VERSITA

Annales UMCS Informatica AI XI, 3 (2011) 71–86
DOI: 10.2478/v10065-011-0017-4

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

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

Non–cryptographic methods for improving real time transmission security and integrity

Grzegorz Oryńczak[1]*, Zbigniew Kotulski[2,3]†

[1] *Jagiellonian University, Department of Physics, Astronomy and Applied Computer Science, Cracow, Poland*
[2] *Institute of Telecomunications, Faculty of Electronics and Information Technology, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland*
[3] *Institute of Fundamental Technological Research of the Polish Academy of Sciences, Warsaw, Poland*

Abstract

In this paper we present a few non cryptographic methods for improving the security, integrity and reliability of real time services. The methods presented in this paper apply to real time transmitting systems, which are based on the Peer-to-Peer (P2P) model. A basic idea of the first technique is to use agents for detecting steganographic content in packet headers, so packets with suspicious entries in the IP header fields will be blocked or the fields will be erased. The two other presented techniques are based on reputation and trust systems, so trust and reputation basic definitions, types and modelling methods are shown. Also a simple design scheme of using these mechanisms in a P2P real-time data transmitting infrastructure is presented. Additionally, we describe an idea of path selecting technique, which can be used to avoid paths that are susceptible to eavesdropping.

**E-mail address: grzegorz.orynczak@uj.edu.pl*
†E-mail address: zkotulsk@tele.pw.edu.pl

1. Introduction

Internet services that are based on real time data transmission have gained much popularity in recent years. Advantages of the most common ones like internet telephony (VoIP), videoconferencing, live media streaming and online multiplayer games are seen not only by ordinary people but also by big companies (where internet conference systems are very common). Rapid growth of popularity and commercial success of those services influenced not only on increased interest in finding new, more efficient ways for real time data transmission, but also aspects of providing appropriate security mechanisms (for data transfer and signalization) started to be much more important (in most of the currently popular implementations this problem is often overlooked or only partially solved).

In this paper we present a few methods for improving the security, integrity and reliability of real time services. We will focus only on non cryptographic methods, as opposite to standard, cryptographic ones like TLS connections, authenticity certificates and hash functions for packet integrity. The techniques presented in this paper apply to real time transmitting systems, which are based on the Peer-to-Peer (P2P) model and where data between users is routed over an overlay network. This model, due to its numerous advantages, is often used by popular services, e.g. Skype. Our reference system, described in [**1**], also uses agents (each agent acts within the single node). The main task of an agent is to constantly control the quality of transmission between peers and to dynamically build a routing table. Here we describe also other tasks that can be preformed by agents in order to improve transmission security. The basic idea of the first technique is to use agents for detecting steganographic content in packet headers, so packets with suspicious entries in the IP header fields will be blocked or the fields will be erased. Another presented technique involves reputation system. In this system peers can rate each other based on their behaviours (transmission quality, failure rate etc.). Aggregated ratings about a given peer are used to determine its reputation score, which is taken into account in building safer and more reliable routing tables. Additionally, we describe a trust system, that by analyzing and combining connections and peers trust relationships is used to measure trustworthiness of specific path, so the system can decide whether it is safe to use it for data transmission. Besides, because nodes participate in data transmission, we have the control over the data flow route. It is possible to specify the desired route, so, regardless of the external routing protocol, we can avoid paths that are susceptible to eavesdropping.

2. Steganography

Steganography is a field of science concerning hiding messages in such a way that only sender and recipient are aware of the existence of this message. The biggest advance of steganography over typical cryptographic methods is that message itself does not attract attention. Normal (plainly visible) messages, no matter how strongly encrypted, will always arouse suspicion. Thanks to the use of steganography we can protect not only the content of the message but also parties involved in communication. However steganography and cryptography may be good partners and it is a common practice to use them both for sending both hidden and encrypted messages.

