Annales UMCS Informatica AI 5 (2006) 417-426

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

New security and control protocol for VoIP based on steganography and digital watermarking

Wojciech Mazurczyk[1*], Zbigniew Kotulski[1,2**]

[1]*Institute of Telecommunications, Faculty of Electronics and Information Technology, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warszawa, Poland*
[2]*Institute of Fundamental Technological Research, Polish Academy of Sciences, Świętokrzyska 21, 00-950 Warszawa, Poland*

Abstract

In this paper new, lightweight security and control protocol for Voice over Internet Protocol (VoIP) service is presented. It is the alternative for the IETF's (Internet Engineering Task Force) RTCP (Real-Time Control Protocol) for real-time application's traffic. Additionally this solution offers authentication and integrity of voice send and it is capable of exchanging and verifying QoS and security parameters. It is based on two information hiding techniques: digital watermarking and steganography. That is why it does not consume additional bandwidth and the data transmitted is inseparably bound to the voice content.

1. Introduction

Nowadays two most important fields which Voice over Internet Protocol (VoIP) lacks provide certain Quality of Service (QoS) parameters and security considerations [1]. In this paper we consider a new protocol that covers both those fields simultaneously. It provides information that is vital to control the network conditions and to verify authentication of the source and data integrity.

In TCP/IP networks VoIP, which is a real-time service, uses RTP (Real-Time Protocol) with UDP (User Datagram Protocol) for transport of digital streams. Currently there is one control protocol for RTP and it is RTCP (Real-Time Control Protocol) [2]. It is designed to monitor the Quality of Service (data delivery) and to convey information about the participants in an on-going session. RTCP operates mainly on exchanging two special reports called: Receiver Report (RR) and Sender Report (SR). Parameters that are enclosed in those reports can be used to estimate the network status. We propose a new protocol that uses two techniques: digital watermarking and network

[*]*E-mail address*: W.Mazurczyk@tele.pw.edu.pl
[**]*E-mail addresses*: Z.Kotulski@tele.pw.edu.pl zkotulsk@ippt.gov.pl

steganography to achieve analogous functionality like RTCP but moreover, offers additional advantages. The most important one is that it includes also security verification of the transmission source and the content sent (authentication and integrity). This solution does not consume transmission bandwidth, because the control bits (a header of the new protocol) are transmitted in a covert (steganographic) channel and data (QoS and the security parameters) is inseparably bound to voice content as a watermark.

The paper is organized as follows. In Section 2 both techniques, digital watermarking and steganography, are described. Next, we give details about the proposed solution in Section 3. Finally, we end with conclusions in Section 4.

2. Information Hiding techniques

Information Hiding has two subdisciplines and they are Steganography and Digital Watermarking. The general difference between those two techniques is that steganography's aim is to keep the existence of the information secret and in watermarking to render it imperceptible. In this section we will characterize shortly both those techniques briefly.

2.1. Steganography

Steganography is a process of hiding secret data inside other, normally transmitted data, so in ideal situation, anyone scanning data will fail to know it contains covert data. In modern digital steganography, data is inserted into redundant (provided but often unneeded) data, e.g. fields in communication protocols, graphic image, etc. For TCP/IP steganography (or network stenography) the most important fact is that a few fields in the packet's headers are changed during transit. So we can exploit for our solution a covert channel, which is a method of communication that is not a part of an actual computer system design, but can be used to transfer information to users or system processes that normally would not be allowed access to the information.

In TCP/IP stack, there is a number of methods available, whereby covert channels can be established and data can be exchanged secretly between hosts as stated in [3]. As we wrote earlier an analysis of the headers of typical TCP/IP protocols e.g. IP, UDP, TCP, HTTP, ICMP results in fields that are either unused or optional. This reveals many possibilities where data can be potentially stored and transmitted.

For VoIP and our solution we will exploit unused/optional fields in IP/UDP/RTP packets because those protocols are used in almost all IP telephony implementations. As described in [4] IP header alone posseses a few fields that are available to be used as a covert channel. The total capacity of those fields exceeds 60 bits per packet. And we can deploy also unused UDP and RTP protocols fields. In [4,5] and [6] different methods of hidden transmission are

presented. We do not limit this solution to using only IP/UDP/RTP protocol. Lower layers of TCP/IP stack also offer steganography possibilities like for example stated in [7]. Furthermore we can distribute those control bits among those fields in a predetermined fashion (this pattern can be exchanged during a signalling phase of conversation). In those chosen fields we will transmit only header (control bits) of our protocol with the use of steganography technique. Header consists of 6 bits per packet, so such a type of transmission is potentially hard to discover. The details will be described in Section 3.

