

LỜI NÓI ĐẦU

Để hoàn thành đề cương nghiên cứu khoa học, tôi xin gửi lời cảm ơn chân thành tới Thầy hướng dẫn – TS. Trương Quang Vinh đã tận tình hướng dẫn, giảng dạy, cung cấp tài liệu và phòng lab để tôi có thể hoàn thành đề tài nghiên cứu khoa học này.

Tôi xin chân thành cảm ơn quý Thầy Cô của Bộ môn Điện tử - Viễn thông nói riêng, Khoa Điện điện tử nói chung đã truyền đạt cho tôi những kiến thức nền tảng thiết yếu để tôi có thể hoàn thành đề cương nghiên cứu khoa học một cách thuận lợi.

Một lần nữa, tôi xin chân thành gửi lời cảm ơn.

Tp. Hồ Chí Minh, ngày tháng năm 2019

MỤC LỤC

| | |
|---|----|
| LỜI NÓI ĐẦU | i |
| DANH MỤC HÌNH ẢNH | iv |
| DANH MỤC BẢNG BIỂU | v |
| TỪ VIẾT TẮT | vi |
| CHƯƠNG 1 | 1 |
| I. Lý do chọn đề tài | 1 |
| II. Đối tượng và phạm vi nghiên cứu | 2 |
| 1. <i>Đối tượng nghiên cứu</i> | 2 |
| 2. <i>Phạm vi nghiên cứu</i> | 3 |
| CHƯƠNG 2 | 4 |
| I. Những vấn đề tồn đọng trong nghiên cứu IPsec hiện thực hoá trên phần cứng FPGA | 4 |
| 1. <i>Nâng cao tốc độ</i> | 4 |
| 2. <i>Tối ưu hoá tài nguyên</i> | 5 |
| 3. <i>Vấn đề về bảo mật</i> | 7 |
| 4. <i>Tối ưu hoá năng lượng tiêu thụ</i> | 7 |
| II. Nghiên cứu của Esam Khan từ đại học Victoria, Victoria, BC, Canada | 8 |
| III. Thiết kế làm thông lượng cao cho giải thuật hàm băm SHA-1 của Jae-woon Kim từ Hanyang University Hàn Quốc | 11 |
| IV. Thiết kế một SoC đã lỗi IPsec thời gian thực cho lưu lượng Internet | 13 |
| CHƯƠNG 3 | 17 |
| I. Nghiên cứu giao thức IPsec | 17 |
| 1. Giới thiệu IPsec | 17 |
| 2. Mục đích của IPsec (IP Security Protocol) | 18 |
| 3. Những tính năng của IPsec (IP Security Protocol) | 19 |
| 4. Cấu trúc bảo mật | 21 |
| 5. Giao thức sử dụng trong IPsec | 22 |
| 6. Các phương thức IPsec | 26 |
| 7. Internet Key Exchange (IKE) | 28 |
| 8. Chính sách bảo mật IPsec | 33 |
| 9. Mối quan hệ giữa chứng chỉ và IPsec | 33 |
| II. Nghiên cứu các cấu trúc phần cứng IPsec thực hiện | 34 |
| 1. IPsec Gateway | 34 |

| | |
|-------------------------------------|----|
| 2. SECURE HASH ALGORITHM -3 (SHA-3) | 35 |
| 3. Kiến trúc IPsec AH | 35 |
| 4. Kiến trúc IPsec ESP | 39 |
| CHƯƠNG 4 | 43 |
| Tài liệu tham khảo | 44 |

DANH MỤC HÌNH ẢNH

| | |
|--|----|
| Hình 1 - Tốc độ dữ liệu, thời gian thực hiện hàm băm | 5 |
| Hình 2 - Mô hình cấu trúc chia sẻ tài nguyên | 6 |
| Hình 3 - Đồ thị sai khác thời gian giữa dùng và không dùng AH protocol | 7 |
| Hình 4 - Đồ thị sai khác thời gian giữ dùng không dùng ESP protocol | 7 |
| Hình 5 - Mô hình hoạt động đơn gian của Clock gating sử dụng cổng AND | 8 |
| Hình 6 - Lưu đồ giải thuật unified hash | 10 |
| Hình 7 - Kiến trúc lõi SHA-1 | 11 |
| Hình 8 - Mô hình OSI (Open Systems Interconnection Reference Model) | 17 |
| Hình 9 - Mô hình AH (Authentication Header) | 23 |
| Hình 10 - AH Packetization Process | 24 |
| Hình 11 - Mô hình ESP (Encapsulated Security Payload Protocol) | 24 |
| Hình 12 - ESP Packetization Process | 25 |
| Hình 13 - Hai phương thức truyền dẫn | 26 |
| Hình 14- Giao thức AH trong phương thức vận chuyển | 27 |
| Hình 15 - Giao thức ESP trong phương thức vận chuyển | 27 |
| Hình 16 - Giao thức ESP trong phương thức đường hầm | 27 |
| Hình 17 - Giao thức AH trong phương thức đường hầm | 28 |
| Hình 18 - Giao thức IKE | 29 |
| Hình 19 - Mô hình Main mode | 31 |
| Hình 20 - Giai đoạn I dành cho cả Main Mode và Aggressive Mode | 31 |
| Hình 21 - Quick Mode | 32 |
| Hình 22 - New Group Mode | 32 |
| Hình 23 - Kiến trúc IPsec Gateway | 34 |
| Hình 24 - Cấu hình lại lõi IPsec AH | 36 |
| Hình 25 - Sơ đồ thực hiện đề xuất | 37 |
| Hình 26 - Các bước xử lý AH | 37 |
| Hình 27 - Kiến trúc tính toán HMAC | 39 |
| Hình 28 - Cấu hình lại lõi IPsec ESP | 39 |
| Hình 29 - Các bước thực hiện của IPsec ESP | 40 |
| Hình 30 - Các bước thực hiện ESP | 42 |

DANH MỤC BẢNG BIỂU

| | |
|--|----|
| Bảng 1 - Kết quả thực hiện bộ xử lý IPsec với 8x8 thanh ngang | 4 |
| Bảng 2 - Kết quả mô phỏng của IPSEC với 8x32 thanh ngang | 5 |
| Bảng 3 - Tài nguyên sử dụng trước và sau khi thực hiện ghép | 6 |
| Bảng 4 - Thống kê tài nguyên sử dụng cho việc sử dụng và không sử dụng DPR Mode (DPR Mode and Full-Cryptography IP Core) | 7 |
| Bảng 5 - So sánh chức năng của giao thức AH và ESP | 26 |
| Bảng 6 - Tải trọng bảo mật đóng gói (ESP) | 41 |

TỪ VIẾT TẮT

CHƯƠNG 1

MỞ ĐẦU

I. Lý do chọn đề tài

Với nhu cầu trao đổi thông tin, giao dịch,... bắt buộc các cơ quan, tổ chức phải hoà nhập vào mạng toàn cầu – Internet. An toàn và bảo mật thông tin là một trong những yếu tố quan trọng hàng đầu trong việc sử dụng hệ thống mạng kết nối thông tin toàn cầu hiện nay. Ngày nay, các biện pháp an toàn thông tin cho máy tính cá nhân cũng như các mạng nội bộ đã được nghiên cứu và triển khai. Tuy nhiên, vẫn thường xuyên có các mạng bị tấn công, xâm nhập không mong muốn, có các tổ chức bị đánh cắp thông tin,... gây nên những hậu quả vô cùng nghiêm trọng.

Những vụ tấn công này nhằm vào tất cả các máy tính có kết nối Internet, các máy tính của các công ty lớn, tập đoàn đa quốc gia như AT&T, IBM, các trường đại học và các cơ quan nhà nước, các tổ chức quân sự, ngân hàng,... một số vụ tấn công với quy mô khổng lồ (có tới 100000 máy tính bị tấn công). Hơn thế nữa những con số này chỉ là phần nổi của tảng băng trôi.

Không thể phủ nhận cách mạng công nghệ 4.0 lan rộng toàn thế giới đã và đang khai thác được nguồn lực to lớn từ Internet nhất là trong lĩnh vực thông tin truyền thông, kinh tế, dịch vụ. Các thành tựu mới về thanh toán điện tử, trao đổi dữ liệu điện tử và các hệ thống tự động thu thập dữ liệu,... giúp kinh doanh điện tử, thương mại điện tử có những bước tiến đột phá trong các năm gần đây.

Tuy nhiên, trong hai năm 2017 và 2018, số lượng lỗ hổng an ninh trong các phần mềm, ứng dụng được công bố tăng đột biến với hơn 15.7000 lỗ hổng, gấp khoảng 2,5 lần những năm trước đó. Đặc biệt, nhiều lỗ hổng nghiêm trọng xuất hiện trên các phần mềm phổ biến như Adobe Flash Player, Microsoft Windows,... và cả trong nhiều dòng CPU của Intel, Applem, AMD,...

Mặc dù các bản vá an ninh đều nhanh chóng được các nhà sản xuất công bố sau khi lỗ hổng xuất hiện, nhưng việc cập nhật bản vá lại chưa kịp thời để giải quyết các lỗ hổng lớn và tối thiểu hoá thiệt hại cho danh nghiệp, cá nhân,... việc cập nhật bản vá thậm chí sau đó nhiều năm vẫn chưa được cập nhật. Điển hình như lỗ hổng SMB, sau hai năm vẫn có tới hơn 50% máy tính tại Việt Nam chưa được vá lỗ hổng này. Đây là lỗ hổng từng bị khai thác bởi mã độc mã hoá tổng tiền WannaCry, lây nhiễm hơn 300.000 máy tính trên thế giới trong vài giờ. Việc cập nhật bản vá chưa kịp thời tạo điều kiện cho hacker lợi dụng lỗ hổng để tấn công hệ thống mạng, từ đó lây nhiễm virus, cài cắm vào phần mềm gián điệp, thực hiện tấn công có chủ đích APT.

Mã độc sử dụng trí tuệ nhân tạo AI có thể xuất hiện trong năm 2019.

Mã độc sử dụng trí tuệ nhân tạo AI có thể xuất hiện trong những năm tới, ban đầu dưới hình thức những mẫu thử nghiệm PoC (Proof of Concept). Tuy nhiên, mối đe dọa lớn nhất của người dùng Internet đến từ mã độc mã hoá tổng tiền, mã độc xóa dữ liệu, mã độc đào tiền ảo

và tấn công APT. Các loại mã độc này có thể kết hợp nhiều con đường lây nhiễm khác nhau để tăng tối đa khả năng phát tán, trong đó phổ biến nhất là khai thác lỗ hổng phần mềm, hệ điều hành và qua email giả mạo.

IP Security (IPSec – Internet Protocol Security) là một giao thức được chuẩn hoá bởi IETF (Internet Engineering Task Force) từ năm 1998 nhằm mục đích nâng cấp các cơ chế mã hoá và xác thực thông tin cho chuỗi thông tin truyền đi trên mạng bằng giao thức IP. Hay nói cách khác, IPSec là sự tập hợp của các chuẩn mở được thiết lập để đảm bảo sự an toàn và bảo mật dữ liệu, đảm bảo tính toàn vẹn dữ liệu và chứng thực dữ liệu giữa các thiết bị mạng.

IPSec cung cấp một cơ cấu bảo mật ở tầng 3 (Network layer) của mô hình OSI vì mọi giao tiếp trong một mạng trên cơ sở IP đều dựa trên các giao thức IP. Do đó, khi một cơ chế bảo mật cao được tích hợp với giao thức IP, toàn bộ mạng được bảo mật bởi vì các giao tiếp đều đi qua tầng 3 (Network layer) trong mô hình OSI. IPSec được thiết kế như phần mở rộng của giao thức IP, được thực hiện thống nhất trong cả hai phiên bản IPv4 và IPv6. Đối với IPv4, việc áp dụng IPSec là một tùy chọn, nhưng đối với IPv6, giao thức bảo mật này được triển khai bắt buộc.

Khi IPSec được triển khai trên bức tường lửa hoặc bộ định tuyến của một mạng riêng, thì tính năng an toàn của IPSec có thể áp dụng cho toàn bộ vào ra mạng riêng đó mà các thành phần khác không cần phải xử lý thêm các công việc liên quan đến bảo mật.

IPSec được thực hiện bên dưới lớp TCP và UDP, đồng thời nó hoạt động trong suốt đối với các lớp này. Do vậy không cần phải thay đổi phần mềm hay cấu hình lại các dịch vụ khi IPSec được triển khai.

IPSec có thể được cấu hình để hoạt động một cách trong suốt đối với các ứng dụng đầu cuối, điều này giúp che giấu những chi tiết cấu hình phức tạp mà người dùng phải thực hiện khi kết nối đến mạng nội bộ từ xa thông qua mạng Internet.

Tuy nhiên, tất cả các gói được xử lý theo IPSec sẽ bị tăng kích thước do phải thêm vào các tiêu đề khác nhau, điều này làm cho thông lượng hiệu dụng của mạng giảm xuống. Vấn đề này có thể được khắc phục bằng cách nén dữ liệu trước khi mã hoá.

IPSec được thiết kế chỉ để hỗ trợ bảo mật cho lưu lượng IP, không hỗ trợ các dạng lưu lượng khác. Việc tính toán nhiều giải thuật phức tạp trong IPSec vẫn còn là một vấn đề khó đối với các trạm làm việc và máy PC năng lực yếu. Việc phân phối các phần cứng và phần mềm mã vẫn còn bị hạn chế đối với chính phủ một số quốc gia.

II. Đối tượng và phạm vi nghiên cứu

1. Đối tượng nghiên cứu

Thiết kế và mô phỏng kiến trúc giao thức IPSec trên KIT FPGA với:

- Mạng cục bộ (LAN): mạng cục bộ dạng chủ khách hay mạng đồng cấp.
- Mạng diện rộng (WAN): mạng WAN giữa các bộ định tuyến (Router to Router) hay giữa các cổng mạng (Gateway to Gateway).
- Truy cập hệ thống mạng đa điểm từ xa: Truy cập Internet từ hệ thống máy tính cá nhân.

Mô phỏng mô hình thiết kế trên KIT FPGA và khảo sát tốc độ truyền dữ liệu và dung lượng truyền dữ liệu.

2. Phạm vi nghiên cứu

- Nghiên cứu bài báo khoa học đăng trên tạp chí, bài báo tham dự hội thảo, luận văn tốt nghiệp,...
- Nghiên cứu cơ sở, kiến trúc, chính sách của hệ thống IPsec trên nền tảng của các hệ điều hành.
- Thiết kế bằng ngôn ngữ phân cứng (VHDL) và kiểm tra thiết kế đảm bảo yêu cầu về chức năng cũng như hiệu suất của thiết kế (assembly language, VCS,...).
- Thực hiện nhúng thiết kế lên KIT FPGA và kiểm tra chức năng, hiệu năng của thiết kế đảm bảo sản xuất thành sản phẩm thực thể.
- Thời gian nghiên cứu:
 - o Nghiên cứu các bài báo khoa học, tài liệu khoa học: Tháng 2 năm 2019 tới tháng 6 năm 2019.
 - o Thiết kế phân cứng và viết báo cáo khoa học: tháng 6 năm 2019 tới tháng 12 năm 2019.
 - o Hiện thực hoá thiết kế kiến trúc IPsec trên KIT FPGA và báo cáo khoa học: tháng 12 năm 2019 tới tháng 4 năm 2020.
 - o Báo cáo khoa học và nộp quyền luận văn tốt nghiệp: tháng 4 năm 2020 tới tháng 6 năm 2020.

CHƯƠNG 2

TỔNG QUAN NGHIÊN CỨU

I. Những vấn đề tồn đọng trong nghiên cứu IPsec hiện thực hoá trên phần cứng FPGA

1. Nâng cao tốc độ

Mỗi giây trôi qua, số lượng dữ liệu được thực hiện trên mạng máy tính lên đến hàng gigabit. Để có thể xử lý một lưu lượng lớn trên lớp IPsec, cần thực hiện các phép biến đổi mật mã ở tốc độ rất cao. Tốc độ này hiện chỉ có sẵn cho các siêu máy tính đắt tiền. Các dự án sẽ tập trung vào việc điều tra khả năng tăng tốc chuyển đổi IPsec bằng phần cứng FPGA, và các phần cứng khác.

- Thiệt hiện thiết kế phần cứng khối Mật mã Rijndael (AES).
- Thực hiện thiết kế phần cứng khối Thuật toán Diffie-Hellman cho Key exchange.
- Thực hiện thiết kế phần cứng cho Hàm Hash functions HMAC-MD5 và HMAC-SHA.

Kết quả khi thực hiện AES trên FPGA:

- Area 986 CLB Slices + 16 Block SelectRAMs
- Max Clock frequency 50.4MHz
- Throughput 615.2Mbps

AES trong kiến trúc cơ bản thực hiện với tốc độ trên 0,5 Gbit / s. Điều này cho thấy rằng thông lượng của 1 Gbit / s có thể đạt được bằng cách sử dụng onestage pipeline và sẽ không làm tăng đáng kể diện tích của mạch.

Tốc độ hàm băm ảnh hưởng lớn đến tốc độ của bộ xử lý IPsec đã nghiên cứu rằng việc đẩy nhanh tốc độ xử lý của hàm băm (hash function) bằng cách sử dụng kiến trúc chuyển đổi thanh ngang (crossbar switch architecture) cho truyền dữ liệu đa lỗi được sử dụng. Với bốn lỗi AH, ESP, AES, HMAC-SHA-1 riêng biệt kết nối với một switch 8x8 ngang trong bộ vi xử lý IPsec, đã đạt được một thông lượng 1.5Gbps ở 200MHz và kiểm tra phần cứng được thực hiện trên FPGA. Bằng cách mô phỏng, hoạt động giao thức IPsec có thể đạt tốc độ trên đường dây là 10Gbps với 32 giao thức IPsec IP và lỗi IP mật mã được cấu hình trong bộ xử lý IPsec.

