass new laws, eventually, to change how music is being distributed…"

~ 50 Cent

PROBLEM STATEMENT

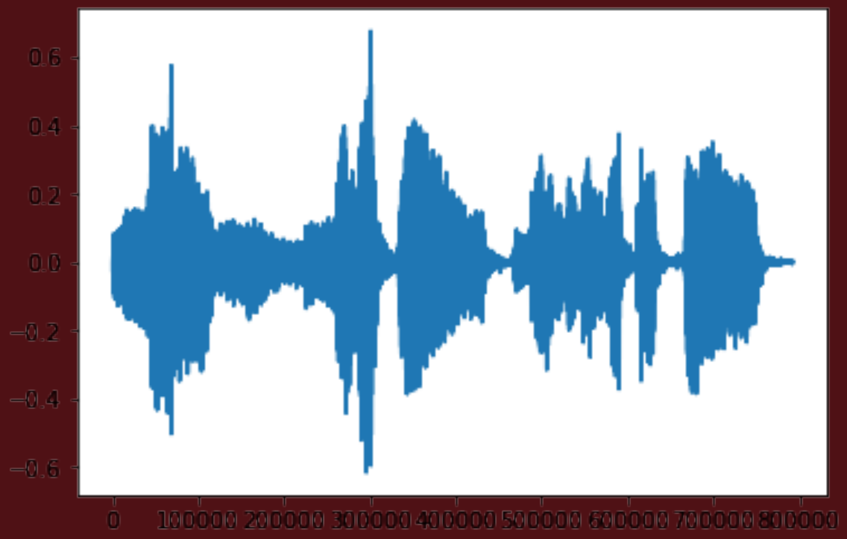
- The source file storage for online media distribution services are single points of failure, prone to attacking/hacking that can lead to the theft of millions of songs.

- Audio files stored in their raw format at the source level of distribution services, can very easily be spread across the internet once accessed, resulting in artists incurring massive loss.

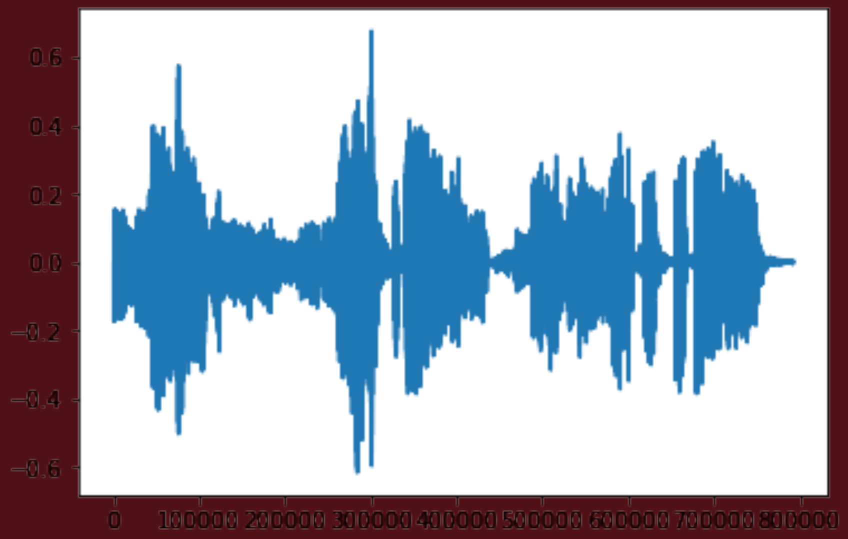
- Recently, a 19-year-old suspect allegedly targeted "award-winning international superstars" by breaking into their websites and cloud-based accounts to access unreleased recorded music. The hacker threatened to release the music files unless the artists paid $150,000.

PROPOSED SOLUTION - PART 1

- A platform where artists can upload their audio files, and also choose a unique key to encode their work. These encoded files are modified versions of the original clips.

