**Bộ tăng tốc phần cứng hiệu quả cho IPSec dựa trên**

**Cấu hình lại một phần trên FPGA Xilinx**

Tác giả:

Ahmad Salman, Marcin Rogawski and Jens-Peter Kaps

Volgenau School of Engineering

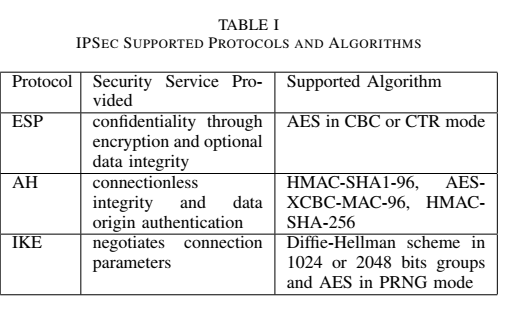
George Mason University

Fairfax, Virginia 22030

email: {asalman, mrogawsk, [jkaps}@gmu.edu](mailto:jkaps%7d@gmu.edu)

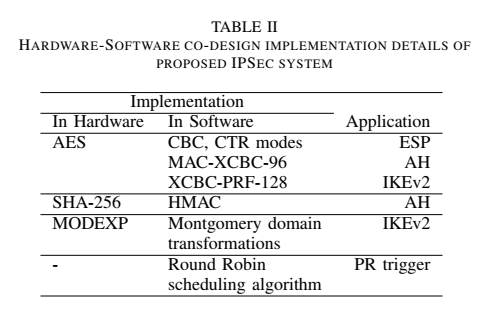
* Tóm tắt

project được thực hiện trên Virtex 4 của Xilinx. Giải pháp được đề xuất hỗ trợ ba giao thức IPSec chính: Bảo mật đóng gói Tải trọng (ESP), Tiêu đề xác thực (AH) và Khóa Internet Trao đổi (IKE). chúng tôi đề xuất một bộ phận các cơ chế cơ bản của giao thức IPSec, cụ thể là mã hóa các thuật toán và phương thức hoạt động của chúng được triển khai trong phần mềm hoặc phần cứng. Qua đó, chúng tôi có thể kết hợp hiệu suất cao được cung cấp bởi giải pháp phần cứng với sự linh hoạt của việc triển khai phần mềm. Chúng tôi chỉ ra rằng một cấu hình giao thức IPSec điển hình có thể được kết hợp với Một phần kỹ thuật cấu hình lại để sử dụng hiệu quả tài nguyên phần cứng.



* Cấu trúc hệ thống

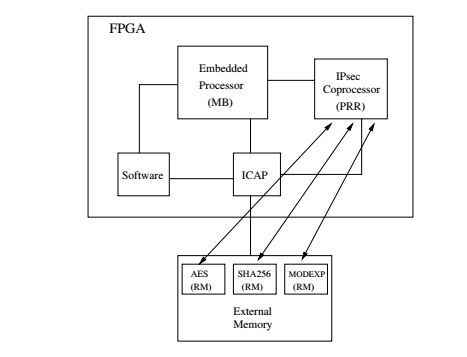
Cấu trúc của hệ thống nhúng IPSec được đề xuất của chúng tôi được tóm tắt trong Bảng II. Chúng tôi đã thực hiện ba mật mã các biến đổi trong phần cứng ở bất kỳ điểm nào đã cho so với thời gian cần thiết để chuẩn bị dữ liệu. Mặt khác, nếu cả hai độ trễ quá cao, khả năng phản hồi của hệ thống sẽ bị suy giảm. Cuối cùng, để tối đa hóa hiệu quả của thời gian tính toán, độ trễ của giao diện cần tối thiểu



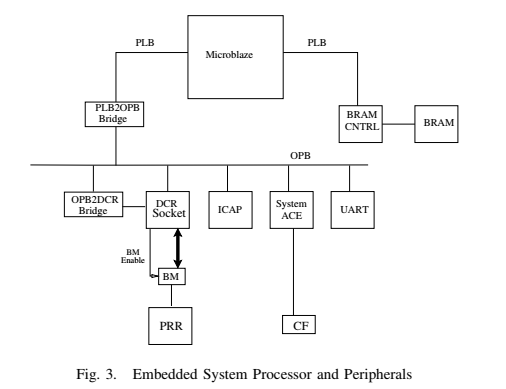
* Cấu hình lại một phần:

Một hệ thống PR điển hình bao gồm các vùng tĩnh đã biết như Base Region (BR) vùng động được gọi là Partial Reconfigurable Region (PRR)

BR giữ phần của thiết kế không bị ảnh hưởng bởi cấu hình lại một phần trong khi PRR giữ phần của thiết kế được đổi chỗ trong quá trình cấu hình lại một phần, được biết đến như Mô-đun có thể cấu hình lại (RM). Một PRR bao gồm tại ít nhất một RM và thường là nhiều RM như trong



* Cấu trúc phần cứng

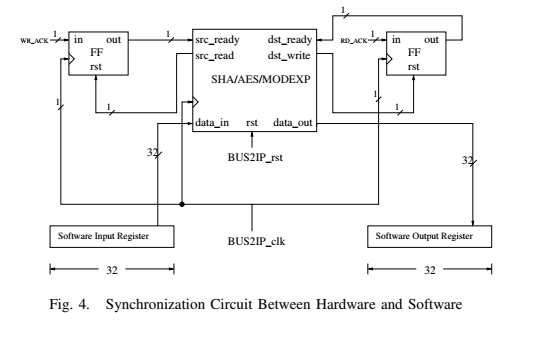


* Phần mền

Phần mềm của hệ thống bao gồm phần mềm trình điều khiển cho các thiết bị ngoại vi phần cứng trong hệ thống, một số các thư viện C cơ bản cũng như các hàm khởi tạo cho HWICAP và API ICAP. Thêm vào đó, tất cả các chế độ hoạt động (ví dụ: CBC và XCBC) và chuẩn bị dữ liệu (ví dụ:Tính toán HMAC) đang được thực hiện trong phần mềm để cung cấp cho

hệ thống linh hoạt như các hoạt động này có thể được sử dụng với thuật toán mã hóa bằng cách chỉ áp dụng phần mềm tối thiểu thay đổi hệ thống

* Mạch đồng bộ



* Phương pháp

IKEv2 vs AH and IKEv2 vs ESP

Trong một đường hầm IPSec Security Policy Database (SPD) điển hình cấu hình, một sự tái đàm phán tham số kết nối được thiết lập không quá một lần mỗi giờ. Mặc dù có một số ứng dụng yêu cầu cấu hình kỳ lạ hơn. AH và xử lý giao thức ESP sẽ mất hơn 99% bất kỳ đồng bộ thực tế IPSec. Do thực tế đó, IKEv2 hiếm khi được sử dụng so với AH / ESP. MODEXP,

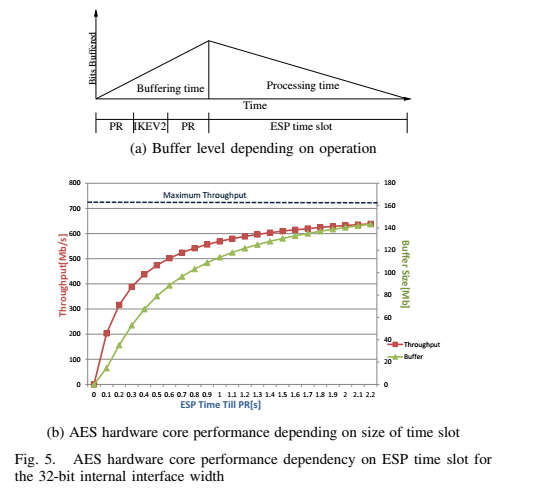
là một mô-đun cơ bản cho giao thức này, có thể được hoán đổi với AH / ESP tăng tốc khi cần thiết. Phần cứng AH và ESP máy gia tốc cũng đang được sử dụng thay thế cho nhau trên chip sử dụng PR nhưng do thực tế là cả hai đều được sử dụng quá mức, các kỹ thuật như sử dụng hàng đợi đầu vào hoặc bộ nhớ trên bo mạch như cũng như các thuật toán lập lịch trình gói như chúng tôi đã mô tả trong phần trước nên được sử dụng để cho phép thông lượng cao.

