TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

CENTRAL CAMPUS PULCHOWK

THESIS NO.: 069MSCS663

Authenticity Based on Physically Unclonable Function in IPv6 Platform

By

Roshan Acharya

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SYSTEM AND KNOWLEDGE ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

NOVEMBER, 2014

Authenticity Based on Physically Unclonable Function in IPv6 Platform

By

Roshan Acharya

(069/MSCS/663)

Thesis supervisor:

Mr. Babu Ram Dawadi

Deputy Head of Electronics and Computer Engineering

Institute of Engineering, Nepal

A thesis submitted in partial fulfillment of the requirements for the

degree of Master of Science in Computer System and Knowledge Engineering

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

Tribhuvan University

Lalitpur, Nepal

November, 2014

COPYRIGHT

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

CENTRAL CAMPUS PULCHOWK

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certify that it has been read and recommended to the Department of Electronics and Computer Engineering, for acceptance, a thesis entitled "Authenticity based on Physically Unclonable Function in IPv6 Platform", submitted by "Mr. Roshan Acharya" in partial fulfillment of the requirement for the award of the degree of "Master of Science in Computer System and Knowledge Engineering".

_____

Supervisor, Babu Ram Dawadi
Deputy Head of Electronics and Computer Engineering

_____

External Examiner, Saroj Shakya
Associate Professor
Nepal College of Information Technology

_____

Committee Chairperson, Dr. Shashidhar Ram Joshi
Professor
Department of Electronics and Computer Engineering

_____

Date

DEPARTMENTAL ACCEPTANCE

The thesis entitled "Authenticity Based on Physically Unclonable Function in IPv6 Platform", submitted by Roshan Acharya in partial fulfillment of the requirement for the award of the degree of "Master of Science in Computer System and Knowledge Engineering" has been accepted as a bonafide record of work independently carried out by him in the department.

_____

Dr. Dibakar Raj Pant

Assistant Professor and Head of the Department

Department of Electronics and Computer Engineering

Central Campus Pulchowk

Institute of Engineering

Tribhuvan University

Lalitpur, Nepal.

ABSTRACT

Computer networks were primarily used by the University researchers for sending emails and by corporate employee for sharing printers. Under these conditions, security did not get a lot of attention. But now, as millions of ordinary citizens are using networks for financial transaction, social networking so there is huge potential threats around network security. Modern cryptographic protocols are based on the premise that only authorized participants can obtain secrets keys and access to information systems. However, various kinds of tampering methods have been devised to extract secret information from smartcards and ATMs. From storage of digital secret key in a chip we came to an idea to use physical property of nonhomogeneous material that make it unique. This property make a secret key unclonable and due to this, the structure created from this material are called physical unclonable function. Moreover our current IP Addressing in IPV4 somewhat addresses security issues however due to limited IP address, it always acting as a hindrance to accommodate most of the users in the earth. So, for the sufficient IP address resources and providing better quality of service with secure services, the technology in networking field should be migrated to IPv6.

Keywords: cryptographic, protocol, nonhomogeneous, tampering, unclonable.

ACKNOWLEDGEMENT

It would not have been possible to write this master thesis without the help and support of the kind people around me. I would like to express my deep sense of gratitude to MSCSKE Program Coordinator Dr. Sanjeeb Prasad Panday for granting me permission in carrying my thesis.

I would like to thank Head of Department Assistant Professor Dr. Dibakar Raj Pant, Deputy Head Babu Ram Dawadi, Professor Dr. Shashidhar Ram Joshi, Professor Dr. Subarna Shakya, Dr. Aman Shakya, Dr. Arun Timilsina all the faculty members and friends for their kind support and their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice.

This thesis would not have been possible without the help, support and patience of my supervisor, Babu Ram Dawadi, who has helped me in transforming my ideas and guiding properly to achieve my goal. I would like to express my deepest gratitude to him for his excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research.

Finally, I would like to thank Mr. Kshitiz Shrestha who as a good friend, is always willing to help and give his best suggestions and help me desperately to complete my thesis.

.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CRP Challenge Response Pair

HTTP Hypertext Transfer Protocol

IANA Internet Assigned Number Authority

IP Internet Protocol

IPng Internet Protocol Next Generation

IPv6 Internet Protocol Version 6

IPsec Internet Protocol Security

ISP Internet Service Provider

LIRs Local Internet Registries

MAC Media Access Control

NAT Network Address Translation

PHP PHP Hypertext Processor

PUF Physically Unclonable Function

RFC Request for comment

RIR Regional Internet Registries

TCP Transmission Control Protocol

CHAPTER ONE: INTRODUCTION

1.1 Background

Network security consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. It does as its title explains: It secures the network, as well as protecting and overseeing operations being done. The most common and simple way of protecting a network resource is by assigning it a unique name and a corresponding password. However it is found that password is often cracked and hacked which lead to serious damage in enterprises reputation. So instead of using only username and password to access critical system, it is often touted to authenticate end devices or using biometric finger's print.

1.2 Internet Protocol Version 6 (IPv6)

The prevailing Internet Protocol standard is IPv4 (Internet Protocol version 4), which dates back to the 1970s. There are well-known limitations of IPv4, including the limited IP address space and lack of security. IPv4 specifies a 32-bit IP address field, and available address spaces are rapidly running out. The only security feature provided in IPv4 is a security option field that provides a way for hosts to send security and handling restrictions parameters [1].

As a result, the Internet Engineering Task Force (IETF) has been working on the IPv6 (Internet Protocol version 6) specifications in order to address these limitations, along with a number of performance, ease-of-configuration, and network management issues.

IPv6 stands for Internet Protocol version 6 also known as IPng (IP next generation) is the second version of the Internet Protocol to be used generally across the virtual world. The first version was IPv4. IPng was designed to take an evolutionary step from IPv4. It was not a design goal to take a radical step away from IPv4. Functions which work in IPv4 were kept in IPng. Functions which didn't work were removed.

Like IPv4, IPv6 is an internet-layer protocol for packet- switched internetworking and provides end-to-end datagram transmission across multiple IP networks. While IPv4 allows 32 bits for an IP address, and therefore has $2^{\wedge 32}$ (about 4.3 billion) possible addresses, IPv6 uses 128-bit addresses, for an address space of $2^{\wedge 128}$ addresses. This expansion allows for many more devices and users on the internet as well as extra flexibility in allocating addresses and efficiency for routing traffic. It also eliminates the primary need for network address translation (NAT), which gained widespread deployment as an effort to alleviate IPv4 address exhaustion [2].

The header format of IPv6 is given below

Bits 4 8 32

Version (4 bits)	Class (4bits)	Flow label(24 bits)	
Payload length (16 bits)		Next header (8 bits)	Hop Limit (8 bits)
Source Address (128 bits)			
Destination Address (128 bits)			
Data			

Figure1.1 IPv6 Header Format

Total length: 40bytes

1.3 IPv6 Features

The feature which IPv6 protocol brings to plate are described in several RFCs and internet drafts could be summarized as follows:

i. New header format
ii. Large address space
iii. Efficient & Hierarchical addressing and routing infrastructure
iv. Stateless and stateful address configuration
v. Security
vi. Better Quality of Service Support
vii. New protocol for neighboring node interaction
viii. Extensibility

1.4 IPv6 Addressing

IPv6 addresses are 128-bit identifiers and sets of interfaces. IPv6 addresses are mainly categorized into three types:

i. **Unicast Address**: an identifier for a single interface. Unicast addresses are regular addresses used for one-to-one communication. A packet sent to a unicast address is delivered to the interface identified by that address.
ii. **Anycast Address**: An identifier for set of interfaces (typically belonging to different nodes). A packet sent to an Anycast address is delivered to one of the nearest interface identified by that address.
iii. **Multicast Address**: An identifier for a set of interfaces (typically belonging to different nodes). Multicast addresses are group addresses; packets sent to such an address are delivered to all the interfaces that are interested and have joined the group. All functions that were performed by broadcasts in IPv4 are performed by using multicast in IPv6.

1.4.1 Address Notation

The 128 bits IPv6 address is divided into 8 blocks each consists of 16-bits represented in hex and each block is separated with colon (:). For example:

2001:0d30:0000:00ef:0020:0000:0000:df3d. The block whose values are zero can be compressed using a double colon (::) to simplify the address notation with a limitation that there will be no more than double colon in an address. The following table shows the correct and incorrect address notation.

No	Notation	Correct	Remarks
1	2001:0d30:0000:00ef:0020:0000:0000:df3d	Yes	
2	2001:d30:0:ef:20:0:0:df3d	Yes	
3	2001:d30:0:ef:20::df3d	Yes	
4	2001:d30::ef:20::df3d	No	Two (::) are not allowed
5	2001:d30::ef:20:0:0:df3d	No	(::) set only to larger zeros

Table 1.1 IPv6 address Notation

IPv6 also uses prefixes to identify subnets and routes, as in IPv4 CIDR. An IPv6 prefix address is written as IPv6-address/prefix-length. If the address in the previous example has a route prefix with length of 48 bits, the prefix is 2001:d30::/48.

1.4.2 Address Type Identification

IPv6 address can be categorized in to different types according to their high order bits.

IPv6 Notation	Binary Prefix	Address Type
::/128	0000…..000 (128bits)	Unspecified
::1/128	0000…..001 (128bits)	Loopback Address
FF00::/8	11111111	Multicast
FE80::/10	1111111010	Link Local Unicast
~~FEC0::/10~~	~~1111111011~~	~~Site Local Unicast~~
FC00::/7	1111111	Unique Local Addresses
Else	Global	Global Unicast Addresses

Table 1.2 IPv6 address types

1.4.3 Unicast Address

The general address format for IPv6 global unicast addresses [RFC 3587] is as follows.

n bits	64-n bits	64 bits
Global Routing Prefix	Subnet ID	Interface ID

Table 1.3 Global unicast address format

Where the routing prefix is a value assigned to identify a site (a cluster of subnets/links), the subnet ID is an identifier of a subnet within the site, and the interface ID is a modified EUI-64 format.

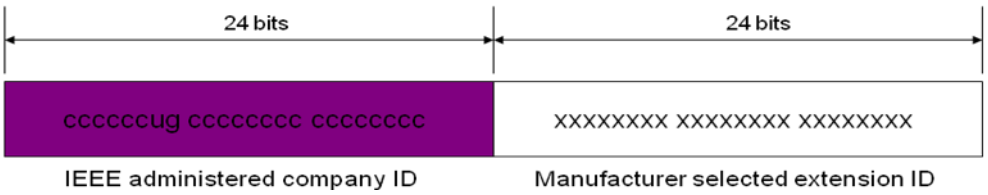
1.4.4 Interface Identifier

Interface identifiers in IPv6 unicast addresses are used to identify interfaces on a link. They are required to be unique within a subnet prefix. It is recommended that the same interface identifier cannot be assigned to different nodes on a link. They may also be unique over a broader scope. In some cases, an interface's identifier will be derived directly from that interface's link-layer address. The same interface identifier may be used on multiple interfaces on a single node, as long as they are attached to different subnets.

For all unicast addresses, except those that start with binary value 000, Interface IDs are required to be 64 bits long and to be constructed in Modified EUI-64 format. Global unicast addresses that start with binary 000 have no such constraint on the size or structure of the interface ID field. Modified EUI-64 format based Interface identifiers may have global scope when derived from a global token (e.g., IEEE 802 48-bit MAC). The modified EUI-64 format of a MAC address is constructed by complementing the second LSB of the first bite of MAC address and inserting 0XFFFE between the third and fourth bytes of the MAC address [3].

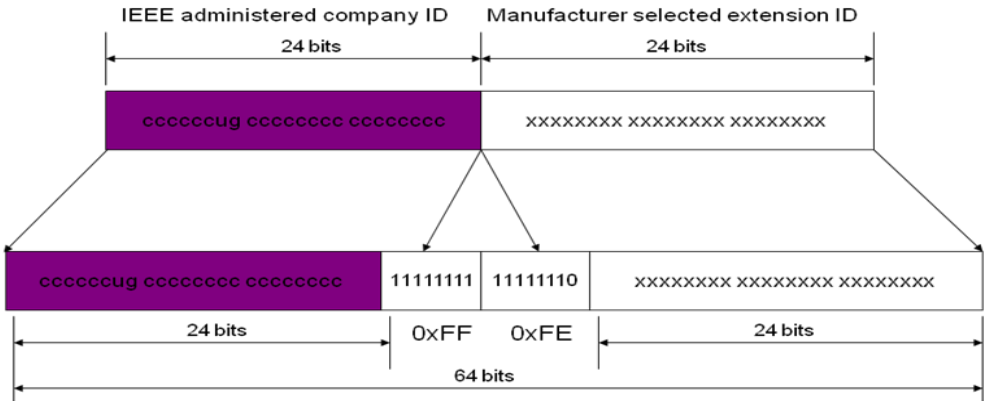
Steps to create modified EUI-64 format and IPv6 interface identifier

i. Let us take IEEE 802 48 bit MAC address which is assigned to each NIC in which the first 24 bits is assigned to vendor and second 24 bits is assigned by vendor.

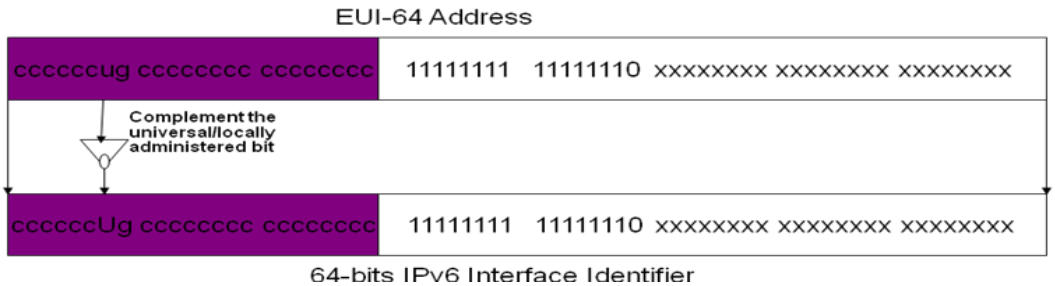


ii. But the IEEE EUI-64 format consists of 64 bits looks like the following.

iii. Now map the MAC address to EUI-64 format by inserting 0xFFFE between third and fourth bytes of the MAC address.



iv. Complement second LSB of first byte to make it IPv6 interface identifiers



v. If an IPv6 address has route prefix of 48 bits. For example 2001:D30::/48 and a PC has NIC with MAC address 00-0D-60-77-DC-04, then its IPv6 address would

be 2001:D30::20D:60FF:FE77:DC04 and the link local address would be FE80::20D:60FF:FE77:DC04

1.4.5 IPv6 Address with Embedded IPv4 Address

To dynamically tunnel IPv6 packets over IPv4 routing infrastructure, a transition mechanism has been developed in which the IPv6 nodes that use this technique are assigned special IPv6 unicast address that carry a global IPv4 address in the low order 32 bits termed as IPv4 compatible IPv6 address.

