UVP: Uncovering WeChat VoIP Peers over Time

Jiazheng Wang
Department of Automation, Tsinghua University
Beijing, China
e-mail: wangjz17@mails.tsinghua.edu.cn

Zhenlong Yuan
Research Institute of Information Technology
Tsinghua University
Beijing, China
e-mail: yuanzl@mail.tsinghua.edu.cn

Jun Li
Research Institute of Information Technology, Tsinghua
Tsinghua National Lab for Information Science and Technology
Beijing, China
e-mail: junl@tsinghua.edu.cn

***Abstract*—VoIP is pervasively used in online voice conversations through internet, and WeChat has become one of the most successful VoIP applications. Any leakage of VoIP call information will threat user confidentiality and privacy, and thus may lead to a catastrophic consequence to VoIP service providers. In this paper, a thorough behavior analysis of WeChat proprietary VoIP protocols is performed and we find the anonymity of users can be compromised by flow correlation attacks. Hereby a new framework UVP is proposed. UVP works in a global monitoring manner and leverages a new, efficient flow distance metric, evolved from Dynamic Time Wrapping, to correlate traffics from caller and callee. Evaluated in both in-house test bench and with real world dataset, UVP demonstrated its capability of uncovering WeChat VoIP call records and further obtaining the contact network of the victims.**

Keywords-VoIP anonymity; WeChat; dynamic time wrapping

I. INTRODUCTION

Voice over Internet Protocol (VoIP) is a technology for transmitting digitalized voice between users through network. With the rapid development of network technologies and facilities, VoIP has become the most successful substitutions for traditional Public Switched Telephone Networks (PSTN). Applications that provide high-quality VoIP services have been emerging in many countries during the last several decades. And WeChat, beyond doubt, is one of the most popular VoIP services providers. According to a statistics report[1], daily active users of WeChat exceeded 902 million compared with Skype's 300 million monthly active users in 2017 and daily number of WeChat calls reached more than 205 million. WeChat VoIP is profoundly reshaping the communication habits of people, especially in China.

In general, VoIP services will make two or more concurrent connections at the same time, and different protocols cooperate together to handle client connectivity and data transmission issues. Voice communication channels are typically divided into *control channel* and *data channel* according to their perspective functionalities. Control channel protocols, Session Initiation Protocol (SIP) [2] for example, are designed to know where the caller and callee are, understand the network status of them before the conversation actually starts. During the conversation, control channel protocols also monitor and adapt to network changes. Based on the information acquired by the control channel, data channel can be set up to deliver audio chucks, whether relayed by a third party server, in a so called *relay based model*, or directly between the VoIP peers, in the so called *peer to peer model.*

VoIP services are responsible to protect the anonymity of users, namely *caller anonymity*, *callee anonymity* and *unlinkability of caller and callee* [3,4] which means the identity of caller and callee should be protected as well as the links (i.e. call records) of two users given traffics generated by VoIP services. Preserving call records is an essential security task for VoIP services. However, there are some previous works showing that unencrypted control channel, for example SIP based signaling is vulnerable to deep inspection and the caller and callee is directly linkable. Apart from that, others existing works also show that data channel, whether encrypted or not, suffers from *active attacks* and *passive attacks* and fails to achieve such anonymities.

For relay based data channel, some in-path attackers can actively manipulate the traffic such as delay packets to generate a special pattern in packets inter-arrival time, which helps attacker to correlate data channels of caller and callee [4,5]. However, such kind of active attacks may fail due to traffic rate limiting or mechanisms that randomize the inter-arrival time. On the other hand, a passive attack do not disturb the traffic but use other flow metrics, for example Dynamic Time Wrapping (DTW) distance, or deep learning approaches to correlate data channels [6,7,8,9], packet size of which typically varies a lot with the input voice. Therefore, both control and data channel of VoIP are plagued by privacy issues in different communication models [5,6,10,11,12,13,14,15,16]

In this paper, we mainly focus on passive attacks which may compromise the anonymity of caller and callee. To our best knowledge, correlation of control and data channels of proprietary VoIP services is not fully researched. Meanwhile, most existing works end up at matching a pair of caller and callee, without investigating more on the structure of contact network of users. Therefore we present a novel framework, UVP, to uncover WeChat VoIP peers over time in a global monitoring manner by correlating partially encrypted control channels and heavily data channels. Furthermore, UVP also reveals the contact network of the users. The contributions of this work are summarized as follows:

- A thorough behavior analysis of WeChat VoIP protocols is performed, indicating that partially encrypted control channel and data channel can be leveraged to link caller and callee.
- Utilizing the low-latency characteristic of VoIP protocol, a data channel distance metric is proposed based on DTW, called Dynamic Flow Wrapping, which is efficient in time and space.
- The contact relationship of users is modeled as weighted graph, according to which communities and key users can be revealed over time. This graph can also provide heuristics to future flow correlation and thus improve its performance.

In the rest of this paper, we will introduce the VoIP services of WeChat and methodology of UVP. Future discussion and conclusion is also provided.

II. Preliminaries

In order to justify the design of UVP, some preliminary VoIP services of WeChat needs to be briefed, and an introduction of the classic time series distance metric DTW is necessary, as the basis of the to be proposed DFW.