2.2. Watermarking

Digital watermarking covers a large field of various aspects, from cryptography to signal processing and is generally used for marking the digital data (images, video, audio or text). There are several applications for digital watermarks, described in [8] and [9], that include:

- Fingerprinting (embedding a distinct watermark into every copy of the author's data),
- Annotation Watermark/Content labelling (embedding information, which describes the digital work that can be later extracted),
- Usage control/Copy control (authors can insert a watermark that indicates the number of copies permitted for each user),
- Authentication and integrity watermark is the most important application for our purposes.

The watermark that will be used in the proposed authentication and integrity solution must possess certain parameters, like: robustness, security, transparency, complexity, capacity, verification and invertibility. Those parameters are well described in [8] and [9]. Their optimization for real-time audio system is crucial. They are often mutually competitive, however there is always a compromise necessary. That is why the embedded watermark, that we will use, must be characterized by **high robustness**, **high security** and must be **nonperceptual**. Not every watermarking technique is applicable for our solution. IP Telephony is the demanding, real-time service. That is why we need the watermarking schemes that really work for the real-time conversations. Such algorithms are described, e.g. in [9,10] and [11].

Generally, audio watermarking algorithm is based on two functions: **embedding** of the watermark into voice and its **extraction**. As soon as the conversation begins, certain information is embedded into the voice samples and sent through the communication channel. Then, the watermark is extracted from those samples before they reach the callee and the information retrieved is verified. If the watermark's data sent is correct, the conversation can be continued.

Most digital watermarking algorithms for the real-time communication are designed to survive typical non-malicious operations like: low bit rate audio compression, codec changes, DA/AD conversion or packet loss. For example, in [9] the watermarking scheme developed at the Fraunhofer IPSI (Institut Integrierte Publikations und Informationssysteme) and the Fraunhofer IIS (Institut Integrierte Schaltungen) were tested for different compression methods. Those results revealed that the large simultaneous capacity and robustness depend on the scale of the codec compression. When the compression rate is high (1:53), the watermark is robust only when we embed about 1 bit/s. With a lower compression rate we can obtain about 30 bit/s, whereas the highest data rate was 48 bit/s with good robust, transparent and complexity parameters. For the monophonic audio signal, which is a default type for IP Telephony the watermark embedding algorithm appeared around 14 times faster and the watermark detector almost 6 times faster than the real-time.

The next important thing for this scheme is how much information we can embed into the original voice data. This will influence the speed of the authentication and integrity process throughout the conversation. This parameter, in our solution, is expected to be high but it is not crucial. With low compression rates, we propose to add a pre-conversation stage. In this stage there will be a few seconds of the RTP packets exchange without the conversation. It will delay the setup of the call but then, during the conversation, the time of verification will be shorter. However, the lowest payload watermarks (about 1 bit/s) cannot be accepted in our scheme because, in this case, the conversation would have to last enormously long to work correctly.

3. New scheme description

For IP Telephony the most important security services are: authentication, integrity and confidentiality. The first two can be provided with the use of our protocol. The third should be guaranteed in a different manner, e.g., with the use of the security mechanisms from a classical security model (the cryptographic mechanisms).

In [12] we proposed authentication scheme based only on digital watermarking, that improves security of both: signalling messages and voice sent in the communication channel. Here we will enhance our approach. As we described earlier here we will use two information hiding techniques: steganography to create covert channel that will be used to transmit header (control bits) and digital watermarking to bound the parameters of the protocol to voice sent into the network (watermark). For the clarity of description we assume that our solution will be used only in IP protocol version 4 networks [13]. But it can be also applicable with minor changes to other protocols as well (e.g. for IPv6).

3.1. Protocol data unit (PDU)

The PDU's (Protocol Data Unit) size that proposed protocol will use must be kept to minimum. It is important because as we said in Section 2 the capacity capability of watermarks is limited if we want to embed watermark that posseses other important parameters like robustness or security. Each PDU consists of header (control bits) and a certain number of data bits that are embedded into sender/receiver voice. Because the capacity of the watermark depends greatly on the codec's compression rate that is used, so it is possible that lot of parameters can be distributed into a number of packets. The size (number of bits) of each parameter that will be transmitted with our protocol should be low. For all parameters it should not exceed 32 bits. This value is taken from RTCP protocol size of the parameters. Only one parameter (NTP timestamp) is greater than the given value. Limited size of every parameter results in shorter time for the parameter to be transmitted and verified. However, we do not dictate this value. It should depend on network bandwidth, status and codec's compression rate. The PDU form is presented in Fig. 1.