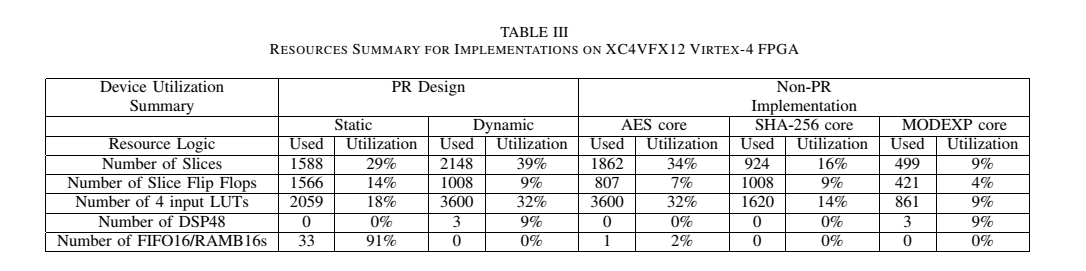
AH vs ESP

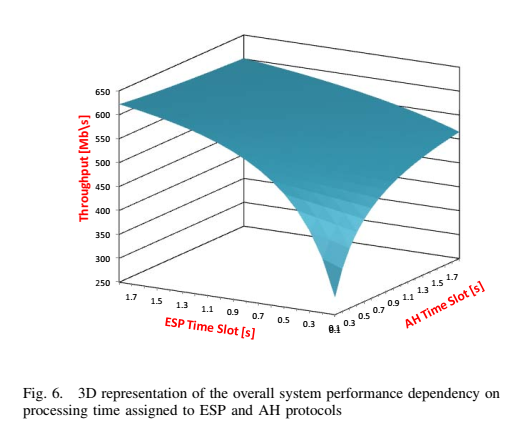
Trong trường hợp của ESP và AH giao thức tình hình với Cấu hình lại một phần hiệu quả là phức tạp hơn. Đầu tiên của tất cả, số lượng các nhiệm vụ liên quan đến AH vs ESP là nhiều hơn cân bằng hơn trong trường hợp của IKEv2. Điều này có nghĩa là theo thứ tự để giảm ảnh hưởng của tương đối đắt tiền, về mặt độ trễ, hoạt động định cấu hình lại một phần, chúng ta phải sử dụng lên lịch các nhiệm vụ. Chúng tôi đã chọn Round Robin theo thời gian thuật toán lập lịch chia sẻ, bởi vì nó rất đơn giản và thuật toán lập lịch trình nổi tiếng. Một điều rất quan trọng đặc tính của thuật toán này là an ninh của nó chống lại vấn đề [18]

* Kết quả

Được thực hiện trên Design Suite 9.1 nạp lên XC4VFX12 Virtex-4 FPGA trên board ML403







Kết luận

Chúng tôi đã triển khai thiết kế PR để thực hiện giao thức IPSec hoạt động trong phần cứng. Thiết kế được chia thành tĩnh vùng đại diện cho một bộ xử lý nhúng Microblaze với một số thiết bị ngoại vi hỗ trợ và khu vực năng động đại diện cho một đồng bộ xử lý IPSec để thực hiện AH, ESP và Tính toán IKEv2 sử dụng bộ tăng tốc phần cứng. Kết quả cho thấy rằng thiết kế PR cho thấy những cải tiến đáng kể về tiết kiệm diện tích so với thiết kế phi PR. Những tiết kiệm trong khu vực có thể được cải thiện hơn nữa nếu có mật mã hơn các thuật toán được hỗ trợ bởi IPSec được thực hiện dưới dạng RM trong đồng xử lý. Thuật toán lập lịch được sử dụng để xử lý nhiệm vụ nhiệm vụ để giảm thiểu tác động của độ trễ bổ sung gây ra bởi quá trình PR. Một ưu điểm khác của giải pháp này có thể được quan sát thấy khi so sánh thời gian sử dụng của IKEv2 chức năng chống lại các chức năng ESP và AH. Triển khai một tăng tốc phần cứng cho IKEv2 như một RM không chỉ tiết kiệm diện tích với hầu như không có hình phạt thời gian, nhưng nó cũng bảo vệ mô-đun từ các cuộc tấn công đặc biệt nếu nó được cài đặt trước bằng các phím. bên trong trường hợp của ESP và AH tăng tốc phần cứng, lưu lượng giao thông nên cao để giảm thiểu PR và tăng thông lượng