A. *WeChat VoIP Protocol Analysis*

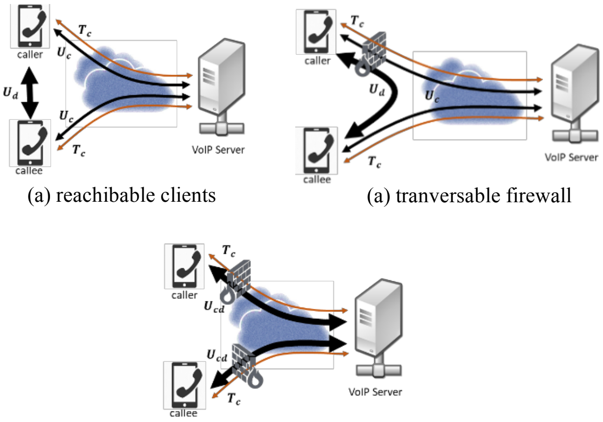
WeChat implements a proprietary version of SIP based services, which also includes a control channel and a data channel.

Each and every WeChat client maintains a long-lived TCP connection, denoted as T_c, with server to transfer application signals and short messages. The server acquires client-side information and acts as name server. As soon as a VoIP call request is made by a client, an encrypted packet is sent through this TCP connection. The VoIP server quickly checks where the callee is and start finding out the network conditions of both clients, such as whether behind a firewall or NAT, and which type of NAT. This works similarly to STUN and TURN [17,18]. These negotiations determine the communication model of the data channels.

According to connectivity of the VoIP peers, i.e. whether the peers can set up a connection directly, VoIP server then decides the communication model to use. Figure 1 illustrates the *relay based model* and *peer to peer model*.

For example, if at least one client has a public IP address or both clients are behind traversable NATs or firewalls, data channelU_d can be setup without help of a third node (Figure 1(a), (b)). Meanwhile, T_c is not specially designed for monitoring VoIP calls so another UDP control channel U_c is set up which also acts as a failover if U_d is broken. U_c is maintained during the entire VoIP call.

The second one is peer to peer model. If both clients are behind symmetric NAT (i.e. a type of NAT that is impossible to traverse even with the help of a third party server), or firewall that forbids peer to peer traffics, a relay server is necessary to set up the data channel U_{cd}. It should be noted that U_{cd} is used for both signaling and data transmission. The two types of communication model result in a difference whether data channel is accessible or not to the attacker in the gateway attack model (more details in section III.A).



(a) reachibable clients (a) tranversable firewall

(c) relay based model

Figure 1. Peer to peer model and Relay based model.

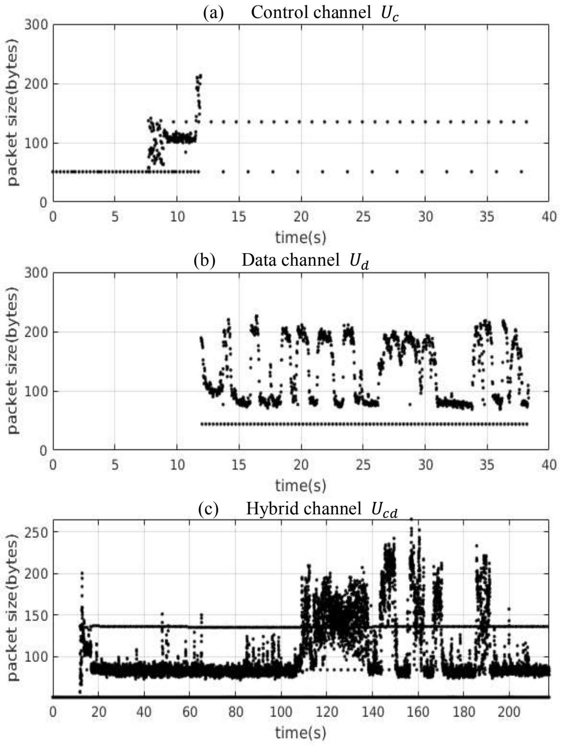
20 bytes	8 bytes	D5 00 00 01 …	… … … … … … … …
IP header	UDP header	WeChat VoIP proprietary header	unknown payload

Figure 2. Packet structure of UDP channels.

Both control and data channel work either in TCP or UDP. WeChat uses UDP as its first choice for efficiency considerations. Meanwhile it supports TCP as a substitution when UDP somehow fails to reach the server, for example due to special firewall rules. This is not very common in mobile networks where most WeChat clients are, hence in this paper, we only focus on UDP based channels.

According to the existing works [19,20], if a VoIP protocol uses UDP for data channel, the payload of a UDP packet tends to be partly encrypted, that is, it seals audio chucks with a plaintext proprietary protocol header. We confirmed that this is exactly the case for the UDP based control and data channel of WeChat. Figure 2 illustrates an example of WeChat VoIP packet in a UDP channel. The first byte of the UDP payload, i.e. the first byte of proprietary header, can be a signature of channel U_c on peer to peer model and channel U_{cd} on relay based model. Thus, all UDP channels are directly identifiable with deep inspection.

Figure 3(a) (b) shows the characteristics of control/data channel packet size in peer to peer model, corresponding to U_c and U_d, respectively. For U_c (Figure 3(a)), small packets are sent in high density before callee accept the call (from 0s to 7s). There is an **initial burst** in packet size once the call actually starts (from 7s to 12s). Then data channel U_d is set up (Figure 3(b)), so U_c converts to "heart-beating mode" (from 12s to end). Packets are slowly sent to VoIP server in order to monitor the liveness of clients. The lifetime of U_c roughly indicates start and end time of a VoIP call. Channel U_d carries audio chucks so the packet size is larger. Voice bursts can be directly observed, whose pattern can be leveraged for data channel correlation with some specific distance metrics (more details in section B).


