https://doi.org/10.2352/EI.2023.35.4.MWSF-377

Privacy preserving leak detection in peer-to-peer communication

Julian Heeger, Simon Bugert, Waldemar Berchtold, Alexander Gruler, Martin Steinebach; Fraunhofer Institute for Secure Information Technology SIT — ATHENE - National Research Center for Applied Cybersecurity

Abstract

During the pandemic the usage of video platforms skyrocketed among office workers and students and even today, when more and more events are held on-site again, the usage of video platforms is at an all-time high. However, the many advantages of these platforms cannot hide some problems. In the professional field, the publication of audio recordings without the consent of the author can get them into trouble. In education, another problem is bullying. The distance from the victim lowers the inhibition threshold for bullying, which means that platforms need tools to combat it. In this work, we present a system, which can not only identify the person leaking the footage, but also identify all other persons present in the footage. This system can be used in both described scenarios.

Introduction

In this paper, we propose a system for leak verification and identifying the parties involved, in a highly fluctuating peer-to-peer video conferencing system, using digital watermarking. The main objective of a peer-to-peer videoconferencing system is to preserve the privacy of the users, and thus contrasts with other well-known solutions that typically neglect exactly this. All video and audio connections are realized directly via peer-to-peer, as the server does not learn anything about the newly created conference.

Video and Audio recording in video conferences have become increasingly common in the last three years as we have been working almost virtually worldwide. Many events remain virtual or hybrid and therefore the need of leakage prevention is still relevant. In addition to data leaks, users must also be given the opportunity to record conversations and report cyberbullying. As the study from Vogels [9] shows, nearly half of the US teens have already experienced cyberbullying or harassing. With this approach, we would like to investigate whether victims can be given the opportunity to report this with the available means and with material that is as reliable as possible if the cyberbullying has taken place in a video conference. Audio watermarking gives the opportunity to find a responsible person after a leak and thus is a meaningful solution for this application. The main challenge is that the watermarking algorithms designed for mastered audio files where the audio is usually not mixed afterwards. In this application we face two major problems. First, the watermarking algorithms are based on a symmetric key and secondly all incoming audio streams from each channel are mixed for each participant. This usually weakens the watermark or even erase it completely. In the conceptual design, both aspects must be considered. Our used watermark approach together with a tailored key exchange is one possible solution to meet these challenges.

Related work

There are several technologies that can be used to detect cyberbullying in video conferences. One option is to use artificial intelligence (AI) and machine learning (ML) to automatically analyze the language in chat messages and audio recordings to detect potentially offensive or harassing content. Another option is to use moderation tools that allow moderators to manually monitor chat messages and audio recordings and report or remove inappropriate content. There are also tools that use emotion recognition to identify inappropriate behaviors in people's communications, by analyzing nonverbal signals such as facial expressions, tone of voice, and body language.

In addition to detecting cyberbullying, it is also necessary to be able to prove that the recording is authentic and thus has taken place on the platform at a certain point in time. Digital watermarking methods are a suitable solution for this, because they can covertly embed data into an audio stream. Thus, they allow for the identification of a person behind a leak.

Cecillon et al. [1] use an approach that combines content- and graph-based characteristics in order to benefit from both information sources to recognize harassing content. Their research on unprocessed chat logs demonstrates that messages' dynamics within a conversation — in addition to their content — contain partially complementary information that can be used to increase performance on an abusive message classification job. This approach can be combined with a speech-to-text conversion solution and then used in audio.

Ye et al. [2] is one out of a huge number of research in the field of speech emotion recognition systems to let a machine understand human emotion during a conversation. The accuracy rates of the best speech emotion recognition systems are in the range of 95%. Detecting emotions and mental state of a person based on the body language is done by e.g. Singh et al. [3] and Slogrove et al. [4]. The accuracy rate in this field is around 92%. The video is not always transmitted and thus the technology does not impose itself to be used in this context.

All of this technology to analyze the content can help with video conferencing systems that have a server through which all communication is routed. The most promising approach is the detection of harassing content from text, so it can be applied not only to chat messages, but also to audio using speech-to-text techniques. However, in peer-to-peer solutions, the mentioned solutions can be used, but it must run on each client. If a client is cyberbullied, it must record the chat, audio and video. But since there is no server as central instance, the authenticity is not verifiable from the recording and the digital watermark is necessary. In this work we consider only the use of watermarking in the audio stream, since this is the integral part of a video conference.