In this section we will show how to protect the infrastructure for transferring real-time data from using it for sending hidden messages. The most common steganographic methods that can be used for creating covert channels in real-time multimedia stream will be described and the method of detecting and blocking this kind of hidden transmission will be proposed. For hiding one piece of data within another we need the following elements:

1. covert media when our hidden message will be held
2. message (m) that we want to hide
3. data encoding function F and its inverse F^{-1}
4. optional key schema (k)

Although the usage of the key is optional, they are recommended (otherwise the system does not obey the Kerckhoffs principle [**2**]). The schematic diagram of steganographic operation is shown in Fig. 1.

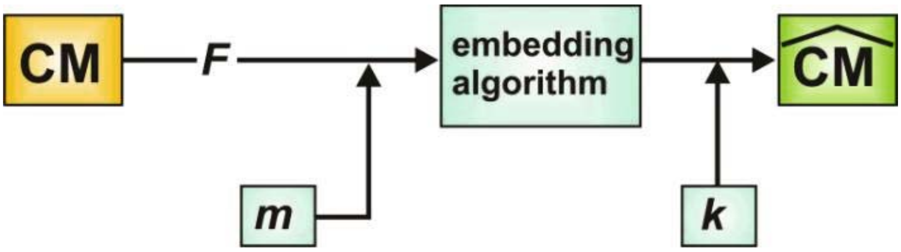


Fig. 1. Diagram of steganographic operation.

Owing to their numerous advantages, steganographic technologies have gained much popularity over the last few years. Much work has been done in the field of finding new covert channels in different media types. Also with the increased interest in steganography, methods of detecting hidden content (steganalysis) and blocking covert channels are intensively studied. There are many different digital media that can be used for embedding messages. Also different embedding algorithms, depending on the medium used, can be applied. The most common covert media types and algorithms are listed below:

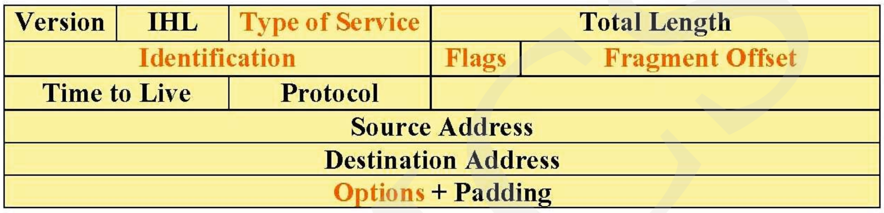
1. **Plain text**
    - using certain characters of covert text - additional key (series of integer numbers) is needed to read a hidden message
    - white space manipulating technique - adding certain amount of white spaces between words or at the end of lines
    - lines shifting technique - shifting certain lines in the document by a small fraction
    - inserting different tags in the XML code
2. **Images**
    - last significant bit (LSB) manipulation
    - simple watermarking (adding pattern on the top of another image)
    - LBS in Direct Cosine Transformation (DCT) and Wavelet Transformation
3. **Audio and video files and streams**
    - simple LSB modifications after analog audio quantization
    - echo hiding - embedding secret messages in covert audio/video as an echo
    - phase coding - using phase shifts in the phase spectrum
    - spread spectrum techniques [**3**]
4. **Network covert channels**
    - using free/unused protocols fields
    - timing based methods
    - many others

More information about general steganography techniques can be found for example in [**4**]. Because in this section we want to discuss the possibility of using real-time infrastructure for sending hidden messages, we will focus on network covert channels. Typically, most popular network channels can be:

(1) **Based on storage** - when information is conveyed by writing or abstaining from writing on certain areas of cover media stream.
(2) **Timing based** - information is conveyed by triggering or delaying certain events at specific time intervals. Receiver needs a timer to decode messages.
(3) **Frequency based** - information is encoded using many channels of cover traffic.

Combination of those channels is a key to access hidden information. It is also possible to use any combination of the methods mentioned above.

