Một giải pháp quản lý kết nối mật cho IPSec trên FPGA

Phan Văn Kỷ, La Hữu Phúc

***Tóm tắt*—IPSec (Internet Protocol Security) là bộ giao thức an toàn nhằm bảo vệ lưu lượng dữ liệu qua mạng Internet. Mỗi kết nối mật trong mô hình triển khai IPSec có một bộ thuật toán, tham số bảo mật riêng. Để đảm bảo các kết nối mật hoạt động ổn định trong môi trường truyền tin với băng thông lớn, việc quản lý nhiều kết nối mật đồng thời trên thiết bị IPSec đóng vai trò vô cùng quan trọng. Do tính phức tạp của quá trình quản lý, thông thường vấn đề này được thực hiện bằng phần mềm trên hệ điều hành. Giải pháp này bị hạn chế do quá trình trao đổi dữ liệu giữa vi mạch Field Programmable Gate Array (FPGA) và bộ vi xử lý. Trong bài viết này, nhóm tác giả đưa ra một giải pháp tổ chức, quản lý kết nối mật sau khi sử dụng giao thức Internet Key Exchange (IKE) để trao đổi khóa cho IPSec trên FPGA sử dụng ngôn ngữ mô tả phần cứng, nhằm đáp ứng yêu cầu tốc độ cao với nhiều kết nối.**

***Abstract*—IPSec (Internet Protocol Security) is a secure protocol aiming to protect data traffic via the Internet. There is a separate set of algorithms and security parameters in each secure connection in the IPSec deployment model. In order to ensure stable connections in high-bandwidth environments, managing multiple secure connections simultaneously on IPSec devices holds a significant role. Due to the complexity of the management process, this is commonly done by software on the operating system. This solution is restricted due to data exchange between field-programmable gate array (FPGA) and microprocessor. In this article, a solution was proposed to organize and manage a confidential connection after using Internet Key Exchange (IKE) to exchange keys for IPSec directly using hardware description language on FPGA, aiming to meet high-speed requirements with many connections.**

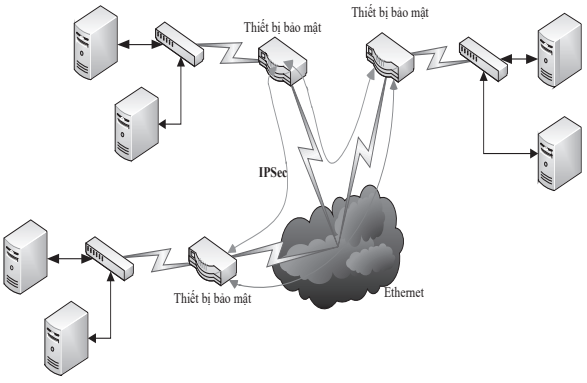
Từ khóa—IPSec; IKE; FPGA; ESP; bảo mật đóng gói.

Keyword—IPSec; IKE; FPGA; ESP; Encapsulating Security Payload.

Bài báo được nhận ngày 14/4/2021. Bài báo được nhận xét bởi phản biện thứ nhất ngày 14/5/2021 và chấp nhận đăng ngày 20/7/2021. Bài báo được nhận xét bởi phản biện thứ hai ngày 13/6/2021 và chấp nhận đăng ngày 05/7/2021.

I. GIỚI THIỆU

IPSec là một bộ giao thức bảo mật được chuẩn hóa bởi Nhóm đặc nhiệm kỹ thuật internet (Internet Engineering Task Force) với mục đích cung cấp các cơ chế mã hóa, xác thực và đảm bảo tính toàn vẹn dữ liệu qua mạng bằng giao thức IP. Một mô hình triển khai IPSec thường sử dụng được thể hiện như trên Hình 1 [2].



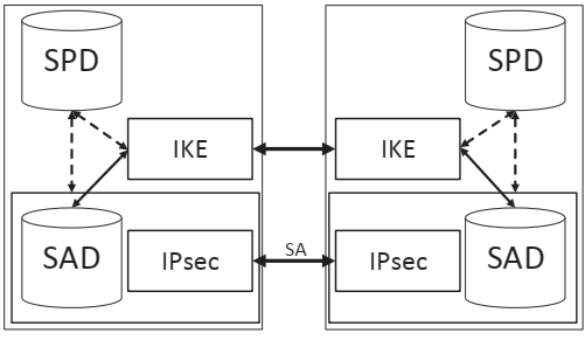
Hình 1. Mô hình triển khai IPSec.

Khi mỗi cặp mạng (thiết bị bảo mật) trong mô hình triển khai IPSec thiết lập kết nối, chúng phải đồng nhất về một tập hợp các giao thức bảo mật, thuật toán mã hóa và khóa mã. Một trong những giao thức thực hiện quá trình trao đổi và thỏa thuận khóa trong chế độ bảo mật IPSec được sử dụng rộng rãi và hiệu quả đó là IKE [3]. Mỗi kết nối trong mô hình triển khai IPSec sử dụng giao thức IKE sẽ có một bộ tham số bảo mật và thuật toán riêng. Do đó, trong một mô hình triển khai có nhiều thiết bị, để đảm bảo các kết nối hoạt động ổn định trong môi trường truyền tin với băng thông lớn, đòi hỏi việc quản lý khóa cho các kết nối đồng thời trên thiết bị bảo mật IPSec đóng vai trò vô cùng quan trọng.

Mặt khác, mỗi một kết nối đòi hỏi nhiều cặp khóa khác nhau nhằm phục vụ cho quá trình mã hóa/giải mã dữ liệu hoặc xác thực tính toàn vẹn của dữ liệu trong cấu trúc bảo mật đóng gói

(Encapsulating Security Payload - ESP) [2]. Như vậy, mỗi kết nối sẽ có 4 cặp khóa khác nhau bao gồm: một khóa cho mã hóa, một khóa cho giải mã và một cặp khóa phục vụ cho quá trình xác thực tính toàn vẹn dữ liệu.

