Applied IPv6 Security

Version 0.8.9
23. Nov 2002

Dominik Schnitzer[1]

This article gives a short introduction to the IPv6 protocol, used for the next generation Internet, and an overview of its security features. IP-Sec guaranteeing Encryption and data Authenticity is a required part of the new protocol stack. In addition to this introduction to IPv6-Sec the article pinpoints weak and dangerous IP-Sec configurations and shows how to securely do it the right way. It closes with a concise how to describing in three steps how to get on line in the already heavily deployed IPv6 Internet, the 6Bone, and start experimenting yourself.

[1] <dominik@schnitzer.at>

Contents

1 Introduction

This document focuses on the next generation Internet protocol (IPv6) and the security mechanisms it implements. In some years IPv6 will fully replace the current IPv4 based Internet and it's sub networks. Some say this will happen soon because of IP address space running out, others say the new features, like IPv6's extreme mobility, simple node configuration or good support for additional extensions will finally be the dead of IPv4 and rise of IPv6. IPv6 essentials are the content the first part of this article, sketching the protocol basics and giving you a sneak preview how the next generation Internet looks like.

Besides the cool IPv6 features listed above, an IPv6 network stack also requires a full implementation of IP Security features. That means that every participant in the IPv6 Internet has the possibility to encrypt and sign it's network traffic, generally and in theory making making all currently deployed insecure Internet services secure! If there wasn't the problem of a non existing Public Key Infrastructure (PKI). IP-Sec is as a protocol already available and deployed for on IPv4 networks and did well in providing security in existing IPv4 networks using it. In IPv4 IP-Sec is just an optional add on, you can or can not install and use. In the second and more in depth going part of this article, the functioning of IP-Sec and it's integration in the IPv6 stack is pictured and the different ways of using IPv6 security features to secure your Internet business are worked out: authentication, types of network tunneling, encryption and choosing the appropriate and most secure solution for different tasks. The second part also details a security analysis of IP-Sec especially commenting on possible dangers of the complexity of the IP-Sec standard. An outlook, showing how to IPv6 connect and start experimenting today, finalizes the article.

2 IPv6 Essentials

Many things in a computers life change fast, usually indicating that things need improvement or are just outdated and old. On the other hand other things, unnoticed like the beloved 1.44MB floppy [4] disk which debuted 1984, just work for years and years.

The IPv4 network protocol is also among those working things. It's in use for over 15 years now and emerged to the most widely used network protocol in the world. We all know how to configure, use it and find network errors with it. So why would one want to replace and update IPv4?

IPv4 survived until today mainly because of it's simplicity and extreme extensibility. To keep track with the new requirements for the worldwide Internet, IPv4 was extended with new optional features often, just take like NAT (Name Address Translation) or IP-Sec as an example. But one limitation of IPv4 could not be worked out: IPv4 addresses are 32 Bit, and therefore have for the current Internet a very limited address range. First concerns about the Internet and it's IP address shortage were raised in a memo 1992 [1]. The IETF even dated the Internet day of doom in March 1994. Eight years after we still use IPv4, since last minute tweaks like NAT and new routing protocols apparently prolonged the life of IPv4 a last time.

Now having learned from the problems occurring with IPv4 deployment and trying to foresee the upcoming growth of the Internet, a new heavily extended and improved IP standard was created: IPv6 – capable of addressing 1 Trillion hosts and 1 Billion different networks [5], should make IPv6 a long used standard.

2.1 Address Notation

IPv6 addresses are 128 Bit, so in the currently known notation an IP address would look like:

```
254.128.0.0.0.0.0.0.2.2.179.255.254.31.131.41
```

not easily remembered by a human. So the hexadecimal representation was chosen – dividing the address into eight 16 Bit blocks. In addition to that leading zeroes in a 16 Bit block can be skipped:

```
FF80:0000:0000:0000:0202:B3FF:FE1E:8329
FF80:0:0:0:202:B3FF:FE1E:8329
```

You can also replace a set of repeated zeroes by a double colon, which would make a typical IPv6 address look like this:

```
FF80::202:B3FF:FE1E:8329
```

2.2 Address Types

IPv6 has three different types of addresses [5], introducing a new type of address and obsoleting the IPv4 broadcast address:

1. *Unicast Addresses* This type of address uniquely identifies an interface on an IPv6 node. Packets sent to an Unicast interface are delivered to this very interface.

