Real-time Transport Protocol (RTP) security

Ville Hallivuori
Helsinki University of Technology
vph@iki.fi

Abstract

This paper describes the Real-time Transport Protocol (RTP). The emphasis is on the security features like confidentiality, integrity and authentication. RTP security features are critically commented and alternative arrangements with their security implications are presented. RTP security is also discussed on multi protocol context where some of RTP's security services are provided by IPsec, SIP, SAP and SDP protocols.

1 Introduction

RTP, Real-time Transport Protocol, is an application level protocol that is intended for delivery of delay sensitive content, such as audio and video, trough different networks. The purpose of RTP is to facilitate delivery, monitoring, reconstruction, mixing and synchronization of data streams. Although RTP does not provide quality of service on IP networks, its mixers can be used to facilitate multimedia delivery on a wide range of link types and speeds. RTP is designed to use both unicast and multicast transport protocols.

Even though RTP is a relatively new protocol, it is widely used by applications like Real Network's RealPlayer, Apple's QuickTime and Microsoft's NetMeeting. Some of the common applications of RTP are audio and video streaming media services and video conferences.

As RTP is usually used trough Internet, the network should be considered as insecure. Although many media streams are publicly available, video conferencing use usually requires confidentiality. In many situations it would be preferable if the user could authenticate the originator and ensure the integrity of media streams.

2 RTP Protocol

2.1 Protocol architecture

RTP is a modular protocol. The base protocol is defined by RFC 1889 [12]. The usage of RTP for a specific purpose requires that also an application area specific RTP profile must be implemented. RFC 1889 defines basic fields for the transportation of real time

data. It also defines RTCP, RTP Control Protocol, whose purpose is to provide feedback on transmission quality, information about participants of RTP session, and enable minimal session control services.

RTP profiles are used for refining the basic RTP protocol to suit for a particular application area. Commonly RTP profiles refine the meaning of the fields provided by the basic RTP protocol. RTP profiles also add new fields and rules. RTP profiles define how and by which formats data is encapsulated to RTP packets.

In contrary to many protocols, RTP is usually implemented by each application, and not by an operating system or by a separate stack. These implementations may, and often are, based on generic RTP libraries. Existence of the application dependent profiles almost mandates that the RTP service must be implemented on an application basis. RTP protocol is transport independent and it can be used over various networking technologies. The most common transport protocols for RTP are IP/UDP, ATM/AAL5 and IPX. Figure 1 shows the RTP on the protocol stack. Note that although RTCP can be considered as a protocol running over RTP, it is actually only a special type of RTP packet.

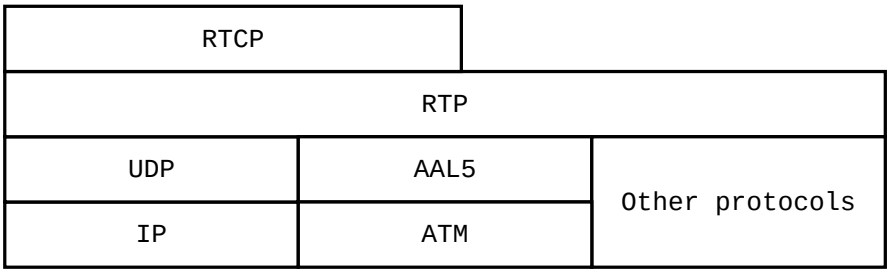

Figure 1: RTP on the protocol stack

On this document we investigate the widely used RTP profile, “RTP Profile for Audio and Video Conferences with Minimal Control”, as defined by RFC 1890 [11]. It should be understood that the basic RTP has never been intended as a complete protocol, but as a framework for building application protocols. As the consequence, a RTP based system commonly relies on non-RTP protocols to negotiate and establish the sessions.

RTP protocol uses synchronization source (SSRC) identifiers as addresses of the peers. SSRCs are unique within the session and are chosen randomly when a participant joins to the session. RTP peers automatically detect and correct SSRC collisions.

2.2 RTP services

RTP defines the roles for two active application level devices that may reside on the network: mixers and translators. In essence the difference between a translator and a mixer is, that mixers change synchronization source identifiers, whereas translators do not.

