Hacia una arquitectura hardware de altas prestaciones para securización de Ethernet

Eduardo Jacob, Purificación Saiz, Armando Astarloa, Jon Matías, Marina Aguado, Christian Pinedo, Enrique Areizaga (Fundación Robotiker)

ETSI de Bilbao- Universidad del País Vasco- Euskal Herriko Unibertsitatea (UPV/EHU)
Dpto. de Electrónica y Telecomunicaciones
C/ Alda. Urquijo S/N, 48013 Bilbao
Telf: 946 014 214, Fax: 946 014 259
E-mail: Eduardo.Jacob@ehu.es

Resumen

El objeto de este artículo es ilustrar nuestra aportación la investigación realizada hasta la fecha en el campo de la seguridad en el nivel Ethernet al amparo de varios proyectos de investigación en curso. En seguridad, el cifrado y la autenticación van ligados. De hecho, el cifrado solo tiene sentido si podemos asegurar el origen de la información. Aun cuando a lo largo del trabajo desarrollado se han estudiado e implementado los mecanismos de autenticación, en este artículo se tratará de la implementación del cifrado (a alta velocidad) en una plataforma para seguridad en Ethernet y de la evolución de la arquitectura inicialmente software a hardware.

1. Introducción

El interés que presenta en la actualidad la utilización de Ethernet en otros escenarios distintos de la red de áreas local es creciente. Después de una etapa inicial en la que el All-IP como tecnología se consideraba única y suficiente para todos los ámbitos, se empieza a considerar que las tecnologías de bajo nivel son complementarias y pueden aparecer en otros escenarios como el backbone ó el acceso. En el caso concreto de la tecnología Ethernet, el trabajo estandarizador se está llevando tanto desde diversos organismos IEEE EFM [1], MEF[2]) e ITU.

2. Evolución del diseño

El objetivo que nos planteamos era diseñar un sistema de cifrado Ethernet extremo a extremo. Esto nos separa ya de la iniciativa del IEEE 802.1ae que persigue seguridad "salto a salto" descifrando y recifrando en cada elemento activo la información.

Para progresar de una manera ordenada se diseñó originalmente una evolución que permitiera ir resolviendo los problemas de manera paulatina.

2.1. Versión Soft-Linux i386

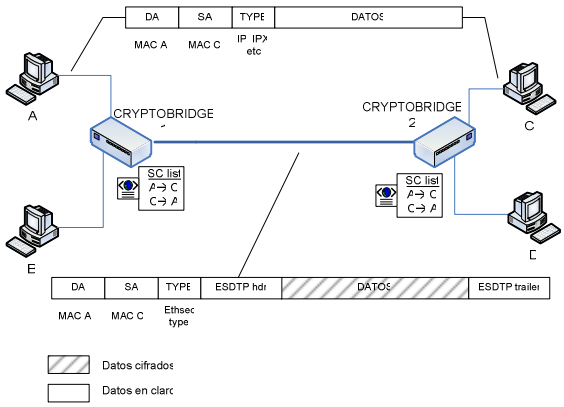
En esta primera fase, se trataba de implementar una primera versión sobre un ordenador personal Linux. Esta versión atacaría el driver de red a través de los módulos de bridging del sistema operativo y realizaría las operaciones de cifrado a través de rutinas software. Desde el punto de vista de protocolo, se diseñó una trama que tenía muchos puntos en común con la trama de 802.1ae e integraba dentro del mismo algunos procedimientos de gestión. Ver Fig. 1. Esta versión vio la luz en forma de bridge cifrante, con dos interfaces Ethernet de 100 Mbs.

2.2. Versión Linux-Padlock

Esta segunda versión buscaba dos objetivos: El primero reducir el tamaño y coste del equipo utilizando una placa base de formato Mini-ITX con una CPU de EPIA que dispone de la funcionalidad Padlock© [3], es decir un juego de instrucciones adicional que posibilita la generación de números aleatorios y cifrado AES. La utilización de AES en modo CBC es soportada directamente y hubiera permitido mejorar el rendimiento de la primera implementación en caso de seleccionar dicho algoritmo.

