Adaptación de la docencia de una asignatura de criptografía a las recomendaciones del Espacio Europeo de Educación Superior.

Pedro M. Alcover, Juan Suardíaz, *Member IEEE*, Pedro J. Navarro.

***Title*— Adapting the teaching methodology of a cryptography subjet to the recommendations of the European Higher Education Area.**

***Abstract*— This paper summarizes the adaptation to the requirements of the Bologna declaration of 19th June, 1999. In this new European context, it is required not only theoretical concepts, but also a learning methodology where professors have a secondary role and students are the main actors of the learning process. We present a proposal of adaptation for a subject of cryptography, which combines a short theoretical review of basics concepts and a set of practical exercises. These exercises are designed with the aim to achieve a global perspective of developing cryptographic systems. A modular approach was selected in order that every new design required results generated in the previous one. A case study of an implementation of a hash function is presented. This exercise demands some skills related with different subjects of their curricula to develop a fully functional system, focusing the learning on a project-based design.**

***Index Terms*— Education, Cryptography, Learning by competencies, Bologna declaration.**

I. INTRODUCCIÓN

La docencia en la Universidad española está inmersa en el proceso de Convergencia Europea, en busca de la integración en el Espacio Europeo de Educación Superior [1]. Se trabaja en la adaptación de las titulaciones, los curricula y los procesos de enseñanza – aprendizaje para lograr alcanzar los objetivos que recoge la declaración de Bolonia.

Entre los másteres de postgrado ofertados en nuestra Universidad [2] está el de Informática Industrial. Este máster profesional se imparte en la Escuela Técnica Superior de Ingenieros de Telecomunicaciones de la Universidad Politécnica de Cartagena; centra su interés en la expansión de las Tecnologías de la Información y las Comunicaciones (TIC) en los campos de la Informática, la Robótica y la Electrónica dentro del sector industrial. Y entre las asignaturas que componen el programa del máster se encuentra una llamada "*Introducción a la Criptografía Aplicada*". Es una asignatura breve, de 2 créditos ECTS, asociada con otra que la sigue, llamada "*Seguridad en Redes y Protección de la Información*".

Presentamos en este artículo una propuesta para la adaptación de los contenidos temáticos de una asignatura de introducción a la criptografía, como es "*Introducción a la Criptografía Aplicada"*, dentro del marco provisto por Bolonia. En la segunda sección se recoge una planificación general de la asignatura, con una distribución de actividades y horas de dedicación, contenidos de la asignatura y documentación entregada a los alumnos. En la tercera sección presentamos el programa de prácticas planteadas. Luego, en la sección cuarta, mostramos el sistema que hemos decidido adoptar para la evaluación final de los alumnos. En la quinta sección se recoge una breve descripción de los proyectos propuestos a los alumnos como trabajo fin de máster. Terminamos con unas breves conclusiones y experiencias del trabajo realizado con los alumnos.

II. PLANIFICACIÓN GENERAL DE LA ASIGNATURA

El objetivo de la asignatura "*Introducción a la Criptografía Aplicada*" es ofrecer a los alumnos del máster una panorámica introductoria de los principales algoritmos criptográficos. La corta duración de la asignatura exige una programación resumida y una planificación de la docencia y de las prácticas que permita al alumno conocer perfectamente los conceptos, comprender su utilidad y verificar su correcta asimilación.

Para la elección de los contenidos, hemos buscado aquéllos que mejor nos permitieran abordar los cuatro objetivos principales de la criptografía: confidencialidad, integridad de la información, autenticación y no repudio [3]. Y, condicionados por la brevedad del curso, hemos seleccionado los que a continuación se relacionan, con la planificación de horas que queda recogida en la Tabla I. La descripción del contenido de las sesiones prácticas quedará recogida en la sección siguiente.

Primera clase teórica (2 horas): Nociones introductorias: Objetivos y logros de la criptografía. / Criptosistemas. / Criptosistemas simétricos y asimétricos. / Criptoanálisis. / Cifrado de bloque. / Cifrado de flujo. / Cifrado de clave asimétrica. / Funciones Hash. / Firma digital. / Cifrado mixto.

Segunda clase teórica (2 horas): Data Encryption Standard – DES: Breve introducción histórica. / Fundamentos teóricos

P. Alcover, P. J. Navarro y J. Suardíaz son profesores en la Universidad Politécnica de Cartagena; los dos primeros en el área de Lenguajes y Sistemas Informáticos; el tercero en el Área de Tecnología Electrónica.