Yun Niu, Liji Wu, Li Wang, Xiangmin Zhang, Jun Xu [2] đã thực hiện phương pháp trên và cho kết quả như sau:

| Mode @Protocol | Frequency (MHz) | Throughput (Gbps) |
|----------------|-----------------|-------------------|
| Transport @AH | 200 | 1.56 |
| Tunnel @ESP | 200 | 1.51 |

Bảng 1 - Kết quả thực hiện bộ xử lý IPsec với 8x8 thanh ngang

Khi thực hiện kết nối 4 IP core AH, ESP, AES, HMAC-SHA-1 với 8x8 thanh ngang trên công nghệ 65nm CMOS kết quả cho thấy data throughput của lên đến 1.5Gbps với tần số 200MHz.

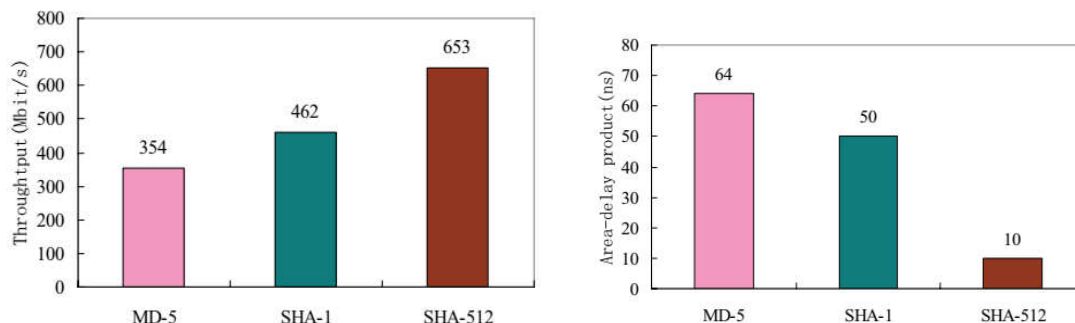
Một IPsec bao gồm 8 lõi giao thức AH / ESP và 24 thuật toán AES / HMAC-SHA-1 được kết nối với chuyển đổi thanh ngang được thực hiện Register Transfer Level (RTL) và được mô phỏng và tổng hợp dựa trên Công nghệ CMOS 65nm. Kết quả mô phỏng cho thấy rằng thông lượng dữ liệu của máy gia tốc IPsec đạt được lên đến 11Gbps với tần số xung nhịp 300MHz. Các kết quả mô phỏng và tổng hợp được thể hiện trong Bảng 2

| Mode @Protocol | Frequency (MHz) | Throughput (Gbps) | Gate Counts | Power Estimation (mW) |
|----------------|-----------------|-------------------|-------------|-----------------------|
| Transport @AH | 300 | 11.89 | 1118,166 | 258.56 |
| Tunnel @ESP | 300 | 11.28 | | |

Bảng 2 - Kết quả mô phỏng của IPSEC với 8x32 thanh ngang

Các hàm băm mới, ví dụ như SHA-512 được sử dụng thay cho hàm băm cũ vì tốc độ truyền dữ liệu nhanh hơn và thời gian thực hiện hàm nhỏ hơn.

LI Hong-qiang and MIAO Chang-yun [3] trong bài báo của mình đã thực hiện cho ra kết quả:



Hình 1 - Tốc độ dữ liệu, thời gian thực hiện hàm băm

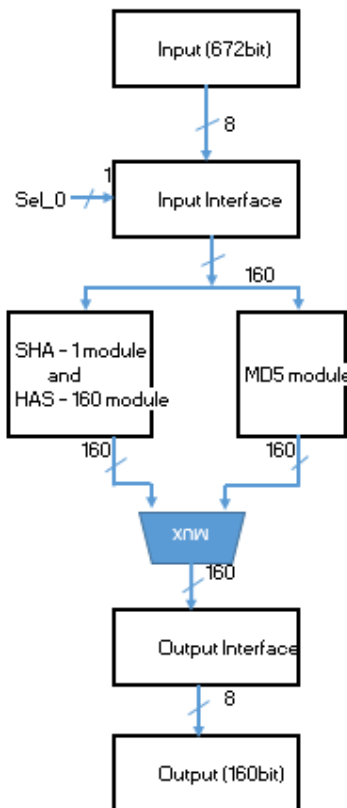
Tốc độ dữ liệu của SHA-512 cao hơn lên đến 653Mb/s so với các hàm cũ khác như SHA-1, MD-5.

Thời gian cũng nhỏ hơn rõ rệt khi thực hiện trên phần cứng so với các hàm khác chỉ 10ns.

2. Tối ưu hoá tài nguyên

Bởi vì các thuật toán băm khác nhau và các loại thuật toán đều cần thiết áp dụng cho IPSEC chip. Do đó, nhiều bài báo viết về triển khai SHA-I, HAS-I60 và MD5 ghép các phần giống nhau lại với nhau từ đó thực hiện trên một chip. Việc này làm giảm số lượng cổng logic trong thiết kế.

Các giáo sư *Yong kyu Kang, Dae Won Kim, Taek Won Kwon and Jun Rim Choi* [4] đã nghiên cứu thiết kế mô phỏng mô-đun SHA-I được kết hợp với mô-đun HAS-I60 cho ra kết quả phía dưới.



Hình 2 - Mô hình cấu trúc chia sẻ tài nguyên

| | Non – resource sharing processor | Resource sharing processor |
|----------------------|----------------------------------|----------------------------|
| Logic Element counts | 14604 | 10573 |

Bảng 3 - Tài nguyên sử dụng trước và sau khi thực hiện ghép

Kết quả là giảm 27% số cổng logic, từ 14604 xuống 10573 cổng.

Với mục tiêu phát triển các hệ thống bảo mật tốc độ cao, các lõi sử dụng kiến trúc pipeline để sử dụng thời gian rảnh của phép chuyển đổi AES để tích hợp nhiều lõi và sử dụng DMA để tăng tốc độ xử lý). Nhưng khi dùng cấu trúc pipeline sẽ làm tăng tài nguyên sử dụng và năng lượng tiêu thụ nên công nghệ DPR được áp dụng.

Trong bài báo [7] thực hiện và so sánh tài nguyên tiêu thụ khi sử dụng và không sử dụng DPR.

| Resource Logic | Crypto with full Modules (non-reconfig) | Crypto at module 1 st AES – 128/SHA-2 | Crypto at module 2 nd AES – 192/SHA-3 | Crypto at module 3 rd AES – 256/SHA-5 |
|---------------------------|---|--|--|--|
| Number of Slice Registers | 8983 (2.98 % of 301440) | 1885 (0.625%) | 3357 (1.11%) | 3741 (1.24%) |
| Number of Slice LUTs | 21306 (14.13 % of 150770) | 4416 (2.93%) | 8109 (5.38%) | 8781 (5.83%) |

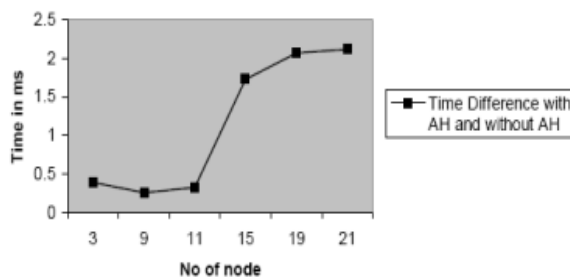
Bảng 4 - Thống kê tài nguyên sử dụng cho việc sử dụng và không sử dụng DPR Mode (DPR Mode and Full-Cryptography IP Core)

Với kết quả thực nghiệm khi áp dụng đã chỉ ra rằng resource Slice 50% và LUT Register 30% so với Full-Cryptography IP Core.

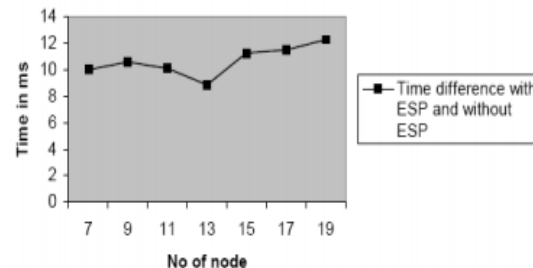
3. Vấn đề về bảo mật

Việc triển khai IPsec được đề xuất nhằm đảm bảo an ninh truyền thông dữ liệu. Gửi và nhận các gói dữ liệu với IPsec cần nhiều thời gian hơn so với việc gửi các gói dữ liệu mà không có IPsec. Giữa AH và ESP khi thực hiện truyền các gói dữ liệu thì ESP tiêu thụ nhiều thời gian hơn do xử lý mã hóa.

Các giáo sư Mr. Hitesh dhall, Mr. Hitesh dhall, Mr. Hitesh dhall, Mr. Hitesh dhall [5] đã nghiên cứu về thời gian sai lệch của khi dùng và không dùng AH và ESP protocol



Hình 3 - Đồ thị sai khác thời gian giữa dùng và không dùng AH protocol



Hình 4 - Đồ thị sai khác thời gian giữ dùng không dùng ESP protocol

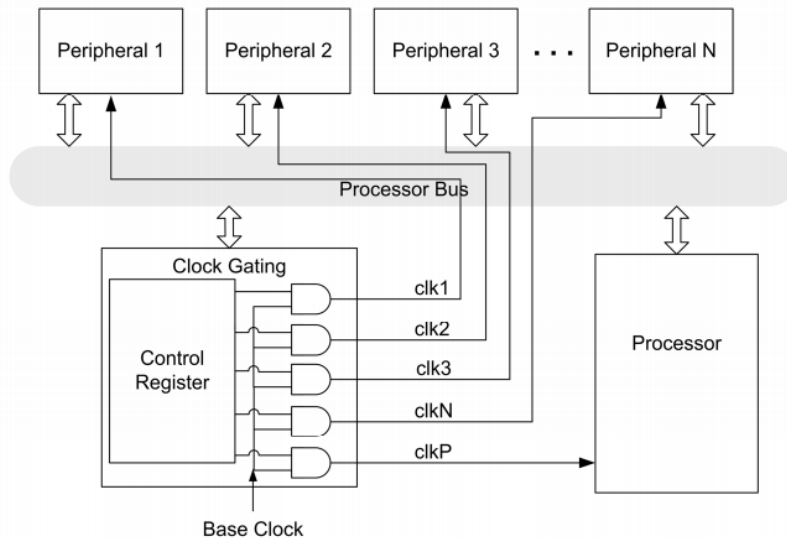
Nếu một ứng dụng chỉ cần xác thực, thì nghiên cứu này đề xuất chỉ sử dụng các gói dữ liệu do AH triển khai với chi phí thời gian bị hạn chế và chỉ cần authentication (xác thực). Nghiên cứu khuyến khích triển khai IPsec với ESP cho tất cả các dịch vụ bảo mật với thời gian trung bình và cần cả confidentiality (bảo mật) và authentication (xác thực)

4. Tối ưu hoá năng lượng tiêu thụ

Để giảm công suất, thiết kế một đơn vị quản lý năng lượng (power management unit - PMU). Đối với các lõi không được sử dụng, năng lượng của chúng sẽ bị tắt bởi tín

hiệu điều khiển từ thiết bị quản lý nguồn. Ngoài ra, phương thức quản lý đồng hồ như đồng hồ hẹn giờ (clock gating) cũng được sử dụng trong thiết kế để giảm công suất.

Clock gating là một phương pháp tắt đồng hồ cho một khối cụ thể khi nó không cần thiết và được sử dụng bởi hầu hết các thiết kế SoC ngày hôm nay như là một kỹ thuật hiệu quả để tiết kiệm năng lượng động.



Hình 5 - Mô hình hoạt động đơn giản của Clock gating sử dụng cổng AND

Tác giả Nguyễn Trọng Tuấn và cộng sự [7] đã đề xuất sử dụng công nghệ Dynamic partial reconfiguration technology (DPR) (DPR) để giảm nguồn tài nguyên FPGA và tiêu thụ điện năng trên chip và kiến trúc đa lõi (Multiple-Core).

Hầu hết các công việc của IPsec tập trung vào bảo mật, và các hàm băm khi thực hiện sẽ tiêu tốn hiệu suất nhiều nhất của hệ thống nên người ta chuyển hàm băm thực hiện hết trên phần cứng thay cho phần mềm.

Các thuật toán yêu cầu tính toán rất nặng trong IPsec nên hiệu năng kết nối mạng sẽ bị giới hạn.

II. Nghiên cứu của Esam Khan từ đại học Victoria, Victoria, BC, Canada

Esam Khan, M. Watheq El-Kharashi, Fayez Gebali, and Mostafa Abd-El-Barr. Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada. Emails: {ekhan, watheq, fayez, mabdelba}@ece.uvic.ca}

Vào năm 2005 nhóm nghiên cứu từ đại học Victoria ở Canada đứng đầu là Esam Khan đã nghiên cứu và thiết kế một hệ thống SOC thực hiện một unified hash engine cho IPsec xác thực dữ liệu trên FPGA. Một thiết kế unified hash engine chứng minh rằng nó hữu ích vì cả ba thuật toán hàm băm gồm MD5, SHA-1, and RIPEMD-160 đều cùng được thực hiện trên 1 lõi IP. Hệ thống SoC có thể với unified hash engine có thể cấu hình được 1 trong 3 hàm băm nào được sử dụng.

Hàm băm được có ứng dụng quan trọng trong việc xác thực và xác định tính trọn vẹn của dữ liệu. Có những giải thuật hàm băm phổ biến như MD4, MD5, SHA-1 và RIPEMD-160. Các hàm này cũng có thể được kết hợp với một số phương pháp mã hóa để tăng cường bảo mật. Một trong những phương pháp đó là HMAC (The Keyed-Hash Message Authentication Code) kết quả của thuật toán sẽ trở thành HMAC-MD5, HMAC-RIPEMD-160 và HMAC-SHA-1. Ba thuật toán này là thuật toán xác thực chuẩn được sử dụng cho IPsec. Trong IPsec cả ESP và AH đều sử dụng các hàm nên thay vì sử dụng cả 3 module hàm băm thì ta chỉ gộp chúng thành một nhằm tiết kiệm được diện tích, diện năng tiêu thụ, thời gian xử lý là chi phí sản xuất.

Dựa trên những điểm tương đồng và khác biệt của ba băm chức năng, đề xuất một thuật toán thống nhất và sau đó triển khai thuật toán này trên một kiến trúc thống nhất. Thuật toán thống nhất bao gồm hai khối, một khối chính và một khối xử lý. Khối chính chịu trách nhiệm khởi tạo, tiền xử lý, và các bộ phận hoàn thành như mô tả trong bộ chung ở trên. Mặt khác, khối xử lý thực hiện xử lý một phần của thuật toán thống nhất.

Thuật toán được mô tả ngắn gọn ở lưu đồ Hình 6.

Kết quả thực hiện mô phỏng trên Xilinx Vertex II FPGA (XC2V3000) được như sau:

| | |
|--|------------|
| LUTs usage | 40.57% |
| Maximum frequency | 37.0MHz |
| Estimated power consumption | 637mW |
| Clock cycles for one 512bit block for MD5 | 130 |
| Clock cycles for one 512bit block for SHA-1 | 162 |
| Clock cycles for one 512bit block for RIPEMD-160 | 162 |
| Average MD5 throughput of a 512bit block | 145.72Mbps |
| Average SHA-1 throughput of a 512bit block | 116.94Mbps |
| Average RIPEMD-160 throughput of a 512bit block | 116.94Mbps |

So với những thiết kế cũ hơn đạt kết quả như sau:

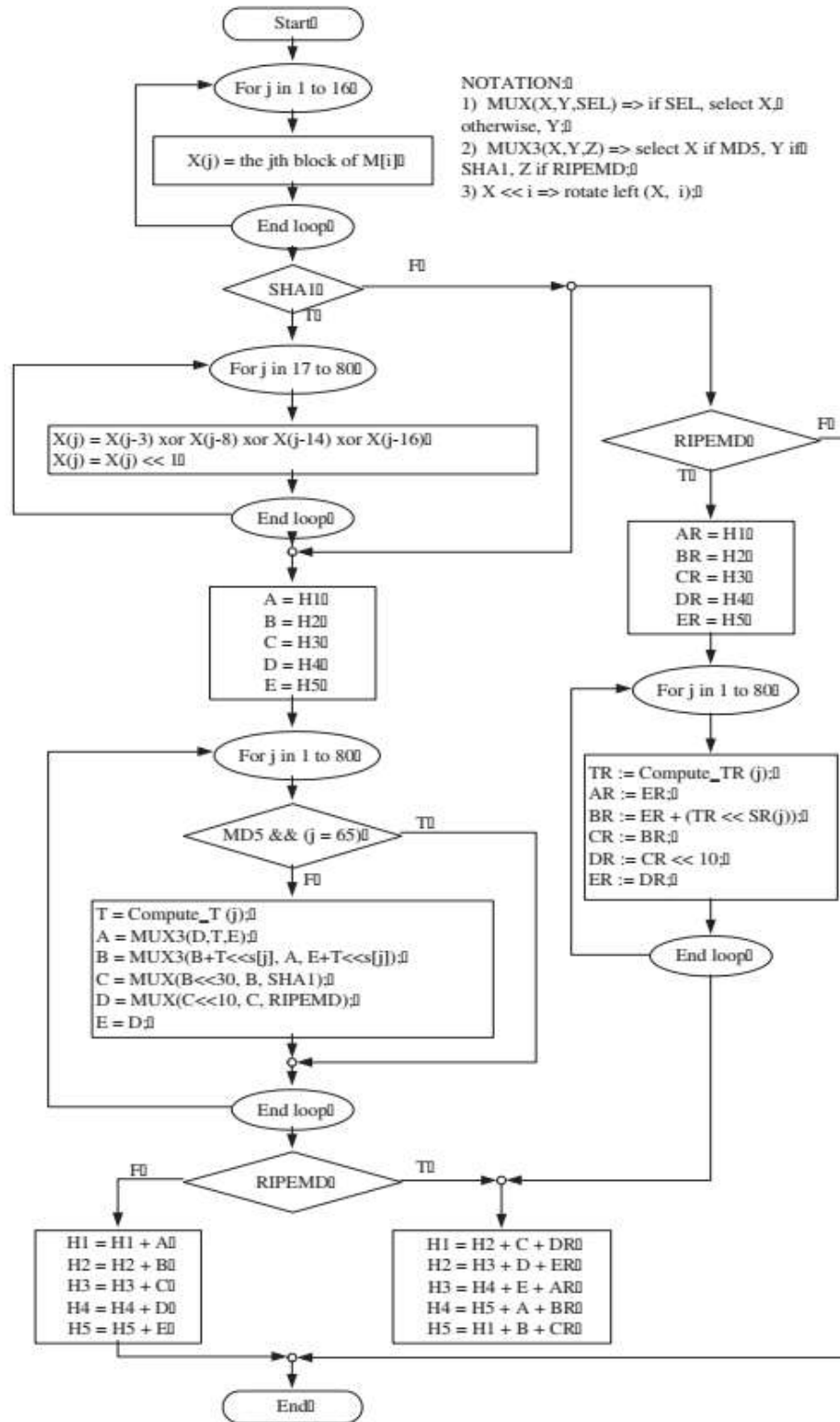
COMPARISON WITH PREVIOUS WORKS.

