Implementación de una blockchain resistente a ataques criptográficos cuánticos

Trabajo Fin de Grado

Autor

María Victoria Granados Pozo

Directores

Gabriel Maciá Fernández

Francisco Javier Lobillo Borrero

Doble grado de Ingeniería Informática y Matemáticas
Universidad de Granada

26 de Noviembre de 2020

UNIVERSIDAD DE GRANADA

Introducción

Motivación

BLOCKCHAIN

ALGORITMO CRIPTOGRÁFICO UOV

@mvictoria1997/TFG
@mvictoria1997/core

Motivación

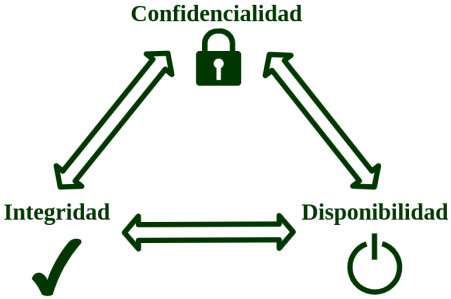


Figure: Pilares de la seguridad informática

Objetivos

Implementación del algoritmo UOV

Funciones propias del algoritmo y aritmética del cuerpo finito de 2^7 elementos.

Integración del algoritmo UOV

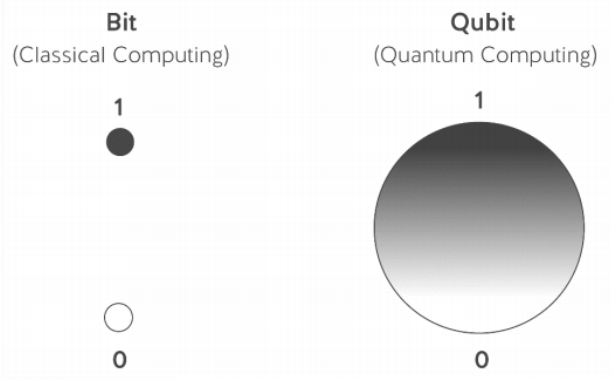
Modificación del algoritmo de firma de la blockchain de ARK por el algoritmo UOV.

Tecnologías utilizadas

Contenidos teóricos

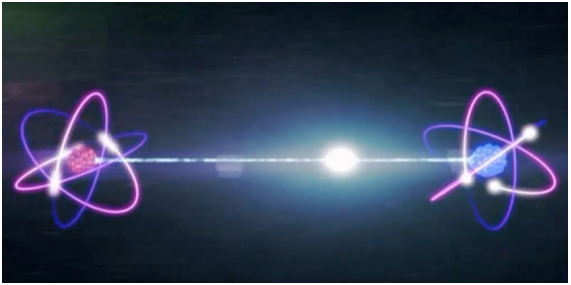
Computación cuántica

Estados de un bit y un cúbit



Propiedades computación cuántica

- Superposición cuántica.
- Entrelazamiento cuántico.
- Teletransporte cuántico.



Comparativa computación cuántica y clásica

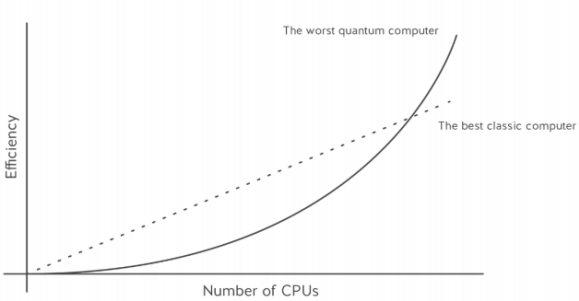


Figure: Comparativa de cómputo de de un ordenador cuántico y clásico

Blockchain

Descripción

Una cadena de bloques es un sistema de almacenamiento de información dividido en bloques de datos enlazados mediante el *hash*.