80 bits	16 bits	32 bits
0000….0000	0000	IPv4 address

Table 1.4 IPv4 Compatible IPv6 global unicast address

A second type of IPv6 addresses which holds an embedded IPv4 address is also defined. This address type is used to represent the addresses of IPv4 nodes as IPv6 addresses. This type of address is termed as "IPv4-mapped IPv6 address" and has the format:

80 bits	16 bits	32 bits
0000….0000	FFFF	IPv4 address

Table 1.5 IPv4-mapped IPv6 address

1.4.6 Multicast Address

IPv6 Multicast address has the following format [RFC 3513]:

8	4	4	112 bits
11111111	Flags	scope	Group ID

Table 1.6 IPv6 multicast address format

Any address starts with FF can be taken as multicast address. Because the multicast address/prefix is defined as FF00::/8. Flags consist of four values:

0	0	0	T

The high-order 3 flags are reserved, and must be initialized to 0. T = 0 indicates a permanently-assigned "well-known" multicast address, assigned by the Internet Assigned Number Authority (IANA). T = 1 indicates a non-permanently-assigned ("transient") multicast address. Scope is a 4-bit multicast scope value used to limit the scope of the multicast group.

It is necessary to limit the propagation of multicast packets. For instance it wouldn't be good if all routers connected to the internet were to receive all the hello packets the OSPF routers use to find their neighbors. These packets are for use on the local subnet only. Restriction on the propagation of multicast packets are encoded in the multicast address in the form of a 4-bit scope value listed below.

Value (binary)	**Value (hex)**	**Scope**
0000	0	Reserved
0001	1	Interface local (for the transmission of loopback multicast packets)
0010	2	Link Local
0011	3	Reserved
0100	4	Admin Local
0101	5	Site Local
1000	8	Organizational Local
1110	E	Global
1111	F	Reserved

Table 1.7 IPv6 Multicast Scope

The remaining scope values 6, 7 & 9 to C may be used by network administrators to define additional scopes where necessary. Interface-local scope spans only a single interface on a node, and is useful only for loopback transmission of multicast. Link-local and site-local multicast scopes span the same topological regions as the corresponding unicast scopes. Admin-local scope is the smallest scope that must be administratively configured, i.e., not automatically derived from physical connectivity or other, non-

multicast-related configuration. Organization-local scope is intended to span multiple sites belonging to a single organization.

Some well-known multicast address:

Address	Description	scope
FF01::1	All nodes addresses	Interface-local
FF01::2	All routers addresses	Interface-local
FF02::1	All nodes addresses	Link-local
FF02::2	All routers addresses	Link-local
FF02::4	DVMRP Routers	Link-local
FF02::5	OSPFIGP	Link-local
FF02::D	All PIM routers	Link-local
FF05::2	All routers addresses	Site-local

Table 1.8 Some Well Known Multicast Addresses

1.4.7 Anycast Address

Anycast addresses are allocated from the unicast address space, using any of the defined unicast address formats. Thus, anycast addresses are syntactically indistinguishable from unicast addresses. When a unicast address is assigned to more than one interface, thus turning it into an anycast address, the nodes to which the address is assigned must be explicitly configured to know that it is an anycast address. One expected use of anycast addresses is to identify the set of routers belonging to an organization providing internet service. Such addresses could be used as intermediate addresses in an IPv6 Routing header, to cause a packet to be delivered via a particular service provider or sequence of service providers. Some other possible uses are to identify the set of routers attached to a particular subnet, or the set of routers providing entry into a particular routing domain. The following restrictions are imposed on IPv6 anycast addresses:

i. An anycast address must not be used as the source address of an IPv6 packet.
ii. An anycast address must not be assigned to an IPv6 host, that is, it may be assigned to an IPv6 router only.

The Subnet-Router anycast address is predefined. Its format is as follows:

n bits	128-n bits
Subnet prefix	00000…0000

Table 1.9 Anycast Address Format [RFC 3513]

The "subnet prefix" in an anycast address is the prefix which identifies a specific link. This anycast address is syntactically the same as a unicast address for an interface on the link with the interface identifier set to zero. Packets sent to the Subnet-Router anycast address will be delivered to one router on the subnet. All routers are required to support the Subnet-Router anycast addresses for the subnets to which they have interfaces.

1.5 IPv6 Address Distribution

An address allocation is defined by allocation size and location. Allocation size specifies how large of the address block or prefix is assigned. Allocation location is where in the address pool this block is allocated to [4].

The IP address allocation hierarchy is shown in figure below. At the top of the hierarchy, the whole address pool is controlled by the Internet Assigned Number Authority (IANA). IANA allocates large address blocks to each the Regional Internet Registries (RIR) serving North America (ARIN), Europe (RIPE), Asia Pacific (APNIC), Africa (AfriNIC) and Latin America & Caribbean (LACNIC). The regional registries divide up these large address blocks into medium blocks to allocate to Local Internet Registries (LIRs), consisting mainly of internet service providers (ISPs). The ISPs further assign smaller address blocks to their users including companies, universities and smaller ISPs etc.

The policies on IP allocation size vary from different registries at different levels. Different RIRs adapt their own policies for allocation to LIR/ISPs with unit size varying from /10 to /20. The size assign to end users by each ISP also vary accordingly. Due to historical allocation schemes, fragmentation is a common problem in IPv4; one ISP is

often left with multiple prefixes. For the 128 IPv6 address, the last 64 bits are assigned to interface ID.

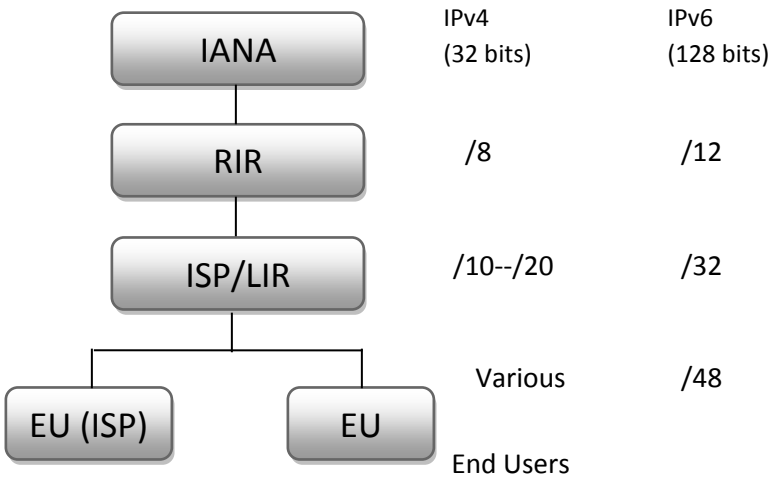


Figure 1.2 Address allocation hierarchy

IPv6 address allocation only considers the top 64 bits. Assigning appropriate address size at different level has been under extensive discussion.

1.6 Security in IPv6

Boarder internet communities must be convinced about the new internet protocol to be deployed. For that we need to deploy on enough systems and used by enough people to show that it is both beneficial and also not a danger to the rest of the network as a whole. Security is the major concern for ISPs after deploying IPv6 successfully.

Extension Headers in IPv6 has two dedicated headers for security, one is Authentication Header (AH) and another is Encrypted Security Payload (ESP). Both the AH and the ESP headers exploit the concept of security association (SA) to agree the security algorithms and parameters between the senders and the receiver. Each IPv6 node manages a set of SAs, one for each secure communication currently active. The Security Parameters Index (SPI) is a parameter contained in both the AH and ESP headers to specify which SA is to be used in decrypting/authenticating packets.

1.6.1 Authentication in IPv6

Authentication alone does not provide all security features but intern, not all application requires all those features. Routing or neighbor discovery information, for instance, is usually not considered a secret, but as a multitude of attacks on the routing system have demonstrated, integrity and authenticity of IP packets carrying routing information would be highly desirable.

The Authentication header is defined under extension header of IPv6 which is identified by value 51 in its next header. The AH header is composed of a 64-bits fixed part followed by a variable number of 32-bit blocks.

Next Header (8)	Payload Length (8)	Reserved (16)
Security Parameters Index (SPI) (32)		
Authenticate Data (Variable Length)		

Table 1.10 Authentication Header

i. **Next Header**: the value for the next type of payload in the daisy chain of headers.
ii. **Payload Length**: the total length of authentication data expressed as a multiple of 32 bits words.
iii. **A Reserved Field** (not used; set to zero) &
iv. **SPI**: indicates which checksum algorithm is to be used.
v. **Authentication Data:** a cryptographically secure checksum over the payload, as well as some fields of the IP and extension headers, concatenated with a shared secret negotiated between the communication partners during the setup of the SA and indexed by the SPI.

The variable part of the AH header is composed of a variable number of 32-bit blocks which contains the actual authentication data. When a destination node receives a packet with an AH header, the packets authority and integrity can be checked by using the procedure as follows:

i. Clear the hop count field

ii. If the packet contains a routing header, then do the following

    a. Set the destination address field the address of the final destination

    b. Set the routing header field to the value that it will have at the final destination

    c. Set the address index field to the value that it will have at the final destination

iii. Clear the all options that have C-bit (Change in route) active.

In-order to protect the packets against deliberate modifications, a reasonable degree of protection can be ensured only by better digest algorithms like MD5 or SHA.

1.6.2 Encrypted Security Payload (ESP)

This header defined under extensions header is identified by the value 52 in next header field for the preceding header. The exact format of the encryption part depends on the encryption algorithm used. The default IPv6 encryption is DES-CBC which is the DES algorithm applied in Cipher Block Chaining (CBC) mode. DES is a private key encryption algorithm that is normally applied to 64-bit data blocks with a 56-bit key. Various techniques have been proposed to apply the DES transformation to blocks bigger than 64-bit blocks and each block is EX-ORed with the result of the previous encryption before being encrypted itself.

1.6.3 IPSEC Framework

IETF IP Security Protocol Working Group (IPSEC) has formally standardized and defined security protocol for the IP protocol layer under RFC 2401. The framework consists of six different layers:

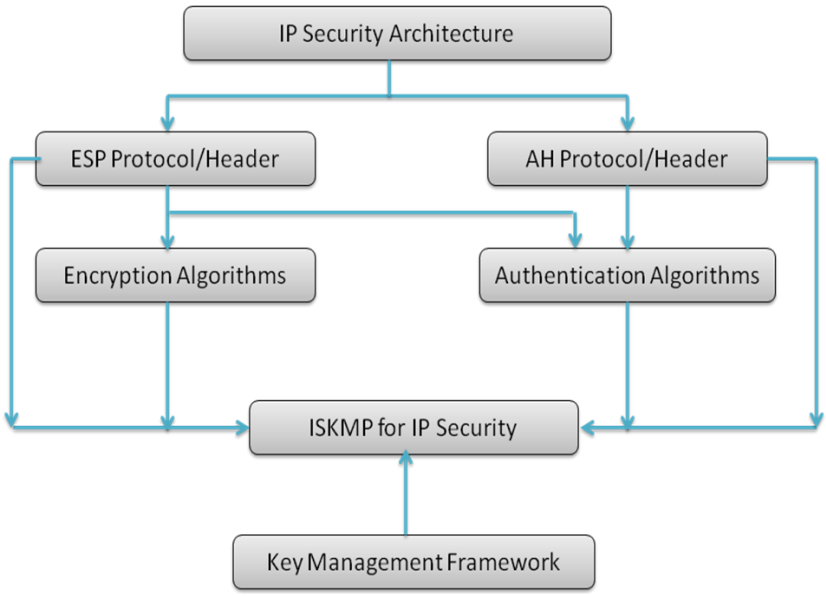


Figure 1.3 IPSEC components (RFC 2411)

It must be noted that these mechanisms are considered generic; they could be used in both IPv4 and IPv6 contexts. They would have to be retrofitted into existing IPv4 software, but they are integrated, mandatory part of the basic IPv6 protocol suite.

1.6.4 IPv6 & Link Local Addresses

One goal of the designers of the IPv6 protocols was to make setting up a system that run the new protocols easier than it was to setup a system running IPv4. IPv6 attempted to do away with the user needing to know any information by specifically including protocols for the auto-configuration of nodes in the network. A node can start running without a globally routable address, and talk to its neighbors to learn about the local network and eventually, with the help of router, automatically bootstrap itself into the global Internet. IPv6 uses link-local addresses for auto-address configuration, neighbor discovery or when there is no router present [2].

Now days, more and more flavors of Linux, BSD and other UNIX operating systems gain IPv6-support in the kernel, it's becoming fairly common for these systems to have link-local connectivity without the owner realizing it. To add insult to injury, existing IP packet filters or software firewalls generally filter only IPv4 and don't get in the way of IPv6 packets. Most systems don't rely on these types of filters to avoid unwanted

connections. While scanning an entire subnet, one address at a time isn't really an option, there are other ways to find IPv6 systems connected to the local subnet. An obvious one is the "broadcast ping" to the all host multicast address.

1.6.5 Internetworking of IPv6 Security with Other Services

The provision of IPSEC in IPv6 is a giant step forward with respect to providing security on the internet. There are a variety of different uses of IPSEC within the basic internet protocol suite, such as general confidentiality of transmission, authentication of peer entities and prevention of DoS and man-in-the-middle attacks. However the provision of IPSEC service also influences the security elements used in application layer protocols. Simple services such as Telnet, FTP, DNS and SNMP-based network management may now rely solely on IPSEC for obtaining sufficient security. Other more complex applications, such as E-Mail require more complex security elements, however, such as non-repudiation of receipt, proof of origin, or specific encryption of information on the application or even user level; these elements are not directly obtainable from IPSEC operating solely on the network layer.

Although these application-level security elements may profit from the provision of IPSEC services, they still need to provide their own security elements, which are not necessarily compatible with the SAs and key exchanges used by IPSEC.

1.6.6 Open issues in IPv6 Security

It seems that some ad-hoc solutions, such as NAT and CIDR, as well as the availability of security elements such as SSL (Standardized by IETF as TLS), SSH, and secure email in IPv4 networks, have delayed the deployment of IPv6. While this delay leaves the internet open for attacks for a longer period of time and leads to additional complexity and management overhead, it has given protocol designers more time to improve their plans. While the base standards for IPSEC are considered stable and will be extended mostly in the area of allowing additional encryption or authentication algorithms, more work remains to be done in the IKE area and in improving protection mechanisms against traffic analysis and denial-of-service/flooding attacks. The full deployment of IPSEC will also largely depend on the availability of a technically, organizationally and

politically acceptable and workable PKI capable of handling a large number of users of software process certificates.

A final issue for IPSEC is the management of its complexity. Many IPSEC mechanisms and their consequences are no longer easy to understand and are thus error-prone in implementation and operation, even though intuitive user interfaces for end-system configuration are available. Adding further functionality, such as various styles of tunneling will also add complexity and thus endanger the original idea of providing simple, ubiquitous security in IPv6.