| Designs and proposals | Ng [12] | Wang [13] | Kang [14] | Dominikus [15] | Sklavos [16] | Proposed design |
|---|-------------|-------------|--------------|----------------|--------------|-----------------|
| Hash functions implemented* | M, R | M, S | M, S | M, S, R | S, R | M, S, R |
| FPGA vendor | Altera | Altera | Altera | Xilinx | Xilinx | Xilinx |
| Area cost** | 1964 LCs | 3040 LCs | 10573 LEs | 1004 CLBs | 2245 CLBs | 1454 CLBs |
| Maximum frequency (MHz) | 26.66 | 22.67 | 18.0 | 42.9 | 55.0 | 37.0 |
| Clock cycles for MD5*** | 66 | 65 | 65 | 206 | - | 130 |
| Average MD5 throughput (Mbps)*** | 206 | 178.6 | 142 | 107 | - | 145.72 |
| Clock cycles for SHA-1*** | - | 81 | 81 | 255 | 20+1 | 162 |
| Average SHA-1 throughput (Mbps)*** | - | 143.3 | 114 | 86 | 1339 | 116.94 |
| Clock cycles for RIPEMD-160*** | 162 | - | - | 337 | 16+1 | 162 |
| Average RIPEMD-160 throughput (Mbps)*** | 84 | - | - | 65 | 1656 | 116.94 |

* We only include results for the three hash functions MD5 (M), SHA-1 (S), and RIPEMD-160 (R)

** LC = Logic Cell, LE = Logic Element, CLB = Configurable Logic Block

*** Per one 512-bit block



Hình 6 - Lưu đồ giải thuật unified hash

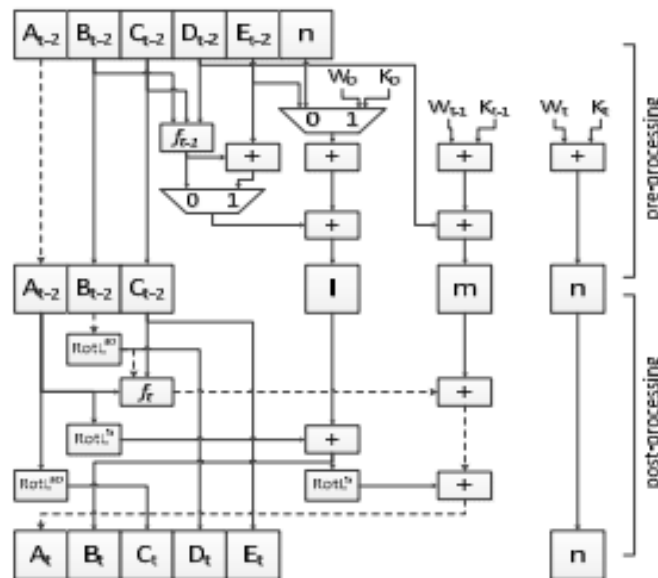
- [12] C.-W. Ng, T.-S. Ng, and K.-W. Yip, "A unified architecture of MD5 and RIPEMD-160 hash algorithms," in Proceedings of the 2004 International Symposium on Circuits and Systems, ISCAS '04, May 2004, pp. 23–26.
- [13] M.-Y. Wang, C.-P. Su, C.-T. Huang, and C.-W. Wu, "An HMAC processor with integrated SHA-1 and MD5 algorithms," in Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC 2004, Jan. 2004, pp. 456–458.
- [14] Y. kyu Kang, D. W. Kim, T. W. Kwon, and J. R. Choi, "An efficient implementation of hash function processor for IPsec," in Proceedings of the 2002 IEEE Asia-Pacific Conference on ASIC, Taipei, Taiwan, Aug. 2002, pp. 93–96.
- [15] S. Dominikus, "A hardware implementation of MD4-family hash algorithms," in Proceedings of the 9th International Conference on Electronic, Circuits and Systems, Sept. 2002, pp. 1143–1146.
- [16] N. Sklavos, G., Dimitroulakos, and O. Koufopavlou, "An ultra high speed architecture for VLSI implementation of hash functions," in Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2003, Dec. 2003, pp. 990–993.

Kết quả của thiết kế cho thấy có thể giảm được thời gian xử lý, diện tích, và điện năng tiêu thụ trên module trên.

III. Thiết kế làm thông lượng cao cho giải thuật hàm băm SHA-1 của Jae-woon Kim từ Hanyang University Hàn Quốc

Năm 2012, Jae-woon Kim và cộng sự tại đại học Hanyang đã đề xuất 1 thiết kế phần cứng SHA-1 đạt được high-throughput, hai kỹ thuật được sử dụng ở đây là loop unfolding và pre-processing và thực hiện trên kit Xilinx Virtex-6 FPGA.

Lỗi SHA-1 cơ bản được mô tả như sau áp dụng 2 unfolding và pre-processing

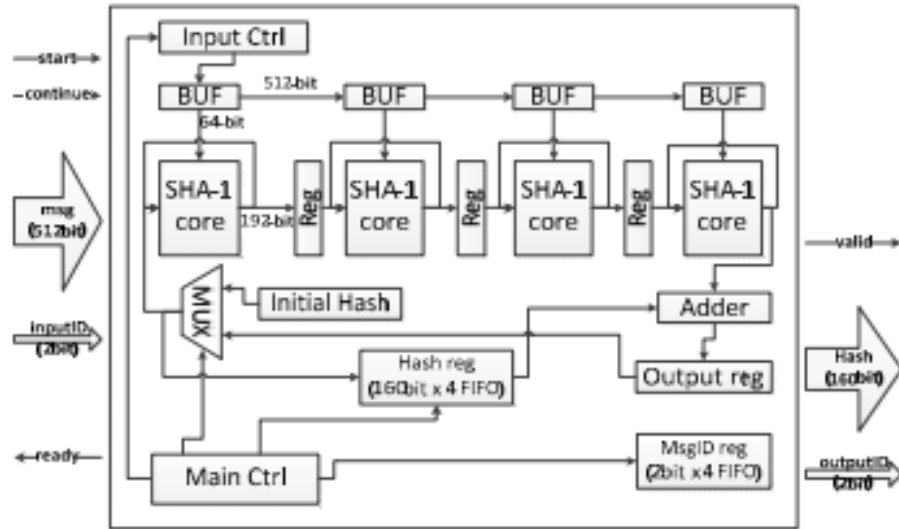


Hình 7 - Kiến trúc lõi SHA-1

Với các biến lưu tạm là:

$$\begin{aligned}
 a_t &= \text{RotL}^5\{\text{RotL}^5(a_{t-2}) + l_{t-2}\} + f_1(a_{t-2}, \text{RotL}^{30}(b_{t-2}), c_{t-2}) + m_{t-2} \\
 b_t &= \text{RotL}^5(a_{t-2}) + l_{t-2} \\
 c_t &= \text{RotL}^{30}(a_{t-2}) \\
 d_t &= \text{RotL}^{30}(b_{t-2}) \\
 e_t &= c_{t-2} \\
 l_t &= f_1(b_t, c_t, d_t) + e_t + W_t + K_t \\
 m_t &= d_t + W_{t+1} + K_{t+1} \\
 n_t &= W_{t+2} + K_{t+2}
 \end{aligned}$$

Lỗi SHA-1 có thể chạy 80 hoạt động trong 41 chu kỳ vì nó được áp dụng vòng lặp mở ra và xử lý trước, do đó, kiến trúc đường ống có thể có tới 40 giai đoạn. Trong nghiên cứu này, có 4 giai đoạn và mỗi giai đoạn hoạt động trong 10 chu kỳ, nhưng giai đoạn đầu tiên xử lý hoạt động bổ sung của 1 chu kỳ do pre-processing.



Kết quả đạt được như sau:

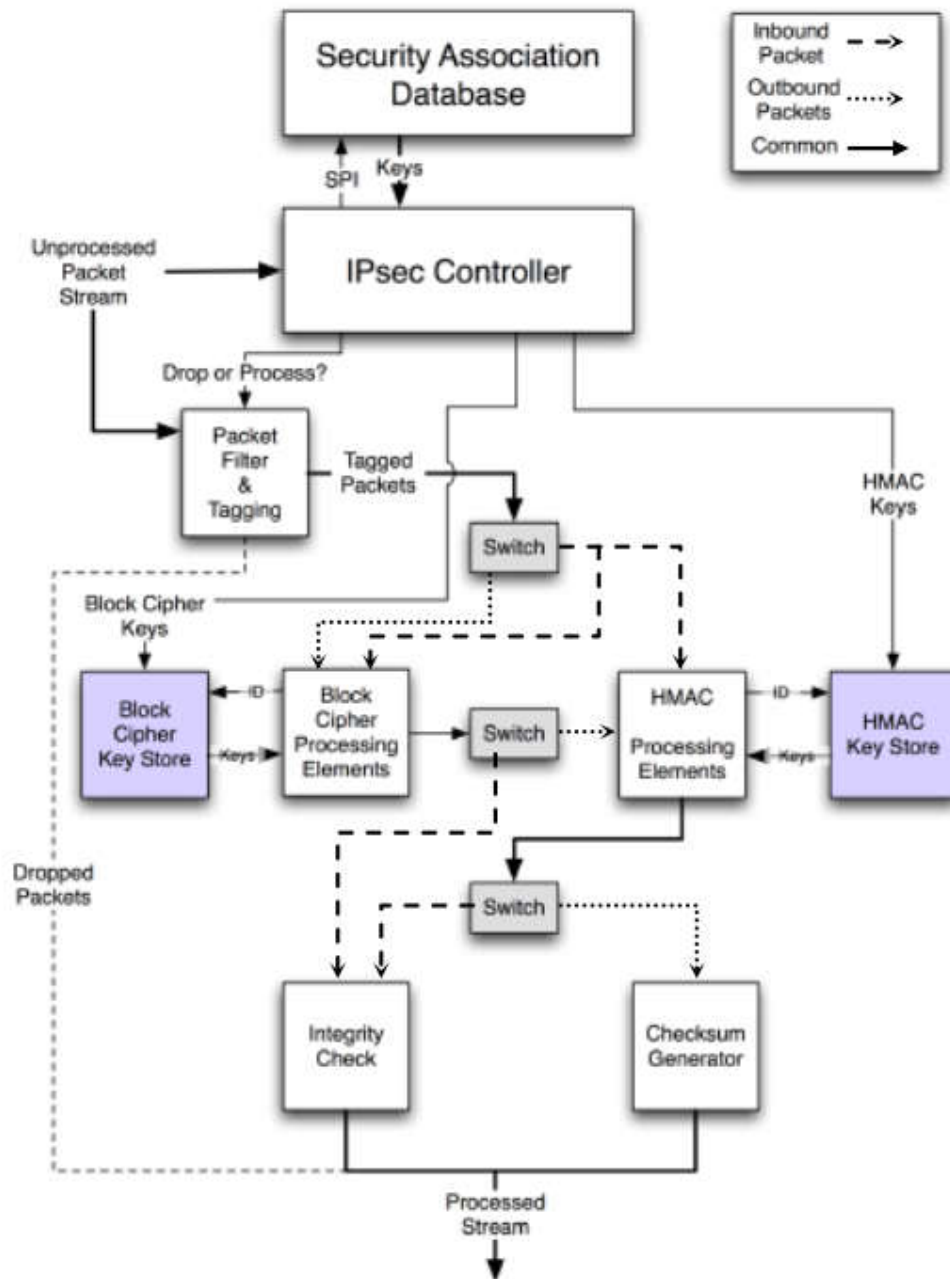
| Design | Clock (Mhz) | Latency | Slice register | Slice LUT | Throughput (Mbps) |
|----------------|-------------|---------|----------------|-----------|-------------------|
| original | 165.2 | 80 | 1,242 | 1,417 | 1,057 |
| unfolding | 142.6 | 40 | 1,270 | 1,647 | 1,825 |
| Pre-processing | 161.2 | 41 | 1,250 | 1,934 | 2,013 |
| Pipeline(4p) | 150.7 | 42 | 3,818 | 5,652 | 7,351 |

Kết quả của đề xuất kiến trúc SHA-1 yêu cầu 1.649 slices và đạt được thông lượng của 7,35 Gbps ở 150,45 MHz.

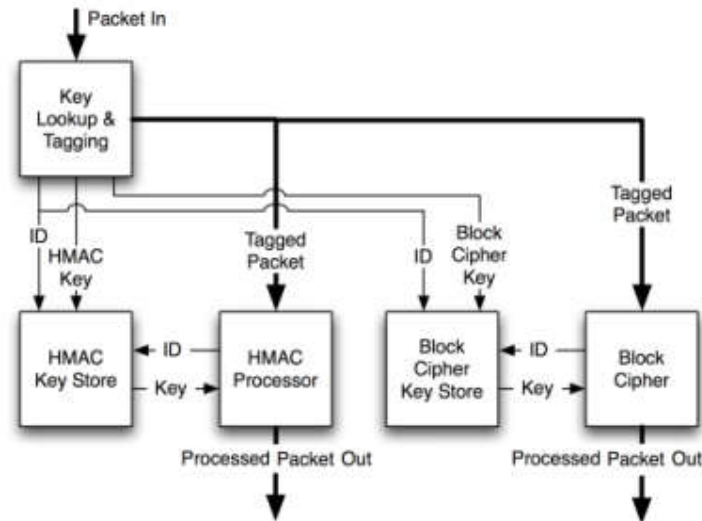
IV. Thiết kế một SoC đã lỗi IPsec thời gian thực cho lưu lượng Internet

Năm 2010, Patrick Moore từ đại học Queen's University Belfast, Mĩ đã nghiên cứu và đề xuất 1 cấu trúc Soc tốc độ cao IPsec cho thông lượng internet.

Thiết kế đề xuất cho HMAC xác thực song song khối, xử lý khóa phân tán và một thiết kế mã hóa khối pipelined cho phép mã hóa phản hồi mode. Điều này cải thiện khi thiết kế tiên tiến nhất cho IPsec, tạo ra một kiến trúc phù hợp cho việc cung cấp mới nổi bật ứng dụng truyền trực tuyến tốc độ cao an toàn qua Internet.



kiến trúc chung đã được đề xuất có thể được sử dụng cho cả đường dẫn dữ liệu trong và ngoài, thể hiện trong hình:



Khi một gói đi vào, nó đi vào trong key-lookup and tagging phần cứng và gói được gắn thẻ. Các phím được gửi đến HMAC Key Store, nơi chúng có thể được truy cập bởi bộ xử lý mật mã. Khi gói thoát khỏi hệ thống của chúng có thể được giải phóng, miễn là không có gói nào từ kết nối đó trong hệ thống tại thời điểm đó.

Về kiến trúc phần cứng xác thực: Để đẩy nhanh quá trình xác thực, một cơ chế mới đã được áp dụng để cho phép song song của các khối HMAC như hình 3.

Thuật toán được chỉ định để xác thực dữ liệu là HMAC-SHA-1. Thiết kế nhanh nhất được báo cáo cho đến nay đạt tới 199Mb/s, ví dụ như hệ thống 8Gb/s sẽ yêu cầu 40 khối HMAC. Điều này phản ánh các giới hạn của Kiến trúc HMAC-SHA-1, tuy nhiên đối với dòng multi-gigabit các hàm băm nhanh hơn.

Về Kiến trúc mã hóa khối:

Là một gói đi vào hệ thống, được hiển thị trong hình 4, nó được thông qua đến một bộ round-robin phân bổ các gói tuần tự đến một FIFO. Các FIFO này được phục vụ bởi một mật mã khối. Bước tiếp theo truy xuất gói tin lần lượt. Có nhiều FIFO như có các giai đoạn đường ống, cho phép hệ thống đồng bộ hóa đầu ra của mật mã khối với gói.

Kết nối:

Khi xử lý được hoàn thành trong HMAC hoặc thuật toán mã hóa cốt lõi, nó được nhập vào một FIFO mà nguồn cấp dữ liệu vào một cửa hàng gói. Khi một gói đến trong kho lưu trữ gói, nó được lưu trữ theo ID thẻ luồng của nó. ID này được chuyển cho người khác gói lưu trữ. Nếu một kết hợp được tìm thấy thì cả hai gói được xuất được kiểm tra trong kiểm tra tính toàn vẹn.