Mixers have many similar characteristic features with routers, as they connect two or more networks together. Mixers process RTCP packets and may perform payload format translations. They also perform remixing of RTP streams. The purpose of mixers is to allow users behind low speed links to receive high speed transmissions by receiving all high speed streams, down-mixing them to one or more lower speed streams and forwarding these low

speed streams to receivers. Reverse will of course be done for possible return packets. Mixers have to regenerate timing information and change SSRC's, as they essentially create new streams based on one or more existing streams. Mixers are non transparent devices. As mixers may combine several encrypted streams, they are capable of encrypting and decrypting RTP streams.

Translators are transparent on RTP level – they leave SSRC identifiers intact. The purpose of the translators is to perform payload format conversions, tunnel the packets though firewalls, add or remove encryption and enable the coexistence of the different networking technologies. Whereas mixers could be described as RTP routers, closest equivalent for RTP translators is an application level proxy.

There exist a few limitations for placing multiple translators or mixers on the same node – their network address (on IP network consisting port number and ip address) must be unique and they may not perform same forwarding task unless they have not been partitioned to prevent them from forwarding same packets to same receivers [12].

2.3 Network organization

RTP is delivered as multicast, singlecast or as both. As most RTP uses have more than two participants, it is preferable that RTP is transported on multicast capable network. Translator can be located in singlecast network boundary, where it will replicate packets for singlecast participants of RTP sessions.

Multiple RTP streams can be down-mixed and combined for slow links. Application level firewall might be bypassed by using two translators with a tunnel though the firewall between them.

As mixers are allowed to change Synchronization Source Identifiers, it is essential that RTP network is loop free (note that underlying network may contain as many loops as it is desired).

A simple RTP network is presented on figure 2. If ATM network would run IP over AAL5, no translator would be necessary. As ATM has quality of service, it might be preferable to run RTP without IP.

3 Desired security features

3.1 Confidentiality

RTP is commonly used for broadcasting content trough network. On such use confidentiality is not necessarily even desired. As RTP is also commonly used for video conferences and for shared white board applications, need for confidentiality should be obvious – for example when RTP based video conference is used for telemedicine, confidentiality is of paramount importance.

Even on confidential tele medicine session not all information should be confidential –

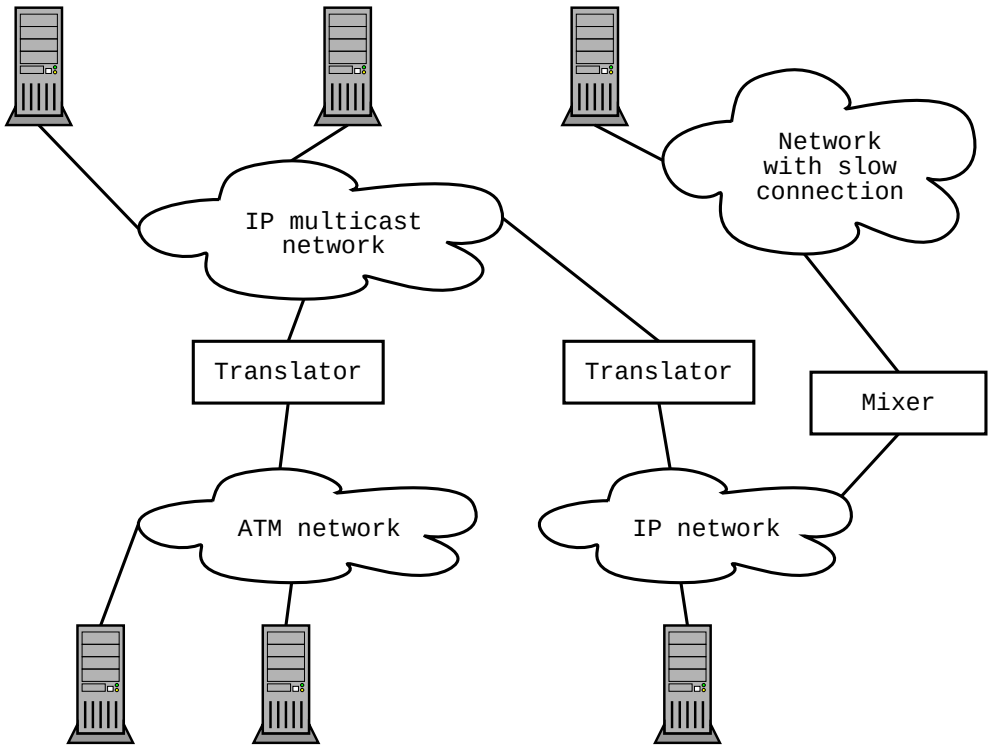

Figure 2: Example RTP network organization

sender and receiver reports containing network performance data can help third party mixers and translators to optimize network usage.