Hiện nay, các sản phẩm bảo mật IPSec thường được thiết kế dựa trên công nghệ FPGA, vì công nghệ này cho phép chế tạo ra các thiết bị có tốc độ mã hóa và độ bảo mật cao. Trong đó, phần mật mã sẽ được thực hiện bằng ngôn ngữ mô tả phần cứng, giao thức trao đổi khóa sẽ được thực hiện bằng phần mềm hoặc bộ vi xử lý nhúng được thể hiện như trên sơ đồ Hình 2 [7].



Hình 2. Cấu trúc hệ thống IPSec.

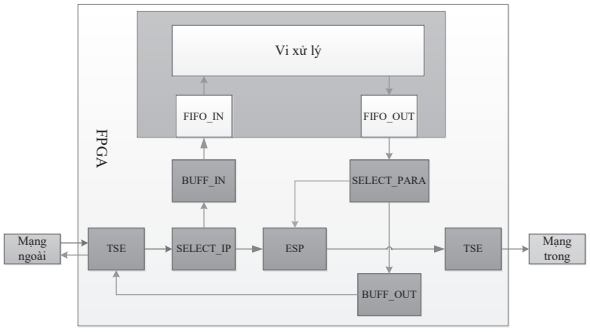
Có rất nhiều giải pháp triển khai IPSec được công bố, tuy nhiên đa phần không trình bày giải pháp tổ chức SAD cho nhiều kết nối mật với nhiều cặp khóa khác nhau đối với từng kết nối [7], [8]. Bên cạnh đó, cũng có một số sản phẩm IPSec thương mại sử dụng FPGA [9] hoặc GPU [10], nhưng các sản phẩm này không cung cấp giải pháp thực thi IPSec.

Để đáp ứng được yêu cầu tốc độ cao cho các kết nối trong mô hình IPSec, sau khi thực hiện trao đổi khóa IKE xong, việc quản lý và phân phối khóa cho từng kết nối trên FPGA là vô cùng quan trọng, nó ảnh hưởng tới quá trình thực hiện đóng gói ESP, mã hóa/giải mã và xác thực tính toàn vẹn dữ liệu, dẫn tới ảnh hưởng tốc độ truyền/nhận dữ liệu của thiết bị. Trong bài báo này, nhóm tác giả trình bày một giải pháp nhằm quản lý khóa cho nhiều kết nối và phân phối khóa cho từng kết nối sử dụng kit DE4 của hãng Intel.

II. MÔ HÌNH THỰC HIỆN FPGA

A. Mô hình giao tiếp giữa bộ vi xử lý và FPGA

Để thực hiện truyền các bộ tham số mật mã cho các kết nối từ bộ vi xử lý xuống cho FPGA, nhóm tác giả thiết kế theo sơ đồ Hình 3.



Hình 3. Giao tiếp giữa bộ vi xử lý và FPGA.

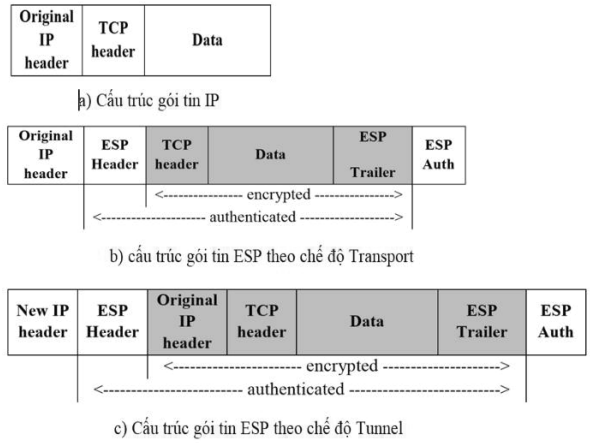
Trong mô hình trên, bộ vi xử lý sẽ thực hiện giao tiếp với FPGA thông qua các bộ nhớ FIFO, làm nhiệm vụ quản lý và trao đổi khóa theo giao thức IKE. Khối TSE [1] sẽ đọc ghi gói tin từ cổng mạng đưa vào thiết bị để sẵn sàng xử lý. Dữ liệu sau khối TSE được đưa vào khối SELECT_IP, khối này có nhiệm vụ phân tích gói tin nhằm xác định các gói tin là ESP hay IKE. Các gói IKE sẽ được đưa đến khối BUFF_IN trước khi gửi đến FIFO_IN để bộ vi xử lý đọc và xử lý gói IKE phục vụ việc trao đổi khóa. Bộ vi xử lý NIOS thực hiện xử lý trao đổi khóa IKE, gửi gói tin IKE xuống FPGA qua FIFO_OUT. Kết thúc quá trình trao đổi khóa, bộ vi xử lý gửi khóa ESP xuống FPGA qua FIFO_OUT. Khối SELECT_PARA đọc dữ liệu do NIOS gửi xuống từ FIFO_OUT, phân tích gói tin nhằm xác định gói tin IKE và gói tin chứa khóa ESP. Nếu là gói tin IKE sẽ được gửi ra BUFF_OUT để chuyển sang khối TSE đóng gói và gửi đi. Nếu là gói tin chứa khóa ESP sẽ được gửi sang khối ESP.

B. Mô hình triển khai ESP trên FPGA

1. Cấu trúc gói tin ESP

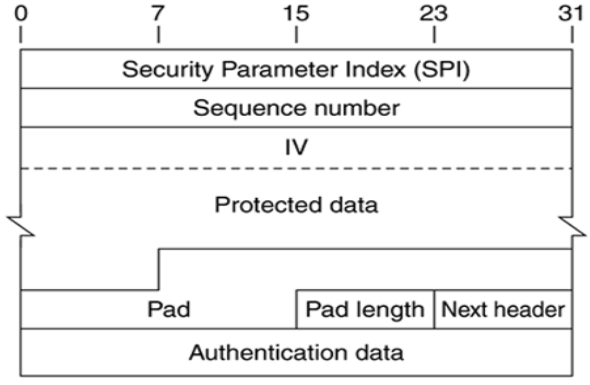
Giao thức ESP thực hiện mã hóa dữ liệu, ký dữ liệu của gói tin, khả năng chống lại kiểu tấn công phát lại (replay) và các dịch vụ bảo đảm cho sự toàn vẹn của các gói tin trong giao thức IP. IPSec được triển khai theo giao thức ESP

theo chế độ Transport (Hình 4b) hoặc Tunnel (Hình 4c).