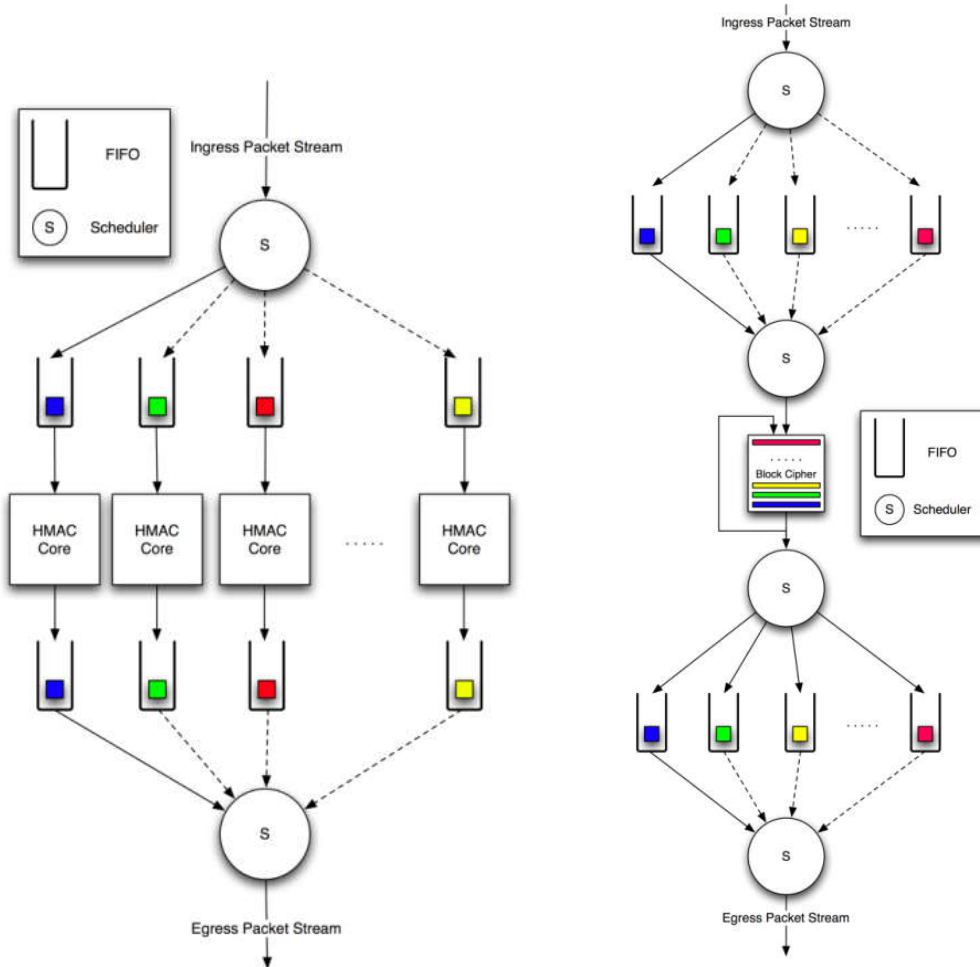
Các ví dụ áp dụng hệ thống Soc đa lõi:

- IPsec cho VoIP(Voice over IP)

Kết quả đạt được như sau:

TABLE I
PROCESSING BLOCKS FOR 500K CONCURRENT VOIP CONNECTIONS

| Protocol | AES | | Whirlpool | | UMAC | |
|----------|--------|------------|-----------|------------|--------|------------|
| | Blocks | Efficiency | Blocks | Efficiency | Blocks | Efficiency |
| G.711 | 3 | 87.1% | 14 | 97.8% | 1 | 82.85% |
| G.732.1 | 1 | 27.1% | 2 | 71.0% | 1 | 8.59% |



IPsec cho VoD

Bây giờ hãy xem xét đến độ nét cao H.264/MPEG-4 với độ phân giải 720p VoD kênh. Kích thước MTU là 1.500 byte được giả định, nhất quán với mạng Ethernet. Do RTP giao thức và thông tin tiêu đề IPsec, trọng tải tối đa kích thước có sẵn là 1,432 byte và số lượng gói trên mỗi thứ hai có thể được tính là 1,159, giả định mức trung bình tỷ lệ nén 50. Từ những giá trị này, quá trình xử lý băng thông trên mỗi kênh có thể được tính là 13,28Mb/s.

Kết luận:

Các ứng dụng truyền thông thời gian thực như stream, truyền hình, yêu cầu quản lý quyền kỹ thuật số và VoIP, yêu cầu kết nối đồng thời an toàn, đang thúc đẩy cần cho kiến trúc IPsec tốc độ cao và do đó trao đổi khóa tốc độ cao. Kiến trúc được đề xuất là lý tưởng để cho phép vô số thương mại mới nổi bật như các ứng dụng yêu cầu phân phối nội dung an toàn trên Internet. Hỗ trợ kiến trúc mạch đa lỗi có thể mở rộng nhiều kênh IPsec đồng thời đã được đề xuất cải thiện đáng kể so với các thiết kế.

CHƯƠNG 3

NỘI DUNG VÀ PHƯƠNG PHÁP HIỆN THỰC HOÁ THIẾT KẾ

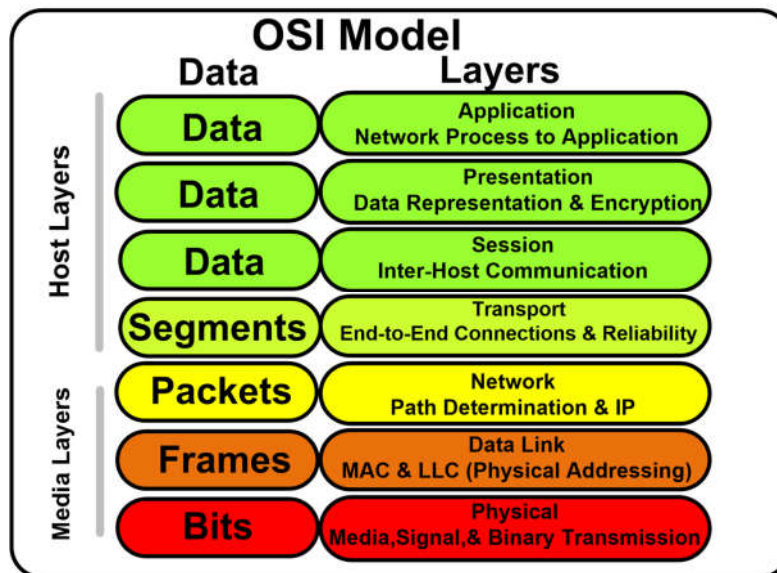
I. Nghiên cứu giao thức IPsec

1. Giới thiệu IPsec

IP Security hoặc còn gọi là IPsec được dựa trên nền tảng chuẩn được cung cấp một mã khoá cho phép bảo mật giữa hai thiết bị mạng ngang hàng.

IPsec là một tập hợp các chuẩn, các nguyên tắc đã được định nghĩa để kiểm tra, xác thực và mã hoá gói dữ liệu IP để cung cấp cho các kênh truyền dẫn mạng bảo mật.

IPsec sử dụng một số bộ giao thức chuẩn (AH, ESP, FIP-140-1 và một số tiêu chuẩn khác) được phát triển bởi Internet Engineering Task Force (IETF). Mục đích chính của việc phát triển IPsec là cung cấp một cơ cấu bảo mật ở tầng Network Layer của mô hình OSI.



Hình 8 - Mô hình OSI (Open Systems Interconnection Reference Model)

Mọi giao tiếp trong một mạng trên cơ sở IP đều dựa trên các giao thức IP. Do đó, khi một cơ chế bảo mật cao được tích hợp với giao thức IP, toàn bộ mạng được bảo mật bởi vì các giao tiếp đều đi qua tầng Network.

Các giao thức bảo mật khác trên Internet như SSL, TLS và SSH được thực hiện từ tầng Transport Layer trở lên. Điều này tạo nên tính chất mềm dẻo cho IPsec, giao thức này có thể hoạt động từ tầng Transport với TCP, UDP và hầu hết các giao thức. IPsec có một tính năng

nổi trội hơn SSL và giao thức bảo mật trên các tầng trên mô hình OSI thì đoạn mã ứng dụng đó sẽ thay đổi lớn.

Với tất cả các ứng dụng chạy ở tầng Application trong mô hình OSI đều độc lập trên tầng Network khi định tuyến dữ liệu từ nguồn tới đích. Bởi vì IPsec được tích hợp chặt chẽ với IP, nên những ứng dụng có thể dùng các dịch vụ kế thừa tính năng bảo mật mà không cần phải có sự thay đổi lớn lao nào. Cũng giống IP, IPsec trong suốt với người dùng cuối – nghĩa là người dùng cuối không cần quan tâm đến cơ chế bảo mật mở rộng liên tục đằng sau một chuỗi các hoạt động.

2. Mục đích của IPsec (IP Security Protocol)

Các **Header** (tiêu đề) của IP, **Transmission Control Protocol** (TCP – Giao thức Kiểm soát truyền dẫn), và **User Datagram Protocol** (UDP – Giao thức gói dữ liệu người dùng) đều chứa một số Kiểm soát (**Check sum**) được sử dụng để kiểm soát tính toàn vẹn (**Integrity**) dữ liệu của một gói IP. Nếu dữ liệu bị hỏng, Số Kiểm soát sẽ thông báo cho người nhận biết. Tuy nhiên, do thuật toán Số Kiểm soát này được phổ biến rộng rãi nên kẻ cả người dùng không có chức năng cũng có thể truy cập vào gói tin một cách dễ dàng thay đổi nội dung của chúng và tính lại Số Kiểm soát, sau đó lại chuyển tiếp gói tin này đến tay người nhận mà không một ai, kể cả người gửi lẫn người nhận biết đến sự can thiệp này. Do các hạn chế về chức năng của số kiểm soát như vậy, tại nơi nhận, người dùng không hề biết và cũng không thể phát hiện ra việc gói tin đã bị thay đổi.

Trong quá khứ, các ứng dụng cần bảo mật sẽ tự cung cấp cơ chế bảo mật cho riêng chúng dẫn tới việc có quá nhiều các chuẩn bảo mật khác nhau và không tương thích.

IPsec là một bộ các Giao thức và Thuật toán Mã hóa cung cấp khả năng bảo mật tại lớp Internet (Internet Layer) mà không cần phải quan tâm đến các ứng dụng gửi hay nhận dữ liệu.

Sử dụng IPsec, chỉ cần một chuẩn bảo mật được áp dụng và việc thay đổi ứng dụng không cần thiết.

IPsec có 2 mục đích chính:

- Bảo vệ nội dung của các gói IP.
- Cung cấp việc bảo vệ chống lại các cuộc tấn công mạng thông qua lọc gói tin và việc bắt buộc sử dụng các kết nối tin cậy.

Cả hai mục tiêu trên đều có thể đạt được thông qua việc sử dụng các dịch vụ phòng chống dựa trên cơ chế mã hóa, các giao thức bảo mật và việc quản lý các khóa động. Với các nền tảng như vậy, IPsec cung cấp cả hai tính năng Mạnh và Uyển chuyển trong việc bảo vệ các cuộc liên lạc giữa các máy tính trong mạng riêng, Miền, Site (bao gồm cả các Site truy cập từ xa), các mạng Intranet, các máy khách truy cập qua đường điện thoại. Thậm chí nó còn được sử dụng để khóa việc nhận hay gửi của một loại lưu thông chuyên biệt nào đó.

Chống lại các cuộc tấn công bảo mật là bảo vệ các gói tin làm cho chúng trở thành quá khó, nếu không nói là không thể, đối với các kẻ xâm nhập để có thể dịch được các dữ liệu mà họ thu giữ được. IPsec có một số các tính năng mà có thể làm giảm đáng kể hay ngăn ngừa được các loại tấn công sau:

- **Do thám gói dữ liệu (Packet Sniffing):** Packet Sniffer là một ứng dụng thiết bị có thể theo dõi và đọc các gói dữ liệu. Nếu gói dữ liệu không được mã hóa, các Packet Sniffer có thể trình bày đầy đủ các nội dung bên trong các gói dữ liệu.
- **Thay đổi dữ liệu:** Kẻ tấn công có thể thay đổi các thông điệp đang được vận chuyển và gửi đi các dữ liệu giả mạo, nó có thể ngăn cản người nhận nhận được các dữ liệu chính xác, hay có thể cho phép kẻ tấn công lấy được thêm các thông tin bảo mật. IPsec sử dụng các khóa mã hóa, chỉ được chia sẻ giữa người gửi và người nhận, để tạo ra các Số Kiểm soát được mã hóa cho mỗi gói IP. Mọi thay đổi đối với gói dữ liệu đều dẫn đến việc thay đổi Số Kiểm soát và sẽ chỉ ra cho người nhận biết rằng gói dữ liệu đã bị thay đổi trên đường truyền.
- **Nhận dạng giả mạo:** Kẻ tấn công có thể làm giả các mã nhận dạng (**Identity Spoofing**) bằng cách sử dụng một chương trình đặc biệt để xây dựng các gói IP mà xuất hiện như các gói dữ liệu gốc từ các địa chỉ hợp lệ bên trong mạng được tin cậy. IPsec cho phép trao đổi và xác nhận lại các mã nhận dạng mà không phơi chúng ra cho các kẻ tấn công dịch. Sự xác nhận lẫn nhau (xác thực) được sử dụng để thiết lập tin cậy giữa các hệ thống cũng tham gia liên lạc với các hệ thống khác. Sau khi các mã nhận dạng được thiết lập, IPsec sử dụng các khóa mã hóa, được chia sẻ chỉ giữa người gửi và người nhận, để tạo các số kiểm soát được mã hóa cho mỗi gói IP. Các số kiểm soát được mã hóa đảm bảo rằng chỉ các máy tính đã biết rõ về các khóa là có thể gửi được từng gói dữ liệu.
- **Tấn công ngang đường (man-in-the-middle attack):** trong dạng tấn công này, một người nào đó, đứng giữa hai máy tính đang liên lạc với nhau, sẽ tiến hành theo dõi, thu nhập và điều khiển các dữ liệu một cách trong suốt. IPsec kết hợp việc xác thực lẫn nhau và các được mã hóa để chống lại dạng tấn công này.
- **Tấn công từ chối dịch vụ (DoS):** Ngăn cản việc vận hành bình thường của các tài nguyên mạng và máy tính. Làm lụt các tài khóa E-mail bằng các thông điệp không mong muốn là một ví dụ của dạng tấn công này. IPsec sử dụng phương pháp lọc các gói IP (IP packet Filtering) làm cơ sở cho việc xác định mối liên lạc nào là được phép, bảo mật hay phải khóa lại. Việc xác định mối liên lạc nào là được phép, bảo mật hay phải khóa lại. Việc xác định trên dựa vào dãy địa chỉ IP, Giao thức IP hay thậm chí một số cổng TCP hay UDP xác định nào đó.

3. Những tính năng của IPsec (IP Security Protocol)

IPsec có rất nhiều tính năng bảo mật được thiết kế để thỏa mãn mục tiêu bảo vệ các gói IP và chống lại các cuộc tấn công nhờ vào các bộ lọc và cơ chế kết nối tin cậy. Một vài trong các tính năng bảo mật của IPsec được liệt kê sau:

Sự kết hợp bảo mật tự động: IPsec sử dụng **Internet Security Association** (Kết hợp bảo mật Internet) và **Key Management Protocol** (ISAKMP – Giao thức Quản lý Khóa) để thỏa thuận một cách tích cực về một tập của các yêu cầu bảo mật cho cả hai phía giữa các máy tính với nhau. Các máy tính không đòi hỏi phải có các chính sách giống hệt nhau, chúng chỉ cần các chính sách đã được cấu hình tùy chọn đã được thỏa thuận để để thiết lập một tập chung các yêu cầu với bảo mật với máy tính kia.

Lọc gói IP: Quá trình lọc này cho phép hay cấm các liên lạc cần thiết bằng cách chỉ định các khoảng địa chỉ IP, các giao thức, hay thậm chí cả những cổng của giao thức.

Bảo mật lớp mạng: IPsec nằm tại lớp mạng, cung cấp cơ chế bảo mật một cách tự động, trong suốt cho các ứng dụng.

Xác thực ngang hàng: IPsec xác nhận lại mã nhận dạng của máy tính đối tác trước khi có bất cứ một gói dữ liệu nào được chuyển. Việc xác thực đối tác IPsec trong Windows Server 2003 được dựa trên các khóa đã chia sẻ, các khóa công khai (ví dụ như các Giấy chứng nhận X509) hoặc **Kerberos** và **Active Directory** để được xác thực bằng **Kerberos**.

Xác thực dữ liệu gốc: Việc xác thực nguồn dữ liệu gốc ngăn cản người dùng không đúng không can thiệp vào gói tin và khai báo họ là người gửi dữ liệu. Mỗi gói tin dữ liệu được bảo vệ bằng IPsec bao gồm một số Số Kiểm Soát bằng mật mã trong định dạng của một giá trị băm có khóa. Số Kiểm Soát bằng mật mã còn được biết đến dưới cái tên **Integrity Check Value** (ICV - Giá trị kiểm soát tính nguyên vẹn) hay **Hash-Based Message Authentication** (HMAC – mã xác thực thông điệp được băm nhỏ).

Tính nguyên vẹn của dữ liệu: Với việc sử dụng số kiểm soát mật mã, IPsec bảo vệ dữ liệu đang vận chuyển không bị sửa đổi bởi các người dùng không được xác thực, hay không phát hiện được trong quá trình vận chuyển, đảm bảo chắc chắn rằng các dữ liệu các dữ liệu mà người nhận có được là chính xác các thông tin mà người gửi đã gửi cho mình. Các người sử dụng có ác tâm muốn thay đổi muốn thay đổi nội dung của gói tin phải cập nhật lại một cách chính xác Số Kiểm Soát bằng mật mã, một điều gần như là không thể thực hiện được nếu không biết được các khóa chia sẻ.