Por otro lado se decidió que la inclusión del protocolo de gestión no era eficiente, generaba problemas de evolución y tenía limitaciones desde el punto de vista de la seguridad. Esto fue la causa de dos decisiones importantes: La primera fue el

Figura 1. ESDTP – Trama Ethsec Secure Data Transport Protocol



abandono del formato inspirado en 802.1ae y la correspondiente separación de las tareas de gestión de las del transporte. La otra fue la utilización de

otra trama distinta, en esta ocasión, inspirada de 802.11i, con los cambios en los algoritmos de cifrado correspondientes.

Esta versión no vio la luz y en su lugar se inició otro desarrollo.

2.3. Versión Linux – Cifrado Hardware

Como resultado de la versión anterior nos planteamos la utilización de plataformas que permitieran el procesado hardware del cifrado. La solución retenida pasaba por la utilización de configuraciones de SOPC con FPGA.

Para esta versión se utilizó una placa de evaluación Xilinx ML310 que lleva una FPGA Virtex II Pro, que contiene en su interior dos procesadores empotrados PowerPC 405. Ver Fig. 2.

En esta placa se proporciona un puerto serie para su programación, una interfaz Ethernet por la que realizar las comunicaciones de red y también memoria RAM y Flash que posibilitan incluso la ejecución de un sistema operativo en la FPGA. Entre los sistemas operativos disponibles se dispone de la distribución MontaVista de Linux con soporte para tiempo real.

Dado que esta placa no dispone de dos puertos Ethernet, no se diseñó la funcionalidad de bridge cifrante sino que se optó por el desarrollo de una API para acceso a aplicaciones seguras sobre redes Ethernet, que pudiera ser empleada posteriormente para la implementación de cualquier tipo de servicio.

La implementación realizada dispone de una API que cubre las necesidades de comunicación:

- Envío de trama
- Recepción de trama

Y las necesidades de gestión de claves y asociaciones de seguridad AS:

- Establecimiento de AS: Clave y MAC destino
- Cambio de Clave
- Eliminación de AS

La trama utilizada está inspirada en la utilizada para 802.11i y proporciona autenticación y cifrado a través de la utilización del algoritmo AES-CCM. La cabecera EthSEC, es bastante similar a la cabecera CCMP de 802.11i. Se soporta el uso de tags Vlan según el estandar 802.1q con objeto de permitir el tránsito de las tramas por infraestructuras que las utilicen. Ver Fig.3

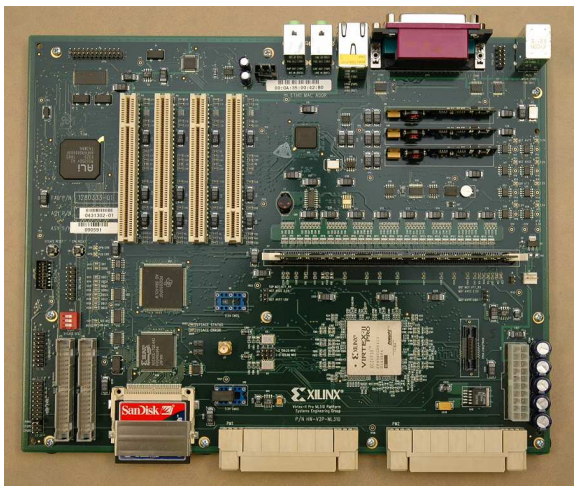


Figura 2. Plataforma de desarrollo ML310

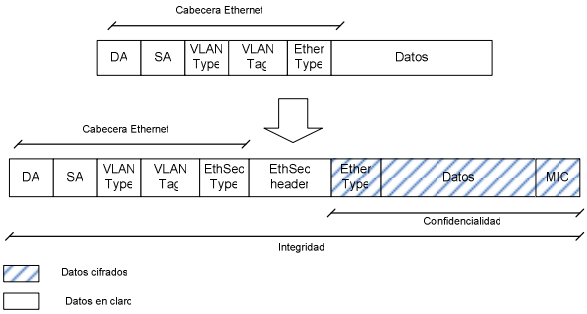


Figura 3. Procesamiento básico de tramas 802.1q en EthSEC

La autenticación y gestión de claves se realizaría por medio de 802.1x o en el caso de cifrado extremo a extremo por una variante de 802.1x desarrollada en el marco del proyecto [4]