2. *Multicast Addresses* identify a group of addresses. When sending packets to a Multicast address, every interface configured with this Multicast address receives the package. For example the predefined Multicast address `FF05::101` specify all NTP time servers on the same site as the sender. Multicast addresses are also used as a broadcast address, known from IPv4.

3. *Anycast Addresses* Groups of hosts are configured with an Anycast address. When sending a package to an Anycast address, only one host of the Anycast group, the nearest one, receives the package. An Anycast address already in use is the address `::192.88.99.1`, specifying your nearest IPv6 to IPv4 router (basically an IPv4 node acting as an entry point to the IPv6 Internet).

2.3 Header Structure

The exact header structure in IPv6 packages is specified in the RFC 2460 [2]. The header information of an IPv6 package has a fixed length of 40 Bytes. To add extended header information to a package IPv6 uses the so called extension headers. Besides IPv6 obsoleted many special IPv4 cases like active fragmentation of packages, in the spirit of simplicity. Since a source and destination address in IPv6 take 2x16 Bytes, only 8 Bytes are left for additional package header information. The following fields characterize a standard IPv6 header [5]:

- *Version, 4 Bits*: The Version of the protocol (6)
- *Traffic Class, 1 Byte*: Used to set priorities to the packages, to privilege certain data packages like voice or video stream packages.
- *Flow Label, 20 Bits*: Information for routers to helping them identifying IPv6 packages that belong together. Enables faster processing of data.
- *Payload Length, 2 Bytes*: Size of data following the IPv6 header
- *Next Header, 1 Byte*: An identification byte for eventual following sub headers in the IPv6 header chain. IPv6 uses header chains (extension headers) to integrate other IP extensions like security, authentication or routing headers.

- *Hop Limit, 1 Byte*: Number of maximum hops. Every time the package passes a router, the router has to decrement this value. If Hop Limit gets zero, the package is discarded and an ICMP error message is sent back to the sender.
- *Source and Destination Address, 32 Bytes*

2.3.1 Extension Headers

At the time of this writing, IPv6 defines six different extension headers [6]: *Hop-by-Hop options header*, *Routing header*, *Fragment header*, *Destination Options header*, *Authentication header* and *Encrypted Security Payload header*. As you may have noticed, the last two extension headers already indicate where encryption and security mechanisms in IPv6 have been implemented.

3 IPv6 Security

During the last years it became clear, that good and standardized security mechanisms will be the requirement for the further success of the Internet. For the networking experts it was very surprising that the Internet with all it's inherited insecurities became a substantial and an important factor in real world/real money business. Many security mechanisms had to be implemented to ensure security for important data transfers via the Internet. SSL on base of the HTTP protocol or SSH as a replacement for the insecure telnet service are the most famous examples of security add ons which were invented to make insecure services usable again. All of these security mechanisms have one thing in common, they are implemented on the Application Layer – written for a special kind of application, using a special more or less secure encryption or authentication method.

IP-Sec takes a different approach in implementing security. It implements security on top of the network layer, thus enabling all services working on top of IP to automatically use it's security mechanisms.

3.1 Internet Threats

Here is a list [5] of the three different archetypes of attacks on networked services. It would be nice if the use of IP-Sec could prevent all of them. The following chapters will come back to these threats and re-analyze them regarding IP-Sec and IPv6.

- *Disruption of Service* or Denial of Service attacks. This kind of attack stops services from running normal by stopping, overloading or simply destroying them. They are easily detected cause they have an immediate and noticeable impact.

- *Fabrication, Modification or Deletion of Information* These attacks are not easily detected and are characterized by infiltrating false information in payment systems, email or any other trusted communication.

- *Electronic Eavesdropping* Passive attacks like this are usually impossible to detect and in a huge network like the Internet impossible to prevent. Attacks like Sniffing IP traffic, or to give a more obvious example, simply silently duplicating and storing all pages sent to a printer, fall in this category of attacks.

How can IP-Sec be used to prevent or complicate attacks like this? Can it prevent those attacks at all?

3.2 IP-Sec Architecture

The IP Security framework has been standardized by the IETF in a large effort. The whole standard is still under development, and citing the official progress document, the IP-Sec standard will be ready in December 2002. The progress document is openly available at `http://www.ietf.org/html.charters/ipsec-charter.html`.