Hình 4. Cấu trúc gói tin ESP.

Dựa theo mô hình triển khai trên Hình 1, nhóm tác giả tập trung vào triển khai giao thức mã hóa gói tin ESP với cấu trúc gói tin như Hình 4c. ESP trong chế độ Tunnel thì IP header được mã hóa, giấu được nguồn phát tin và đích nhận tin. Do vậy, giao thức ESP cung cấp đầy đủ các dịch vụ bảo mật đó là bí mật, toàn vẹn và chống được tấn công phát lại. Các thành phần của gói tin ESP được thể hiện như Hình 5.



Hình 5. Các thành phần gói tin ESP.

Để đóng gói tin theo giao thức ESP như trên Hình 5, mỗi kết nối cần phải có bộ tham số riêng như chỉ số bảo mật (SPI), số thứ tự (Sequence number), vector khởi tạo (IV) và dữ liệu xác thực. Trong bài báo này, mỗi kết nối sẽ phải có 2 khóa cho quá trình mã hóa/giải mã và 2 khóa cho quá trình xác thực dữ liệu gói tin.

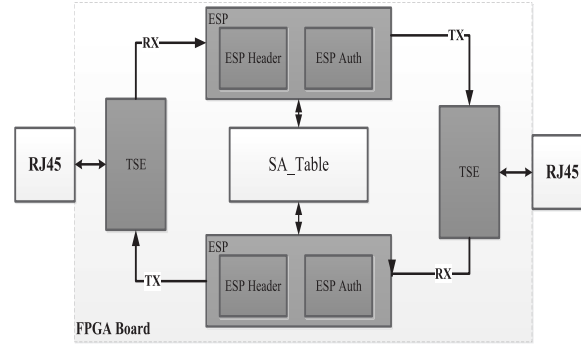
Như vậy, bộ tham số cho mỗi kết nối sau khi thực hiện trao đổi khóa IKE xong sẽ được bộ vi xử lý đẩy xuống FPGA sử dụng thuật toán mã khối GOST và HMAC_SHA_256 như Bảng 1.

BẢNG 1. CÁC THAM SỐ CHO ESP

stt	Ý nghĩa	Kích thước, Byte
1	Địa chỉ IP public đầu xa	4
2	Địa chỉ IP mạng trong đầu xa	4
3	SPI cho ESP mã hóa	4
4	SPI cho ESP giải mã	4
5	SeqNumber gói tin đi	4
6	SeqNumber gói tin đến	4
7	Khóa mã hóa	32
8	Khóa giải mã	32
9	Khóa HMAC gói tin đi	32
10	Khóa HMAC gói tin đến	32
11	Dữ liệu IV mầm	8

2. Mô hình triển khai ESP trên FPGA

Dựa trên cấu trúc gói tin dữ liệu IPSec đóng gói theo giao thức ESP như Phần 1, nhóm tác giả đề xuất mô hình thực hiện đóng gói và xác thực tính toàn vẹn dữ liệu gói tin theo giao thức ESP trên FPGA như sơ đồ Hình 6.



Hình 6. Mô hình thực hiện trên FPGA.

Như vậy, trên bo mạch FPGA sẽ có 2 luồng dữ liệu chính. Luồng thứ nhất sẽ thực hiện đóng gói tin IP theo giao thức ESP và thực hiện mã hóa dữ liệu, luồng còn lại sẽ thực hiện loại bỏ giao thức ESP và thực hiện giải mã. Mô hình trên được xây dựng không áp dụng cho các gói tin điều khiển như ARP.

Nguyên lý hoạt động của sơ đồ được xây dựng như sau: Sau khi thực hiện trao đổi khóa IKE xong, trên bộ vi xử lý sẽ thực hiện đẩy các tham số mật mã xuống cho FPGA và được lưu trong khối SA_Table. Khi các gói tin cần được mã hóa hoặc giải mã, các tham số của gói tin sẽ được ánh xạ vào trong các tham số được lưu

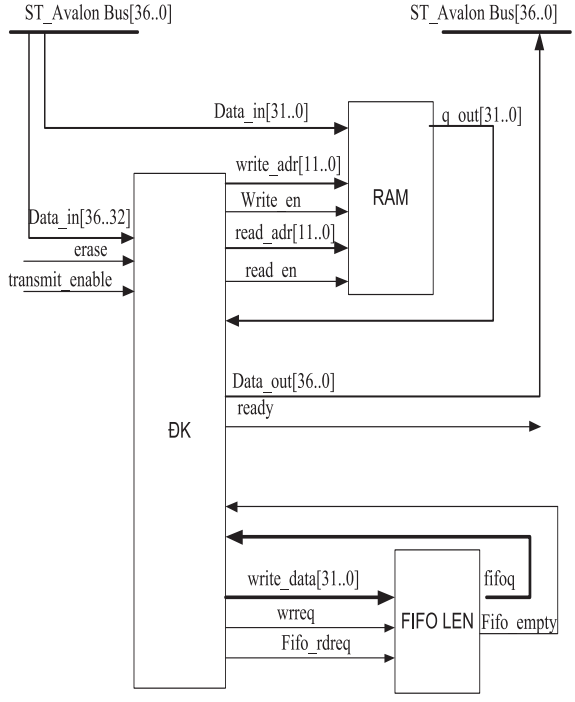
trong khối SA_Table. Nếu các tham số này thỏa mãn thì sẽ được xử lý bởi khối ESP, ngược lại sẽ bị loại bỏ.

III. GIẢI PHÁP THỰC HIỆN TRÊN FPGA

Trong bài báo này, nhóm tác giả sử dụng thuật toán mã khối GOST R34.12-2015 để mã hóa/giải mã dữ liệu và HMAC_SHA_256_128 bit để xác thực tính toàn vẹn của gói tin.