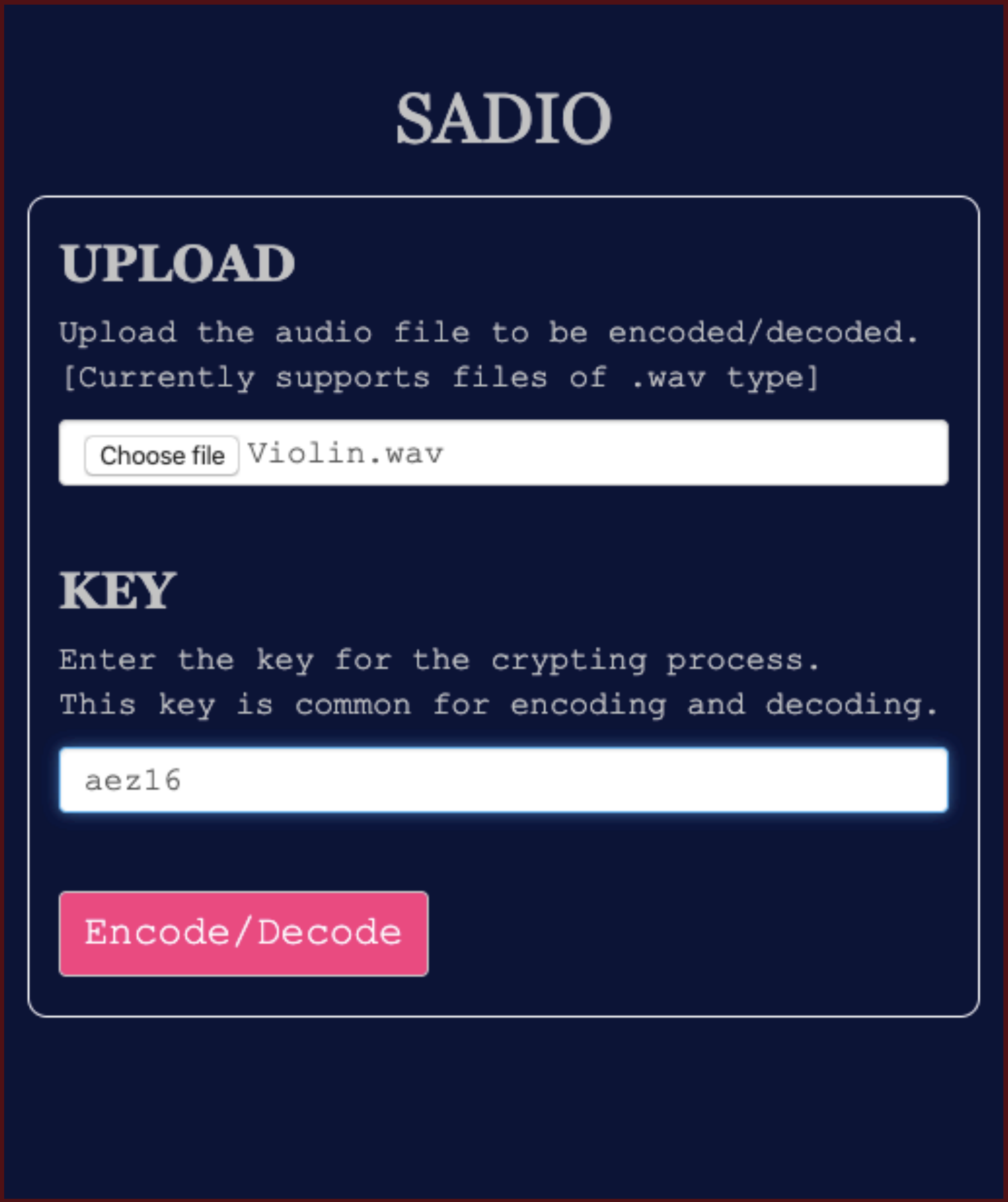


Original
audio sample



Modified (encoded)
audio sample

- The unique key to be used to decode the file, can be shared as a part of the service agreement between an artist and the distribution service.



