Multimed Tools Appl (2019) 78:3493–3509
https://doi.org/10.1007/s11042-018-6281-1

A sensitive network jitter measurement for covert timing channels over interactive traffic

Quanxin Zhang[1] · Hanxiao Gong[1] · Xiaosong Zhang[1,2] · Chen Liang[1] · Yu-an Tan[1]

Received: 25 January 2018 / Revised: 13 May 2018 / Accepted: 15 June 2018 /
Published online: 30 June 2018

Abstract In order to reflect the network transmission quality, some network state feedback mechanisms are provided in the network protocol. In the RTP, the jitter of the packet transmission delay is fed back through the jitter field in the RTCP packet. This feedback value is a very important reference data when the covert timing channel is established. However, the sending frequency of the RTCP packet is low and the feedback value of the RTCP packet are only the jitter value of the last RTP packet associated with this RTCP packet when it is sent. Therefore, the jitter feedback mechanism in the existing RTCP protocol has the problem of lack of feedback on the network state during the period between two RTCP data packets. As a result, the feedback value is highly susceptible to extreme values, which prevents it from providing an accurate numerical reference for establishing covert channels. Therefore, in this paper, a buffer was established between the last RTCP packet and the current RTCP packet. And we choose to set the interval is n RTP packets and record the corresponding position jitter value in the buffer. The data in the buffer is averaged, and the mean value is weighted and averaged with the jitter value of the current RTCP packet as a new jitter feedback value. The effect of the extreme value on the feedback value is reduced, thereby

✉ Yu-an Tan
tan2008@bit.edu.cn

Quanxin Zhang
zhangqx@bit.edu.cn

Hanxiao Gong
13718106764@163.com

Xiaosong Zhang
zxs0224@163.com

Chen Liang
1342313537@qq.com

[1] School of Science and Technology, Beijing Institute of Technology, Beijing 100081, China

[2] Department of Computer Science and Technology, Tangshan University, Tangshan 063000, China

it contribute to the improvement of the feedback energy for the state of the network. In addition, the bit error rate generated by establishing a simple covert timing channel for data transmission under different network conditions is compared with the change of two jitter feedback values. It is verified that there is a positive correlation between the feedback value of the new feedback mode and the error rate. through the comparison It is verified that the new feedback method can provide a more accurate reference for the establishment of covert channels.

Keywords Jitter · RTCP · Covert timing channel · IPD

1 Introduction

As the prevalence of multimedia social network, the leakage of privacy information become ever more serious [21, 27, 30, 36, 37, 41–43, 45, 46]. An enormous amount of research effort goes into privacy protection mechanisms to preserve the users' personal information [17, 18, 28, 29, 44].Information monitoring technology can prevent information leakage of covert channels to a certain extent [8, 12, 13]. However, system inevitably has many covert channels because of system design, equipment design and implementation of loopholes [31]. The covert channels can be used to disclose information to unauthorized devices in violation of the security policy. The covert channels will pose a threat to the network information security [40]. And the covert channels include covert storage channels and covert timing channels in trusted Computer System Evaluation Criteria(TCSEC) [32].

Covert storage channel refers to different processes reading and writing the same shared variable for information transmission. And covert timing channels refers to the adjustment of the data state during information transmission, and the acquisition of information by comparing the data state with its unregulated state during information reception. There are many ways to design a covert timing channel, and they are perfect methods [5]. The main methods are: (1) based on the number of packets sent during a period of specified time [6]. (2) based on the time delay between the transmissions of data packets [16].

Due to the existence of the jitter, different degrees of offset may occur in the time latitude of the data packet during transmission [20]. As a result, the data transmitted by the covert channels may generate errors during demodulation. Actually, jitter is a deviation between the ideal time and the actual time of an event. In network communication, the transmission of data is influenced by the delay of the network, resulting that the time spacing of arrivals from different data packets are different. For example, in order to send a series of data packets, the transmission interval of these data packets is determined at the sending end, but the interval of the data packets varies as the data packets arrive at the receiving end due to the dynamic changes of the network. This change is the jitter that will be discussed in this paper. The dynamic changes in the network have caused many problems in data transmission in the network [22].

In general, the Inter-Packet Delay (IPD) in the network follows an unknown random distribution, such as normal distribution, poisson distribution and so on. However, the temporal covert channel algorithm is mainly based on modifying the distribution of IPD, which generates a given IPD or a given distribution of IPD to create a covert channel. The IPD is usually changed within the range indicated by the jitter of the network [33]. If the jitter is

too large, it will lead to the original method has a great bit error rate. Thus the transmission of information cannot be demodulated correctly.

Real-time Transport Control Protocol (RTCP) packets can report the network quality in the network transmission process, including jitter. However, due to the restrictions of calculation equation and the reported mechanism of the RTCP packets, the jitter data of the RTCP packets have some shortcomings,which are mainly in the following two aspects: (1) Due to the (2), when the network changes sharply, the change of jitter in the RTCP is still small. (2) RTCP periodically sends control packets to members of the network session to report the network quality to each member,and the reported result is calculated by the current location. Therefore, even if the network conditions are different, the jitter values calculated by the RTCP will not be different because their report on the calculation result of the current location.

There are many researches in the fields of multimedia data transmission and multimedia data detection [4, 14, 47]. Through these methods for multimedia file comparison and feature selection [1, 9–11, 23], the security of multimedia transmission can be guaranteed to a great extent. Many existing technologies can prevent information leakage. However, covert channels are unavoidable and pose a great threat to the security of multimedia data transmission. The jitter value is an important reference for the establishment of covert channels, but the jitter reporting mechanism in the network is flawed. However, until now, no research is relevant to the jitter reporting mechanism in the network.