Research on digital audio watermarking has been going on for over two decades [7] and there are many algorithms that can

be used. We use an algorithm based on the work of Zmudzinski and Steinebach [6] and [5].

Platform

During the pandemic, we built a proof-of-concept platform called *JAMS* (just another meeting space), which connects different video and collaborative tools on one platform to work more efficient with each other remotely. The intended target market includes educational institutions and businesses with a large percentage of home office as conferencing solution. While conducting our tests, we considered potential issues that might occur in these environments and considered solutions that would address these issues directly and not merely as an additional feature. One of the results of these considerations is the use of audio watermarks in the meetings, while at the same time ensuring the privacy of the users.

Figure 1: Example map for JAMS

Users on *JAMS* have an avatar, which they can move on a 2D map similar to a video game, as seen in Figure 1. Figure 2 illustrates the group membership. If two users U_1, U_2 are close to each other, a video connection is established. When a third user U_3 moves close to the two, he also shares video and audio with the previous two. If one user U_2 moves out of the group range, displayed as a gray area, he will be removed from the group and will stop sharing his video and audio. This allows for ad-hoc conferences between multiple users to be set up and broken down quickly. For smaller groups of around 6-8 users, a peer-to-peer connection using *WebRTC* [8] is used to interconnect all members. Typical conferencing software is embedded into the map for larger gatherings. A server is used to authenticate users, assigning each a unique identification number (**id**), and as a signaling server for the *WebRTC* connection. After a successful connection between to peers the server the server is no longer needed.

Requirements

In this section we describe the properties that a watermarking algorithm must possess in order to be used in our solution. The main requirements for audio watermarking are

Imperceptibility/Transparency: In order to avoid lowering the audio quality and upsetting the listener, the watermark should be challenging or impossible to spot.

Robustness: The watermark should be able to withstand typical audio post-processing procedures like compression, recording with a microphone by an external device and noise reduction without losing it.

Capacity: The watermark should be able to store enough data for a unique identification of each user.

Security: To protect the data it contains, the watermark should be hard to remove or change by unauthorized parties.

Although the other standards must also be addressed, the security of the watermark is the main concern in this work. In particular, certain attacks are to be prevented by the watermark. These includes the scenario where a user wants to frame an other user or a spreader of hate speech try to delete or fake the watermark. Even if a user add noise or use a bandpass filter, the watermark should remain readable. The mentioned scenarios are relevant for cyberbullying as well as in the event of a leak.

The watermark algorithm needs to provide enough capacity so that a watermark can be detected reliably in a 6 second audio snippet independent the number of participants. The watermark should be robust and thus detectable even after it is down-sampled to 8kHz, as the audio quality of speech is acceptable at this sample rate. Among all the requirements mentioned, the watermark should be imperceptible, so that it is not noticed by the participants.

Proposed Approach

In this section, we outline the suggested strategy, which is broken up into two separate processes. The first process is the embedding of a message with a key into the audio stream of a user. The second describes the extraction of the message from the audio stream, when a someone wants to determine the people present in a recording.

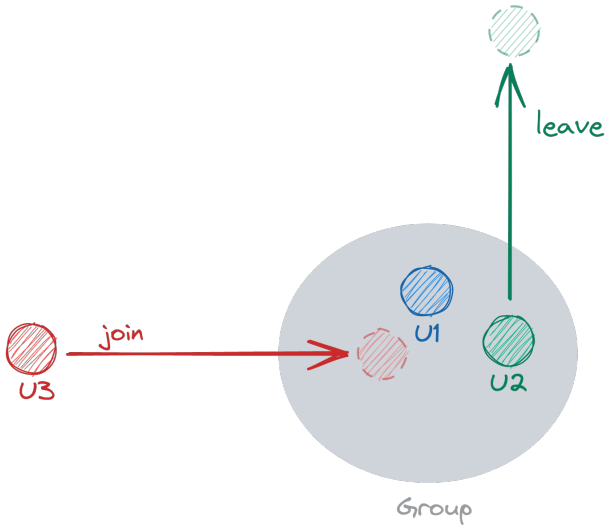

Figure 2: Group proximity defines the membership

Embedding

Two or more users start a group by moving in close proximity. The server is used to handle the *WebRTC* connection establishment and allows both users to exchange information, like their **id**. For testing purposes the id is an 8-bit positive integer, which uniquely identifies a single user. The message, which will be embedded, consists of a concatenation of the group members ids, as seen in Figure 4. The order of ids is irrelevant and only determined by the order in which the users connect to the group. If one of the members disconnects the message is updated and its id is removed. Each time a user logs in, they produce the key that is needed to embed the message into the audio stream and send it to the server to be stored in a database along with the current time, accurate to the day. The database does not associate the password with a user, as its only needed in case of the extraction process. The daytime accuracy of the time field allows for faster lockups after an incident. As a result, each stream sent to the other group members has the same message and key, as seen in Figure 3.