REFERENCES

[1] RFC-4301, “http://www.ietf.org/rfc/rfc4301.txt,” 2005.

[2] RFC-4308, “http://www.ietf.org/rfc/rfc4308.txt,” 2005.

[3] RFC-4309, “http://www.ietf.org/rfc/rfc4309.txt,” 2005.

[4] J.-P. Kaps, “Cryptography for ultra-low power devices,” Ph.D. Dissertation, ECE Department, Worcester Polytechnic Institute, Worcester,

Massachusetts, USA, May 2006.

[5] Early Acess Partial Reconfiguration, User Guide, Ug208 (v1.1) ed.,

Xilinx, Inc., Mar 2006.

[6] K. Project, “http://www.kame.net/project-overview.html,” 2006.

[7] 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.

[8] J. Lu and J. Lockwood, “Ipsec implementation on xilinx virtex-ii pro

fpga and its application,” in Reconfigurable Architecture Workshop,

RAW, 2005.

[9] 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.

[10] P. R. Schaumont, A Practical Introduction to Hardware/Software Codesign. Springer, 2010.

[11] 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.

[12] Fortinet, “http://www.fortinet.com/products/fortigate/,” 2011.

[13] An Introduction to the Helion IPsec ESP Engine, v. 1.0.0 ed., Helion

Technology Limited, 2006.

[14] Sun Crypto Accelerator 4000 Board, v. 1.1 ed., Sun Microsystems, Inc.,

2004.

[15] Increasing Design Functionality with Partial and Dynamic Reconfiguration in 28-nm FPGAs, v. 1.0 ed., Altera Corporation, 2010.

[16] Partial Reconfiguration, User Guide, Ug702 (v12.1) ed., Xilinx, Inc.,

May 2010.

[17] A. A. Salman, “IPSec implementation in embedded systems for partial

reconfigurable platforms,” Masters Thesis, ECE Department, George

Mason University, Fairfax, Virginia, USA, May 2011.

[18] A. Silberschatz, P. B. Galvin, and G. Gagne, Operating System Concepts.

Wiley, 2008.

[19] ML401/ML402/ML403 Evaluation Platform, User Guide, Ug080

(v2.5) ed., Xilinx, Inc., May 2006.

[20] K. Gaj and P. Chodowiec, Cryptographic Engineering. Springer, 2009,

ch. FPGA and ASIC Implementations of AES, pp. 235–294.

[21] 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.

[22] H. Orup, “Simplifying quotient determination in high-radix modular

multiplication,” in Proceedings of the 12th Symposium on Computer

Arithmetic, Jul 1995, pp. 193–199.

[23] 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, C¸ etin K.. Koc¸, and C. Paar,

Eds., vol. 2523. Springer-Verlag, 2002, pp. 291–302.

[24] 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.

[25] E. Oks ¨ uzo ¨ glu 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.

[26] Hardware Interface of a Secure Hash Algorithm (SHA), v. 1.4 ed.,

Cryptographic Engineering Research Group, George Mason University,

Jan 2010.

[27] Secure Hash Standard (SHS), National Institute of Standards and

Technology (NIST), Oct. 2008, http://csrc.nist.gov/publications/fips/

fips180-3/fips180-3\ final.pdf.

[28] Advanced Encryption Standard (AES), National Institute of Standards

and Technology (NIST), FIPS Publication 197, Nov 2001, http://csrc.

nist.gov/publications/fips/fips197/fips-197.pdf.

[29] Openssl, “http://www.openssl.org/docs/apps/openssl.html,” 2009.