SADIO

UPLOAD

Upload the audio file to be encoded/decoded.
[Currently supports files of .wav type]

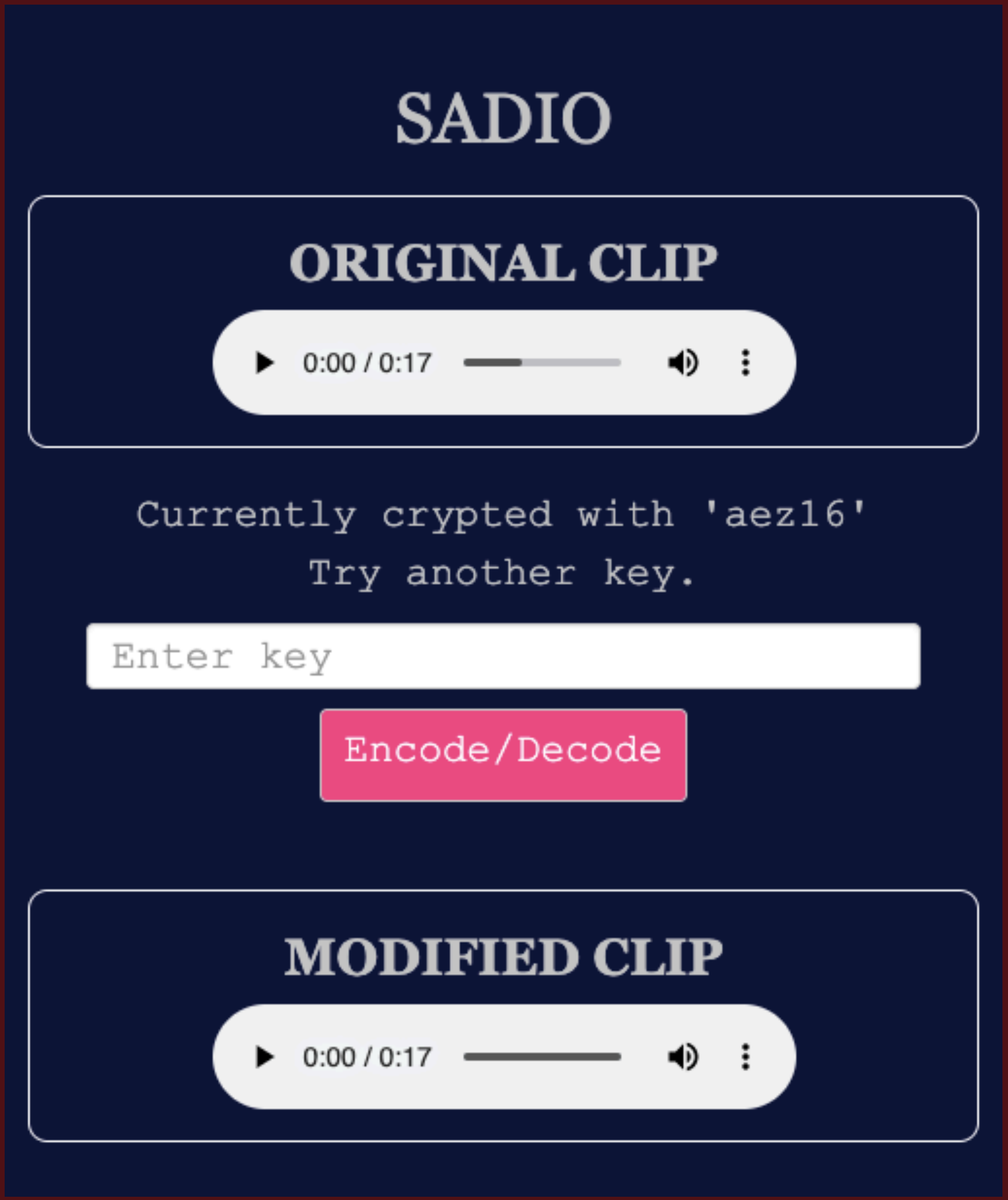
Choose file Violin.wav

KEY

Enter the key for the crypting process.
This key is common for encoding and decoding.

aez16

Encode/Decode



SADIO

ORIGINAL CLIP

0:00 / 0:17

Currently crypted with 'aez16'
Try another key.