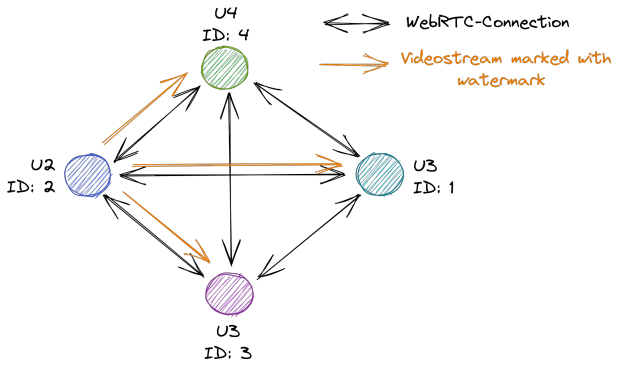

Figure 3: Group connection

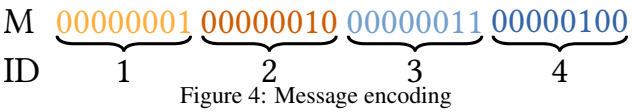

Figure 4: Message encoding

Extraction

The extraction is based on two different scenarios. The first being a leak occurred and a recording surfaces on the internet from an meeting on our platform. In this scenario the relevant party to identify is the user, which created this recording. To do this the keys stored in the database are used to extract the message. Multiple keys should return a similar message, as each message contains the members of the group minus the embedder. Based on this information the only group member in every message is the one, who leaked the meeting.

In a cyberbulling case the relevant party to identify are the other users, who were present during the mobbing. As the user know when the meeting took place, the number of keys to check is reduced. With the message the ids of group members can be extracted and looked up in the user database.

Evaluation

Since our system is completely implemented in the user's browser and thus under their control, a wide variety of attacks must be considered. In doing so, the performance of both the embedding of the watermark and the defense against attacks must not be compromised. Even a short delay of the audio track leads to a noticeable desynchronization with the video track on the receiving side.

An obvious problem is the overlapping of multiple audio signals. Our requirement for the digital watermark is a robust recognition of the message from the resulting audio.

Attacks

Based on the scenarios described in the introduction, we identified attacks against which the solution should be resistant. Some attacks could be applied to both scenarios, while others are only relevant to one. To present a general solution, the solution should still be resistant to all described attacks.

Delete Watermark

A participant spreads hate speech in a video conference. This is recorded by another participant as a watermarked audio track. Using common audio effects (such as filtering, time stretching, downsampling) applied to the marked audio track, the attacker tries to make the watermark non-readable to prevent the hate speech from being attributed. The attack takes place offline, i.e. not during the video conference.

Noise and Collusion Attack

The attacker sends a jamming signal (e.g., white noise or sine tone) during a hate speech that is taking place in real time, with which he overlays the marked audio track of the hate speaker. This interference signal can be sent either permanently or specifically only during the hate speech. The overlay takes place locally at each participant, where the incoming audio streams of the hate speaker and the attacker are mixed together. The goal of the attacker here is to make watermark detection more difficult for recordings consisting of this mixed overall audio track or to make the watermarks illegible. In the collusion attack, the attacker send from two different sources e.g. with a second device controlled by him or his accomplice. The audio of two different sources are marked with different keys and watermark messages. The incoming audio is mixed and thus the energy of the signals as well as each watermark are lowered.

Re-embed Watermark

The attacker wants to frame a user for hate speech that he or she did not commit. To do so, he first takes part in a video conference where the chosen victim speaks for a longer period of time. He then has a watermarked audio track of the victim. Offline (after the conference), he edits the audio track so that it sounds as if the participant speaking in it is committing hate speech or is violates the guidelines of the platform operators. Depending on the recording material, this can be easier or more difficult. Since the marked track is only cut and rearranged, the watermarks should leave this process largely unscathed. The attacker now submits the manipulated audio track for analysis, where the watermark of the actually innocent user is detected. It is thus exploited here that the watermark message is constantly the same and serves as a reference to a user/conference.