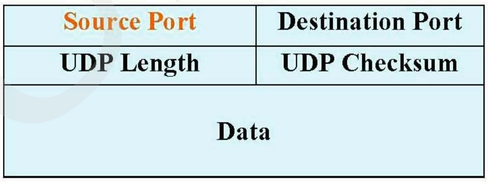
For the standard network communication we can use the TCP or UDP protocol. However, because of the low latency requirements of real time traffic, due to its speed and simplicity, UDP is a preferable choice to base on while building real-time protocols. So, in the case of storage based steganography, some fields in IP and UDP frame headers can be exploited. A typical IP header is shown in Fig. 1. Fields selected in red can be used for inserting steganography data.



Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol		
Source Address				
Destination Address				
Options + Padding				

Fig. 2. IP header with marked fields that can be used for inserting steganography data.

As it was said before, steganography in the UDP header is also possible, but due to the small and very simple header, we do not have such a wide selection of fields that can be used as a carrier for a steganographic covert channel. A typical UDP field in shown in Fig. 3.



Source Port	Destination Port
UDP Length	UDP Checksum
Data	

Fig. 3. UDP header with the marked field that can be used for inserting steganography data.

The header is 64 bits long and has only 4 fields of bytes each. For correct packet delivery, destination port needs to be unmodified, so it cannot be used for steganography. Also length of the field cannot be used in any circumstances because it represents the length of entire packet and if modified, data contained in this packet will not be correctly read. On the other hand, third field - checksum is not necessarily needed for correct communication because UDP specification does not require checksum calculation. However, if checksum is not calculated, according to RFC, it should be set to 0, otherwise the receiver will assume that it was calculated, and if it is not correct, packet will be dropped. Finally it leaves us only with one field: source ports, and because UDP is a connectionless protocol, it does not need to send any replies, so only this field can be successfully used for steganography.

Also other protocol headers, for example those dedicated to real-time transmission like RTP/RTCP, can be used for embedding hidden messages. The main problem with steganography in this form of communications is associated with the usage of non reliable communication (packets can be lost, and they will not be retransmitted). However, as it was shown in many papers (e.g. [**5**]) steganography is possible.

Moreover, real time transmission, because of its nature (continuously sending many relatively short packets) is a good environment for implementing timing and frequency based steganography. The basic idea of timing based covert channels is shown in Fig. 4.

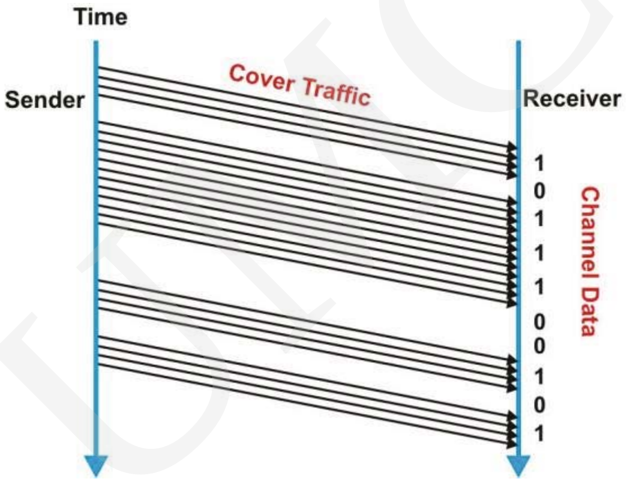


Fig. 4. Idea of timing based covert channels.

In this case a hidden message is encoded by intentional delays of the time of sending certain packets. Unfortunately, due to low bandwidth of this way constructed covert channel, this method is suitable primarily for sending short text messages. Additionally, in order not to impair the quality of real time transmission by introducing additional delays, it should be used only with low latency links or with pre buffered streams (i.e. Video on Demand services).