The security of electronic devices is of crucial importance to companies as well as to users. Moreover, companies that develop Intellectual Property also want to protect them from counterfeiting and overbuilding. Company profits, brand reputations and personal information of the users are at stake if there is a breach in the security of these electronic devices. In the classical approach, a system is secured by storing the cryptographic keys permanently in the non-volatile memories that are present in the security devices. However, this permanent storage of the key makes them easy targets for physical attacks; hence compromising the security of the system. A more secure, cost-effective and elegant solution to this permanent key storage is the use of Physical Unclonable Functions (PUFs).

PUF is a method of producing a signature from a physical object, such as an Integrated Circuit, by relying on the non-reproducible physical attributes of a device. These signatures are unique because fabricated circuits exhibit slightly different electrical behavior from one another even if their design, mask and manufacturing process are identical. Various kinds of PUFs exist; examples are Optical PUF, Butterfly PUF and SRAM PUF. Physical Unclonable Functions (PUFs) are defined as functions based on physical characteristics which are unique for each chip, difficult to predict, easy to evaluate and reliable. These functions should also be individual and practically impossible to duplicate. PUFs can serve as a root of trust and can provide a key which cannot be easily reverse engineered. In principle, any physical device characteristic that fluctuates can be turned into a PUF.

A PUF device provides a unique challenge-response capability. That is, when two PUFs are provided the identical challenge, they will each produce unique responses. In this way, a PUF, and the system it contains, can be identified by the response value it generates to a specific challenge. A more formalized definition of this relationship is given below.

PUF1(C) = R1

PUF2(C) = R2

R1 = R2

This relationship can thus be used to bind certain information to a given system by adding a PUF to it. That is, when a system produces a specific response, it is possible to unique identify that specific system from another [5].

1.7 Types of PUFs

A PUF device provides this sort of relationship by leveraging the physical properties of the materials in which it is instantiated. There are several different ways of doing this, from measuring the distortions of reflected light to leveraging the manufacturing inconsistencies from one chip to another. The Ring Oscillator PUF is presented first and in somewhat greater detail than 20other types of PUF since this is the type of PUF that the author worked with primarily.

1.7.1 Ring Oscillator PUF

A Ring Oscillator PUF is a PUF design that utilizes a circuit called a Ring Oscillator (RO). An RO is an odd number of inverter gates tied together. Because there are an odd number of gates, this will produce a continuously changing, or oscillating, signal. Because it is a combination of circuits, the RO PUF can be instantiated on a piece of silicon, such as an FPGA or ASIC device.

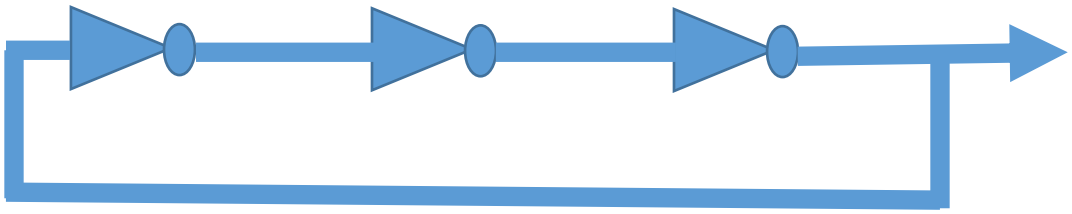


Figure 1.4 A 3-gate ring oscillator

Depending on the number of inverter gates being used as well as the propagation delay of every individual inverter, the output frequency of one RO may be different from another RO. In Figure 1.4, this output signal corresponds to the signal marked Q.

When used as part of a PUF, the unique behavior of an RO will be examined. Consider again the 3 stage RO shown in Figure 1.4. All three inverter gates are assumed to have the same propagation delay and the interconnecting wires are assumed to impose a negligible delay. However, in an actual instantiation of an RO, these assumptions are invalid. All three inverters should have the same propagation delay, but, due to uncontrollable manufacturing inconsistencies and tolerances, they do not. In a similar vein, the interconnecting wires will also impose a non-zero delay time in signal propagation. Both of these factors will combine so that even if two ROs are produced on the same manufacturing line, they will generate a slightly different output frequency. The slightly different output frequencies of two ring oscillators forms the basis of randomness for the Ring Oscillator PUF. Because the output frequencies of the ROs cannot be predicted, their actual frequency at run time gives a way to uniquely identify the individual PUF that contains them.

The ring oscillator PUF shown above uses a challenge bit and feeds it to a multiplexer. If the challenge bit is zero, the top ring oscillator will be fed to the top counter and the bottom ring oscillator to the bottom counter. If the challenge bit is one, the top ring oscillator will be fed to the bottom counter and the bottom ring oscillator will be fed to the top counter. The counters will then be executed for a given amount of time. At the expiration of this time duration, the values of the counters are compared. If the top counter has a larger total, a zero is output as the response.

If the bottom counter has a larger total, a one is output as the response. While the diagram only displays two ring oscillators and only 1 bit of challenge and response, this diagram can be extrapolated to form arbitrarily large PUFs. That is the most basic design of an RO PUF. In practice, this design is somewhat inefficient, since for an N-bit PUF, 2*N ring oscillators are needed, which is fairly expensive. There has been work done for alternative designs of an RO PUF to reduce the number of ring oscillators needed [5]. Typically, this involves using a pool of ring oscillators and then using a multi-bit challenge to select some permutation of them.

1.7.2 **Butterfly PUF**

Another design of a PUF is called a Butterfly PUF. This design is similar to the previous RO design in that it can be instantiated on a piece of silicon. This allows for easy incorporation into existing FPGA designs or through the production of a custom ASIC chip. The Butterfly PUF works by tying the output of two D flip flops to each other's inputs. By applying the CLR signal to one flip flop and the PRE signal to the other flip flop, the circuit will enter an undefined state. It will eventually go to one of two defined states (0 or 1). The circuit will typically settle in the same state, which forms the basis for the PUF response. Interested readers are referred to [6] for more information about the Butterfly PUF.

1.7.3 **Optical PUF**

An optical PUF is a design that leverages physical randomness that is explicitly introduced during a manufacturing process. The typical optical PUF is constructed by taking a transparent material and randomly coating it with particles to disperse the light. A laser light is then shone on the material and the resulting pattern is recorded. The image is then processed and this is the response of the PUF.

1.7.4 **Coating PUF**

A coating PUF works by creating a mesh of wires and then filling the cavities with some sort of dielectric material. Based on how the dielectric is applied, there will be varying levels of capacitance between the wires in the mesh. This capacitance can then be

measured as the unique response for the given PUF design. Details are presented in [7]. The coating PUF is typically used as a sort of anti-tamper device. The coating PUF is wrapped around an existing circuit and is used to enable its operation. That is, the proper PUF response is needed to unlock the device. If an adversary is to alter the coating in anyway (though reverse engineering for example), this will alter the PUF response and cause the underlying circuit to not function.

1.8 Problem Definition

In the real world scenario, there is data transmission over the insecure channel and always exist potential threat that may alter or modify the message contents or data. Moreover if anyone cannot maintain authenticity, there is chance of fraud in real time transaction or communication system. To address the source verification which means to authentic device or users to logon on any system and provide the right access to him/her and to avoid the message modification or alteration, it is seemed to be useful to have system incorporated with physically unclonable function which maintain authenticity avoiding cloning problem in IPv6 platform. Moreover IPv6 addressing is based on MAC and random number generation process which may suffer from MAC cloning or IP spoofing problem. However by the use of PUF in IPv6, it is possible to verify source and can provide access to the authentic devices or users.

1.9 Objective

Source Verification using physically unclonable function.

1.10 Scope of the work

This thesis allow to detect the IP spoofing, MAC cloning, and prevent from unwanted damages in data security. It maintains the privacy and benefits both the sender and receiver and avoid the problem of non-repudiation.

CHAPTER TWO: LITERATURE REVIEW

2.1 Previous work

There are three main method for source address validation: cryptographic authentication, ingress/egress packet filtering and various trackback techniques [3]. Some of them are given below.

2.2 Two-factor authentication

When elements representing two factors are required for authentication, the term two-factor authentication is applied — e.g. a bankcard (something the user has) and a PIN (something the user knows). Business networks may require users to provide a password (knowledge factor) and a pseudorandom number from a security token (ownership factor). Access to a very-high-security system might require a mantrap screening of height, weight, facial, and fingerprint checks (several inherence factor elements) plus a PIN and a day code (knowledge factor elements), but this is still a two-factor authentication.

2.3 Product Authentication

Counterfeit products are often offered to consumers as being authentic. Counterfeit consumer goods such as electronics, music, apparel, and Counterfeit medications have been sold as being legitimate. Efforts to control the supply chain and educate consumers to evaluate the packaging and labeling help ensure that authentic products are sold and used. Even security printing on packages, labels, and nameplates, however, is subject to counterfeiting. A secure key storage device can be used for authentication in consumer electronics, network authentication, license management, supply chain management, etc. Generally the device to be authenticated needs some sort of wireless or wired digital connection to either a host system or a network.

Various systems have been invented to allow authors to provide a means for readers to reliably authenticate that a given message originated from or was relayed by them. These involve authentication factors like:

i. A difficult-to-reproduce physical artifact, such as a seal, signature, watermark, special stationery, or fingerprint.

ii. A shared secret, such as a passphrase, in the content of the message.

iii. An electronic signature; public-key infrastructure is often used to cryptographically guarantee that a message has been signed by the holder of a particular private key.

2.4 Internet Protocol security (IPsec)

It uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin authentication, data integrity, and data confidentiality (encryption), and replay protection. IPsec is mandatory in IPv6 and optional in IPv4

2.5 Digital Signatures

A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, such that the sender cannot deny having sent the message (authentication and non-repudiation) and that the message was not altered in transit (integrity). Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.

In cryptography, a message authentication code (often MAC) is a short piece of information used to authenticate a message and to provide integrity and authenticity assurances on the message. Integrity assurances detect accidental and intentional message changes, while authenticity assurances affirm the message's origin.

Message integrity code (MIC) is frequently substituted for the term MAC as usage of the term the MIC operation does not use secret keys. This lack of security means that any MIC intended for use gauging message integrity should be encrypted or otherwise be protected against tampering. MIC algorithms are created such that a given message will always produce the same MIC assuming the same algorithm is used to generate both. Conversely, MAC algorithms are designed to produce matching MACs only if the same message, secret key and initialization vector are input to the same algorithm. MICs do not

use secret keys and, when taken on their own, are therefore a much less reliable gauge of message integrity than MACs. Because MACs use secret keys, they do not necessarily need to be encrypted to provide the same level of assurance.

CHAPTER THREE: RESEARCH METHODOLOGY

3.1 Authenticity Based on PUF

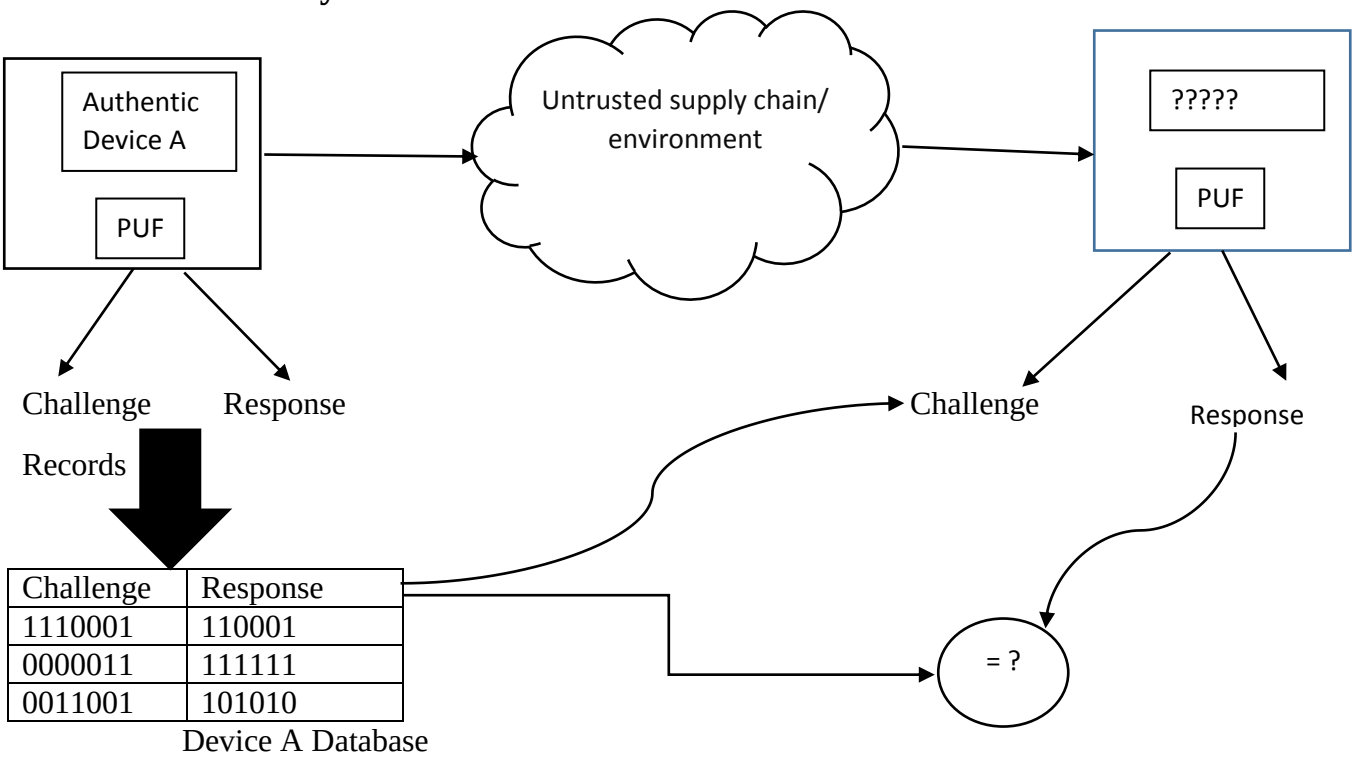


Figure 3.1 Overview of PUF Based Authentication

In this method, source is verified and will allow to communicate to the particular system after verification. As we know that during manufacturing process, due to process variation, every challenge input results different responses, therefore, we can maintain the records of challenge and response. So it can be easier for the system to lookup the challenge response pair (CRP) in order to authenticate the device [5].

3.2 Authentication Based on IPv6

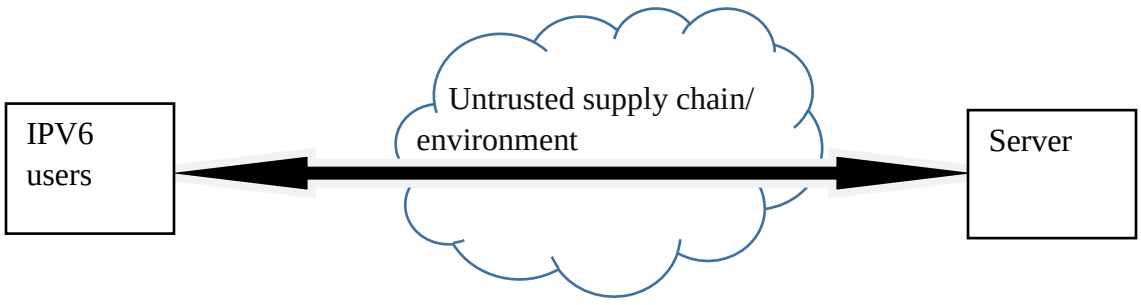


Figure 3.2 IPsec tunnel using IPv6

In this model, IPsec is mandatory. In IPv6, the packet datagram is designed in such a manner that it can maintain authenticity, integrity and confidentiality. It means whenever the IPv6 users try to communicate with other system or IPv6 users, there is IPv6 tunnel which will encapsulate the entire IP packet and send to the destination.

