International Journal of Engineering Research and General Science Volume 3, Issue 2, March-April, 2015
ISSN 2091-2730

FPGA BASED OPTIMAL SECURED COMMUNICATION

Mr.G. Manikandan [1], Mr. M.Paramasivan [2] , Mr.N.Mathavan [3] ,Ms.X.Benedict priyanka [4]

[1]Research Scholar, St. Peter's University, St. Peter's, Institute of Higher Education and Research, Avadi,

Chennai-600 054, Tamilnadu, India

[2,3]Assistant Professor, Department of Electronics and Communication Engineering,

[2]Aksheyaa college of engineering ,Tamilnadu, India

[3]Nadar Saraswathi College of Engineering and Technology ,Tamilnadu, India

[4] UG Student, Department of Electronics and Communication Engineering,

Kodaikanal Institute of Technology, Tamilnadu, India

Email: mrg.manikandan@gmail.com

Abstract— In real time systems the FPGA can implement with soft core, hard core and many embedded applications. It is a major platform for reconfigurable, high execution speed and low power consumption. The design resources can also be reached by using this FPGA. This can be implemented by using blow fish algorithm (encryption and decryption) based on security purposes. Earlier we are implementing this process in single FPGA system for multiple process tasks with low operating speed. But now, the multiple FPGA system has been implemented over here for high execution speed. This process can be communicated through the RS232 communication link.

Keywords- Multi-protocol Label Switching (MPLS); Label Distribution Protocol (LDP); Last In First Out (LIFO);Synchronous Optical Networking (SONET).

I. INTRODUCTION

To design a cryptographic protocol to protect the Multi-protocol Label Switching (MPLS) header used in an Internet Service Provider (ISP) network. This protocol should protect the MPLS header primarily against tampering for purposes of hijacking ISP resources. Secondary goals are protection against replay attack and traffic analysis of ISP traffic. The protocol should be fast so as to minimize delay introduced into the high-speed MPLS routers.One goal has been to compile an introduction to the subject of cryptography. There exist a number of studies of various parts of the cryptographic standards, but complete treatments on a technical level are not as common. Material from papers, journals, and conference proceedings are used that best describe the various parts.Another goal has been to search for algorithms that can be used to implement the suitable cryptography for MPLS label switching.
A third goal is to evaluate their performance of various cryptographic protocols and to select a best protocol that can be implemented best for MPLS switching. These properties were chosen because they have the greatest impact on the implementation effort.

A final goal has been to design and simulate an cryptographic protocol. This should be done in C or MATLAB. The source code should be easy to understand so that it can serve as a reference on the standard for designers that need to implement a system

II.MULTIPROTOCOL LABEL SWITCHING (MPLS) HEADER

In computer networking and telecommunications, Multi-protocol Label Switching (MPLS) is a data-carrying mechanism, which emulates some properties of a circuit-switched network over a packet-switched network. MPLS operates at a OSI Model layer that is generally considered to lie between traditional definitions of Layer 2 (data link layer) and Layer 3 (network layer), and thus is often referred to as a "Layer 2.5" protocol. It was designed to provide a unified data-carrying service for both circuit-based clients and packet-switching clients, which provide a datagram service model. It can be used to carry many different kinds of traffic, including IP packets, as well as native ATM, SONET, and Ethernet frames. Fig 1.3 shows the MPLS Header in ISP network model.
A number of different technologies were previously deployed with essentially identical goals, such as frame relay and ATM. MPLS is

now replacing these technologies in the marketplace, mostly because it is better aligned with current and future technology and needs.

