Forensic Behavior Analysis in Video Conferencing Based on the Metadata of Encrypted Audio and Video Streams - Considerations and Possibilities

Robert Altschaffel, Jonas Hielscher, Christian Kraetzer, Kevin Lamshöft and Jana Dittmann

Otto von Guericke University Magdeburg
Magdeburg, Germany
Email: robert.altschaffel@iti.cs.uni-magdeburg.de

***Abstract*—This paper discusses the possibility to perform a forensic behavior analysis on the network recordings of video conferences in order to identify different activities taking part during such conferencing. This behavior analysis is based on the audio- and video streams of such software. While the connections are usually encrypted, the possibility of using and deriving heuristic metadata from the encrypted stream in order to identify various activities (use cases) is explored. This paper shows first results of such an approach to identify various activities, which could then be used to construct a biometric pattern. Furthermore, a model for communication flows during video conferences is introduced, formalizing which specific data can be gathered at various points by an observer. A first case study employs a set of four different test cases applied to two different solutions for video conferencing.**

***Keywords*—Security, Video conferencing, Zoom, Big Blue Button, User and traffic profiling, Network forensics.**

I. Introduction

Video Conferencing (VC) systems are used to communicate using video, audio and text streams. They are used in business contexts, as well as in the private life of many people, receiving additional relevance during the trend of social distancing associated with the COVID-19 epidemic.

This paper evaluates the possibility to perform a forensic behavior analysis on network data captures of VC. This forensic behavior analysis aims at identifying certain activities during VC sessions. The analysis is based on the audio-, video- and text streams common in this kind of software. While the network traffic is usually encrypted, the possibility of using metadata still available, as well as heuristic metadata derived from the encrypted stream has been explored in different application scenarios (see Section II-B). This paper discusses first results of this approach tested with two different VC solutions and the impact these results have for forensic use and also the potential impact on the privacy of the users during such conferencing sessions is discussed.

The paper is structured as follows: Section II of this paper gives an overview on the VC systems used for the first tests discussed in this paper, the impact of biometrics on privacy, approaches used to perform activity identification in encrypted traffic and general considerations on using such an approach during a forensic investigation. Section III introduces the overall approach used during these tests. Section IV describes the tools required to conduct such forensic activity identification. An exemplary case study based on the two selected VC systems is provided in Section V. A discussion of the potential impacts of the results presented in this work as well as a discussion on potential future topics conclude this paper in Section VI.

II. State of the Art

This section gives some general background to establish the foundations of the work done during this paper.

A. Video conferencing

This paper discusses systems used to enable the communication between users utilizing text, audio and video. These systems are referred to as *VC solutions* during the course of this paper (based on the definition of this term in [1]). They are used for business discussions as well as for private conversations. These two use cases establish a clear need of confidentiality of the communication and in some cases also of the resulting metadata of the communication (this includes the identity of the participants or the date and time of the communication). If the contents of the communication are disclosed, private information and business secrets might be revealed. In this paper, we focus on conferencing systems which rely on a central (conferencing) server which handles the communication (in contrast to peer-to-peer systems).

B. Activity and Content Identification in Encrypted Traffic

This section provides an overview on the various approaches designed to identify activities within encrypted traffic.

An early approach was published in 2001 by Song et al. [2]. The study found that the reconstruction of single user inputs in an SSH session is possible based on the packet size and keystroke dynamics. Statistical models to successfully extract biometric features from the encrypted traffic of the TeamViewer application were researched in Altschaffel et al. [3]. White et al. [4] separated encrypted VoIP-traffic into single phonemes and were able to show that at least parts of the encrypted spoken conversations can be reconstructed.

Dupasquier et al. [5] used a dynamic time warping algorithm based on the size of encrypted packets, as well as the timing

 ISBN: 978-1-61208-821-1

information as input in order to predict between 60%-83% of spoken sentences correctly. Korczynski et al. [6] applied statistical protocol identification on encrypted Skype traffic in order to distinguish different types of communication (audio, video, chat, file sharing, etc.). With their approach they were able to predict the type of communication with accuracy between 60.3% and 100%, depending on the type. Zhu et al. [7] used packet inter-arrival-times and packet size as parameters to identify speakers in a Skype session. They also proposed countermeasures to restore privacy, e.g. the harmonization of packet sizes between different speakers. Zhang et al. [8] were able to track Skype users mobility (determine their ever-changing public IP-Addresses), by creating unique fingerprints of their Skype-network-traffic.

Besides the potential risks originating from such transcription attacks and user tracking analyses, Whiskerd et al. [9] demonstrated that biometric user information can also be easily derived from such encrypted communication sessions. In their work they achieved an EER of roughly 5% while doing keystroke dynamics (KD) analyses in the encrypted domain, a performance which, for their setup, is not significantly worse than biometric KD verification on the ground truth (i.e. unencrypted) input.