Enter key

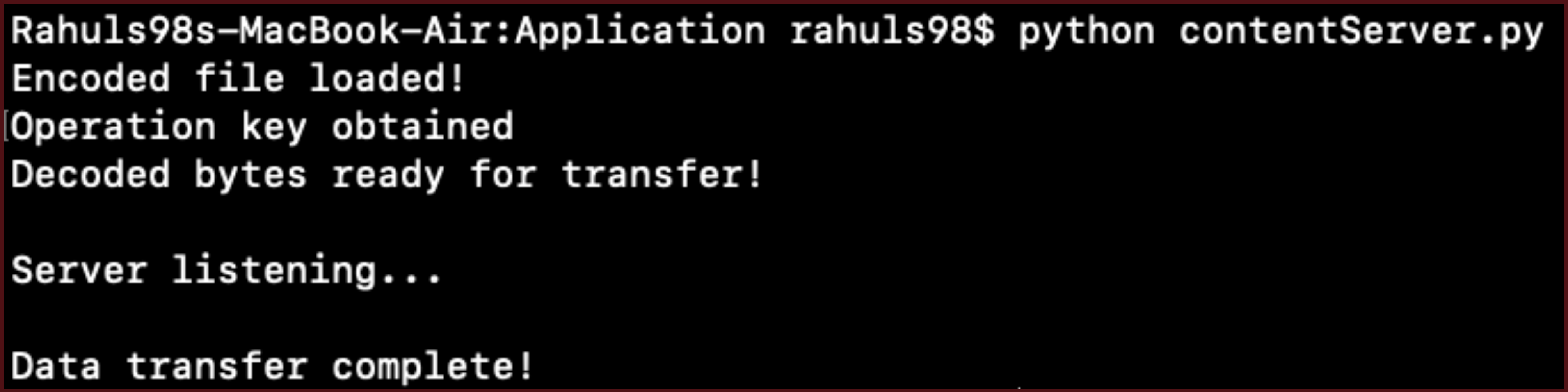
Encode/Decode

MODIFIED CLIP

0:00 / 0:17

PROPOSED SOLUTION - PART 2

- A plug and play solution for media serving hosts and clients, enabling the live decoding of an encoded audio file using the corresponding unique key.

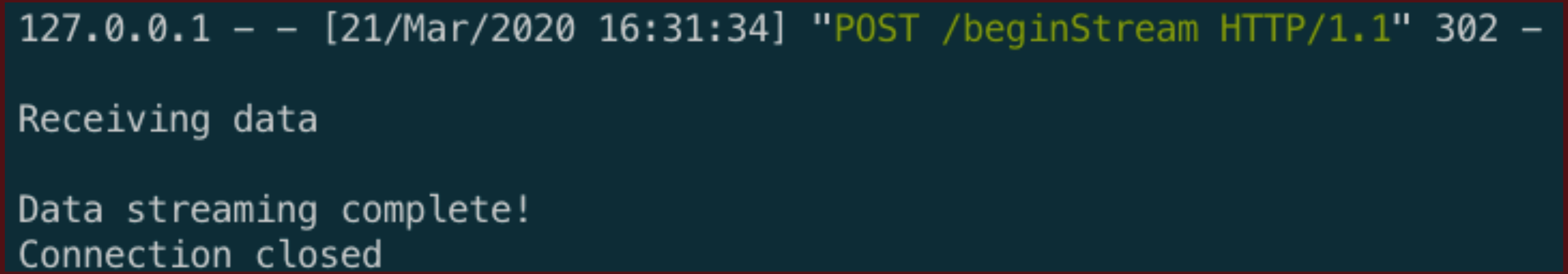


```
Rahuls98s-MacBook-Air:Application rahuls98$ python contentServer.py
Encoded file loaded!
Operation key obtained
Decoded bytes ready for transfer!

Server listening...

Data transfer complete!
```

Live decoding at the content server on client's request.



