Support of NAT traversal in IKEv2 implementation

Miljen Mikić, Stjepan Groš, Vlado Glavinić
Department of Electronics, Microelectronics, Computer and Intelligent Systems
Faculty of electrical engineering and computing
Unska 3, 10000 Zagreb, Croatia
Telephone: +385 1 6129-935, miljen.mikic@fer.hr

Summary – Network Address Translation (NAT) is a mechanism introduced with a primary purpose to postpone the problem of IPv4 addresses shortage, but it also has some other uses, most notably, easier site renumbering. From its introduction in the middle of 90's up until now, usage of NAT solution on the Internet has surged, but in the same time it is a very controversial subject since it introduces many problems into existing protocols. While working on IKEv2 (Internet Key Exchange v2) implementation we had to analyze NAT interrelation with IKEv2 protocol, enumerate problems, analyze them and finally propose appropriate solutions, if they are possible at all. This paper is a result of this analysis and it summarizes our findings. We also describe implementation we've done based on the analysis.

I. Introduction

One of the bigger problems of the Internet is a shortage of IPv4 addresses. The solution developed during past decade is a next generation IP architecture, IPv6, that has bigger address space. This address space should be large enough for a foreseeable future. Still, there is one big problem with this solution, namely, it is a big change which needs much time to take place and still it is not clear when it will "take control" over the Internet. Second solution is a Network Address Translation (NAT) [9]. This solution was meant to be temporary, but it's now in widespread use and it's actually holding back wider deployment of IPv6.

Apart from the address shortage, Internet also has security related problems. There are different solutions for these problems currently in use, of which we are particularly interested in IPsec. IPsec is an architecture [2], currently in a second generation, that defines behavior of compliant IPsec nodes, and a set of three main protocols. Two protocols offer traffic protection at Layer 3 of ISO/OSI reference model. Those are Encapsulating Security Payload (ESP) [12] used for traffic encryption and integrity protection, and Authentication Header (AH) [13]. Of those two, ESP is mandatory to implement, while AH was mandatory, but now is optional. The third protocol is Internet Key Exchange version 2 (IKEv2) [1] and it is used for authentication, authorization and key exchange within IPsec (Internet Protocol Security) architecture.

Widespread deployment of NAT based devices creates substantial problems to IPsec protocols. As we implemented NAT in IKEv2 protocol we had to do thorough analysis of possible problems and their solutions. This paper summarizes our findings and gives overview of their application in IKEv2 implementation.

The paper is organized as follows. First we give a short overview of NAT mechanism in the following section. Then, in the third section we overview basic mechanisms behind IPsec based virtual private networks. The fourth section enumerates problems that are caused by NAT devices between VPN gateways, or hosts and gateways that establish VPNs. In section five we give an overview of NAT traversal in IKEv2 implementation. The paper finishes with a conclusion in section six, and list of references.

II. Network Address Translation

NATs were introduced primarily because of the shortage of IPv4 addresses. IP nodes that are "behind" a NAT device have IP addresses that are not globally unique. They are more often assigned from some space that is unique within the network behind the NAT but which are likely to be reused by nodes behind other NATs. Node behind a NAT, which wants to communicate with other node on the Internet, is assigned a global IP address by NAT box which results with change of source IP address for outgoing packets. Similar situation is when destination node is behind a NAT, then for incoming packets NAT box changes destination IP address to the private IP address of node on the internal network. NAT box keeps the mapping for the duration of the communication. This duration is estimated by NAT box heuristically. Mapping is often achieved by additional translation based on UDP or TCP ports. In that case, NAT box is known as a NAPT box.

Figure 1 shows a basic concept of NAT mechanism. At the left side we have a private network which means that addresses of the internal nodes cannot emerge on the public Internet. Therefore, when some host from the internal network wants to access a server on the public Internet, NAT box translates these addresses to addresses of its public interfaces.