DOI (Digital Object Identifier) Pendiente

TABLA I
ESTIMACIÓN DE HORAS DE DEDICACIÓN

Tarea docente o de aprendizaje	horas
Exposición del profesor en el Aula	12
Nociones introductorias	2
Cifrado Simétrico. DES	2
Modos de cifrado	2
Funciones Hash	2
RSA: nociones matemáticas y algoritmo.	4
Presentación de los trabajos a realizar	2
Prácticas de laboratorio (Trabajo presencial)	10
Trabajo presencial en el aula	24
Trabajos de programación en horario libre	12
Estudio y preparación del examen	12
Tutorías y consultas	1
Examen	1
Total trabajo fuera del aula	26
TOTAL	50

previos: cifrado de producto y cifrado de Feistel. / Algoritmo DES. / Seguridad del DES. Ataque por fuerza bruta. / DES no es un grupo: Triple DES.

Observación al contenido de este capítulo o clase: Aunque conocemos que el algoritmo DES es ya obsoleto, hemos preferido mostrar éste en lugar de otros más actuales (por ejemplo, AES) por dos motivos: por el indudable interés histórico del algoritmo DES; y por la abundante documentación sobre los algoritmos S–DES (algoritmo DES simplificado) que, como diremos más adelante, hemos utilizado como elemento base para toda la implementación de algoritmos y sus correspondientes ataques de criptoanálisis.

Primera clase práctica (2’5 horas).

Tercera clase teórica (2 horas): Modos de operación en el cifrado por bloques: modo de cifra ECB: Electronic Codebook Mode. / Modo de cifra CBC: Cipherblock Chaininig Mode. / Modo de cifra CFB: Cipher Feedback Mode. / Modo de cifra OFB: Output Feedback Mode.

Segunda clase práctica (2’5 horas).

Cuarta clase teórica (2 horas): Funciones Hash: Funciones Hash de uso criptográfico. / Ataque y Paradoja del cumpleaños. / Funciones de comprensión desde funciones de cifrado. / Funciones hash desde funciones de comprensión.

Tercera clase práctica (2’5 horas).

Quinta y sexta clases teóricas (4 horas): Nociones matemáticas para RSA: Definición del conjunto de los enteros: algunas propiedades. / Algunas nociones algebraicas sobre conjuntos. / Relación de Equivalencia. Relación de Congruencia. / Cálculo del inverso en una aritmética modular con módulo compuesto del que se desconocen sus factores. / Algoritmo RSA. / Criptoanálisis de RSA. Intercambio de claves: protocolo Diffie – Hellman. Firma digital.

Cuarta clase práctica (2’5 horas)

Además de la documentación que desarrolla este temario, a cada alumno se le ha entregado la siguiente documentación complementaria: para el capítulo y clase de Data Encryption Standard-DES, la bibliografía de la referencia [4] y [5]; para el capítulo y clase de Funciones Hash, la documentación de la referencia [6]; y para el capítulo y clase de RSA los artículos [7] y [8].

III. PROGRAMACIÓN DE PRÁCTICAS

Los trabajos o prácticas han sido diseñados de manera que permitieran al alumno comprender los conceptos propuestos para la asignatura, y comprobar la complejidad de los algoritmos tanto criptográficos como de criptoanálisis. A cada alumno se le han planteado cuatro trabajos o prácticas, para desarrollar de forma individual o en grupo con otro alumno. Cada alumno puede elegir el lenguaje de programación en el que quiere trabajar. Se han desarrollado los trabajos, principalmente en C y en Java. Algunos pocos alumnos lo han realizado en C++, y uno de ellos en *Mathematica*. Los trabajos se han realizado en un laboratorio con equipos PC, con SO Windows o Linux. Bastantes de los alumnos, sin embargo, han preferido trabajar en el aula directamente sobre su portátil.

A. Primer trabajo o práctica.

Programar un DES simplificado (llamado S-DES). Se han tomado como posibles algoritmos de DES simplificado dos presentados por Edward Schaefer, de la Universidad de Santa Clara.

Uno de ellos (S-DES1) está presentado en [9] pp. 56 a 63. El otro (D-DES2), similar, lo hemos tomado de [10] pp. 98 a 102. El primero de ellos cifra mediante una clave de 10 bits, y con un tamaño de bloque de 8 bits. El tamaño de las subclaves es de 8 bits. El segundo de ellos cifra mediante una clave de 9 bits, y con un tamaño de bloque de 12 bits. El tamaño de las subclaves es de 8 bits. En ambas referencias se describe el algoritmo pormenorizadamente.

El objetivo de este primer trabajo es doble. Por un lado, el alumno debe implementar uno de los dos modelos de S-DES y realizar cifrado y descifrado de mensajes mediante bloques. Además, debe implementar la forma de ataque por fuerza bruta que le permita obtener la clave una vez tiene un bloque plano y su correspondiente cifrado. Evidentemente, con estos tamaños de clave (10 y 9 bits respectivamente) este ataque es eficaz y muestra la debilidad de estos criptosistemas y la necesidad de usar claves de mayor longitud de bits.

