Analysis of Network Encryption Schemes

John Mark Matson
Department of Electrical & Computer Engineering,
Oregon State University, Corvallis, Oregon 97331 -USA.
E-mail: matson@ece.orst.edu
May 22, 2002

Abstract— **This paper attempts to shed light onto the topic of hardware implementations of network security and their relevance to IPSec. Several research papers are analyzed to gain an understanding of this field. Architecture, security level, speed, key agility, and IPSec weaknesses are a few of the topics covered. The goal is to educate the reader about the hardware details required to generate network security.**

I. Introduction

The rapid expansion of the Internet over the past decade has benefited society in many ways. Individuals have a level of access to information unlike any generation before, and they are allowed to buy, sell, and trade with ease. Businesses, by way of virtual private networks, are able to span their resource centers over a large geographical location.

This freedom, however, does not come without risk. Security risks are present in almost any use of the Internet, and creating a secure environment for information exchange is a non-trivial task. Hackers may be interested in something small, like reading a persons email, to something complex as session hijacking. These concerns have generated interest in the applications of cryptography for the Internet.

Many individuals and business use cryptography to insure secure transfer of important data, such as emails and banking transactions. However this covers only a small portion of Internet traffic and does not insure that entire sessions are safe.

Recent attention has been paid to IP Security (IPSec) and its role in creating secure Internet sessions. IPSec is the encryption of data on the transport layer using a secret key generated via a key exchange algorithm (Diffie-Hellman) between the two parties. This results in the encryption of every IP packet as it leaves the source and is then decrypted at the destination.

This paper attempts to shed light onto the topic of hardware implementations of network security and their relevance to IPSec. Several research papers are analyzed to gain an understanding of this field. Architecture, security level, speed, key agility, and IPSec weaknesses are a few of the topics covered.

The goal is to educate the reader about the hardware details required to generate network security.

II. Architectures & Performance

The design choice of a cryptographic architecture takes into account factors such as speed, level of security, and area. These are all important factors for a good IPSec system.

DES

Until recently the AES (Advanced Encryption Standard) had been DES (Data Encryption Standard). DES has proved, until recently, to be a secure method of encryption, and is commonly used by industry and the government. A survey put forth by Sandia National Labs (SNL) showed that previous implementations of DES were unable to cross the 0.5 Gbps threshold. [1] Current high-speed networks are in the range of 1-10 Gbps, therefore previous implementations were not suitable for network packet encryption. SNL researchers designed an ASIC specifically for the use of IPSec, using the DES architecture. By fully pipelining (16 stages) the ASIC, SNL was able to solve the problem of key agility. Here, key agility means the ability to encrypt data with one key on one clock cycle, and on the next clock cycle encrypt new data with a different key. This is a common problem in IPSec because many packets are arriving and they all may have different destinations, and thus requires encryption via different keys.

One of the main performance barriers was the feedback mode of operation called Cipher Block Chaining (CBC). This is where the output of the pipeline must be combined with previous rounds. This requires the pipeline to be empty at the start of every encryption. Therefore, the throughput for the pipeline was only 0.08 Gbps. To overcome this performance issue, SNL used a non-feedback mode called Electronic Codebook (ECB). However, ECB is inherently less secure than CBC.

The SNL DES ASIC was targeted for ATM (Asynchronous Transfer Mode) traffic that uses 384bit packet lengths. Using ECB, and running 6 pipelines in parallel (64-bits each), SNL was able to achieve 6.7 Gbps, and simulations showed the chip capable of operating at up to 9.28 Gbps.

Furthermore, the SNL designers believe that improvements could boost the speed in future productions of the ASIC. Additional pipeline stages, higher performance IO buffers, source synchronous clocking, and Gallium Arsenide (GaAs) technology would push the performance close to 160 Gbps.

However, DES and Tripple DES are no longer considered to be secure enough for future applications. As a result, the NIST (National Institute of Standards and Technology) decided to evaluate new possible solutions for the AES.

CRYPTON

One such proposed candidate for the AES was CRYPTON. [2] This 128-bit block encryption algorithm is touted as being very "hardware-friendly" in terms of speed and area.