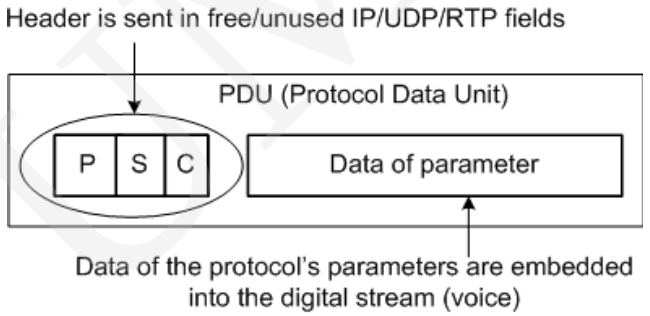

Fig. 1. General PDU characterization

As we see in Fig. 1 the PDU consists of two parts: the header (control bits) and the watermark data. The header/control fields are transmitted in a covert channel in unused/optional fields of IP/UDP/RTP protocol's headers. Actual value of the parameter is embedded into voice as a watermark. The header (control bits) used in PDU are organized in the fields as shown in Table 1.

Table 1. Header fields and their function

Type of field	Number of bits	Function
P (Parameter)	4	Describes parameter that is transmitted in the watermark
S (Side)	1	Describes the side of the communication (1 – sender, 0 – receiver report)
C (Continuity)	1	Describes if a packet contains the beginning or continuation of the parameter indicated in the field P (1 – beginning of new parameter, 0 – continuation of the last parameter)

Exemplary values of the field P are shown below (analogous parameters like in RTCP [2]):

0001 – authentication or integrity parameter (32 bits)
0010 – parameter: LSR – Last sender report (32 bits)
0011 – parameter: DLSR – Delay of last sender report (32 bits)
0100 – parameter: Interarrival jitter (32 bits)
0101 – parameter: Extend the highest sequence number received (32 bits)
0111 – parameter: Cumulative number of packet lost (24 bits)
1000 – parameter: Fraction lost (8 bits)
1001 – parameter: Sender's packet count (32 bits)
1011 – parameter: NTP timestamp (64 bits)
1010 – parameter: RTP timestamp (32 bits)
…

Moreover, the PDU can have one of two payload types: security or informational. Security payload means that PDU contains certain authentication and/or integrity information that should be verified after its extraction. Two kinds of security payloads are available, the first is used to provide authentication and integrity of the voice and its source. The role of the second is to authenticate protocol parameters that were sent earlier (both security and informational). Another payload type is informational. Each PDU carries one of the parameters that are used to monitor the quality of service and the network conditions.

The classification of PDU's available payloads is presented in the figure below.

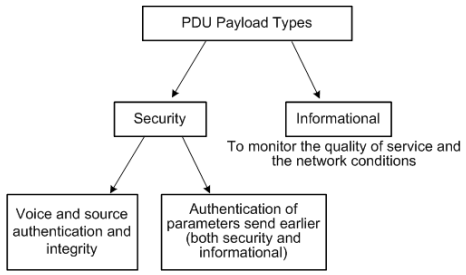

Fig. 2. Available PDU payload types

Usually in one IP/UDP/RTP packet there are about 20-30 milliseconds of voice, which is about 20-30 bytes, depending on type of codec used. Let us say that we are able to embed at the average about 10 bits/s of watermark into the voice stream. With that assumption we must send about 3-4 packets to achieve those 10 bits. In this protocol we set parameter's value to 32 bits, so this parameter will be transmitted in about 9-12 packets in more than 3 seconds of the voice. In the example scenario in Fig. 3, we see how the exemplary

parameter: Interarrival jitter (32 bits) is transmitted for the assumption: 10bits/packet.

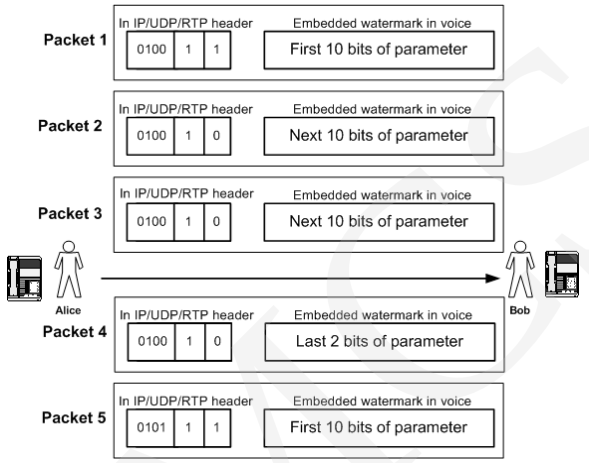

Fig. 3. Example of transmission of the Interarrival jitter parameter

As we can see in Fig. 3 the parameter characterized by code 0100 (Interarrival jitter) was sent in four IP/UDP/RTP packets. In the first packet both fields S and C were set to 1. In the next packet field C changed its value to 0 because it is a continuation of the parameter's data that was sent in the last packet. At the destination there must be a buffer to extract all data from each packet. After transmitting all packets for one parameter data is available to be used (for QoS monitoring) or to be verified (for security reasons).