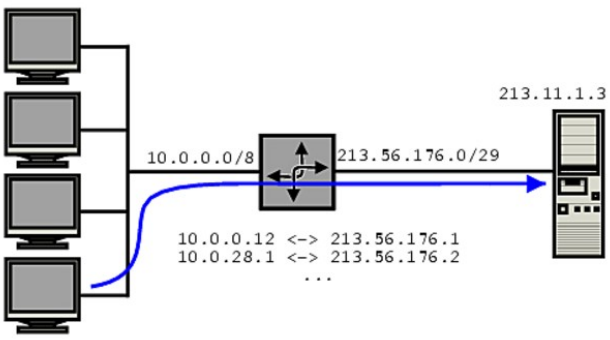

Figure 1. Multiple hosts behind a NAT box accessing a server on the public Internet

There are many protocols having complications with NAT [10]. Applications such as FTP, H.323, SIP, RTSP

use a control connection to establish a data flow and they are usually broken by NAT devices enroute. This is because these applications exchange address and port parameters within control session to establish data session and session orientations. Most likely reasons for failures are that addressing information in payload could be realm specific and second, that control sessions permit data sessions to originate in a direction that NAT might not permit. Peer to peer applications also have problems with NAT. They can be originated by any of the peers and external peers will not be able to locate their peers in private realm unless they know the externally assigned IP address or FQDN ahead of the time. IP fragmentation with NAPT enroute is also an issue, as described later. Applications requiring retention of address mapping or requiring more public addresses than available are broken by NAT for obvious reasons. Namely, in the first case NAT cannot know this requirement and may assign external addresses between sessions to different hosts and in the second case NAT is limited by number of available public addresses.

III. Virtual Private Networks and IPsec

Although there are different definitions depending on the scope of the network, we could say that virtual private network (VPN) is a secure, private communication tunnel between two or more devices across a public network (Internet for example). The traffic within the VPN tunnel is encrypted so that other users cannot read it even if it has been intercepted.

By implementing a VPN, a company can provide access to the internal private network to clients around the world. Before VPNs, remote workers accessed company networks over private leased lines or through dial-up remote access servers. With VPN, they got the most cost effective way to connect by tunneling through the public network.

An Internet VPN solution is based on client/server architecture. A client (remote host) wanting to connect to the company internal network first needs to gain access to the Internet by any public ISP. After that, he initiates a VPN connection to the company VPN server via a VPN client installed on the remote host. Once the connection has been established, the remote client can communicate with the rest of the corporate network over the Internet just as if it were a local host.

Layer 2 protocols that are used to establish a VPN connection are PPTP (Microsoft's proprietary solution) and L2TP (Cisco's proprietary solution). L2TP is often used in conjunction with IPsec for encryption and therefore, better security. As we already mentioned, IPsec operates on the Layer 3 and provides data authentication, confidentiality, integrity protection and anti replay protection. As such, IPsec is one of the most widespread VPN technologies in today's enterprise, service provider and government networks.

IPsec keeps records about traffic which needs to be protected and how to protect it in two databases – SPD (Security Policy Database) and SAD (Security Association Database). SPD contains entries about security policy – which traffic to protect, which protocol to use, level of protection etc. The most important part of SP database are *traffic selectors*. Traffic selectors specify which packets to protect by specifying source and destination addresses, upper layer protocols and ports. IPsec is based on SA (Security association), which is a set of security parameters, for instance crypto algorithms used in communication. SA is uniquely defined by protocol (AH or ESP), destination IP address and SPI (Security Parameters Index). Two sides will establish connection if and only if they successfully negotiate security parameters for the connection.

ESP and AH are two main security protocols in the IPsec architecture which assure traffic protection. AH is used for authentication and integrity check, while ESP is used primary to enable confidentiality and optionally, authentication and integrity check. As we can see in Figure 2, AH protects IP payload and whole IP header except variable fields like TTL. AH header is situated between IP header and IP payload (which is actually header and payload from the upper layer). Integrity and authentication are obtained via ICV (Integrity Check Value) which is calculated from the data that is protected together with shared secret.

