GIẢI PHÁP PHÁT TRIỂN HỆ XÁC THỰC PHẦN CỨNG HMAC-SHA-512 ỨNG DỤNG TRONG AN TOÀN THÔNG TIN MẠNG

ThS. Nguyễn Trường Sơn - Trung tâm KHKT & CNQS
TS. Keisuke Iwai, PGS. TS. Takakazu Kurokawa
Học viện phòng vệ Nhật Bản

Tóm tắt

Trong những năm gần đây, xác thực thông tin (Message Authentication) có vai trò vô cùng quan trọng trong an toàn thông tin mạng máy tính. Bài báo trình bày một phương án phát triển hệ xác thực phần cứng HMAC-SHA-512 mới, ứng dụng trong an toàn thông tin mạng trên vi mạch FPGA (Field Programmable Gate Array). Tác giả đề xuất cấu trúc phần cứng cho phép đạt được tốc độ xử lý dữ liệu cao trên cơ sở sử dụng các bộ đếm song song (Parallel Counters - PCs) và các bộ cộng cất nhớ (Carry Save Adders - CSAs). Kết quả thực nghiệm cho thấy tốc độ xử lý dữ liệu của hệ thống dựa trên thiết kế của tác giả đạt 1.267 Gbps, nhanh gấp 3.8 lần so với hệ xác thực phần mềm HMAC-SHA-512 mạnh nhất hiện nay. Điều này chứng tỏ thiết kế của tác giả cho phép cung cấp một hệ xác thực phần cứng có khả năng ứng dụng cao trong an toàn thông tin mạng.

1. Giới thiệu

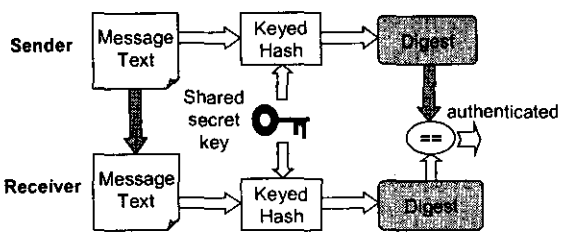
Trong những năm gần đây, với sự phát triển nhanh chóng các mạng máy tính trên qui mô toàn thế giới, an toàn thông tin ngày càng trở nên quan trọng. Các thuật toán an toàn thông tin như xác thực (Authentication) và mã hóa (Encryption) được cung cấp bởi các giao thức bảo mật (Security Protocols) cho phép tăng cường độ tin cậy cho mạng máy tính. Trong đó, ba giao thức bảo mật cơ bản được biết đến rộng rãi bao gồm: IPSec (Internet Protocol Security), SSL (Secure Sockets Layer), TSL (Transport Layer Security) [1]. Một trong những thuật toán quan trọng nhất được sử dụng bởi các giao thức này là xác thực thông tin. Thuật toán xác thực thông tin cung cấp khả năng cho phép xác nhận rằng, các thông tin thu nhận được, thực sự đến từ nguồn tin tin cậy và không bị làm giả trên đường truyền. Xác thực thông tin thường được xây dựng trên cơ sở kỹ thuật hàm Hash bảo mật. Trong đó, phương thức xác thực thông tin căn bản là phương thức sử dụng mã xác thực thông tin (Message Authentication Code - MAC) [1]. Kỹ thuật xây dựng MAC trên cơ sở sử dụng hàm Hash được gọi là HMAC [2]. HMAC-MD5, HMAC-SHA-1 cũng như HMAC-SHA-512 là những thuật toán xác thực cơ bản được sử dụng trong IPSec, cho phép các máy chủ hoặc mạng cục bộ truyền nhận dữ liệu đã được mã hóa và xác thực thông qua các mạng không tin cậy [1].

Hiện nay có nhiều giải pháp thực hiện HMAC-MD5 và HMAC-SHA-1 mang tính học huật và thương mại đã được đề xuất [3]. Tuy nhiên, các giải pháp này bộc lộ một số vấn đề như hạn chế về tốc độ xử lý dữ liệu và mức độ bảo mật, đồng thời chi phí phát triển hệ thống cao, gây khó khăn cho khả năng ứng dụng rộng rãi HMAC trong các lĩnh vực kỹ thuật.