La implementación de uno de estos algoritmos no es muy costosa para un alumno que ya tiene conocimientos de programación. Ha supuesto de dos a cuatro horas de trabajo del alumno por cada práctica presentada. No es sencillo determinar una duración estándar porque cada alumno realiza su trabajo con el lenguaje que prefiere y porque no es homogénea la pericia a la hora de crear el programa.

La realización de esta práctica ayuda a caer en la cuenta de dos aspectos básicos de la criptografía simétrica: (1) que una vez implementado el algoritmo de cifrado, es casi inmediato, con las mismas funciones de cifrado, obtener la implementación del descifrado; y (2) que la implementación del ataque por fuerza bruta es también muy sencilla una vez se tiene implementado el algoritmo de cifrado. Los alumnos han podido obtener las claves de cifrado, en un ataque conocido como *Known plaintext attack* (ataque del que se conoce un

mensaje original cualquiera y su correspondiente cifrado: cfr. [3] § 1.13.1), en un tiempo mínimo, imperceptible en la ejecución.

B. Segundo trabajo o práctica.

Una vez implementado correctamente el algoritmo S-DES, se les propone a los alumnos que completen su pequeña aplicación permitiendo una entrada de texto a cifrar de cierta longitud (un documento de texto ASCII) y generando una cadena de longitud similar a la entrada de bloques cifrados. Se les plantea que elijan entre uno de los cuatro modos de operación de cifrado de bloque que se les ha presentado en clase (ECB, CBC, CFB, OFB comentados anteriormente) y se les pide también que justifiquen su elección.

Con esta práctica se pretende que el alumno vea el cifrado de un documento completo, y su posterior descifrado. También deben enfrentarse a la decisión del modo de operación de cifrado, aunque frecuentemente el alumno determina usar el modo CBC por ser más seguro que el inmediato ECB y notablemente más sencillo de implementar que los modos CFB y OFB.

El tiempo empleado en la implementación de esta práctica ha sido también variable de un trabajo a otro; además de las causas indicadas antes en el apartado anterior, en este caso el alumno ha podido elegir diferentes modo de cifrado. Para el caso más habitual de elección, el modo CBC, el alumno ha empleado también entre dos y cuatro horas de trabajo.

Esta práctica, además de ayudar a comprender bien el concepto de los modos de cifrado, ha exigido a algunos alumnos algunas modificaciones en el código de la práctica anterior: les ha enseñado que, además de saber implementar el algoritmo, un criptosistema requiere de la construcción de un protocolo de proceso. Además, todos los modos de cifrado (excepto el modo ECB) requieren de un valor inicial de tantos bits como el tamaño de bloque del criptosistema. Ese valor es aleatorio y, por tanto, supone una incertidumbre más en el uso del criptosistema: se comporta como una segunda clave añadida a la clave del criptosistema. Ahora el ataque por fuerza bruta antes implementado ya no sirve: los alumnos comprueban que así pueden incrementar la seguridad del criptosistema.

C. Tercer trabajo o práctica.

Una vez los alumnos tienen implementado el algoritmo S-DES y en correcto funcionamiento, el siguiente trabajo que se les propone es el diseño e implementación de una función de compresión a partir de una función de cifrado. Y luego el diseño e implementación de una función hash a partir de una función de compresión. La idea de implementar una función Hash sin clave a partir de una función de compresión fue propuesta por Ralph Merkle. Queda descrita en [10], pp. 235 – 242.

Las funciones de compresión que se obtienen de los algoritmos de cifrado implementados por los alumnos en su primera práctica, serán:

$g_1: \{0,1\}^m \longrightarrow \{0,1\}^n$, con $m = 20$ y $n = 8$, creada a partir de S-DES1.

$g_2: \{0,1\}^m \longrightarrow \{0,1\}^n$, con $m = 21$ y $n = 12$, creada a partir de S-DES2.

A partir de estas funciones de compresión se diseñan e implementan las funciones hash. En [11] viene descrito el modo en que se puede definir esas funciones hash. A partir de g_1 se puede llegar a la función hash $h_1: \{0,1\}^* \longrightarrow \{0,1\}^n$, con $n = 8$. A partir de g_2 se llega a una función hash h_2 similar a h_1, ahora con $n = 12$.