3.2. Authentication and integrity parameter calculation and security payload

Authentication and integrity calculation will be performed similarly, but simplified as described in [14] and with watermark specific considerations. Generally we take from Miner's and Staddon's solution the creation of security relations between blocks of data that are sent. It means that we additionally authenticate the parameters that are used for authentication.

In Section 3.1 we mentioned that two security payloads are available:

- one is used to provide authentication and integrity of the voice and its source,
- the other is to authenticate protocol parameters (both security and informational) that were sent earlier.

he first security parameter is a combination of user global identification and features that were extracted from the voice stream. It is expected that this parameter will have 32 bits. So if the concatenation of those two values exceeds this number of bits, there will be a hash function (marked as H) performed. Then only predetermined bits will be transmitted as a security parameter.

The second security payload is a special parameter that will be used to provide greater security of the whole digital stream and transmission. The general idea of its calculation is presented in Figure 4.

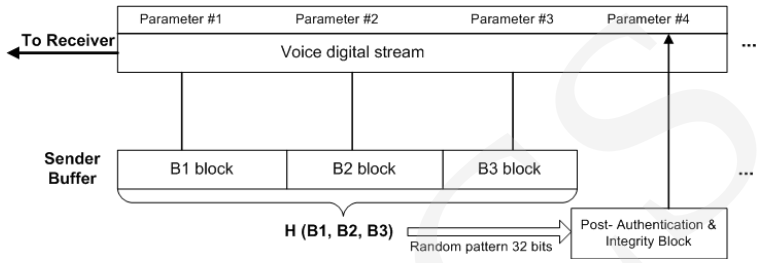

Fig. 4. Example of authentication and integrity mechanism for transmitted parameters

First, we must emphasize that during a conversation (RTP packets flow) there will be constant two-way exchanging of certain sequence of parameters. Those parameters can be susceptible to e.g. modifications or other attacks. To prevent this situation every n-th packet is used to authenticate and provide integrity of n-1 parameters that were transmitted earlier.

For the situation in Fig. 4 $n = 4$. Three parameters that can contain informational or first kind of security payload are stored in the sender buffer. After they all are placed there (B1, B2 and B3 blocks) the hash function (H) can be calculated, if the result value is too long. Because we assumed certain parameter length that is why we have to choose only 32 bits from the hash to be transmitted. For every conversation this pattern, in which bits are chosen, should be changed and its determination should be set and sent in a signalling phase of the connection.

3.3. Level of Trust (LoT) mechanism

However, we can still imagine a situation, in which the attacker disrupts transmission of the header/controls bits. In this situation the receiver is unable to retrieve any parameters that were transmitted by the sender. Here we will describe a mechanism that will prevent such a situation. This mechanism was introduced in [12]. The general idea is for both parties to update special parameter named LoT (Level of Trust), during a conversation. If a parameter (security or informational) is received and verified, LoT value increases. In any other situation its value decreases. Additionally, the parameters that are exchanged during conversation influence LoT value differently. Informational parameters (QoS) add/subtract to LoT's value 1, the first kind security parameters 2 and the second kind security parameter 5. If A sends a parameter to B, the algorithm of handling the LoT parameter (on B side) works, as described below in the pseudo-code:

```
START /* CL - Critical Level, LoT - Level of Trust, T - timer */
CL = a; LoTA = x; TA = 0; /* Initiating values */
StartTimer(TA);
FOR (i = 0; i++; i< End of Transmission) /* i - Time slot */
   {
   IF (ParameterA correct) THEN
      {
      LoTA + {1 or 2 or 5}; *
      ResetTimer(TA);
      }
   ELSE (LoTA – {1 or 2 or 5}); *
   IF (LoTA <= CL) OR (TA > k) THEN STOP; (1)
   IF (LoTA = a*x) THEN LoT = x; (2)
}
* value depends on the type of parameter (QoS, security)
```

As we can see, the breakage of the call (or notification to the calling parties) will take place if the value of the LoT parameter is equal to or below the given threshold (CL value) or if the timer TA expires (1). The LoT value changes during the conversation time. If every signalling message is successfully verified, the LoT value rises. To prevent its increase from reaching the infinity, we lower it, as soon as it reaches the value of the critical level multiplied by the start value of LoT (2).