(a) Control channel U_c

(b) Data channel U_d

(c) Hybrid channel U_{cd}

Figure 3. Distribution of packet size in control/data channels.

Figure 3(c) shows the packet size distribution of U_{cd} in relay based model. U_{cd} can be regarded as the mixture of U_c and U_d, which is also vulnerable to correlation attacks.

Based on observations above, both control and data channel may leak information about "who called whom, when and how long". Revealing a pair of caller and callee by correlating control channels and data channels is feasible.

B. DTW Distance in Flow Correlation

Flow correlation is a typical kind of passive attack to break anonymity of mixing or proxy systems [7,8,21]. If the packet size or inter-arrival time consists of distinguishable patterns, the traffic of sender and receiver will demonstrate these detectable patterns, which helps an attacker to correlate them. In the scenario of VoIP, packet size is a suitable feature because it varies with input voice. Hence, a metric is demanded to compute the distance of two flows, in order to obtain the correlation, and in turn to match the peers.

Dynamic Time Wrapping (DTW) is a classic algorithm to measure distance between two temporal sequences, typically applied in audio signal processing, which plays an important role in data channel correlation. Some existing works [15,22,23] adopted this measurement to VoIP data channel because it conforms well to network delay and UDP packet loss.

Given two time series S and T, respectively of length n and m, namely $S = \{s_1, s_2 \ldots s_n\}$ and $T = \{t_1, t_2 \ldots t_m\}$. Computing DTW distance of these two time series can be reduced to a dynamic programming problem. A matrix γ of size $n \times m$ needs to be constructed, where $\gamma_{i,j}$ is the DTW distance between the subsequences $S[1:i]$ and $T[1:j]$. The recurrence relation complies with equation (1), where $d(s_i, t_j)$ is the distance between two points, a typical Euclidean distance.

$$\gamma_{i,j} = d(s_i, t_j) + \min\{\gamma_{i-1,j}, \gamma_{i,j-1}, \gamma_{i-1,j-1}\} \qquad (1)$$

Then $\gamma_{nm}$is the DTW distance of S and T. The time and space complexity of classic DTW is $O(n \times m)$. Several papers show that the dynamic programming table size can be reduced by using the characteristics of audio series. In this paper, we leverage the low latency of VoIP service to adapt classic DTW to flow correlation of data channels in real-time and propose a new algorithm Dynamic Flow Wrapping (more details in section III.B).

III. Methodology

Targeting the vulnerabilities in WeChat VoIP services, an attacker can deploy his program on a gateway and then acquire the its call records of all users inside the local network by means of flow correlation monitoring manner.

Based on our gateway threat model, the methodology to uncover WeChat's VoIP peers is proposed, with peer matching as its core and contact discovery as outcome.

A. Threat Model

Figure 4 illustrates the gateway threat model. For example, WeChat clients C_1, C_2, C_3, C_4 are WeChat clients and all their traffics need to go through a malicious gateway GW to reach a VoIP server. The attacker uses deep inspection to identify control and data channels. This is doable because UDP channels have signatures as analyzed in section II.A.

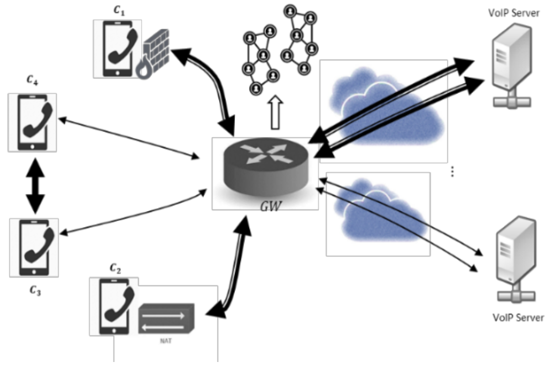


Figure 4. Gateway threat model.

Data channel and control channel of C_1 and C_2 are accessible to the attacker because they work in relay based model. Therefore a flow correlation attack can be performed on data channels. The attacker monitors all VoIP traffics of the subnet. Obviously, the correlation algorithms and distance metrics must be efficient enough, in order to keep up with the network speed.

For C_3 and C_4, only control channel is accessible to the attacker because they work in peer to peer model and no plaintext information can be used to match the caller and callee. However, the control channel implies the start and end time of a VoIP call. So a time range based correlation can be performed on control channels. It should be noted that there is a small chance that different VoIP peers that work in peer to peer model both start a phone call at t_s and end at t_e, i.e. a *period collision* happens for time range based correlation.

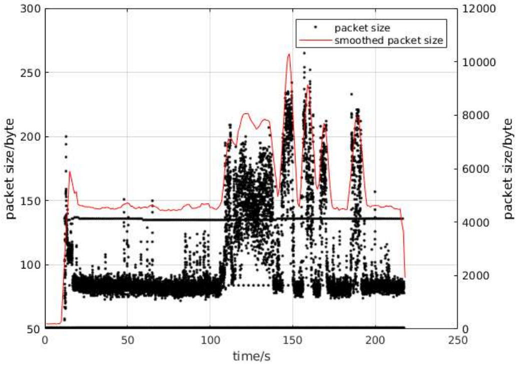


Figure 5. Packet size smooth. Y axis on the left denotes the packet size in bytes. Y axis on the right denotes total packet bytes in one second.