3.3 Model Development

As it is known that to have a secure communication, the three fundamental parameter are needed.

1. Confidentiality
2. Authenticity
3. Integrity

In this proposed thesis, we integrate PUF authentication with IPv6. In general, if we use IPV4 platform for PUF authentication, it requires to implement additional security features to transfer data from source to destination. It means, it requires encryption and data integrity which makes IPv4 packet size to be varied and complex. So instead of using IPV4, IPv6 has security features inbuilt on it. So there is less complexity in adopting encryption and authenticity while transferring data from source to destination in IPv6 platform

3.4 Generation of Challenge Response Pair using XOR Gates

The challenge response pair (CRP) is produced by the combination of XOR gates by using its gate delay property. The below diagram shows the method of producing 8 different 8bits responses at different time with 8bits challenge input. It uses the physically unclonable function principle which means that there exists delay characteristics property where no two IC chips or gates have identical delay value due to manufacturing variability. In more general term, if we design the same circuit diagram and implemented in the real hardware, for the same challenge, the response is different in two circuit. This is the beauty of the physically unclonable function.

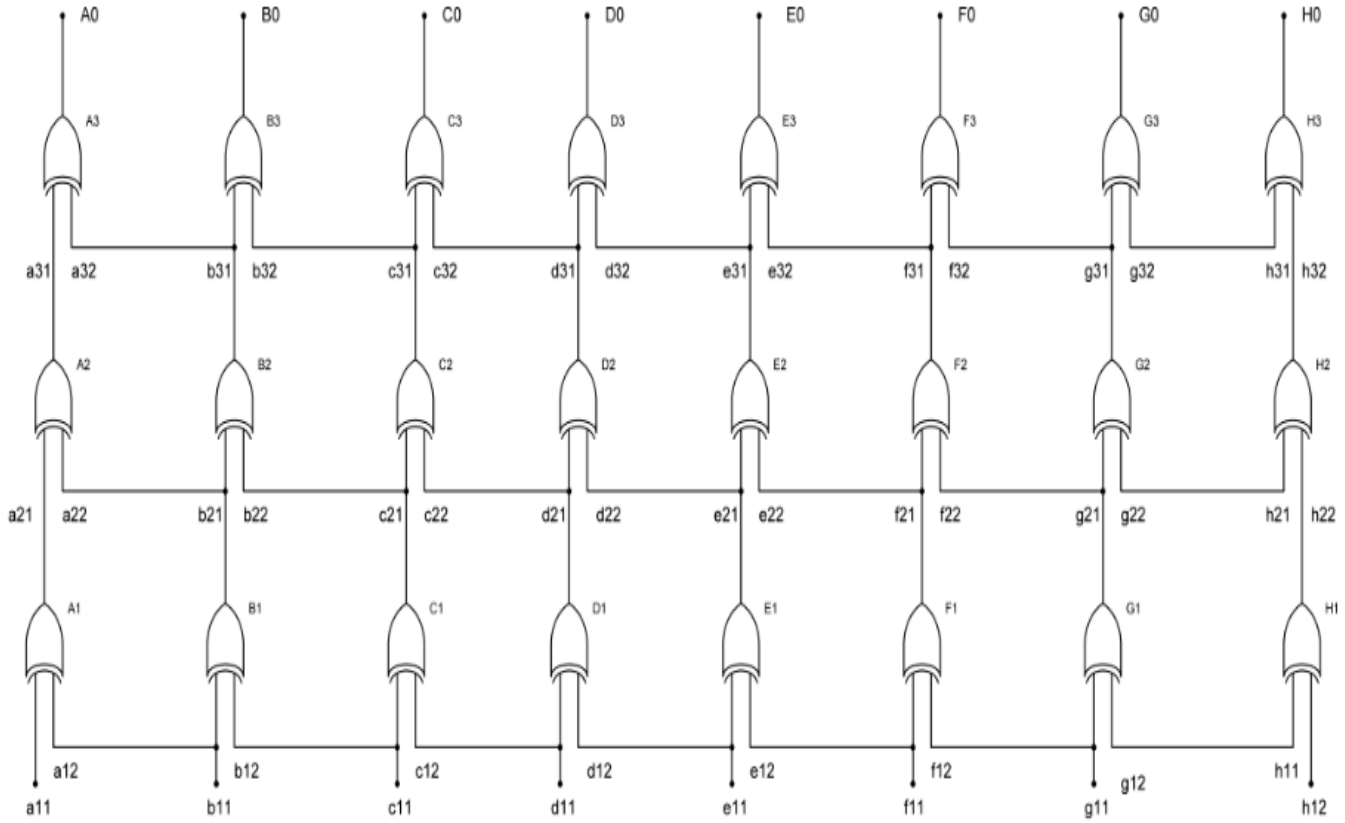


Figure 3.3 Challenge Response Pair generator

To generate CRP, the whole circuit should be in stable condition at first. As it consider the previous input and xor-ing the current input with the previous input. The truth table of 2 input XOR gate is given below [6].

a11(input)	a12(input)	a21(output)
0	0	0
0	1	1
1	0	1
1	1	0

Table 3.1 Truth table of XOR Gate

For simplicity, we assume that there are three rows (row1, row2 and row3). Each row has 16 different input gates and have different gates delays. The gate which has the least delay value is computed first with the previous input and so on. However the circuit structure make one input equals to other input at different level. It means

Input a12 = input b11, input b12=input c11, input c12=input d11, input d12=input e11, input e12=input f11, input f12=input g11, input g12=input h11 for row1 and so on row2 and row3 respectively. The gate delay is given below:

Serial	Gate	Delay in nanoseconds
1	Gate a11	0.9
2	Gate a12	0.92
3	Gate b11	0.89
4	Gate b12	1.01
5	Gate c11	1.02
6	Gate c12	1.03
7	Gate d11	1.04
8	Gate d12	1.05
9	Gate e11	1.06
10	Gate e12	0.99
11	Gate f11	0.95
12	Gate f12	0.96
13	Gate g11	0.97
14	Gate g12	0.89
15	Gate h11	0.78
16	Gate h12	1.1
17	Gate a21	1.1
18	Gate a22	1.02
19	Gate b21	1.03
20	Gate b22	1.05

21	Gate c21	1.06
22	Gate c22	1.07
23	Gate d21	1.08
24	Gate d22	1.09
25	Gate e21	1.02
26	Gate e22	1.06
27	Gate f21	1.07
28	Gate f22	1.01
29	Gate g21	0.81
30	Gate g22	0.82
31	Gate h21	0.85
32	Gate h22	0.99
33	Gate a31	0.89
34	Gate a32	0.79
35	Gate b31	0.9
36	Gate b32	0.91
37	Gate c31	0.92
38	Gate c32	0.93
39	Gate d31	1.1
40	Gate d32	1.2
41	Gate e31	1.15
42	Gate e32	1.16
43	Gate f31	1.16
44	Gate f32	1.17
45	Gate g31	1.2
46	Gate g32	1.01
47	Gate h31	1.02
48	Gate h32	1.03

Table 3.2 Gates and its delay value

The combination of a11, b11, c11, d11, e11, f11, g11 and h12 is one 8 bits challenge and A0, B0, C0, D0, E0, F0, G0 and H0 is one 8 bits response at certain time. Since there are three rows so the number responses is $2^{\wedge 3}$ is equal to 8. It means one 8 bits challenge will produce 8 different 8 bits response at 8 different average time [6].

In order to calculate the result of XOR from row1, row2 and row3, we need to consider two things.

i. Gate Delay
ii. Previous input

The gate which has the least value is computed first with the previous input and so on. Moreover gate delay is additive in nature so for the result of xor in row2, the delay from row1 is also considered as per the circuit structure.

Some php code to compute challenge response pairs are

$row1=array(0.9,0.92,0.89,1.01,1.02,1.03,1.04,1.05,1.06,0.99,0.95,0.96,0.97,0.89,0.78,1.1);

$row2=array(1.1,1.02,1.03,1.05,1.06,1.07,1.08,1.09,1.02,1.06,1.07,1.01,0.81,0.82,0.85,0.99);

$row3=array(0.89,0.79,0.9,0.91,0.92,0.93,1.1,1.2,1.15,1.16,1.16,1.17,1.2,1.01,1.02,1.03);

These $row1, $row2 and $row3 are the array of gate delay. Xor begins from row1 with 0.78 delay value with the previous input.

In this thesis, I have given 10110011 as previous in which the circuit has stable output. So the computation is made between current input and previous input.

The possible outcomes are the value obtain by xor-ing as below

$h2h12=($h11 xor $h);$g2g12=($g12 xor $g);$b2b11=($b11 xor $c);$a2a11=($a11 xor $b11);$a2a12=($a12 xor $a11);$f2f11=($f11 xor $g);$f212=($f12 xor $f11);

$g2g11=($g11 xor $g12);$e2e12=($e12 xor $e);$b2b12=($b12 xor $a12);$c2c11=($c11 xor $d);$c2c12=($c12 xor $c11);$d2d11=($d11 xor $e);$d2d12=($d12 xor $d11);

$e2e11= ($e11 xor $e12); $h2h11= ($h11 xor $h12);

There are 16 possible outcomes

Also if we consider the delay of row1 and row2 and compute the additive time delay as per the circuit structure we can obtain time delay and xor resulting formula as follow

For time delay,

$A2arr=array($trow1[0]+$trow2[0],$trow1[1]+$trow2[0],$trow1[2]+$trow2[1],$trow1[3]+$trow2[1]);

$B2arr=array($trow1[2]+$trow2[2],$trow1[3]+$trow2[2],$trow1[4]+$trow2[3],$trow1[5]+$trow2[3]);

$C2arr=array($trow1[4]+$trow2[4],$trow1[5]+$trow2[4],$trow1[6]+$trow2[5],$trow1[7]+$trow2[5]);

$D2arr=array($trow1[6]+$trow2[6],$trow1[7]+$trow2[6],$trow1[8]+$trow2[7],$trow1[9]+$trow2[7]);

$E2arr=array($trow1[8]+$trow2[8],$trow1[9]+$trow2[8],$trow1[10]+$trow2[9],$trow1[11]+$trow2[9]);

$F2arr=array($trow1[10]+$trow2[10],$trow1[11]+$trow2[10],$trow1[12]+$trow2[11],$trow1[13]+$trow2[11]);

$G2arr=array ($trow1[12]+$trow2[12],$trow1[13]+$trow2[12]);

$H2arr=array ($trow1[14]+$trow2[15],$trow1[15]+$trow2[15]);

For Xor result,

$g3g21=($g2g12 xor $g22);$h3h21=($h2h11 xor $h21);$g3g22=($g2g11 xor $g22);

$f3f21= $g2g21 xor $f21;

$a3a21=$b2b11 xor $a21;$b3b21=$b2b11 xor $b22;$f3f22=$g2g11 xor $f21;

$a3a22= $b3b21 xor $a2a11; $e3e21=$e2e12 xor $d22;$e3e22=$f2f11 xor $e3e21;

$e3e23= $f2f12 xor $e2e21;

$f3f23=$f2f11 xor $f3f22;$a3a23=$b2b12 xor $a3a22;$a3a24=$b2b12 xor $a3a23;

$f3f24=$f2f12 xor $f3f22;$b3b22=$b2b12 xor $b22;$b3b23=$c2c11 xor $b3b22;

$b3b24= $c2c12 xor $b3b22;

$c3c21=$c2c11 xor $c22;$e3e24=$e2e11 xor $f3f23;$d3d21=$e2e12 xor $d21;

$c3c22=$c2c12 xor $c22;$h3h22=$h2h12 xor $h3h21;$c3c23=$d2d11 xor $c3c22;

$c3c23= $d2d11 xor $c3c22;

$c3c24=$d2d12 xor $c3c22;$d3d22=$d2d11 xor $d3d21;$d3d23=$d2d12 xor $d3d21;

$d3d24= $d2d11 xor $d3d23;

So there are 32 intermediate xor result at 32 different delay time.
At final stage, that means after considering the row3 gate delay value,
The time delay is computed as follow:

$A3arr=array($trow1[0]+$trow2[0]+$trow3[0],$trow1[1]+$trow2[0]+$trow3[0],$trow1[2]+$trow2[1]+$trow3[0],$trow1[3]+$trow2[1]+$trow3[0],$trow1[2]+$trow2[2]+$trow3[1],$trow1[3]+$trow2[2]+$trow3[1],$trow1[4]+$trow2[3]+$trow3[1],$trow1[5]+$trow2[3]+$trow3[1]);

$B3arr=array($trow1[2]+$trow2[2]+$trow3[2],$trow1[3]+$trow2[2]+$trow3[2],$trow1[4]+$trow2[3]+$trow3[2],$trow1[5]+$trow2[3]+$trow3[2],$trow1[4]+$trow2[4]+$trow3[3],$trow1[5]+$trow2[4]+$trow3[3],$trow1[6]+$trow2[5]+$trow3[3],$trow1[7]+$trow2[5]+$trow3[3]);

$C3arr=array($trow1[4]+$trow2[4]+$trow3[4],$trow1[5]+$trow2[4]+$trow3[4],$trow1[6]+$trow2[5]+$trow3[4],$trow1[7]+$trow2[5]+$trow3[4],$trow1[6]+$trow2[6]+$trow3[

5],$trow1[7]+$trow2[6]+$trow3[5],$trow1[8]+$trow2[7]+$trow3[5],$trow1[9]+$trow2[7]+$trow3[5]);

$D3arr=array($trow1[6]+$trow2[6]+$trow3[6],$trow1[7]+$trow2[6]+$trow3[6],$trow1[8]+$trow2[7]+$trow3[6],$trow1[9]+$trow2[7]+$trow3[6],$trow1[8]+$trow2[8]+$trow3[7],$trow1[9]+$trow2[8]+$trow3[7],$trow1[10]+$trow2[9]+$trow3[7],$trow1[11]+$trow2[9]+$trow3[7]);

$E3arr=array($trow1[8]+$trow2[8]+$trow3[8],$trow1[9]+$trow2[8]+$trow3[8],$trow1[10]+$trow2[9]+$trow3[8],$trow1[11]+$trow2[9]+$trow3[8],$trow1[10]+$trow2[10]+$trow3[9],$trow1[11]+$trow2[10]+$trow3[9],$trow1[12]+$trow2[11]+$trow3[9],$trow1[13]+$trow2[11]+$trow3[9]);