3.2.1 Security Associations (SAs)

The Security Association is the internal base construct of IP-Sec [6]. For each connection a SA specifies the communication mode to be used. For example, let's assume an IP-Sec data package comes in, now the recipient of this package uses ¡protocol, source-IP, source port, destination-IP, destination port¿ as a key to lookup the matching SA associated with this very connection. Security Associations are stored in the so called *Security Association Database* (SADB). Static system wide security policies, explicitly telling a system to use certain SA when a connection occurs are stored in the Security Policy Database (SPD) and can be adopted by the system administrator. For instance in Microsoft Windows the service managing all these security rule sets is called "IP-Sec Policy Agent". In the Linux IP-Sec Free/SWAN package SAs are managed by the `spi` command.

These are the most important parameters which specify a Security Association [5]:

- The Security Parameter Index (SPI)
- The desired IP-Sec service (AH or ESP)
- The transmission mode (Tunnel/Transport)
- Source and Destination address
- Used authentication/encryption method
- Used Keys

Establishing SAs with another host is done by using the IKE (Internet Key Exchange) protocol discussed later.

3.2.2 Authentication

Knowing how security is managed in IP-Sec we now request authenticity for our packages, meaning we want to be certain that we receive a data package unmodified and from the very computer we requested it. To ensure authenticity IP-Sec adds a new extension header to the header chain: the *Authentication Header (AH)*. This header adds cryptographic information to the data package, so the sender of the data package can clearly be verified. The Authentication Extension header includes the following informational fields in the data package [5]:

- *Next Header, 1 Byte*: Like every extension header this parameter specifies the next IPv6 header to come after the AH.
- *Length of Payload, 1 Byte*: Describes how many 32 Bit fields follow the SPI field, necessary because different authentication algorithms are allowed.
- *Reserved, 2 Bytes*: not used yet, zeroed data
- *Security Parameter Index (SPI), 4 Bytes*: Indicates the checksum algorithm used. Currently two different checksum algorithms are required to be implemented: `HMAC-MD5-96` and `HMAC-SHA-1-96`. HMAC is a a mechanism for message authentication using cryptographic hash functions. It is used in combination with MD5 and SHA-1 or other iterative cryptographic hash function.

- *Sequence Number, 4 Bytes*: The sequence number is there to prevent replay attacks. The known limitation of this replay attack prevention number is that connections using more than 2^{32} packets can theoretically be compromised.
- *Authentication Data, Variable Byte Length*: a cryptographically secure checksum over the payload and some header fields of the IP and extension headers.

Basically there exist two modes of Authenticating a data package:

1. **Payload Authentication**: In this Mode only the carried data plus some IP headers are signed and thereby authenticated. This usually happens in the so called Transport Mode. *Figure 1* illustrates this behavior. All signed header information is filled with dark gray color.

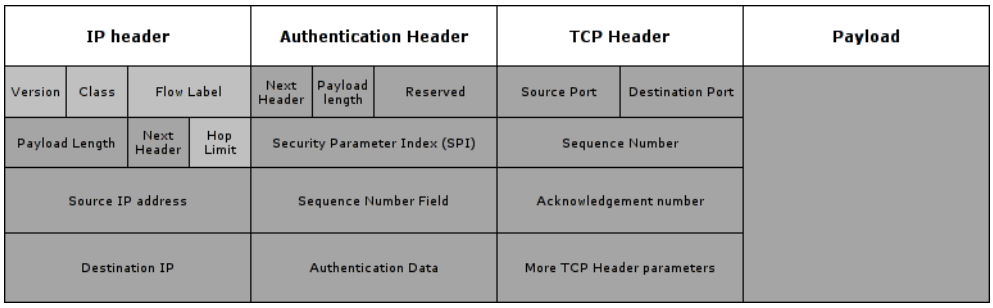

Figure 1: Transport Mode Authentication header (dark gray = authenticated data)

2. **Header and Payload Authentication**: This authentication mode signs the whole IP packet. To achieve this, the whole original data package has to be signed and encapsulated in a new IP package. This mode is called the Tunnel Mode, coming from the fact, that the original IP package is transferred tunneled in a new package. *Figure 2* shows the authenticated data in this mode.

By using IP authenticated traffic, one can prevent that an attacker can infiltrate his spoofed packages in the communication. Moreover even replay attacks can be prevented, rendering overtaking of sessions by a listening attacker impossible. Authenticated IP traffic is for example sufficient for exchanging public, well known information like routing traffic. But it's totally insufficient for transmitting important data, since it still can be read by an eavesdropper.