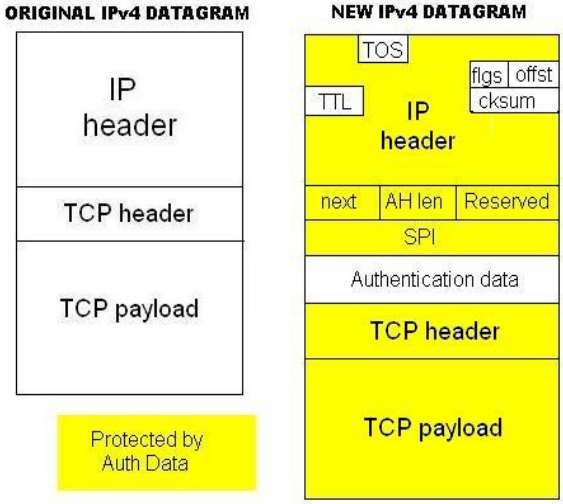

Figure 2. AH packet in transport mode

ESP assures confidentiality through encryption with some of the negotiated symmetric cryptographic algorithm. Encryption includes IP payload, padding and "Next header" field (Figure 3). Authentication covers less fields then AH – only ESP header and IP payload.

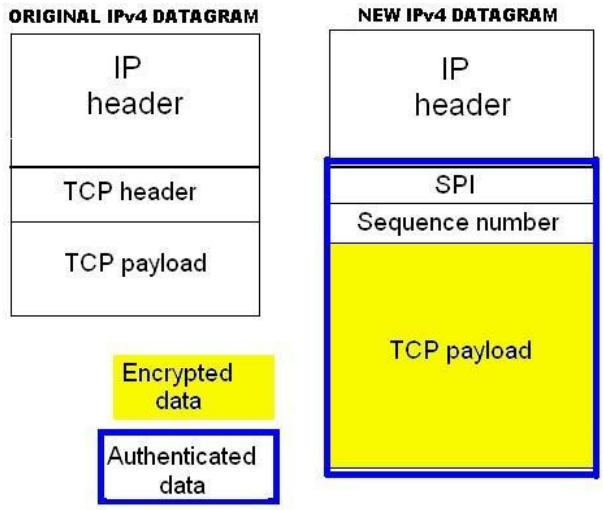

Figure 3. ESP packet in transport mode

There are two IPsec modes: transport mode and tunnel mode. Transport mode is appropriate for usage when communication is end-to-end. In this mode, we have only one source and destination IPv4 address, which are in AH protected by ICV. This leads to problems with NAT, as described later. ESP doesn't have this problems because his integrity check doesn't cover IPv4 header where are these addresses situated. Tunnel mode is better where communication takes place between security gateways. In this case communication is maintained within the IPsec tunnel. This leads to another pair of IP addresses and therefore, to another header besides the original one: "outer" IP header. ESP encryption now covers whole IPv4 datagram, with inner header also. AH authentication checks integrity of the both inner and outer IPv4 header, and off course, IPv4 payload. Integrity check successfully reveals attempts of packet change by intruder on insecure network.

AH and ESP require cryptographic keys to be in SA database. Though possible manual key management isn't particularly secure and doesn't scale well. These problems are solved by automatic key exchange, specifically by Internet Key Exchange version 2 (IKEv2) protocol. Daemon, which runs IKEv2 protocol, generates symmetric keys and does rekeying after some period. Authentication in IPsec is also performed by the IKEv2 protocol using either pre-shared keys, digital certificates or EAP.

IKEv2 messages are transfered via UDP protocol in pairs, requests and response. Each pair is known as *exchange*. Communication between two IKEv2 entities is established via two exchanges. First, those entities, called Initiator and Responder, negotiate the cryptographic suite (crypto algorithms and other parameters important for traffic protection), exchange nonce numbers and perform Diffie-Helman exchange in the IKE_SA_INIT phase. This exchange provides for protection of all the subsequent exchanges. The second exchange, IKE_AUTH, provides mutual authentication for peers, validates the previous pair of messages (IKE_AUTH) and also establishes first SA. Establishment of SA includes traffic selectors and cryptographic algorithms to use for data protection.

IV. Interrelation of IPsec and NAT devices

There are two design possibilities with the respect to interrelation of IPsec and NAT devices. The first one is for IPsec protocols to completely ignore NAT, while the other one is to introduce mechanisms in the protocol that will allow IPsec compliant devices to communicate in spite of NAT devices. As it was already mentioned, NAT boxes are so widespread on today's Internet and there are so many networks behind some kind of a NAT gateway that it's not possible to ignore them. So, some mechanisms to alleviate NATs were introduced. Still, it's not possible to come to complete solution.