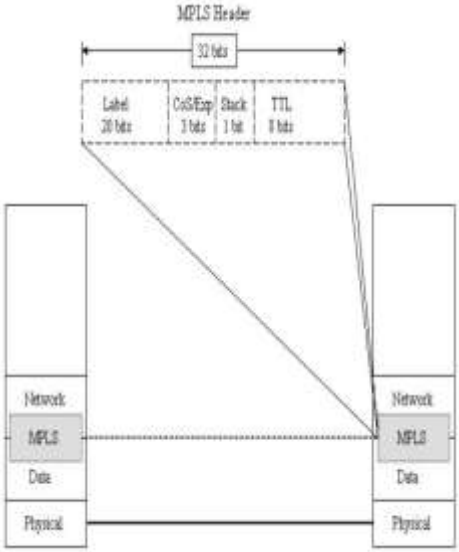


Fig 1 MPLS Header in ISP Network

In particular, MPLS dispenses with the cell-switching and signaling-protocol baggage of ATM. MPLS recognizes that small ATM cells are not needed in the core of modern networks, since modern optical networks (as of 2001) are so fast (at 10 gbit/s and well beyond) that even full-length 1500 byte packets do not incur significant real-time queuing delays (the need to reduce such delays, to support voice traffic, having been the motivation for the cell nature of ATM).

At the same time, it attempts to preserve the traffic engineering and out-of-band control that made frame relay and ATM attractive for deploying large scale networks.

MPLS was originally proposed by a group of engineers from Cisco systems, inc.; it was called "tag switching" when it was a Cisco proprietary proposal, and was renamed "label switching" when it was handed over to the IETF for open standardization.

One original motivation was to allow the creation of simple high-speed switches, since it was at one point thought to be impossible to forward IP packets entirely in hardware. However, advances in VLSI have made such devices possible. The systemic advantages of MPLS, such as the ability to support multiple service models, do traffic management, etc, remain.

III. PROBLEM STATEMENT

In conventional IP forwarding, the router uses a longest-prefix match on the destination IP address to determine where to forward a packet. With MPLS, labels are attached to packets at the ingress point to an MPLS network. Within the network, the labels are used to route the packets, without regard to the original packet header information. These labels can be stacked as a last in first out (LIFO) label stack, enabling MPLS flows to be combined for transport and separated later for distribution.

Current proposed protocols for MPLS security, Behringer [2] and Senevirathne et al. [3] discuss two approaches to securing MPLS. Behringer [2] makes the assumption that the core MPLS network is "trusted and provided in a secure manner." We make no such assumption in our work. We assume that only the MPLS nodes themselves are secure. The physical links connecting the nodes are assumed to not be secure – we protect them using our protocol. Senevirathne et al. [3] proposes an encryption approach using a modified version of IPsec. IPsec is defined by the IETF [4], and is an all-purpose encryption protocol that includes key distribution, authentication for the IP header, and authentication and encryption for the IP payload. Senevirathne et al. [3] translate these capabilities to an MPLS environment. Their proposed system does not meet the requirements specified above for our problem for two reasons:

1. It adds at least 128 bits to each MPLS header. This is four times the size of the MPLS header itself. This level of overhead on every packet would probably prove unacceptable to an ISP. It would almost certainly add significantly more processing delay to each packet when compared to a simple encryption scheme.

2. It does not encrypt the MPLS header (but provides authentication). Therefore the header is vulnerable to traffic analysis. We require fast and inexpensive operation, since MPLS routers are mainly routers without the full capability to do Layer 3 routing operations or are Layer 2 (ATM) switches with some additional capability. For this reason, application layer distribution designs are not applicable in our case. We also require the key exchange algorithm to be aware of the computation burden it imposes on the underlying system and communications performance

Currently, MPLS does not provide header or payload encryption. The only security function employed in MPLS is the use of MD5 [5] to sign and authenticate the control messages sent using TCP. MPLS control messages are transported using IP and do not fall under the scope of this research. They can be secured either by IPsec or any other proprietary method. Integration of the Label Distribution Protocol (LDP) security is an open issue for future study. Nevertheless, MD5 could be used for MPLS header security, since it is already present in the routers' software. MD-5 is particularly suitable in fast re-keying and for the hash or keyed-hash functions that may need to be used

IV.PERFORMANCE ANALYSIS OF ENCRYPTION ALGORITHMS

Our encryption system must be as fast as possible (to minimize processing delay) while meeting the stated security objectives. The protocol should not add bits to the MPLS header or require an additional header to be inserted into each data packet. MPLS routers must be able to recognize a valid, decrypted (received) MPLS header. There will only be a small number of different MPLS headers exchanged between nodes, however there will be potentially hundreds of millions of copies of these same headers exchanged. Therefore, the protocol must not encrypt the same plaintext to the same ciphertext. This, along with the fact that each MPLS header is only 32 bits long implies that some kind of stream protocol is necessary. Packets can be lost or damaged in transmission, so the encryption protocol must be self-synchronizing. This further implies that a cipher feedback mode of operation is required.