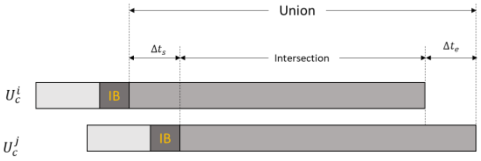


Figure 6. IOU and start/end gap.

In order to reduce the complexity of flow correlation on data channel and possibilities of period collisions on control channel, a user contact network can be constructed, where each node and edge represent a user and the frequency of two users making a WeChat VoIP call, respectively.

After observation over time, records of "who calls whom, when and for how long" of all users behind the GW are obtained. Furthermore, the relationships among those users are leaked this way and their community structure and communication habits can be further analyzed.

B. Peer Matching

Peer Matching (PM) module is designed to correlate control channel and data channel of a given pair of caller and callee, and to determine if they are VoIP peers. For both channels, information directly observed includes source/destination IP, port, start/end time of the UDP session and the uplink/downlink packet size series. No content information is observed, as they are in ciphertext.

Then the packet size series is smoothed to correlate channels.

Packet size smooth: According to Figure 5, the pattern of a channel is strongly indicated by packet size burst. As a preprocessing, packet sizes are added up every Δt and constitute a *flow size series*, so that the burst can be better recognized and size of signaling packets is negligible for multiplexed channel U_{cd}.

Control channel correlation: For control channel U_c^i, packet size cannot provide much heuristics except for indicating the *start time* of a VoIP call. As Figure 5 shows, there is an initial signaling burst, denoted as *IB*, representing callee accept the call and data channel is about to be set up. A threshold δ_{ib} is predefined to detect IB and hereby start time is decided as t_s^i. In additional, end of control channel represents *end time* of the VoIP call, denoted as t_e^i. The overlap rate of two control channels U_c^i, U_c^j is defined as Intersection-over-Union (IOU) in equation (2).

$$IOU\left(U_c^i, U_c^j\right) = \frac{\min\left(t_e^i, t_e^j\right) - \max\left(t_s^i, t_s^j\right)}{\max\left(t_e^i, t_e^j\right) - \min\left(t_s^i, t_s^j\right)} \quad (2)$$

$$l\left(U_c^i, U_c^j\right) = IOU\left(U_c^i, U_c^j\right) \times e^{-k\left(\left|t_s^i - t_s^j\right| + \left|t_e^i - t_e^j\right|\right)} \quad (3)$$

Taking absolute duration of the VoIP call into consideration, the confidence of matching is defined as equation (3). A VoIP call lasts at least more than a few seconds, typically much longer, and the network delay is several orders of magnitude lower than that. Therefore, the start and end time of caller and callee should be close enough, less than one seconds in general. We named this phenomenon as start/end proximity. In such case, the later term in equation (3) should be extremely close to 1 so IOU becomes the final confidence. However, if $\left|t_s^i - t_s^j\right| + \left|t_e^i - t_e^j\right|$ is not small enough, for example 10 seconds, but IOU can still be large due to very long VoIP calls, then this probably a false positive. So the confidence is adjusted by multiplying a number close to 0. Control channels U_c^i, U_c^j are considered matched if the confidence is above a threshold δ_c.

In theory, there must be collisions for control channel correlation. That is two pairs of users making a VoIP call simultaneously and finishing at the same time. In some cases, such collisions can be handled by building a contact network (more details in section 3.3). But in practice, control channel correlation should be effective and efficient.

Data channel correlation: Data channel provides much more information that can be leveraged to correlate caller and callee. As mentioned above, the packet size of data channel varies with the input audio. Given a pair data channels U_d^k and V_d^k from a caller and its callee, the smoothed uplink/downlink packet size series after initial burst are then denoted as $\overrightarrow{U_d^k}, \overleftarrow{U_d^k}, \overrightarrow{V_d^k}, \overleftarrow{V_d^k}$. The channel distance $\gamma\left(\overrightarrow{U_d^k}, \overleftarrow{V_d^k}\right)$ and $\gamma\left(\overleftarrow{U_d^k}, \overrightarrow{V_d^k}\right)$ should be small while $\gamma\left(\overrightarrow{U_d^k}, \overleftarrow{V_d^l}\right)$ and $\gamma\left(\overleftarrow{U_d^k}, \overrightarrow{V_d^l}\right), \forall l \neq k$ should be large.

The attacker works in a global monitoring manner. Channel distance should be computed in real-time. The attacker can leverage the low-latency characteristic of data channel and adapt classic DTW for real-time computing.

Figure 7 shows the DTW paths of data channels from a pair of peers and also another pair of unrelated users. Obviously, a good path always follows the diagonal line of the dynamic programming table while a bad path does not. The space table can also be restricted in a band that is much smaller than the original table. In addition, there is no need to compute DTW distance over the entire sequences. A packet size series can be chopped into some shorter subsequences and compute DTW distance independently, without introducing significant error. Therefore, **Dynamic Flow Wrapping** is proposed.