La versión disponible en este momento realiza el procesado de la trama Ethernet en espacio de usuario entregando y recuperando del nivel físico las tramas con llamadas al sistema y cifrando las mismas con rutinas software de dominio público [5]

En las pruebas de rendimiento preliminares y en comparación con una implementación basado en un PC i386, aparentemente la implementación en SOPC presenta un rendimiento superior en relación a la velocidad de reloj de la CPU. Todavía no disponemos de conclusiones definitivas que nos permitan validar estos resultados o las causas.

En este momento nos encontramos integrando rutinas de cifrado basado en FPGA para realizar el cifrado AES, para lo que se van a utilizar cores Open Source. Existen diversos cores, en los que se permuta ocupación de la FPGA (slices y RAM) por rendimiento. Se ha estudiado en el ámbito del proyecto diversas implementaciones bajo este punto de vista [6]. Véase la tabla 1 para visualizar las alternativas.

Tabla 1. Comparación de COREs AES

Resources	R. Usselman	J. Castillo	H.V. Kampen
4 input LUTs	8.184 (74%)	3.034 (27%)	1.242 (11%)
Slice Flip-Flops	3.705 (33%)	1.362 (12%)	846 (7%)
Virtex-4 Slices	5.470 (99%)	1.923 (35%)	922 (16%)
18K Block Ram	4 (11%)	0 (0%)	2 (5%)
Equivalent gate count	94.480	30.703	17.436
Maximum running speed	135 MHz	139 MHz	149 MHz
Data throughput	1,44 Gbit/s	35 Mbit/s	1,9 Mbit/s

2.4. Versión con Procesado HW de tramas y cifrado

La versión siguiente de la plataforma incluye un cambio fundamental: El tratamiento por hardware no sólo del cifrado de la trama Ethernet, sino también de la inserción y recuperación del nivel de enlace de la trama en cuestión. De esta manera, se puede mantener dentro de la FPGA y de manera óptima todos los procesos intensivos. Cada una de estas funciones va a ser realizadas por un core específico.

Con este diseño la utilización de un sistema operativo completo como Linux sólo se justifica en el caso de querer implementar servicios de alto nivel como autenticadores 802.1x o interfaces de usuario basados en web.

En este caso para el prototipo no se ha incluido un sistema con Linux. En su lugar se ha utilizado una plataforma con la FPGA Xilinx Virtex-4 FX12, que contiene un PowerPC de 32-bits como microprocesador y del OPB (On-chip Peripheral Bus) conectado al EMAC (Ethernet Media Access Controller)

En el ámbito del proyecto, se ha desarrollado un driver EMAC que utiliza DMA para las transmisiones y que implementa las siguientes funcionalidades:

- Inicialización del driver del EMAC
- Definición del handler general del EMAC
- Definición del handler de recepción de una trama
- Definición del handler de transmisión de una trama
- Definición del handler de error
- Captación de una trama
- Transmisión de una trama
- Actualización de las estadísticas del EMAC
- Obtención de las estadísticas del EMAC

También se ha desarrollado un core EthSEC que realiza las funciones de filtrado y cifrado de las tramas que necesitan ser tratadas. A este módulo se le pueden cargar los parámetros que definen una asociación de seguridad, como se ha visto antes. El procesador se encarga de orquestar todos los dispositivos. La arquitectura se muestra en Fig. 4.