2.1. Detecting and blocking methods

We propose two approaches for eliminating possibility of steganography communication in infrastructure for real-time data transmission. The first one is based on the statistical analysis of packet header fields for storage based covert channels and packet send time analysis for time based steganography. Based on the tests that we have performed on our reference system, we chose to use the same agents that are involved in P2P routing to additionally analyze routed data for a hidden content. Thanks to the use of agents, the detection process is faster, more efficient, easier to integrate with other mechanisms (as described in the next section reputation mechanism) and more flexible (agents can inform each other about new detecting criteria). Agents in the nodes to which the end users are connected, before routing real-time traffic to the next peer, search for anomalies in the routed real-time data. Traffic becomes suspected if excessive fragmentation occurs, unusual IP flags are set, type of service field in the IP header is constantly changing, excessive packet re-ordering is detected or other predefined events are triggered. On the other hand, for detecting covered messages conveyed by packets send time manipulations, agents measure packet delay variability. If that variability has an unusual distribution (compared to the network model or to other routed traffic) traffic can be permanently blocked or packets send time can be additionally disturbed to make encoding a hidden communication impossible. For our tests, we created a disturbing mechanism that works by delaying transmission of a randomly chosen packet by a small amount of time. This delay time is calculated during delay variability analysis and it should be equal to ”binary zero” in the encoded covert message. In the case of steganography based on packet re-ordering, this delay time should be correspondingly longer. Additionally, for a detecting process agents may benefit from the additional knowledge obtained from other agents. This knowledge can be about new threat models, new possible covert channels or packet delay variability in the routed traffic (to improve detection of real-time traffic with unusual delay distribution).

The second method of eliminating steganographic communication is a simple preventive approach that can be used for example when P2P nodes have limited CPU and memory resources, so additional tasks may be undesirable. In this case agents do not analyze real time traffic in terms of finding hidden messages. Instead, they erase (set to standard values) every header field, that can be potentially used for creating a covert channel. Although this method is fast and simple to implement, it works only with storage based covert channels (using a packet delay disturbing mechanism for every routed traffic is very inefficient).

3. Trust

Apart from the posibility discussed in the previous sections of using real-time stream for sending hidden messages, there are many other security threats related to the real-time data transmission. At the same time, with the increasing popularity of services like internet telephony or videoconferences and the use of these services also for sensitive content delivery (i.e. company secrets during the videoconference meeting), the aspect of providing appropriate security level becomes even more important. However, in many commonly used real-time data transmission systems this aspect is still in its infancy. Furthermore, even if implemented, standard security techniques are used for example in the currently most popular internet telephony systems like SIP or H.323 are based only on cryptographic mechanisms. Therefore, encrypted connections based, for example, on TLS can be used for transmitting signalling messages to control connection state; also real-time traffic can be encrypted (for example with AES in the integer counter mode). However, those security technologies can provide protection against some, but not all security threats. Let us consider Peer-to-Peer reference system described in section one. By analyzing possible security risks it can be easily noticed, that if any of the peers starts to act deceitfully by providing false or misleading information, the traditional security mechanisms will be unable to protect against this type of threat. In the case of such an event, we will face a serious threat, against which there will not be any effective protection.

In this section we describe a trust mechanism that can be used as a complement to the standard security mechanisms, and further protect the system against the threats described above.

3.1. Trust definition

Trust can be defined as direct relationship between the two parties, where one is a trustor (trusting part) and another is trustee. The trustee can be for example an application, a server, some element of a distributed system, a peer in the P2P network or simply a person. The trustor can be defined similarly, but additionally, it should be able to make assumptions about the trustee, based on previous experience or other information that it possesses. The existence of this trust relation can influence on trustor's actions (i.e. it may prefer to cooperate with trustees). Although, this relationship is not symmetrical, it may affect trustee's opinion about the trustor and lead to reciprocal trust. The trust relationship is normally limited only to a specific range of actions or states, such as trust in the authenticity of something, trust in reliability of the received data, in security of the communication channel, etc.; therefore we usually define the scope of trust. In addition, the meaning of the term ”trust” can be interpreted in many ways. However, for the purpose of using this term in information theory, we will confine ourselves to only two most popular interpretations:

1. **(Reliability) Trust** - is understood as a perceived reliability of the trustee. In 1988 Gambetta [**6**] defined this form of trust as: *”subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends”*.

2. **(Decision) Trust** - in Ref. [**7**] is defined as the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible.