3.2 Integrity and authenticity

Even on public broadcast it is useful to be able to verify both the integrity and the origin of the transmission. This information helps the receiver to assess the trustworthiness of the received information. If anyone could modify CNN news broadcast, which many consider reliable information source, results could be potentially disastrous. It should be noted that the possibility for anonymous broadcast is also important, even though not always politically acceptable feature in some countries.

It is desired that no participant of the same session could masquerade as an other.

3.3 Performance considerations of security features

As RTP is often used for transferring huge amounts of time critical data (eg. video) it is essential that all security features are implemented with minimal delay and jitter. It should be evident that with huge transmission rates even a small timing overhead easily amounts to huge loss of bandwidth.

Although encryption, especially using DES, causes overhead, it is minor compared to CPU

requirements of modern compression algorithms for voice and video. Such a small overhead is hardly noticeable, especially as connection speeds available for general population are much lower than the encryption speed of modern desktop computer [3, 16].

4 Security features provided by RTP

4.1 Confidentiality

The basic RTP protocol, as defined in RFC 1889, provides flexible facilities for encrypting RTP packets. This facility allows to split packets to encrypted and unencrypted parts, and therefore facilitates the need for unencrypted performance statistics.

The default algorithm, which all encryption capable RTP clients must support, is DES-CBC. Algorithm usage is same as defined in RFC 1423 [1]. To prevent known plain text attacks, RTCP headers are obfuscated with 32 bit random prefix. CBC mode has random access property for decryption, which guarantees that the lost packet only prevents decoding of itself and the following packet [15]. This feature is vital, as time sensitive data can usually not be resent.

RTP allows the use of any other encryption algorithm, but the algorithm must be negotiated on non RTP means. An application profile may specify additional methods for encrypting the payload.

It is suggested by RTP standards that when multicast supporting encryption is offered by the network layer, it should be used instead.[12]

In terms on IPsec RTP encryption corresponds using only ESP headers, within pre-established security association.

4.2 Authentication and integrity

RTP standard does not specify any authentication, except that implicit authentication is assumed if encryption key is known. RTP assumes that the lower layers of the network will handle more sophisticated authentication. Integrity is verified by sanity checking decrypted headers. Sanity checks verify the known field values, such as protocol version number, packet length and payload type. The details of the sanity checking are presented on the appendices A1 and A2 of RFC 1889.

RTP limits issuing certain commands, like bye (which is used for session termination) to the peers which command affects. This is enforced by checking the SSRC identifiers of the packets containing the commands. This command authentication mechanism only works, if authenticity of RTP stream is ensured by other means.

4.3 Key management

The only key management feature of RTP is specified in RFC 1890, which specifies a MD5 based method for deriving the encryption key from the password. RTP assumes that more complex key management is either handled by other protocols or by application specific profiles.

On conference type applications (video, audio or even only a shared white board) key management is handled by combination of SIP, SAP and SDP protocols [6, 8, 9, 13]. These protocols feature strong authentication and key exchange features, and provide standardized way to establish encrypted conferences using RTP as the transport protocol.

5 Analysis of RTP security

5.1 Confidentiality

The default DES-CBC algorithm is inadequate, as it can easily be broken with specialized hardware [4]. As DES is mainly designed for hardware implementations it is difficult to implement efficiently on software, and therefore is not optimal choice for application that needs to transfer huge amount of time sensitive data. If encryption is used, RFC 2508[2] compliant header compression for slow links can not be utilized, unless compressing translator privy decryption key for the session. Such may lead to significant performance loss or cause one more node to privy the encryption key. Some bandwidth will be lost to packed padding, but it is not expected to be of significant magnitude, as RTP payload is usually much larger than 64 bit padding (which is the worst case). Also as encryption is performed on transmission units, which may contain several RTP packets, the effect of padding is still further reduced. Otherwise encryption causes no transmission overhead.