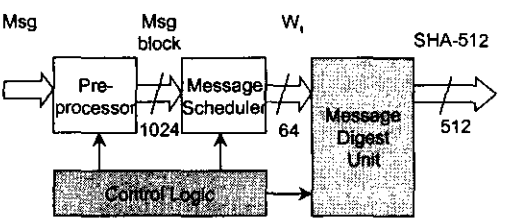
Trong bài báo này, tác giả đề xuất giải pháp phát triển một hệ xác thực mới HMAC-SHA-512 nhằm khắc phục những vấn đề nêu trên. HMAC-SHA-512 được phát triển trên vi mạch FPGA Xilinx Virtex-II XC2V6000 và được thử nghiệm trên Board phát triển KAC-02 của hãng Mitsubishi.

2. Thuật toán HMAC-SHA-512

Chúng ta biết rằng, HMAC cung cấp phương pháp xác thực và kiểm tra tính toàn vẹn của thông tin truyền nhận hoặc lưu trữ trong những môi trường không tin cậy dựa trên khóa mật. Mục đích của HMAC là cho phép xác thực thông tin mà không cần thêm bất kỳ một kỹ thuật hỗ trợ nào. Phương thức xác thực HMAC được biểu diễn ở hình 1.



Hình 1. Phương thức xác thực HMAC



Hình 2. Sơ đồ khối SHA-512

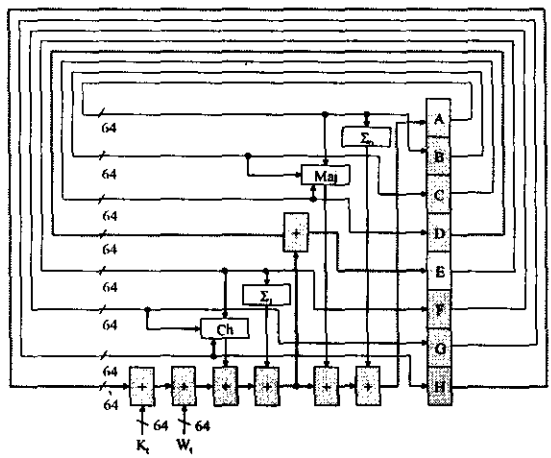
HMAC giả định rằng có hai đối tác truyền thông *Người gửi* (Sender) và *Người nhận* (Receiver) chia sẻ cùng một khóa mật. *Người gửi* sử dụng hàm Hash và khóa mật để tạo ra bản tóm lược (Digest) từ thông điệp ban đầu (Message Text). Sau đó cả thông điệp và Digest đều được gửi tới *Người nhận*. *Người nhận* sẽ sử dụng cùng một hàm Hash và khóa mật giống với *Người gửi* để tạo ra Digest từ thông điệp nhận được. Kết quả thu được đem so sánh với Digest nhận được. Tùy thuộc vào kết quả so sánh, thông điệp sẽ được chấp nhận hay từ chối.

HMAC-SHA-512 là thuật toán xác thực HMAC sử dụng hàm Hash SHA-512, có hai tham số đầu vào là khóa mật K và thông điệp M, cho kết quả đầu ra là giá trị HMAC có độ dài lớn nhất là 512-bit. Hai hằng số *ipad* và *opad* được cung cấp để bảo vệ khóa mật chống lại khả năng bị phát hiện hoặc thay thế. HMAC được tính toán theo công thức sau đây:

$$HMAC(K,M) = H((K \oplus opad)||H((K \oplus ipad)||M)) \quad (1)$$

Ở đây ký hiệu $\oplus$ chỉ phép toán XOR và || chỉ sự kết nối. Hàm SHA-512 (ký hiệu là H trong (1)) nhận đầu vào là một bản tin Msg có độ dài cực đại nhỏ hơn 2^{128} và cho kết quả là Digest có độ dài 512-bit [4]. Để tạo ra Digest 512-bit (hình 2), đầu tiên Msg được xử lý bởi Pre-processor, tại đây nó được đệm (padding) thêm các bit "0" và chia thành các Block có độ dài cố định bằng 1024-bit, gồm 16 từ (word) 64-bit. Sau đó, Pre-

processor chuyển các Block 1024-bit tới Message Scheduler để chuẩn bị các giá trị tính toán. Khối xử lý trung tâm của SHA-512 là Message Digest Unit (MDU), tại đây thực hiện 80 bước tính toán giá trị Digest. Hình 3 biểu diễn sơ đồ khối một bước tính toán của MDU.