Tính riêng tư của dữ liệu (Data integrity): Các gói dữ liệu khi được gửi là mã hóa bằng các kỹ thuật mã hóa khóa bí mật qui ước. Điều này làm cho dữ liệu trở nên riêng tư. Thậm chí ngay cả khi dữ liệu bị truy nhập và quan sát, kẻ truy nhập cũng chỉ nhìn thấy các dữ liệu đã được mã hóa. Nếu không biết các khóa bí mật đã sử dụng thì các dữ liệu gốc vẫn là ẩn. Do khóa bí mật chỉ được chia sẻ giữa người gửi và người nhận, tính riêng tư của dữ liệu đảm bảo rằng chỉ người nhận đã định trước của gói tin là có thể giải mã và trình bày được gói tin.

Tính không lặp: Bằng cách sử dụng số thứ tự trên mỗi gói tin đã được bảo vệ gửi giữa các đối tác có sử dụng IPsec, dữ liệu được trao đổi giữa các đối tác không thể bị lại để thiết lập các quan hệ bảo mật khác hay nhận được sự truy cập không xác thực đến các thông tin hay tài nguyên.

Quản lý khóa (Key management): Việc xác thực nguồn gốc dữ liệu, tính nguyên vẹn, tính riêng tư hoàn toàn phụ thuộc vào các thông tin được chia sẻ của khóa bí mật. Nếu khóa bí mật bị tổn thương, liên lạc sẽ không còn là bảo mật nữa. Để giữ các khóa không bị các người sử dụng có ác tâm phát hiện IPsec cung cấp một phương thức an toàn cho việc trao đổi thông tin khóa để nhận được khóa bảo mật chia sẻ và thay đổi khóa một cách định kỳ cho các liên lạc cần bảo mật.

Các giao thức IPsec cung cấp sự bảo mật dựa trên việc sử dụng kết hợp các giao thức, trong đó có giao thức AH và giao thức ESP. Các giao thức này được sử dụng độc lập hay cái trước cái sau còn phụ thuộc vào các yêu cầu của việc giữ tính riêng tư và xác thực.

Giao thức AH cung cấp tính xác thực, nguyên vẹn và không lặp cho toàn bộ gói tin (gồm cả phần tiêu đề của IP và các dữ liệu được chuyển trong toàn bộ gói tin). Nó không cung cấp tính riêng tư, có nghĩa là nó không mã hóa dữ liệu. Dữ liệu vẫn có thể đọc được, nhưng chúng được bảo vệ chống lại việc thay đổi. Giao thức AH sử dụng các thuật toán **Keyed hash** để đánh dấu gói dữ liệu nhằm đảm bảo tính toàn vẹn của nó.

Giao thức ESP cung cấp tính riêng tư (thêm vào cho tính xác thực, toàn vẹn và không lặp) cho dữ liệu (IP Payload). ESP trong trạng thái vận chuyển sẽ không đánh dấu lại toàn bộ gói tin. Chỉ các thân gói tin IP (IP Payload) – không phải là **IP Header** – là được bảo vệ. ESP có thể được sử dụng độc lập hay kết hợp với AH. Ví dụ, khi sử dụng kết hợp với AH, các gói **IP Payload** được gửi từ máy tính A đến máy tính B được mã hóa và đánh dấu để đảm bảo tính nguyên vẹn. Khi nhận được, phần dữ liệu được truyền sẽ được giải mã sau khi quá trình xác nhận tính xác nhận tính toàn vẹn được thực hiện thành công. Và người nhận có thể biết chắc chắn rằng ai đã gửi gói dữ liệu, dữ liệu không bị thay đổi và không ai khác có thể đọc được chúng.

4. Cấu trúc bảo mật

IPsec được triển khai sử dụng các giao thức cung cấp mật mã (cryptographic protocols) nhằm bảo mật gói tin (packet) trong quá trình truyền, phương thức xác thực và thiết lập các thông số mã hoá.

Xây dựng IPsec sử dụng khái niệm về bảo mật trên nền tảng IP. Một sự kết hợp bảo mật rất đơn giản khi kết hợp các thuật toán và các thông số (ví như các khóa – keys) là nền tảng trong việc mã hoá và xác thực trong một chiều. Tuy nhiên trong các giao tiếp hai chiều, các giao thức bảo mật sẽ làm việc với nhau và đáp ứng quá trình giao tiếp.

Thực tế lựa chọn các thuật toán mã hoá và xác thực lại phụ thuộc vào người quản trị IPsec bởi IPsec bao gồm một nhóm các giao thức bảo mật đáp ứng mã hoá và xác thực cho mỗi gói tin IP.

Trong các bước thực hiện phải quyết định cái gì cần bảo vệ và cung cấp cho một gói tin outgoing (đi ra ngoài), IPsec sử dụng các thông số Security Parameter Index (SPI), mỗi quá trình Index (đánh thứ tự và lưu trong dữ liệu – Index ví như một cuốn danh bạ điện thoại) bao gồm Security Association Database (SADB), theo suốt chiều dài của địa chỉ đích trong header của gói tin, cùng với sự nhận dạng duy nhất của một thoả hiệp bảo mật (tạm dịch từ - security association) cho mỗi gói tin. Một quá trình tương tự cũng được làm với gói tin đi vào (incoming packet), nơi IPsec thực hiện quá trình giải mã và kiểm tra các khoá từ SADB.

Cho các gói multicast, một thoả hiệp bảo mật sẽ cung cấp cho một group, và thực hiện cho toàn bộ các receiver trong group đó. Có thể có hơn một thoả hiệp bảo mật cho một group, bằng cách sử dụng các SPI khác nhau, tuy nhiên nó cũng cho phép thực hiện nhiều mức độ bảo mật cho một group. Mỗi người gửi có thể có nhiều thoả hiệp bảo mật, cho phép xác thực, trong khi người nhận chỉ biết được các keys được gửi đi trong dữ liệu.

5. Giao thức sử dụng trong IPsec

IPsec bảo mật kết nối mạng bằng việc sử dụng hai giao thức và cung cấp bảo mật cho các gói tin của hai phiên bản IPv4 và IPv6:

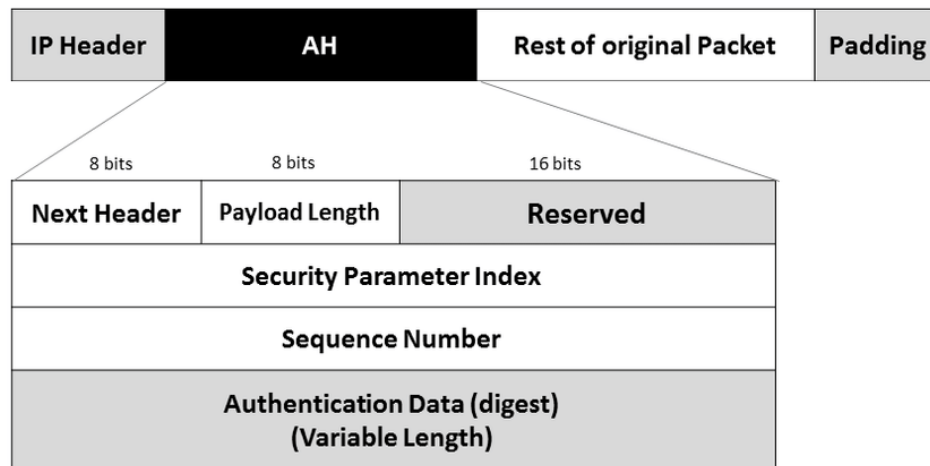
IP Authentication Header giúp đảm bảo tính toàn vẹn và cung cấp xác thực.

IP Encapsulating Security Payload cung cấp bảo mật và người dùng có thể lựa chọn những tính năng Authentication và Integrity đảm bảo tính toàn vẹn dữ liệu.

Các thuật toán mã hoá được sử dụng trong IPsec bao gồm HMAC-SHA1 cho tính toàn vẹn dữ liệu (Integrity protection), và thuật toán TripleDES-CBC và AES-CBC cho mã hoá và đảm bảo độ an toàn của gói tin. Toàn bộ thuật toán này được thể hiện trong RFC 4305.

5.1. Authentication Header (AH)

AH được sử dụng trong các kết nối không có tính đảm bảo dữ liệu. Hơn nữa đó là lựa chọn nhằm chống lại các cuộc tấn công replay attack bằng cách sử dụng công nghệ tấn công Sliding windows và Discarding older packets. AH bảo vệ quá trình truyền dữ liệu khi sử dụng IP. Trong IPv4, IP header có bao gồm TOS, Flags, Fragment Offset, TTL và Header Checksum. AH được thực hiện trực tiếp trong phần đầu tiên của gói tin IP.



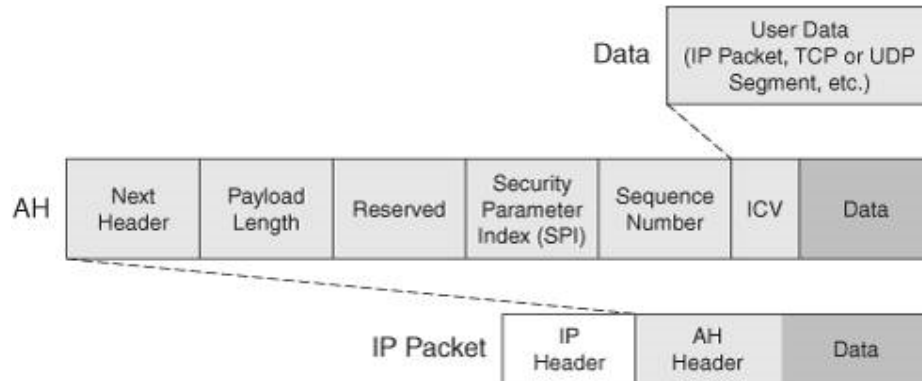
Hình 9 - Mô hình AH (Authentication Header)

Trong đó:

- Next header: Nhận dạng giao thức trong sử dụng truyền thông tin.
- Payload length: Độ lớn của gói tin AH.
- Reserved: Sử dụng trong tương lai (cho tới thời điểm này nó được biểu diễn bằng những giá trị 0).
- Security Parameters index (SPI): Nhận ra các thông số bảo mật, tích hợp địa chỉ IP, nhận dạng các thương lượng bảo mật được kết hợp với gói tin.
- Sequence number: Một số tự động tăng lên mỗi gói tin, sử dụng nhằm chống lại tấn công dạng replay attacks.
- Authentical data: Bao gồm thông số Integrity check value (ICV) cần thiết trong gói tin xác thực.

AH cung cấp tính xác thực, tính nguyên vẹn và khâu lắp cho toàn bộ gói tin bao gồm cả phần tiêu đề của IP (IP header) và các gói dữ liệu được truyền trong các gói tin.

AH không cung cấp tính riêng tư, không mã hoá dữ liệu. Như vậy, dữ liệu có thể được đọc nhưng chúng sẽ được bảo vệ để chống lại sự thay đổi. AH sẽ sử dụng thuật toán Key AH để đánh dấu gói dữ liệu nhằm đảm bảo tính toàn vẹn của gói dữ liệu.

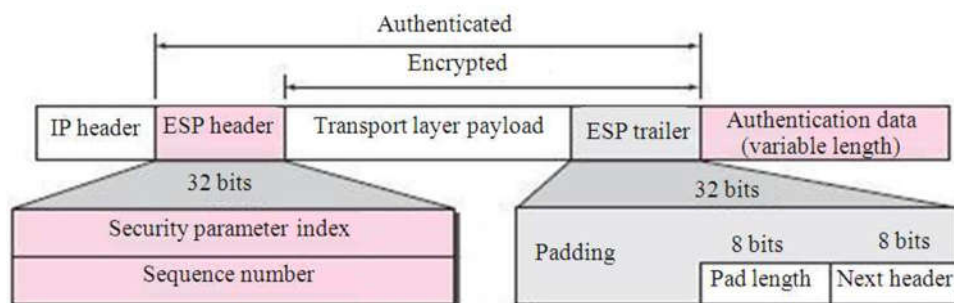


Hình 10 - AH Packetization Process

Do giao thức AH không có chức năng mã hoá dữ liệu nên AH ít được dùng trong IPsec vì nó không đảm bảo tính an ninh.

5.2. Encapsulating Security Payload (ESP)

Giao thức ESP cung cấp xác thực, độ toàn vẹn, đảm bảo tính bảo mật cho gói tin. ESP cũng hỗ trợ tính năng cấu hình sử dụng trong tình huống chỉ cần bảo mã hoá và chỉ cần cho Authentication. ESP sử dụng IP protocol number là 50 (ESP được đóng gói bởi giao thức IP và trường Protocol trong IP là 50).



Hình 11 - Mô hình ESP (Encapsulated Security Payload Protocol)

Trong đó:

- Security parameters index (SPI): Nhận ra các thông số được tích hợp với địa chỉ IP.
- Sequence number: Tự động tăng, có chức năng chống tấn công kiểu replay attacks.
- Payload data: Cho dữ liệu truyền đi.

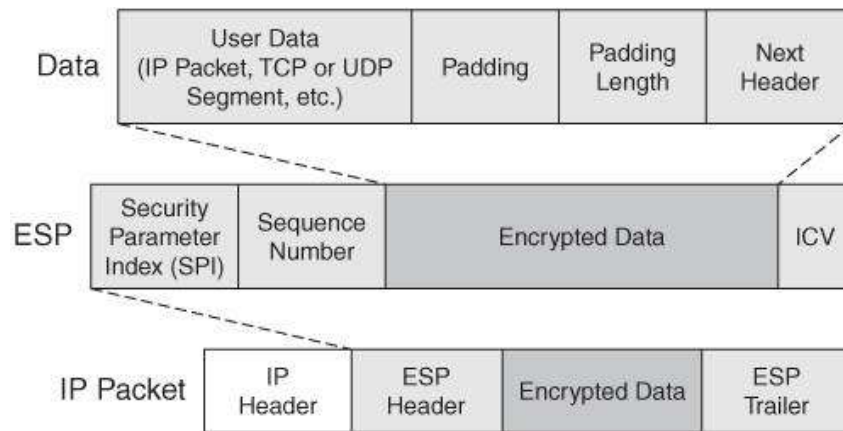
- Padding: Sử dụng vài khối mã hoá.
- Pad length: Độ lớn của padding.
- Next header: Nhận ra giao thức được sử dụng trong quá trình truyền thông tin.
- Authentication data: Bao gồm dữ liệu để xác thực cho gói tin.

Các thuật toán mã hoá được sử dụng bao gồm DES, 3DES, AES.

Các thuật toán xác thực bao gồm: MD5 hoặc SHA-1.

ESP còn cung cấp tính năng anti-relay để bảo vệ các gói tin bị ghi đè lên nó.

ESP trong trạng thái vận chuyển sẽ không đánh toàn bộ gói tin mà chỉ đóng gói phần thân IP.ESP có thể sử dụng độc lập hay kết hợp với AH.



Hình 12 - ESP Packetization Process

So sánh hai giao thức AH và ESP hỗ trợ các tính năng:

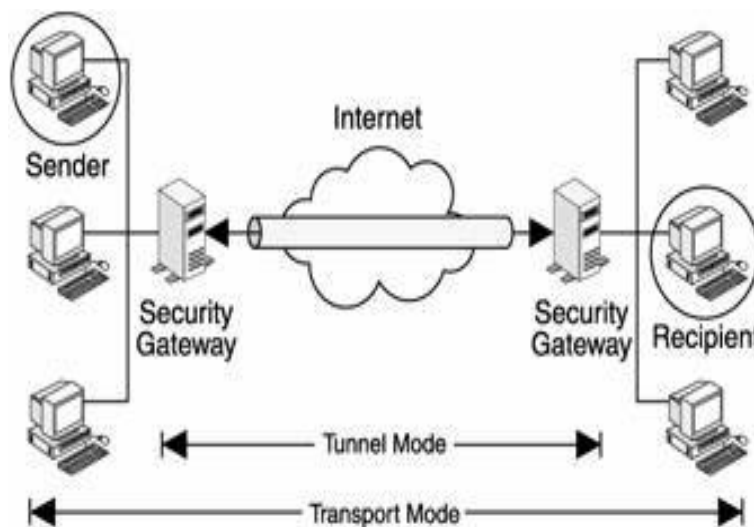
| Đặc tính bảo mật | AH | ESP |
|--|-------|-----|
| Số giao thức IP lớp Network Layer-3 IP protocol number | 51 | 50 |
| Cung cấp dịch vụ xác định tính nguyên vẹn dữ liệu Provides for data integrity | Có | Có |
| Cung cấp dịch vụ xác thực dữ liệu Provides for data authentication | Có | Có |
| Cung cấp dịch vụ mã hoá dữ liệu Provides for data encryption | Không | Có |
| Bảo vệ chống lại các cuộc tấn công replay attacks trên dữ liệu Protects against data replay attacks | Có | Có |

| | | |
|--|-------|-------|
| Hỗ trợ biên dịch địa chỉ mạng Works with NAT (Network Address Translation) | Không | Có |
| Hỗ trợ làm việc trên giao thức PAT Work with PAT (Port Address Translation) | Không | Không |
| Bảo vệ gói IP Protects the IP packet | Có | Không |
| Bảo vệ mỗi dữ liệu Protects only the data | Không | Có |

Bảng 5 - So sánh chức năng của giao thức AH và ESP

6. Các phương thức IPsec

IPsec được cấu hình để hoạt động bằng hai chế độ: chế độ Transport và chế độ Tunnel. Cả hai giao thức AH và ESP đều hỗ trợ để có thể làm việc với một trong hai chế độ này.