One of the main differences between these two definitions, is caused by the fact that decision trust can be changed depending on the given situation, while reliability trust is always constant. For example, let us assume that we have a working web server, and because it is relatively old, the probability of failure is high, so we decide not to use it any more and buy a new one. However, after a short time, a new server has crashed, so to avoid unavailability of a service, we decide to run the old one once again. In the example given above, reliability trust in the old server is always constant (and low), while decision trust changed as a result of a situation in which we find ourselves (we trust that the old server will help us avoid unavailability of a service, although we are aware of its high failure rate). In general, reliability trust reflects the degree of reliability of something, while decision trust is expressed as a binary decision. Most IT applications using trust models are based on the reliability trust definition, however, decision trust also can be used, for example in decision making systems.

3.2. Trust transitivity

Trust transitivity is one of the main features of the trust system, particularly important in large distributed systems. Its meaning can be explained by a simple example: if Alicia trusts Bob, and Bob trusts Carol, then Alicia will also trust Carol. As can easily be noticed, this transitivity takes place only in the case of the same trust scope. Thus, if Alicia trusts in Bob's authenticity and Bob trusts that Carol provides reliable information, this does not imply, that Alicia can trust in Carol's authenticity. In most cases, the concept of trust transitivity is based on recommendations, so, for example, when a trusted node in P2P network recommends another node, this recommended node also deserves some trust. For managing trust, analyzing transitivity, and determining trusted paths, trust matrices [**8**] and graphs can be build [**9**].

Depending on the adopted trust model, we can also assume the existence of negative trust (distrust). Similarly, to the previous case, this negative trust by transitivity, may influence the opinions of other parties, inducing them to reduce derived trust for that untreated peer. However, depending on the used model and formalism, in some situations this negative trust accordance with the principle that the enemy of my enemy is my friend, can positively influence on the opinion of some parties on this specific peer. In particular, if peer A does not trust peer B and peer B does not trust peer C, based on that information, A may decide to begin to trust C. But it turns out, that trust managing systems that can take into account these dependencies become very complex, therefore they are still not popular.

3.3. Trust modeling

Depending on the purpose for which we are building a trust system, we can use many different trust models. Currently, popular systems often consider trust as a term in sociology (e.g. [**10**]), where trust is understood as a time-varying positive attitude of one party toward the other. This model can be successfully used in multiagent systems, where similar to human society, agent's social nature and beliefs result in how they interact with the environment. Therefore, trust depends on previous experiences, its value can be increased as a result of successful experiences and decreased as a result of failures, so it continually evolves over time.

The other interesting model is proposed in Ref. [**9**], where trust system is used in Web services to automatically select service that best meets user's requirements. In this case, user quality of service (QoS) preferences is used to determine trust levels. However, this model should be used to improve service reliability rather than security.

There are also models that can take possible risk into account, so they can be used for building decision systems. The model combining risk and trust was presented by Manchala [**10**] in which he expresses trust by the values like cost of transaction and result of previous operations. Then risk-trust decision matrices are built and by applying fuzzy-logic rules it is possible to determine whether it is safe to transact with a particular party in the future. Instead of analyzing the number of positive experiences, also more complex economic model which takes into account the values like probabilities of success, consequences of failure, gain, asset values and more can be considered. This kind of trust system, based on economic modelling, was presented e.g by Josang and Lo Presti in Ref. [**11**].

The trust can be also interpreted as a measure of reliability. In this case we can use many types of metrics for determining trust or trustworthiness levels. Two most popular models are listed below:

1. **Probabilistic models** are very popular because of a very wide range of probabilistic methods that can be applied, such as simple probabilistic calculations, Bayesian methods and other statistical approaches. This model is used, for example, in the popular Google's PageRank link analyzing algorithm [**12**] in the Google Internet search engine. More information about probabilistic models can be found in Ref. [**13**].
2. **Belief model** is in many aspects similar to the probabilistic, but it can also deal with existence of some uncertainty. This uncertainty may one occur when the sum of probabilities of all possible outcomes differs from 1. The

most popular belief models are using a triple: belief, disbelief and uncertainty for determining the confidence values. The belief models are described for example in Ref. [**7**].