Hình 3. Sơ đồ khối một bước tính toán của MDU

Trong đó, tám từ 64-bit A ÷ H được sử dụng để tính toán giá trị Digest. Tại mỗi bước, một từ W_t được tạo ra từ Message Scheduler và một hằng số K_t được sử dụng để hoàn thành phần tính toán. Trong Hình 3, Ch, Maj, Σ_0, Σ_1 là các hàm logic được tính toán theo các công thức sau đây:

$$Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z) \quad (2)$$

$$Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \quad (3)$$

$$\Sigma_0(x) = ROTR^{28}(x) \oplus ROTR^{34}(x) \oplus ROTR^{39}(x) \quad (4)$$

$$\Sigma_1(x) = ROTR^{14}(x) \oplus ROTR^{18}(x) \oplus ROTR^{41}(x) \quad (5)$$

Trong đó, các ký hiệu $\wedge$, $\neg$ và ROTR tương ứng với các phép toán AND, phủ định và quay phải. Tất cả các phép cộng trong hình 3 là những phép tính Modulo 2^{64}.

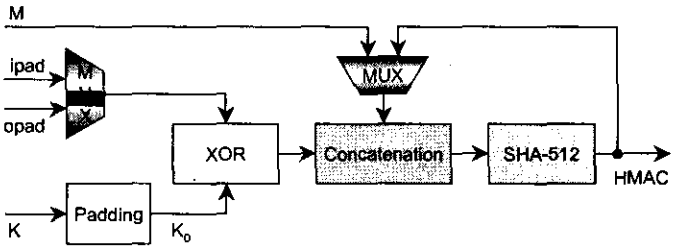
3. Phát triển HMAC-SHA-512 trên FPGA

Ngày nay, các vi mạch điện tử có khả năng tái cấu trúc như FPGA và CPLD (Complex Programmable Logic Device) đang là những lựa chọn hấp dẫn cho giải pháp phát triển phần cứng các thuật toán xác thực-mã hóa, vì chúng cung cấp khả năng xử lý mềm dẻo cho các hệ thống động, cũng như khả năng thực hiện tổng hợp nhiều thuật toán xác thực-mã hóa một cách dễ dàng.

3.1. Thiết kế hệ thống

Sơ đồ khối HMAC-SHA-512 trong thiết kế của chúng tôi được trình bày ở hình 4. Trong đó, M là thông điệp đầu vào, K là khóa mật. K_0 là khóa có độ dài 1024-bit được tạo ra từ K bằng việc đệm thêm các bit "0" khi K có độ dài nhỏ hơn 1024-bit. Đầu ra

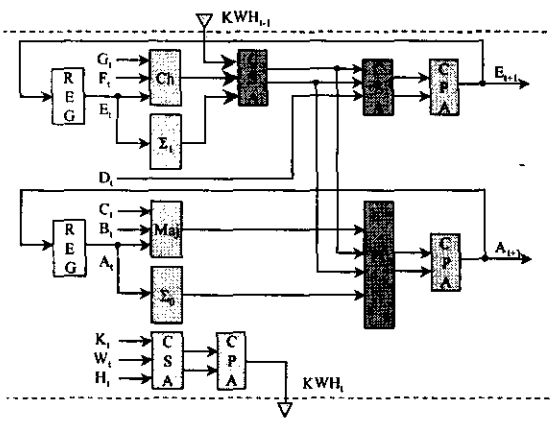
HMAC có độ dài lớn nhất là 512-bit. Như đã thảo luận, phần tính toán quan trọng nhất của thuật toán HMAC-SHA-512 là Message Digest Unit, bao gồm 80 bước tính toán. Để tính toán Digest, 8 thanh ghi 64-bit A ÷ H được sử dụng (hình 3). Đầu tiên các thanh



Hình 4. Sơ đồ khối HMAC-SHA-512

ghi này được khởi tạo bởi các hằng số cho trước, sau đó được cập nhật các giá trị mới tại mỗi bước tính toán. Giá trị của sáu thanh ghi B, C, D, F, G và H đơn giản được chuyển từ A, B, C, E, F và G tới. Tuy nhiên, giá trị của A và E đòi hỏi phép tính cộng đa toán tử và lấy số dư Modulo 2^{64} khá phức tạp. Số toán hạng trong các phép tính A và E tương ứng là 7 và 6. Dễ dàng nhận ra rằng, đường tới hạn (critical path) của thuật toán HMAC-SHA-512 là phần tính toán các giá trị mới của A và E.