$F3arr=array($trow1[10]+$trow2[10]+$trow3[10],$trow1[11]+$trow2[10]+$trow3[10],$trow1[12]+$trow2[11]+$trow3[10],$trow1[13]+$trow2[11]+$trow3[10],$trow1[12]+$trow2[12]+$trow3[11],$trow1[13]+$trow2[12]+$trow3[11]);

$G3arr=array($trow1[12]+$trow2[12]+$trow3[12],$trow1[13]+$trow2[12]+$trow3[12]);

$H3arr=array($trow1[14]+$trow2[15]+$trow3[15],$trow1[15]+$trow2[15]+$trow3[15]);

The Xor result is computed as follow:

$a4a31=$b2b11 xor $a31;$a4a32=$b2b12 xor $a4a31;$h4h31=$h3h21 xor $h31;

$b4b31= $b2b11 xor $b32;$a4a33=$b2b12 xor $a4a32;

$a4a34=$c2c11 xor $a4a32;$a4a35=$c2c12 xor $a4a32;

$f4f31= $g2g12 xor $f31;$a4a36=$a3a11 xor $a4a35;$g4g31=$g2g12 xor $g32;

$f4f32= $g2g12 xor $f31;$b4b33=$c2c11 xor $b32;$g4g32=$g2g11 xor $g32;

$b4b34=$c2c12 xor $b32; $b4b35=$c2c11 xor $b4b34;

$c4c31=$c2c11 xor $c32;$b4b36=$c2c12 xor $b4b34;

$c4c32= $c2c12 xor $c32;$c4c33=$e2e12 xor $c4c32;$b4b37=$d2d12 xor $b4b34;

$c4c34= $d2d11 xor $c4c33;$b4b38=$d2d12 xor $b4b34;

$c4c35=$d2d11 xor $c4c33;$c4c36=$d2d11 xor $c4c35;

$c4c37=$d2d12 xor $c4c35;$f4f33=$g2g12 xor $f4f32;$e4e31=$g2g12 xor $e31;

$c4c38= $e2e11 xor $c4c35;$h4h32=$h2h12 xor $h31;$e4e32=$g2g11 xor $e31;

$f4f34=$g2g11 xor $f4f32; $e4e33=$e2e12 xor $e4e32;

$e4e34=$f2f11 xor $e4e32;$e4e35=$f2f12 xor $e4e32;

$e4e36=$f2f11 xor $e4e35; $f4f35=$f2f11 xor $f4f32;$d4d31=$e2e12 xor $e4e35;

$f4f36= $f2f12 xor $f4f32;$e4e37=$f2f12 xor $e4e35;

$d4d32=$e2e12 xor $d4d31;$d4d33=$f2f11 xor $d4d34;

$d4d34=$d2d11 xor $d4d33; $d4d35=$f2f12 xor $d4d34;$d4d36=$d2d12 xor $d4d35;

$e4e38=$e2e11 xor $e4e37;$d4d37=$e2e11 xor $d4d35; $d4d38=$e2e11 xor $d4d37;

So after row3, there are 64 possible responses at 64 possible time so in order to produce 1btye word we need to categorize the possible outcome in such a way that it result certain challenge response pair. It means the combination of first output bit of each row3 gives one 8 bits response at average time. The average time is computed by taking average of 8 different time of first output bits and so on. So, finally, we will get 8 different 8bits responses for one 8 bits challenge. If we increase the number of rows we can get different possible responses at different time.

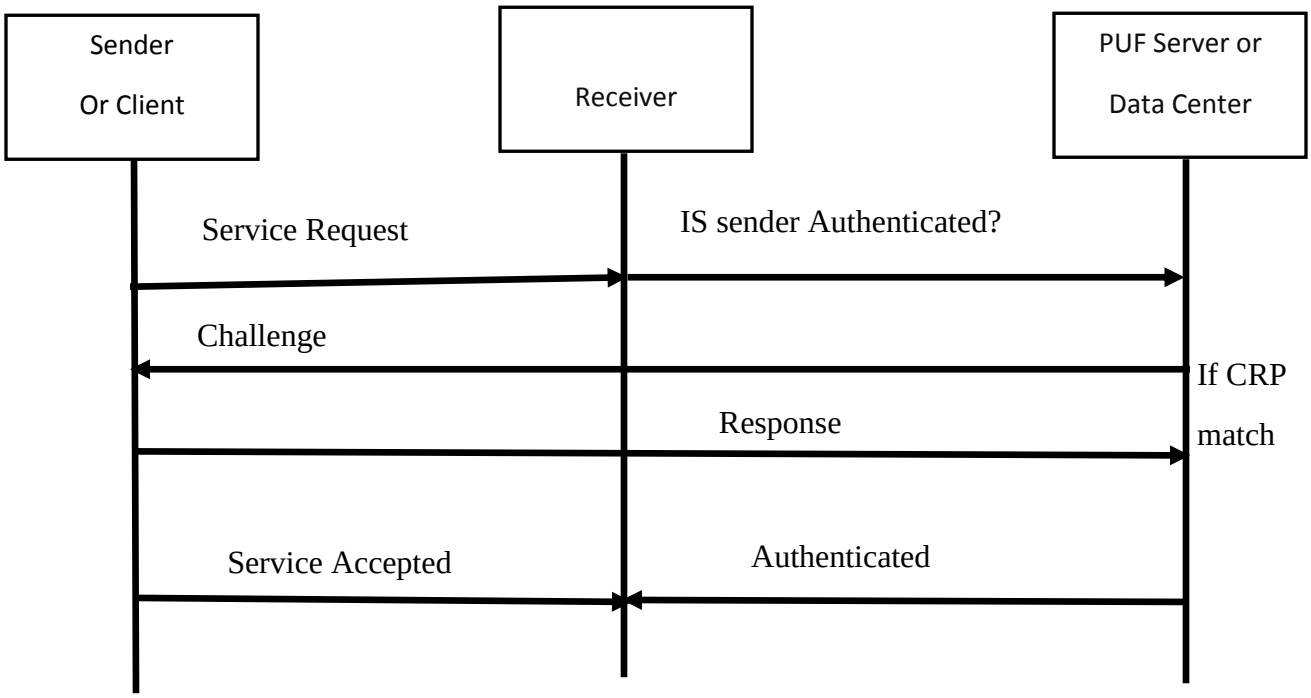


Figure 3.4 Sequential diagram of User/device authentication

3.5 Algorithm

Step1 Client will try to access secure web server for some services

- Whenever the client try to access remote web application, it will communicate at port 80 with the web server.

Step2 Web server will make query to Authenticate server whether client is authenticated or not

- At Authenticate server, challenge response pair is stored in MySQL databases and authenticate server will make query to client as client's IP is provided by web server.

Step3 Authenticate server will make direct communication with client.

- After obtaining source IP of client, authenticate server will communicate at port 80 with the client for its authenticity. This is done by sending some random challenge to the client.

Step4 Client will compute the challenge and gives responses and send it back to server

- The sender will compute the challenge and produce the response and it will transmit to the authenticate server at port 80.

Step5 Authenticate server will look up on its database and if challenge-response pair is matched, then it will inform the web server that the client is authenticated

- In this step, the query made by the web server at the early stage is acknowledged, and depending upon this query result, the web server will accept or reject the requested service.

Step6 Web server will allow the client to log in.

- After successful source authentication, the client will login into web server and grant the requested services

3.6 **Realization of PUF in IPv6**

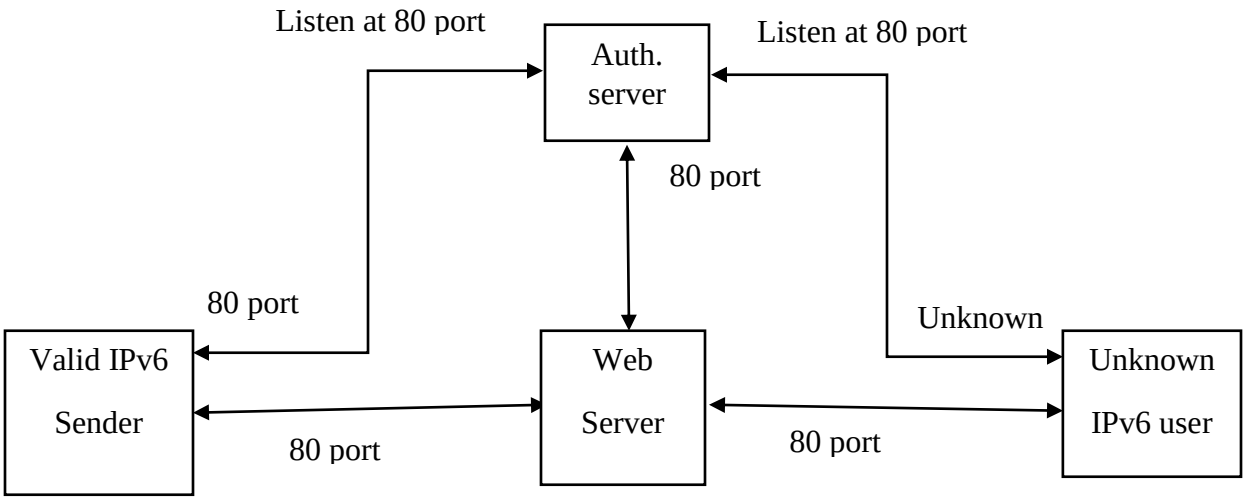


Figure 3.5 Overall Diagram of PUF authentication in IPv6

In order to maintain the challenge response pair database, we use php to generate the challenge response pair and stored in PUF server. Then whenever the web server would like to know about client's authenticity, it will make query on the authenticate server through port 80 using header ( ) function. After that, authenticate server make direct communication to client for verification. The authenticate server send random challenge to the client and the client compute the challenge and produce the response and send it back to the server. The authenticate server compare the challenge response pair at that instant of time with its recorded database and if CRP matches, it will give query result to the web server indicating whether source is verified or not. At last, depending upon the query result, web server will accept client request or reject.

The coding is done using php. As we don't have PUF hardware available so I simulate XOR gate circuit diagram as shown in figure 3.3 in php and neglecting temperature variation factors.

In client side, there are two php files

a. crpforward.php

b. crpgenerator.php

Similarly, there are two files in web server

a. index.php

b. home.php

At last, the authenticate server contain a single php file

a. ipcheck.php

The process of php file flow goes like this

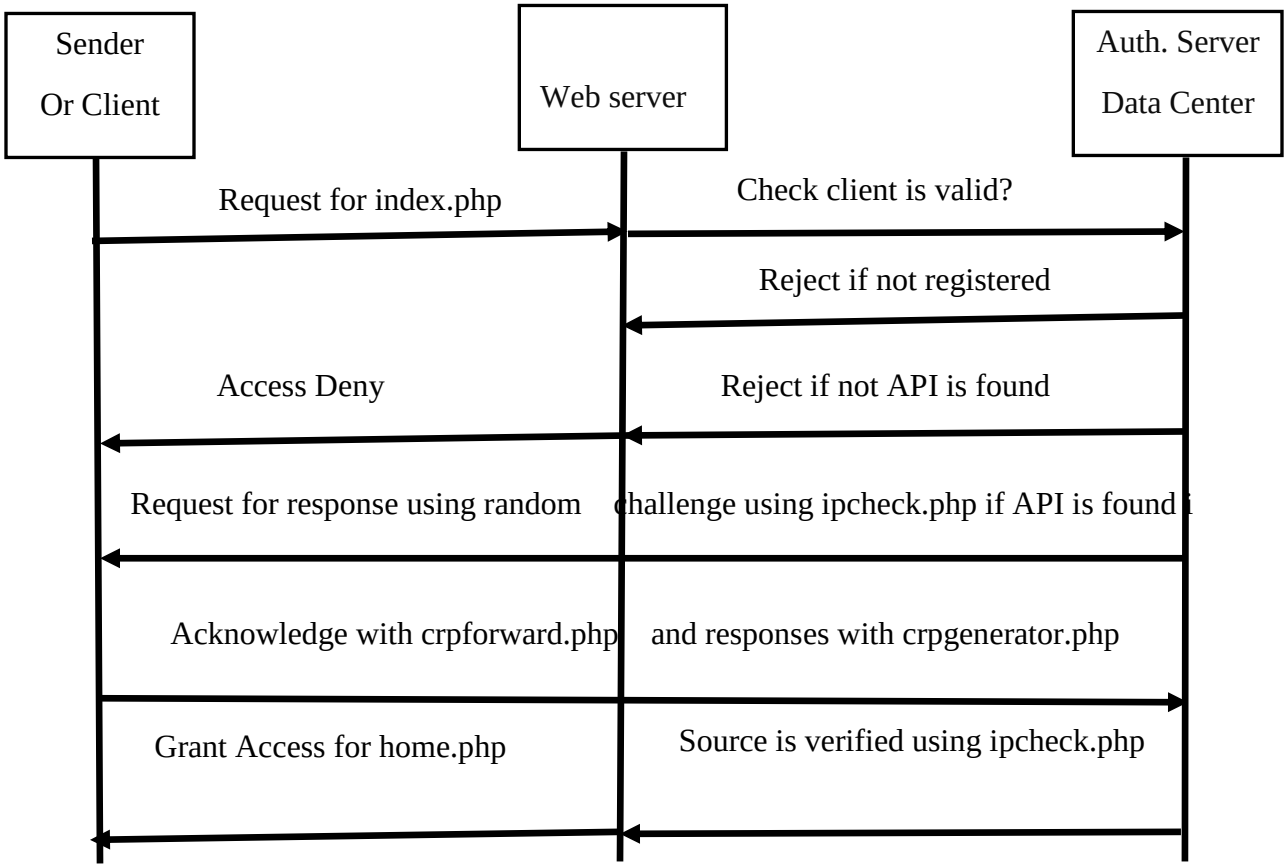


Figure 3.6 Sequential PHP file flow for authentication

3.7 **Verification**

In order to verify the authentic user/sender and non-authentic user/sender, the user whose challenge response pairs are already stored in the central database are allowed to communicate with the receiver and other whose CRP are not stored in the DB are not allowed. The recorded CRP is matched against the instant CRP generation and if it is matched, the source is verified and further access is allowed.

CHAPTER FOUR: RESULTS AND DISCUSSIONS

4.1 Assumptions and outputs

The test is done in Linux OS (Centos OS 6.0), MySQL database, PHP programming language and GNS3. The whole system is done in the ipv6 environment. The client, web server and authenticate server ip is given below

i. Client ip is 2003:db8:0:f101:130/64

ii. Web server ip is 2001:db8:0:f101:131/64

iii. Authenticate server ip is 2001:db8:0:f101:129/64

The database is maintained at server which verify the sender/client and provide information about sender/client authenticity.