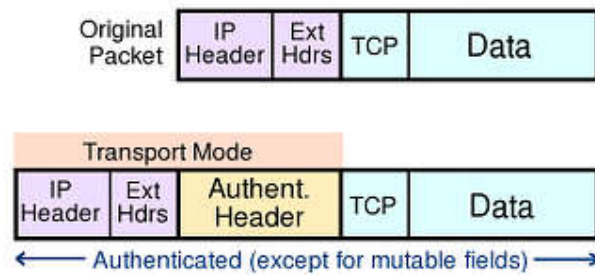


Hình 13 - Hai phương thức truyền dẫn

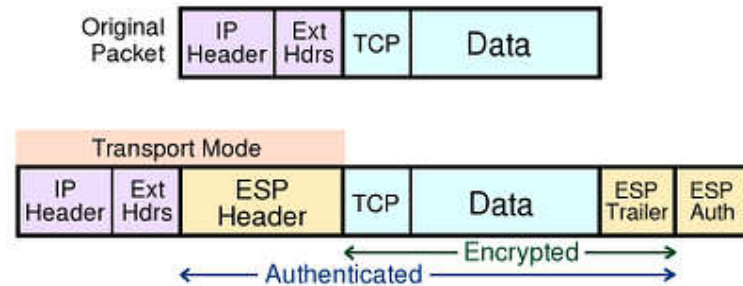
6.1. Phương thức vận chuyển (Transport Mode)

Phương thức vận chuyển (Transport Mode): bảo vệ giao thức tầng trên và các ứng dụng. Trong Transport mode, phần IPsec header được chèn vào giữa phần IP header và phần header của giao thức tầng trên.

Sử dụng bảo mật điểm tới điểm. Cả hai trạm cần hỗ trợ IPsec sử dụng cùng giao thức xác thực, bắt buộc phải sử dụng các bộ lọc IP tương thích và không đi qua một giao thức NAT nào. Liên lạc đi qua giao tiếp NAT sẽ chỉ đổi IP trên phần tiêu đề và làm mất hiệu lực của ICV (Giá trị kiểm soát tính nguyên vẹn).



Hình 14- Giao thức AH trong phương thức vận chuyển

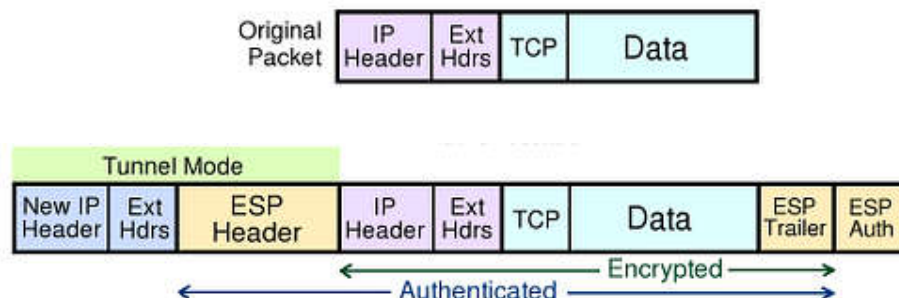


Hình 15 - Giao thức ESP trong phương thức vận chuyển

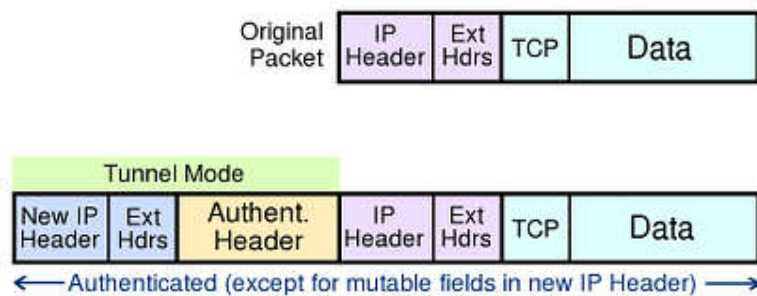
Transport mode thiếu mất quá trình xử lý phần đầu, do đó nó nhanh hơn. Tuy nhiên, nó không hiệu quả đối với giao thức ESP có khả năng không xác nhận mà cũng không mã hoá phần đầu IP.

6.2. Phương thức đường hầm (Tunnel mode)

Không giống với phương thức vận chuyển (Transport mode), phương thức đường hầm (Tunnel mode) bảo vệ toàn bộ gói dữ liệu. Toàn bộ gói dữ liệu IP được đóng gói trong một gói dữ liệu IP khác và một IPSec header được chèn vào giữa phần đầu nguyên bản và phần đầu mới của IP.



Hình 16 - Giao thức ESP trong phương thức đường hầm



Hình 17 - Giao thức AH trong phương thức đường hầm

7. Internet Key Exchange (IKE)

Về cơ bản được biết như ISAKMP/Oakley, ISAKMP là chữ viết tắt của Internet Security Association and Key Management Protocol, IKE giúp các bên giao tiếp hòa hợp các tham số bảo mật và khóa xác nhận trước khi một phiên bảo mật IPsec được triển khai. Ngoài việc hòa hợp và thiết lập các tham số bảo mật và khóa mã hóa, IKE cũng sửa đổi những tham số khi cần thiết trong suốt phiên làm việc.

IKE cũng đảm nhiệm việc xóa bỏ những SAs và các khoá sau khi một phiên giao dịch hoàn thành.

Thuận lợi chính của IKE bao gồm:

- IKE không phải là một công nghệ độc lập, do đó nó có thể dùng với bất kỳ cơ chế bảo mật nào.
- Cơ chế IKE, mặc dù không nhanh, nhưng hiệu quả cao bởi vì một lượng lớn những hiệp hội bảo mật thoả thuận với nhau với một vài thông điệp khá ít.

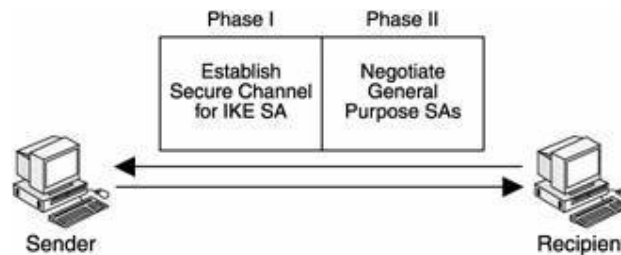
Như vậy, nếu không có giao thức này thì người quản trị phải cấu hình thủ công và những chính sách an ninh trên những thiết bị này được gọi là SA (Security Associate).

Do đó, các thiết bị trong quá trình IKE sẽ trao đổi với nhau tất cả những SA mà nó có và giữa các thiết bị này sẽ tự tìm ra cho mình những SA phù hợp với đối tác nhất.

Những key được trao đổi trong quá trình IKE cũng được mã hoá và những key này sẽ thay đổi theo thời gian (Generate key) để tránh tình trạng bruteforce của những người tấn công vào hệ thống mạng (Attacker).

7.1. IKE Phases

Giai đoạn I và II là hai giai đoạn tạo nên phiên làm việc dựa trên IKE, hình 6-14 trình bày một số đặc điểm chung của hai giai đoạn. Trong một phiên làm việc IKE, nó giả sử đã có một kênh bảo mật được thiết lập sẵn. Kênh bảo mật này phải được thiết lập trước khi có bất kỳ thỏa thuận nào xảy ra.



Hình 18 - Giao thức IKE

7.1.1. Giai đoạn I của IKE

Giai đoạn I của IKE đầu tiên xác nhận các điểm thông tin, và sau đó thiết lập một kênh bảo mật cho sự thiết lập SA. Tiếp đó, các bên thông tin thỏa thuận một ISAKMP SA đồng ý lẫn nhau, bao gồm các thuật toán mã hoá, hàm băm, và các phương pháp xác nhận bảo vệ mã khoá.

Sau khi cơ chế mã hoá và hàm băm đã được đồng ý ở trên, một khoá chỉ sẽ bí mật được phát sinh. Theo sau là những thông tin được dùng để phát sinh khoá bí mật:

- Giá trị Diffie – Hellman.
- SPI của ISAKMP SA ở dạng cookies.
- Số ngẫu nhiên known as nonces (used for signing purposes).

Nếu hai bên đồng ý sử dụng phương pháp xác nhận dựa trên public key, chúng cũng cần trao đổi IDs. Sau khi trao đổi các thông tin cần thiết, cả hai bên phát sinh những key riêng của chính mình sử dụng chúng để chia sẻ bí mật. Theo cách này, những khoá mã hoá được phát sinh mà không cần thực sự trao đổi bất kỳ khoá nào thông qua mạng.

7.1.2. Giai đoạn II của IKE

Trong khi giai đoạn I thỏa thuận thiết lập SA cho ISAKMP, giai đoạn II giải quyết bằng việc thiết lập SAs cho IPsec. Trong giai đoạn này, SAs dùng nhiều dịch vụ khác

nhau thỏa thuận. Cơ chế xác nhận, hàm băm, và thuật toán mã hóa bảo vệ gói dữ liệu IPSec tiếp theo (sử dụng AH và ESP) dưới hình thức một phần của giai đoạn SA.

Sự thỏa thuận của giai đoạn xảy ra thường xuyên hơn giai đoạn I. Điển hình, sự thỏa thuận có thể lặp lại sau 4-5 phút. Sự thay đổi thường xuyên các mã khóa ngăn cản các hacker bẻ gãy những khóa này và sau đó là nội dung của gói dữ liệu.

Tổng quát, một phiên làm việc ở giai đoạn II tương đương với một phiên làm việc đơn của giai đoạn I. Tuy nhiên, nhiều sự thay đổi ở giai đoạn II cũng có thể được hỗ trợ bởi một trường hợp đơn ở giai đoạn I. Điều này làm quá trình giao dịch chậm chạp của IKE tỏ ra tương đối nhanh hơn.

Oakley là một trong số các giao thức của IKE. Oakley is one of the protocols on which IKE is based. Oakley lần lượt định nghĩa 4 chế độ phổ biến IKE.

7.2. *IKE Modes*

4 chế độ IKE phổ biến thường được triển khai:

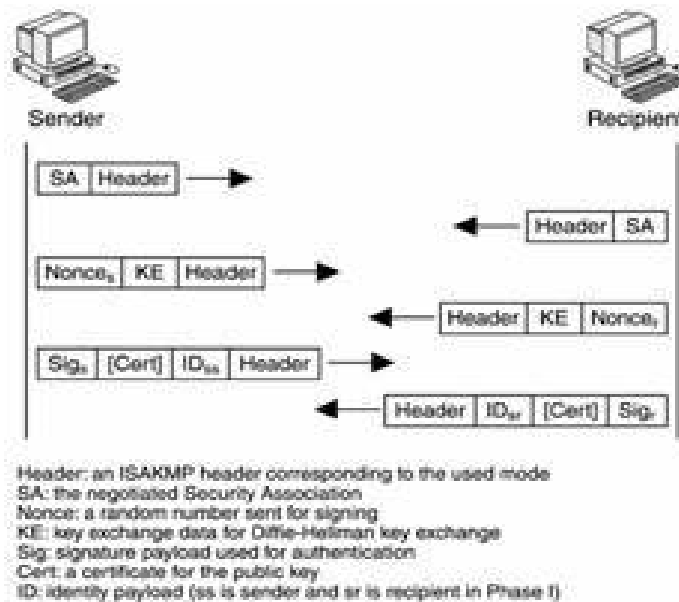
- Chế độ chính (Main mode).
- Chế độ linh hoạt (Aggressive mode).
- Chế độ nhanh (Quick mode).
- Chế độ nhóm mới (New Group mode).

7.2.1. *Main Mode*

Main mode xác nhận và bảo vệ tính đồng nhất của các bên có liên quan trong qua trình giao dịch. Trong chế độ này, 6 thông điệp được trao đổi giữa các điểm:

- 2 thông điệp đầu tiên dùng để thỏa thuận chính sách bảo mật cho sự thay đổi.
- 2 thông điệp kế tiếp phục vụ để thay đổi các khóa Diffie-Hellman và nonces. Những khóa sau này thực hiện một vai trò quan trọng trong cơ chế mã hóa.

Hai thông điệp cuối cùng của chế độ này dùng để xác nhận các bên giao dịch với sự giúp đỡ của chữ ký, các hàm băm, và tùy chọn với chứng nhận.

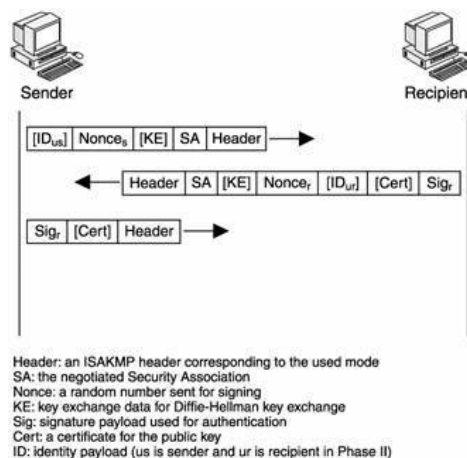


Hình 19 - Mô hình Main mode

7.2.2. Aggressive Mode

Aggressive mode về bản chất giống Main mode. Chỉ khác nhau thay vì main mode có 6 thông điệp thì chế độ này chỉ có 3 thông điệp được trao đổi. Do đó, Aggressive mode nhanh hơn main mode. Các thông điệp đó bao gồm:

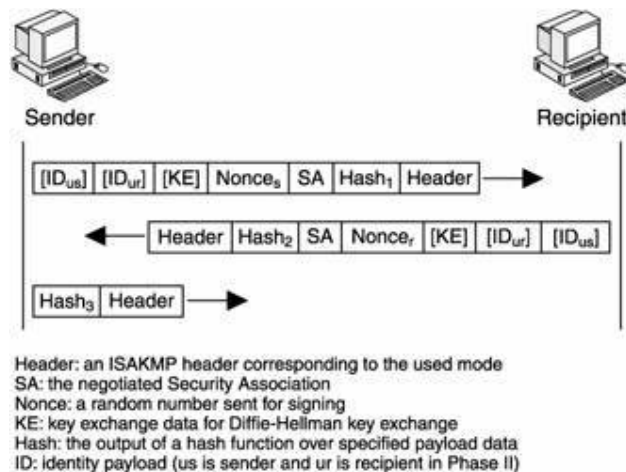
- Thông điệp đầu tiên dùng để đưa ra chính sách bảo mật, pass data cho khóa chính, và trao đổi nonces cho việc ký và xác minh tiếp theo.
- Thông điệp kế tiếp hồi đáp lại cho thông tin đầu tiên. Nó xác thực người nhận và hoàn thành chính sách bảo mật bằng các khóa.
- Thông điệp cuối cùng dùng để xác nhận người gửi (hoặc bộ khởi tạo của phiên làm việc).



Hình 20 - Giai đoạn I dành cho cả Main Mode và Aggressive Mode

7.2.3. Quick Mode

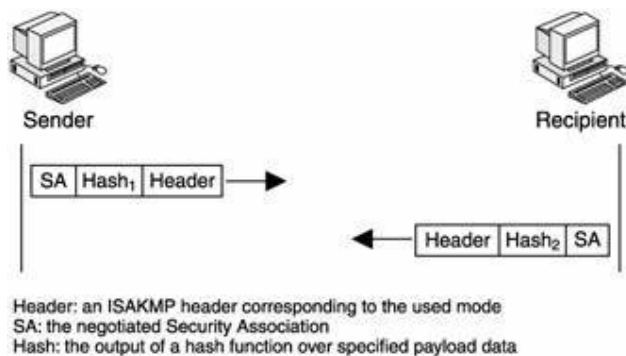
Chế độ thứ ba của IKE, Quick mode, là chế độ trong giai đoạn II. Nó dùng để thỏa thuận SA cho các dịch vụ bảo mật IPSec. Ngoài ra, Quick mode cũng có thể phát sinh khóa chính mới. Nếu chính sách của Perfect Forward Secrecy (PFS) được thỏa thuận trong giai đoạn I, một sự thay đổi hoàn toàn Diffie-Hellman key được khởi tạo. Mặt khác, khóa mới được phát sinh bằng các giá trị băm.



Hình 21 - Quick Mode

7.2.4. New Group Mode

New Group mode được dùng để thỏa thuận một private group mới nhằm tạo điều kiện trao đổi Diffie-Hellman key được dễ dàng. Hình 6-18 mô tả New Group mode. Mặc dù chế độ này được thực hiện sau giai đoạn I, nhưng nó không thuộc giai đoạn II.



Hình 22 - New Group Mode

Ngoài 4 chế độ IKE phổ biến trên, còn có thêm Informational mode. Chế độ này kết hợp với quá trình thay đổi của giai đoạn II và SAs. Chế độ này cung cấp cho các bên có liên quan một số thông tin thêm, xuất phát từ những thất bại trong quá trình thỏa thuận. Ví dụ, nếu việc giải mã thất bại tại người nhận hoặc chữ ký không được xác minh thành công, Informational mode được dùng để thông báo cho các bên khác biết.

8. Chính sách bảo mật IPsec

Mỗi chính sách bao gồm một vài nguyên tắc hay một danh sách các bộ lọc. Ta chỉ có thể gán một chính sách tới một máy tính.

8.1. Một nguyên tắc bao gồm các thành phần sau (*Rules are composed of*)

- Bộ lọc filter (A filter).
- A filter action.
- An Authentication Method (Phương pháp xác thực): Mỗi nguyên tắc có thể chỉ ra nhiều phương pháp xác thực khác nhau.

8.2. Chính sách mặc định của IPsec (*Default policies*)

- Client (Respond Only).
- Server (Request Security).
- Secure server (Require Security).

9. Môi quan hệ giữa chứng chỉ và IPsec

Chứng chỉ X.509 còn được gọi là một chứng chỉ số, là một dạng giấy hình dạng điện tử được trao đổi rộng rãi trong việc xác thực và trao đổi thông tin trên các mạng mở như internet, Extranet (Mạng liên ngành), Intranet (Mạng cục bộ).