A. Giải pháp thiết kế thực hiện can thiệp gói tin trên FPGA

Để thực hiện xử lý các gói tin trên FPGA, nhóm tác giả thiết kế 2 bộ nhớ trên FPGA là bộ nhớ RAM và bộ nhớ FIFO, được thiết kế như sơ đồ Hình 7.



Hình 7. Giải pháp xử lý gói tin trên FPGA.

Với thiết kế như trên, dữ liệu các gói tin sẽ được ghi vào trong bộ nhớ RAM, tính từ khi tín hiệu bắt đầu gói tin cho đến khi tín hiệu kết thúc gói tin bật lên mức cao (1'b1). Độ dài của các gói tin này (tính theo word 32 bit) sẽ được ghi vào trong bộ nhớ FIFO.

B. Tổ chức dữ liệu cho khối SA_Table

1. Giải pháp tổ chức khung dữ liệu SA

Sau khi thực hiện trao đổi khóa IKE xong, các tham số mật mã phục vụ cho quá trình đóng gói theo ESP và xác thực tính toàn vẹn dữ liệu của gói tin sẽ được bộ vi xử lý gửi xuống cho FPGA thông qua bộ nhớ FIFO được thiết kế như sơ đồ Hình 3.

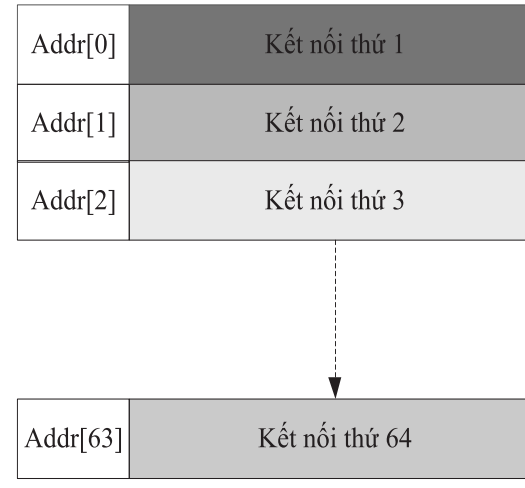
Để cho quá trình đóng gói ESP và xác thực dữ liệu đáp ứng được đường truyền tốc độ cao Gigabit, đòi hỏi khối SA_Table phải xử lý và phân phối các tham số một cách nhanh chóng và hiệu quả.

Trong mô hình triển khai thiết bị IPSec, mỗi kết nối sau khi thực hiện trao đổi khóa IKE xong tổng tất cả các tham số là 160 Byte (1280 bit).

Trong mô hình triển khai thực tế, sẽ có rất nhiều kết nối được thiết lập, trong phạm vi bài báo này, nhóm tác giả xây dựng giải pháp quản lý cho 64 kết nối đồng thời. Để thực hiện giải pháp này trên FPGA, theo nhóm tác giả, có rất nhiều giải pháp khác nhau để lưu trữ và phân phối các tham số mật mã cho các kết nối IPSec như dùng bộ nhớ RAM, FIFO, hay là các thanh ghi được thiết kế trên FPGA. Nhóm tác giả đề xuất giải pháp sử dụng thanh ghi thiết kế theo mô hình Vector. Các tham số của mỗi kết nối sẽ được lưu trữ trên một mảng của Vector đó. Cấu trúc khai báo như sau:

Reg [1279:0] SA_Table [0:63];

Tham số của các kết nối nhận được sau khi thực hiện trao đổi khóa IKE xong sẽ được truyền xuống cho FPGA để thực hiện lưu và phân phối cho từng kết nối. Các tham số này sẽ được lưu lần lượt vào trong mảng lần lượt từ địa chỉ 0 cho tới địa chỉ 63 như sơ đồ Hình 8.



Hình 8. Mô hình lưu trữ tham số cho 64 kết nối.

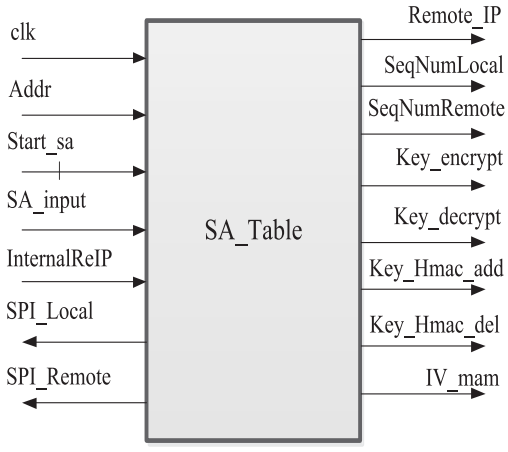
Các tham số của từng kết nối được tổ chức như Bảng 2.

BẢNG 2. TỔ CHỨC THAM SỐ CHO TỪNG KẾT NỐI

STT	Ý nghĩa	Thứ tự bit
1	Địa chỉ IP public đầu xa	1279-1248
2	Địa chỉ IP mạng trong đầu xa	1247-1216
3	SPI cho ESP mã hóa	1215-1184
4	SPI cho ESP giải mã	1183-1152
5	SeqNumber cho mã hóa	1151-1120
6	SeqNumber cho giải mã	1119-1088
7	Khóa mã hóa	1087-832
8	Khóa giải mã	831-576
9	Khóa HMAC chiều mã hóa	575-320
10	Khóa HMAC chiều giải mã	319-64
11	Dữ liệu IV mầm	63-0

2. Giải pháp thiết kế khối SA_Table

Cấu trúc thiết kế khối SA_Table được thể hiện như Hình 9.



Hình 9. Sơ đồ cấu trúc khối SA_Table.