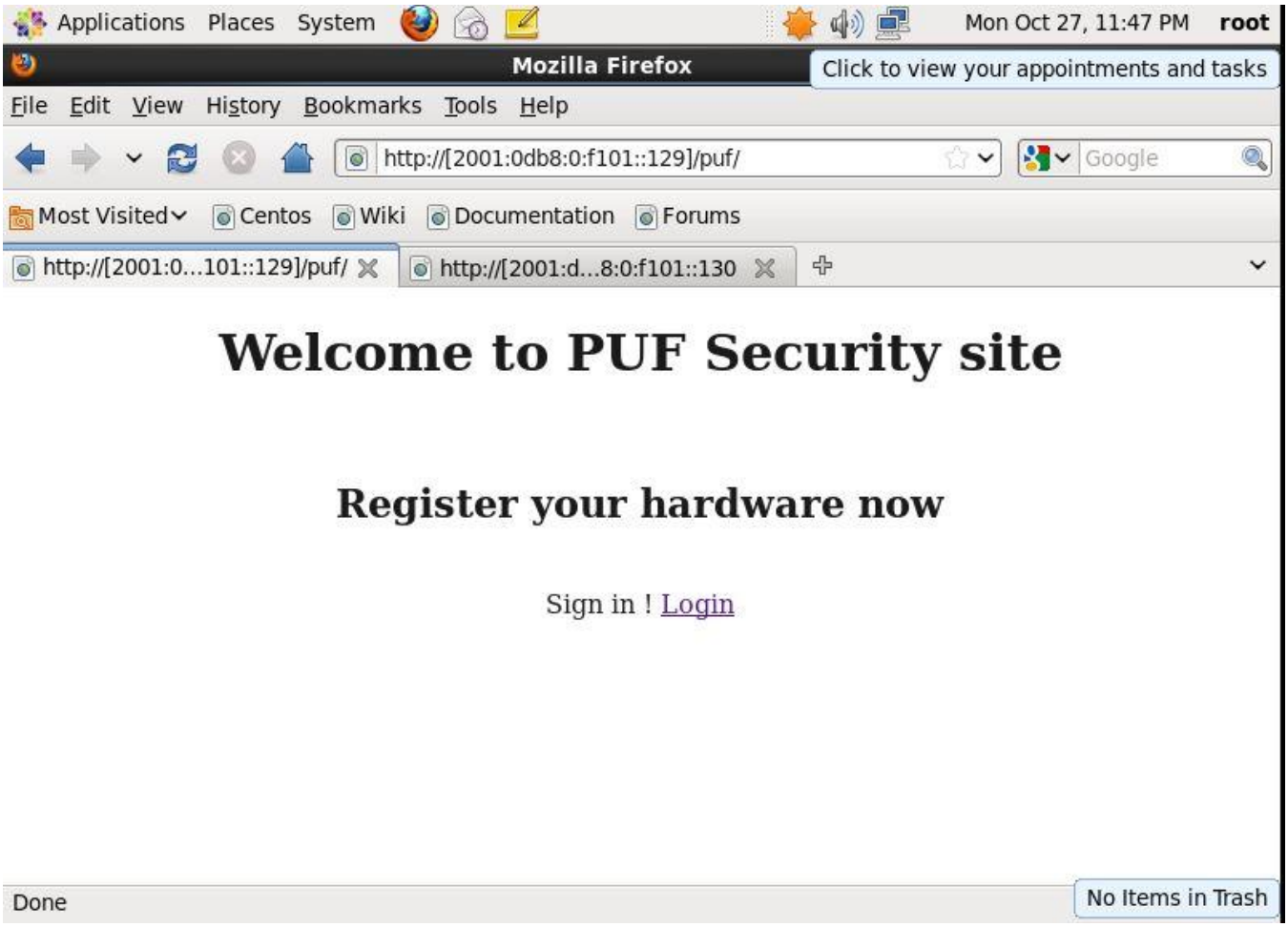


Figure 4.1 Login page for Hardware Registration

At first, the users/client are provided to register their hardware in the authenticate server. This is the web portal in which the client will register his/her hardware. Here, only for this thesis, the client will generate challenge and response pair and it is stored in the authenticate server database.

```
mysql> select * from tbl_crp where ip='2003:db8:0:f101::130';
+--------+-----------+----------+---------+----------------------+
| serial | challenge | response | time    | ip                   |
+--------+-----------+----------+---------+----------------------+
|    113 | 11100010  | 11100000 | 2.9175  | 2003:db8:0:f101::130 |
|    114 | 11100010  | 11001010 | 3.01875 | 2003:db8:0:f101::130 |
|    115 | 11100010  | 11000010 | 2.28    | 2003:db8:0:f101::130 |
|    116 | 11100010  | 10000110 | 2.29875 | 2003:db8:0:f101::130 |
|    117 | 11100010  | 01011010 | 2.30875 | 2003:db8:0:f101::130 |
|    118 | 11100010  | 10000110 | 2.3175  | 2003:db8:0:f101::130 |
|    119 | 11100010  | 10001110 | 1.92875 | 2003:db8:0:f101::130 |
|    120 | 11100010  | 10000110 | 1.9425  | 2003:db8:0:f101::130 |
|    121 | 11001111  | 11010000 | 2.9175  | 2003:db8:0:f101::130 |
|    122 | 11001111  | 01011010 | 3.01875 | 2003:db8:0:f101::130 |
|    123 | 11001111  | 00111010 | 2.28    | 2003:db8:0:f101::130 |
|    124 | 11001111  | 00001110 | 2.29875 | 2003:db8:0:f101::130 |
|    125 | 11001111  | 00011110 | 2.30875 | 2003:db8:0:f101::130 |
|    126 | 11001111  | 10011110 | 2.3175  | 2003:db8:0:f101::130 |
|    127 | 11001111  | 11111110 | 1.92875 | 2003:db8:0:f101::130 |
|    128 | 11001111  | 01111110 | 1.9425  | 2003:db8:0:f101::130 |
|    129 | 11001110  | 11010000 | 2.9175  | 2003:db8:0:f101::130 |
|    130 | 11001110  | 01011010 | 3.01875 | 2003:db8:0:f101::130 |
|    131 | 11001110  | 00111010 | 2.28    | 2003:db8:0:f101::130 |
|    132 | 11001110  | 00001110 | 2.29875 | 2003:db8:0:f101::130 |
|    133 | 11001110  | 00011110 | 2.30875 | 2003:db8:0:f101::130 |
|    134 | 11001110  | 10011110 | 2.3175  | 2003:db8:0:f101::130 |
|    135 | 11001110  | 11111110 | 1.92875 | 2003:db8:0:f101::130 |
|    136 | 11001110  | 01111110 | 1.9425  | 2003:db8:0:f101::130 |
+--------+-----------+----------+---------+----------------------+
24 rows in set (0.00 sec)

mysql>
```

Figure 4.2 IPv6 with challenge response pairs store in database

The above figure shows that the client's challenge response is pre-computed and stored at the central databases in the authenticate server. Besides these, time and corresponding IPv6 address is also stored in the database of authenticate server.

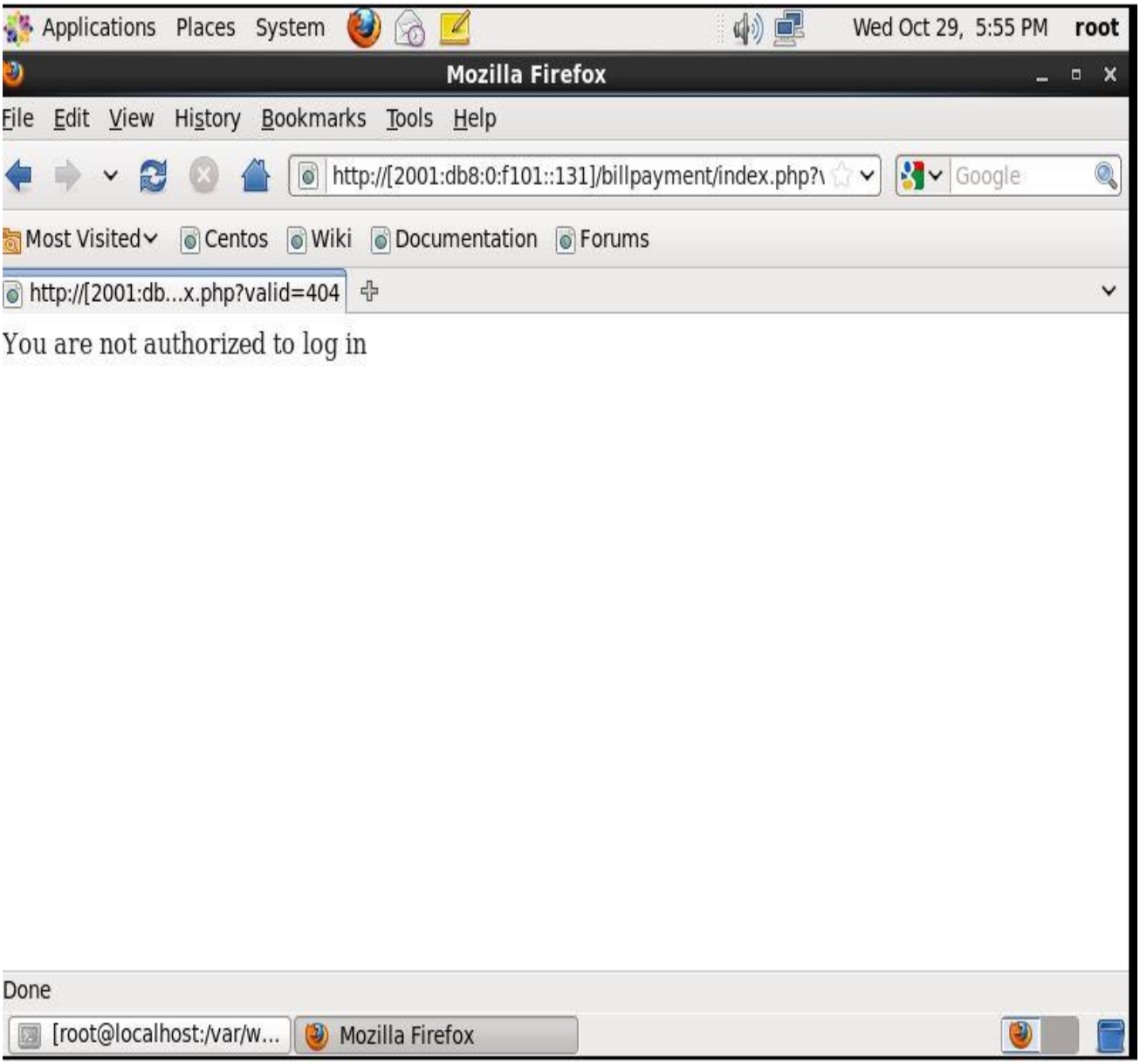


Figure 4.3 Prevention of unauthorized access to web server

Figure 4.3 signifies that client will not able to access web server in two cases

1. If the client IP address along with challenge response pair are not stored in the authenticate server

2. Although the client's IP is stored or an intruder spoofed authenticate client's IP, he/she will not able to access to the web server because authenticate server always back track the client in order to compute challenge response pair instantly and compare with the stored record. So it is less likely to grant access to unknown or IP spoofed users in the web server.

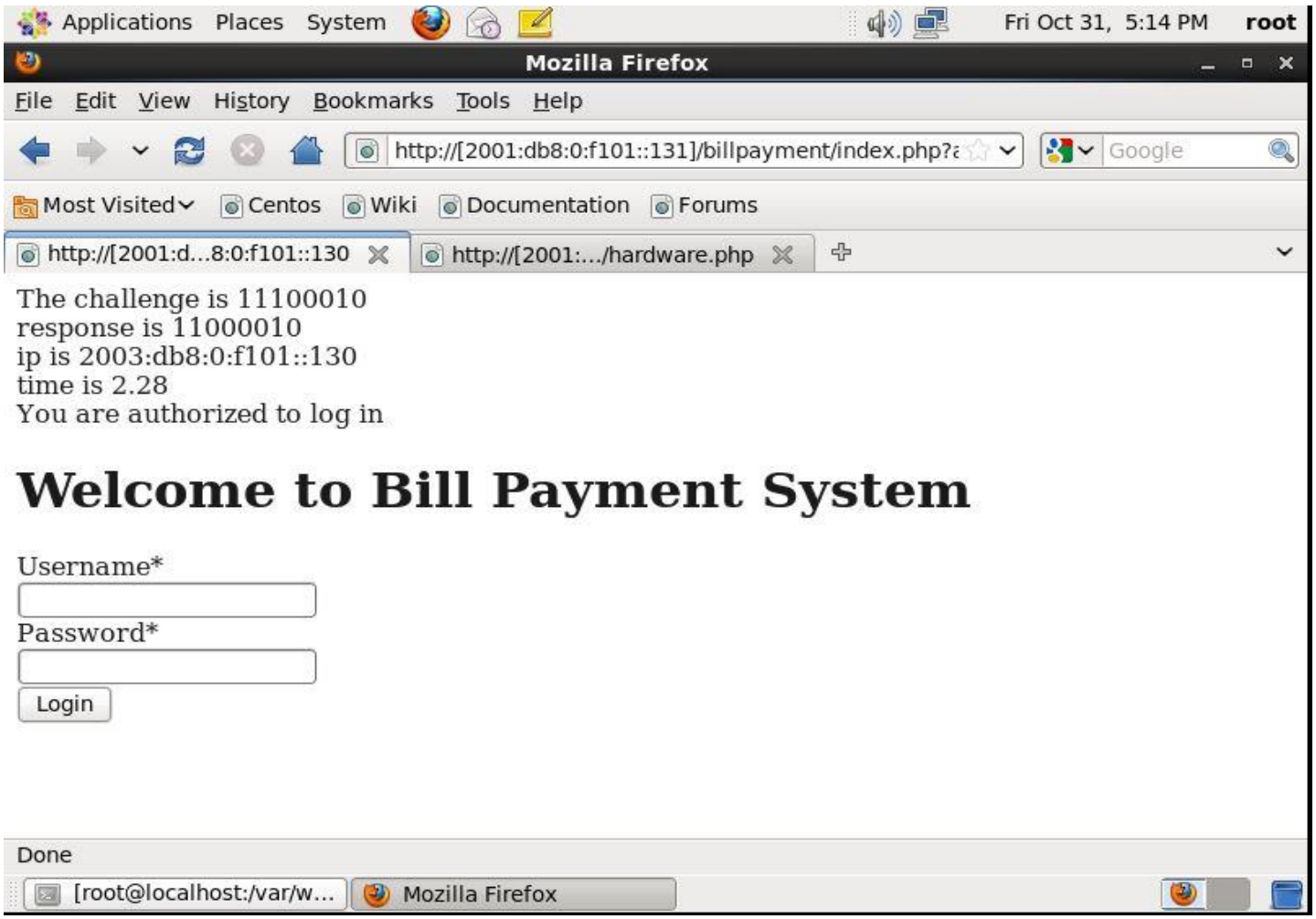


Figure 4.4 Providing access to web server for authentic client

Figure 4.4 shows that for the authentic users, proper challenge response pair generation at particular instant of time will validate the client and he/she will able to access to the web server and can acknowledge the requested services. Also the total time taken to execute the whole cycle is less than milliseconds.

```
mysql>
mysql> select * from tbl_crp where ip='2003:db8:0:f101::130'
    -> and challenge='11100010' and time='2.28';
+--------+-----------+----------+------+----------------------+
| serial | challenge | response | time | ip                   |
+--------+-----------+----------+------+----------------------+
|    115 | 11100010  | 11000010 | 2.28 | 2003:db8:0:f101::130 |
+--------+-----------+----------+------+----------------------+
1 row in set (0.00 sec)

mysql>
mysql>
mysql>
```

Figure 4.5 Database record at particular CRP and time

At last, cross verification is also important. The recorded database is matched against the instant challenge response pair generation of the client. It is found that the instant computation of challenge response pair is matched with the recorded database and source is verified and access is granted.