C. Digital forensics

Digital forensics is defined by [10] as *"[...] the science of identifying and analyzing entities, states, and state transitions of events that have occurred or are occurring"*. In essence, forensics is used to reconstruct events. Digital forensics performs this reconstruction in the digital domain. This event reconstruction might be useful in various contexts, including judicial proceedings.

Forensic proceedings follow a distinct pattern. Forensic process models describe this pattern. In this paper, the forensic process model as described in [11] is used. This model has two primary advantages useful for the approach used within this model. At first, the forensic process described in this model contains a *Strategic Preparation* (**SP**) which consists of activities taken before a specific forensic investigation in order to support or even enable the possibility of said investigation. The other advantage is the use of various *Data Types* which represent groups of data handled in a specific way during a forensic investigation.

During this paper, the enhanced definitions presented in [12] are used. Hence, the following definitions for the *Investigation Steps* are used to describe the forensic behavior identification:

- **Strategic preparation** (**SP**) represents measures taken by the operator of an IT-system, prior to an incident, which might support a forensic investigation
- **Operational preparation** (**OP**) represents the preparation for a forensic investigation after a suspected incident
- **Data gathering** (**DG**) represents measures to acquire and secure digital evidence
- **Data investigation** (**DI**) represents measures to evaluate and extract data for further investigation
- **Data analysis** (**DA**) represents the detailed analysis and correlation between digital evidence from various sources
- **Documentation** (**DO**) represents the detailed documentation of the investigation

This identification is based on data which belongs to a range of different *Data Types*. The *Data Types* relevant for this approach are defined as follows by [13]:

- **Raw data** (**DT2**): *A sequence of bits within the Data Streams of a computing systems not (yet) interpreted.*
- **Details about data** (**DT3**): *Data added to other data, stored within the annotated chunk of data or externally*
- **Functional data** (**DT9**): *Data content created, edited, consumed or processed as the key functionality of the system*

These *Data Types* are related to each other. **DT2** is the pure not interpreted data flow between the systems. This contains some **DT3** in order to facilitate the communications between the systems (in this case network addresses or ports). If the **DT2** could be completely interpreted (incl. decryption and decoding), the **DT9** would be obtained. This would include the video, audio and text streams. This data might also contain annotations (**DT3**). The approach presented in this paper relies on *Strategic Preparation* (providing among others the evaluation setup) and has to address the fact that different *Data Types* are available at different locations of the use case scenarios common with VC systems.

III. Usability of audio and video streams in video conferencing for activity analysis

Section II-B has shown that different activities might lead to a notable change in communication behavior. This change in network behavior might enable the identification of various activities in a communication flow without the ability to interpret the communication flow directly (e.g. by using the audio codec to reconstruct the transmitted audio). Such an approach might be useful in cases in which the used codec is unknown or not available or where such reconstruction is not possible since the communication channel itself is encrypted.

In order to implement such an approach several steps are necessary. At first, the various activities in VC which might lead to discernible difference in communication behavior have to identified. This is performed in Section III-A. After this, the discernible difference in the communication behavior has to be identified. These differences can be used to establish the properties usable for an identification of the various activities. This process is described in Section III-B. Section III-C describes where in the communication structure these properties are available during the use of a VC solution. Section III-D then explores the technical process used in order to identify different activities within communication flows. This approach is based on pattern recognition which evaluates said properties. During the use of the system, these properties are compared to a model created during a training phase. This process is akin to the pattern recognition pipeline.

 ISBN: 978-1-61208-821-1

A. Activities in video conferencing

First considerations on the various activities usually performed in VC are listed here. This list describes some basic behaviors which we suspect to have a notable impact on the observable communication behavior. For example, these activities each require a different amount of data to be transmitted to the other participants during the video conference. This list contains eleven elements at this point of time but is expandable. However, a distinction between more similar activities might require additional insight into the communication behavior and the resulting properties.

Activities in Text

TE_1 inactive / not typing
TE_2 typing
TE_3 sending text

Activities in Audio

A_1 deactivated / muted
A_2 unmute and silent
A_3 unmute and speaking fluently
A_4 unmute and speaking chopped off

Activities in Video

V_1 deactivated
V_2 black screen
V_3 one person in front
V_4 multiple persons in front

B. Features

Various features usable to distinguish activities within encrypted communication sessions have been proposed by the literature examined in Section II-B. A common theme here is the size of the transmitted packets in correlation to the timing information. Our approach uses these two sets of inputs as the foundation for the resulting features. The specific features used in this approach are based on the hierarchy of features proposed in [3]. Here, *Window-based features using fixed time windows* are selected as best representing the results found in other research work presented in Section II-B. These features are based on *Packet features* and aggregated over a specific amount of packets. Hence, the used features consist of information about the specific *packet size*. This is then aggregated to the *minimum*, *maximum*, *mean* and *standard deviation* of *packet size* over a specific window of packets.