Upper cutoff frequency	Lower cutoff frequency			
	0	1000	2000	3000
16000	1	1	1	1
8000	1	1	1	1
7000	1	1	1	1
6000	1	0.96	1	0.06
5000	0.866	0.93	0.03	0
4000	0	0	0	0

Table 1: The table shows the relative frequency of detection of a watermark when varying the lower and upper cutoff frequencies using a bandpass filter.

Test setup

We use an audio watermarking solution from Fraunhofer SIT for the tests. It is especially designed for applications in the broadcast and based on various published works (e.g. [5] and [6]), although any other spread spectrum method can also be used. Podcasts were used and cut into different lengths of 3, 6, 10, 15, 20 and 30 second snippets. For each user we use a 8 bit message length. The embedding strength is a parameter and a higher number let the algorithm embed the watermark more robust. While with a 4 the watermark is not perceived, with a 10 the watermark is perceptible but not annoying. The embedding of the watermark has taken place in the frequency band of 1-14kHz.

For each parameter combination 30 tests are performed. For each parameter combination, 30 tests are performed and the probability of success is given as to whether the correct watermark was detected or not. This does not include how often a watermark was detected, but only whether it was detected at least once.

Results

In the following, the individual attacks are performed and the results are presented in tables. The results show the relative frequency of detection success, where 0 means that no watermark was successfully detected and 1 means that a watermark was successfully found in all 30 tests.

Bandpass filter

For the bandpass filter, we used audio snippets with a length of 10 seconds, chose a watermark strength of 8, and a watermark length of 32, simulating four participants in the conversation. The attack could be performed before a leak, with the hope that the watermark would be destroyed while the quality of the audio was still acceptable. In this test, the lower and upper cutoff frequency was varied between 0 and 3kHz heart at the lower and 4-16kHz at the upper cutoff frequency.

Table 1 shows good results as long as 4kHz of the embedded frequency range was not destroyed by the bandpass filter. It does not have a big impact Where the range starts and ends.

Time stretching

To evaluate the impact of time stretch we used different stretch factors. The audio snippets had a length of 10 seconds, we used a watermark strength of 8, and a watermark length of 32, again simulating four participants in the conversation. The attack could be performed from an attacker before a leak.

We lowered to half of the speed and accelerated to twice the speed without trying to bring it back to the original speed. The results in table 2 show only a good performance as long as the speed is not varying a lot. But as soon as we stretched the audio back again, we could detect the watermarks successfully.

Stretch factor	Success rate
0.5	0.16
0.6	0.26
0.7	0.6
0.8	0.76
0.9	1
1.0	1
1.1	0.3
1.2	0
1.4	0
1.6	0
1.8	0
2	0

Table 2: Time stretch performed to the audio files, where the speed is lowered to half of the original speed and accelerated to twice the speed.

Type	Gain (dB)			
	-20	-30	-40	-50
whitenoise	0	0.16	0.86	1
pinknoise	0,5	0.96	1	1
brownnoise	0.7	1	1	1
sine 1000	1	1	1	1
sine 2000	0.76	1	1	1
sine 3000	0.96	1	1	1

Table 3: The table shows the relative detection success of an watermark depending on the noise type and it level as well es jamming signals like sine signals.

Noise and Collusion Attack

If an attacker adds noise to its audio after they captured the audio, they can typically choose between different noise patterns, e.g. white, pink and brown noise. We also tested the scenario where the attacker add a sine tone of different frequencies.

In collusion attacks, attackers send simultaneously so that their signal is mixed on the receiver. This requires at least two attackers. This attack can occur in the case of a leak and in the case of cyberbullying, with the aim of making the watermark unreadable.

For this tests we used audio snippets with a length of 10 seconds, chose a watermark strength of 8, and a watermark length of 32, simulating four participants in the conversation.

Table 3 shows the results with only poor detection results with a loud noise. Besides, white noise is more challenging than brown or pink noise.

In the coalition attacks, 30 different 10-second snippets were used to embed the watermark with different keys. Each file was overlaid with another and in only three cases could no watermark be detected. This corresponds to a success rate of 0.9967%.

Downsampling

To evaluate the effect of downsampling on detection probability, we varied the sampling rate between 8 and 14 kHz. We used again audio snippets with a length of 10 seconds, chose a watermark strength of 8, and a watermark length of 32, simulating four participants in the conversation.