Our contribution includes three parts:

(1) We introduce the effect of jitter for covert timing channels and present that the traditional jitter calculation method is lack of the distinction among some network status.
(2) We put forward a new jitter calculation method called sensitive jitter.
(3) We compare the two algorithms and prove the shortcomings of the traditional algorithm and the effectiveness of the new algorithm through experiments.

The remainder of our paper is organized as follows: In Section 2, the principle and disadvantage of traditional jitter in RTCP is analyzed. In Section 3, Sensitive Jitter Algorithm is proposed to avoid the shortcomings and disadvantages. Then in Section 4, the effectiveness of the Sensitive Jitter algorithm is verified and the Sensitive Jitter with the jitter in the RTCP are compared through experiments Finally, the conclusion is reached and the future research direction is discussed. in Section 5.

2 Background

2.1 Jitter measurement

Real-time transport protocol (RTP) [2] It is a transport protocol for multimedia data streams on the Internet, which is published by the IETF as RFC1889(The latest version is RFC3550). RTP is defined to work in case of one-to-one or one-to-many transmission for the purpose of providing time information and enabling stream synchronization [34]. A typical application of RTP is UDP, which can also work on the top of other protocols such as TCP [7], VoIP or other real-time transmission software by using the RTP protocol [15]. Hence the VoLTE call used in this experiment is based on RTP. RTP only guarantees the transmission of real-time data but may not provide a reliable transport mechanism for the

sequential delivery of data packets nor a traffic control or congestion control, which can provide these services by RTCP [25, 35].

Real-time transport control protocol (RTCP) It is responsible for the management of transmission quality, exchange of control information among the current application processes and provision of services about traffic control and congestion control [38]. During an RTP session, each participant periodically transmits RTCP packets containing statistics about the number of packets sent and the number of packets lost and etc. Therefore, the server can use this information to dynamically change the transmission rate, or even the payload type. RTP and RTCP work together to optimize transmission efficiency with efficient feedback and minimal overhead. It is particularly suitable for real-time data transmission network.

The 32 bits in the header of an RTCP packet represent the estimated statistical variance of arrival times of RTP packets, which is measured in unit of time and expressed as an unsigned integer. Jitter (J) is defined as the average deviation (smoothed absolute value) of the time difference between the receiving end and the sending end in a pair of packets. At the sending end, the structure of the header of the data packet is shown in Fig. 1.

The calculation equation in the RTP protocol is as shown in (1) and (2), and calculation result equals to the difference between the relative transmission time of two packets. The relative transmission time refers to the RTP timestamp of the packet and the clock of receiving end at the arrival time, S_i is the RTP timestamp of packet i, and R_i is the value of the arrival time of packet i in units of RTP timestamp, for two packets i and j, D can be expressed as:

$$D(i, j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_i) - (R_i - S_i) \tag{1}$$

$$J_i = J_{i-1} + \frac{|D(i-1, i)| - J_{i-1}}{16} \tag{2}$$

J_i is recursively calculated by each package i, which is the jitter value reported in the RTCP protocol.

2.2 Noise of a covert timing channel

Jitter is a very important indicator for the design of covert timing channels such as the paper by Archibald et al. [3] The concept of guard band is designed to prevent false demodulation results due to jitter in the network. For example, the data is modulated at the sending

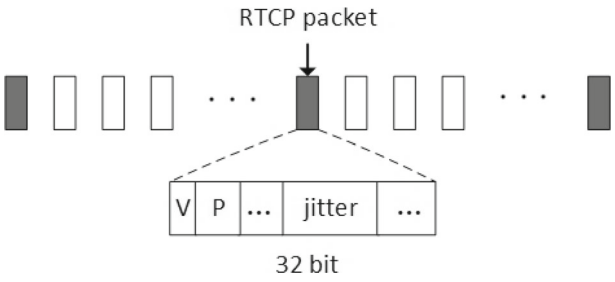


Fig. 1 Jitter value reported by RTCP packets

end, using the packet interval of 10ms and 30ms to represent the binary value of 0 and 1 respectively. The threshold is set to 20ms when demodulated at the receiving end. when the interval between two data packets is less than 20ms, the binary value of 0 was received. On the contrary, the binary value of 1 is received. As shown in Fig. 2, when the jitter is less than 10ms, the packet interval change will not impact the demodulation process.

However, when the jitter value is over 10ms, demodulation data will be wrong.Because of the jitter, which can change the binary value, the information will not be correctly restored.

From this, It can be seen that the jitter index has a great influence on the design of covert timing channels. It is necessary to know the jitter in the network for the concrete realization of covert channel.

2.3 Criteria of desirable jitter measurement

When data is transmitted through the network, the network protocols try to provide the user with a reliable network environment. And through some mechanisms, they provide the user with the network transmission quality and report the network delay, jitter and other information. However, a practical jitter algorithm in reporting networks should have the following features: (1) Simplicity: calculation algorithms should be simple and still contain enough useful information. (2) Distinction: the jitter generated by the algorithm should reflect the state of the network and distinguish the different network status. 3) Low overhead: when giving feedback about jitter, the network cost should not be increased so as not to affect the normal network transmission.

RTCP periodically sends control packets and reports the network quality to members of the network session to. The RTCP packets report the result calculated by the current location. This feedback algorithm is simple calculation method and can reflect the trend of jitter in the network to a certain extent, which has better simplicity. Since the RTCP packets are about 1% of the total packets sent with less overhead.

However, this feedback algorithm performs poorly in terms of distinction because of the low frequency of RTCP packets sending. The value of the feedback result is the state of the jitter in the local networks. If the local networks are similar, the jitter algorithm in RTCP cannot distinguish the different network status, and thus the performance of the distinction is poor.