Giải pháp ghi tham số SA trên FPGA

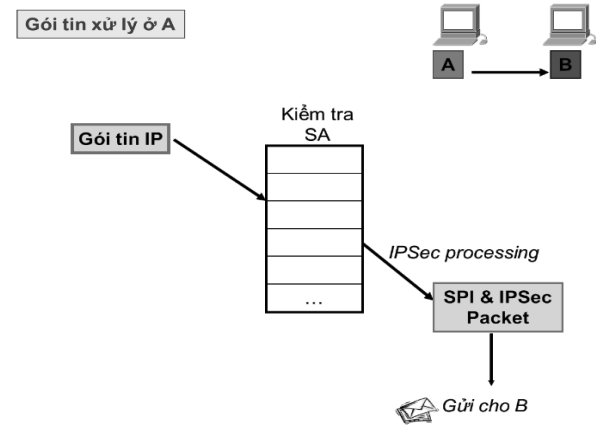
Khi các tham số mật mã của mỗi kết nối được truyền từ bộ vi xử lý xuống cho FPGA thông qua bộ nhớ FIFO_Out, khối SELECT_PARA sẽ xử lý và gửi các tham số *SA_input[1279:0]* kèm theo tín hiệu điều khiển *addr[6:0]* và *start_sa* tới khối SA_Table để thực hiện lưu trên FPGA. Trong đó *addr[6:0]* là địa chỉ cho từng kết nối. Để tránh sự trùng lặp các kết nối việc quản lý các địa chỉ này sẽ được thực hiện bởi bộ vi xử lý, mỗi một kết nối sẽ được lưu tại một địa chỉ cụ thể.

```
Always @(posedge clk)
Begin
 If (start_sa)
  Begin
   If (addr[6:0] = 7'di)
     SA_Table[i] <= SA_input;
  end
end
```

Khi tín hiệu *start_sa* bật lên ở mức cao (1'b1), khối SA_Table sẽ thực hiện kiểm tra địa chỉ được gửi kèm theo bộ tham số, sau đó thực hiện cập nhật vào bảng được thiết kế như trong Hình 8.

Giải pháp lựa chọn tham số SA giao thức ESP

Đối với quá trình mã hóa, gói tin IP trước khi được đóng gói theo giao thức ESP chế độ Tunnel sẽ được kiểm tra trước. Nếu địa chỉ IP đầu xa đã có trong bảng SA, thì sẽ thực hiện đóng gói, xác thực theo cấu trúc ESP và thực hiện mã hóa dữ liệu. Quá trình thực hiện được thể hiện như sơ đồ Hình 10.



Hình 10. Sơ đồ kiểm tra SA và đóng gói tin ESP.

Khi địa chỉ IP đầu xa kiểm tra có trong dữ liệu SA của SA_Table, khối SA_Table sẽ truyền các tham số mật mã của kết nối cho quá trình đóng gói ESP, xác thực và mã hóa dữ liệu.

```
Always @(posedge clk)
Begin
 if (IntReIP == SA_Table[i][1247:1216])
  begin
  RemotIP <= SA_Table[i][1279:1248];
  SPILocal <= SA_Table[i][1215:1184];
  SeqNum <= SA_Table[i][1151:1120];
  KeyEncryp <= SA_Table[i][1087:832];
  KeyHmac <= SA_Table[i][575:320];
  IV_Mam <= SA_Table[i][63:0];
 end
end
```

Tương tự như quá trình mã hóa, gói tin đã được mã hóa và đóng gói theo giao thức ESP sẽ thực hiện kiểm tra dữ liệu SA trước khi thực hiện giải mã.

Khác với quá trình mã hóa, khi gói tin ESP được truyền đến, khối SA_Table sẽ kiểm tra địa chỉ IP Public đầu xa và tham số SPI ở trong gói tin có trong dữ liệu của SA_Table hay không, nếu có khối SA_Table sẽ truyền các tham số mật mã cho quá trình xác thực, ESP và giải mã dữ liệu. Nếu các tham số này không có trong dữ liệu SA, gói tin này sẽ bị loại bỏ và không thực hiện giải mã.

```
Always @(posedge clk)
Begin
 if (RemotIP == SA_Table[i][1279:1248])
& (SPI == SA_Table[i][1183:1152])
 begin
  SeqNum <= SA_Table[i][1119:1088];
  KeyDecrypt <= SA_Table[i][831:576];
  KeyHmac <= SA_Table[i][319:64];
 end
end
```

C. *Giải pháp thực hiện mã hóa/giải mã trên FPGA*

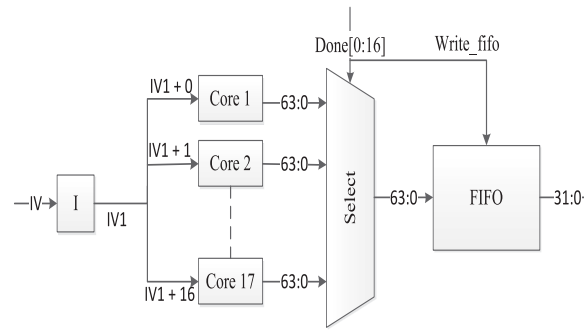
Để thực hiện mã hóa/giải mã đáp ứng được đường truyền ethernet tốc độ cao, nhóm tác giả sử dụng thuật toán mã khối theo chế độ CTR. Trong phạm vi bài viết này, nhóm tác giả sử dụng thuật toán GOST R34.12-2015.

1. Giải pháp thiết kế lõi GOST theo chế độ CTR

Đối với thuật toán GOST R34.12-2015, để thực hiện tạo ra 64 bit bản mã từ 64 bit dữ liệu rõ, sẽ mất 32 clock trên FPGA.

Đầu vào:	**Đầu ra:**
+ 64 bit bản rõ	+ 64 bit bản mã
+ Khóa 256 bit	+ số clock: 32

Để thuận tiện cho việc ghép nối khối mã hóa vào mô hình đường truyền ethernet 1Gbps, nhóm tác giả đề xuất mô hình thiết kế như Hình 11.



Hình 11. Cấu trúc thiết kế GOST theo CTR.

Khi tín hiệu *start* của khối mã hóa bật lên, các core GOST bên trong sẽ thực hiện lần lượt theo thứ tự từ 1 cho đến 17.