De nuevo el objetivo planteado a los alumnos con este segundo trabajo es doble. Por un lado, implementar una función hash y comprobar con él que se cumplen las propiedades funcionales básicas de estas funciones. Específicamente que al variar en un bit la entrada a la función sufre modificación aproximadamente del 50% de los bits de salida. Y además, los alumnos deben implementar una aplicación que, mediante el ataque de cumpleaños (*birthday attack*) (cfr. [3] § 9.7.1., [11] § 11.2) construya dos documentos diferentes con el mismo hash. Evidentemente es posible plantear este ataque sobre nuestra función hash porque el tamaño del resumen es demasiado pequeño para evitarlo.

El tiempo empleado para la implementación de la función de compresión a partir del S-DES1 o S-DES2, y la implementación de la función hash a partir de la previa función de compresión ha sido largo, de más de 4 horas. Los alumnos han encontrado dificultad en el diseño y posterior implementación de un procedimiento para la construcción de la cadena de entrada al algoritmo de resumen.

Para la parte de criptoanálisis, los alumnos han tenido más problemas. La paradoja del cumpleaños es sencilla de comprender pero exige bastante programación para llevarla a la práctica. No todos los alumnos han llegado a terminar este ejercicio. Los trabajos entregados han resultado brillantes. Se han empleado en clase de teoría para que todos los alumnos pudieran ver el comportamiento del ataque. Aunque no todos los alumnos han logrado culminar este trabajo con éxito, el esfuerzo realizado sí ha merecido la pena desde un punto de vista didáctico: les ha permitido asimilar con precisión todos los conceptos presentados sobre las funciones hash. Hemos evaluado a los alumnos de manera que no se ha penalizado el no haber entregado la parte de ataque en esta práctica. El tiempo medio dedicado a esta parte ha sido elevado, en una media de entre 8 y 10 horas.

D. Cuarto trabajo o práctica.

De forma sencilla, para valores enteros admitidos en el dominio del tipo de dato ***unsigned long*** del C o del Java (lenguajes que mayoritariamente han elegido los alumnos para confeccionar sus prácticas), se les ha propuesto a los alumnos que implementen un sencillo generador de bits aleatorios por entrada de teclado con el que generar una secuencia de bits (por ejemplo, el algoritmo presentado en [12]) para, a partir de ella, tomar dos enteros primos, generar las claves pública y privada del criptosistema RSA y firmar documentos mediante la transformación RSA sobre su correspondiente hash con la clave pública.

También han realizado el proceso de obtención de la clave privada de otro usuario mediante la factorización del módulo de la transformación RSA.

Esta práctica ha resultado quizá demasiado sencilla, pero ha permitido a los alumnos comprender y verificar el cambio de filosofía presente en los criptosistemas que deben su robustez no tanto al tamaño de las claves como sí a la dificultad actual para resolver en un tiempo aceptable determinados retos matemáticos por desconocer un algoritmo que sea computacionalmente eficiente. El tiempo medio dedicado para la implementación de esta práctica ha sido de dos horas.

TABLA II
ESTIMACIÓN DE HORAS DE DEDICACIÓN PARA LAS PRÁCTICAS

Práctica	horas
Primera práctica: S-DES y ataque por fuerza bruta	3
Segunda práctica: modos de cifrado (ECB, CBC, CFB, OFB)	3
Tercera práctica: función de compresión y función hash.	5
Tercera práctica: ataque a la función hash: birthday attack	9
Cuarta práctica: criptosistema RSA	2
TOTAL	22

IV. EVALUACIÓN FINAL DE LOS ALUMNOS

Los alumnos matriculados en la asignatura han sido 42. Todos ellos son titulados en ingeniería técnica telemática, ingeniería técnica en electrónica industrial, u otra especialidad de ingeniería técnica industrial. Algo más de la mitad de ellos con los estudios recién terminados; el resto con ya algunos años de ejercicio profesional.

Siete de ellos se han visto obligados a abandonar el desarrollo del máster, por motivos diversos, independientes del correcto desarrollo de la docencia. Otros seis han quedado sin evaluar en la convocatoria de febrero, por no haber logrado presentar a tiempo todos los trabajos o por no haberse presentado al examen. 28 de los 42 alumnos que se matricularon inicialmente han superado la asignatura en primera convocatoria.

Los alumnos han sido evaluados en cuatro aspectos de su trabajo, a lo largo del desarrollo de la asignatura. El primer día de clase fueron informados del método de evaluación.

A. Primer criterio: la participación en clase.

Mediante el planteamiento de breves ejercicios o de cuestiones de pronta contestación a lo largo de las exposiciones teóricas de la materia. Estas puntuaciones han supuesto un 15 % de la nota final de la asignatura.

B. Segundo criterio: resolución de ejercicios planteados.

Al final de cada clase teórica se ha dejado planteado uno o varios ejercicios para que los alumnos los pudieran trabajar por su cuenta y los presentaran en la siguiente sesión teórica. Estos ejercicios entregables han supuesto otro 15 % de la nota final de la asignatura.