Đối với phép tính cộng đa toán tử, tốc độ tính toán sẽ bị hạn chế nếu chỉ sử dụng các bộ cộng thông thường Carry Propagate Adder (CPA), do thời gian để hoàn thành bit nhớ carry tại mỗi CPA lớn. Vì vậy, để nâng cao tốc độ xử lý dữ liệu cho hệ thống, chúng tôi quyết định thay thế CPA bằng các bộ cộng tốc độ cao như Carry Save Adder (CSA) và Parallel Counter (PC). Mặt khác, do $H_t = G_{t-1}$ nên tổng của 3 toán hạng $K_t + W_t + H_t = K_t + W_t + G_{t-1} = KWH_t$ được tách riêng và tính trước một chu kỳ xung nhịp, nhằm giảm bớt lượng tính toán trên đường tới hạn. Hình 5 biểu diễn cấu trúc phần cứng được tác giả đề xuất cho phần tính toán các giá trị A và E.



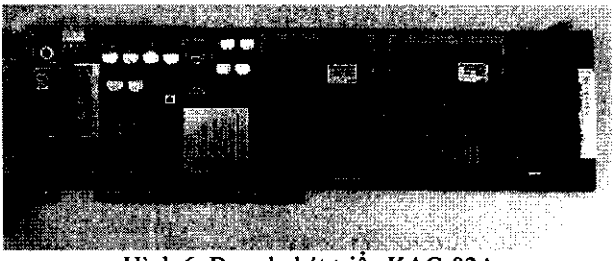
Hình 5. Cấu trúc mạch thiết kế phần tính toán giá trị A và E

Ở đây, CSA là bộ cộng 64-bit được tạo thành từ 64 bộ cộng đầy đủ 1-bit (Full Adder - FA) không liên kết với nhau, có 3 đầu vào và 2 đầu ra (64-bit sum và 64-bit carry).

Không giống các bộ cộng thông thường (như CPA), CSA không chuyển bit nhớ carry trong quá trình cộng giữa các bit. Vì vậy, thời gian thực hiện phép tính của CSA chỉ tương đương với thời gian thực hiện của một FA [5]. PC(4,2) là bộ cộng 64-bit, tiến hành giảm số lượng các digit nhị phân tại mỗi bit trọng số từ 4 đầu vào thành 2 đầu ra. Bộ cộng này có thời gian thực hiện phép tính tương đương với CSA [6]. CPA chỉ cần thiết tại phép tính cuối cùng của biểu thức nhằm đưa ra một kết quả duy nhất.

3.2. Môi trường phát triển

Chúng tôi sử dụng ngôn ngữ Verilog-HDL (Hardware Description Language) cho thiết kế mạch logic, trong môi trường tích hợp công cụ phát triển Xilinx ISE Foundation 5.2i trên Windows XP. Vi mạch FPGA Virtex-II XC2V6000 của hãng Xilinx được lựa chọn để phát triển HMAC-SHA-512. Kết quả thực nghiệm được kiểm chứng theo hoạt động thời gian thực trên board phát triển KAC-02A của hãng Mitsubishi (hình 6).



Hình 6. Board phát triển KAC-02A

KAC-02A có một chip XCV300 dùng để điều khiển giao tiếp PCI và hai chip XC2V6000 cho thiết kế phát triển của người dùng. Board có khả năng giao tiếp với máy tính qua giao diện PCI 64-bit/66 MHz. Dữ liệu cấu hình mạch thiết kế người dùng sẽ được tải xuống chip FPGA (XC2V6000) qua bộ nhớ Flash [7].

4. Kết quả thực nghiệm

Các Module của HMAC-SHA-512 được kiểm chứng theo hoạt động thời gian thực trên cơ sở sử dụng board KAC-02A. Các giá trị Test cung cấp trong [2, 4] cho các thuật toán SHA và HMAC được sử dụng để đảm bảo rằng mạch thiết kế hoạt động một cách chính xác. Bảng 1 đưa ra kết quả so sánh tốc độ xử lý dữ liệu giữa hệ xác thực HMAC-SHA-512 của tác giả và các hệ xác thực đã được phát triển khác.