The basic premise of CRYPTON is a substitution-permutation network cipher based on the structure of a square. Each 128-bit block (4x4 byte array) is processed using a sequence of round transformations. Each round consists of four steps that take place in parallel - byte-wise substitution, column-wise bit permutation, column-to-row transposition, and key addition. The encryption process involves 12 repetitions on the same round transformation.

A key to making the CRYPTON algorithm work efficiently is the parallel execution of key generation and data encryption. This means that per-computation and storage of round keys are not required, which cuts down on cycle time and memory size.

The 12 rounds can be "loop unrolled" into smaller rounds at the cost of additional hardware. The authors tried several different cases and found that the speed and area varied significantly. A two-round model took 28K gates and ran at 1.66 Gbps, and the full-round model took 94K gates and ran at 2.69 Gbps. Both of which are speeds that could keep up with a giga-bit network protocol.

RIJNDAEL

In October of 2000 the NIST choose the Rijndael algorithm as the successor to DES as the new AES. The Rijndael algorithm proved highly secure and highly flexible, allowing variable key and block lengths of 128, 192, and 256 bits. Furthermore, it allows encryption rounds of 10, 12, and 14.

Researchers at UCLA [3] proposed an ASIC implementation of Rijndael for high-speed encryption. The algorithm is broken into two main sections, encryption and key generation. Like previous algorithms, key generation in Rijndael is done on the fly to avoid large buffers to store generated keys. Likewise, pipelining and unrolling are not implemented in the system.

The encryption process goes through four steps for each round. First a S-box substitution is done, followed by a row shift, followed by a row-column GF($2\hat{8}$) multiplication, and lastly the data is XORed with the key to generate the output cipher text.

In their implementation there is only hardware for one encryption that is reused for each round. Furthermore, encryption was designed to operate in one clock cycle. This was to ensure that the design ran at the lowest clock frequency possible with the same throughput. As a result, lookup tables had to be duplicated in order to insure encryption in one clock cycle, and the designers chose several 256 entry asynchronous ROM modules to handle this task.

Their results showed that a design of 173K gates produced and optimal speed of 1.82 Gbps. Interestingly enough, the critical path was 10ns and was not in the encryption module, but rather in the key generation module. The encryption module took only 6ns. The designers felt that they could reduce the delay of the key generation module, but at the sacrifice of more rounds.

In order to achieve the kinds of speeds needed for very high-speed networks (multiple gigabit), designers may have to sacrifice security for performance.

Designers of an FPGA version of Rijndael chose to sacrifice security and used ECB mode in order to fully pipeline the algorithm. [4] A pipelined version requires large amounts of area due to registers, as a result, the designers chose the Virtex-E (Extended memory) FPGA series by Xilinx because it has built in RAM blocks. Their results showed that with 128-bit key lengths they could achieve approximately 7 Gbps on an FPGA and with variable key lengths they could read 3.2 Gbps.

A similar study done by Weeks, Bean, Rozylowiez, and Ficke showed that a fully pipelined, variable key length, CMOS implementation of Rijndael could reach 5 Gbps. [5]

III. IPSec Weaknesses

Even though virtually all information is encrypted in IPSec, there are still fundamental issues that aid hackers in their quest to penetrate security systems. One cannot forget that TCP/IP is a protocol. As such, users must adhere to the guidelines set forth by the protocol. This opens the system up for various attacks.

Steven Bellovin at AT&T Research Labs performed a study [6] to analyze the weaknesses of IPSec. His research showed that as few as one or two packets could be enough to successfully attack a session that was encrypted using DES.

For instance take a single-packet attack. Measurements have shown that 30-40% of all TCP/IP packets are 40-byte TCP ACK (acknowledge) packets. The first word in one such packet is the replay counter. This number starts at 1 and goes up from there. If the attacker intercepts the packet at the beginning of the transmission, then 30 or even 32 bits are known. Next is the IP header. Inside the IP header are some very predictable numbers. The version is always 4, the length is almost always 5, and the type-of-service field is generally 10.

Further analysis of the TCP header renders, under favorable circumstances, 88 bits of the 160 bits predictable. Hence, an ACK packet would be a prime target for a crypt-analysis device. If the analysis is increased to cover two packets, it is shown that of the 160 bits, about 124 bits of probably plaintext can be predicted.