Another, quite popular model is based on discrete trust in which trust is expressed by discrete values or statements. Although, due to the use of discrete values, it may be difficult to apply standard computational algorithms to them, however the methods like lookup tables can be successfully used for analyzing trust transitivity paths. The examples of discrete trust models are shown in Ref. [**14**].

3.4. Usage of trust system for improving real time systems security

As it was mentioned at the beginning of this section, trust systems can be used as a complement to the standard, based on cryptographic techniques, security mechanisms. A properly designed trust managing system will have a significant impact on improving overall security of the system under consideration. Especially, in the case of P2P based applications for real time data transmission, applying trust as a component of security mechanism brings many benefits.

Because of the nature of those P2P systems, in a normal operation mode, many peers participate in each data transmission session. Additionally, flexible infrastructure architecture allows peers to leave and, what is more important from the security point of view, to join the infrastructure at any time. Therefore, those system properties make it vulnerable to peers that act deceitfully. However, by using the trust system and analyzing trust transitivity, it is possible to determine trusted nodes and paths, that will be used for transmitting real-time data which require high level of security. In this case, we propose to use agents (acting in each node), to manage trust relationships between nodes. Initially, every agent, based on its own knowledge, will determine which nodes are trusted for it. Then it will share information about its trust relationship with the main server (and depending on the assumed model, also with its closest neighbours). In the next step, the main server will use the data obtained from agents to build trust matrices [**8**], it will also allow other agents to download information about trust dependences between nodes in the infrastructure. Finally, each agent, based on its own knowledge and additional, downloaded from the server, data about other trust relationships in the infrastructure, will determine its own trusted paths by applying the trust transitivity principle. In the normal system operation we will consider two types of real time traffic. The first is a typical traffic which does not require high security levels (such as public internet radio or TV streams). For that any node which is currently available can be used for data routing, routing protocol will be focused on finding best paths in the overlay network (lowest latency, minimal packet losses etc.) and thus offer the best possible connection quality. In the case of the second type of transmission - where security of transmitted data is very important - agent that decides about the next routing step, will reduce the number of considered peers only to those which are trusted (determined as a result of trust transitivity) and a routing protocol will operate only on those peers. Although the presented method is only a simple schema and needs some improvements, it has shown how the consideration of the trust relationship in the routing process increases transmission security.

4. Reputation

Reputation is another useful tool, that can provide some benefit when integrated in the real-time transmission systems. In this section we will show how reputation mechanism can affect overall system stability, integrity and reliability.

Reputation is often defined as the ”overall quality or character as seen or judged by people in general” [**15**]. In many aspects reputation, sometimes introduced as another dimension of trust, is similar to previously described trustworthiness. However, there are some important differences between these two terms. As we mentioned before, trust can be based on personal experiences and subjective opinions. On the other hand, reputation represents the collective opinion of the community members about the reliability of a specific unit. To obtain a reputation score, aggregated subjective opinions (referrals or ratings from members) are processed by a reputation system. So, as opposite do trust, scores reflect community's opinion in general. In most cases this score affects our trust decisions, we are more likely to trust to a certain service because of its good opinions from other community members. Although, we can also choose to trust to another service, which has bad reputation, but our experiences with this service are positive. Those differences make reputation a more suitable decision support tool for the parties with none or limited knowledge about other infrastructure members. Therefore, for example, integrating a rating mechanism into distributed services, which are based on P2P architecture, helps choose those peers that are more reliable, thus improving transmission quality and overall service stability and efficiency.

From the standpoint of construction, we can distinguish two types of reputation systems:

1. **Centralized reputation systems** - which is characterized by the existence of centralized controlling unit that collects reputation ratings from other members and derives computed reputation scores to all interested parties. It is important that rating can be collected only from the members

that had direct experience with the rated party. During the transmisson agents observe each other's behaviour, and thereafter send rating to a collecion center, where, based on the obtained data, reputation scores are being updated. The centralized reputation systems, mainly due to their relatively simple implementation, are very often used. One of the most well known services, which uses this mechanism, is the eBay.

