TRACKING ENCRYPTED VOIP CALLS VIA ROBUST HASHING OF NETWORK FLOWS

Baris Coskun

Polytechnic Institute of NYU
Electrical and Computer Engineering
Brooklyn, NY

Nasir Memon

Polytechnic Institute of NYU
Computer Science and Engineering
Brooklyn, NY

ABSTRACT

In this work we propose a Voice over IP (VoIP) call tracking scheme based on robust hashing of VoIP flows. In the proposed scheme the audio content of a possibly encrypted VoIP flow is identified by a short binary string, called the robust hash, using variations on the flow's bitrate over time. These robust hashes are then used to detect pairs of parties communicating with each other. In summary, if two parties are communication with each other then they have a pair of VoIP flows having similar robust hashes with high probability. The basic intuition behind the proposed hash function is twofolds: i)The variable bitrate codec employed in most VoIP applications result in a distinctive bitrate variations over time for each VoIP flow depending on the underlying audio content. ii) Encryption typically doesn't change the bitrate of a VoIP flow. Our experiments show that the proposed scheme is able to identify Skype VoIP flows even under various network impairments such as packet delays, jitter and packet drops.

***Index Terms*—** VoIP Security, Robust Hash, Call Tracking

1. INTRODUCTION

VoIP calls are often anonymized using cryptographic tools and various proprietary protocols for privacy purposes. Furthermore most VoIP networks have peer-to-peer (P2P) architecture (i.e. Skype) and VoIP packets pass through several peers before reaching their destinations. Therefore, it is very hard to determine who is talking to whom on a VoIP network. However, such information can be very crucial for many scenarios, especially in law enforcement. In such cases, one could determine if two parties were talking to each other if he could identify audio contents of VoIP calls by an identifier. For instance, suppose that the traffic of a number of suspected networks are monitored. More specifically, some sort of content identifiers of VoIP flows are stored for every host in those networks. Then, one can simply search the stored data for pairs of concurrent VoIP flows having similar content identifiers, which suggest that the corresponding parties are communicating with each other with high probability.

Motivated by this, in this paper we propose a novel robust hashing scheme for VoIP flows to identify their audio content by short binary strings. Notice that, to identify the contents of a VoIP flow, one cannot use traditional audio hashing schemes [1, 2], since in most cases VoIP flows are encrypted. The proposed robust hash is based on VoIP flows' packet timings and payload sizes, which are typically invariant under encryption and reflect the content information of VoIP flows. The basic idea is that, variable bitrate (VBR) audio codecs employed in most VoIP applications result in the packet timings and the payload sizes to be highly dependent on the contents of the input VoIP flow. This is because vowels typically require more bandwidth than consonants or fricative sounds [3]. Therefore, depending on the ordering of these sounds in an input audio signal, a VBR codec determines what size packets are sent on what times. Furthermore, this information is often invariant under encryption, since most VoIP applications employ length preserving ciphers.

In summary, the proposed robust hashing scheme takes the timestamps and payload sizes of the packets of a VoIP flow as the input and computes a short constant-length binary string, called **the hash value**, as the flow's content identifier. However, in order this scheme to be employed in VoIP call tracking applications, it has to possess the following crucial properties:

• **Robustness:** Network flows often encounter various network impairments such as packet delays, jitter and packet drops. Therefore, the packet timings and variations on the bitrate slightly differs as a VoIP flow transmitted through networks. The proposed scheme should output the same or similar hash value even if a VoIP flow undergoes such impairments, in order to still be able to identify a VoIP flow at the other end of the network.

• **Low Collision-Probability:** Two different and uncorrelated VoIP flows should have completely different robust hash values.

• **Efficient Computation:** In order for the proposed scheme to be deployed in very large networks, where hundreds of thousands of VoIP have to be monitored, it has to compute hash values very efficiently in real-time as the packets are captured. Furthermore, resulting hash values has to be

©2010 IEEE. Published in the IEEE 2010 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2010)

short enough to avoid any scalability issues during storage or search.