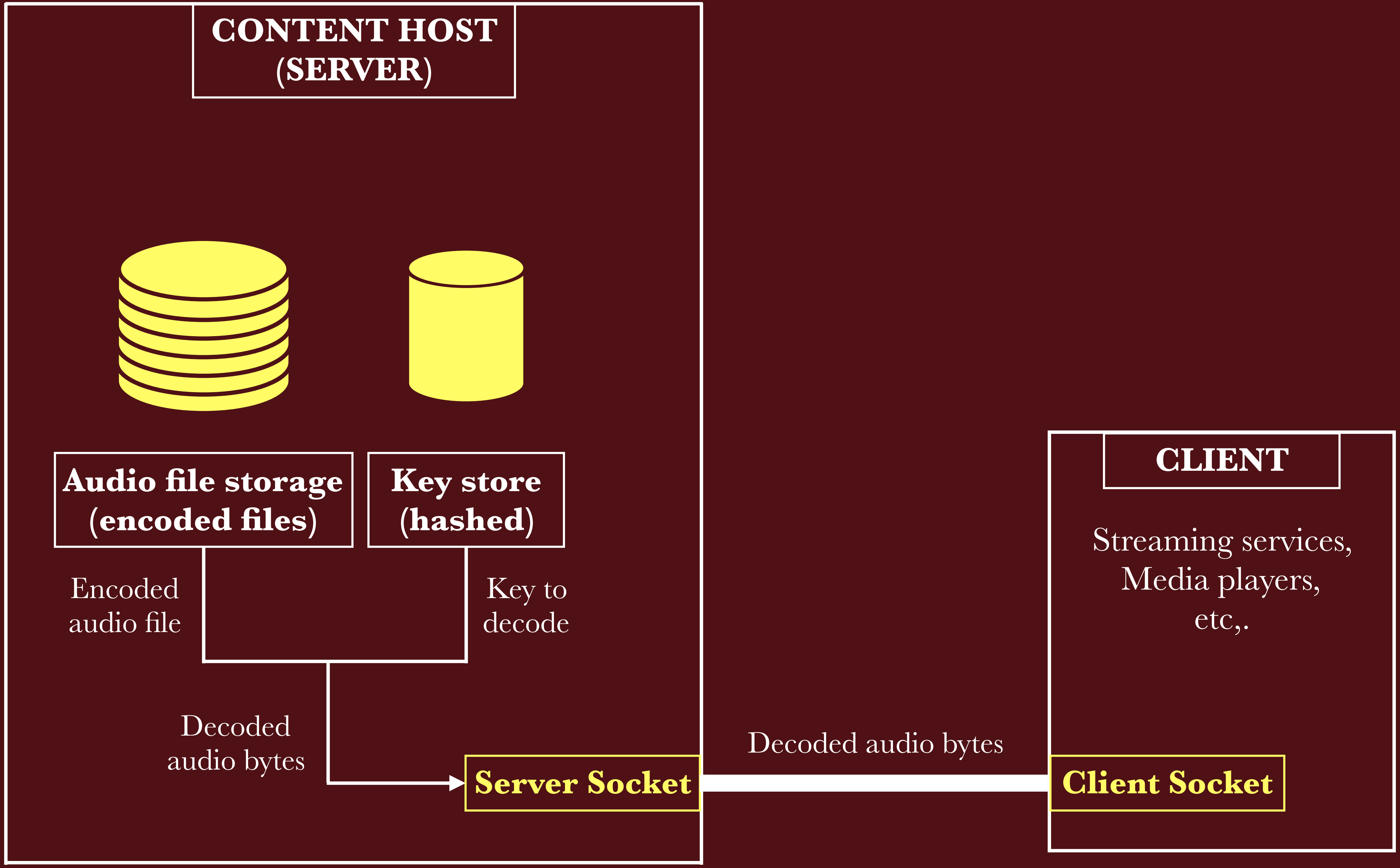
```
127.0.0.1 - - [21/Mar/2020 16:31:34] "POST /beginStream HTTP/1.1" 302 -

Receiving data

Data streaming complete!
Connection closed
```

Client receiving decoded bytes and streaming the media.

- In the events of illegal download attempts from the host, without the right keys for decoding, the attacker only gets access to modified files which are useless.



Thank You.