Incompatibilities between NAT and IPsec can be divided into the following three categories:

1) intrinsic NAT incompatibilities,
2) NAT implementation weaknesses, and
3) helper incompatibilities.

In the following subsections we describe each incompatibility in more detail and the we list possible solutions.

A. Intrinsic NAT incompatibilities

This group of incompatibilities derives directly from the NAT functionality and therefore will be present whenever there is any kind of NAT device.

First, we have incompatibility between IPsec AH and NAT. This is because AH header incorporates the IP source and destination addresses in the keyed message integrity check and therefore NAT will invalidate message integrity check by changing address fields. This makes AH unusable in the presence of NAT devices. ESP doesn't have this problem because it doesn't incorporate the IP header in its keyed message integrity check, so changes in IP header will not conflict with ESP's integrity protection.

A more serious problem that affects both AH and ESP is incompatibility between transport layer checksums and NAT. This problem occurs with AH and ESP in transport mode. Namely, TCP and UDP checksums depend on source and destination addresses and ports because of their inclusion in "pseudo-header" during transport protocol checksum calculations and verifications. When a NAT box changes source and/or destination address/port it can not change the checksum since it's protected by either or both ESP and AH. The partial solution is to use ESP in tunnel mode, but in this case inner IP address, protected by NAT, is visible to the node node.

There is also incompatibility between IKEv2 address identifiers and NAT. Some phases of IKEv2 use IP addresses as identifiers. Modification of these addresses through NAT will cause mismatch between identifiers and the addresses in the IP header. The solution to this problem might be the use of user IDs or FQDNs that are independent of IP addresses.

IKE protocol requires use of fixed source and destination ports. This also causes incompatibility with NAT. One possible scenario that might cause problems is when multiple hosts behind the NAT (NAPT to be precise) initiate IKE SAs to the same responder. Mechanism is needed to allow the NAPT to demultiplex the incoming IKE packets from responder. This is typically accomplished by translating the IKE UDP source port on outbound packets from initiator. Because of that,

responders must be able to accept IKE traffic from a various source ports (other than 500 on which is IKE running), and must reply to that port.

The somewhat similar problem, but in it's own category, are the incompatibilities between overlapping SPD entries when using NAT. There are situations when responder could send packets down the wrong IPsec SA because of overlapping SPD entries. This could happen when initiating hosts behind NAT use their source IP addresses in Phase 2 identifiers.

Another problem between ESP and NAT lies in the incompatibility between IPsec SPI selection and NAT. Since IPsec ESP traffic is encrypted, only way for NATs to know how to demultiplex incoming packets is to inspect informations in the IP and ESP header. These informations are usually destination IP address, IPsec SPI and security protocol (every SA is uniquely determined by this three elements). Because of independent selection of incoming and outgoing SPIs it is possible that the NAT will deliver the incoming IPsec packets to the wrong place when two hosts behind the NAT attempt to create IPsec SAs at the same destination simultaneously. Namely, this destination could choose same SPIs and there is no way for NAT to know to which SA belong incoming packets from this destination. Thus, NAT neither knows where to deliver these incoming packets. Technique that can alleviate this problem is that receiving host (the one who generated same SPIs in our case) allocate a unique SPI to each unicast SA.

We have already mentioned that protocols who embed IP address within payload have a problem with NAT boxes. IPsec and its protocols aren't exception – payload could be integrity protected and NAT couldn't translate IP addresses embedded within payload. Solution for this problem is to install ALGs (Application Layer Gateways) on the host or security gateway. These ALGs should have a possibility of operating on application traffic before IPsec encapsulation and after IPsec decapsulation.

B. NAT implementation weaknesses

These incompatibilities are not intrinsic to NAT, but are never the less present in many current NAT implementations. This makes this implementation weaknesses something that has to be dealt with.

One of those weaknesses is that some NATs discard any traffic which is not UDP or TCP. Obviously, ESP or AH traffic cannot pass through such NATs.