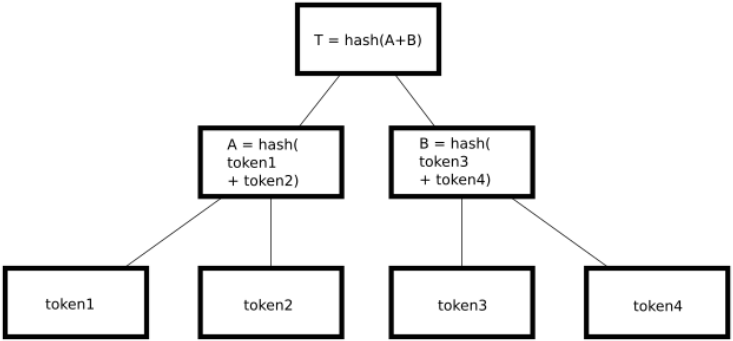


Figure: Estructura árbol de Merkle

Aplicaciones

- Área financera o criptomonedas.
- Centros de salud.
- Firma de documentos.
- Cadenas de suministro.

Algoritmo UOV *(Unbalance Oil and Vinegar)*

Ventajas del algoritmo UOV

- Problema NP-duro.
- No se conoce un algoritmo eficiente para la resolución de sistemas multivariados en un ordenador cuántico.
- Simplicidad de las operaciones.
- Requiere bajos recursos *hardware*.

Esquema UOV

$$\mathcal{P} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^m$$

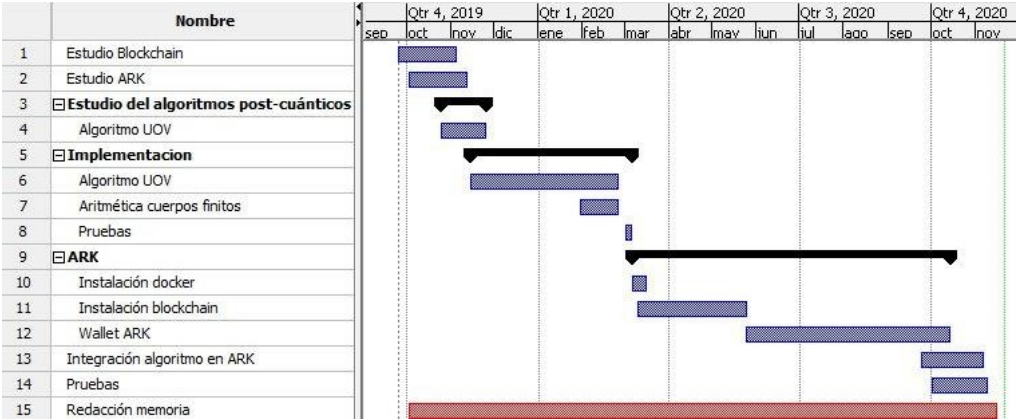
$\mathcal{P} = \mathcal{F} \circ \mathcal{T}$, donde $\mathcal{T} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^n$ y $\mathcal{F} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^m$

$$f_k(x) = \sum_{i=1}^{v} \sum_{j=i}^{n} \alpha_{i,j,k} x_i x_j + \sum_{i=1}^{n} \beta_{i,k} x_i \tag{1}$$

donde $\alpha_{i,j,k}$ y $\beta_{i,k}$ se toman aleatoriamente en $\mathbb{F}_2$ siendo $(\alpha_{i,j,k})_{\substack{1 \leqslant i \leqslant v \\ 1 \leqslant j \leqslant n}}$ un vector de matrices triangulares superiores.

