UNIVERSIDAD DE GRANADA

TRABAJO FIN DE GRADO

DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS

Implementación de una blockchain resistente a ataques criptográficos post-cuánticos

Subtitulo del Proyecto

Autor
María Victoria Granados Pozo

Director
Gabriel Maciá Fernández
Francisco Javier Lobillo Borrero

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Facultad de Ciencias

—

Granada, mes de 201

Implementación de una blockchain resistente a ataques criptográficos post-cuánticos

Subtítulo del proyecto.

Autor
María Victoria Granados Pozo

Director
Gabriel Maciá Fernández
Francisco Javier Lobillo Borrero

Implementación de una blockchain resistente a ataques criptográficos post-cuánticos: Subtítulo del proyecto

María Victoria Granados Pozo

Palabras clave: palabra_clave1, palabra_clave2, palabra_clave3, ...

Resumen

Poner aquí el resumen.

Implementation of a blockchain resistant to post-quantum cryptographic attacks: Project Subtitle

María Victoria Granados Pozo

Keywords: Keyword1, Keyword2, Keyword3, ...

Abstract

Write here the abstract in English.

Yo, **María Victoria Granados Pozo**, alumno de la titulación Doble Grado de Ingeniería Informática y Matemáticas de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación y Facultad de Ciencias de la Universidad de Granada**, con DNI 77137043, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: María Victoria Granados Pozo

Granada a X de mes de 201 .

D. **Gabriel Maciá Fernández**, Profesor del Área de Ingeniería Telemática del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

D. **Francisco Javier Lobillo Borrero**, Profesor del Área de XXXX del Departamento YYYY de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***Implementación de una blockchain resistente a ataques criptográficos post-cuánticos, Subtítulo del proyecto***, ha sido realizado bajo su supervisión por **María Victoria Granados Pozo**, y autoriza la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a X de mes de 201 .

El director:

Gabriel Maciá Fernández **Francisco Javier Lobillo Borrero**

Agradecimientos

Poner aquí agradecimientos...

Índice general

Índice de figuras

Índice de tablas

Listados de código

Guía de estilo para escribir un TFG/TFM

Este capítulo no forma parte del TFG/TFM. Su único objetivo es aportar algunas recomendaciones y plantillas para tener claro cómo redactar el TFG/TFM. Una vez se haya comprendido, se puede comentar la siguiente línea en el fichero proyecto.tex añadiéndole al principio el carácter %:

```
\input{guiaDeEstilo} --> %\input{guiaDeEstilo}
```

Recomendaciones generales

A la hora de escribir el TFG/TFM es importante seguir las siguientes recomendaciones:

1. La memoria debe realizarse con el **máximo cuidado**, y debe proporcionar de forma consistente -y por sí misma- una idea clara y concisa de lo que se ha realizado.

2. No debe tener errores tipográficos ni ortográficos. Este es un aspecto que penaliza muchísimo el trabajo en la evaluación del tribunal.

3. Siempre que se utilice alguna figura no elaborada por el autor del proyecto debe indicarse la fuente de la que se ha sacado mediante una cita en la bibliografía.

4. La lectura debe ser fluida. Por ello, dada la dificultad que tiene afrontar la escritura de un texto largo casi por primera vez, se recomienda elaborar un índice rellenando los títulos de los diferentes apartados de que constará este documento. En segundo lugar, para cada apartado, se indicarán a modo de resumen las diferentes ideas que se desarrollarán posteriormente (una línea de texto por idea). Después, se desarrollan las ideas (cada idea en un párrafo). Cuando se termina, se realiza una lectura completa y detallada del texto para comprobar que es coherente y no tiene fallos ortográficos, tipográficos ni gramaticales, antes de pasarlo al tutor.

5. Una extensión normal está entorno a las 100-120 páginas. Esto no quiere decir que tengamos que escribir por escribir, ni meter contenido adicional sin sentido. Hay que escribir el proyecto de forma coherente, pero sin ser telegráfico, esto es, realizando una descripción detallada del trabajo realizado.

6. Evitar afirmaciones del tipo "El sistema diseñado es bastante bueno". Esa misma frase debería ser escrita tal que responda a las preguntas: ¿Qué parte del sistema? ¿En qué sentido? ¿Cuánto de bueno? ¿Comparado con qué?