This paper will validate these questions experimentally, propose improvements in terms of poorly differentiated performance, and experimentally validate the effectiveness of the algorithm.

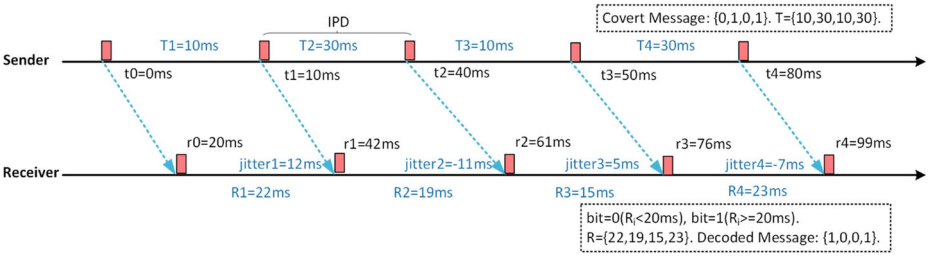


Fig. 2 Jitter noise of covert timing channels

3 Sensitive jitter algorithm

The proposed sensitive jitter algorithm is modified based on the traditional jitter feedback algorithm in the RTCP protocol to ensure the sensitive jitter algorithm stronger distinction of network state without adding extra network overhead.

3.1 Design of algorithm

The lack of distinction of jitter algorithms in RTCP is mainly due to the low frequency of RTCP packets sending. The RTCP packets are sent at large intervals so that they can only express the jitter value of the network in a much more localized area. For the purpose of improving distinction, the sampling frequency is increased, but this method will result to the increase of the number of packets transmitted in the network. In order not to increase the extra network load, we still use the traditional jitter reporting frequency, but make the changes on the reported values.

The traditional jitter calculation method is based on the statistic variance estimation jitter value, whose advantage is that it is easy to calculate and reflect jitter data dramatic changes to some extent. The method of calculating jitter still uses the traditional statistical variance estimation value to represent the jitter value, but uses a larger sampling frequency. The jitter values of every n packets are recorded. If all these values are sent to the user for reporting, the network overhead will be greatly increased. The traditional RTCP packet is at intervals of m TCP packets and the new one is at intervals of n TCP packets. Then, the number of RTCP packets becomes m/n times as many as the traditional one. Instead of sending all the recorded values, AVG_i is the mean value of the recorded values between the RTCP packets $i-1$ and i,which reflects the network status between two RTCP packets to some extent. The equation of AVG_i is as shown in (3), and the algorithm is as shown in Fig. 3.

$$AVG_i = \frac{1}{r} \sum_{r=1}^{r=\lfloor m/n \rfloor} J_{k+rn} \tag{3}$$

When transmitting RTCP data packets, jitter values of the current position need to be considered for feedback of the local information. Therefore, the jitter value evaluation equation at the feedback position SJ_i is as shown in (4).

$$SJ_i = a \times AVG_i + (1-a) \times J_i \tag{4}$$

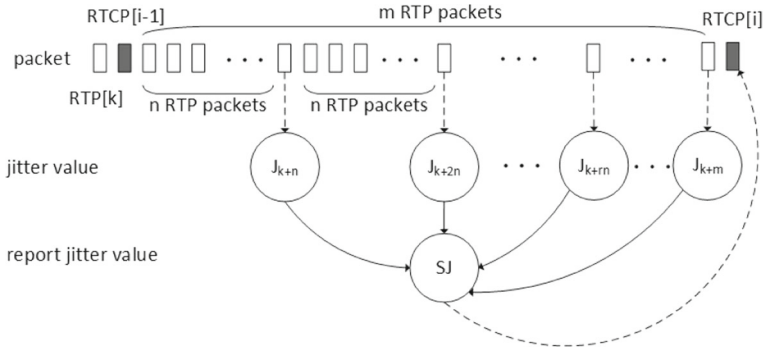


Fig. 3 Sensitive measurement by averaging multiple sample values

The RTCP packet is sent at the original feedback position, and SJ_i is sent for reporting. The average of the time between two RTCP packets is about 2 seconds in our experiment, which means that the traditional jitter algorithm cannot provide feedback on the jitter value for a long period of time. The time for each record is approximately 0.19 seconds. This will shorten the reported blank time to 10% of the original. When calculating SJ_i values, only simple averaging operations and weighted summation operations are needed. The Time of calculation is very short and can be ignored.

3.2 Comparative analysis

The traditional jitter algorithm is the jitter algorithm used in RTCP. The sensitive jitter algorithm is a new algorithm based on the jitter algorithm used in RTCP. The differences between them are mainly as follows.

Firstly, the jitter algorithm used in the RTCP performs poorly on the distinction. In order to enhance the distinction of the jitter algorithm, sensitive jitter adds the sampling points and mathematically averages the jitter of these sampling points and the jitter of reported locations. It leads to increase the state feedback of jitter between RTCP packets, retain the ability of feedback on local data and make the sensitive jitter algorithm behave better than the traditional one on distinction.

Secondly, the sensitive jitter algorithm inherits the calculation method used in the traditional jitter, so it also possesses the simplicity of the traditional jitter algorithm. What is more, it also has low overhead as compared with the traditional jitter, because it does not add extra packet transmission.

4 Experiment results and analysis

4.1 Experimental setup

In the experiment, two smart mobile devices were used as the sending end and the receiving end of the data packets respectively. The operation system is the Android 5.1, which installs VoLTE software for network video communications. TCPdump is used to assist capture operations. Wireshark is used to capture and analyze the packet data and python is used to analyze data and simulate the network transmission status.