C. Tercer criterio: elaboración de prácticas.

La elaboración de todas las prácticas, tanto las realizadas en el laboratorio en horario lectivo de la asignatura como las que cada alumno ha desarrollado por su cuenta, ha supuesto el máximo porcentaje de evaluación de la asignatura. Cada alumno debía presentar un trabajo o memoria de prácticas, así como los ejecutables de todos los algoritmos implementados: cifrado y descifrado con el algoritmo DES simplificado; ataque al DES por fuerza bruta; función hash obtenida a partir de un algoritmo de compresión, diseñado a su vez de un algoritmo de cifrado (el DES simplificado); y finalmente el ataque a la función hash mediante la estrategia de la paradoja de cumpleaños. La evaluación de este trabajo ha supuesto el 50% de la nota final de la asignatura. Casi todos los alumnos pudieron terminar el ejercicio de RSA en el aula de clase práctica. Como el planteamiento de la práctica era muy sencillo, y el objetivo quedó —a nuestro juicio— cubierto con la asistencia a esa clase, no les exigimos entrega de una memoria definitiva sobre esta parte del trabajo.

La asistencia a las clases de prácticas ha sido optativa, dejando a criterio del alumno la decisión de afrontar todos los trabajos como trabajo particular o contando con la ayuda y asesoramiento de unas clases de prácticas. La totalidad de los alumnos matriculados ha optado por la asistencia a las clases prácticas.

D. Cuarto criterio: cuestionario y test finales.

El 20% restante de la nota se ha tomado del examen final de la asignatura. Este examen tenía dos partes. En la primera el alumno debía entregar un cuestionario con 30 preguntas cortas, que recogía todas las partes teóricas de la asignatura. Ese cuestionario fue entregado a los alumnos unas semanas previas a la fecha del examen. La segunda parte del examen era un cuestionario de preguntas tipo test que fue entregado a todos los alumnos a la vez y para el que dispusieron de una hora para contestar.

El cuestionario de preguntas cubría todas las partes del temario de la asignatura, y para su correcta contestación los alumnos han necesitado realizar una lectura completa de toda la documentación entregada y accesible vía web en la OCW (*OpenCourseWare*) de la Universidad Politécnica de Cartagena [13]. Las contestaciones, de extensión limitada, exigían al alumno un ejercicio de síntesis. La revisión del cuestionario ha permitido en algunos casos clarificar algunos conceptos que, de su lectura, se deducían mal asimilados.

La prueba tipo test, se ha realizado sin posibilidad de consulta de documentación. Cada pregunta ofrecía 4 soluciones, y sólo una de ellas era correcta. Cada respuesta errónea penalizaba medio punto de la nota final. Se les formuló un total de 30 preguntas, y la prueba tuvo una duración de una hora. La nota media de esas pruebas ha sido de 5.3.

E. Cálculo de la nota final.

Todos los alumnos debían obtener una nota mínima, en los criterios B, C y D, de 3.5 sobre 10. Y una nota media de todas las calificaciones igual o superior a cinco.

Las notas finales en la primera convocatoria quedan recogidas en la Tabla III.

TABLA III
RESULTADOS DE LA EVALUACIÓN DE LOS ALUMNOS

Número de alumnos…	
…matriculados	42
…que han abandonado los estudios del máster	7
…no presentados en primera convocatoria	6
…suspensos	1
…que han superado la asignatura con APROBADO	13
…que han superado la asignatura con NOTABLE	10
…que han superado la asignatura con SOBRESALIENTE	5

V. DESCRIPCIÓN DE LOS PROYECTOS FIN DE MÁSTER PLANTEADOS

El máster en el que se encuadra la asignatura presenta, en su último cuatrimestre, una asignatura de fin de máster, en la que cada alumno debe presentar un trabajo o proyecto de investigación, dirigido por algún profesor del máster. El objetivo del proyecto es que el alumno aplique los conocimientos adquiridos en una o varias de las asignaturas del máster.

Los profesores implicados en la asignatura "*Introducción a la Criptografía Aplicada*" hemos presentado también algunos proyectos que permiten a los alumnos aplicar mejor los conocimientos adquiridos en la asignatura y a lo largo del máster.

Se han presentado diferentes propuestas para el trabajo fin de máster, todas ellas englobadas en un proyecto más general donde se aplican disciplinas de varias asignaturas y titulado "Implementación de algoritmos criptográficos sobre dispositivos lógicos programables (FPGA)". Para el desarrollo de esos trabajos el alumno debe haber adquirido, además de los conocimientos impartidos en nuestra asignatura, algunos otros sobre diseño electrónico.