NAT's behavior in the absence of traffic could be configured in many different ways. One way, which is fatal for IPsec, is that UDP mapping is too early removed when no traffic passes through the NAT.

Although most NATs can properly fragment outgoing IP packets when that is necessary, translation of packets that are already fragmented is difficult and many NATs fail in doing this. Specific issue in this situation is identifier collision because fragment identifiers can overlap when two hosts originate fragmented packets to the same destination. To address this issue, NATs could support identifier translation, but unfortunately not many of them support this kind of protection against identifier collision. Nevertheless, when NAT perform the fragmentation, collision is not a problem since the fragment identifier needs only be unique within source/destination address pair.

Except with outgoing, NATs have also problem with incoming IP packets when they arrive fragmented. Namely, headers often may be split between fragments and fragments could arrive reordered. Hence, NATs should have a possibility of reassembly before completing the translation. Again, not many of them have these features.

C. Helper incompatibilities

These issues are ironically present only in NAT devices which attempt to provide "helper" functionality for IPsec NAT traversal.

There is a certain number of NATs that attempt to use SPI values in header of IKEv2 to demultiplex incoming IKE traffic, which results in problems with rekeying. Problems arise because there is a small probability that would be used same SPI values as earlier.

IKE daemon is usually running at UDP port 500. Some IKE implementations are not able to handle UDP ports other than 500 which hence forces some NATs not to translate packets with a UDP source port of 500. Result is obvious – these NATs are limited to one IPsec client. One solution to this problem is inspection of the ISAKMP header to examine cookies but this leads to the problem explained above.

D. Existing solutions

Some solutions are already mentioned together with problems, other are listed below. Specifically, these solutions are *IPsecTunnelMode* (i), *RSIP* (ii) and *6to4* (iii).

i) IPsec Tunnel Mode

IPsec tunnel mode implementation is possible to traverse NAT successfully in a limited set of circumstances. Unfortunately, the requirements for successful traversal are limited, so more general solution is needed:

- IPsec ESP
- no address validation
- "any to any" SPD entries
- single client operation
- no fragmentation
- active sessions

ESP tunnels: they do not cover the outer IP header within the message integrity check and therefore will not suffer Authentication Data invalidation due to address translation.

No address validation: incompatibilities between IKE identifiers and source addresses will not be detected.

"Any to Any" SPD entries: these entries are not invalidated by address translation.

Single client operation: with only a single client behind NAT there is no risk of overlapping SPDs and also no risk of re-key mis-translation or improper SPI or cookie demultiplexing.

No fragmentation: when certificate authentication is used, IKE fragmentations can be encountered. With pre-shared keys used for authentication fragmentation is less likely.

Active sessions: most VPN sessions typically maintain ongoing traffic flow during their lifetime so that UDP port mappings are unlike to be removed due to inactivity.

ii) RSIP

RSIP, described in [6] and [7] includes mechanisms for IPsec traversal. Issues of IPsec SPI demultiplexing as well as SPD overlap are addressed by enabling host-NAT communication. Its usage is appropriate for enterprises as well as home networking scenarios. This approach assures interoperability with protocols carrying embedded IP addresses because it enables hosts behind a NAT to share the external IP address of NAT (the RSIP gateway).

Changes to the IKE and IPsec protocols are avoided by tunneling IKE and IPsec packets. This assures compatibility with all existing protocols (AH,ESP) and modes (transport and tunnel).

RSIP puts up special requirement in order to successfully handle demultiplexing of IKE rekeys. IKE source port should be floating as well as rekeying to the floated port. Therefore compatibility with existing IPsec implementations is not assured.

iii) 6to4

In this approach, described in [8], the NAT provides IPv6 nodes with specific IPv6 prefix. This prefix is retrieved from the NAT external IPv4 address. Apart from providing the prefix, NAT also encapsulates IPv6 packets in IPv4 for transport to other 6to4 nodes or relays. This method gives a possibility for communication without problems between an IPv6 node using IPsec and other nodes inside the IPv6 or 6to4 area.