TCPdump TCPdump is one of the most powerful network data acquisition and analysis tools in Linux, which provides the source code and exposes the interface because of its strong scalability. TCPdump exists in the basic Linux system, so ordinary users cannot execute it, but users with root privileges can execute it directly to get the information on the network. In this experiment, TCPdump was loaded into two smart mobile devices at the same time in order to obtain the data packets at both ends of the communication.

Wireshark Wireshark (formerly known as Ethereal) is a software for network packet analysis, whose function is to extract the network packet and display the most detailed network packet information as much as possible. User can find related content by entering search criteria (such as protocol type, IP address, port number and etc.) in the display filter to get related packets and filter out irrelevant packets.

VoLTE VoLTE [24, 39]calls only have RTP packets and RTCP control packets and the data packets have a strict sequence relationship [19, 26]. Therefore, in this experiment, the out-of-order problem of data packets in the transmission process can be dealt with according to the serial number of them and calculate the accurate IPD value. Because VoLTE calls only contain RTP and RTCP packets, it is easier to find the packets required by the experiment through the IP address, the port number and the protocol type, after capturing the packets

4.2 Network traffic capture and IPD analysis

Firstly, because ordinary users can not normally execute TCPdump, root privileges for the two smart mobile devices are obtained. But users with root privileges can do that. After that, TCPdump is installed in these two devices to prepare for the capture operation needed in the experiment. In the experiment, the two devices were used to make VoLTE calls in the LTE network. When two devices started transmission, TCPdump is used to capture the packets from the sending end and the receiving end at the same time. In the process of experiment, the capture duration of each group was 5 minutes and a total of 16 groups of data were obtained.

Secondly, the sending end and receiving end of the captured packet results were imported to Wireshark for analysis. Through the IP address and the port number at the time of communication, the data packets during the VoLTE calls were separated from all the data packets. Wireshark is used to parse out each packet and the analytical results were exported according to the time property column and the serial number property column. At the same time, the RTCP data packets in the data packet were filtered, the jitter in the RTCP header was parsed and the serial number property column of the RTCP data packets and the reported jitter value data column were derived.

Next, the export result is processed. The packets captured at the sending end and receiving end were matched according to the sequence number with the purpose to solve the disorder of data packets caused by the delay of the network. If an out-of-order situation occurs in the network, an error would occur when calculating the IPD value without adjusting the order of data packets in both ends. If a packet loss occurs in the network, the corresponding unmatched packet is deleted at the sending end. The calculated IPD equation is shown in (1), and the difference corresponding to the rest packets with the same location will not be affected.

Finally, the IPD value sequence and the jitter value sequence separately after previous processes are calculated according to the (1) and (2).

4.3 A covert timing channel for experiment

In this experiment, the packet interval of packets was modulated according to the way of changing the time delay among transmission data packets to realize the covert channel. The packet interval of the sending end was adjusted by using the packet interval of 10ms and 30ms to represent the binary value of 0 and 1 respectively. The threshold was set to 20ms when demodulating at the receiving end, which means that when the interval between two data packets was less than 20ms, the binary value of 0 was received. On the contrary, the binary value of 1 was received.

Before simulating the data sending through the covert channel, the data sent in this experiment was an alternating binary sequence of binary 0 and 1.

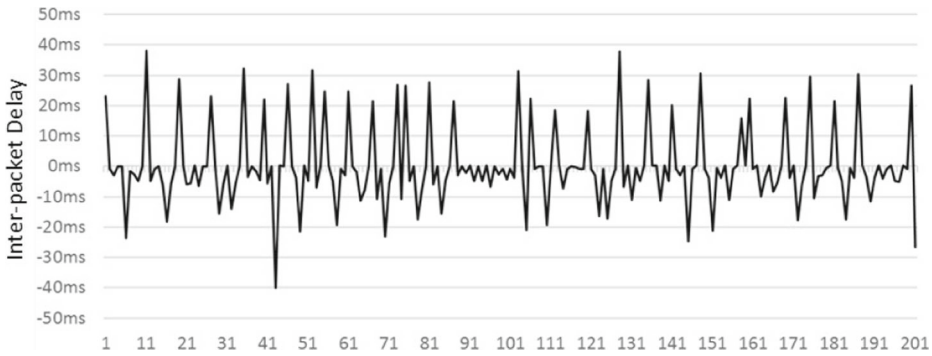


Fig. 4 IPD sequence of test traffic

4.4 Jitter measurement

The obtained jitter sequence and IPD value sequence were analyzed. IPD value sequence is shown in Fig. 4, the abscissa is the packet number, the ordinate is the IPD calculation result in milliseconds. The corresponding jitter value sequence is shown in Fig. 5, where the abscissa is the sequence number of the packet and the ordinate is jitter. The calculation result is in milliseconds.

In Fig. 4, the gap between the sending end and the receiving end in the network is relatively large. The maximum value in Fig. 4 reaches 0.04s, and many values exceed 0.02s. However, in Fig. 5, the corresponding jitter value calculated by (2), shows that the network status of the response is much smoother and the maximum value does not exceed 0.012s. All of above show that the jitter calculation method in RTCP makes the network with drastic changes be shown in a gentler way.

Then, the covert channel was established and the disturbance to the modulation result of the transmitted data to simulate the network transmission affected was added. First, the IPD value calculated was according to the actual captured packets data added to the modulation result. Then, the jitter calculated by the jitter equation specified in the RTCP protocol was added to the same modulation result.