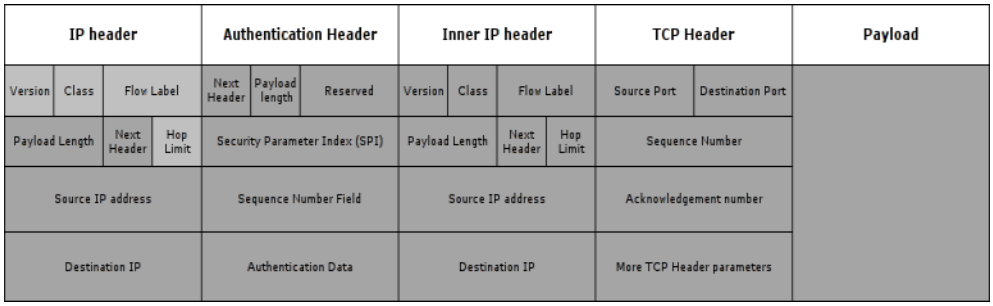

Figure 2: Tunnel Mode Authentication header (dark gray = authenticated data)

3.2.3 Encryption

Another extension header in IPv6, namely the Encryption Security Payload (ESP) extension header, finally adds encryption to the transmitted package. Typical applications which then could use this encryption are FTP, telnet or mail sending/retrieving sessions. Making all the named services secure without the need of changing the protocol. The ESP extension header has the following required fields [6]:

- *Security Parameter Index (SPI), 4 Bytes*: The encryption algorithm used
- *Sequence Number, 4 Bytes*: Prevents attacks relying on replaying traffic
- *Payload Data, Variable Byte Length*: The encrypted data.

IPv6 IP-Sec requires one encryption algorithm to be available on every system implementing IPv6: DES-CBC (Data Encryption Standard in Cypher Block Chaining mode). DES operates with a key length of 56 Bit and is not recommended to be used anymore. On the 19th January 1999 for example a 56 DES key was cracked in about 22 hours by distributed.net, a distributed computing project. The encrypted message stated: *See you in Rome (second AES Conference, March 22-23, 1999)* [7]. AES is the next generation encryption standard and is, since 26th of May 2002, the official encryption standard of the USA [8]. AES will be available for IP-Sec too. However today you should in any case use Triple-DES in your IP-Sec connections which is the best encryption algorithm, required to be implemented in a current IPv6-Sec protocol stack.

Like authenticated data, encrypted data can be sent in two different ways:

1. **Payload Encryption**: This is the so called transport mode, which just encrypts the payload (including TCP information like the destination port). Figure 3 shows the encrypted parts of a package sent in transport mode.

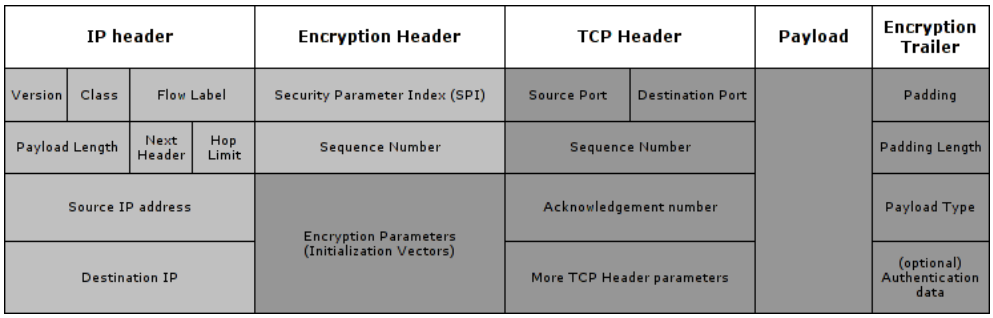

Figure 3: Transport Mode Encryption (dark gray = encrypted data)

2. **Header and Payload Encryption**: If it is required to encrypt the whole IP package, the encrypted IP package has to be wrapped with an outer IP package (tunneled), enabling routers to read the important outer IP header information.

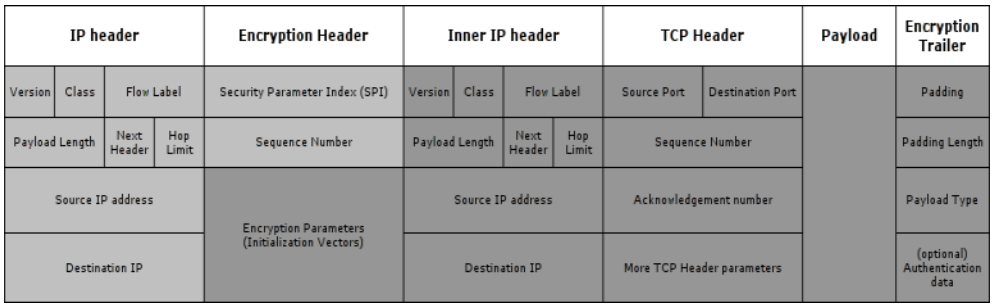

Figure 4: Tunnel Mode Encryption (dark gray = encrypted data)