Unfortunately, 6to4 is not universally usable. It is an appropriate solution where a single NAT separates a client and the VPN gateway but it cannot be used where multiple NATs are deployed between client and VPN gateway. Reason for this is that forming of mentioned IPv6 prefix requires the assignment of a routable IPv4 address to the NAT.

Peers that support IPv6 need a little upgrade to be 6to4 compatible, while NATs require many extensions to support 6to4.

V. IKEv2 and NAT

NAT in its original form was designed to be transparent for end nodes. Nodes behind the NAT just like the others who are on the Internet don't need any modification to communicate through NAT. Problem arises when IP address is encapsulated inside the payload of the packet because NAT shouldn't have knowledge about this payload (it would be violation of independence between network layers and it often results with more problems).

As we have seen before, opening an IPsec connection through NAT can cause a lot of problems. Communicating in transport mode, checksums fail because of change of IP address and correction is not possible because checksums are cryptographically protected. In tunnel mode there are routing problems. For those reasons IKEv2 can negotiate UDP encapsulation of IKE and ESP packets. Although this encoding is little less efficient, it is easier for NATs to process.

Ports may be modified as the packets pass through NAT devices. Thus, even though IKEv2 packets must be sent from and to UDP port 500, they must be accepted coming from any port. Also, responses must be sent to the port from whence they came. NATs could not transparently modify IKE addresses included in payload because the payload is cryptographically protected and therefore IP addresses of the end nodes are generally not included in the IKE payload.

We have seen that some NATs try to handle cleverly IKE traffic on port 500 so it is better to use some other port to pass IKE packets when traffic is flowing through a NAT. This port was selected to be 4500 and IKEv2 must also listen on port 4500 and that port is reserved for UDP-encapsulated ESP and IKE.

E. NAT detection

Among other payloads important in first phase of creating IKE SA, there are two which are used for NAT detection. They are Notify payloads of type NAT_DETECTION_SOURCE_IP and NAT_DETECTION_DESTINATION_IP. Except the detection whether is NAT between the peers, they are also used to determine which peer is behind the NAT. Content of these payloads is hash of initiator and responder SPI, source/destination IP address and source/destination port (depending on type of payload, it is used source or destination port and address). There could be more than one NAT_DETECTION_SOURCE_IP Notify payload if the sender of the packet does not know his own IP address (in the case of multiple interfaces for instance). If the initiator and responder instructed to initiate NAT traversal, then the following sequence of steps takes place during IKE_SA_INIT and IKE_AUTH exchanges:

i) Prepare hashes for other side and include NOTIFY payloads of type NAT_DETECTION_SOURCE_IP and NAT_DETECTION_DESTINATION_IP in IKE_SA_INIT packets. IKEv2 defines state machine and these actions take place in IKE_SMI_INIT state (initiator) and IKE_SMR_INIT state (responder).

ii) Compare hashes obtained from other side with your own hashes. In case of a couple of NAT_DETECTION_SOURCE_IP NOTIFY payloads, NAT is detected if and only if none of the hashes match and in that case other side is behind the NAT. If the NAT_DETECTION_DESTINATION_IP payload does not match the hash of SPIs and destination IP address and port found from the IP header of the packet containing the payload it means that this end is behind a NAT. In that case this end starts sending keepalive packets as described later. Complete IKE_SA_INIT exchange, when NAT is included, is shown in Figure 4.

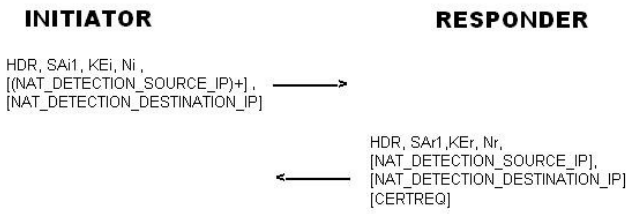

Figure 4. IKE_SA_INIT with NAT included

IKEv2 initiator checks whether these payloads are present and if they do not match the addresses in the outer packet it tunnels all future IKE and ESP packets associated with this IKE_SA over UDP port 4500. To distinguish ESP and IKE messages, there are observed four octets following the UDP header. In case of IKE packet, they are all zeros and in the case of ESP packet on that place is SPI which is never zero.