Fig. 5 Jitter sequence of test traffic

Table 1 Generating function coefficient selection

Network status number	Random coefficient p	Jitter coefficient q
SCENARIO 1 (S1)	0.2	0.3
SCENARIO 2 (S2)	0.4	0.3
SCENARIO 3 (S3)	0	1
SCENARIO 4 (S4)	1	0.5
SCENARIO 5 (S5)	1.25	0.75

After the disturbance was added, the simulation of the demodulation process was performed according to the preset demodulation mode and the data was demodulated to obtain the received binary sequence. Compared the binary sequence data of the sending end with that of the receiving end, the bit error rate of analog transmission results of two groups according to different disturbance standards were calculated. In the experiment, the bit error rate of the data set of the IPD perturbation calculated according to the actual captured packet data was 14.06%, and the bit error rate of the data set of jitter perturbation was 12.51%. Therefore, it can be concluded that when design a covert timing channel, if the feedback of jitter value in the RTCP packet is taken as a basis, the actual result will be worse than the theoretical result.

Next, according to the analysis result of Wireshark, the position numbers reported by RTCP packets were extracted. Under the condition of ensuring the same jitter values at corresponding numbers reported by RTCP packets, another four sets of jitter value sequences were randomly generated to simulate the other four network conditions. Data generating equation is shown in (5). Generating function coefficient selection is shown in Table 1.

$$J_{Si} = abs(random.gauss(m, v)) \times p_{Si} + J_{S3} \times q_{Si} \tag{5}$$

Then, the five network results were compared. Figure 6 shows the comparison among the jitter value of the four network status and the jitter value of the original captured packet network. Among them, the SCENARIO 3 (S3) is jitter value series of the original network status, the SCENARIO 1 (S1), SCENARIO 2 (S2), SCENARIO 4 (S4) and SCENARIO 5 (S5) are the generated simulation data.

The $random.gauss(m, v)$ function is a Gaussian random number generator which generates a random number satisfying the Gaussian distribution with m as the mean and v as the variance of the Gaussian function. In this experiment, the variance of jitter is taken as the variance of Gaussian random numbers and the mean of jitter is taken as the mean of Gaussian random numbers.The value of J_{S3} is the jitter value calculated from the actual packet capture data.The parameter p in the equation represents the randomness of the generated data, and the parameter q represents the correlation between the generated data and the original data. When the parameter in the equation is larger, the calculated value of equation 5 is larger, which means the larger the jitter value is, the more unstable the network is. The new jitter sequence value is generated by controlling coefficient , $S1 < S2 < S3 < S4 < S5$. The results of the five sets of jitter sequences at number 89, 183 and 284 are shown in Fig. 6.

The jitter values calculated by the RTCP protocol are different. However, they report only the calculation result of the current location. It can be clearly seen in Fig. 6 that the five network status are different. From S1 to S5, it becomes more jittery in turn and the average

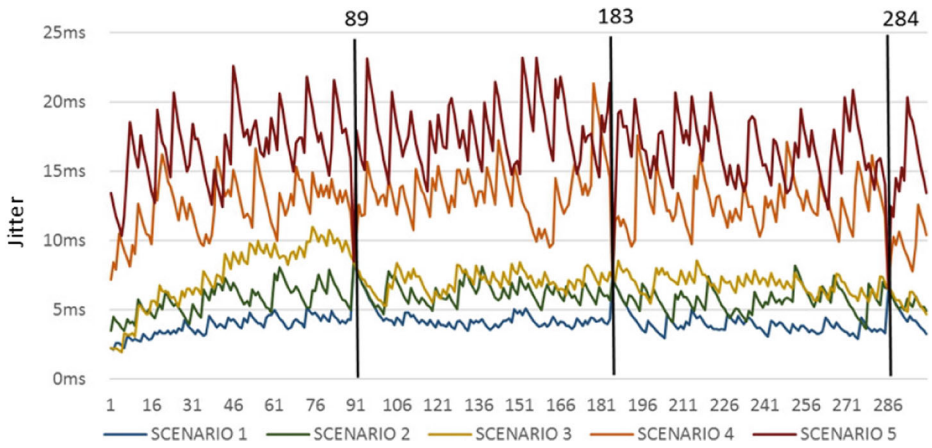


Fig. 6 Five network scenarios of same jitter

is gradually increasing.The reason for this phenomenon is that the sending interval of RTCP packets is too large. In our experimental packets capture data, RTCP packets are sent every about 200 RTP packets. The interval is approximately 2s. What a large time interval results in feedback values is that the feedback value are highly susceptible to extreme values and do not give a correct reference for establishing covert channels, which Affects covert channel data transmission and threatens data transmission. However, these five network status have the same jitter value in the same RTCP feedback position, though the network jitter values of S1, S2, S4 and S5 are randomly generated by S3. The probability of this coincidence in the actual transmission is small. It can also indicate that the feedback network of RTCP uses the current feedback value as the jitter value, which poses a problem that the feedback frequency is low.

After that, the data sending in the covert channel was simulated. Jitter values of five different networks were generated. Perturbations were added to the modulated data sequence according to the jitter values of the five different networks generated in the previous experiment. The transmission results in these five networks were then simulated.

After the perturbations were added, the simulation of the demodulation process was performed. And according to the preset demodulation mode, when the convert channel was established, the data was demodulated to obtain the five received binary sequences. The binary sequence data of the sending end and that of the receiving end were compared to calculate the bit error rates. The bit error rate of the data set perturbed by the S1, S2, S3, S4 and S5 were 8.84%, 12.75%, 14.06%, 17.17% and 19.07% respectively.

The jitter values of these five network status were used as the basis of the disturbance. When the simulation was performed on the covert channel, the bit error rate varied greatly. The bit error rate in the S1 data set was only 8.84%, but the bit error rate in the S5 data set reached 19.07% which is more than twice the S1 data set. However, the jitter values in the RTCP packets are the same. That is, the original jitter feedback mechanism does not show the differences among these five networks. In addition, the establishment of covert channels for data transmission in these five types of networks has a wide range of error rates, which reflected the RTCP performs poorly in terms of distinction.