If application uses a strong cryptographic algorithm, confidentiality on RTP is preserved.

As all participants (including all mixers) need to privy encryption keys, compromise on one participant leads immediately to the collapse of both the confidentiality and the integrity. In practice this means that it is unlikely that confidentiality could be implemented for large RTP sessions, as most computers are both buggy and poorly maintained.

No encryption related information is available at RTP MIB [10]. Some security problems could be caused by RTP MIB implementing SNMP agents, if public access is allowed: all RTP meta data (senders, receivers, etc) is available. If encryption key generation is (wrongly) seeded by timestamp, accurate session start times on MIB could prove fatal.

5.2 Authentication and integrity

Implicit authentication is totally inadequate, even if stronger (than DES-CBC) algorithm is chosen, as any participant can easily claim to be an other (as participants use same encryption key). Participants are recognized from their SSRCs (as only nearest translator or mixer knows network level address), which obviously can be forged just by writing different SSRC to the packet to be sent. As RTP is a connectionless protocol, forging IP

addresses is trivial. Tracing a forged IP address on RTP session is quite difficult, as both translators and mixers change underlying network addresses. There is also the fact that not all packets originate from IP networks. Tracing forged packet might be possible, if all RTP nodes and routers (and switches on ATM networks) would cooperate, a situation that is very unlikely to happen on Internet.

On the other side of the coin is of course the fact that this poor authentication facilitates anonymity, as it is impossible to say which one of the session members is transmitting.

Another problem is, that change in one block encrypted using DES in CBC mode scrambles that particular block fully, but only propagates predictable bit changes to the next block (the change propagated to the next block is reversing those bits on clear text that were modified on the previous cipher text bloc) [14]. As integrity is verified by header sanity check, it might be possible to exploit this flaw to change RTP payload. For example if client could roughly guess what kind of content a packet contains, it can then modify any block (of encrypted data), and cause scrambling of one block and predefined changes to another. Would the modification be placed properly, the scrambled block would contain voice or video, but the next block might be a header, which would be modified deterministically. To make useful changes, attacker must have a strong idea of the content of the encrypted packets, as otherwise flipping unknown bits will not produce deterministic results.

Better authentication and integrity for singlecast sessions can easily be provided, and is even suggested by standard, using the authentication headers of IPsec. For multicast session situation is more problematic, as there is currently no specification for IPsec for multicast environment. IPsec can neither offer help for session that span multiple network technologies – for example if there is a session between ATM and IP participant, IPsec can only provide services between participant on the IP network and translator on the boundary of the ATM network.

As RTP is network independent, using IPsec will not solve authentication problems for other technologies. Even though ATM networks could be considered secure on certain assumptions, when multiple networks are connected with translators, peer authentication becomes quite difficult. Application profile has to define a peer specific signature, as there is no other way of ensuring that a peer is the one it claims to be.

5.3 Denial of service

As any participant can easily send commands using the SSRC of an other participant, it is possible to disturb other participants on the same session. This could, for example, be used for (repeatedly) dropping certain participants from the session, and thereby denying the service. Same effect can easily be accomplished by repeatedly claiming the SSRC of the other participant, which, to be in accordance with RTP specs, has to stop transmitting and reselect a new SSRC. Both of the above attacks can be limited to the participants admitted to the session by using implicit authentication.

As reception reports are unencrypted, and therefore unauthenticated, it is possible on some cases to reduce the quality of service by injecting forged reception reports to RTP session. For example, an adaptive encoder could be tricked to produce poorer quality sound with reports that indicate huge packet losses.

RTP guidelines for application profiles mandate that application profiles and payload formats should be defined in such a way that they are robust in the real world conditions, even if implementation is poorly done [7].

Obviously normal DoS attacks against RTP aware devices and other IP infrastructure responsible for transporting RTP apply, but those are not within the scope of this paper.

5.4 Complexity

As RTP network performs many services (eg mixing), the network software is complex and introduces a certain risk. As RTP translators are needed for bypassing firewalls, it is probable that these translators will become an easy way for breaching firewalls – even http proxies, which are much simpler than RTP translators, have been a cause of many breakins.

RTP sessions for a large network could easily be forged, if the chosen RTP mixer could be compromised or tricked to mix wrong streams.