However, obtaining these features relies on the possibility to separate the various communication streams as understood by the feature hierarchy. In order to do so, the presence of additional information might be necessary. This information includes identifiable network addresses.

C. Availability of features in the communication infrastructure

Some of the information discussed in III-B is only available at certain points within the communication infrastructure. This is due to the fact that the media streams are encrypted on their path between the various clients and the conference server. In addition, the conference server fuses the various media streams of the clients together and delivers them back to the attached clients. This is illustrated in Figure 1.

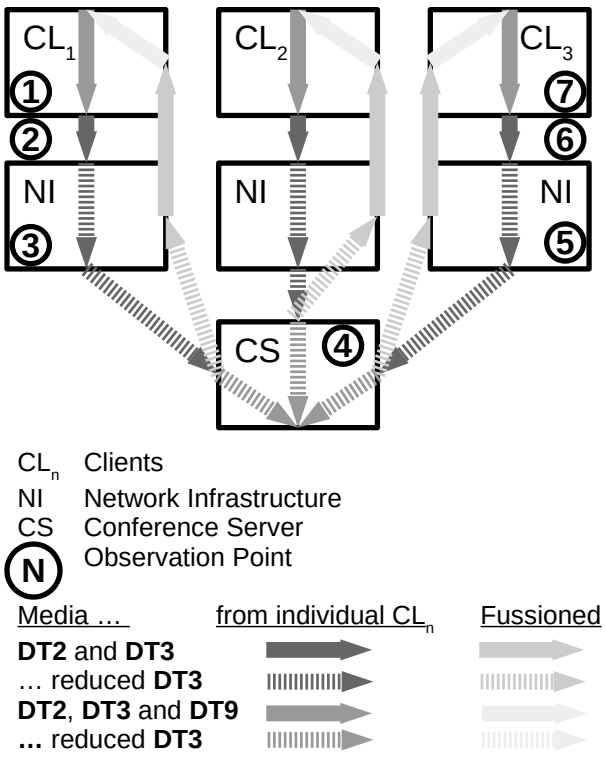

Fig. 1. Communication infrastructure in VC solutions and potential available *Data Types* at various observation points

It is of relevance to understand where which exact *Data Types* (and hence features) are available within this communication infrastructure. Hence, assuming Client CL_1 is the target of the activity identification, the available *Data Types* at the specific Observation Points (**OP**) are:

- $\mathbf{OP}_1$ provides access to the entire unencrypted media streams front Client CL_1 (**DT9**) as well as its representation on the network (**DT2**) and the metadata necessary for the network communication (**DT3**).
- $\mathbf{OP}_2$ provides access to the raw representation of the encrypted media streams on the network (**DT2**) as well as the metadata necessary for the network communication (**DT3**).
- $\mathbf{OP}_3$ provides the same access as $\mathbf{OP}_2$ but might alter some of the metadata necessary for the network communication (**DT3**).
- $\mathbf{OP}_4$ can decrypt the encrypted media streams and has access to all unencrypted media streams from all clients (**DT9**). In addition, access to the individual encrypted streams (**DT2**) and the individual metadata from the network communication (**DT3**) is available, although potentially in a form altered by the network infrastructure.
- $\mathbf{OP}_5$ has access to the encrypted and combined media

 ISBN: 978-1-61208-821-1

streams (**DT2**) as well as to some metadata necessary for the communication from the Conference Server to Client CL_3 which might be altered due to intervening Network Infrastructure (**DT3**).

- **OP**$_6$ has the same access to **DT2** as **OP**$_5$ but has access to the original metadata for the communication between Conference Server and Client CL_3 (**DT3**).
- **OP**$_7$ can decrypt the combined **DT2** in order to achieve the combined media streams from all Clients CL_n (**DT9**). In addition, the metadata from the communication between Conference Server and Client CL_3 is available (**DT3**).