5 Evaluation of sensitive jitter measurement

According to the algorithm, a buffer was added to the sending end to record the jitter value of every 20 packets in the jitter sequence. At the position to be reported, all the recorded values after the last reported position were calculated by the (5). The coefficient p is 0.5 and the coefficient q is 0.5. The weighted average results were reported as jitter values at the position to be reported. In Fig. 7, The jitter value of each position before the improvement is compared with the improved jitter value, and the jitter values calculated by the two algorithms in the five types of networks are compared with the bit error rates calculated in previous experiments. Then, the modified jitter values of different locations in the five network status are displayed.

Through the jitter value of the improvement before and after, and the change of bit error rate in different network status, it can be seen that, in the five network status, the trend of the jitter value in the corresponding location and the change trend of the bit error rate are the same. The bit error rates of these five groups of networks increase in sequence and the jitter feedback values of the corresponding positions after the improvement also perform a similar trend. However, the unimproved jitter feedback value at the corresponding location is invariant. In the five network status, the values sent by the three RTCP packets are values of the first group and have not been changed, as shown in the Fig. 7. The above results show that the improved jitter reporting method can better reflect the overall status of the network compared with the traditional method and can distinguish the network status that cannot be distinguished previously. In order to record the jitter value in the network, only a buffer is added in the sending end, which does not increase the number of RTCP packets in the network and the additional network load are not added to meet the low overhead of the previous RTCP jitter algorithm. The calculation method has not changed with the addition of the mean calculation of sampling points. It meets the simplicity and is effective.

Limitation: The paper only uses a single covert channel establishment method, and only uses the bit error rate to measure the network performance.

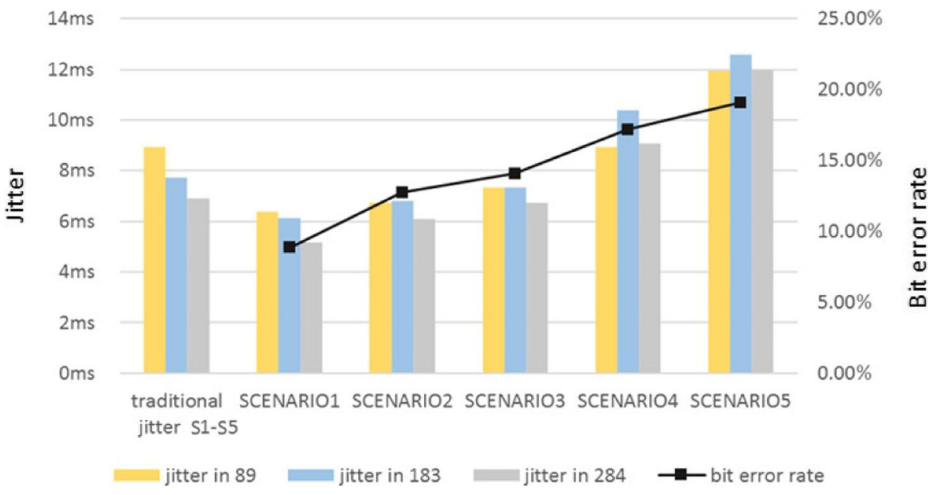


Fig. 7 Relevance between Bit-error-rate and jitter

6 Conclusion

This paper focuses on the RTCP algorithm jitter feedback mechanism and jitter numerical calculation algorithm. The features of RTCP feedback jitter algorithm are evaluated from three aspects: simplicity, distinction and low overhead by analyzing multimedia data transmission during a VoLTE call. Aiming at solving these problems Of poor performance in distinction of the traditional jitter, an improved algorithm is proposed. From these three aspects, this paper analyzes the jitter algorithm of the RTCP feedback mechanism, and verifies the advantages of the new algorithm compared with the traditional one through experiments over covert timing channels.

In the future, more experiments with different ways of establishing covert channels are needed to be done. The algorithm in this paper mainly improves the shortcomings of jitter values in RTCP packets in RTP. The problem of jitter feedback in different network transport protocols will be taken into consideration in the next step. And covert timing channels can be combined with other research results to produce much more interesting results.

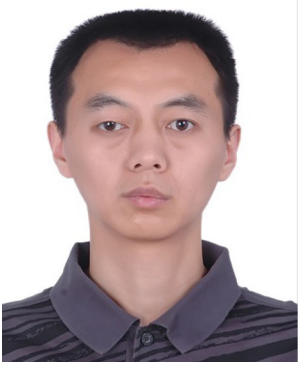
Acknowledgements This paper was supported by the National Natural Science Foundation of China (No.U1636213).