Each MPLS router in an MPLS network must be able to read and change the label in the MPLS headers it processes. Therefore, a link encryption scheme is necessary. A disadvantage of link encryption is that the MPLS header message is vulnerable at each router. A link encryption device must be present at each end of the link. Each pair of nodes that share a link could have a unique key with a different key for each link. As an alternative, the same key can be used for all links in an administrative domain and different keys can be used at the edges of the domain. This can expose a large number of nodes (belonging to the same domain) if the key is stolen.

	Key Length (Nk words)	Block Size (Nb words)	Number of Rounds (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Table 1 Key-Block-Round Combination

V. IMPLEMENTATION OF THE SYSTEM

Blowfish is a variable-length key, 64-bit block cipher. The algorithm consists of two parts: a key-expansion part and a data-encryption part. Key expansion converts a key of at most 448 bits into several sub-key arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network. Each round consists of a key-dependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round. Encryption and Decryption Blowfish is a Feistel network consisting of 16 rounds. The input is a 64-bit data element, x.

5.1 Design Considerations

In our design, we consider the types of attacks we intend to protect from, and operational considerations including speed, span of encryption, and key distribution. What kinds of attacks are we protecting from? We are protecting the header in an ISP provider's network so that an attacker cannot collect and analyze traffic data, understand route configuration, and eventually create a covert channel. This protocol will protect the links between MPLS routers by protecting the MPLS headers. All MPLS headers will be encrypted with this protocol.

There are four general categories of attacks described by Stallings [6]: Interruption -- An asset on the system is destroyed or becomes unavailable. This does not have to be a physical asset. This is an attack on availability. We are not protecting from this type of attack with our protocol. Interception -- An unauthorized party gains access to an asset and can capture data. This is an attack on confidentiality. We are not protecting from this type of attack with our protocol. Modification -- An attacker modifies the contents of a message. This is an attack on integrity. We are protecting the MPLS header contents with our protocol. Fabrication -- An attacker inserts counterfeit objects in to the system. This is an attack on authenticity. We are protecting an MPLS network from this type of attack with our protocol.

Our encryption system must be as fast as possible (to minimize processing delay) while meeting the stated security objectives. The protocol should not add bits to the MPLS header or require an additional header to be inserted into each data packet. MPLS routers must be able to recognize a valid, decrypted (received) MPLS header. There will only be a small number of different MPLS headers exchanged between nodes, however there will be potentially hundreds of millions of copies of these same headers exchanged. Therefore, the protocol must not encrypt the same plaintext to the same ciphertext.

5.2 Speeding Processing:

If the Blowfish algorithm was to prove to add too much processing delay to the MPLS routers, we could use a reduced number of

rounds to speed the processing. This would likely reduce the security afforded by 16 rounds; however, the published research to date has not been able to break five rounds or more of Blowfish. There is also a trade-off to be made between the time required to break Blowfish (by brute force) and the time between re-keying. If we are willing to re-key more frequently, we can use fewer rounds in our Blowfish encryption and speed the processing at each MPLS router.

Determining if Decrypted MPLS header is ValidThe method of determining if an MPLS header is valid is to check if the label portion of the header is a valid MPLS label in the context of that particular router. Unfortunately, if we are using a large number of labels there is a small but significant chance that an intruder's MPLS header or a valid header with an error will produce a valid label (though a random one). For example, if we are using1024 labels there is a one in a thousand chance for a random "encrypted" label to produce a valid decrypted label. Our proposed solution to this problem is to "steal" bits from the CoS and TTL fields to increase our chances of detecting a bad MPLS header. Using the three bits from the CoS field (which assumes we create separate paths for each class of service and therefore do not need these bits) and two bits from the TTL field would improve our odds of detecting bad MPLS headers. Redoing the example above using the "stolen" bits, we determined that 15 in 1,000,000 bad MPLS headers would decrypt to a valid label. Changing the expected bit pattern of the unused bits from zero to some other value will not improve the security or ability to detect a valid header, since any value has an equally likely chance of occurring.