By using encryption it's possible to hide sensitive data from an attacker, making it impossible for the attacker to read the information sent. But by just using encryption for our data, as you can see in *Figure 4*, the attacker can again change or spoof important header information, cause the packet is not authenticated. It's not recommended to solely rely on encryption when transmitting data with IP-Sec.

3.2.4 Authentication and Encryption

Using the mechanisms shown above it's possible to first encrypt and then authenticate the encrypted package. This is done by preceding an AH header before an ESP header. Because integrity, authenticity and confidentiality is wanted in the most cases, it's possible to append an AH trailer (see *Figure 4*) to the ESP trailer which results in smaller IP packets. This combination is the most secure way of ensuring packet integrity and security and should be used whenever encryption is wanted.

3.3 Putting it Together

To establish a Security Association between two hosts for furthermore using authentication and/or encryption in communication, the two hosts must first agree upon the common security policy and used cryptographic algorithms. Therefore the IKE (Internet Key Exchange [3]) protocol was defined. IKE is implemented on application layer, working on UDP port 500 and generally is an adaption of basically three more general protocols [5] and [6]:

1. *ISAKMP*: The Internet Security Association and Key Management Protocol (defined in RFC 2408) manages the initialization of connections and defines their SAs by describing the negotiated connection properties.

2. *IP-Sec DOI for ISAKMP*: A tight specification which tells how to interpret the rather abstract ISAKMP specification in regard to IP-Sec

3. *Oakley key determination protocol*: Is defined in RFC 2412 and bases on the Diffie/Hellman key exchange.

IKE can basically be described as a negation protocol, that uses ISAKMP to exchange key and SA information. Besides IKE there exist other proposals negotiation like Photuris (experimental RFC 2522, 2523) and SKIP.

IKE works in two phases. In the first phase The two machines involved set up a secure authenticated channel. For instance this channel could be set up by encrypting the data with an RSA key. After obtaining the ISAKMP SA, the communicating partners use the secure channel in the second phase to exchange IP-Sec SAs necessary for the upcoming IP-Sec traffic. IKE is defined as a very flexible protocol, allowing further extension to allow fetching eventually needed public keys from a PKI (Public Key Infrastructure) in the Internet, which does not exist yet.

3.4 Current Implementations

Today there already exist many more or less stable IPv6 implementations. Not all of them implement IP-Sec yet. The following table is a summary of the capabilities of the 4 main IPv6 stacks being implemented.

GNU/Linux Vanilla Kernel (2.4) does not support IP-Sec out of the box. An IPv6-Sec stack is being developed by the USAGI project. AH in transport mode is fully working, ESP and AH tunnel mode are in development. See `http://www.linux-ipv6.org/`

Windows XP does support IPv6 and IP-Sec out of the box with AH tunnel and transport mode working. ESP mode works too, but ironically does not encrypt any data sent. Has no IKE support, currently SAs have to be set manually. See `http://www.microsoft.com/windowsxp/pro/techinfo/administration/ipv6/default.asp`

**BSD* Most complete and best freely available implementation of an IPv6 and IP-Sec stack. The KAME project is working on it's implementation. Their work was already merged into FreeBSD 4.0, OpenBSD 2.7, NetBSD 1.5 and BSD/OS 4.2. See `http://www.kame.net/`

Cisco Latest Cisco IOs releases (starting with version 12.2) have full IPv6 and IP-Sec support. There are still missing some special RFCs (like IP-Sec over a NAT-ed connection) but basically their stack is ready for use in production environment. See `http://www.cisco.com/warp/public/732/Tech/`

3.5 Open Problems

It has already been mentioned, that NAT is problematic when used in conjunction with IP-Sec. NAT is actively rewriting packet headers, changing source and destination address in the packet header. This makes authentication impossible. Another point not yet addressed 100% is the IKE process, where for instance problems arise when the mobility features or Quality of Service come into play.