Las tareas que debe acometer el alumno son:

1. Realizar un acopio de información coherente con objeto de lograr un estado del arte actualizado sobre los algoritmos criptográficos seleccionados.
2. Programar los diferentes dispositivos que integran el proyecto.
3. Generar la documentación del trabajo realizado.

A. *Breve descripción del proyecto.*

En la seguridad de la información juega un papel clave la criptografía, que logra cifrar la información y hacerla inasequible para aquellos que no tengan las claves de cifrado y descifrado.

Pero aunque la seguridad que ofrecen los algoritmos criptográficos es muy alta, el factor humano y el uso habitual que de la criptografía hace el usuario final, es el talón de Aquiles de estos sistemas. Con frecuencia los usuarios guardan de forma poco o nada segura información sensible en sus discos duros.

Sería deseable disponer de un sistema de entrada y salida de datos al ordenador que, de forma transparente al usuario, cifrase la información después de introducirla por teclado y antes de su ingreso en la memoria del mismo, y lo descifrara a la salida por pantalla.

Los dispositivos de cifrado y descifrado se instalarían en la conexión del teclado con el ordenador y en la conexión del ordenador con el dispositivo de visualización del texto. En el ordenador jamás entraría la información en forma de texto original sin cifrar. Y jamás se almacenaría esa información en soporte alguno que no fuera de forma cifrada. En la Figura 1 se recoge un esquema del sistema a implementar.

El trabajo a desarrollar consistirá en la implementación de un algoritmo criptográfico a elegir entre las siguientes propuestas:

1. Algoritmo simétrico de cifrado por bloques: DES [4], AES [5], Skipjack, RC4, RC5, RC6, AES, Twofish (Para todos estos algoritmos ver [3]).
2. Algoritmo simétrico de cifrado por flujos: LFSR [3] § 6.2.

El objetivo final del trabajo es conectar un dispositivo lógico programable con el algoritmo de cifrado en el camino de conexión entre teclado y ordenador, y el dispositivo con el algoritmo de descifrado en el camino de conexión entre el ordenador y el dispositivo de visualización.

Otros algoritmos que se podrían implementar, para tareas de autenticación en caso de que se desee enviar la información por la red, son las funciones hash. Por ejemplo, si se desea enviar el documento resumido ("hasheado"), el ordenador no puede realizar esta tarea porque no dispone del texto original sin cifrar: debería hacerlo una tercera FPGA instalada más allá de la que descifra la información enviada.

Se puede elegir entre uno de estos dos algoritmos de resumen o hash:

1. Función Hash. SHA-1, SHA160. [6]
2. Hash obtenido a partir de la implementación de un algoritmo compresor creado a su vez a partir del algoritmo de cifrado simétrico (cfr. [11], capítulo 11.)

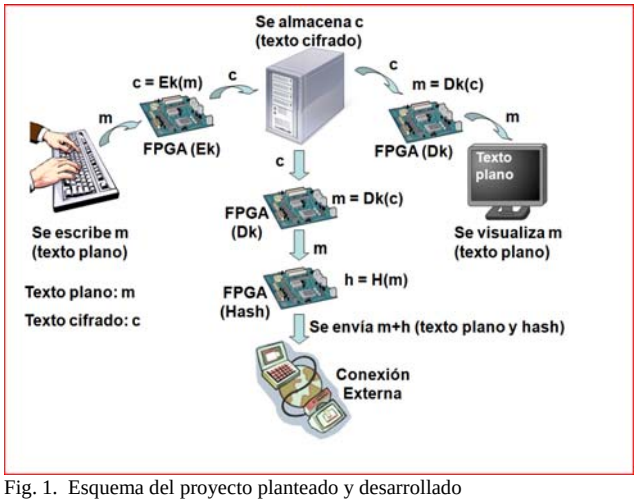

Fig. 1. Esquema del proyecto planteado y desarrollado

Cada alumno debe implementar y dejar funcional uno de los algoritmos propuestos: cifrado y descifrado, o función hash.

Los alumnos deberán defender oralmente el trabajo realizado.

B. Resultados de los proyectos terminados hasta la fecha.

A lo largo de estas líneas se describe uno de los proyectos mencionados anteriormente, en el que se lleva a cabo la implementación de una función hash. El trabajo ha sido realizado por dos alumnos en equipo. De esta forma se posibilita que trabajen en dos implementaciones diferentes. En una se utiliza un enfoque software en el que se lleva a cabo una implementación utilizando el lenguaje de programación C. Posteriormente, y casi en paralelo, se lleva a cabo una implementación hardware del mismo algoritmo utilizando el lenguaje de descripción de hardware VHDL.