References

1. Al-Ayyoub M, AlzuBi S, Jararweh Y, Shehab MA, Gupta B (2016) Accelerating 3d medical volume segmentation using gpus. Multimed Tools Appl 77(4):4939–4958. https://doi.org/10.1007/s11042-016-4218-0
2. Andreadis A, Rizzuto S, Zambon R (2016) A cross-layer jitter-based tcp for wireless networks. Eurasip J Wireless Commun Network 2016(1):191. https://doi.org/10.1186/s13638-016-0695-0
3. Archibald R, Ghosal D (2012) A covert timing channel based on fountain codes. In: IEEE International conference on trust, security and privacy in computing and communications, pp 970–977. https://doi.org/10.1109/TrustCom.2012.21
4. Atawneh S, Almomani A, Bazar HA, Sumari P, Gupta B (2017) Secure and imperceptible digital image steganographic algorithm based on diamond encoding in dwt domain. Multimed Tools Appl 76(18):18,451–18,472. https://doi.org/10.1007/s11042-016-3930-0
5. Biswas AK, Ghosal D, Nagaraja S (2017) A survey of timing channels and countermeasures, 50. https://doi.org/10.1145/3023872
6. Cabuk S, Brodley CE, Shields C (2004) Ip covert timing channels: design and detection. In: Proceedings of the 11th ACM conference on computer and communications security, pp 178–187. https://doi.org/10.1145/1030083.1030108
7. Carle G, Biersack EW (1997) Survey of error recovery techniques for ip-based audio-visual multicast applications. IEEE Netw 11(6):24–36. https://doi.org/10.1109/65.642357
8. Chang X, Yang Y (2017) Semisupervised feature analysis by mining correlations among multiple tasks. IEEE Trans Neural Netw Learn Syst 28(10):2294–2305. https://doi.org/10.1109/TNNLS.2016.2582746
9. Chang X, Nie F, Wang S, Yang Y, Zhou X, Zhang C (2014) Compound rank-k projections for bilinear analysis. IEEE Trans Neural Netw Learn Syst 27(7):1502–1513. https://doi.org/10.1109/TNNLS.2015.2441735
10. Chang X, Nie F, Yang Y, Zhang C, Huang H (2016) Convex sparse pca for unsupervised feature learning. Acm Trans Knowl Discov Data 11(1):3:1–3,16. https://doi.org/10.1145/2910585
11. Chang X, Ma Z, Lin M, Yang Y, Hauptmann AG (2017) Feature interaction augmented sparse learning for fast kinect motion detection. IEEE Trans Image Process 26(8):3911–3920. https://doi.org/10.1109/TIP.2017.2708506
12. Chang X, Ma Z, Yi Y, Zeng Z, Hauptmann AG (2017) Bi-level semantic representation analysis for multimedia event detection. IEEE Trans Cybern 47(5):1180–1197. https://doi.org/10.1109/TCYB.2016.2539546

13. Chang X, Yu YL, Yang Y, Xing EP (2017) Semantic pooling for complex event analysis in untrimmed videos. IEEE Trans Pattern Anal Mach Intell 39(8):1617–1632. https://doi.org/10.1109/TPAMI.2016.2608901
14. Chen Z, Peng L, Gao C, Yang B, Chen Y, Li J (2017) Flexible neural trees based early stage identification for ip traffic. Soft Comput 21(8):2035–2046. https://doi.org/10.1007/s00500-015-1902-3
15. Denby L, Landwehr JM, Mallows CL, Meloche J, Tuck J, Xi B, Michailidis G, Nair VN (2007) Statistical aspects of the analysis of data networks. Technometrics 49(3):318–334. https://doi.org/10.1198/004017007000000290
16. Gianvecchio S, Wang H, Wijesekera D, Jajodia S (2008) Model-based covert timing channels: automated modeling and evasion. In: International Symposium on recent advances in intrusion detection, pp 211–230
17. Guan Z, Li J, Wu L, Zhang Y, Wu J, Du X (2017) Achieving efficient and secure data acquisition for cloud-supported internet of things in smart grid. IEEE Internet Things J 4(6):1934–1944. https://doi.org/10.1109/JIOT.2017.2690522
18. Guan Z, Li J, Zhu L, Zhang Z, Du X, Guizani M (2017) Towards delay-tolerant flexible data access control for smart grid with renewable energy resources. IEEE Trans Indus Inform 13(6):3216–3225. https://doi.org/10.1109/TII.2017.2706760
19. Hastyo WJ, Kang CG (2014) Lte network emulator for volte service. Nmr Biomed 22(2):191–198
20. He B, Yan S, Zhou X, Lau VKN (2017) On covert communication with noise uncertainty. IEEE Commun Lett 21(4):941–944. https://doi.org/10.1109/LCOMM.2016.2647716
21. Huang Z, Liu S, Mao X, Chen K, Li J (2017) Insight of the protection for data security under selective opening attacks. Inform Sci 412-413:223–241. https://doi.org/10.1016/2017.05.031
22. Imputato P, Avallone S (2018) An analysis of the impact of network device buffers on packet schedulers through experiments and simulations. Simul Model Pract Theory 80:1–18. https://doi.org/10.1016/2017.09.008
23. Jararweh Y, Al-Ayyoub M, Fakirah M, Alawneh L, Gupta B (2017) Improving the performance of the needleman-wunsch algorithm using parallelization and vectorization techniques. Multimed Tools Appl 3:1–17. https://doi.org/10.1007/s11042-017-5092-0
24. Jouihri Y, Guennoun Z, Chagh Y, Zahi D (2017) Towards successful volte and vowifi deployment: network function virtualization solutions benefits and challenges. Telecommun Syst 64(3):467–478. https://doi.org/10.1007/s11235-016-0186-y
25. Jung TJ, Seo KD (2016) A client-driven media synchronization mechanism for rtp packet-based video streaming. J Real-Time Image Proc 12(2):455–464. https://doi.org/10.1007/s11554-015-0497-3
26. Kumar R, Ganguly S, Izmailov R, Dan R (2006) Performance optimization of voip using an overlay network. J Non Cryst Solids 144(2):247–252
27. Li J, Li J, Chen X, Jia C, Lou W (2015) Identity-based encryption with outsourced revocation in cloud computing. IEEE Trans Comput 64(2):425–437. https://doi.org/10.1109/TC.2013.208
28. Li J, Zhang Y, Chen X, Xiang Y (2017) Preserving privacy with probabilistic indistinguishability in weighted social networks. IEEE Trans Parallel Distrib Syst 28(5):1417–1429. https://doi.org/10.1109/TPDS.2016.2615020
29. Li J, Zhang Y, Chen X, Xiang Y, Li J, Zhang Y, Chen X, Xiang Y (2017) Secure attribute-based data sharing for resource-limited users in cloud computing. Comput Secur 72:1–12. https://doi.org/10.1016/2017.08.007
30. Li P, Li J, Huang Z, Gao CZ, Chen WB, Chen K (2017) Privacy-preserving outsourced classification in cloud computing. Clust Comput, 1–10. https://doi.org/10.1007/s10586-017-0849-9
31. Qi W, Ding W, Wang X, Jiang Y, Xu Y, Wang J, Lu K (2018) Construction and mitigation of user-behavior-based covert channels on smartphones. IEEE Trans Mob Comput 17(1):44–57. https://doi.org/10.1109/TMC.2017.2696945
32. Qiu L, Zhang Y, Wang F, Kyung M, Mahajan HR (1985) Trusted computer system evaluation criteria. In: National Computer security center
33. Rezaei F, Hempel M, Sharif H (2017) Towards a reliable detection of covert timing channels over real-time network traffic. IEEE Trans Depend Secur Comput 14(3):249–264. https://doi.org/10.1109/TDSC.2017.2656078
34. Schulzrinne H (1995) Internet services: from electronic mail to real-time multimedia. Springer, Berlin, pp 21–34
35. Schulzrinne H, Casner S, Frederick R, Jacobson V (1996) Rtp: a transport protocol for real-time applications. Ietf Rfc 2(2):459C482
36. Shen J, Gui Z, Ji S, Shen J, Tan H, Tang Y (2018) Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. Journal of Network and Computer Applications. https://doi.org/10.1016/j.jnca.2018.01.003