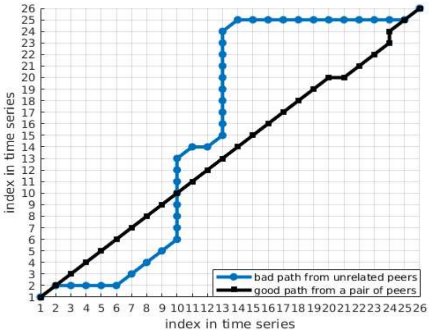


Figure 7. Good path and bad path in DTW

In Algorithm 1 input flow size series is divided into small chucks. Each chuck is in length C and the last chuck may be shorter. Then DFW computes distance of each pair of chucks by calling function Chuck-DFW. In the classic DTW, a state matrix of size n^2 is maintained to solve the dynamic programming problem, where n denotes the length of input series. By introducing restricted band with width w around the diagonal line, the size of state matrix is reduced to $2nw + n - w^2 - w$, where w denotes the width of restricted band. Only length of the previous row is needed to recover the distance of points $(i-1, j), (i, j-1)$ and $(i-1, j-1)$. Thus time and space complexity of the algorithm are optimized.

Figure 8 illustrates the state table of classic DTW and DFW. It is obvious that the time and space complexity is reduced to $O(w\sqrt{m^2+n^2})$ compared to original $O(mn)$.

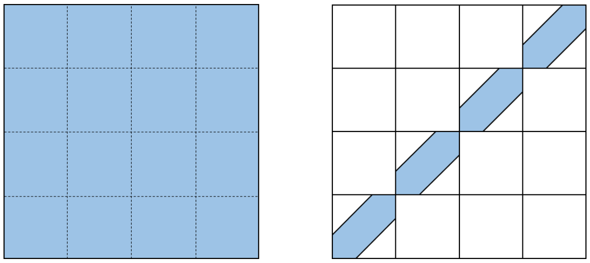


Figure 8. Legitimate space of DTW and DFW.

PM module uses DFW distance to match data channels in real-time. After observing U_d^i for $K\Delta t$, PM calculates distance of U_d^k against all lived data channels V_d^l and get $\{d_{kl}, l \neq k\}$, where $d_{ij} = \frac{\gamma\left(\overrightarrow{U_d^k}, \overleftarrow{V_d^l}\right) + \gamma\left(\overleftarrow{U_d^k}, \overrightarrow{V_d^l}\right)}{2}$. PM then computes the ratio r of the smallest distance d_{ip} and the second smallest distance d_{iq} and report U_d^k is matched with V_d^l if r is below a threshold δ_d.

Algorithm 1 Dynamic Flow Wrapping

```
Input: $T_1, T_2$ flow size series, $c$ chuck size, $w$ band width
Output: $d$ flow distance of $T_1, T_2$
function DFW($T_1, T_2, c, w$)
    $d \leftarrow 0$
    Split $T_1$ into $\{U_i, i = 1 \ldots n\}, \forall i \neq n, len(U_i) = c, len(U_n) <= c$
    Split $T_2$ into $\{V_j, j = 1 \ldots m\}, \forall j \neq m, len(V_j) = c, len(V_n) <= c$
    for $i = 1$ to $min(m, n)$ do
        $d += $ Chuck-DFW($U_i, V_i, w$)
    end for
    if $n \neq m$ then
        Take the remaining chucks as $R$
        $d += ||R||$
    end if
    return $d$
end function

function Chuck-DFW($U, V, w$)
    $n \leftarrow$ length of U or V
    ▷ $dist$ is a 1-d array of size $(2nw + n - w^2 - w)$
    $dist \leftarrow [\ \ ]$
    for $(i, j) \leftarrow$ all indices in the restricted band do
        if $(i-1, j)$ is legitimate point then
            $d_1 \leftarrow$ extract distance of $(i-1, j)$ from $dist$
        else
            $d_1 \leftarrow \infty$
        end if
        same logic for $d_2$ and $d_3$ of $(i, j-1), (i-1, j-1)$
        $min\_d \leftarrow ||u_i - v_j|| + min\{d_1, d_2, d_3\}$
        put $min\_d$ to next position of $dist$
    end for
    return last element of $dist$
end function
```

Algorithm 1 Dynamic Flow Wrapping

So far the attacker can effectively uncover callers and callees and he then wants to know the relationship of these.

C. Contact Network

PM module will generate a sequence of call records, indicating “who called whom, when and for how long”, denoted as (t_s, u, v, d), where u, v are user IDs (IP addresses here), d is duration of the call. The attacker builds a weighted graph G with users as nodes and call frequencies as edges to better understand the relationship of users, called FN module.

After observing the local network for some time, graph G can express the communication relationship more precisely, which, to some extent, can help us handle the collisions of control channel correlation. For example, user u_1 makes a VoIP call with user v_1, time range of which happens to be exactly same with u_2 and v_2. There is a chance attacker mistakenly paired the four users by merely leveraging control channel information. But if lucky enough, graph G

can give us hints about the relationship among these four users, like u_2 called v_2 frequently before so they are more likely to be the VoIP pair.

Data channel correlation also benefits from FN module. The attacker compare flow similarities in a real-time manner. So to correlate a data channel d_i, he needs to compute its distance against every other lived data channel, which may be computationally intensive. Once again, graph G can give heuristics about which data channel to compare first.