In this work, we formally explain the proposed scheme and demonstrate that it possesses the above properties. The remaining of this paper is organized as follows: After presenting related work in Section 2, we formally discuss the details of the proposed robust hashing scheme in Section 3. Also in Section 3, we give an efficient algorithm which computes the proposed hash values in real-time. Then we present our experiments and results in Section 4. Finally, we conclude the paper and discuss future work in Section 5.

2. RELATED WORK

Wright *et. al.* exploited the information leakage through variable bitrate codecs in [4] to identify the spoken language in encrypted VoIP calls. In a more ambitious work [3], Wright *et. al.* were able to identify spoken phrases from a standard speech corpus in a VoIP call with on average $\%50$ accuracy. In that work, authors trained a Hidden Markov Model using packet sizes of VoIP segments containing phrases in the corpus. These two works strongly suggests that audio contents of a VoIP call can be identified from the resulting packet sizes. Similar to robust hashing, in [5], Coskun and Memon propose a flow sketch to identify packet-timing characteristics of network flows for real time stepping-stone detection. In [6], Wang *et. al.* uses watermarking techniques to identify two communicating parties over VoIP. Their technique actively modifies packet timings of a VoIP flow and try the detect the same modification pattern on the other end of the network. However, real-time modification of packet-timings of all VoIP flows poses potential scalability issues for large networks.

3. ROBUST HASHING OF VOIP FLOWS

In this section, we explain the details of the proposed robust hash function and a real-time algorithm, which efficiently computes robust hash of a VoIP flow. We begin with presenting a brief background.

Background:
In a typical VoIP call, call setup and voice transmission are handled by different network flows. A standard protocol, such as Session Initiation Protocol (SIP), Extensible Messaging and Presence Protocol (XMPP), or a proprietary protocol (i.e. Skype) is often used for call setup. The actual voice data is typically transmitted using Real-time Transport Protocol (RTP) over UDP packets. In this work, we refer these UDP flows carrying actual voice data as **VoIP Flows**. More formally, *a VoIP flow is defined as a collection of UDP packets which possess the same quadruple of ⟨source IP, source port, destination IP , destination port⟩*. Transmitted voice data is often encoded by a Variable Bit Rate (VBR) codec to minimize average bit-rate thereby increasing overall service quality. To ensure low latency constraints, inter-arrival time between UDP packets is typically set between 10 and 50 milliseconds. The size of each UDP packet depends on the output bit-rate of the VBR codec.

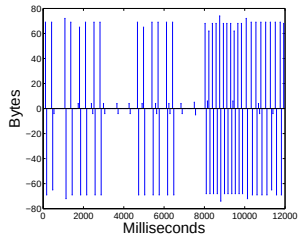


Fig. 1. An example bitrate variation signal constructed from 200 packets of a Skype call.

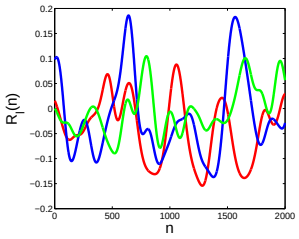


Fig. 2. Three example random bases each shown in different color.

Robust VoIP Hash Overview:
In the proposed robust hashing scheme, to compute the robust hash value of a VoIP flow, we follow three major procedures:

• **Representation:** Packet payload sizes of a VoIP flow varies according to the underlying audio content information. Therefore, in order to identify the audio content of a VoIP call, we first represent the variations on the bit rate of a VoIP flow across time by a sparse discrete signal called **bitrate variation signal**. The bitrate variation signal is basically shows the difference between payload sizes of consecutive packets at corresponding times. Hence, it captures both packet timings and bitrate variations of a VoIP flow in one dimension.

• **Projection:** To compute the hash value, we then project the bitrate variation signal onto pseudorandom smooth bases. Projecting the input signal onto smooth bases functions is widely used in multimedia hashing [7] [8], as it provides robustness to slight changes in the input.