Introducción al algoritmo hash. Una función hash (H) es una transformación que toma una entrada de bits de tamaño variable (m) y genera un salida de bits de un tamaño predeterminado, denominado valor hash (h), es decir $h = H(m)$. Las funciones hash con sólo esta propiedad tienen una gran variedad de usos en computación. Los requerimientos básicos para una función hash son los siguientes: (1) la entrada puede ser de cualquier longitud. (2) la salida presenta una longitud fija. (3) $H(m)$ es relativamente fácil de calcular para cualquier valor m. (4) H es una función de una vía, esto es, a partir del valor de h (salida de la función hash) no es posible obtener el vector de entrada. (5) H es una función libre de colisiones, lo que significa que no es posible encontrar dos entradas distintas con idéntico resumen hash h.

Ejemplos bien conocidos de transformaciones hash son MD2, MD5 y SHA. En este caso se ha optado por la implementación del algoritmo SHA, siguiendo los pasos descritos más profundamente en [3], algoritmo 9.53.

Implementación del algoritmo utilizando el lenguaje de programación C. En primer lugar, se ha implementado el algoritmo hash utilizando el lenguaje de programación C, diseñado en Bloodshed Dev-C++. Para la simulación se utilizó un mensaje de 3825 caracteres, lo que suponía una cantidad de 60 bloques. El primer paso a desarrollar por los alumnos a la hora de efectuar el diseño de la versión en C fue proponer un diagrama de flujo del algoritmo. Con ese diagrama, validado por los tutores del proyecto, los alumnos lograron definir el proceso en pequeños módulos de programación, que implementaron tanto en C como en VHDL.

Una vez implementado el algoritmo en C, se evaluó su rendimiento. Para ello, los alumnos utilizaron la herramienta *Performance Application Programming Interface* (PAPI). [14] Se trata de una API que permite un acceso a los contadores de rendimiento presentes en la mayor parte de microprocesadores comerciales, tales como Intel Pentium, AMD Opteron, etc. PAPI permite a los diseñadores de software ver, en tiempo casi real, la relación entre el rendimiento del software y los eventos del procesador.

En el caso que nos ocupa, se llevaron varias pruebas con una entrada de texto y se obtuvo una media de tiempo de cálculo para la implementación C del algoritmo hash de 4.51μs.

Implementación hardware del algoritmo utilizando el lenguaje de descripción hardware VHDL. Con este trabajo previo, los alumnos logran un conocimiento amplio del funcionamiento de la función SHA-1 y de su modularización. De esta forma, pudieron afrontar con éxito su implementación hardware.

El principal objetivo de esta segunda fase del trabajo era la implementación del algoritmo sobre un dispositivo FPGA de la familia Spartan 3 de Xilinx (XC3S200). Para ello se utilizó la tarjeta Spartan 3 Starter Board comercializada por la empresa Digilent [15]. Como entornos de programación y verificación se utilizaron el ISE Foundation de Xilinx para la implementación y síntesis y el ModelSim para la verificación y simulación.

La implementación hardware se puede considerar compuesta de un contador que obtiene los diferentes bloques de 512 bits (Mi) que se introducen en un subsistema ("*Hash Calculator*"), dedicado al cálculo y reproducción de los diferentes pasos del algoritmo hash.

A su vez, el subsistema "*Hash Calculator*" se implementa de acuerdo con la estructura modular definida previamente en el diagrama de flujo. De esta forma, los alumnos ponen en práctica sus conocimientos de diseño y usan las herramientas adecuadas para la simulación y verificación de todo lo desarrollado.

Una vez implementados, simulados y verificados los bloques constitutivos de la implementación hardware, es posible obtener medidas de rendimiento con objeto de llevar a cabo una comparativa con los resultados obtenidos de la versión software.

La herramienta de síntesis ofrece unos valores para la implementación hardware en los que el cálculo de la función hash dura 83 ciclos de reloj, con una frecuencia máxima de 75MHz. A partir de este se obtiene un tiempo de cálculo de 1.10μs para esta nueva implementación hardware, lo que supone una aceleración de un factor de 4 respecto a su equivalente software.

VI. RESULTADOS Y CONCLUSIONES

Una vez cursada la asignatura, y realizados todos los trabajos propuestos, los alumnos han desarrollado las siguientes competencias básicas:

A. Competencias específicas.

Los alumnos se han enfrentado a una nueva asignatura con contenidos nuevos y, aunque brevemente, han podido adquirir un conocimiento básico suficiente para poder continuar un estudio personalizado.