D. Using pattern recognition to identify activities in VC to support a forensic investigation

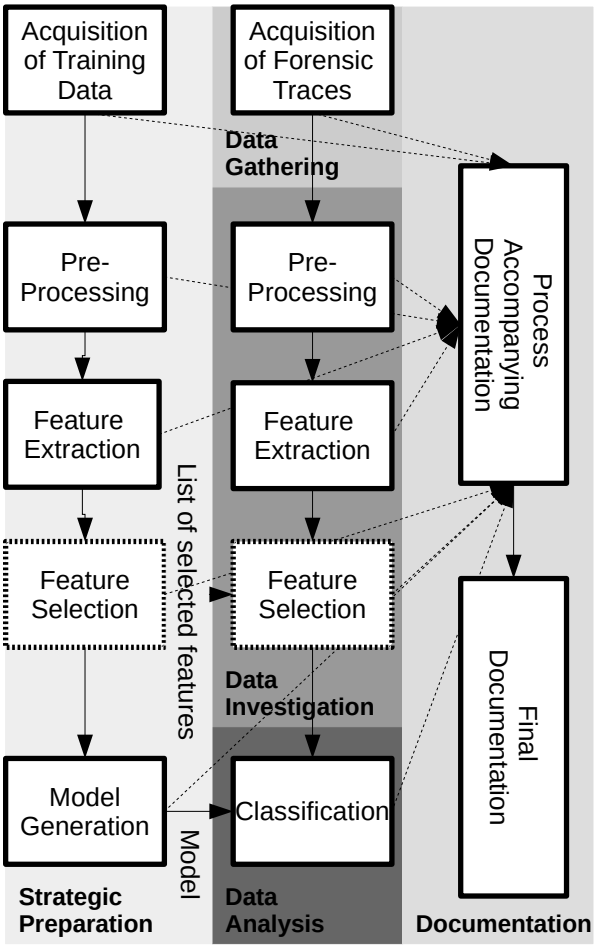

Fig. 2. Mapping of the pattern recognition approach to the forensic *Investigation Steps*, based on [12].

In order to use pattern recognition to identify activities within an encrypted VC session, various activities during the forensic process are necessary.

At first, the pattern recognition relies on a classification model. This model has to be created before a specific forensic investigation takes place. This has to occur during the **SP**. In addition to the steps already described in the pattern recognition pipeline, the acquisition of training data is a complicated topic and requires special attention. Section III-C discussed the availability of features at various spots within the communication infrastructure. The training data should be as close as possible to the data acquired during the forensic investigation in order to achieve an usable classification model. Hence, the training data should be captured at the spot most likely used by the forensic investigator during a forensic investigation. The rest of the **SP** covers the creation of the model which involves the Pre-Processing, Feature Extraction, Feature Selection and the Model generation itself.

Once a specific forensic investigation is started the potential forensic data sources are identified during the **OP**. This leads to the exact locations where forensic data is acquired during **DG** and will aim at acquiring the greatest extent of possible data while maintaining integrity and authenticity of the captures. Hence, a capture location which is entirely under the control of the investigator might enable a greater degree of integrity and authenticity. In this step, **DT2** is acquired.

Once these forensic traces are acquired, the data is interpreted. **DT3** is extracted from **DT2**. If no encryption is used (or the encryption keys are known to the investigator), **DT9** could also be reconstructed. This data is then used for the Feature Extraction.

Finally, the model generated during the **SP** is used to classify the forensic traces yielding information about the activities performed during the VC session investigated here. During the entire process the performed actions and steps are documented in order to create a final documentation usable to judge the potential evidentiary value of the classification results.

IV. Exemplary Implementation and preliminary results

In order to test our hypothesis that the features presented in III-B are sufficient to identify the various possible activities in VC (see III-A) even within encrypted communication a test setup using two different VC solutions was created. First preliminary tests have been conducted in order to show the validity of this approach.

The two video conference solutions Zoom [14] and Big Blue Button (BBB) [15] are chosen for a first practical case study. Zoom is selected for its high market share among commercial VC solutions (as can be seen in [16]). BBB is an Open Source VC solution. The prevalence of self-hosted instances of BBB hinders extensive use statistics for this solution. However, it is broadly used in various educational institutes. Both follow the communication structure as laid out in Section III-C and are hence representative for the technologies used in VC.

In case of Zoom we use the educational accounts of our university and in case of BBB a self-hosted instance (denoted as **BBB**). Zoom offers a desktop-client and a web-client which are both used (denoted as **Zoom-Web** and **Zoom-App** respectively). BBB offers only a web-client.