Planificación y presupuesto

Diagrama de Gantt



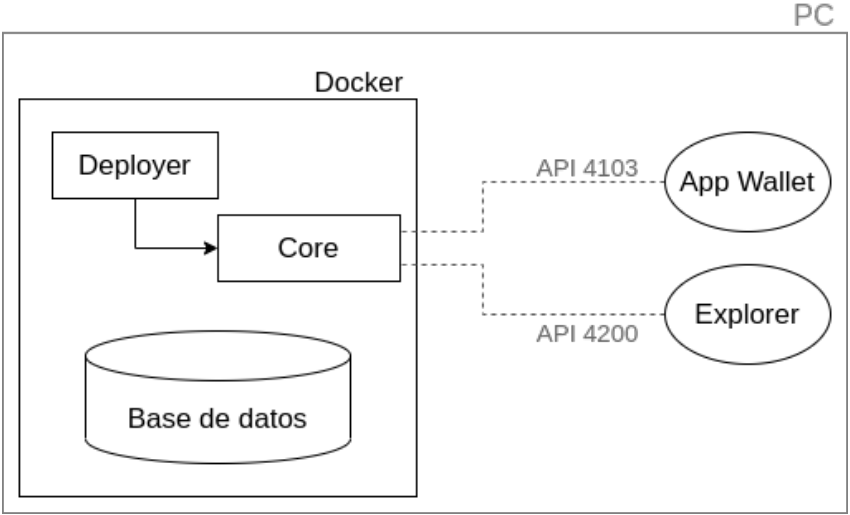
Presupuesto desglosado

Tipo de costes	**Cantidad**
Recursos humanos tutores	4.830€
Recursos humanos alumna	10.720€
Indirectos	1.578,24€
Directos	210,40€
Viajes	22€
Gastos imprevistos	868,03€
TOTAL (€)	18.228,67€

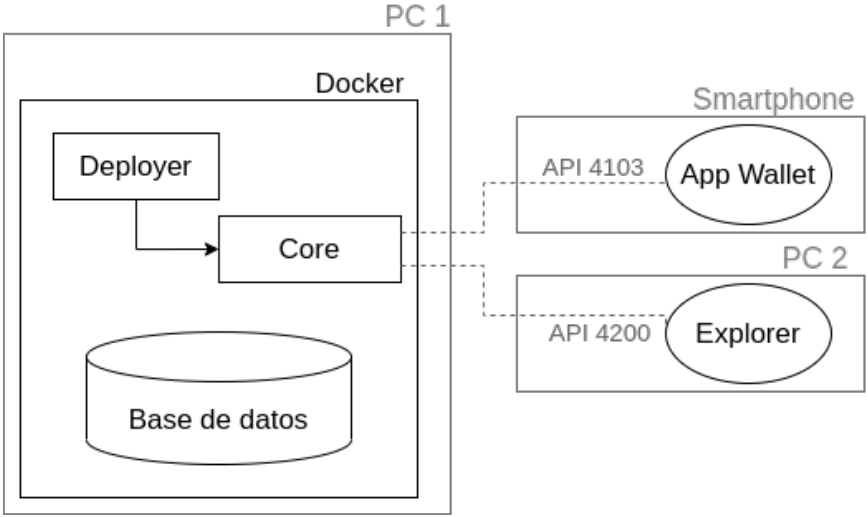
Table: Presupuesto total desglosado

Diseño

Configuración de los bloques



Otra posible configuración de los bloques



Implementación

Estructura directorio core-bridgechain/packages/crypto/src/crypto

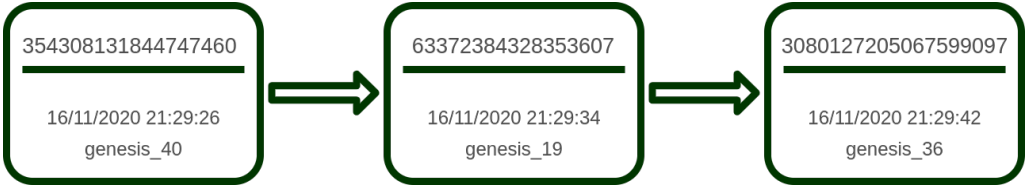
```
core-bridgechain/packages/crypto/src
└─ crypto
   ├─ hash.ts
   ├─ uov.py
   ├─ signature.py
   ├─ verify.py
   ├─ data.json
   └─ signature.json
```

Problemas encontrados

- Necesidad de implementar la aritmética del cuerpo finito de 128.
- Incompatibilidad de las claves de la *blockchain* de ARK con las del algoritmo UOV.
- La firma llega truncada la función de verificación.