Critics of IP-Sec [9] also state that IP-Sec is too complex to be secure: T̈he simplest solution is the best.̈ The IP-Sec standard is a collection of more than 35 RFC documents, introducing a great level of interpretation range and complexity. The many different ways of reaching the same goal with IP-Sec are also a point of criticism. Adding further functionality such as the

various modes of tunneling will certainly add complexity and thus endanger the original idea if providing a simple but secure security mechanism for IPv6.

3.6 Conclusion

Despite all criticism IP-Sec is the best network security solution currently available. It allows two networks to securely connect over the Internet, or just enabling secure data transmission for network services operating in clear text. It should be noted, however, that IP-Sec does not automatically secure everything, it's as secure as the computer, operating system or application it is working on. IP-Sec does attempt to standardize security mechanisms in the Internet and is a great step toward a more secure Internet.

4 Getting Started

This section introduces you how to start your own experiments with IPv6 and IP-Sec. The steps are sketched in a general way assuming that you have a static IPv4 address, no existing IPv6 infrastructure. Besides the here described 6Bone many other (mostly commercial) testing networks like the 6net (Cisco) exist.

To connect to the 6Bone, you generally have to follow those three steps described here:

1. *Enabling the IPv6 stack* The first step before experimenting with the things described above, you have to download download your latest IPv6 vendor stack and enable it. For instance In GNU/Linux, IPv6 support has to be compiled into the kernel, in Windows XP IPv6 and IP-Sec support just has to be activated with a special command line tool.

2. *Getting on the 6Bone* After enabling the IPv6 protocol, your Ethernet cards already automatically have an IP-address assigned: the so called link local address. An unique IP address computed using your MAC address. To get on the 6Bone, you'll need a global IPv6 IP. You can now compute your personal 6to4 IPv6-address:

   ```
   2002:hhhh:hhhh::1
   ```

hhhh:hhhh is the hexadecimal equivalent of your global IPv4 address. You now have an address space of 2^{80} (globally valid) different IPv6 IPs, which gives you more IPs than the current IPv4 Internet has for experiments. After setting the calculated IP on your system, just add:

```
::192.88.99.1
```

as your default gateway to the 6Bone. `::192.88.99.1` is a special notation for old IPv4 addresses in IPv6. 192.88.99.1 is an Anycast address (as described in the first Section) which always points to the next 4to6 router. Your closest entry point in the 6Bone could be right at your ISP, if you are lucky.

3. *Starting Experiments* Using the `ping6` command, you can now try to ping me on the 6Bone or visit an IPv6-only website: `http://zoidberg.ipv6.chello.at/` using an IPv6 enabled browser like Mozilla.

```
aeneas@blackhole:~$ ping6 -c2 -n atlantis.relax.ath.cx
PING atlantis.relax.ath.cx(2002:3eb2:51cf::1) 56 data bytes
64 bytes from 2002:3eb2:51cf::1: icmp_seq=1 ttl=64 time=0.35 ms
64 bytes from 2002:3eb2:51cf::1: icmp_seq=2 ttl=64 time=0.31 ms
```

Welcome to the next generation Internet!

References

[1] P. Gross and P. Almquist, 1992. *IESG Deliberations on Routing and Addressing*, IETF, `http://www.ietf.org/rfc/rfc1380.txt`.

[2] S. Deering and R. Hinden, December 1998. *Internet Protocol, Version 6 (IPv6) Specification*, IETF, `http://www.ietf.org/rfc/rfc2460.txt`.

[3] D. Harkins and D. Carrel, November 1998. *The Internet Key Exchange (IKE)*, IETF, `http://www.ietf.org/rfc/rfc2409.txt`.

[4] Computerhope, 2002. *Computer Hardware - Information about computer floppy drives*, computerhope.com, `http://www.computerhope.com/help/floppy.htm`

[5] S. Hagen, July 2002. *IPv6 Essentials - Integrating IPv6 into your IPv4 Network*, O'Reilly, p 1-4, 12-16, 77-104.

[6] W. Stallings, 1998. *Cryptography and Network Security: Principles and Practice*, Prentice Hall, p 399-432.

[7] D. McNett, Janurary 1999. *Press release - US government's encryption standard broken in less than a day*, distributed.net, `http://www.distributed.net/des/release-desiii.txt`

[8] US Computer Security Devision, January 2002. *Advanced Encryption Standard (AES) - Questions and Answers*, National Institute of Standards and Technology, `http://csrc.nist.gov/encryption/aes/aesfact.html`

[9] N. Ferguson and B. Schneier, 2002. *A Cryptographic Evaluation of IPsec*, Counterpane Labs, `http://www.counterpane.com/ipsec.html`