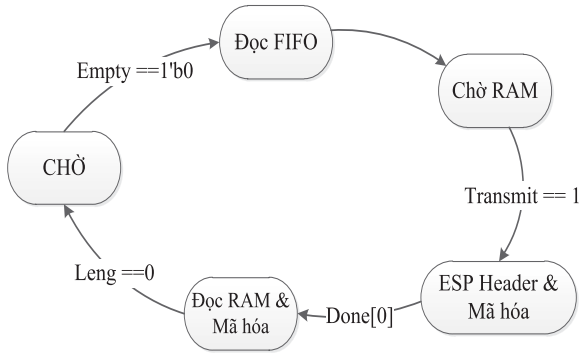
Khi tín hiệu *done* của của core nào bật lên, thì bản khóa dòng của tín hiệu đó sẽ được ghi vào bộ nhớ FIFO được điều khiển bởi khối Select. Tín hiệu ghi vào FIFO (*write_fifo*) là phép *OR* các tín hiệu *done[0:16]* của các core GOST.

Sử dụng FIFO bất đồng bộ với đầu vào là 64 bit và đầu ra là 32 bit. Khi tín hiệu kết thúc khối mã hóa *stop* bật lên mức cao, toàn bộ dữ liệu trong FIFO sẽ được xóa hết để bắt đầu quá trình mã cho gói tiếp theo.

Bộ biến đổi I, thực hiện biến đổi dữ liệu IV (đầu vào của khối mã hóa, và dữ liệu này được đóng vào gói tin). Khi có tín hiệu *start* bật lên ở mức cao, dữ liệu IV này sẽ được gán vào IV1. IV1 là một thanh ghi 64 bit, khi tín hiệu *start* của core 17 bật lên, IV1 sẽ được cộng lên 17.

2. Giải pháp ghép nối khối mã hóa/giải mã

Quá trình mã hóa và giải mã nhóm tác giả thiết kế thực hiện đồng thời trong quá trình đóng gói ESP header, như trên sơ đồ Hình 6. Đối với quá trình mã hóa, các gói tin sau khi kiểm tra thỏa mãn các tham số trong bảng SA_Table. Đối với quá trình giải mã, ngoài kiểm tra các tham số đi kèm gói tin IP đến thỏa mãn các tham số trong bảng SA_Table, thì gói tin IP phải được kiểm tra tính toàn vẹn của dữ liệu ở khối ESP Auth trước. Nếu thỏa mãn thì gói tin mới được giải mã, nếu không sẽ bị loại bỏ. Giải pháp ghép nối khối mã hóa được thực hiện theo sơ đồ máy trạng thái như trên Hình 12.



Hình 12. Sơ đồ thực hiện ghép nối mã hóa.

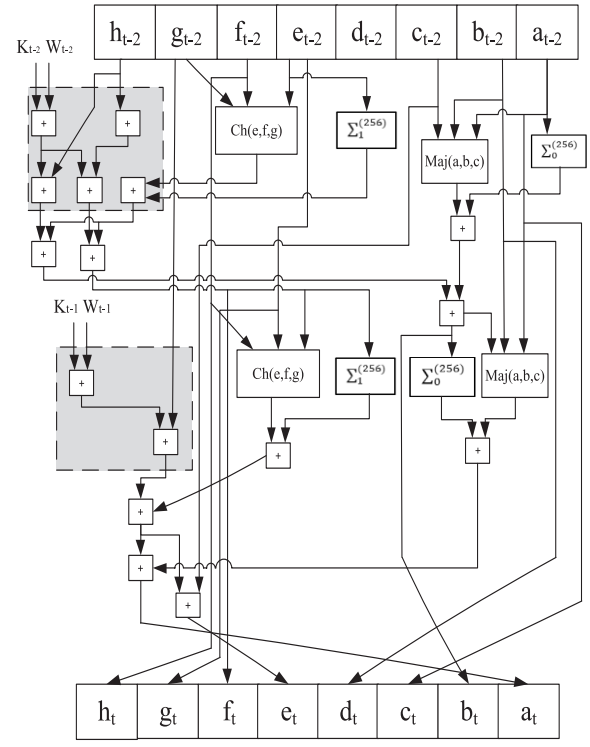
Nguyên lý thực hiện ghép nối như sau: khi toàn bộ gói tin được lưu hết vào bộ nhớ RAM, đồng nghĩa độ dài gói tin cũng được lưu vào FIFO (Hình 7), tín hiệu *empty* của FIFO bật về mức thấp, lúc này máy trạng thái chuyển sang trạng thái *Đọc FIFO*. Độ dài gói tin IP sẽ được đọc từ trong FIFO ra, và máy trạng thái chuyển sang trạng thái *Chờ RAM*. Khi có tín hiệu cho phép đọc (*transmit == 1'b1*), máy trạng thái chuyển sang trạng thái *ESP Header & Mã hóa*. Ở trạng thái này sẽ thực hiện đóng gói ESP Header cho gói tin IP theo chiều mã hóa và loại bỏ ESP Header cho gói tin IP theo chiều giải mã theo các tham số trong bảng SA_Table và thực hiện tạo các bản khóa dòng bởi thuật toán GOST theo chế độ CTR. Khi tín hiệu *done[0]* bật lên, đồng nghĩa đã có dữ liệu khóa dòng từ thuật toán GOST, máy trạng thái chuyển sang trạng thái *Đọc RAM & Mã hóa*. Ở trạng thái này, dữ liệu gói tin từ RAM sẽ được đọc ra và thực hiện XOR với các dòng khóa được tạo ra từ khối mã GOST, ở trạng thái này lõi mã hóa vẫn tiếp tục tạo ra các dòng khóa từ dữ liệu IV. Khi dữ liệu của gói tin trong RAM được đọc hết (*leng == 0*), lúc này máy trạng thái chuyển sang trạng thái *CHỜ* ban đầu và khối mã hóa cũng dừng hoạt động.