Demostración práctica

Cadena de bloques



Bloque con ID 63372384328353607

63372384328353607

Hash: e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b34ca495991b7852b855

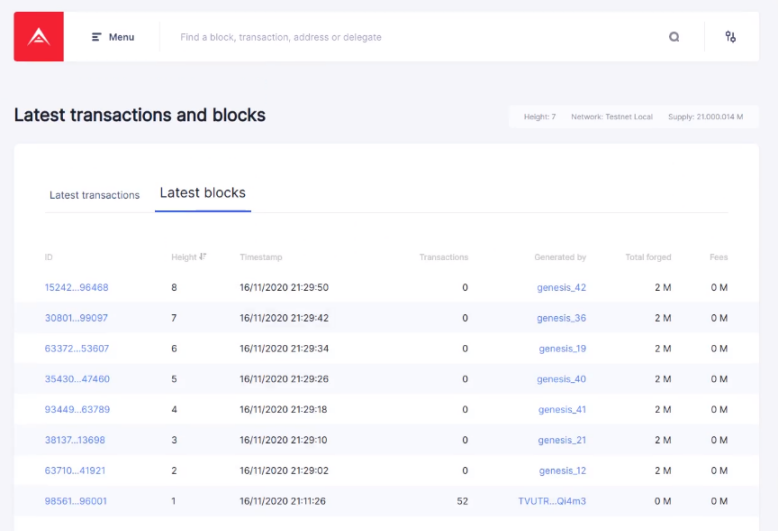
Firma Vector: [[1, 1, 1, 0, 1, 0, 1], [0, 1, 0, 1, 1, 1, 1], [1, 0, 0, 0, 0, 0, 1], [0, 0, 1, 1, 1, ...

Firma Hex: 5b5b312c20312c20312c20302c20312c20302c20312c205d2c205b302c2....

Logs terminal

```
2|bridgechain-forger | signature UOV
2|bridgechain-forger | python /home/deployer/core-bridgechain/packages/crypto/dist/crypto/../../src/crypto/signature.py 182,85,243,222,157,82,40,158,70,45,235,73,201,34,
239,216,202,182,62,76,120,58,108,176,141,155,20,225,79,192,7,184 02fbefc34e84ca97fa3f1393a7309a4e6964296ae0c91049de57d7365372c89295 96b6d0ee7372521079de79ef94a62f4c4143e
21e5c9e2b59ec2a6b0e696999a6
2|bridgechain-forger | [[1, 1, 1, 0, 1, 0, 1], [0, 1, 0, 1, 1, 1, 1], [1, 1, 0, 1, 1, 1, 1], [1, 0, 0, 0, 0, 0, 1], [0, 0, 1, 1, 1, 0, 0], [1, 0, 1, 0, 1, 1, 0]]
2|bridgechain-forger | 5b5b312c20312c20312c20302c20312c20302c20312c20302c20315d2c205b302c20312c20302c20312c20312c20312c20315d2c205b312c20312c20302c20312c20312c20315d2c205b312c
20302c20302c20302c20302c20302c20315d2c205b302c20302c20312c20312c20312c20302c20305d2c205b312c20302c20312c20302c20312c20312c20305d5d0a
2|bridgechain-forger | verifica UOv
2|bridgechain-forger | [[1,1,1,0,1,0,1],[0,1,0,1,1,1,1],[1,1,0,1,1,1,1],[1,0,0,0,0,0,1],[0,0,1,1,1,0,0],[1,0,1,0,1,1,0]]
2|bridgechain-forger | python /home/deployer/core-bridgechain/packages/crypto/dist/crypto/../../src/crypto/verify.py 182,85,243,222,157,82,40,158,70,45,235,73,201,34,239
,216,202,182,62,76,120,58,108,176,141,155,20,225,79,192,7,184 [[1,1,1,0,1,0,1],[0,1,0,1,1,1,1],[1,1,0,1,1,1,1],[1,0,0,0,0,0,1],[0,0,1,1,1,0,0],[1,0,1,0,1,1,0]] 02fbefc34
e84ca97fa3f1393a7309a4e6964296ae0c91049de57d7365372c89295
2|bridgechain-forger | True
2|bridgechain-forger | [2020-11-16 20:29:36.442] INFO : Forged new block 63372384328353607 by delegate genesis_19 (02fbefc34e84ca97fa3f1393a7309a4e6964296ae0c91049de57d7
365372c89295)
2|bridgechain-forger | [2020-11-16 20:29:36.444] DEBUG: Broadcasting block 6 (63372384328353607) with 0 transactions to 127.0.0.1
1|bridgechain-relay  | [2020-11-16 20:29:36.456] INFO : Received new block at height 6 with 0 transactions from 127.0.0.1
1|bridgechain-relay  | [2020-11-16 20:29:36.456] INFO : Previous block 5 pinged blockchain 0 times
1|bridgechain-relay  | [2020-11-16 20:29:36.460] DEBUG: event 'NEWBLOCK': "id18" -> "newBlock"
```