F. Keepalive packets

Purpose of sending NAT-keepalive packets is to keep NAT mappings alive for the duration of a connection between the peers. The receiver ignores a received NAT-keepalive packet. Structure of keepalive packet is shown in Figure 5.

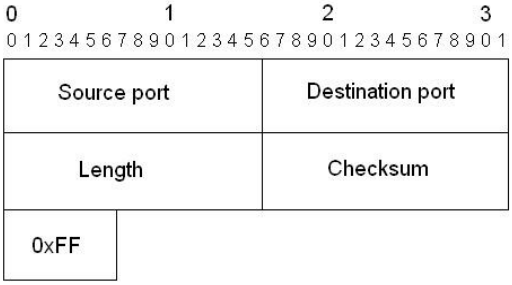

Figure 5. Structure of NAT-keepalive packet

G. Additional remarks

Transport mode TCP and UDP packet checksum fixup is required. For this purpose the original source and destination IP address are needed and they are obtained from the Traffic Selectors associated with the exchange. Traffic Selectors define IP addresses, ports and protocols which will be transmitted through some CHILD SA. CHILD SA is a security association which comes after IKE SA. In the case of NAT traversal, the Traffic Selectors contain exactly one IP address which is then used as the original IP address.

In some special cases, NAT box does the removal of still valid mappings (for instance, when NAT box is rebooted). In these cases hosts that are not behind a NAT send all packets to the IP address and port retrieved from the last valid and authenticated packet received by other peer. Host behind a NAT does not do this because of the danger of DoS (Denial of Service) attack.

VI. CONCLUSION

NAT is a mechanism which brought temporary rescue to the problem of shortage of IPv4 addresses. Unfortunately, it also brought some problems which we solved as described. IKEv2 with its state machine is a very complex system and it was necessary to carefully select states for NAT detection and traversal. Security is an essence part of this protocol and therefore hashes were used in procedure of NAT detection. After detection of NAT box, there are appropriate actions as described.

Support of NAT traversal in IKEv2 implementation solved one of the important demands for IKEv2 implementations described in [1] and made this implementation more general and therefore, more appropriate to use in the IPsec.

VII. Acknowledgment

This work has been carried out within projects 03603619941995 Universal Middleware Platform for Intelligent eLearning Systems funded by the Ministry of Science and Technology of the Republic of Croatia, and IKEv2 Step2 project funded by Siemens Networks.

LITERATURE

[1] C. Kaufman, Ed., *Internet Key Exchange Key (IKEv2) Protocol*, 2005., URL: http://www.ietf.org/rfc/rfc4306.txt

[2] Kent S, and K.Seo, *Security Architecture for Internet Protocol*, RFC 4301, December 2005.

[3] A.Kukec, *Introduction to IKEv2 and IPsec*, March 2006. URL: http://os2.zemris.fer.hr/ns/2006_kukec/

[4] Aboba, B. and W. Dixon, *IPsec-Network Address Translation (NAT) Compatibility Requirements*, RFC 3715, March 2004.

[5] Kivinen, T., *Negotiation of NAT-Traversal in the IKE*, RFC 3947, January 2005.

[6] Borella, M., Grabelsky, D., Lo, J. and K. Taniguchi, *Realm Specific IP: Protocol Specification*, RFC 3103, October 2001.

[7] Borella, M., Lo, J., Grabelsky, D. and G. Montenegro, *Realm Specific IP: Framework*, RFC 3102, October 2001.

[8] Carpenter, B. and K. Moore, *Connection of IPv6 Domains via IPv4 Clouds*, RFC 3056, February 2001.

[9] Egevang, K. and P.Francis, *The IP Network Address Translator (NAT)*, RFC 1631, May 1994.

[10] Holdrege, M. and P.Srisuresh, *Protocol Complications with the IP Network Address Translator,* RFC 3027, January 2001.

[11] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, *UDP Encapsulation of IPsec Packets*, RFC 3948, January 2005.

[12] Kent, S., *IP Encapsulating Security Payload (ESP)*, RFC 4303, December 2005.

[13] Kent, S., *IP Authentication Header*, RFC 4302, December 2005.

[14] D. Shinder, *Comparing VPN options*, 2004.