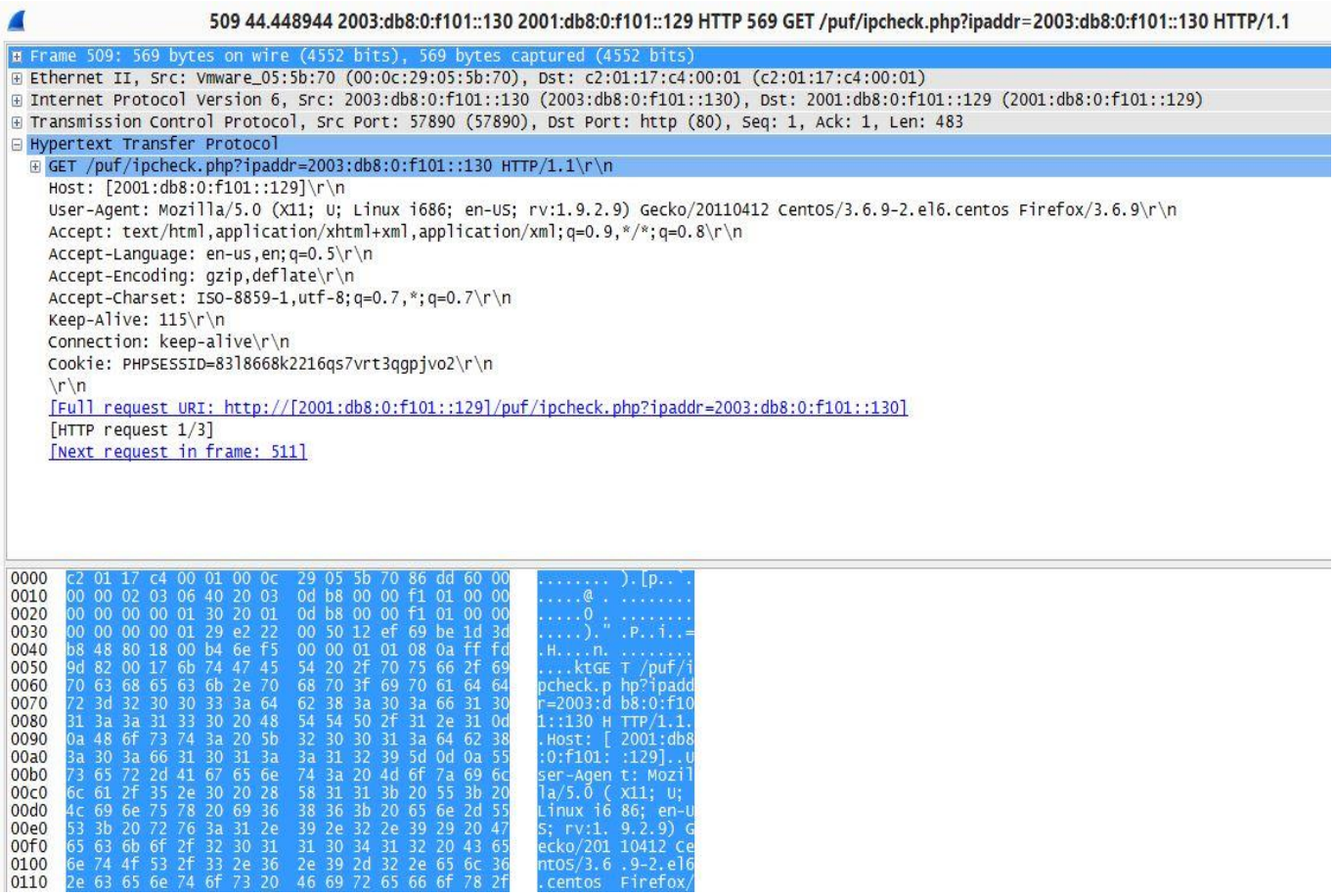


Figure 4.6 Packet Capture at authenticate server

The figure above shows that whenever the client want to get the services, the web server always request the authenticate server to provide his/her genuine identity. The decision of the web server to give access/deny always depends upon the response from the authenticate server.

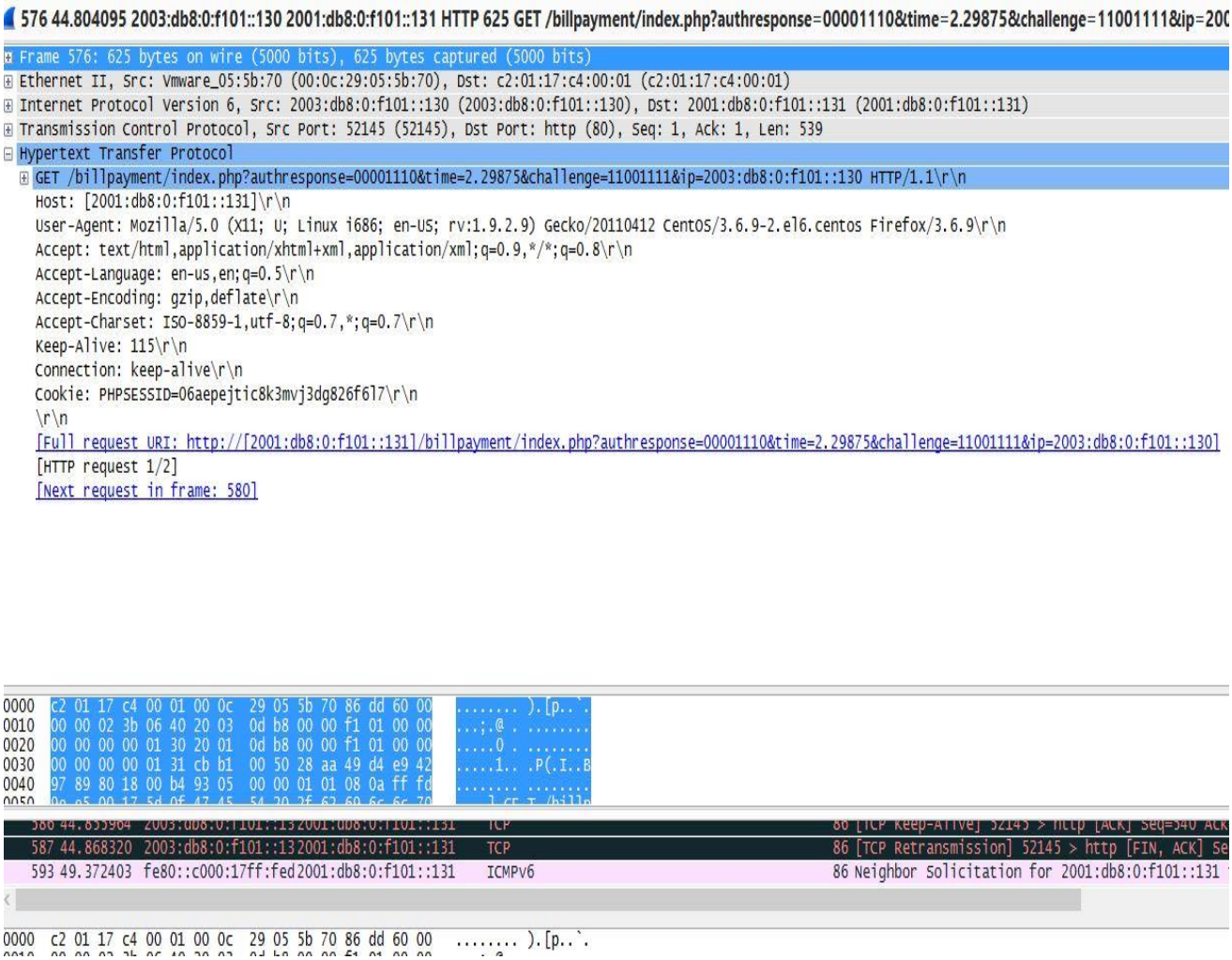


Figure 4.7 Packet Capture at client

The above diagram shows that after successful comparison of challenge response pair with the instant generation of challenge response pair at the client, the web server is acknowledged to give permission to the client by the authenticate server. So, as a result, the client is getting the service as per his request.

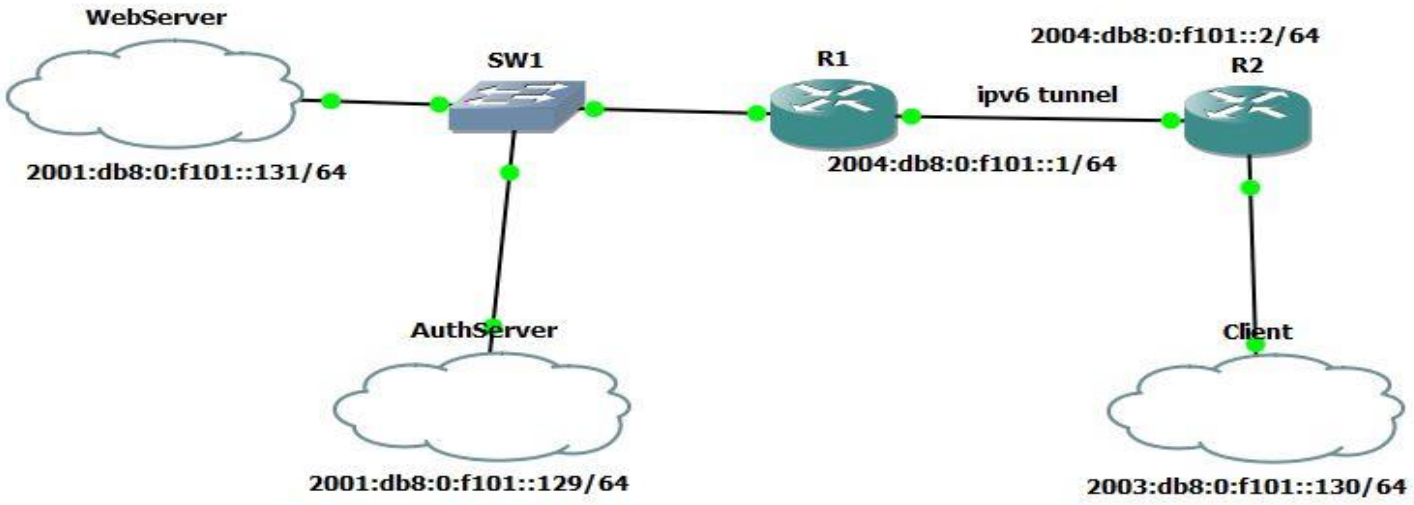


Figure 4.8 Client connecting web sever

It is clear from the above diagram that client is communicating to the web server through communication channel. The ipv6 channel is made ipsec tunnel so that everything is encrypted and encapsulated in that tunnel. So, whenever challenge response pair is computed and transfer across this tunnel, packet is fully protected. Here, ipv6 ipsec tunnel is created using tunnel interface. The router R1 has tunnel interface with ipv6 address 2012::1/64 and R2 has ipv6 address 2012::2/64. The ip address facing to each other of Router R1 and R2 as shown in figure is global ip address which is similar as public ip address in IPv4. The web server and authenticate server can be placed at different network address but for simplicity, it is placed in the same network. After setting up tunnel, we can verify the tunnel status by executing following command in router.

#show crypto isakmp sa

And the output of this command is

```
state: MM_NO_STATE     conn-id:      0 slot:    0 status: ACTIVE (deleted)

dst: 2004:DB8:0:F101::1
src: 2004:DB8:0:F101::2
state: QM_IDLE         conn-id:   1001 slot:    0 status: ACTIVE
```

Figure 4.9 IPSEC tunnel status in ipv6

From the above output, it is known that there is a tunnel setup between two ipv6 address whose source is 2004:db8:0:f101::1 and 2004:db8:0:f101::2. So, everything that transfer from this tunnel is protected by the ipsec mechanism. Also we need to make a policy so that our desire client/packets are forwarded into this tunnel. As soon as the tunnel is setup, first we need to trace the route from source end (client ip) to the web servers and authenticate server.

```
[root@localhost puf]# traceroute 2001:db8:0:f101::131
traceroute to 2001:db8:0:f101::131 (2001:db8:0:f101::131), 30 hops max, 80 byte packe
ts
 1  2003:db8:0:f101::120 (2003:db8:0:f101::120)  23.086 ms  33.207 ms  43.357 ms
 2  2012::1 (2012::1)  63.666 ms  73.691 ms  83.836 ms
 3  2001:db8:0:f101::131 (2001:db8:0:f101::131)  66.725 ms  76.864 ms  86.953 ms
[root@localhost puf]# traceroute 2001:db8:0:f101::129
traceroute to 2001:db8:0:f101::129 (2001:db8:0:f101::129), 30 hops max, 80 byte packe
ts
 1  2003:db8:0:f101::120 (2003:db8:0:f101::120)  25.159 ms  34.589 ms  44.701 ms
 2  2012::1 (2012::1)  188.838 ms  198.974 ms  208.783 ms
 3  2001:db8:0:f101::129 (2001:db8:0:f101::129)  218.781 ms  228.761 ms  238.437 ms
[root@localhost puf]#
[root@localhost puf]#
[root@localhost puf]#
```