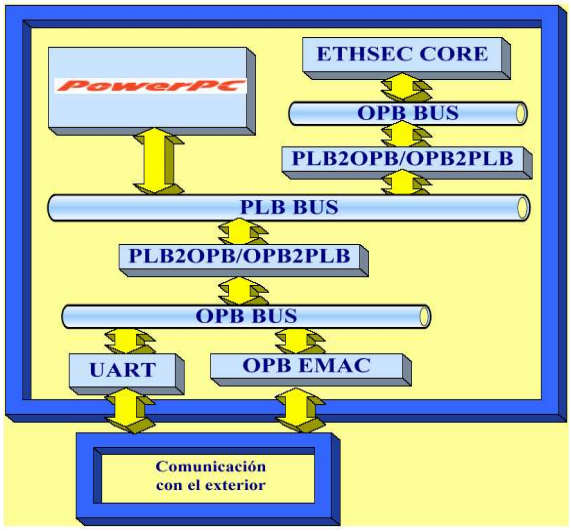


Figura 4. Arquitectura con Core EthSEC

Una vez correctamente configurado el sistema, el módulo EthSEC opera de manera autónoma cifrando y descifrando las tramas en función de las necesidades, sin necesidad de consumir ciclos de la CPU. Se dispone lógicamente de la posibilidad de modificar los parámetros de asociación de seguridad.

El prototipo conviene para la realización de comunicaciones cifradas de tipo industrial y que se configuran externamente.

En este momento, el core EthSEC no implementa el formato de trama EthSEC vista previamente, sino uno simplificado basado en AES-CBC. El nuevo core EthSEC-II capaz de trabajar con el nuevo formato de trama está en desarrollo.

3. Conclusiones.

Este artículo ha mostrado la evolución de una arquitectura de cifrado desde una solución puramente software hasta una solución puramente hardware. La aproximación a través de un equipo interdisciplinar ha permitido obtener buenos resultados.

4. Líneas futuras

El trabajo futuro se articula en base a dos líneas.

4.1. Utilización del core EthSEC-II en un SOPC con sistema operativo Linux

Esta línea supone utilizar el core EthSEC-II dentro de la arquitectura presentada en el punto 2.3. Esto mejorará el rendimiento y la seguridad de manera notable.

4.2. Adaptación de mecanismos de autenticación a la plataforma

Precisamente vinculado al punto anterior, se van a integrar los mecanismos de autenticación 802.1x adecuados a los dos escenarios presentados previamente: Redes de acceso y enlaces punto a punto.

Agradecimientos

El trabajo presentado en este artículo ha sido posible entre otros a la financiación recibida del Plan Nacional de I+D+I (2000-2003), a través del proyecto MCYT "**EthSEC.** Seguridad en Redes Ethernet Extremo a Extremo" (2000-2003) TIC2003-09585-C02-01 y del proyecto del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (I+D+I 2004-2007) "**PlaNetS**. Platform for Networked Service Delivery". FIT-330220-2005-111

Queremos también agradecer a Aitor Álvarez Gila, Esther Torvisco Zarandona y Naroa González Sánchez la ayuda prestada durante el desarrollo de los prototipos.

Referencias

[1] IEEE P802.3ah Ethernet in the First Mile Task Force. Disponible en: http://www.ieee802.org/3/efm/

[2] Metro Ethernet Forum. Disponible en: http://www.metroethernetforum.org/

[3] Via Padlock Initiative. Disponible en: http://www.via.com.tw/en/initiatives/padlock/

[4]P. Sáiz, J. Matías, E. Jacob, J. Bustamante, A. Astarloa, *"Adaptation of IEEE 802.1X for secure session establishment between Ethernet peers"*. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg, 2006. Vol: 4332 pp.220-234

[5] AES and Combined Encryption/Authentication Modes. Disponible en: http://fp.gladman.plus.com/AES/

[6] A. Astarloa, P. Saiz, J. Lázaro, E. Jacob y U. Bidarte *"Multi-architectural 128 bit AES-CBC Core based on Open-Source Hardware AES Implementations for Secure Industrial Communications"*. Proceedings "The International Conference on Communication Technology (ICCT'2006)"