• **Quantization:** Finally the signs the projection values are output as the resulting hash value. The basic intuition is that the slight modifications on the VoIP flow are usually not powerful enough to flip the signs of the projection values. On the other hand, the signs of projection values of two uncorrelated VoIP flows are expected to be uncorrelated.

In the following subsections, we present each procedure in formal details.

3.1. Bitrate Variation Signal

To formally present bitrate variation signal, consider a VoIP flow with P packets, where the packets are in chronological order[1]. For these P packets, let T_i denote the global timestamp of i^{th} packet as the number of milliseconds passed since the global epoch, where $i = 0, 1, 2, ..., P-1$. Using this, we represent the relative timestamp of i^{th} packet by $\widehat{T_i}$ indicating the elapsed time since the beginning of that particular flow, such that:

[1] Since the packets are captured in real time, they are essentially in chronological order by default.

$$\widehat{T}_i = T_i - T_0 \quad (1)$$

On the other hand, let B_i denote the size of the payload of the i^{th} packet in **bytes**. Then, we represent the difference of the sizes of consecutive packets with B_i^{Δ}, such that:

$$B_i^{\Delta} = B_i - B_{i-1} \quad (2)$$

Following that, to represent a VoIP flow's bitrate variations over time we define the **bitrate variation signal** (V) as follows:

$$V(n) = \sum_{i=1}^{P-1} B_i^{\Delta} \left[\delta(n - \widehat{T}_i)\right] \quad (3)$$

where $\delta(n)$ is the Dirac delta function. Note that, $V(n)$ is a discrete signal which has N-1 nonzero values at relative timestamps of all packets except the first one. Note that bitrate variation signal contains both packet-timing and packet payload size information, which represents the underlying content of VoIP flows. As an example, the bitrate variation signal extracted from 200 consecutive packets of a VoIP flow is illustrated in Figure 1.

3.2. Smooth Bases and Hash Computation

Once the bitrate variation signal is constructed, the robust hash value is calculated by projecting V onto L smooth pseudorandom bases, which are denoted by $R_l(n)$, such that:

$$H_l = \sum_n V(n) R_l(n) \quad (4)$$

where $l = 1, 2, 3, ..., L$. Each pseudorandom base is generated independently by smoothing an array of i.i.d Gaussian random variables. To smooth pseudorandom arrays, we used a Gaussian lowpass filter with $\sigma = 50$. As an example, few of the bases we used are plotted Figure 2.

Finally, we use the signs of these projection values to compute the robust hash value of the input VoIP flow. More formally, each bit of L-bit robust hash (h) is computed by 1-Bit quantization of a corresponding projection value such that:

$$h_l = \begin{cases} 1, & \text{if } H_l \geq 0 \\ 0, & \text{if } H_l < 0 \end{cases} \quad (5)$$

where $l = 1, 2, 3, ..., L$.

3.3. Online Hash Calculation in Real-Time

In order the proposed scheme to scale up to large networks, where hundreds of thousands of VoIP are monitored, a fast algorithm is required to compute robust hash values in real-time. In this section we present an efficient algorithm, which computes the robust hash of a VoIP flow cumulatively as the packets are captured requiring minimal memory.

The proposed algorithm assumes that the smooth pseudorandom bases are previously computed and stored in the memory. Note that, in practice these pseudorandom basis arrays should be longer than bitrate variation signals of all possible VoIP flows. Therefore, length of the input signals should be bounded by a certain value. For instance, in our experiments we mostly use 200 consecutive packets of VoIP flows, which approximately corresponds to 12 seconds. Hence, in the experiments we safely employed 15-second long pseudorandom bases.