Các chứng chỉ được sử dụng như là một phương pháp xác thực của IPsec. Lợi ích của việc sử dụng chứng chỉ với IPsec là:

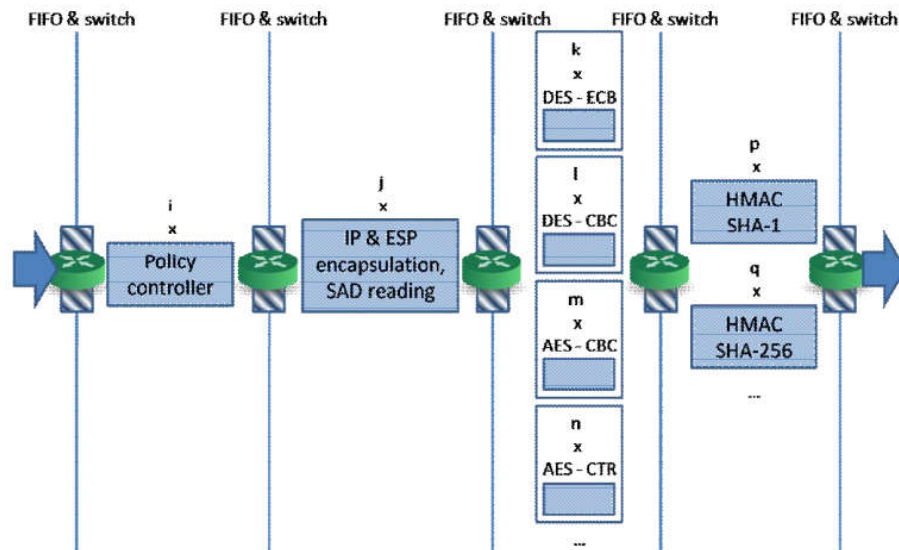
- Như là một phương pháp xác thực giữa 2 host sử dụng IPsec cho phép chúng ta có thể kết nối tin cậy đến một doanh nghiệp này với các tổ chức khác với cùng một CA.
- Sử dụng chứng chỉ cho phép dịch vụ Routing Remote Access có thể truyền dữ liệu qua mạng bảo mật giống như mạng Internet với 1 Router có hỗ trợ IPsec.
- Sử dụng chứng chỉ khi sử dụng mô hình VPN Client và Mô hình VPN Site – to – site Sử dụng giao thức đường hầm L2TP/IPsec.

Để sử dụng chứng chỉ trong quá trình xác thực IPsec thì cả 2 máy tính đều phải cung cấp một chứng chỉ hợp lệ dùng cho mục đích truyền thông của IPsec. Khi một máy tính cấu hình chứng chỉ thì ta phải sử dụng chính sách IPsec để hỗ trợ chứng chỉ sử dụng như là một phương pháp xác thực hợp lệ.

II. Nghiên cứu các cấu trúc phần cứng IPsec thực hiện

1. IPsec Gateway

Tối ưu hoá thiết kế phần cứng đã được áp dụng cho các thiết kế trước đó, dẫn đến tăng tốc độ xử lý và sử dụng các cổng logic tốt hơn.



Hình 23 - Kiến trúc IPsec Gateway

Thiết kế các chức năng IPsec chia thành nhiều phần. Mỗi phần bao gồm nhiều module thực hiện một công việc, một giai đoạn cụ thể của điều chế, làm việc đồng thời với nhau. Các phần được phân chia bằng FIFO và vòng robin switch. Như thế, nhiều gói tin có thể xử lý cùng một lúc, tăng hiệu năng của thiết kế cũng như giảm thiểu thời gian xử lý các gói tin tức. Song bên cạnh đó thiết kế cũng có thể kiểm soát số lượng gói tin trong lúc xử lý. Việc tăng số lượng các module tính toán HMAC theo thời gian nhất định dẫn đến việc có thể tăng thông lượng.

Tối ưu hoá kiến trúc thiết kế: Độ rộng của kiến trúc BUS nội tăng từ 64bit lên đến 128bit (kích thước của khối thiết kế AES) để nâng cao hiệu suất của thiết kế. Hai đường thiết kế DES xử lý song song (độ rộng 64bit) được đặt trong các module mã hoá DES-ECB.

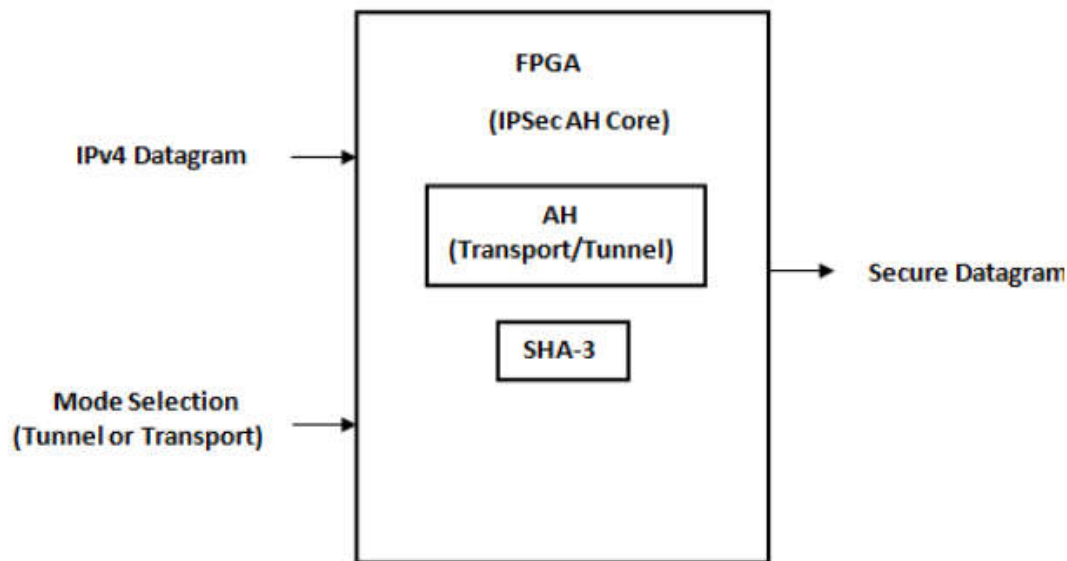
Việc sử dụng nhiều lõi xử lý song song được tạo ra một vấn đề trong việc cung cấp quyền truy cập và Hiệp hội bảo mật Cơ sở dữ liệu (Security Association Database). Để giải quyết vấn đề này, hệ thống phân định mức độ quan trọng được thêm vào trong thiết kế. Hơn nữa, các truy cập vào Cơ sở dữ liệu đã được di chuyển đến một miền clock khác, vì bộ nhớ DDR được thiết kế với tần số hoạt động cao hơn rất nhiều lần so với bộ phận thiết kế.

2. SECURE HASH ALGORITHM -3 (SHA-3)

Hàm băm có thể được sử dụng để xác minh tính đúng đắn của dữ liệu. Đây là một thủ tục xác định một chiều mà đầu vào là một khối dữ liệu tùy ý và có đầu ra là một chuỗi bit có kích thước cố định, được gọi là giá trị băm. Một tín hiệu được mã hoá với giá trị băm là một dấu hiệu nhận biết đặc trưng riêng của dữ liệu và nếu dữ liệu thay đổi, nó sẽ thay đổi dấu hiệu nhận biết đặc trưng riêng của tín hiệu. Băm của một phần dữ liệu được tính toán và gắn vào dữ liệu. Khi tín hiệu gửi đến đích, hàm băm được tính toán lại từ dữ liệu và so sánh nó với băm, đã được thêm vào thông điệp ban đầu. Nếu các giá trị không khớp nhau, điều đó có nghĩa là dữ liệu đã được bị hỏng và sẽ không được xử lý. Các hàm băm được sử dụng trước đây (SHA-0, SHA-1, SHA-2, RIPEMD và MD5) đã được tìm thấy có lỗ hổng trong thiết kế được mô tả trong [14] [15] [16]. Mặc dù, không có cuộc tấn công nào vào SHA-2 đã báo cáo, nhưng sự giống nhau về thuật toán của SHA-2 với SHA-1, làm cho thuật toán này dễ bảo mật. Đó tại sao chúng tôi triển khai băm mật mã mới được chọn chức năng SHA-3, để cung cấp xác minh tính toàn vẹn dữ liệu. Điều này thuật toán mới được công bố bởi Viện quốc gia Tiêu chuẩn và Công nghệ (NIST), Hoa Kỳ năm 2012 [17]. Của chúng tôi Kỹ thuật triển khai SHA-3 được trình bày trước đó [18] là được sử dụng để thực hiện giao thức AH của IPsec.

3. Kiến trúc IPsec AH

Việc triển khai IPsec AH được trình bày liên quan đến phương thức vận chuyển và đường hầm của hoạt động và điều này lõi có thể cấu hình lại có khả năng bảo mật datagram IPv4 như hình bên dưới:



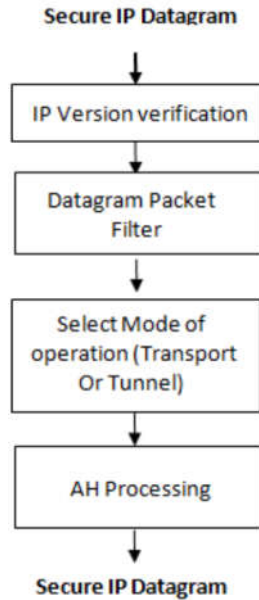
Hình 24 - Cấu hình lại lõi IPsec AH

Lõi IPsec AH có thể cấu hình lại này có thể được cấu hình thủ công cho chế độ hoạt động mong muốn. Việc thực hiện các bước của sơ đồ được trình bày trong Hình 25. 04 bit ban đầu của gói datagram được sử dụng để kiểm tra phiên bản IP. Điều này kiểm tra là cần thiết để có ý tưởng về cấu trúc của datagram. Vì vậy, các trường datagram khác nhau có thể là truy cập và cập nhật trong quá trình xử lý AH. Tiêu đề IP bit được trích xuất từ datagram IP. Khi phiên bản IP được xác minh, gói datagram được truyền qua gói bộ lọc, được sử dụng để quyết định xử lý AH là bắt buộc hay không. Lọc gói này hoạt động như một chính sách bảo mật cho IPsec. Sử dụng lọc gói này, gói datagram là bỏ, bỏ mà không áp dụng IPsec hoặc chuyển tiếp với xử lý IPsec. Trong quá trình thực hiện này, chúng tôi đã kiểm tra Field Giao thức mạng trường của datagram IP, nếu trường này bằng '51, (số giao thức của giao thức AH), có nghĩa là đã nhận được datagram đã được bảo mật. Trong trường hợp này, datagram là chuyển tiếp mà không áp dụng giao thức AH. Tương tự, bởi áp dụng kiểm tra địa chỉ IP của một mạng cụ thể, IP datagram có thể được loại bỏ và chuyển tiếp có và không có áp dụng giao thức AH. Khối tiếp theo của Hình 25 liên quan đến lựa chọn chế độ của operation được cấu hình thông qua đầu vào. Bước cuối cùng là xử lý giao thức AH chính, được mô tả dưới đây.

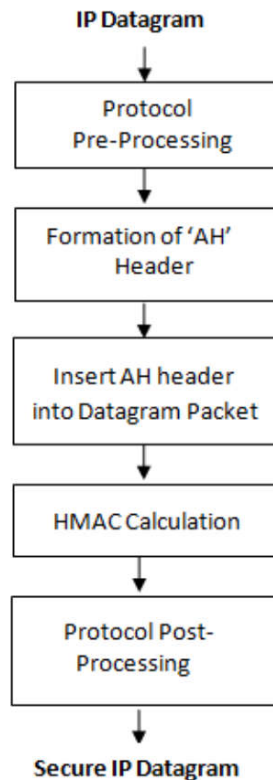
3.1. Protocol Pre-processing

Giao thức tiền xử lý được sử dụng để chuẩn bị IP tiêu đề để tính toán ICV và phụ thuộc vào lựa chọn phương thức hoạt động. Trong chế độ vận chuyển, chỉ có tiêu đề IP là được cập nhật, trong khi ở chế độ tunnel, một tiêu đề IP mới được tạo. Trong chế độ transport, tiêu đề IP được cập nhật bằng cách thay thế các trường có thể thay đổi (được sửa đổi trong quá trình vận chuyển) của tiêu đề IPv4 bởi không và cũng bằng cách cập nhật một số lĩnh vực khác; như "Giao thức" trường của tiêu đề IPv4 bởi '51, (trong đó 51 51 là giao thức số cho AH). Ngoài ra, trường "Tổng chiều dài của IPv4 tiêu đề" được cập nhật bằng cách thêm độ dài AH trong bản gốc chiều dài của datagram. Trong chế độ tunnel, các trường của tiêu đề IP mới được đặt thành các trường của tiêu đề IP gốc và các trường có thể thay đổi được thay thế bằng không. Trường "Giao thức" mạng của IPv4 mới tiêu đề được thay thế bằng 51, giống như chế độ vận chuyển. Nhưng Tổng chiều dài trường của tiêu đề IPv4 mới được cập nhật bởi thêm chiều dài AH và

độ dài tiêu đề IP theo chiều dài ban đầu của datagram (vì đóng gói tiêu đề IP trong đường hầm chế độ). Địa chỉ nguồn và đích của tiêu đề IP mới địa chỉ IP của các thiết bị giữa chúng là đường hầm hình thành. Các chi tiết về các lĩnh vực có thể thay đổi và bất biến của Tiêu đề IPv4 được đưa ra trong [7]. Tiêu đề IP được xử lý trước này là được sử dụng để tính giá trị ICV trong quá trình hình thành Tiêu đề AH.



Hình 25 - Sơ đồ thực hiện đề xuất



Hình 26 - Các bước xử lý AH

3.2. AH header

Tiêu đề AH liên quan đến việc tạo ra 06 lĩnh vực. 06 lĩnh vực này được chi tiết trong bảng 1. Trong giao thông vận tải chế độ 'Trường tiêu đề tiếp theo của tiêu đề AH được đặt thành Field Giao thức trường của tiêu đề IPv4. Trong chế độ đường hầm, 'Tiếp theo trường tiêu đề của AH đại diện cho IP được đóng gói datagram, đó là lý do tại sao nó được đặt thành "4", trong đó 4 là giao thức số lượng IPv4. Trường độ dài AH tổng chiều dài của Tiêu đề AH, phụ thuộc vào độ dài của giá trị HMAC. Các độ dài của giá trị HMAC dựa trên mật mã đã chọn hàm băm. Trong công việc này, để tính giá trị HMAC, chúng tôi chọn biến thể 256-bit của SHA-3. Ngoài ra, độ dài AH là được đề cập trong các từ 32 bit trừ đi 2. Điểm trừ này là hai bởi vì thực tế AH về cơ bản là một tiêu đề mở rộng của IPv6, độ dài thường được đo bằng các từ 64 bit, không phải bằng từ 32 bit. Do đó, ở đây cho IPv4, AH len = [3 (Các trường cố định 32 bit của tiêu đề AH) + 8 (các từ 32 bit cho Giá trị HMAC)] - 2. Trường tiếp theo của tiêu đề AH là 'Dành riêng được đặt thành không phân biệt chế độ đã chọn. Trường 'SPI cũng được đặt thành 0 cho biết không hiệp hội an ninh tồn tại. Trường thứ tự 'số thứ tự là được tạo bằng cách sử dụng bộ đếm không dấu 32 bit, làm tăng bởi một bất cứ khi nào tiêu đề AH được tạo. Trình tự số gói datagram an toàn đầu tiên là 1. Trường cuối cùng của Tiêu đề AH là field Trường dữ liệu xác thực, được đặt thành Giá trị HMAC. Như đã đề cập trước đó, ở đây chiều dài của Giá trị HMAC là 256-bit và được đặt thành 0 cho HMAC phép tính.

3.3. Chèn tiêu đề AH vào IP datagram

Trước khi tính toán giá trị HMAC, AH được tạo tiêu đề được chèn vào giao thức IP datagram được xử lý trước. Bước này dẫn đến datagram được cập nhật được sử dụng để tính toán HMAC.

3.4. Tính toán HMAC

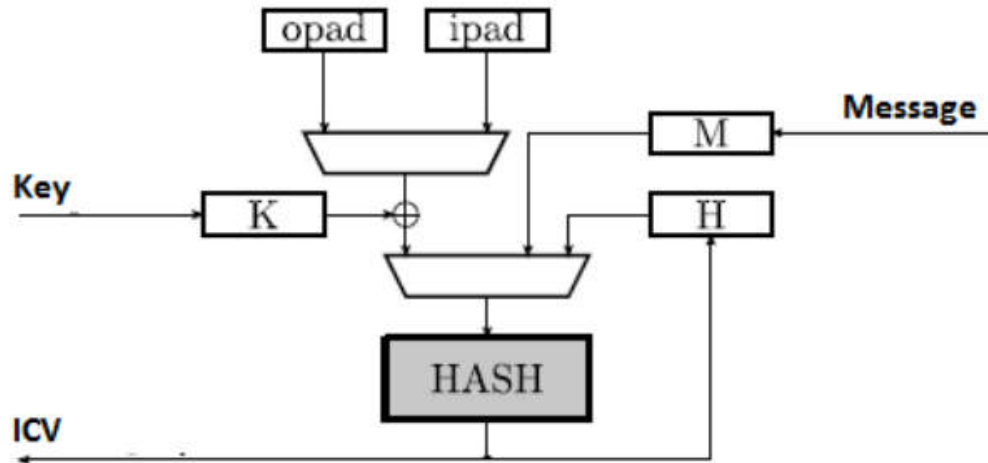
Để cung cấp tính toàn vẹn và tính xác thực của dữ liệu trong Giao thức AH, IPsec định nghĩa một HMAC đối xứng [19] được sử dụng để tạo ICV. Ưu điểm chính của một HMAC là tốc độ tương đối cao so với kỹ thuật số chữ ký. IPsec yêu cầu sử dụng HMAC xây dựng, cung cấp một thẻ xác thực kích thước cố định cho các tin nhắn tùy ý. Giá trị HMAC hoặc ICV của một tin nhắn x được tính là

$$H(K \text{ XOR } opad, H(K \text{ XOR } ipad, x))$$

Trong đó:

- H là hàm băm mật mã
- K là chìa khóa bí mật

Để làm cho việc xây dựng an toàn, cần phải có độ dài khóa được sử dụng bằng hoặc lớn hơn độ dài đầu ra của hàm băm được sử dụng H. Việc thực hiện một trình bao bọc HMAC, được cung cấp hàm băm H, được hiển thị trong Hình 9. Ở đây, 'tin nhắn được cập nhật datagram sau khi chèn tiêu đề AH.