Furthermore, graph G is a kind of social network. The attacker discovers communities and also the communication habits of users on it. Contact network gives attacker a higher perspective understanding of relationship among those users.

IV. Implementation

After verifying feasibility of the system, the attacker begins to implement it and proposes several optimizations to make it more efficient.

The attacker choose nDPI [24], an open source deep inspection engine maintained by ntop, as traffic identification engine. The attacker deploys nDPI on a gateway to identify control and data channels of WeChat VoIP services, based on the signature analyzed in section II.A. It continually calculates the flow size in byte per Δt (one second in the paper) for each channel and generate a sequence of events to PM module.

PM module runs an event loop and create two time windows, namely *data events* and *control events*. The time window is implemented by linked list with a timer for each node. When receiving a VoIP call *start event* or *end event* captured by DPI, it pushes the event into the corresponding window depending on the channel type.

PM module also receives flow statistics from DPI, called bytes-counting event. For data channel U_d^i (also for U_{cd}^i), it triggers a distance computation for U_d^i against all lived data channels after receiving k bytes-counting events of it instead of waiting for the end event. For control channel U_c^i, PM module must wait until the end event of it to perform a control channel correlation.

The attacker also comes up with some optimizations to reduce the complexity of computation and storage in PM, listed as follow.

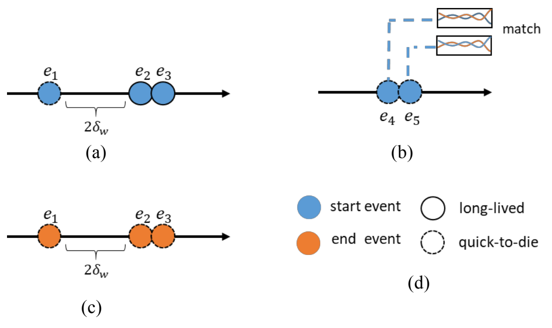


Figure 9. Long-lived and quick-to-die events.

Time Window: Considering low-latency characteristic of VoIP services, start/event proximity is a requirement for two peers being matched. So there is no need to compare distance of data channel d_i against all lived data channels but those within a time window of size δ_w. This also works for control channels when calculating confidence.

Quick-To-Die: Event proximity also helps us to avoid resources occupation for those channels that will never get a match. Figure 9 shows some examples about long-lived and quick-to-die events. For example Figure 9(a), PM gets a start event e_1 at t_1 and there is no other start event in window $[t_1 - \delta_w, t_1)$, timeout of this entry will be set to T_s which is short, such as seconds. Similar situation for e_2. Then e_3 comes and e_2 is found in its time window, then timers of e_2 and e_3 will be reset to T_l, which is longer, such as hours. And events in data channel are quick-to-die whenever they get matched (Figure 9(b)). All end events are quick-to-die regardless of proximity (Figure 9(c)). PM module will ignore bytes-counting packets for dead and matched entries for saving memory.

PM module sends messages about matched peers to FN module, which also runs an event loop. FN module maintains graph G and provide information to PM when necessary, for example control channel collisions happen or massive data channel comparisons need to be computed. FN also performs community analysis regularly and plot the network with echarts[25], a visualization tool maintained by Baidu.

V. Evaluation

The UVP prototype is evaluated in several aspects as following (i) The evaluation of DFW in real world traffics compared with classic DTW and windowed DTW. (ii) Parameters selection and the performance of control/data channels correlation. (iii) The evaluation contact network.

Dataset and experiment setup. We collect two datasets for different proposes and also design a simulation for further validation. **(i) Dataset A**: we deploy the traffic identification engine on a real world campus gateway and collected flow statistics of WeChat VoIP control/data channels from September 25th to October 30th, 2018. The number of WeChat users inside campus is around two thousand. All IP addresses are replaced by untraceable IDs to preserve the anonymity of users. 27330 records of WeChat VoIP calls are observed in total. **(ii) Dataset B**: The real world traffics is not labeled so the accuracy cannot be directly obtained. We generated Dataset B, containing 113 bi-directional data channels and validate data channel correlation on this dataset. **(iii) Simulation**: No control channel is observed on in the real world data so we implement another simplified control channel protocol of WeChat to test the ceiling capability of UVP control channel correlation.

A. Performance of DFW

Table 1 illustrates the time and error comparisons among DTW, window based DTW and DFW. The classic DTW has a quadric time and space complexity depends on the length of input sequences but metric result is the most accurate. WDTW and DFW both have linear time and space complexity but introduce error for reducing those possibly useless states. It should be noted that WDTW and DFW introduce more error when comparing flow similarities of unrelated traffics, i.e. not data channels from a pair of caller and callee. This is because such flows are more likely to result in a bad path for DTW, which is contradicted with the assumption of WDTW and DFW. However, in a low-latency system such as VoIP services, the error is negligible for flow correlation.

TABLE I. TIME AND ERROR COMPARISON OF DTW, WDTW AND DFW WITH DIFFERENT INPUT SERIES LENGTH

		Matched data channels			Unrelated data channels		
		100	500	900	100	500	900
Time (μs)	DTW	296.9	5455.6	14871.1	300.9	5561.6	15321.1
	WDTW	250.1	1095.0	2005.2	263.2	1130.5	2013.5
	DFW	38.1	158.1	227.7	40.1	163.2	230.2
Error (%)	DTW	0.00	0.00	0.00	0.00	0.00	0.00
	WDTW	2.31	0.46	0.31	4.94	4.37	3.90
	DFW	1.00	0.02	0.07	7.45	6.79	4.35

B. *Parameter Selection and Performance of Correlation*

In section III.B, several parameters are predefined. Based on the analysis of WeChat VoIP protocols, δ_{ib} is chosen as 2500 bytes for detecting Initial Burst and δ_c is chosen as 0.8 for correlating control channels.