7. Evitar la primera persona (incluso del plural). No obstante para resaltar la autoría de algo o enfatizar una posición personal sí se puede usar.

8. Numerar estructuradamente los capítulos, secciones y subsecciones. Evitar más de tres niveles de anidamiento.

9. Toda afirmación categórica o se demuestra (teórica o experimentalmente) o se incluye una referencia en la que se haya previamente demostrado.

10. Toda tecnología, teorema, institución, norma, documento que se mencione debe estar referenciado. No incluir referencias a la wiki.

11. Los términos en ingles que no tenga sentido traducir se pondrán en cursiva al menos para indicar que es un término no castellano.

Recomendaciones específicas para determinados contenidos

Inserción de figuras

Esta es una plantilla de código para adjuntar una figura.

```
\begin{figure}[t]
\centering
\includegraphics[width=0.6\textwidth]{figuras/prueba.eps}
\caption{Pie de figura. Poner aquí cita del lugar de donde
se ha tomado la imagen en caso de que sea así. }
\label{fig:prueba}
\end{figure}
```

Si se pone el modificador [t] (top) latex ubicará la figura en la parte de arriba de la página. Ver otros modificadores como [h] (here) o [b] (bottom).

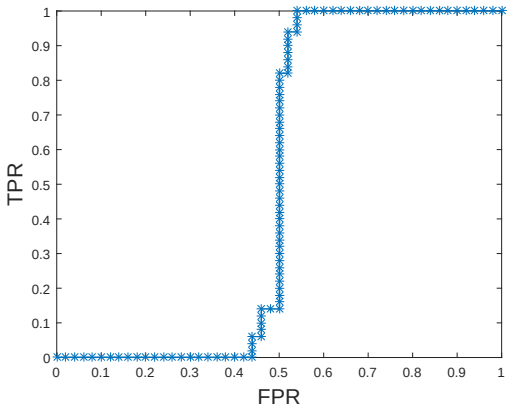

Figura 1: Pie de figura. Poner aquí cita del lugar de donde se ha tomado la imagen en caso de que sea así.

Tabla 1: Descripción de la tabla.

Tipo de ataque	**Etiqueta**
DoS11	dos
Exf1MBp	exf1KB

Se pueden usar otras plantillas para, por ejemplo, poner dos figuras una al lado de otra. Consultar en Internet diferentes plantillas en caso de necesidad.

Cuando en el texto nos refiramos a la figura en cuestión por el número, debemos usar la mayúscula y utilizar referencia a la figura. Esto hará que no nos tengamos que preocupar de la numeración de las figuras. Ej. Como se puede comprobar en la Figura 1.

Sustituir expresiones del tipo: "En la siguiente figura..." por "En la Figura 2.2..."

Inserción de tablas

Este sería un ejemplo de una tabla. Se puede modificar el formato y contenido (ver en Internet algún enlace sobre cómo formatear tablas en latex).

La forma de referirse a las tablas es similar a las de las figuras (usar mayúsculas y referencia a la etiqueta(label) de la tabla). Ej. Como se puede ver en la Tabla 1, ...

Citas de bibliografía

Ejemplo de cita de bibliografía. Primero se va a google.scholar y se busca la referencia. Después se da al enlace citar, y se elige el formato bibtex. Se copia ese texto en el fichero bibliografia.bib. Un ejemplo de referenciar una cita es [**?**].

Referencias a secciones

Para referirnos a secciones, primero debemos tener una etiqueta de tipo `label` en dicha sección. Posteriormente, pondremos una referencia a dicho label, igual que hacemos para las figuras y las tablas. Ej. Como se ha mencionado en la Sección 1.1 (Nótese que la palabra Sección va con mayúscula).

Glosario y acrónimos

Cuando se utilice un acrónimo se debe definir en el fichero glosario/entradas_glosario, tal y como está el ejemplo en dicho fichero. Al referirse en el texto se indicará así: support vector machine (SVM) (ver que la primera vez lo pondrá completo). La segunda vez que se referencie a SVM ya no aparece completo. También se puede nombrar en plural así: SVMs. Otros ejemplos de acrónimo son: Greatest Common Divisor (GCD), Least Common Multiple (LCM), Gabriel Maciá Fernández (GMF).