6 Supplementary protocols

As RTP is only a framework for implementing protocols, it is not even expected that it would provide all necessary security services. Chapters below describe various solutions that have been used to secure RTP. Use of IPsec is not duplicated below, as its usage is described in other chapters of this document.

It should be noted that conceptually SDP is a higher level protocol than SIP and SAP, which are protocols that can be used to transport SDP [5].

In figure 3 is presented a few common possibilities for arranging secure RTP based conference. Authentication can be handled by IPsec, or left to RTP encryption. IPsec can also provide encryption and keying services (picture does not show IPsec association management). Keys for RTP are distributed using SDP. RTP can also use its own encryption and utilize authentication services of IPsec. Terms AH and ESP on picture refer to the headers used by IPsec (when AH is used IPsec offers authentication services and adding ESP extends services to encryption).

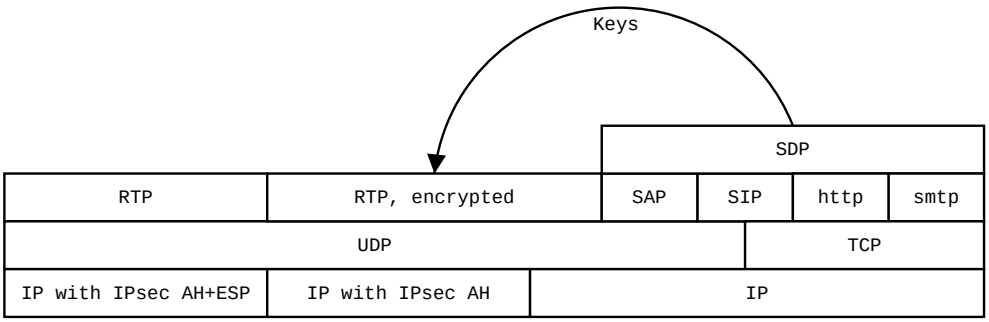

Figure 3: Possible arrangements for secure RTP sessions

6.1 Session Description Protocol

SDP offers facilities to distribute the encryption keys and other security parameters (such as encryption algorithm). SDP itself is neither encrypted nor authenticated and is usually used with SAP or SIP, which in turn provides confidentiality and authentication of keys and other session parameters.

As SDP is a non-interactive protocol, it can also be transported over secure http (https) or over encrypted emails.

6.2 Session Initation Protocol

SIP protocol can be used alone or in conjunction with SAP and SDP to distribute the encryption keys and other security parameters. SIP supports various strong authentication and encryption methods.

6.3 Session Announcement Protocol

SAP protocol is intended for broadcasting information, such as SDP packets to multicast groups. Interested parties may listen such group or contacts database server listening announcements to obtain the session announcements. SAP supports variety of authentication methods, such as PGP. As SAP is intended for public session announcements, broadcasting encrypted announcements is discouraged.

7 Conclusions

Security facilities provided by the RTP protocol are inadequate alone. RTP represents excellent design as it does not try to re-implement security features that can be provided by other layers, but instead limits itself to the task it is supposed to solve. It is clear that RTP is intended to be used over IPsec (when on IP networks). This separation removes a need for complex key and association management on RTP, and reduces the probability of bugs on RTP implementations.

RTP encryption feature is clearly intended to be transitional and is intended to be replaced by services provided by the lower protocol layers. As there is no multicast capable IPsec yet, RTP security must be facilitated by other means. The currently accepted mean is to leave the key and algorithm management to upper layer session establishment protocols and use RTP encryption for confidentiality. Authentication on multi participant RTP sessions is inadequate. As RTP is networking technology independent IPsec will not be the magic bullet that solves all security needs – security solutions for other network layers are needed. Some form of authentication has to be done on levels above RTP, as the lower levels can not accommodate identifying and authenticating transmitting peer on a network which has mixers, translators and possibly multiple network level technologies.

Table 1 summarizes security features provided by some of the common protocol configurations utilizing RTP protocol. The table assumes that pure RTP has some non RTP mean to

distribute the keys (such as MD5 based derivation method discussed earlier on this paper). It is also assumed that SDP has some secure transport, such as SIP or https.