5.3 Key Exchange System

A cryptographic key exchange method developed by Whitfield Diffie and Martin Hellman in 1976. Also known as the "Diffie-Hellman-Merkle" method and "exponential key agreement," it enables parties at both ends to derive a shared, secret key without ever sending it to each other.

Using a common number, both sides use a different random number as a power to raise the common number. The results are then sent to each other. The receiving party raises the received number to the same random power they used before, and the results are the same on both sides.

The simplest, and original, implementation of the protocol uses the multiplicative group of integers modulo p, where p is prime and g is primitive mod p. Modulo (or mod) simply means that the integers between 1 and p − 1 are used with normal multiplication, exponentiation and division, except that after each operation the result keeps only the remainder after dividing by p.

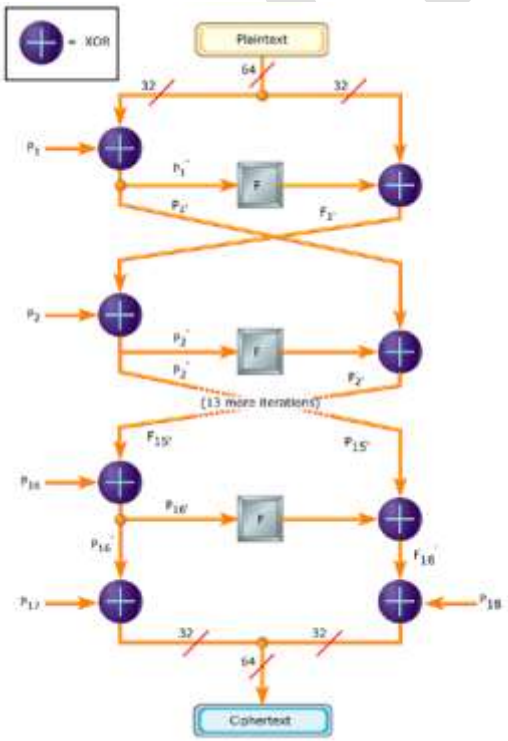


Fig 2 Blowfish Algorithm Fiestel Network.

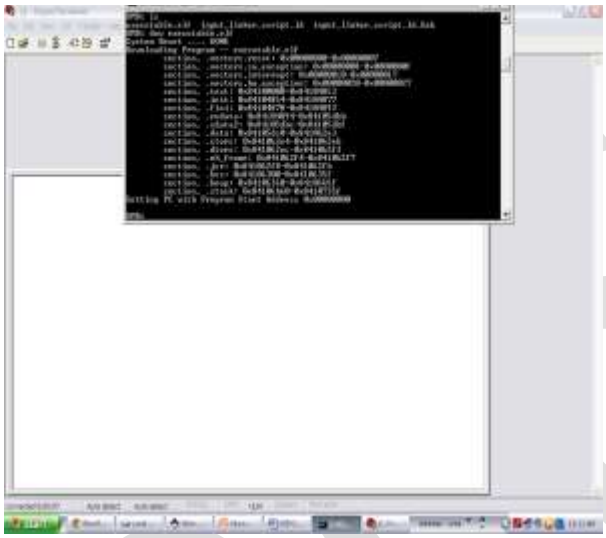
Both Alice and Bob have arrived at the same value, because gab and gba are equal. Note that only a, b, gab and gba are kept secret. All the other values are sent in the clear. Once Alice and Bob compute the shared secret they can use it as an encryption key, known only to them, for sending messages across the same open communications channel. Of course, much larger values of a,b, and p would be needed to make this example secure, since it is easy to try all the possible values of gab mod 23 (there will be, at most, 22 such values, even if a and b are large). If p was a prime of more than 300 digits, and a and b were at least 100 digits long, then even the best known algorithms for finding a given only g, p, and ga mod p (known as the discrete logarithm problem) would take longer than the lifetime of the universe to run. g need not be large at all, and in practice is usually either 2 or 5.