B. Competencias genéricas. Competencias Instrumentales.

Los alumnos han tenido que llevar a cabo un esfuerzo de síntesis reflejado en los cuestionarios anteriormente comentados; un esfuerzo de planificación y organización de su tiempo para la entrega de los sucesivos trabajos. Han

adquirido habilidades básicas de manejo de entornos de desarrollo utilizados en sectores industriales. Han tenido que decidir sobre distintas posibilidades algorítmicas a la hora de desarrollar los trabajos planteados. Han profundizado y afianzado sus conocimientos de programación, con objeto de llevar a cabo una implementación de un código criptográfico. Han incrementado sus conocimientos del lenguaje de descripción hardware VHDL con objeto de migrar la implementación software a un módulo hardware.

C. Competencias genéricas. Competencias Sistémicas.

Han trabajado con documentación científica, toda ella publicada en inglés. Han aportado perspectivas originales de diseño y programación que se ha reflejado en la calidad y progresiva mejora de los trabajos presentados. La planificación de las prácticas, en las que los resultados de cada una de ellas han sido necesarios para la correcta solución de las siguientes, ha exigido a los alumnos cuidar la calidad de su desarrollo software. También les ha permitido adquirir una visión amplia de su trabajo como desarrolladores, necesaria para gestionar la implementación de los trabajos con un enfoque modular. Han verificado las diferentes fases del flujo de diseño hardware sobre un sistema real, y han comprobado que al final los resultados obtenidos mejoran las prestaciones del desarrollo inicialmente logrado.

REFERENCIAS

[1] M.A. Zabalza. "La enseñanza universitaria. El escenario y sus protagonistas". Ed. Nancea, 2002

[2] www.teleco.upct.es/posgrados/tic/index.php.

[3] "Handbook of Applied Cryptography". A. Menezes, P. van Oorschot, and S. Vanstone. CRC Press, Inc. 1997.

[4] FIPS PUB 46–3, 1999 October 25. "Data Encryption Standard (DES)"

[5] FIPS PUB 197, 2001 November 26. "Annoucing the Advanced Encryption Standard (AES)".

[6] FIPS PUB 180–1, 1993 May 11. "Secure Hash Standard".

[7] Diffie, W., and Hellman, M. "New directions in cryptography". *IEEE Trans. Inform. Theory IT-22*, 6 (Nov. 1976), 644–654

[8] Rivest, R., Shamir, A., and Adleman, L. "A method for obtaining digital signatures and public key cryptosystems", Comm, ACM 21 (1978), 120–126.

[9] "Cryptography and Network Security. Principles and practices". William Stallings. Prentice Hall. Pearson Education. Third edition. 2003.

[10] "Introduction of Cryptography with coding theory". Wade Trappe and Lawrence C. Washington. Prentice Hall, 2002.

[11] "Introduction to Cryptography". Johannes A. Buchmann. Springer Verlag, 2004. Second Edition.

[12] Pedro Alcover, José Manuel García, Luis Hernández. "Diseño de un nuevo generador de secuencias de bits aleatorios por entrada de teclado". Novática, n. 174, marzo–abril 2005. Pg 59 – 65.

[13] Pedro Alcover, Disponible en <http://ocw.bib.upct.es/course/view.php?id=30&topic=1>, abril, 2009.

[14] Disponible en <http://icl.cs.utk.edu/papi/>, abril, 2009.

[15] Disponible en <http://www.digilentinc.com>, abril, 2009.

Pedro María Alcover Garau. Nació en Palma de Mallorca (España), el 6 de agosto de 1964. Licenciado en Ciencias Físicas por la Universidad de Valencia (España) el año 1987, y Doctor en Informática por la Universidad de Murcia (España) el año 2004.

Actualmente, y desde el año 1999, es profesor en el área de Lenguajes y Sistemas Informáticos en la Universidad Politécnica de Cartagena (España). Sus principales líneas de investigación son la criptografía y los patrones de diseño.

Juan Suardíaz Muro (M'02). Nació en Madrid (España), el 28 de julio de 1971. Ingeniero Industrial por la Universidad Politécnica de Madrid (España) el año 1997, y Doctor Ingeniero Industrial por la Universidad Politécnica de Cartagena (España) el año 2001.

Actualmente, y desde el año 2001, es profesor en el área de Tecnología Electrónica en la Universidad Politécnica de Cartagena (España). Sus principales líneas de investigación son el diseño digital basado en dispositivos lógicos reconfigurables (FPGA's) y el desarrollo de sistemas electrónicos de control.

Pedro Javier Navarro Lorente. Nació en Cartagena (España), el 26 de marzo de 1972. Ingeniero Industrial por la Universidad Politécnica de Cartagena (España), el año 1999.

Actualmente, y desde el año 2004 es Profesor en el área Lenguajes y Sistemas Informáticos en la Universidad Politécnica de Cartagena (España). Sus principales líneas de investigación son los algoritmos de visión artificial.