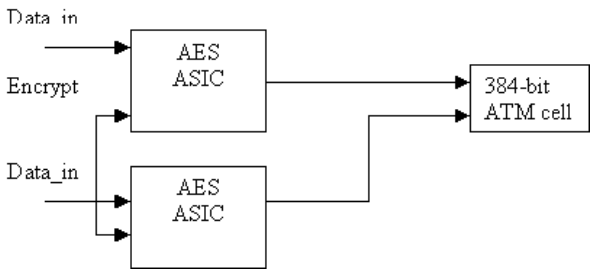
Although, probable plaintext attacks are not carried out on a whim. They require expensive, special-purpose hardware, and a lengthy setup and monitoring time. Furthermore, if the hosts change keys on a somewhat regular basis (on the order of minutes) an attacker would not be able to monitor traffic flow, and might - at best - get bits and pieces of a session.

IV. AES Rijndael ASIC Implementation for ATM Networks

In response to the need for a high-speed encryption/decryption scheme for the Internet, I propose an AES ASIC for ATM networks. The goal is to develop a very high-speed implementation of AES Rijndael algorithm for

the use of packet level security of an ATM Network. ATM was chosen as a starting point because it is a fixed, small-packet length protocol that usually runs at high speeds. ATM is a popular protocol in businesses, and would provide an excellent starting point toward an Internet security device. An ATM cell breaks down into a 5-byte header and a 48-byte (384-bit) payload. The 384-bits of data break down into two 192-bit chunks, which are the right size for encryption using two ASICs in parallel.

Figure 1: AES ASIC Architecture.



We chose the Rijndael algorithm because it has received the approval of the NIST as the cryptographic algorithm of the future. It has been shown that the use of hardware devices for brute force decryption coupled with the predictability of network packets can drastically decrease the time adversaries require to break cryptographic algorithms. However we opted to use ECB mode rather than CBC mode because added encryption time would not make it possible to achieve the high throughputs needed for network packets. Our ASIC touts such features as single cycle encryption, fully pipelined implementation, on the fly generation of sub-keys, variable bit key lengths, and encryption/decryption on the same chip. Furthermore, the encryption requires a fixed data length (192-bit) and thus simplifies the design.

The S-box transformation was made by breaking down the 192bits of data into 24 8-bit chunks, and each of them is the address for a S-box table lookup. New zero-access-time memory cells were used for the S-box memory cells, and fabrication took place in another galaxy and was performed using a high-speed GaAs process on a 10nm technology.

The result was an ASIC that had an encryption cycle time of 1ns and a corresponding throughput of just over 104 Gbps. Using two in parallel to encrypt 384-bit payloads require a little overhead and resulted in an astounding throughput of 190 Gbps.

V. Conclusions

Security is a rising concern in modern society, and the Internet is no exception. IPSec provides a much-needed level of security over complete sessions on the Internet and not just single pieces of data. New advances in dedicated hardware cryptographic algorithms have allowed the possibility of dynamic, transport level encryption of data at very high speeds. It is the author's belief that IPSec will become a standard in the not too distant future.

References

[1] Perry Roberston et al. D. Craig Wilcox, Lyndon Pierson, "A des asic suitable for network encryption at 10 gbps and beyond," *Cryptographic Hardware and Embedded Systems (CHES)*, pp. 37–48, August 1999.

[2] J.-H. Chung E. Hong and C. H. Lim, "Hardware design and performance estimation of the 128-bit block cipher crypton," *Cryptographic Hardware and Embedded Systems (CHES)*, pp. 49–60, August 1999.

[3] Henry Kuo and Ingrid Verbauwhede, "Architectural optimization for a 1.82 gbit/sec vlsi implementation of the aes rijndael algorithm," *Cryptographic Hardware and Embedded Systems (CHES)*, pp. 51–64, May 2001.

[4] Maire McLoone and J. V McCanny, "High performance single-chip fpga rijndael algorithm implementations," *Cryptographic Hardware and Embedded Systems (CHES)*, pp. 65–76, May 2001.

[5] Tom Rozylowiez Bryan Weeks, Mark Bean and Chris Ficke, "Hardware performance simulations of round 2 advanced encryption standard algorithms," *AES3: The Third Advanced Encryption Standard (AES) Candidate Conference*, April 2000.

[6] Steven M. Bellovin, "Probable plaintext cryptanalysis of the ip security protocols," *Symposium on Network and Distributed Systems Security*, 1997.