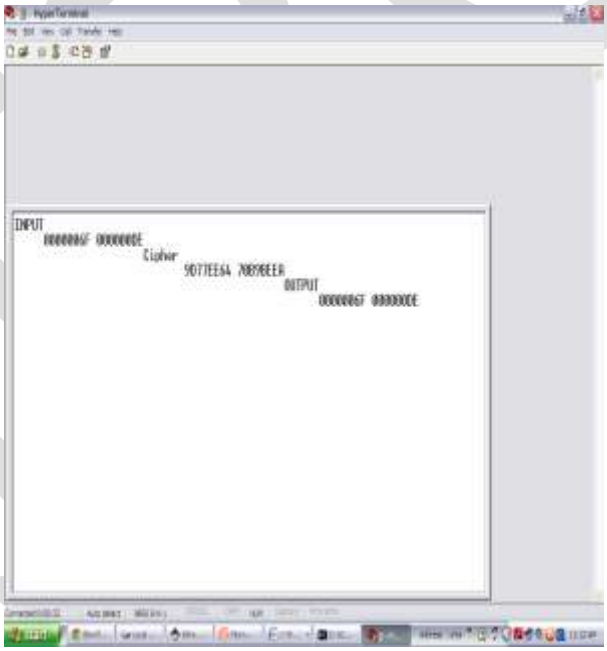
5.4 Authentication Diffie Hellman Key Exchange

In the original description, the Diffie-Hellman exchange by itself does not provide authentication of the parties, and is thus vulnerable to man in the middle attack. The man-in-the-middle may establish two distinct Diffie-Hellman keys, one with Alice and the other with Bob, and then try to masquerade as Alice to Bob and/or vice-versa, perhaps by decrypting and re-encrypting messages passed between

them. Some method to authenticate these parties to each other is generally needed.
A variety of cryptographic authentication solutions incorporate a Diffie-Hellman exchange. When Alice and Bob have a public key infrastructure they may digitally sign the agreed key, or ga and gb, as in MQV, STS and the IKE component of the IPsec protocol suite for securing Internet Protocol communications. When Alice and Bob share a password, they may use a password-authenticated key agreement form of Diffie-Hellman

RESULTS





CONCLUSIONS

Our proposed cryptographic system for the protection of MPLS headers would likely prove successful at its primary task – preventing theft of services by an intruder with interior access to an ISP's MPLS network. Our encryption system uses the (to date) very strong Blowfish algorithm. Without actually implementing and testing this system, we cannot know for sure what delay our encryption protocol would produce in an MPLS router. This delay would be a critical issue for ISP's supporting Quality of Service protocols, like Differentiated Services. Our key distribution system strives to be as simple as possible to administer, yet provide a high level of security for the keys themselves.

REFERENCES:

[1] Rosen, E., Viswanathan, A., and Callon, R. "Multiprotocol Label Switching Architecture," IETF RFC 3031, January 2001.
[2] Behringer, M., "Analysis of the Security of the MPLS Architecture," Internet Draft, IETF Network Working Group, February 2001.
[3] Senevirathne, T. and Paridaens, O., "Secure MPLS – Encryption and Authentication of MPLS
[4] Kent, S. and Atkinson, R., "Security Architecture for the Internet Protocol," IETF RFC 2401, November 1998.
[5] Rivest, R., " The MD5 Message-Digest Algorithm," IETF RFC 1321, April 1992.
[6] Stallings, W., Cryptography and Network Security Principles and Practice, 2nd Edition, Prentice Hall, New Jersey, 1999.
[7] Schneier, B., Blowfish symmetric block cipher, 1993.
[8] Counterpane Internet Security Web Site, Copyright Counterpane Internet Security, Inc., 2001.
[9] Boyd C., "A Framework for Design of Key Establishment Protocols," IEEE ACISP, 1996.
[10] Ellison, C. and Schneier, B., "Ten Risks of PKI: What You're not BeingTold about Public Key Infrastructure," Computer Security Journal, Volume XVI, November 2000