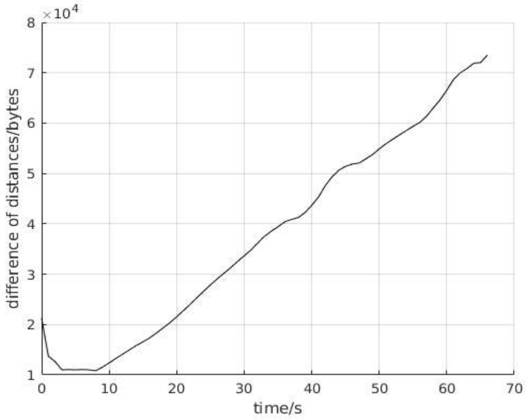


Figure 10. Difference of channel distances. Y axis denotes the difference between $\boldsymbol{\gamma}(\overrightarrow{\boldsymbol{u_k}}, \overleftarrow{\boldsymbol{v_k}})$ and average of $\boldsymbol{\gamma}(\overrightarrow{\boldsymbol{u_k}}, \overleftarrow{\boldsymbol{v_l}}), \forall \mathbf{l} \neq \mathbf{k}$.

For data channel correlation, DFW algorithm chops input series into chuck of length C. After comparing K chucks, PM module will have enough confidence according to the distance ratio threshold δ_d. Figure 10 explains how we decide these parameters. The x axis is time and y axis is the difference between distance $\left(\overrightarrow{U_d^k}, \overleftarrow{V_d^k}\right)$ and the average distance of $\gamma\left(\overrightarrow{U_d^k}, \overleftarrow{V_d^l}\right), \forall l \neq k$, where U_d^k and V_d^k are data channels from a pair of caller and callee, U_d^k and V_d^l from unrelated channels. The larger difference is, the more confident PM module is. There is no obvious difference during the first 10 seconds of VoIP call. But the gap continually increases with time. So the chuck size C is decided as 10 seconds and K as 4 chucks. Figure 10 also shows that the difference increases linearly with time. The average ratio is around 0.01 so we choose δ_d as 0.05 to avoid false negatives.

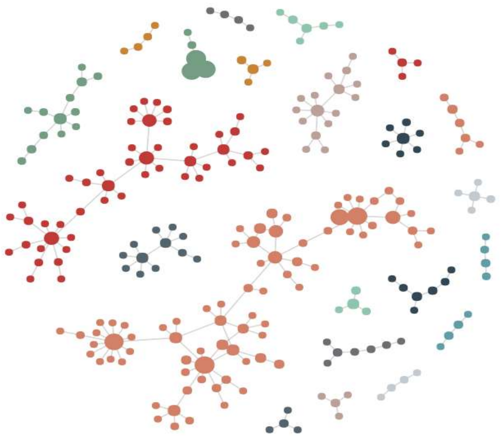


Figure 11. Contact network in real world dataset.

The real world traffics are not labeled but UVP correlates control channel and data channel with considerably high confidence. For 672 pairs of correlated control channels in dataset A, the minimum confidence is 89.5%, corresponding with a 10-second VoIP call. All confidences are above 96% for VoIP calls longer than one minutes. In dataset B, UVP achieves 100% accuracy and all confidences are above 97%. So it is reasonable to assume UVP also works for dataset A.

Control channel correlation may suffer from high-density VoIP calls. It is beyond our ability to find such a test bench so a simplified control channel protocol is implemented based on analysis in section II.A. Supposing the arriving time of VoIP calls is subjected to Poisson distribution and network delay to uniform distribution. Table 2 shows the result. A drop on accuracy is observed at 273.5 events per second, which far exceeds the density of campus environment (about 0.1 event per second). So control channel correlation is feasible in real world.

TABLE II. CONTROL CHANNEL CORRELATION ACCURACY (%)IN HIGH-DENSITY VOIP EVENTS

Density	<100	113.6	115.8	181.8	226.8	273.5	318.1
Accuracy	100	95.9	94.7	94.5	94.0	91.1	87.2

C. *Evaluation of Contact Network*

A contact network is built based on dataset A, shown in Figure 11. The node size denotes to the frequency and duration of users making VoIP calls. And nodes are colored according to their connectivity. FN reveals the structure of interpersonal relationships of victims. For example, two big communities (orange, red) can be directly observed which may represent the key characters in the subnet, such as students and facilities. And some users, with larger circle sizes, are playing more important roles in a community. FN helps the attacker understand the inner relationships from a higher perspective, which is a greater threat than linking a single pair of caller and callee.

VI. Conclusion and Discussion

In this paper, we thoroughly analyze the proprietary VoIP services of WeChat and propose UVP to uncover callers and callees also their relationship based on the gateway threat model. The experiments on real-world traffics and simulation test bench validate the effectiveness and efficiency of UVP.