	RTP	RTP+IPsec	RTP+SDP
Key / algorithm setup	no	yes	yes
Confidentiality	yes	yes	yes
Session authentication	implicit	yes	yes
Session integrity	some	yes	some
Transmitter authentication	singlecast	singlecast	singlecast
Multicast support	yes	no	yes

Table 1: Security features provided by common RTP configurations

For singlecast sessions using IPsec is an adequate solution. For multicast sessions much is expected from the future development of IPsec. Author is not aware of any other development for securing RTP.

It appears that as it is with IPsec and other encryption technologies, the largest problem with RTP encryption is the key distribution. The key distribution is usually the heel of Akhiles, as it can not be solved just by using smart protocols – it requires infrastructure for the distribution and certification of the keys.

8 Terms

AAL5 ATM Adaptation Layer 5. An ATM sublayer for computer networks.

AH Authentication Header, and IPsec header that authenticates by cryptographic means datagram's origin and verifies it's integrity.

ATM Asynchronous Transfer Mode. A common cell based network technology.

CBC Cipher Block Chaining. Encryption mode which hides reoccuring patterns from encrypted stream.

DES Data Encryption Standard. Robust cipher which is currently considered somewhat weak due to it's small key size.

ESP Encrypted Security Payload, and IPsec header that indicates that datagram is encrypted.

IPsec Network level encryption and authentication service for IP networks. IPsec is not widely used, but it is expected to gain popularity in near future.

IPX Internetwork Packet Exchange. A legacy network protocol by Novel.

Mixer RTP device that changes synchronization source identifiers.

SAP Session Announcement Protocol.

SDP Session Description Protocol.

SIP Session Initiation Protocol.

SNMP Simple Network Management Protocol

RTCP RTP Control Protocol.

RTP Real-time transport Protocol.

SSRC Synchronization source. An unique 32 bit identifier associated with single RTP packet stream originator. SSRC is independent of transport address.

Translator RTP device that does not change synchronization source identifiers.

References

[1] D. Balenson. RFC 1423: Privacy enhancement for Internet electronic mail: Part III: Algorithms, modes, and identifiers, February 1993.

[2] S. Casner and V. Jacobson. RFC 2508: Compressing IP/UDP/RTP headers for low-speed serial links, February 1999.

[3] W. Dai. Crypto++ 3.1 benchmarks. http://www.eskimo.com/ weidai/benchmarks.html, April 1999.

[4] Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*. O'Reilly & Associates, Inc., 103a Morris Street, Sebastopol, CA 95472, USA, Tel: +1 707 829 0515, and 90 Sherman Street, Cambridge, MA 02140, USA, Tel: +1 617 354 5800, July 1998.

[5] M. Handley, J. Crowcroft, C. Bormann, and J. Ott. The internet multimedia conferencing architecture, draft-ietf-mmusic-confarch-03.txt, work in progress, July 2000.

[6] M. Handley and V. Jacobson. RFC 2327: SDP: Session description protocol, April 1998.

[7] M. Handley and C. Perkins. RFC 2736: Guidelines for writers of RTP payload format specifications, December 1999.

[8] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. RFC 2543: SIP: Session initiation protocol, March 1999.

[9] Mark Handley, Colin Perkins, and Edmund Whelan. Session announcement protocol, draft-ietf-mmusic-sap-v2-06.txt, work in progress, March 2000.

[10] IETF Audio Video Transport Group. Real-time transport protocol management information base, draft-ietf-avt-rtp-mib-13.txt, work in progress, March 1999.

[11] IETF Audio-Video Transport Working Group and H. Schulzrinne. RFC 1890: RTP profile for audio and video conferences with minimal control, January 1996.

[12] IETF Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 1889: RTP: A transport protocol for real-time applications, January 1996.

[13] Peter T. Kirstein, Ian Brown, and Edmund Whelan. A secure multicast conferencing architecture. http://www.cs.ucl.ac.uk/staff/I.Brown/pimms/secure-conferencing.html, December 1999.

[14] A. J. (Alfred J.) Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. The CRC Press series on discrete mathematics and its applications. CRC Press, 2000 N.W. Corporate Blvd., Boca Raton, FL 33431-9868, USA, 1997.

[15] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley and Sons, Inc., New York, NY, USA, second edition, 1996.

[16] Bruce Schneier. Speed comparisons of block ciphers on a pentium. http://www.counterpane.com/speed.html, February 1997.