Hình 27 - Kiến trúc tính toán HMAC

3.5.Protocol Post-Processing

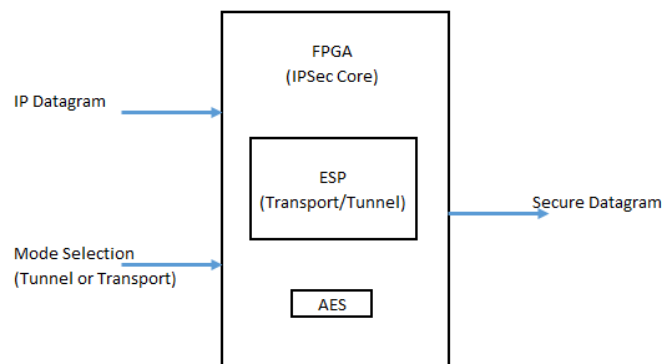
Việc thực hiện hậu xử lý giao thức liên quan đến ba bước:

- (1) Trường data Dữ liệu xác thực của tiêu đề AH được đặt thành giá trị ICV được tính toán.
- (2) Các trường có thể thay đổi, được thay thế bằng 0 để tính toán ICV được đặt trở lại giá trị ban đầu của chúng.
- (3) Một bước bổ sung là đã triển khai i.e để tính lại tổng kiểm tra tiêu đề và cập nhật trường checks tổng kiểm tra tiêu đề của IPv4.

Bây giờ, an toàn datagram đã sẵn sàng để chuyển tiếp. Để tính toán tổng kiểm tra tiêu đề, 160 bit của IP được cập nhật tiêu đề được chia thành các từ 16 bit. Những từ 16 bit này được thêm vào với nhau và kết quả là thông số 16 bit được đảo ngược. Đầu ra đảo ngược này là tổng kiểm tra tiêu đề cần thiết.

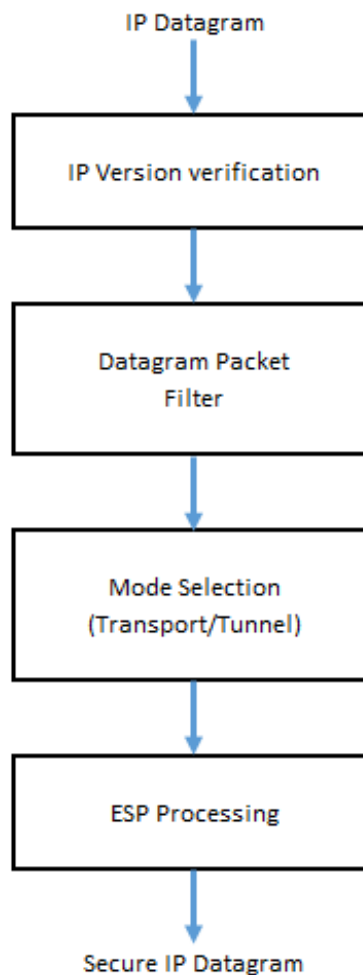
4. Kiến trúc IPsec ESP

Việc triển khai IPsec ESP được trình bày liên quan đến vận chuyển và chế độ đường hầm của các hoạt động như trong Hình 29. Lỗi có thể cấu hình lại này có khả năng bảo mật datagram IPv4.



Hình 28 - Cấu hình lại lõi IPsec ESP

Lỗi IPsec ESP có thể cấu hình lại này là thủ công cấu hình để hoạt động trong chế độ hoạt động mong muốn. Các các bước chương trình đã thực hiện được trình bày trong Hình 28. Ban đầu 04 bit của gói datagram được sử dụng để kiểm tra phiên bản IP. Điều này kiểm tra là cần thiết để có ý tưởng về cấu trúc của datagram. Vì vậy, các trường datagram khác nhau có thể được truy cập và được cập nhật trong quá trình xử lý ESP. Các bit tiêu đề IP là trích xuất từ datagram IP. Khi phiên bản IP được xác minh, gói datagram được truyền qua bộ lọc gói, nghĩa là được sử dụng để quyết định xử lý ESP có cần thiết hay không. Điều này lọc gói làm việc như một chính sách bảo mật cho IPsec. Sử dụng cái này lọc gói, gói datagram bị bỏ, chuyển tiếp không áp dụng IPsec hoặc chuyển tiếp với xử lý IPsec. Trong triển khai này, chúng tôi đã kiểm tra field giao thức mạng IP của IP datagram, nếu trường này bằng “50”, (số giao thức của ESP giao thức), nó có nghĩa là datagram nhận được đã được bảo mật. Trong này trường hợp datagram được chuyển tiếp mà không áp dụng giao thức ESP. Tương tự, bằng cách áp dụng kiểm tra trên địa chỉ IP của một cụ thể mạng, IP datagram có thể được loại bỏ và chuyển tiếp với và mà không áp dụng giao thức ESP. Khối tiếp theo của Hình 28 liên quan đến lựa chọn chế độ hoạt động được cấu hình thông qua đầu vào. Bước cuối cùng là xử lý giao thức ESP chính Giao thức ESP bao gồm cả, mã hóa và xác thực (tùy chọn). Trong công việc này, chúng tôi đã thực hiện chức năng mã hóa của ESP. Việc xử lý ESP được đưa ra trong hình 29 và nó thẳng tiến so với AH xử lý [14] vì nó không liên quan đến giao thức trước và sau chế biến.



Hình 29 - Các bước thực hiện của IPsec ESP

4.1. Trích xuất bit Payload IP từ datagram

Các bit Payload IP được trích xuất từ datagram, bởi vì những bit được sử dụng trong tiêu đề ESP và cũng cần được mã hóa. Trong IPv4, độ dài của các trường được trích xuất này bằng “Tổng chiều dài - Chiều dài tiêu đề”.

4.2. Chuyển đổi của ESP header/trailer

Các trường giao thức ESP được trình bày chi tiết trong Bảng 6. ‘SPI, trường được đặt thành 0, cho biết không có liên kết bảo mật tồn tại Trường ‘số thứ tự được tạo bằng cách sử dụng 32-bộ đếm bit không dấu, tăng lên bất cứ khi nào ESP tiêu đề được tạo ra. Số thứ tự an toàn đầu tiên gói datagram là 1. Trường ‘Padding của đoạn giới thiệu ESP được sử dụng ở đây để đảm bảo độ dài của văn bản tuân theo yêu cầu của lỗi AES để mã hóa. Trường ‘Padding chiều dài của đoạn giới thiệu ESP được đặt đến số byte được chèn vào trường đệm. Trong giao thông vận tải chế độ field Trường tiêu đề tiếp theo của tiêu đề ESP được đặt thành ‘Giao thức lĩnh vực của tiêu đề IPv4. Trong chế độ đường hầm, trường header Tiêu đề tiếp theo của ESP đại diện cho datagram IP được đóng gói, đó là lý do tại sao được đặt thành ‘4, trong đó 4 là số giao thức của IPv4. Tải trọng IP được chèn giữa tiêu đề ESP và trailer ESP trong Chế độ vận chuyển, trong khi ở chế độ đường hầm, IP datagram hoàn thành được thêm vào giữa tiêu đề ESP và trailer.

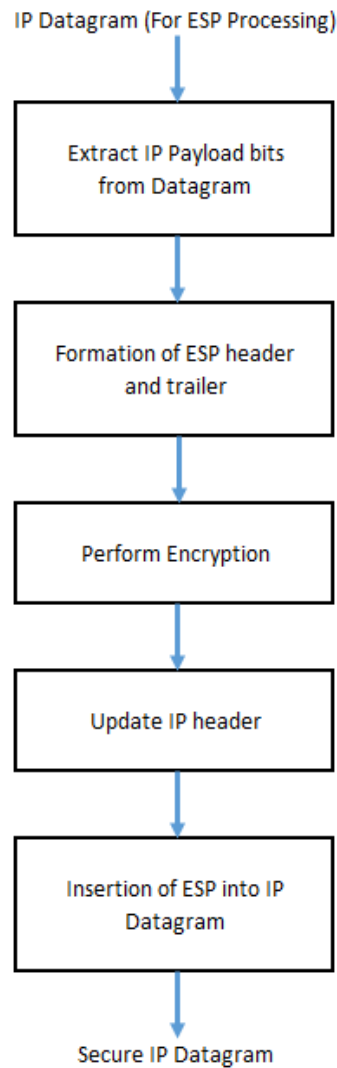
| Field | Length |
|--------------------------------|------------|
| Security Parameter Index (SPI) | 32bit |
| Sequence Number | 32bit |
| Payload Data | Variable |
| Padding | 0-255bytes |
| Padding length | 8bit |
| Next header | 8bit |
| Authentication Data | Variable |

Bảng 6 - Tải trọng bảo mật đóng gói (ESP)

4.3. Thực hiện mã hoá

AES là một thuật toán mã hóa khối đối xứng xử lý dữ liệu cố định của khối 128 bit. Mật mã khối có nghĩa là số byte mà nó mã hóa được cố định, tức là 16 byte. Nó hỗ trợ kích thước khóa 128, 192 và 256 bit với các vòng lặp 10, 12 và 14 tương ứng. Số lượng các vòng này được chọn tùy thuộc vào kích thước khóa Một đơn vị mở rộng Khóa riêng biệt là được sử dụng để tạo khóa cho mỗi vòng thuật toán AES. Một chút chuỗi liên quan đến đầu vào, đầu ra và khóa mật mã là được xử lý như các mảng byte; gọi là Nhà nước. Mảng nhà nước bao gồm bốn hàng byte và mỗi hàng gồm 4 hàng byte. Trong mỗi vòng AES, khối dữ liệu 128 bit là được biến đổi bởi một chuỗi các hoạt động như được đưa ra trong [12]. Của chúng tôi trước đó đã trình bày kỹ thuật thực hiện AES [13], trong đó sử dụng tài nguyên BRAM và LUT của FPGA, được sử dụng trong AES Chế độ truy cập ở đây để cung cấp mã hóa cần thiết cho IPsec Lỗi ESP.

Phạm vi mã hóa phụ thuộc vào chế độ hoạt động được lựa chọn. Đối với IP chế độ vận chuyển tải trọng và trailer ESP được mã hóa và cho chế độ đường hầm hoàn thành IP datagram và trailer ESP được mã hóa. AES lỗi được sử dụng trong chế độ bộ đếm, sao cho số lượng 128 bit mã hóa khối có thể được thực hiện trong paralle.



Hình 30 - Các bước thực hiện ESP

4.4. Cập nhật IP header

Trong IPv4, hai trường tiêu đề được cập nhật:

- (1) Giao thức.
- (2) Kiểm tra tiêu đề.

Trường “Giao thức” được đặt thành “Giao thức” số lượng ESP 50 cho cả chế độ vận chuyển và đường hầm. Do thay đổi này, một tổng kiểm tra tiêu đề mới được tính toán và được đặt vào trường Tiêu đề tổng kiểm tra của tiêu đề IPv4.









4.5. Chèn ESP vào IP datagram

Bước cuối cùng trong xử lý ESP là chèn ESP vào IP datagram và nó phụ thuộc vào chế độ hoạt động và IP được chọn phiên bản. Bây giờ, datagram IP an toàn đã sẵn sàng để chuyển tiếp.

CHƯƠNG 4

KẾ HOẠCH LUẬN VĂN

Kế hoạch dự kiến thực hiện đề tài:

| Công việc | Jul.2019 | Aug.2019 | Sep.2019 | Oct.2019 | Nov.2019 | Dec.2019 |
|--------------------------------------|--|---|--|---|---|----------|
| Viết báo |  | | | | | |
| Thiết kế bộ SHA-3 |  | | | | | |
| Kiểm tra thiết kế bộ SHA-3 |  | | | | | |
| Thiết kế kiến trúc IPsec AH | |  | | | | |
| Kiểm tra thiết kế kiến trúc IPsec AH | | |  | | | |
| Thiết kế kiến trúc ESP | | |  | | | |
| Kiểm tra thiết kế kiến trúc ESP | | | |  | | |
| Hiện thực hoá – mô phỏng lên FPGA | | | | |  | |

Tài liệu tham khảo

- [1] IPsec performance, OpenWRT project webpage (accessed 17.05.16), <https://oldwiki.archive.openwrt.org/doc/howto/vpn.IPsec.performance>
- [2] StrongSwan project webpage, (accessed 17.05.16), <https://www.strongswan.org>
- [3] Klassert Steffen, 2010, Parallelizing IPsec, https://www.strongswan.org/docs/Steffen_Klassert_Parallelizing_IPsec.pdf
- [4] Intel Corporation, 2012, Fast Multi – buffer IPsec Implementations on Intel Architecture Processor.
- [5] Cisco Systems, Inc., 2008, Cisco IPsec and SSL VPN Solutions Portfolio.
- [6] Juniper Networks, 2015, Security Products Comparison Chart.
- [7] Helion Technology Limited, IPsec ESP IP Core for FPGA – Product Brief, (accessed 19.05.16), <http://www.heliontech.com/IPsec.htm>
- [8] Mateusz Korona, 2015, Implementation of IPsec protocol suite using field – programmable devices, bachelor thesis.
- [9] Krawczyk H., Bellare M., Canetti R., RFC 2404, 1997, HMAC: Keyed Hashing for Message Authentication.
- [10] Eastlake D. 3rd, Jones P., RFC 3174, 2001, US Secure Hash Algorithm 1 (SHA1).
- [11] A. Ferrante, V. Piuri, and J. Owen, “IPsec Hardware Resource Requirements Evaluation”, Next Generation Internet Networks (NGI 2005), April 2005. pp.240-246, “DOI:10.1109/NGI.2005.1431672”.
- [12] W. Vander, K. Benkrid, “High-Performance Computing Using FPGAs”, Springer book, ISBN: 978-1-4614-1790-3.
- [13] Rao M., Newe T. and Grout I., “AES implementation on Xilinx FPGAs suitable for FPGA based WBSNs”. 9th International Conference on Sensing Technology (ICST 2015), 08 Dec - 10 Dec 2015, Auckland, New Zealand.
- [14] Rao M., Newe T. and Grout I., Lewis E., Mathur A. “FPGA Based Reconfigurable IPsec AH Core Suitable for IoT Applications”. 13th IEEE International Conference on Pervasive Intelligence and computing (PICOm - 2015), Liverpool, United Kingdom.
- [15] M. McLoone and J. McCanny, “A single-chip IPSEC cryptographic processor,” in *Signal Processing Systems, 2002. (SIPS '02). IEEE Workshop on*, Oct 2002, pp. 133–138.
- [16] J. Lu and J. Lockwood, “Ipsec implementation on xilinx virtex-ii pro fpga and its application,” in *Reconfigurable Architecture Workshop, RAW*, 2005.

- [17] A. P. Kakarountas, H. Michail, A. Milidonis, C. E. Goutis, and G. Theodoridis, “High-speed fpga implementation of secure hash algorithm for ipsec and vpn applications,” *The Journal of Supercomputing*, vol. 37, no. 21, pp. 179–195, Aug 2006.
- [18] P. R. Schaumont, *A Practical Introduction to Hardware/Software Codesign*. Springer, 2010.
- [19] H. Michail, G. Athanasiou, A. Gregoriades, C. L. Panagiotou, and S. Goutis, “High throughput hardware/software co-design approach for sha-256 hashing cryptographic module in ipsec/ipv6,” *Global Journal of Computer Science and Technology*, vol. 10, no. 4, pp. 54–59, June 2010.
- [20] An Introduction to the Helion IPsec ESP Engine, v. 1.0.0 ed., *Helion Technology Limited*, 2006.
- [21] A. A. Salman, “IPSec implementation in embedded systems for partial reconfigurable platforms,” Masters Thesis, ECE Department, George Mason University, Fairfax, Virginia, USA, May 2011.
- [22] R. Chaves, G. Kuzmanov, L. Sousa, and S. Vassiliadis, “Improving sha-2 hardware implementations,” in *Cryptographic Hardware and Embedded Systems - CHES 2006*, Oct 2006, pp. 298–310.
- [23] H. Orup, “Simplifying quotient determination in high-radix modular multiplication,” in *Proceedings of the 12th Symposium on Computer Arithmetic*, Jul 1995, pp. 193–199.
- [24] M. Joye and S.-M. Yen, “The montgomery powering ladder,” in *Cryptographic Hardware and Embedded Systems CHES 2002*, ser. Lecture Notes in Computer Science, B. Kaliski, Cetin K. Koc., and C. Paar, Eds., vol. 2523. Springer-Verlag, 2002, pp. 291–302.
- [25] D. Suzuki, “How to maximize the potential of fpga resources for modular exponentiation,” in *Workshop on Cryptographic Hardware and Embedded Systems—CHES 2007*. Berlin: Springer-Verlag, 2007.
- [26] E. Oksa and E. Savas, “Parametric, secure and compact implementation of RSA on FPGA,” in *Reconfigurable Computing and FPGAs, 2008. ReConFig '08. International Conference on*, Dec. 2008, pp. 391–396.
- [27] Hardware Interface of a Secure Hash Algorithm (SHA), v. 1.4 ed., *Cryptographic Engineering Research Group, George Mason University*, Jan 2010.
- [28] R.L. Rivest, M. J. B. Robshaw, R. Sidney, and Y. L. Yin. The RC6 Block Cipher. In First Advanced Encryption Standard (AES) Conference, 1998.
- [29] Jean-Luc Beuchat. FPGA Implementations of the RC6 Block Cipher. In P. Y. K. Cheung, G. A. Constantinides, and J. T. de Sousa, editors, *Field-Programmable Logic and Applications*, number 2778 in Lecture Notes in Computer Science, pages 101–110. Springer, 2003.