Hệ xác thực	Giải pháp thực hiện	Tốc độ xử lý dữ liệu (Mbps)
HMAC-SHA-512	Hardware	1267
HMAC-SHA-512 [8]	Software	333.5
HMAC-SHA-1 [9]	Hardware	700

Bảng 1. Kết quả so sánh tốc độ xử lý dữ liệu giữa các hệ xác thực

Kết quả cho thấy tốc độ xử lý dữ liệu của hệ thống dựa trên thiết kế của tác giả đạt 1.267 Gbps, nhanh gấp 3.8 lần so với hệ xác thực phần mềm HMAC-SHA-512 và

nhanh gấp 1.8 lần so với hệ xác thực phần cứng HMAC-SHA-1 mạnh nhất hiện nay. Kết quả này chỉ ra tầm quan trọng của cấu trúc thiết kế trong việc tăng hiệu năng cho các hệ thống phần cứng. Ngoài ra, nó cũng chứng minh lợi ích của việc sử dụng các bộ cộng tốc độ cao như CSA, PC nhằm tăng tốc độ tính toán của các phép tính cộng đa toán tử.

5. Kết luận

Trong bài báo này tác giả đã đề xuất giải pháp phát triển một hệ xác thực mới HMAC-SHA-512 ứng dụng trong an toàn thông tin mạng trên FPGA. Một trong những hàm Hash có mức bảo mật cao nhất SHA-512 đã được lựa chọn để tăng cường độ an toàn cho hệ thống. Ngoài ra, tác giả cũng đã đề xuất cấu trúc phần cứng cho phép nâng cao tốc độ xử lý của hệ xác thực dựa trên các bộ cộng tốc độ cao như CSA và PC, đồng thời đưa ra những kết quả phân tích đánh giá và kiểm chứng hiệu năng hoạt động của hệ HMAC-SHA-512. Kết quả thực nghiệm cho thấy tốc độ xử lý dữ liệu của HMAC-SHA-512 dựa trên thiết kế của tác giả đạt 1.267 Gbps trên vi mạch Xilinx Virtex-II XC2V6000. Điều này chứng tỏ thiết kế của tác giả cho phép cung cấp một hệ xác thực phần cứng có khả năng ứng dụng cao trong an toàn thông tin mạng.

Tài liệu tham khảo

[1].S. KENT and R. ATKINSON, (1998), *Security Architecture for the Internet Protocol,* Request for Comments (RFC) 2401, Internet Activities Board, Internet Privacy Task Force, available at http://www.cis.ohio-state.edu/cgi-in/rfc/rfc2401.html.

[2].FIPS PUB 198, (2002), *The Keyed-Hash Message Authentication Code (HMAC),* available at http://csrc.nist.gov/publications/fips/fips198/fips198a.dpf.

[3].FIPS PUB 140, (2004), *Validated FIPS 140-1 and FIPS 140-2 Cryptographic Modules*, available at http://csrc.ncsl.nist.gov/cryptval/140-1/1401val2004.htm.

[4].FIPS PUB 180-2, (2002), *Secure Hash Standard,* available at http://crypto.nknu.edu.tw/crypto/fips180-2.dpf.

[5].TAEWHAN KIM, WILLIAM JAO, and STEVE TJIANG, (Oct. 1998), *Circuit Optimization Using Carry-Save-Adder Cells*, IEEE Transactions on Computer-Aided Design of Intergrated Circuits and Systems, Vol. **17**, No. 10, pp. 974-984.

[6].GIANLUCA CORNETTA, JORDI CORADELLA, (Sept. 2001), *A synchronous Multipliers with Variable-Delay Counters*, 8th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Vol. **II**, pp. 701-705, Malta.

[7].MITSUBISHI Co., (2002), *KAC-02A Large Scale FPGA Board,* available at http://www.mee.co.jp/pro/sales/fpga/fpga.html.

[8].B. PRENEEL, B. VAN ROMPAY... and M. PARKER, (2003), *Performance of Optimized Implementations of the NESSIE Primitives,* available at http://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/D21-v2.pdf.

[9].ELLIPTIC SEMICONDUCTOR Co., *CLP-06 HMAC Core,* available at http://www.ellipticsemi.com/CLP-06_30814.pdf.