 ISBN: 978-1-61208-821-1

A. Setup

Three clients (CL_1, CL_2, CL_3) are used during these tests. While CL_1 is used in every test-run and actively using functionalities like the microphone or webcam to generate corresponding data, CL_2 is only used in those occasions in which multiple active clients are required for the tests. CL_3 is a passive observer (participating in the conferences but not actively using any functionality). The Observation Point for forensic data is located outside the decryption performed by the client located on this system. Hence, the observation takes place at $\mathbf{OP}_6$. All incoming traffic is captured using Wireshark. Different hardware is used for the clients (and therefore the possible quality of audio and video data differs): An *iPad (2017)* is used for CL_1, an *iPhone 11* for CL_2 and a *Lenovo Thinkpad Carbon X1 3th Gen* for CL_3. The following tests are performed in this setup:

[T1] - Audio: CL_1 is using the microphone to send audio data. Different levels of audio usage are compared: 1. The microphone is muted in the conference client and on the hardware (A_1). 2. The microphone is activated in the conference client but deactivated on the hardware (A_2). 3. The microphone is fully activated and a monotone voice is recorded (A_3). 4. The microphone is fully activated and a voice which varies in vocal pitch and volume is recorded (A_4). **[T2] - Video**: CL_1 is using the built-in webcam to send video data. Different levels of video usage are compared: 1. The webcam is deactivated in the client (V_1). 2. The webcam is activated and a black image is recorded (V_2). 3. The webcam is activated and a static video (without visible movement) is recorded (V_3). 4. The webcam is activated and and a moving video (movement of a person) is recorded (V_4). The aim is to test whether an observer can identify user behavior, e.g. whether a person is visible in the camera or not. **[T3] - Video2x**: CL_1 and CL_2 are using the built-in camera and both perform tests like in [T2] (V_5). The aim is to test whether an observer can identify the number of active participants or not. **[T4] - Video-Audio**: CL_1 is using different audio- and video features like described in [T1] and [T2]. The aim is to test whether the stream of audio and video data can be distinguished on network level in order to evaluate them separately. This adds up to twelve test cases from $\mathbf{OP}_6$ (four tests with three different clients). Additional tests cases are performed to test the validity of our approach in regards to keystroke dynamics. **[T5] - Keystroke** CL_1 is using the chat functionality of the client and in case of BBB also the *Shared Notes*. Simple text strings are typed and sent. In this test the outgoing traffic is captured at CL_1 (observation point 2) as well as the incoming traffic at CL_3 (observation point 6). In addition traffic without any user interaction is captured at both points. Each of the tests presented here is repeated twice in order to reduce the risk of data corruption.

B. Training

For each of these tests, an initial run during the preparation (as discussed in III-D) is performed in order to obtain training data. Pre-processing is done using the filter options of Wireshark in order to clean up the captures from easily identifiable noise and in order to separate the specific streams. The necessary steps are different according to the specific conferencing software used. These pre-processing steps are provided here in order to present an overview on potentially necessary actions:

BBB: The audio and video streams are transferred using WebRTC[17] which is based on UDP. Other data, like information about the session itself and the text chat is transferred using Websockets based on TCP. Hence, TCP packets are removed. **Zoom-Web**: The video streams are transferred exclusively via UDP. The audio streams are transferred using Websockets (TCP) and WebRTC (UDP). Depending on the test scenario, UDP or TCP packets are filtered out. **Zoom-App**: The audio and video streams are transferred over UDP exclusively. Additional data is transferred partly via TCP. Therefore all TCP packets are filtered out. In all cases, only the incoming traffic is observed, filtered by the source IP address. While the IP address of the BBB server is unique over all conference sessions, the addresses of the Zoom server change every time. The filters need to be adapted accordingly.

Since the data is collected at $\mathbf{OP}_6$, only a subset of potential data is available. As discussed in III-C, the encrypted and combined media streams (**DT2**) as well as some metadata necessary for the communication between the Conference Server and Client CL_3 is available which prevents the separation of (**DT2**) into streams specific to either CL_1 or CL_2. This data is then used as training data in order to create models. Here, a self-created feature extractor which extracts the features mentioned in III-B is used. This feature extractor also extracts additional features which were initially deemed not useful for the decision task at hand. The WEKA machine learning workbench[18] is used to create the model. In addition, the captures are visualized using the I/O-Graph tool inherent in Wireshark in order to perform a first visual inspection of the features predicted to be relevant. Wireshark allows for the visualization of (*F1*) number of packets per time (*pck/sec*) and (*F2*) the throughput (*byte/sec*).

C. Tests and Results

The test cases cover **[T1]** - **[T4]** either using **BBB**, **Zoom-App** or **Zoom-Web**. The J48 classification algorithm in WEKA was employed in order to identify the various user activities. This algorithm uses decisions trees which are interpretable in order to identify the relevant feature (as has been demonstrated in [3]). However, this was preceded by a purely visual inspection of the respective data.

a) Visual Inspection: A first visual inspection was performed using the Wireshark I/O Graph for every test case as shown in Figure 3. The identifiable activities for each test case (see Section IV-A) are shown in the following way: "/" means a distinction is possible between the activities on the left and the right side. "∧" indicates that the activities on the left and right side can not be distinguished from each other,

but together form a combined class. An X denotes that no separation was possible.