A la hora de compilar con el glosario, se debe abrir una terminal CMD en el directorio de los fuentes latex del proyecto, y ejecutar el siguiente comando: `makeglossaries proyecto`. Esto generará los ficheros auxiliares que contienen el glosario.

Listados de código

Aquí se puede ver un ejemplo de listado de código:

```
1  import numpy as np
2
3  def incmatrix(genl1,genl2):
4    m = len(genl1)
5    n = len(genl2)
6    M = None #to become the incidence matrix
7    VT = np.zeros((n*m,1), int)  #dummy variable
8
9    #compute the bitwise xor matrix
10   M1 = bitxormatrix(genl1)
11   M2 = np.triu(bitxormatrix(genl2),1)
12
13   for i in range(m-1):
14     for j in range(i+1, m):
15       [r,c] = np.where(M2 == M1[i,j])
16       for k in range(len(r)):
17         VT[(i)*n + r[k]] = 1;
18         VT[(i)*n + c[k]] = 1;
19         VT[(j)*n + r[k]] = 1;
20         VT[(j)*n + c[k]] = 1;
```

```
21
22    if M is None:
23      M = np.copy(VT)
24    else:
25      M = np.concatenate((M, VT), 1)
26
27    VT = np.zeros((n*m,1), int)
28
29    return M
```

Listado de código 1: Ejemplo de Python

Nos podemos referir a él como Listado de código 1. Si queremos que aparezca como un flotante en la página debemos poner la palabra `float` así:

```
\begin{lstlisting}[float,language=Python,caption=Ejemplo de Python,
label=listado:pythonPrueba]
```

Enlaces URL

Podemos poner un enlace así `http://dtstc.ugr.es/~gmacia`

Capítulo 1

Introducción

El objetivo de este proyecto es la implementación de un algoritmo criptográfico resistente a ordenadores cuánticos, denominado LUOV, para la firma de documentos. Con ayuda de este algoritmo modificaremos la implentación de la *blockchain* ARK.

1.1. Motivación y contexto del proyecto

A lo largo del desarrollo de este trabajo tendremos presentes dos dos tecnologías, la computación cuántica y las cadenas de bloques.

La computación cuántica constituye un nuevo paradigma de la informática basado en los principio de la teoría cuántica. La computación clásica funciona con bits cuyos valores pueden ser 0 o 1, mientras que la computación cuántica funciona con bit cuánticos o cúbits, donde son una combinación de 0 y 1, pudiendo tomar ambos valores a la vez, esto se denomina la superposición cuántica de los estados.

La superposición cuántica aporta gran capacidad de procesamiento, lo que hace posible resolver de manera eficiente problemas de mayor complejidad como la factorización de enteros, el algoritmo discreto y la simulación cuántica, que a día de hoy con los ordenadores clásicos son difíciles de romper.

La evolución de la tecnología se ha basado principalmente en la reducción de los transistores para aumentar la velocidad, llegando a escalas de tan solo algunas decenas de nanómetros. Esto tiene un límite y es la eficiencia, puesto que al seguir disminuyendo el tamaño podrían dejar de funcionar correctamente. De ahí surge la necesidad de descubrir nuevas tecnologías, la computación cuántica.

El estudio de las tecnologías cuánticas se inició en 1980, donde surgieron teorías con la posibilidad de realizar cálculos cuánticos. En la década de los 90 se empezó a poner en práctica algunas teoría, apareciendo los primeros algoritmos cuánticos, primeras aplicaciones cuánticas y las primera máquinas

diseñadas para realizar cálculos cuánticos.

La otra tecnología que vamos a utilizar son las cadenas de bloques, *blockchain*. Esta tecnología permite verificar, validar, rastrear todo tipo de información, como contratos inteligentes, transacciones financieras, certificados digitales y firmas, que será el centro de este proyecto.

Los datos que almadenan cada bloque almacena son transacciones válidas, información referente a ese bloque y la relación con el bloque anterior mediante el *hash*, por tanto el bloque tiene un lugar específico dentro de la cadena. De esta forma si hay una alteración en un determinado bloque se verá reflejado en su *hash* y en el de los bloques posteriores, haciendo que la cadena que la información de la cadena no se pueda perder, modificar o eliminar.