This way of decreasing the LoT value has one serious disadvantage: it allows an attacker to wait until LoT = (a*x)-1. But we must assume that he is able to posses information about its value and then safely spoof ((a*x) - 1 – (CL + 1)) audio packets without LoT's falling below the threshold (CL). To prevent it, one must choose the initiating values (a and x) carefully. Their values should depend on network's parameters: the packet loss and possible delays. If the network does not suffer heavily from the packet loss, those values must be low. In the other case, they must be set to a higher level. For example, the network administrator or service provider can circumscribe those parameters for a certain network/user.

Summary and conclusions

New security and control protocol for VoIP service was presented. It uses two information hiding techniques: steganography to create a covert channel in which the header (control bits) is passed and digital watermarking to transmit the actual data (parameter's value) in the voice stream. The most important advantages of our solution are no consuming of available bandwidth, providing security, parameters to monitor QoS and network status in one protocol. What we want to emphasize is that the process of sending information for this protocol

is continuous in time and although the bit rate per second offered by watermarking is usually not very high, when we consider the whole conversation we see that we are able to exchange quite a large amount of data.

The variety of different kind of parameters that can be used in our solution is not limited to security/monitoring status of the network ones. That is why this protocol can be freely extended to other data e.g. to support other and more detailed statistics as described in [11].

References

[1] Kuhn D.R, Walsh T.J., Fries S., *Security Considerations for Voice Over IP Systems*, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, (2004).

[2] Schulzrinne H., Casner S., Frederick R., Jacobson V., *RTP: A Transport Protocol for Real-Time Applications*, IETF, RFC 3550, (2003).

[3] Llamas D., Allison C., Miller A., *Covert Channels in Internet Protocols: A Survey*, In Proceedings of the 6th Annual Postgraduate Symposium about the Convergence of Telecommunications, Networking and Broadcasting, PGNET 2005, (2005).

[4] Murdoch S.J., Lewis S., *Embedding Covert Channels into TCP/IP*, Information Hiding, (2005) 247.

[5] Ahsan K., Kundur D., *Practical Data Hiding in TCP/IP*, In: Proceedings of Workshop on Multimedia Security at ACM Multimedia '02, Juan-les-Pins (on the French Riviera), (2002).

[6] Anderson R., (Ed.): Proceedings of: Information Hiding .First International Workshop, Cambridge, U.K., May 30, June 1, 1996, Springer-Verlag Inc., 1174 (1996).

[7] Szczypiorski K., *HICCUPS: Hidden Communication System for Corrupted Networks*, In Proc. of: The Tenth International Multi-Conference on Advanced Computer Systems ACS'2003, October 22-24, 2003 Miedzyzdroje, Poland, ISBN 83-87362-61-1, (2003) 31.

[8] Dittmann J., Mukherjee A., Steinebach M., *Media-independent Watermarking Classification and the need for combining digital video and audio watermarking for media authentication*, Proceedings of the International Conference on Information Technology: Coding and Computing, IEEE Computer Science Society, Las Vegas, Nevada, USA, (2000) 62.

[9] Steinebach M., Siebenhaar F., Neubauer C., Ackermann R., Roedig U., Dittmann J., *Intrusion Detection Systems for IP Telephony Networks*, Real time intrusion detection symposium, Estoril, Portugal, 17 (2002) 1.

[10] Mizrahi T., Borenstein E., Leifman G., Cassuto Y., Lustig M., Mizrachi S., Peleg N., *Real-Time Implementation for Digital Watermarking in Audio Signals Using Perceptual Masking*, 3rd European DSP Education and Research Conference, ESIEE, Noisy Le Grand, Paris, (2000).

[11] Yuan S., Huss S., *Audio Watermarking Algorithm for Real-time Speech Integrity and Authentication*, International Multimedia Conference Proceedings of the 2004 Multimedia and security workshop on Multimedia and security, Magdeburg, Germany, (2004) 220.

[12] Mazurczyk W., Kotulski Z., *New VoIP traffic security scheme with digital watermarking*, In Proceedings of SafeComp 2006, Lecture Notes in Computer Science 4166, Springer-Verlag, Heidelberg, (2006) 170.

[13] Information Sciences Institute University of Southern California : IP (Internet Protocol), IETF, RFC 791, September (1981).

[14] Miner S., Staddon J., *Graph-based authentication of digital streams*, In Proceedings of the IEEE Symposium on Research in Security and Privacy, (2001) 232.

[15] Friedman T., Caceres R., Clark A., *RTP Control Protocol Extended Reports (RTCP XR)*, IETF, RFC 3611, (2003).