This approach allows for the extraction of user behavior in ten out of twelve test cases. An overview of the results is provided in Table I.

During the inspection, only $F2$ shows significant differences between the different user behavior features and thus only this feature is used for the visualization. **[T4]** in **Zoom-Web** does not show any difference between the activation of one or two cameras. This is due to the fact that **Zoom-Web** can only show one incoming stream per time and the Zoom server decides which is chosen. **[T3]** in **BBB** cannot be used to differentiate between audio and video streams, since both are sent over the same protocol and port. In **[T3]** in **Zoom-Web** the audio and video streams can easily be distinguished based on the protocol (video: UDP only, audio: UDP and TCP). In the **Zoom-App**, the distinction is possible based on different port numbers per stream. In **[T4]** in both **Zoom-Web** and **Zoom-App**, $F1$ differs heavily, based on the quality of the used camera which potentially allows to extract additional information about the used hardware. The same is not possible in case of **BBB**.

Fig. 3. In the case of Zoom, the amount of incoming UDP data at $\mathbf{OP}_6$ varies, depending on the audio-channel usage of CL_1. An activated microphone (A_2) can be distinguished from a deactivated (A_1) and a continuous voice (A_3) from a choppy voice (A_4).

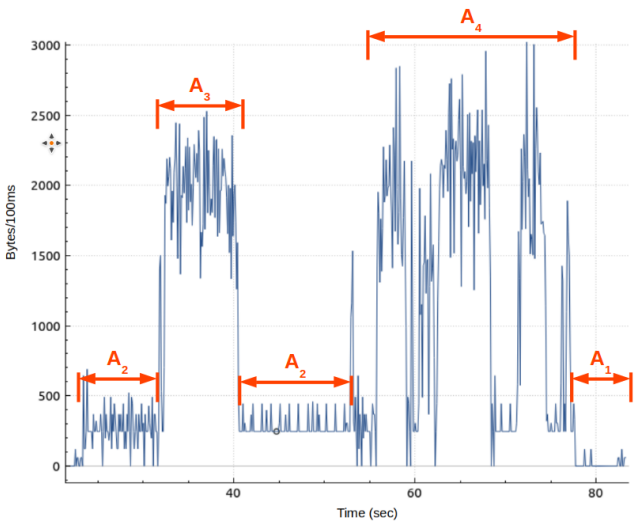

TABLE I. RESULTS OF THE VISUAL INSPECTION EMPLOYING THE WIRESHARK I/O GRAPH FOR ACTIVITY IDENTIFICATION

Test	Zoom-Web	Zoom-App	BBB
[T1]	A_1 / A_2 / A_3 / A_4	A_1 / A_2 / A_3 / A_4	$(A_1 \wedge A_2)$ / $(A_3 \wedge A_4)$
[T2]	V_1 / V_2 / V_3	V_1 / V_2 / V_3	V_1 / $(V_2 \wedge V_3)$
[T3]	V_1 / V_2 / V_3 / A_1 / A_2 / A_3 / A_4	V_1 / V_2 / V_3 / A_1 / A_2 / A_3 / A_4	X
[T4]	X	V_1 / V_2 / V_3 / V_4	$(V_1 \wedge V_2 \wedge V_3)$ / V_4

[T5] shows that no keystroke biometrics is possible in either **Zoom-App** or **Zoom-Web** at all. Single keystrokes are not transferred over the network and other clients have no indication about, whether a client is currently typing or not. In **BBB** keystroke biometrics is possible on different levels: At OP_2 every keystroke in the text chat is visible on the network as an TCP request to the server, which is directly answered. The size of the transferred packets does not differ, since the concrete keystroke (character) is not transferred but only the information about the occurence of a keystroke event. At OP_6 packets are received whenever the status of another user changes between *writing* and *not writing*. The first event is triggered after the first keystroke is received by the server, the second approximately 1.5 sec after the last keystroke was received. Even stronger keystroke biometrics are possible with the shared notes: Here every keystroke is also visible at OP_6 and the network packets differs in size, based on the transferred content (e.g. the packet is larger if one chunk of text was inserted by just one keystroke like using the insert shortcut).

b) Classification Results: The promising first results from the visual inspection are confirmed using pattern recognition in the form of the J48 algorithm. In this case, the derivation of user behavior is possible in nine out of the twelve test cases as can be seen in Table II. Here, Kappa Statistics are used. They range between 0 and 1 with 1 indicating optimal classification.