D. Giải pháp xây dựng khối xác thực tính toàn vẹn của gói tin IP

1. Giải pháp tối ưu cho SHA_256

Đối với thuật toán hàm băm SHA_256, nếu thiết kế theo tiêu chuẩn cho mỗi khối thông điệp 512 bit trên FPGA sẽ cần tới 64 vòng tương đương với 64 clock để thực hiện. Khi đó, với mỗi gói tin IP có kích thước lớn (1500 Byte), để thực hiện băm cho một gói tin sẽ cần tới 64*((1500*8)/512) clock, xấp xỉ 1500 clock. Với thiết kế như vậy, tốc độ đường truyền mạng của hệ thống sẽ bị giảm xuống.

Để tối ưu giải pháp thiết kế cho thuật toán hàm băm SHA_256, nhóm tác giả thực hiện mô hình thiết kế như trên Hình 13 [6].

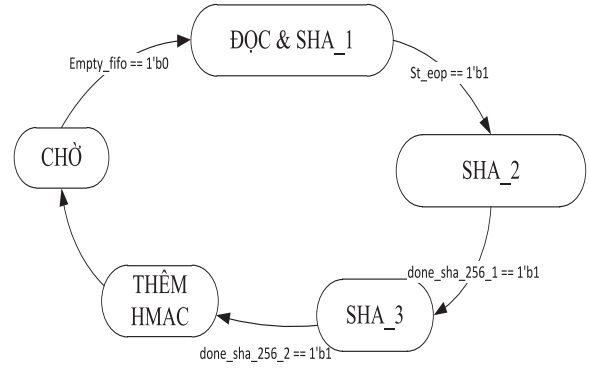


Hình 13. Giải pháp tối ưu SHA_256.

Với thiết kế như trên, với mỗi một thông điệp đầu vào 512 bit, sẽ chỉ mất 32 clock để thực hiện. Đối với một gói tin có kích thước lớn, giải pháp trên mang lại hiệu quả rất lớn trong việc tăng tốc độ đường truyền.

2. Giải pháp ghép nối HMAC_SHA_256

Nhóm tác giả đề xuất sơ đồ thiết kế máy trạng thái thực hiện HMAC_SHA_256 [4,5] trên FPGA như trên sơ đồ Hình 13.

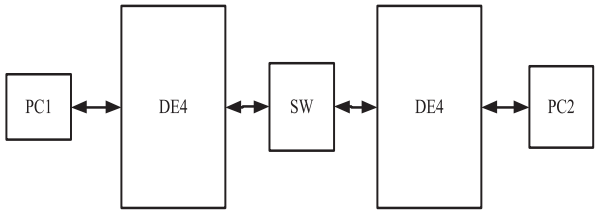


Hình 14. Máy trạng thái thực hiện HMAC.

Khi gói tin IP được lưu vào bộ nhớ RAM, đồng nghĩa với độ dài của gói tin cũng được lưu vào bộ nhớ FIFO như sơ đồ Hình 7. Lúc này tín hiệu *empty* của FIFO bật về mức thấp (1'b0), máy trạng thái chuyển từ trạng thái *CHỜ* sang trạng thái *ĐỌC & SHA_1*. Ở trạng thái này, quá trình đọc dữ liệu của gói tin trong RAM sẽ được đọc ra, đồng thời 2 hàm băm SHA_256 sẽ thực hiện song song. SHA_256 thứ nhất sẽ băm khóa K1 = K⊕IPAD và dữ liệu gói tin (*SHA-256(K1||D)*). SHA_256 thứ 2 sẽ thực hiện băm K2 = K⊕OPAD (*SHA-256(K2*)). Khi dữ liệu của gói tin đọc ra từ RAM xong, tức tín hiệu *st_eop* bật lên ở mức cao (1'b1), máy trạng thái chuyển sang *SHA_2*. Ở trạng thái này, hàm băm SHA_256 thứ nhất tiếp tục thực hiện băm cho tới khi hết dữ liệu của gói tin và tạo ra giá trị băm H1. Khi đó tín hiệu *done_sha_256_1* bật lên mức cao, máy trạng thái lúc này sẽ chuyển sang trạng thái *SHA_3*. Ở trạng thái này, hàm băm SHA_256 thứ 2 sẽ thực hiện băm SHA_256(H1). Khi quá trình này thực hiện xong, đồng nghĩa với HMAC_SHA_256 cũng đã băm xong dữ liệu (*done_sha_256_2 == 1'b1*), máy trạng thái chuyển sang trạng thái *THÊM HMAC*. Ở trạng thái này sẽ thêm dữ liệu HMAC vừa thực hiện xong vào cuối của gói tin IP và truyền đi.

IV. KẾT QUẢ THỰC HIỆN

Để kiểm tra và đánh giá kết quả thực hiện giải pháp của nhóm tác giả, chúng tôi sử dụng 2 bo mạch DE4 của hãng Intel được ghép nối như trên sơ đồ Hình 15. Trong đó 2 bo mạch DE4 thực hiện đóng gói ESP, mã hóa/giải mã và xác thực tính toàn vẹn của dữ liệu.



Hình 15. Mô hình thử nghiệm với 2 bo mạch.

Tài nguyên thiết kế trên FPGA cho các khối được tổng hợp trên chip DE4 của Intel, được thể hiện trên Bảng 3. Đối với các dòng chip có tài nguyên hạn chế, thiết kế này cần thiết phải tối ưu thêm.

BẢNG 3. TÀI NGUYÊN THIẾT KẾ TRÊN FPGA

Khối	ALUTs	ALMs	Block memory bit	Dedicated logic registers
ESP	886	1098	132336	1165
HMAC	16439	16988	299360	7482

Kết quả kiểm tra theo mô hình 2 bo mạch (Hình 15) đối với đường truyền 1 Gbps cho gói tin 1500 Byte sử dụng phần mềm Iperf được thể hiện trên Hình 16. Tốc độ đường truyền phụ thuộc vào kích thước của từng gói tin, để đánh giá thực tế, nhóm tác giả thiết lập MTU trên máy tính là 1500 cho mỗi gói tin qua cổng Ethernet. Kết quả thực thi của nhóm tác giả cho đường truyền 1 Gbps là 710 Mbps ở tần số 125 MHz.

```
------------------------------------------------------------
Client connecting to 192.168.2.101, TCP port 5001
TCP window size: 2.00 MByte
------------------------------------------------------------
[  3] local 192.168.0.99 port 57476 connected with 192.168.2.101
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  96.2 MBytes   807 Mbits/sec
[  3]  1.0- 2.0 sec  97.4 MBytes   817 Mbits/sec
[  3]  2.0- 3.0 sec  88.2 MBytes   740 Mbits/sec
[  3]  3.0- 4.0 sec  73.5 MBytes   617 Mbits/sec
[  3]  0.0- 5.0 sec   422 MBytes   710 Mbits/sec
```

Hình 16. Kiểm tra băng thông bằng Iperf.