Once the pseudorandom bases are computed, the proposed algorithm updates the projection values each time a packet is captured. The intuition is based on the fact that the bitrate variation signal '$V(n)$' has nonzero entries only when $n = \widehat{T}_i$. Therefore, we can rewrite Equation (4) as:

$$H_l = \sum_{i=1}^{P-1} V(\widehat{T}_i) R_l(\widehat{T}_i) \quad (6)$$

combining this with Equation (3), we get:

$$H_l = \sum_{i=1}^{P-1} B_i^{\Delta} R_l(\widehat{T}_i) \quad (7)$$

Using the above equation one can compute the projection values cumulatively as the packets arrived. A simple algorithm, named $Compute\_Hash$, to compute the robust hash in real-time using this equation is given below.

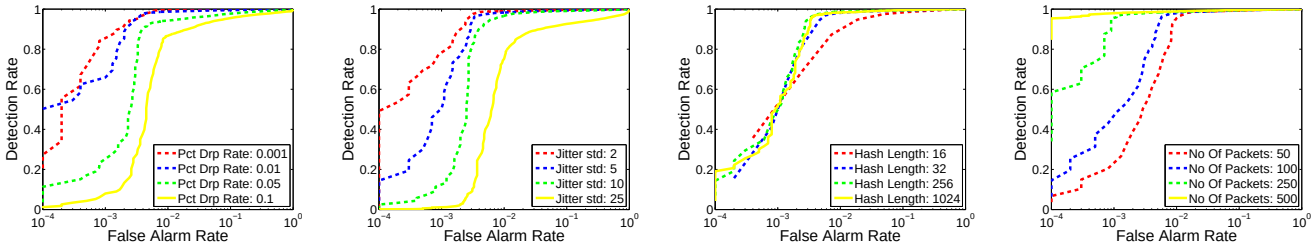
Algorithm 1 $h = Compute\_Hash$

$H \leftarrow [0, 0, 0, ..., 0]$ {*//Initialize projection values* $H_1, H_2, ..., H_L$}
for all captured packet i such that $i = 0, 1, ..., P-1$ **do**
  if i=0 **then**
    $flowStart \leftarrow T_i$ {*//first packet timestamp is when the flow starts*}
  else
    $\widehat{T}_i \leftarrow T_i - flowStart$ {*//relative timestamp*}
    $B_i^{\Delta} = B_i - B_{i-1}$ {*//difference in the bitrate*}
    $H \leftarrow H + B_i^{\Delta} \left[R_1(\widehat{T}_i), R_2(\widehat{T}_i), ..., R_L(\widehat{T}_i)\right]$ {*//update projections*}
  end if
end for
$h = sign\,(H)$
Output h

The $Compute\_Hash$ algorithm is very efficient and runs in $O(N)$ time where N is the number of packets. On the other hand, it requires a constant memory space as it has to keep only the pseudorandom bases and the $flowStart$.

4. EXPERIMENTS

To evaluate the efficacy of the proposed scheme, we set up an experiment where we evaluated the detection performance under various network impairments. For this purpose, we first played 2 hours of recorded speech over a single Skype call and captured its UDP packets. We employed Skype because it is one of the most widely used VoIP application. Then, to obtain a VoIP flow segment, we picked 200 consecutive packets starting from a randomly selected packet. Using this strategy we obtained $10,000$ VoIP flow segments. After that, we applied packet delays, jitter and packet drops to the original segments to simulate the network impairments and consequently



(a) For different packet drop rates (b) For different jitter sta. dev. (c) For different hash lengths (d) For different input signal lengths

Fig. 3. ROC curves for $Dl = 2000$ ms, $\sigma_{Jt} = 5$, $Pd = 0.01$, hash length $L = 256$, input length $= 200$ packets.

obtained $10,000$ modified segments. More specifically, we delayed every packet by Dl milliseconds. Then we added jitter Jt on each packet's timestamp where Jt was drawn from a zero-mean Laplacian distribution with standard deviation σ_{Jt}. Finally, we randomly dropped packets with probability Pd and obtained the modified VoIP segments. After that, we computed the Hamming distance between the robust hashes of the original segments and the modified segments. If the Hamming distance is below a threshold, then we consider it as a successful detection. On the other hand, we also compared the robust hash of each VoIP segment with another randomly selected segment. In this case, if the Hamming distance is above a threshold, then we consider it as a false alarm.