The identification is possible to a greater degree than those based on visual inspection. Indeed, the missing three entries are the result of non-sufficient training data and are not indicative of the impossibility to discern the user behavior in these cases.

The first results lead to the surprising observation that the most distinct features are those based on the transfer dynamic during the communication (especially a feature referred to as *syn_transfer_drop_freq* which denotes the rate with which windows of a certain amount of packets each have a lower throughput than the previous window). The extraction of this feature is present in the feature extractor used here. However, when filtering out the STUN protocol, these features lose their usefulness in the case of BBB. STUN - or Session Traversal Utilities for NAT - is commonly used protocol which enables clients to access servers which are based behind a NAT-firewall.

Therefore, we suspect that these features are highly susceptible to changes in the network infrastructure. However, this question remains open for additional work and for this paper we removed these features from further considerations.

TABLE II. KAPPA STATISTICS FOR DIFFERENT TEST CASES IN THE CONTEXT OF ACTIVITY IDENTIFICATION

Test	Zoom-Web	Zoom-App	BBB
[T1]	0.9989	0.9947	1
[T2]	0.9993	0.9953	1
[T3]	NA	NA	1
[T4]	0.9993	0.9947	NA

A detailed example of this can be seen in Table III where the confusion matrix for **[T2]** using **Zoom-Web** is provided.

 ISBN: 978-1-61208-821-1

The confusion matrix shows a clear distinctness between the various classes of activities.

TABLE III. CONFUSION MATRIX FOR **[T2]** USING **ZOOM-WEB** IN THE CONTEXT OF ACTIVITY IDENTIFICATION

Classified as	V_1 **deactivated**	V_2 **black screen**	V_3 **one person in front**
V_1 **deactivated**	44	0	0
V_2 **black screen**	1	1221	0
V_3 **one person in front**	1	0	2898

The J48 decision tree allows for an interpretation of the features this distinction is based on:

```
winn_totlen_stddev1 <= 124
|   str_totlen_stddev1 <= 6: NoCamera
|   str_totlen_stddev1 > 6: Black
winn_totlen_stddev1 > 124: MovingImage
```

The feature *winn_totlen_stddev* denotes the standard variation of the total length of the payload of various packets during windows of 500 packets. Even when these features are removed, other features would be used without degrading the classification quality.

Beyond the possibilities of visual inspection is for example the identification of all specific activities in the case of **[T2]** using **BBB**:

```
str_entropy_mean1 <= 6.38381: NoCamera
str_entropy_mean1 > 6.38381
|   winn_kbps1 <= 0.069221: Black
|   winn_kbps1 > 0.069221
|   |   winn_ia_pl_mean1 <= 0.01912: MovingImage
|   |   winn_ia_pl_mean1 > 0.01912
|   |   |   str_totlen_mean1 <= 796: MovingImage
|   |   |   str_totlen_mean1 > 796: Black
```

Here, *str_entropy_mean1* (the entropy) is used to distinguish between NoCamera and a Black image. This seems to be the case since there is less variance in the video stream. Hence, additional features might be useful.

V. DISCUSSION

The results presented in this paper can be seen as a preliminary trend indicating the potential benefits of a forensic behavior analysis in encrypted VC sessions (the resulting capture files can be found online [19]). While it is freely and openly admitted by us authors that the amount of training and test data used within this first evaluation is not sufficient for any generalization, the used features (mainly those covering the transfer rate of data) seem promising. The use of $\mathbf{OP}_6$ for the **DG** impacts the attribution of a given behavior to a specific client since a separation of the specific streams is often impossible after they have been aggregated by the conference server. On one hand this might make the approach presented in this paper impractical for a larger amount of communication participants, on the other hand is this segmentation problem not that hard: For example in the case of audio streams in VC, only one participant is likely to speak at any given point of time. For larger analyses, the separation of different types of media streams on the network layer, like it is possible in both Zoom clients, is nevertheless very important.

VI. CONCLUSION

In this paper we have demonstrated an approach for forensic behavior analysis on encrypted VC communications. First results indicate that an activity identification is possible. This approach relies on the creation of knowledge (in the form of decision models) before a specific investigation takes place. Then, these models are applied to the captured data. The first tests conducted in this paper have to be repeated on a greater amount of data for training and test in order to increase the generalization of the potential this approach offers in terms of forensic activity identification.

In addition, the first tests identified potentially useful features beyond the use of those based on network throughput. The multitude of different communication scenarios and the specific data available at the various points of the network has been discussed in Section III-C providing a clear definition of the specific Observation Points. It has to be investigated how accurate a model created with training data from one specific Observation Point is when applied to test data obtained from another observation point. Also, the network connection characteristics might have an influence on the specific models since such a connection could limit bandwidth. This might influence the investigation into the use of specific features (like for example *syn_transfer_drop_freq*).