Figure 4.10 Trace route from client to servers

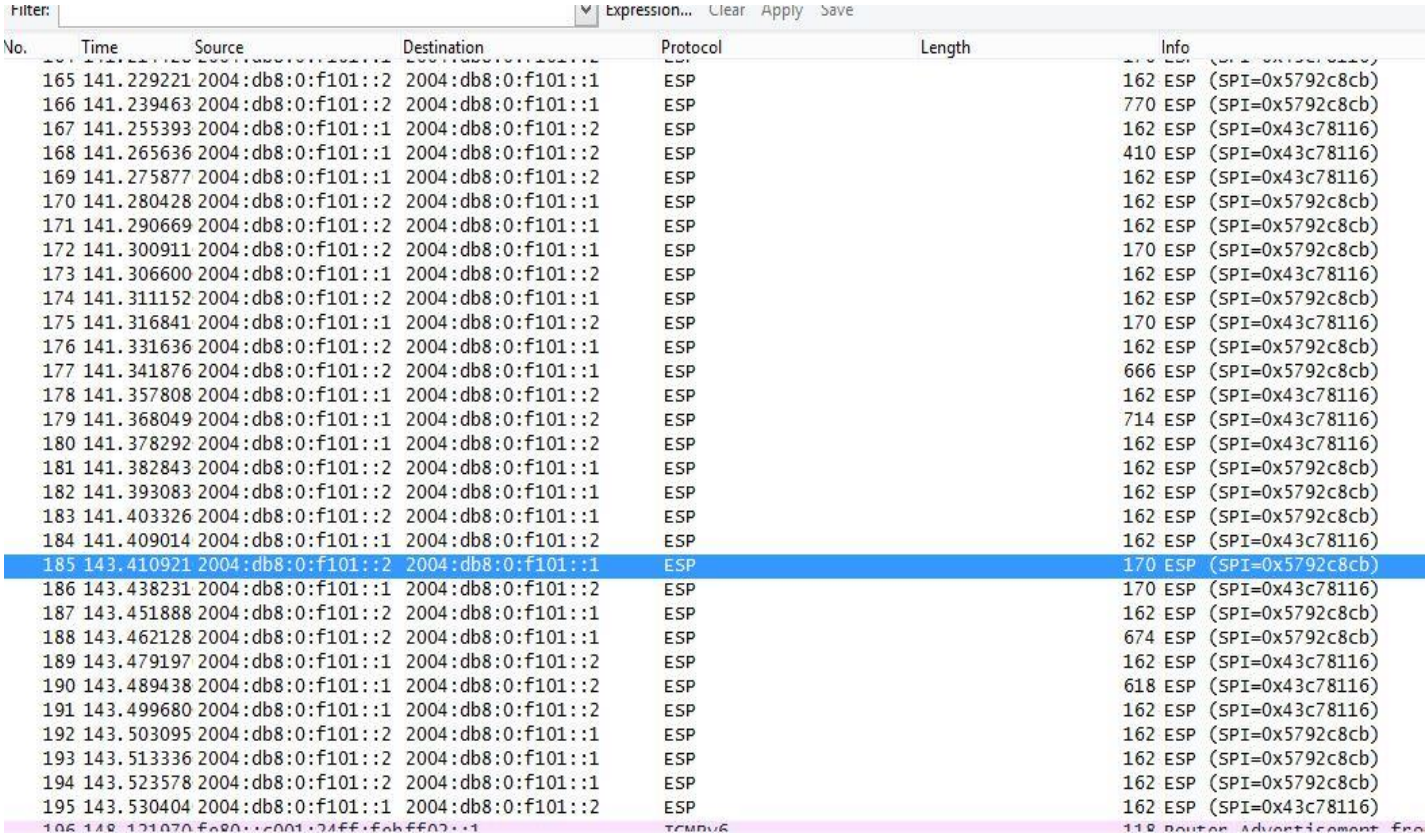
The above trace route validate that the client's packet traffic would be transferred to servers using tunnel interface which has 2012::1 ip address and which has actual physical interface as 2004:db8:0:f101::1. Thus, if the client request http service, the request are reached to the servers using tunnel and get acknowledge as per the requirement.

Similar trace route is also obtained if we would trace the client from servers. The only different is the leaving interface. Here in this step, 2012::2 is used to reach the client which has the physical interface as 2004:db8:0:f101::2.

```
[root@localhost Desktop]#
[root@localhost Desktop]#
[root@localhost Desktop]#
[root@localhost Desktop]# traceroute 2003:db8:0:f101::130
traceroute to 2003:db8:0:f101::130 (2003:db8:0:f101::130), 30 hops max, 80 byte
packets
 1  2001:db8:0:f101::120 (2001:db8:0:f101::120)  12.603 ms  22.731 ms  32.733 ms
 2  2012::2 (2012::2)  94.054 ms  104.125 ms  114.226 ms
 3  2003:db8:0:f101::130 (2003:db8:0:f101::130)  124.353 ms  134.440 ms  144.573
 ms
[root@localhost Desktop]#
[root@localhost Desktop]#
[root@localhost Desktop]# clear
```

Figure 4.11 Trace route from Auth. server to client

At the same instant of time, if we capture the packet entering and leaving in the communication channel, the following output is obtained.



Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
165	141.229221	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
166	141.239463	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		770 ESP (SPI=0x5792c8cb)
167	141.255393	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		162 ESP (SPI=0x43c78116)
168	141.265636	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		410 ESP (SPI=0x43c78116)
169	141.275877	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		162 ESP (SPI=0x43c78116)
170	141.280428	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
171	141.290669	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
172	141.300911	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		170 ESP (SPI=0x5792c8cb)
173	141.306600	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		162 ESP (SPI=0x43c78116)
174	141.311152	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
175	141.316841	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		170 ESP (SPI=0x43c78116)
176	141.331636	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
177	141.341876	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		666 ESP (SPI=0x5792c8cb)
178	141.357808	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		162 ESP (SPI=0x43c78116)
179	141.368049	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		714 ESP (SPI=0x43c78116)
180	141.378292	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		162 ESP (SPI=0x43c78116)
181	141.382843	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
182	141.393083	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
183	141.403326	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
184	141.409014	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		162 ESP (SPI=0x43c78116)
185	143.410921	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		170 ESP (SPI=0x5792c8cb)
186	143.438231	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		170 ESP (SPI=0x43c78116)
187	143.451888	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
188	143.462128	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		674 ESP (SPI=0x5792c8cb)
189	143.479197	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		162 ESP (SPI=0x43c78116)
190	143.489438	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		618 ESP (SPI=0x43c78116)
191	143.499680	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		162 ESP (SPI=0x43c78116)
192	143.503095	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
193	143.513336	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
194	143.523578	2004:db8:0:f101::2	2004:db8:0:f101::1	ESP		162 ESP (SPI=0x5792c8cb)
195	143.530404	2004:db8:0:f101::1	2004:db8:0:f101::2	ESP		162 ESP (SPI=0x43c78116)

Figure 4.12 Packet Capture in the IPsec tunnel

The figure signifies that all the packet which is entering and leaving in the tunnel is fully encapsulated and encrypted. The source ip and destination ip which is the real physical interface ip of the router is shown to the external world. It means whether the client ping or have http request or grant other services, it is completely unknown to others. If we picked up one of the packet and tried to see the content inside of that packet, we could see the following contents.

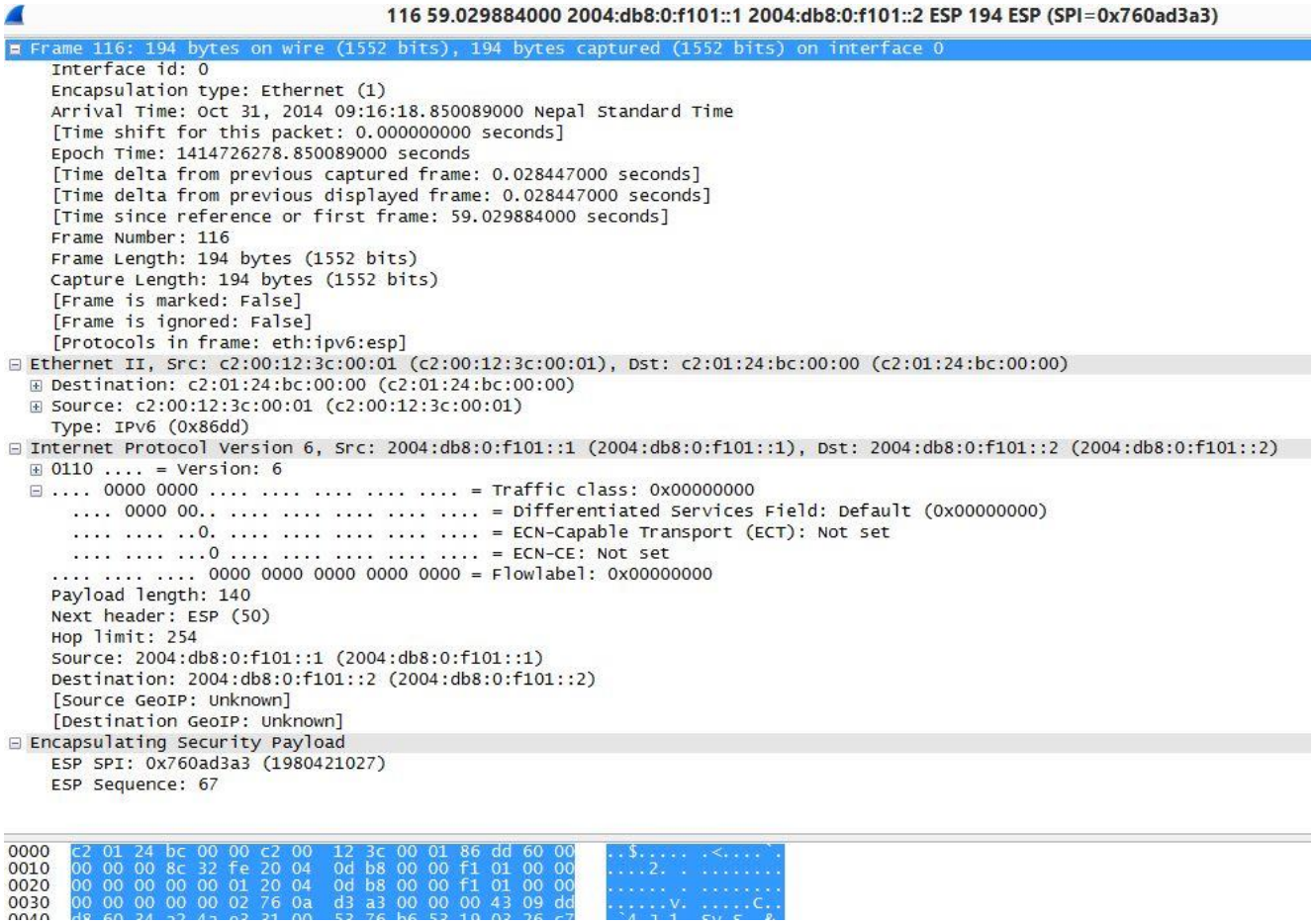


Figure 4.13 Packet Capture at communication channel

It only shows the real router interface ip from where the packet is leaving and coming. The actual data is not exposed to others. So, it prevents unauthorized access of packets and modification of the contents. The above packet capture shows that everything is encrypted and although the packet get transfer from client end to the authenticate server and web server, it only shows the tunnel end ip address. It prevents the intruder to know about the exact ip address of the client, web hosting ip address of web server and authenticate server.

CHAPTER FIVE: CONCLUSION AND RECOMMENDATION

5.1 Conclusion

From the above result it is found that source users/device can be track back and his/her/its identity can be identify and can prevent IP spoofing. Security is the prime issue in our modern world. So it is essential to avoid intruders to enter into secure system. On implementation of physically unclonable function on device which is commonly known as device signature, it really make easier, safer and quicker transaction or service grant to the authorized one. Moreover in this thesis, due to lack of hardware resources, it done in software (PHP) neglecting all the environmental effects so the delay of propagation may be varied and process of complete cycle is little bit longer that it is found. In this thesis, time to execute whole code is near about milliseconds, so the whole process time execution depend upon the transmission path if we would implement it in the real world scenario. Also if the number of devices/users/clients increases and more CRP are to be computed frequently, then it depends upon the server capacity like memory and CPU speed.

Moreover IPV6 is the next generation technology in networking system, and we could get benefit of built in security system of IPV6. Here, in my thesis, client is track back successfully and the packet capture at the communication channel are completely encrypted and encapsulated. However due to unavailability of the real hardware, it does not consider the external environmental effects like temperatures and hardware error is also not computed. It seemed that although there is 100 % success rate in track back of authentic and non-authentic client in this thesis, this success rate may be varied in real hardware implementation however the concept is still the same.

5.2 Recommendation

In this thesis, the web server, client and authenticate server are all created within the virtual machine which somehow use same hardware resources and memory resulting the outputs in one form that may be varied if one could use independent hardware entities. As the main principle of physically unclonable function relied upon the delay time of hardware which generate challenge response pairs (CRP), it is seemed that the outputs could be more analyzed and further research can be done in broad way on using real hardware. Moreover, the number of rows to generate CRP can be increased to get more

possible combination of CRP and also the input could be varied from two to higher order to get more complex output.

Although there are various factors that affects the performance of hardware devices, it is not considered here due to hardware resources unavailability. So, in the future, it is better to test this concept and enhanced this thesis to generate higher number of combination of challenge response pair and use it for source verification.

REFERENCES

[1] Yu Zhu, Jun Bi and Yayuan Sun. A light-weighted Source Address Validation Method in IPv4/IPv6 Translation. The Seventh International Conference on Internet Monitoring and Protection, 2012

[2] Al-Zobbi Mohammed. Comparision between IPv4 and IPv6 in Adopting Differentiated Services. International Journal of Science and Technology Research Volume 3, Issue 2, February 2014

[3] M. Leber, "Global IPv6 Deployment Progress Report". Hurricane Electric. 2010. http://bgp.he.net/ipv6-progress-report.cgi [retrieved: May, 2012]

[4] DePetr Fojtu,Vulnerabilities and Threats in IPv6 Environment,Department of Computer Science and Engineering,University of West Bohemia Faculty of Applied Sciences,2013

[5] K. Dmitriy,G.Gurtov. SAVAH: Souce Address Validation with Host Identity Protocol. Helsinki Institute for Information Technology, 2009.

[6] Nathan Beckmann and Miodrag Potkonjak, Hardware-Based Public-key Cryptography with Public Physically Unclonable Functions, 2009

[7] A. Maiti, J. Casarona, L. McHale, and P. Schaumont. A large scale characteriza- tion of RO-PUF. In 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pages 94—99, June 2010

[8] Michael S. Kirkpatrick, Sam Kerr, and Elisa Bertino. PUF ROKs: A hardware approach to read-once keys. In Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ASIACCS '11, pages 155—164, 2011.

[9] Kazuhiko Minematsu, Stefan Lucks, Hiraku Morita, and Tetsu Iwata. Crypt-analysis of EAXprime. Cryptology ePrint Archive, Report 2012/018, 2012.

[10] Zaur TARIGULIYEV, Reliability and security of arbiter based physically unclonable function, 2010

[11] Shohreh Sharif Mansouri and Elena Dubrova, Ring Oscillator Physical Unclonable Function with Multi Level Supply Voltages, 2012

[12] Q. Chen, G. Csaba, P. Lugli, and et Al., "The bistable ring puf: A new architecture for strong physical unclonable functions," in HOST, june 2011, pp. 134 –141.

[13] A. Maiti and P. Schaumont, "Improving the quality of a physical unclonable function using configurable ring oscillators," in FPL, 2009, pp. 703–707. [14] A.-R. Sadeghi and D. Naccache, Towards Hardware-Intrinsic Security: Foundations and Practice, 1st ed. Springer-Verlag New York, Inc., 2010.

BIBILOGRAPHY

1 R. Merkle, “Secrecy, Authentication, and Public Key Systems,” UMI Research Press, 1979.

2 R. Merkle, “A certified digital signature,” Advances in Cryptology, Proc. Crypto’89, LNCS 435, G. Brassard, Ed., Springer-Verlag, 1990, pp. 218–238.

3 http:/en.wikIPedia.org/

4 “PUF Based RFIDs”http://www.verayo.com/product/pufrfid.html

5 http://w3schools.com/