We present the resulting ROC curves in Figure 3. We initially fix the parameters to $Dl = 2000$ ms, $\sigma_{Jt} = 5$, $Pd = 0.01$, hash length $L = 256$, input length $= 200$ packets. To demonstrate the algorithms sensitivity to each parameter, we varied only one parameter on each of the Figures 3(a)-3(d). We didn't vary delay since the resulting robust hashes are invariant under delay because the hash function uses relative packet timestamps with respect to the start of each flow. We observe in these figures that, the proposed scheme performs satisfactorily under reasonable jitter and packet drop rates. Also, it is observed in Figure 3(c) that increasing hash length improves the performance up to some point and then saturates. Hence $L = 256$ seems to be appropriate since the longer the hash values are the harder to manage them (storage, computation etc.) Finally, Figure 3(c) suggests that longer input signals are easier to detect. This is expected since longer signals have more content information which distinguishes it from others.

5. CONCLUSION AND FUTURE WORK

In this work, we presented a robust hashing scheme for VoIP flows, which could be used for VoIP call tracking applications. The proposed hashing scheme exploits a VoIP flow's packet timings and bitrate variations to identify the audio contents of that flow with a short binary string. Our experiments show that the proposed hash function can successfully identify VoIP calls under various network impairments, such as delay, jitter and packet drops. We also showed that the proposed hash values can be efficiently computed in real-time, thereby allowing us to potentially employ it in large scale VoIP tracking applications.

The proposed robust hashing scheme can potentially be used in other VoIP security application, such as VoIP spam[2] (SPIT) detection. For instance, one can monitor a VoIP network to detect if any of the currently active VoIP calls has a robust hash similar to the hash of one of the previously known SPIT messages. This case would indicate that a particular SPIT message is being played at that moment with high probability. In another scenario, one can keep track the histogram of the robust hashes of VoIP calls that was made within a certain time window. Ideally, that histogram should be close to a uniform distribution. However, any spike in the histogram would potentially indicates a brand new SPIT message. We leave the exploration of these potential applications as future work.

6. REFERENCES

[1] Jaap Haitsma Ton and Ton Kalker, "Robust audio hashing for content identification," in *In Content-Based Multimedia Indexing (CBMI*, 2001.

[2] Hamza Özer, Bülent Sankur, Nasir Memon, and Emin Anarim, "Perceptual audio hashing functions," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 1780–1793, 2005.

[3] Charles V. Wright, Lucas Ballard, Scott E. Coull, Fabian Monrose, and Gerald M. Masson, "Spot me if you can: Uncovering spoken phrases in encrypted voip conversations," *IEEE Symposium on Security and Privacy*, 2008.

[4] Charles V. Wright, Lucas Ballard, Fabian Monrose, and Gerald M. Masson, "Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob?," in *SS'07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, 2007.

[5] B. Coskun and N. Memon, "Online sketching of network flows for real-time stepping-stone detection," in *ACSAC'09: 25th Annual Computer Security Applications Conference, Honolulu, HI*, Dec 2009.

[6] Xinyuan Wang, Shiping Chen, and Sushil Jajodia, "Tracking anonymous peer-to-peer voip calls on the internet," in *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*, 2005, pp. 81–91.

[7] B. Coskun, B. Sankur, and N. Memon, "Spatio-temporal transform-based video hashing," *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1190–1208, 2006.

[8] J. Fridrich, "Robust bit extraction from images," in *ICMCS '99, Florence, Italy*, June 1999.

[2] VoIP spam, often called SPam over Internet Telephony (SPIT), are prerecorded messages played unsolicitedly over VoIP calls.