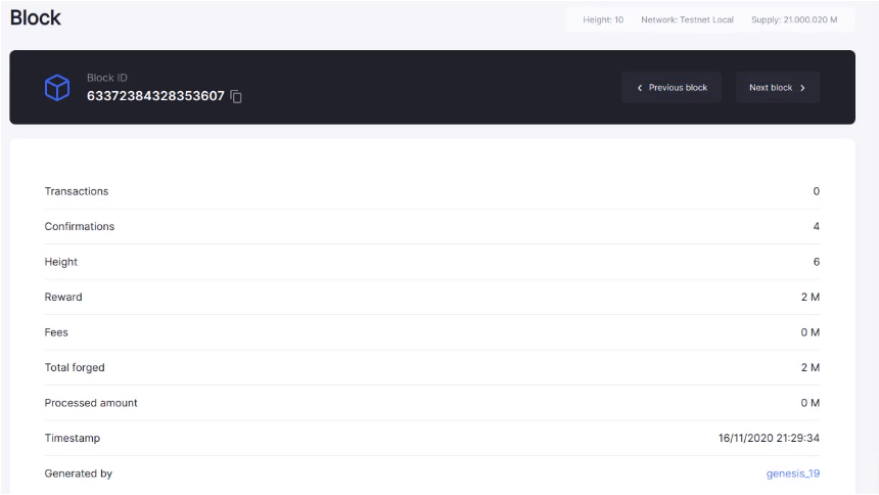
Visualización ARK Explorer



Visualización ARK Explorer ampliada

30801...99097	7	16/11/2020 21:29:42
63372...53607	6	16/11/2020 21:29:34
35430...47460	5	16/11/2020 21:29:26

Visualización ARK Explorer del bloque con ID 63372384328353607



Visualización ARK API del bloque con ID 63372384328353607


```
{"id":"63372384328353607","version":0,"height":6,"previous":"3543081318344747460","forged":
{"reward":"200000000","fee":"0","total":"200000000","amount":"0"},"payload":
{"hash":"e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855","length":0},"generator":
{"username":"genesis_19","address":"TFy6ASk5k5cohnrV32LAZRoBNiDqPVTXXe","publicKey":"02fbefc34e84ca97fa3f1393a7309a4e6964296ae0c91049de57d7365372c89295"},"signa
ture":"5b5b312c20312c20312c20302c20312c20302c20315d2c205b302c20312c20302c20312c20312c20312c20315d2c205b312c20312c20302c20312c20312c20312c20315d2c205b312c20302c2
0302c20302c20302c20302c20315d2c20","confirmations":6,"transactions":0,"timestamp":{"epoch":1088,"unix":1605558574,"human":"2020-11-16T20:29:34.125Z"}},
```

Conclusiones e investigaciones futuras

Conclusiones

- ✓ Implementación algoritmo UOV y aritmética del cuerpo finito de 128 elementos.
- ✓ Comparación de los tiempos de ejecución en `python` y `SageMath`.
- ✓ Integración del algoritmo en la *blockchain* de ARK.
- ✓ Ejecución de la *blockchain* de ARK modificada.
- ✓ Ver los bloques firmados en el *explorer* de ARK y en la API.
- ✓ Cadena de bloques más segura a costa de perder rendimiento.

Investigaciones futuras

- Trabajar con la base de datos.
- Integrar la *blockchain* ARK modificada en otra cadena de bloques.

¡Gracias por su atención!