Table 4 shows good results as long as the sampling rate is

Sample rate	Success rate
8000	0
9000	0.03
10000	0.53
11000	0.9
12000	0.96
13000	0.96
14000	0.96

Table 4: The success rate is shown depending on the sampling rate. The audio was down sampled from 44.1kHz the mentioned sample rate as stereo signal.

Duration (seconds)	Strength	Watermark length (bit) 16	24	32	48
3	4	1			
	6	1			
	8	1			
	10	1			
6	4	1			
	6	1			
	8	1			
	10	1			
10	4	1	0.79166		
	6	1	1		
	8	1	1		
	10	1	1		
15	4	1	0.85833	1	
	6	1	1	1	
	8	1	1	1	
	10	1	1	1	
20	4	1	0.90333	1	1
	6	1	1	1	1
	8	1	1	1	1
	10	1	1	1	1
30	4	1	0.99166	1	1
	6	1	1	1	1
	8	1	1	1	1
	10	1	1	1	1

Table 5: Success rates are shown depending on the parameter variations.

above 11kHz stereo. The audio quality below 12kHz stereo is low, but one can understand what was spoken.

Parameter evaluation

In order to evaluate the impact of the different parameters, we varied the audio length between 3, 6, 10, 20 and 30 Seconds, changed the embedding strength between 4, 6, 8 and 10 as well as changed the watermark length. The results in table 5 are not surprising as they reflect the well known fact that the longer the audio and higher the embedding strength as well as shorter the watermark length the better the results.

Conclusion and future work

In this paper we describe a video platform, which marks the audio of a meeting with watermarks to provide a means of dealing with problems such as bullying and leakage of voice recordings. This looks at how, in the aftermath of an incident, the individuals involved can be identified without compromising the privacy of the users. We defined requirements, like robustness and security, our solution needs to address and defined attacks against it. We showed that as long as the 4kHz frequency range was not destroyed our solution is robust against filtering, but stretching the audio too much it reaches a poor detection rate. Furthermore adding noise to the audio only lowers the detection rate if the noise is loud or white noise.

In the future we will develop a new password sharing schema between multiple parties to reduce the number of keys stored in the database and potentially helping the watermark detection and robustness.

An extension of the approach in this work by an automated detection of emotions in speech and body language as well as content detection of hate speech in chats and audio can be added to the video platform and may a meaningful addition. This way, further aspects can be integrated, such as the client not displaying the hate speech in the chat, or the audio being muted at the client to protect the potential victim.

Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) in the Fraunhofer Cybersecurity Training Lab (LLCS) and by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

[1] Cécillon, Noé and Labatut, Vincent and Dufour, Richard and Linarès, Georges Abusive Language Detection in Online Conversations by Combining Content- and Graph-Based Features, Frontiers in Big Data 2019

[2] Ye, Jiaxin and Wen, Xincheng and Wei, Yujie and Xu, Yong and Liu, Kun-Hong and Shan, Hongming. (2022). Temporal Modeling Matters: A Novel Temporal Emotional Modeling Approach for Speech Emotion Recognition. https://arxiv.org/pdf/2211.08233v1.pdf

[3] S. Singh, V. Sharma, K. Jain and R. Bhall, "EDBL - algorithm for detection and analysis of emotion using body language," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 2015, pp. 820-823, doi: 10.1109/NGCT.2015.7375234.

[4] K. Slogrove and D. van der Haar, "Group Emotion Recognition in the Wild using Pose Estimation and LSTM Neural Networks," 2022 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 2022, pp. 1-6, doi: 10.1109/icABCD54961.2022.9856227.

[5] Zmudzinski, S., Steinebach, M. (2009). Perception-Based Audio Authentication Watermarking in the Time-Frequency Domain. In: Katzenbeisser, S., Sadeghi, AR. (eds) Information Hiding. IH 2009. Lecture Notes in Computer Science, vol 5806. Springer, Berlin, Heidelberg.

[6] Sascha Zmudzinski, Martin Steinebach, "Perception-based authentication watermarking for digital audio data," Proc. SPIE 7254, Media Forensics and Security, 725414 (4 February 2009)

[7] J. Cox et al. Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing 6.12 (1997), S. 1673–1687

[8] Ali, H. (2018). Real-time Communication Using WebRTC.

[9] Pew Research Center, December 2022, “Teens and Cyberbullying 2022”