Với giải pháp thực thi IPSec trên FPGA, tùy thuộc vào từng dòng chip được sử dụng, hay là các thuật toán mật mã, xác thực được triển khai trên thiết bị. Với giải pháp triển khai IPSec trên FPGA [7] sử dụng thuật toán mã hóa DES và HMAC_SHA_1, thực thi trên kit DE2 của Intel đối với gói tin 1500 Byte ở tần số 86,72 MHz tốc độ đạt 437,44 Mbps cho đường truyền 1Gbps. Khi thực thi trên kit DE5 của Intel tốc độ đường truyền có thể lên tới 10 Gbps đối với đường truyền SFP+, tuy nhiên bài báo này không đề cập đến giải pháp phân phối khóa cho từng kết nối trong mô hình triển khai IPSec.

KẾT LUẬN

Bài báo đã trình bày một giải pháp quản lý kết nối mật cho IPSec trên FPGA sử dụng ngôn ngữ mô tả phần cứng Verilog HDL. Mỗi một kết nối trong hệ thống IPSec sẽ có các bộ tham số bảo mật riêng. Trong bài báo này, nhóm tác giả đã trình bày một giải pháp quản lý bộ tham số ESP cho nhiều kết nối và giải pháp kiểm tra, phân phối khóa cho từng kết nối trong mô hình triển khai IPSec.

Các kết quả thử nghiệm trên bo mạch DE4 cho thấy, giải pháp quản lý kết nối và phân phối tham số cho từng kết nối của nhóm tác giả hoạt động ổn định, đáp ứng các yêu cầu đặt ra. Do tài nguyên trên FPGA của bo mạch DE4 là tương đối lớn, do đó thiết kế của nhóm tác giả còn chưa được tối ưu hết tài nguyên trên FPGA. Trong thời gian tới, nhóm sẽ tiếp tục tối ưu tài nguyên thiết kế để đáp ứng cho các loại Chip có tài nguyên nhỏ hơn.

TÀI LIỆU THAM KHẢO

[1] Altera Corp., *"Triple-Speed Ethernet MegaCore Function User Guide"*. https://www.altera.com/literature/ug/ug_ethernet.pdf. (Truy cập 17/3/2021).

[2] RFC 4303, "IP Encapsulating Security Payload (ESP)".10/2005.

[3] RFC 7296, "Internet Key Exchange Protocol Version 2 (IKEv2)". 10/2014.

[4] FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code (HMAC)". 07/2008.

[5] RFC 4634 "US Secure Hash Algorithms (SHA and HMAC-SHA)". 7/2006.

[6] H.E.Michail, A.P.Kakarountas, E.Fotopoulou, C.E.Goutis, "*High-Speed and Low-Power Implementation of Hash Message Authentication Code through Partially Unrolled Techniques*", Proceedings of the 5th WSEAS Int. Conf. on multimedia, internet and video technologies, Corfu, Greece, 17-19/8/2005, pp. 130-135.

[7] Mateusz Korona, Krzysztof Skowron, Mateusz Trzepinski, Mariusz Rawski, "*High-performance FPGA Architecture for Data Streams Processing on Example of IPsec Gateway*", Intl journal of electronics and telecommunications, 2018, Vol. 64, No. 3, pp. 351-356.

[8] Muzaffar Rao, Joseph Coleman and Thomas Newe "*An FPGA based reconfigurable IPSec ESP core suitable for IoT applications*" Conference: 2016 10th International Conference on Sensing Technology, 11-13/11/2016.

[9] Helion Technology Limited, IPsec ESP IP Core for FPGA – Product Brief, http://www.heliontech.com/ipsec.htm. (Truy cập 17/3/2021).

[10] Sangjin Han, Keon Jang, Kyoung Soo Park, Sue Moon, PacketShader, *"A GPU-accelerated Software Router"*, http://shader.kaist.edu/packetshader, 2010 (Truy cập 17/3/2021).

[11] Ky Phan Van, Thang Tran Van, Phuc La Huu, "A solution for packet security 1 Gbps on layer 2 with technology FPGA", Journal of Science and Technology on Information security, ISSN 2615-9570, Vol. 08, No.02, 2018, pp. 19-24.

SƠ LƯỢC VỀ TÁC GIẢ

Phan Văn Kỷ

Đơn vị công tác: Viện Khoa học – Công nghệ mật mã, Ban Cơ yếu Chính phủ.

Email: pvk.hvktqs@gmail.com

Quá trình đào tạo: Nhận bằng Đại học năm 2013 và Thạc sĩ năm 2017 chuyên ngành Vô tuyến kỹ thuật, Đại học tổng hợp kỹ thuật Điện St. Peterburg, Liên Bang Nga.

Hướng nghiên cứu hiện nay: Công nghệ vi mạch, FPGA.

La Hữu Phúc

Đơn vị công tác: Viện Khoa học – Công nghệ mật mã, Ban Cơ yếu Chính phủ.

Email: phucpvkt@hotmail.com

Quá trình đào tạo: Nhận bằng Đại học năm 1998 và Thạc sĩ năm 2002 chuyên ngành Kỹ thuật điện tử, Học viện Kỹ thuật mật mã. Nhận bằng Tiến sĩ năm 2015, Viện Khoa học - Công nghệ quân sự.

Hướng nghiên cứu hiện nay: Thiết kế chế tạo máy mã, thiết bị mã chuyên dụng.