Las cadenas de bloques se pueden aplicar en diversos ámbitos, como en la salud, que podrían tener el historial médico de un paciente en cualquier centro de salud, de una forma segura y evitando falsificaciones. También se puede aplicar en la firma de documentos o transacciones por parte de un notario, que a día de hoy es un problema ya que las firmas digitales son fáciles de copiar, pero con *blockchain* no podrían ser falsificadas.

1.2. Objetivos del proyecto y logros conseguidos

El objetivo de este proyecto es modificar el algoritmo de firma y verificación de las transacciones de la *blockchain* ARK, para hacerla resistente a atáques cuánticos.

- Implementación del algoritmo UOV: Se ha implementado las funciones de generación de claves tanto públicas como privadas, la función de firma a partir de la clave privada y la función de verificación de la misma con la clave pública. Además ha sido necesario implementar la aritmética de cuerpo finito de 2^7 elementos.

- Estudio de la *blockchain* ARK: Crear y probar la *blockchain* ARK como usuario final, también estudiar cuales son y donde se encuentran las funciones que queremos modificar.

- Crear un docker ARK: Crear un docker para almacenar la blockchain ARK.

- Modificar el algoritmo de firma y verificación: Cambiar el actual algoritmo que aplica la *blockchain* ARK por el algoritmo LUOV

- Ver las transacciones realizadas en el *explorer* de ark: Finalmente la prueba que se realizará es ver que las transacciones que realicemos con la API se muestran en el explorer de ARK.

1.3. Estructura de la memoria

Se describirán los capítulos que tiene la memoria, indicando qué contenidos habrá en cada uno de ellos, para permitir al lector situarse ante el documento.

1. Introducción: Breve explicación de la motivación de este proyecto y de los objtetivos planteados.

2. Planificación y costes: Definición de las entregas y seguimiento del proyecto. Además incluye el presupuesto del proyecto.

3. Análisis del problema: Descripción de las funcionalidades y requisitos, y análisis de los objetivos que se muestran en la sección 1.2.

4. Diseño:

5. Implementación

6. Evaluación y pruebas

7. Conclusiones: Aportaciones al proyecto

1.4. Contenidos teóricos para la comprensión del proyecto

A continuación se muestran algunos contenidos claves para la mejor comprensión del proyecto.

- Algoritmo cuántico: Son los algoritmos que pueden ser resueltos por un computador cuántico en tiempo polinómico.

- Contratos inteligentes: Base de "propiedades inteligentes"que permiten definir mediante códigos de la Blockchain, la forma en la que los dispositivos reaccionan ante eventos que tienen lugar en su entorno. Llevan incorporada una máquina virtual que habilita la codificación y ejecución de programas software para determinar las condiciones sobre el intercambio de activos entre agentes.

- Firma electrónica: Sirve para controlar la integridad de los datos y asegurar que la información procede de quien dice ser su remitente, garantiza que la información que se almacena o se envía no ha sido modificada. Controla la auditoría del documento. Se requieren criptosistemas aritméticos.

- Criptosistemas aritméticos: Cada usuario posee una clave pública y otra privada. El usuario cifrará y descifrará el mensaje con su llave privada y pública, además de la llave pública del usuario que descifrará o que haya cifrado el mensaje.

- Resumen o Hash: Es el resultado de aplicar una función que transforma un mensaje que longitud variable en uno de longitud fija, denominada función hash. Es el resultado de calcular el resto módulo n con n la longitud fija. Al aplicar la función hash a un fichero, si se modifica algún dato del mismo cambiará su hash y por tanto se sabrá si ha sido manipulado desde que se envió. Así podemos conseguir la integridad del mensaje

- *Blockchain*: Sistemas de almacenamiento de información que se divide en bloque de datos enlazados mediante los hash. A cada bloque se le asocia un hash a partir del bloque anterior, creando una lista enlazada, la búsqueda de información no es muy óptima si hay un número elevado de bloques. Para la búsqueda eficiente en *blockchain* se usan los árboles merkle. *Blockchain* por si mismo no solucionan los problemas de los sistemas de la información y la comunicación. Pero permiten impulsar modificaciones orientadas a crear soluciones más robustas, implicando conocer donde hay que usar las blockchain y cual es la infraestructura.