It should be noted that UVP has several limitations. First, IP address is used as user ID in this paper. But this may fail in DHCP and NAT scenarios, which means a user can have different IP addresses and vice versa. An attacker can further leverage special ID carried in WeChat traffics to further correlate users from different IP addresses or distinguish users sharing one IP address. Also, VoIP traffics of WeChat are identified by deep inspection, which may sometimes be false due to changeful signatures. Last but not least, UVP can only correlate caller and callee whose VoIP traffics are both accessible. For example, if user u_a in campus A calls a user u_b in campus B, UVP cannot correlate them unless both control channels or both data channels of u_a and u_b are accessible to the attacker.

References

[1] MOMENTS, W., 2017. WeChat statistics of 2017.

[2] ROSENBERG, J., SCHULZRINNE, H., CAMARILLO, G., JOHNSTON, A., PETERSON, J., SPARKS, R., HANDLEY, M., and SCHOOLER, E., 2002. *SIP: session initiation protocol.*

[3] ZHANG, G. and FISCHER-HÜBNER, S.J.E.C.R., 2013. A survey on anonymous voice over IP communication: attacks and defenses, 1-33.

[4] ZHANG, L., KONG, Y., GUO, Y., YAN, J., and WANG, Z.J.I.C., 2018. Survey on network flow watermarking: model, interferences, applications, technologies and security *12*, 14, 1639-1648.

[5] WANG, X., CHEN, S., and JAJODIA, S., 2007. Network flow watermarking attack on low-latency anonymous communication systems. In *2007 IEEE Symposium on Security and Privacy (SP* IEEE, 116-130.

[6] ZHU, Y. and FU, H.J.C.C., 2011. Traffic analysis attacks on Skype VoIP calls *34*, 10, 1202-1212.

[7] NASR, M., BAHRAMALI, A., and HOUMANSADR, A., 2018. DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* ACM, 1962-1976.

[8] ZHU, Y., FU, X., GRAHAM, B., BETTATI, R., ZHAO, W.J.I.T.O.P., and SYSTEMS, D., 2010. Correlation-based traffic analysis attacks on anonymity networks *21*, 7, 954-967.

[9] BERNDT, D.J. and CLIFFORD, J., 1994. Using dynamic time warping to find patterns in time series. In *KDD workshop* Seattle, WA, 359-370.

[10] SRIVATSA, M., IYENGAR, A., LIU, L., JIANG, H.J.I.T.O.P., and SYSTEMS, D., 2011. Privacy in voip networks: Flow analysis attacks and defense *22*, 4, 621-633.

[11] WRIGHT, C.V., BALLARD, L., COULL, S.E., MONROSE, F., and MASSON, G.M., 2008. Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on* IEEE, 35-49.

[12] VERSCHEURE, O., VLACHOS, M., ANAGNOSTOPOULOS, A., FROSSARD, P., BOUILLET, E., and PHILIP, S.Y., 2006. Finding" who is talking to whom" in voip networks via progressive stream clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on* IEEE, 667-677.

[13] WRIGHT, C.V., BALLARD, L., MONROSE, F., and MASSON, G.M., 2007. Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob? In *USENIX Security Symposium*, 43-54.

[14] CHEN, S., WANG, X., and JAJODIA, S.J.I.N., 2006. On the anonymity and traceability of peer-to-peer VoIP calls *20*, 5, 32-37.

[15] SHAUN COLLEY, D.C., 2013. Practical Attacks Against Encrypted VoIP

[16] WANG, X., CHEN, S., and JAJODIA, S., 2005. Tracking anonymous peer-to-peer VoIP calls on the internet. In *Proceedings of the 12th ACM conference on Computer and communications security* ACM, 81-91.

[17] ROSENBERG, J., WEINBERGER, J., HUITEMA, C., and MAHY, R., 2003. STUN-simple traversal of user datagram protocol (UDP) through network address translators (NATs).

[18] MAHY, R., MATTHEWS, P., and ROSENBERG, J., 2010. Traversal using relays around nat (turn): Relay extensions to session traversal utilities for nat (stun).

[19] YUAN, Z., DU, C., CHEN, X., WANG, D., and XUE, Y., 2014. SkyTracer: Towards fine-grained identification for Skype traffic via sequence signatures. In *ICNC*, 1-5.

[20] BONFIGLIO, D., MELLIA, M., MEO, M., ROSSI, D., and TOFANELLI, P., 2007. Revealing skype traffic: when randomness plays with you. In *ACM SIGCOMM Computer Communication Review* ACM, 37-48.

[21] LEVINE, B.N., REITER, M.K., WANG, C., and WRIGHT, M., 2004. Timing attacks in low-latency mix systems. In *International Conference on Financial Cryptography* Springer, 251-265.

[22] FANG, J., ZHU, Y., and GUAN, Y., 2016. Voice Pattern Hiding for VoIP Communications. In *Computer Communication and Networks (ICCCN), 2016 25th International Conference on* IEEE, 1-9.

[23] DUPASQUIER, B., BURSCHKA, S., MCLAUGHLIN, K., and SEZER, S.J.I.J.O.I.S., 2010. Analysis of information leakage from encrypted Skype conversations *9*, 5, 313-325.

[24] DERI, L., MARTINELLI, M., BUJLOW, T., and CARDIGLIANO, A., 2014. ndpi: Open-source high-speed deep packet inspection. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International* IEEE, 617-622.

[25] BAIDU, echarts.