2. **Distributed reputation systems** - in this kind of system there is no central unit where all ratings can be submitted. Instead, many local units collecting ratings can exist or simply parties can record their opinion about past experiences with other parties. Parties are also equipped with mechanism for sharing their collected ratings among themselves, so a reputation score can be calculated by obtaining as many as possible ratings directly form other community members. Distributed reputation systems are ideal for implementation in the services based on the P2P model, so they can be found in the services like Napster, KaZaA, or Gnutella. It is also used in Mobile Ah Hoc Networks for determining the truthfulness of peers.

Due to the considerable interest in reputation systems in recent years, many different models have been presented. Their thorough comparison can be found in Ref. [**16**]. In this paragraph we shortly describe only a few most popular ones that can be used in the real-time data transmitting systems.

1. **Simple summation reputation system** - those are the simplest possible systems, where a reputation score is a difference in the number of positive and negative ratings. Another version of this system, also very simple to implement, can calculate reputation as an average of all collected ratings.
2. **Reputation systems based on the probabilistic approach** - two main types of those systems can be found in the literature: the Bayesian network based systems and the Subjective Logic based system. The first uses statistical updating of probability density functions. In this type of systems rating data is collected in a central unit and scores are calculated accordingly to the obtained ratings, so they are considered to be objective. On the other hand, the Subjective Logic based systems provide local and subjective reputation, so they are a good choice for the distributed systems. More information about those systems can be found in Ref. [**17**].
3. **Fuzzy logic base reputation systems** - fuzzy logic systems are often used in the situation where we are unable to collect required amount of statistical data to use the probability based mechanism (for example when trying to classify rare events). Models of fuzzy logic system are described in Ref. [**18**].

4.1. Reputation in real time data transmission

For the testing purpose we designed and implemented a simple reputation mechanism that was then built into our VoIP testing platform [**1**]. Just as in the case of our trust mechanism, agents acting in each node turned out to be ideal for managing operations related to reputation. Firstly, agents are used to observe behaviour of nodes which they communicate with (direct neighbours on the routing path), they analyze parameters like reliability and node long time behaviour. As our VoIP application has already had a centralized Login Server designed mostly for the purpose of users and nodes authentication and authorization, we choose to build a centralized reputation system with Login Server as a reputation center. Each agent sends its ratings about other peers to this Login Server, where they are processed and reputation scores of each node are updated (at this stage, we use a simple average of the ratings). Agents download those scores from the server, and next they are used for assisting routing decision: as an input to a routing algorithm, a graph created by assigning to each edge its cost index, which is calculated by multiplying the corresponding link quality index by its reputation with the appropriate weight factor, is given (more about routing can be read in [**1**]). Although the described reputation is very simple, it has a significant impact on the stability and reliability of transmission, since more reliable nodes are preferred. However, for normally operating systems more advanced reputation mechanism should be used (which for example can detect and ignore wrong or deceitful ratings).

5. Paths selecting technique

Another designed by us mechanism for improving real-time transmission security is a paths selecting technique. Although it is still in the early testing stage, first tests in a simulated environment has given promising results. The idea itself is rather simple. We use the fact that P2P based infrastructures, especially those popular, often have a large amount of available nodes (usually in the case where service users can also act as nodes -as it is in Skype), additionally, those nodes are often scattered over different geographical locations. Moreover, because nodes participate in data transmission, we have bigger control over the data flow route and can become, to some extent, independent of external routing protocols (like BGP). Therefore if a higher security level is required, it is possible to specify the desired route, so regardless of the external routing protocol we can avoid paths that are vulnerable to eavesdropping. For this purpose, a special path-reservation protocol was designed. The user, who wants to avoid unsafe paths/geographic locations, with help of a patch selecting mechanism chooses an alternative route, based on location of other infrastructure nodes. Then a special protocol informs every node on the path, for this particular traffic, not to use the typical routing mechanism, but to choose the predefined one. However, this mechanism, to work properly, needs an adequate number of nodes, full knowledge about their geographic locations, and routes between them provided by the external routing protocols.