- Árboles Merkle: Árboles binarios con funciones hash, cada nodo tiene con máximo dos hijos, no hay ciclos. El cálculo de los hash de los padres se hace combinando los hash de los hijos. La integridad se obtiene inclueyendo en los bloques el valor del nodo raíz en lugar de añadir el valor de todos los datos protegidos por los bloques, reducimos además la información de la cabecera.

Capítulo 2

Planificación y costes

Definir claramente de acuerdo con el tutor los "paquetes de trabajo" (PTs), identificando claramente los entregables resultantes de cada uno de ellos. Esto definirá claramente los resultados del proyecto. Pueden usarse Diagramas de Gantt o cualquier herramienta o metodología siempre que facilite la visualización secuencial y dependencias entre los PT. En este mismo capítulo se incluir un presupuesto –ajustado en lo posible a la realidad- que incluya recursos humanos y materiales, así como cualquier dato que determine la viabilidad del proyecto.

Capítulo 3

Análisis del problema

3.1. Especificación de requisitos

Debe incluir una clara descripción de las funcionalidades que se esperan alcanzar, así como las restricciones o condicionantes que puedan determinar el diseño o solución adoptada. Tras eso deben especificarse claramente los requisitos.

Los requisitos pueden ser funcionales (e.g. La herramienta debe mostrar las medidas de la red en tiempo real), o no funcionales (e.g. se debe garantizar el acceso seguro a la herramienta; el rendimiento debe ser alto; el consumo de memoria debe ser bajo; etc.)

3.2. Análisis

El objetivo de este apartado es mostrar en diferentes subapartados los diferentes subproblemas que han aparecido al realizar el proyecto, describiendo las alternativas que se han considerado, y justificando las decisiones que se han adoptado. A veces, especialmente cuando los conceptos utilizados en este apartado son extensos, es necesario clarificarlos previamente en el Capítulo de *Introducción* (sección *Contenidos teóricos para la comprensión del proyecto*).

Capítulo 4

Diseño

Es uno de los capítulos más importantes. Debe explicar claramente la solución propuesta justificando la aproximación adoptada. Este capítulo, según el caso, es aconsejable que defina claramente la arquitectura del sistema propuesto, identificando los roles o partes o actores del sistema. Pueden emplearse metodologías basadas en diagramas de clases, paquetes, diagramas secuenciales, diagramas de relación, etc.

Si se ha diseñado una interfaz gráfica debe también describirse su estructura, dónde se mostrará la información, etc.

Capítulo 5

Implementación

Aquí se deben proporcionar los detalles de cómo se ha llevado a la práctica el diseño propuesto en el capítulo anterior. Deben identificarse claramente herramientas, tecnologías, equipamientos, etc. utilizados o necesarios para el buen funcionamiento de la solución. Se pueden describir los fragmentos de código más importantes, con el fin de clarificar la funcionalidad que proporcionan. En general este capítulo debe facilitar la reutilización de nuestra solución, por lo que debe estar bien documentada. Puede incluir un manual de uso.

Capítulo 6

Evaluación y pruebas

En este capítulo se debe proporcionar una medida objetiva de las bondades y beneficios de la solución propuesta, tanto en términos absolutos, como –en la medida de lo posible- comparándola con otras soluciones. Dependiendo del tipo de proyecto, debe incluir los resultados experimentales obtenidos al probar la solución; también puede incluir una tabla o diagrama de los costes reales del desarrollo, para así establecer conclusiones respecto a la planificación y costes estimados a priori. Finalmente, cuando se trata del desarrollo de una aplicación software, se pueden definir baterías de pruebas a realizar, de modo que en este capítulo se especificarán qué pruebas se han realizado, los resultados esperados y los resultados obtenidos.

Capítulo 7

Conclusiones

Capítulo en el que deben resumirse las principales aportaciones del trabajo realizado.

7.1. Valoración personal

Se puede incluir una valoración personal del proyecto (opcionalmente)

Siglas

GCD Greatest Common Divisor.

GMF Gabriel Maciá Fernández.

LCM Least Common Multiple.

SVM support vector machine.

Apéndice A

Manual de usuario