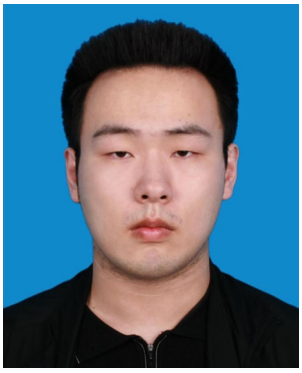
37. Sun L, Li Z, Yan Q, Srisa-An W, Pan Y (2017) Sigpid: significant permission identification for android malware detection. In: International Conference on malicious and unwanted software, pp 1–8. https://doi.org/10.1109/MALWARE.2016.7888730
38. Szpyrka M (2013) Fast and flexible modelling of real-time systems with rtcp-nets. Comput Sci 6(5):81
39. Wang YH, Chow TH (2016) Applying patent-based fuzzy quality function deployment to explore prospective volte technologies. Int J Fuzzy Syst 18(3):424–435
40. Wu Z, Xu Z, Wang H (2015) Whispers in the hyper-space: high-bandwidth and reliable covert channel attacks inside the cloud. IEEE/ACM Trans Netw 23(2):603–614. https://doi.org/10.1109/TNET.2014.2304439
41. Xue Y, Tan YA, Liang C, Zhang C, Zheng J (2018) An optimized data hiding scheme for deflate codes. Soft Comput 22(13):4445–4455. https://doi.org/10.1007/s00500-017-2651-2
42. Zhang X, Tan YA, Xue Y, Zhang Q, Li Y, Zhang C, Zheng J (2017) Cryptographic key protection against frost for mobile devices. Cluster Comput 20(3):2393–2402. https://doi.org/10.1007/s10586-016-0721-3
43. Zhang X, Tan YA, Zhang C, Xue Y, Li Y, Zheng J (2018) A code protection scheme by process memory relocation for android devices. Multimed Tools Appl 77(9):11137–11157. https://doi.org/10.1007/s11042-017-5363-9
44. Zhu H, Tan YA, Zhang X, Zhu L, Zhang C, Zheng J (2017) A round-optimal lattice-based blind signature scheme for cloud services. Futur Gener Comput Syst 73:106–114. https://doi.org/10.1016/2017.01.031
45. Zhu R, Zhang B, Mao J, Zhang Q, Tan YA (2017) A methodology for determining the image base of arm-based industrial control system firmware. Int J Crit Infrastruct Prot 16:26–35. https://doi.org/10.1016/2016.12.002
46. Zhu H, Tan YA, Yu X, Zhang XY, Zhu QL, Li Y (2018) An identity-based proxy signature on ntru lattice. Chinese J Electron 27(2):297–303(6). https://doi.org/10.1049/2017.09.008
47. Zkik K, Orhanou G, Hajji SE (2017) Secure mobile multi cloud architecture for authentication and data storage IGI global. https://doi.org/10.4018/IJCAC.2017040105



Quanxin Zhang, received the Ph.D. degree in School of Computer Science from Beijing Institute of Technology, in 2003. He is the faculty of the School of Computer Science and Technology at Beijing Institute of Technology. From 2011 to 2012, he was the visiting scholar at University of Connecticut. His research interest includes the ad hoc network and mobile computing.

Hanxiao Gong, M.S. degree candidate of the School of Computer Science and Technology at Beijing Institute of Technology. Her research interests include information security and mobile computing.

Xiaosong Zhang, received MS degree in computer technology from University of Science & Technology Beijing. Currently, he is a Ph.D. candidate of Beijing Institute of Technology. His main research interests include information security and mobile computing.



Chen Liang, Ph.D. candidate in Beijing Institute of Technology, his main research interests include information security, coding theory.

Yu-an Tan, professor and Ph.D. supervisor in Beijing Institute of Technology, senior member of China Computer Federation. His main research interests include network storage, storage security and embedded system.