6. Conclusions

This paper deals with non cryptographic methods that can be used for improving security, reliability and integrity of real time data transmission. The techniques presented here can be a complement to the standard, based on cryptographic techniques, security mechanisms and in conjunction with them, help create more secure and more resistant to attacks real time services as internet telephony (VoIP), videoconference and multimedia streaming systems, online gamming applications etc. In particular, we decided to focus on P2P based applications because of their growing popularity and greater flexibility than traditional client-server applications. Additionally, our design schemes, show usefulness of agents and a significant role that they can play in managing factor like trust or reputation. It turned out that, by using agents, we can greatly simplify the implementation, while maintaining overall application flexibility.

It is important to mention that the described techniques, although very useful, by no means exhaust the subject of security. Other tools, in many cases more suitable for a certain type of thread, still may be found. There are also many other useful security related tasks, to which we may use agents. For example that can check packet delivery time - if this time is longer than normally, we can suspect the original data was swapped or modified.

Acknowledgement: Research reported in this paper was partially supported by the 7FP NoE EuroNF project.

References

[1] Oryńczak G., Kotulski Z., Agent-based VoIP application with reputation mechanisms, Proceedings of the First European Teletraffic Seminar., February 14-16, Poznań (2011): 203.

[2] Cayre F., Bas P., Kerckhoffs-Based Embedding Security Classes for WOA Data Hiding, Information Forensics and Security 3(1) (2008).

[3] Marvel99 Marvel L.M., Boncelet C. G. Jr., Retter C.T., Spread spectrum image steganography, Image Processing 8(8) (1999).

[4] Wang H., Wang S., Cyber warfare: steganography vs. steganalysis, Communications of the ACM - Voting systems 47(10) (2004).

[5] Mazurczyk W., Kotulski Z., New security and control protocol for VoIP based on steganography and digital watermarking, Annales UMCS Informatica AI 5 (2006): 417.

[6] Gambetta D., Can We Trust Trust?, Trust: Making and Breaking Cooperative Relations, Oxford (1990): 213.

[7] A. Josang, Trust and Reputation Systems, Lecture Notes in Computer Science 4677/2007 (2007).

[8] Guha R., Kumar R., Raghavan P., Tomkins A., Propagation of trust and distrust, Proceedings of the 13th international conference on World Wide Web, NY, USA (2004).

[9] Maximilien E.M., Singh M.P., A framework and ontology for dynamic web services Selection, IEEE Internet Computing 8 (2004).

[10] Manchala D.W., Trust Metrics, Models and Protocols for Electronic Commerce Transactions, Proceedings of the 18th International Conference on Distributed Computing Systems (1998).

[11] Josang A., Presti S. L., Analysing the Relationship Between Risk and Trust, iTrust 2004. LNCS 2995 Springer, Heidelberg (2004).

[12] Krukow K., Nielsen M., From Simulations to Theorems: A Position Paper on Research in the Field of Computational Trust, Proceedings of the Workshop of Formal Aspects of Security and Trust, Ontario, Kanada, August (2006).

[13] Josang A., A Logic for Uncertain Probabilities, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9 (2001): 279.

[14] Cahill V., Shand B., Gray E., et al., Using Trust for Secure Collaboration in Uncertain Environments, Pervasive Computing 2 (2003).

[15] Merriam-Webster, Merriam-Webster Online (accessed august (2011)), Available from http://www.m-w.com/

[16] Carter J., Bitting E., Ghorbani A. A.,Reputation Formalization for an Information-Sharing Multi-Agent System, Computational Intelligence 18(4) (2002).

[17] Despotovic Z., Aberer K., P2P reputation management: Probabilistic estimation vs. social networks, Computer Networks 50(4) (2006).

[18] Sabater J., Sierra C., Reputation and social network analysis in multi-agent systems, proceedings of the first international joint conference on Autonomous agents and multiagent systems, USA (2002).