While the presented approach might prove useful in reconstructing potential security events in a given network, it could also lead to certain privacy related risks. Although the use of biometrics to identify persons within these streams is not explored in this paper, this seems like a potential field for additional research.

ACKNOWLEDGMENT

The research shown in this paper is partly funded by the European Union Project "CyberSec LSA_OVGU-AMSL".

REFERENCES

[1] "Merriam-Webster.com Dictionary - Videoconferencing," Merriam-Webster, Tech. Rep., 2020.

[2] D. X. Song, D. Wagner, and X. Tian, "Timing analysis of keystrokes and timing attacks on ssh," in *Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10*, ser. SSYM'01. USA: USENIX Association, 2001.

[3] R. Altschaffel, R. Clausing, C. Kraetzer, T. Hoppe, S. Kiltz, and J. Dittmann, "Statistical pattern recognition based content analysis on encrypted network: Traffic for the teamviewer application," 03 2013, pp. 113–121.

[4] A. M. White, A. R. Matthews, K. Z. Snow, and F. Monrose, "Phonotactic Reconstruction of Encrypted VoIP Conversations : Hookt on fon-iks," in *2011 IEEE Symposium on Security and Privacy*, 2011.

[5] B. Dupasquier, S. Burschka, K. Mclaughlin, and S. Sezer, "Analysis of information leakage from encrypted skype conversations," *Int. J. Inf. Sec.*, vol. 9, pp. 313–325, 10 2010.

[6] M. Korczynski and A. Duda, "Classifying service flows in the encrypted skype traffic," 06 2012, pp. 1064–1068.

[7] Y. Zhu and H. Fu, "Traffic analysis attacks on skype voip calls," *Computer Communications*, vol. 34, pp. 1202–1212, 07 2011.

 ISBN: 978-1-61208-821-1

[8] S. Le Blond, C. Zhang, A. Legout, K. Ross, and W. Dabbous, "I know where you are and what you are sharing: Exploiting p2p communications to invade users' privacy," in *Proc. 2011 ACM SIGCOMM Conference on Internet Measurement Conference*. New York, NY, USA: ACM, 2011, p. 45–60.

[9] N. Whiskerd, N. Körtge, K. Jürgens, K. Lamsöft, S. Ezennaya-Gomez, C. Vielhauer, J. Dittmann, and M. Hildebrandt, "Keystroke biometrics in the encrypted domain - a first study on search suggestion functions of web search engines," in *EURASIP J. on information security*, 2020.

[10] M. Bishop, *Computer Security - Art and Science*, 2nd ed. Addison-Wesley, 2018.

[11] S. Kiltz, J. Dittmann, and C. Vielhauer, "Supporting Forensic Design - A Course Profile to Teach Forensics," in *Proc. 9th Int. Conf. on IT Security Incident Management & IT Forensics (IMF 2015)*. IEEE, 2015.

[12] R. Altschaffel, K. Lamshöft, S. Kiltz, M. Hildebrandt, and J. Dittmann, "A Survey on Open Forensics in Embedded Systems of Systems," *Int. Journ. on Advances in Security*, vol. 11, pp. 104–117, 2018.

[13] R. Altschaffel, M. Hildebrandt, S. Kiltz, and J. Dittmann, "Digital Forensics in Industrial Control Systems," in *Proceedings of 38th International Conference of Computer Safety, Reliability, and Security (Safecomp 2019)*. Springer Nature Switzerland, 2019, pp. 128–136.

[14] Zoom Video Communications, Inc., "Zoom - Video Conferencing, Web Conferencing, Webinars," 2020, https://zoom.us [September 20. 2020].

[15] BigBlueButton, "BigBlueButton - Open Source Web Conferencing," 2020, https://bigbluebutton.org/ [September 20. 2020].

[16] datanyze, "Web Conferencing Market Share," 2020, https://www.datanyze.com/market-share/web-conferencing--52 [November 05. 2020].

[17] C. Jennings, H. Boström, and J.-I. Bruaroey, "WebRTC 1.0: Real-Time Communication Between Browsers," 2020, https://www.w3.org/TR/webrtc/ [September 20. 2020].

[18] University of Waikato, "WEKA - The workbench for machine learning," 2020, https://www.cs.waikato.ac.nz/ml/weka/ [September 20. 2020].

[19] Jonas Hielscher, "Forensic in Video Conferencing Results," 2020, https://gitti.cs.uni-magdeburg.de/jhielscher/forensicinvideoconferencingresources [November 06. 2020].

 ISBN: 978-1-61208-821-1