



eryx

zkCity

Día 4: De oráculos a PCS

¿Qué vamos a ver?

- Commitments.
- KZG: chau oráculos.



El martes dijimos:

A partir de esto, alcanza con hacer la verificación...

$$q_L(z)[a]_Z + q_R(z)[b]_z + q_M(z)[a]_z[b]_z + q_O(z)[c]_z + q_C(z) = z_D(z)[t]_z$$

El martes dijimos:

A partir de esto, alcanza con hacer la verificación...

$$q_L(z)[a]_Z + q_R(z)[b]_z + q_M(z)[a]_z[b]_z + q_O(z)[c]_z + q_C(z) = z_D(z)[t]_z$$

Commit



Commitments

- Estoy resolviendo un problema de ajedrez con amigos.
- Los que lo resuelven en menos de 5 minutos ganan platita.



Commitments

Cuando alguien resuelve el problema, escribe la jugada en un papel y lo guarda en un sobre.



Commitments

Al final abrimos todos los sobres y revisamos quiénes anotaron la jugada correcta.



Propiedades

Los commitments sirven para comprometerse a algo, en un instante de tiempo.

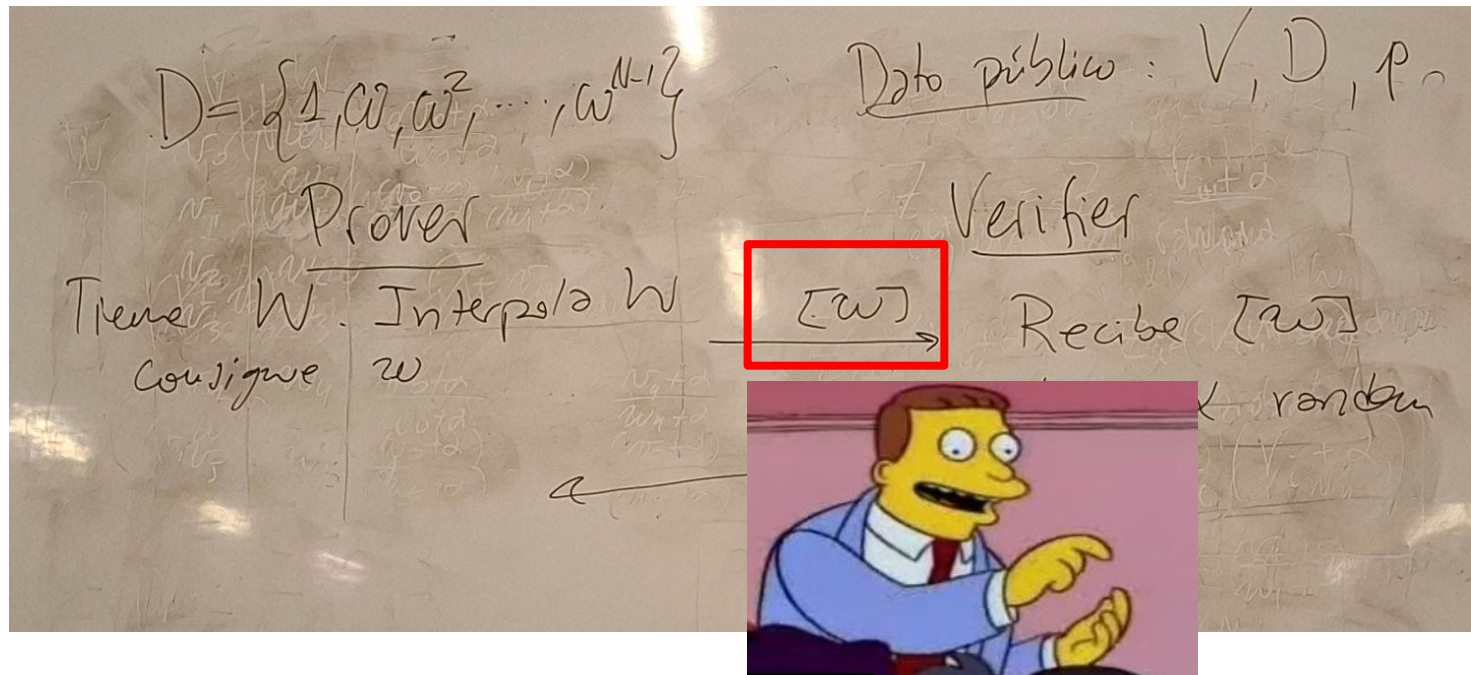
- **Binding**: me comprometo a **X** y ya no lo puedo cambiar.
- **Hiding**: no revela información sobre **X**.



Binding



Binding



Fases

- **Commit:**
 - Me comprometo a **X**.
- **Open:**
 - Revelo info sobre **X** y genero una prueba.
- **Verify:**
 - El resto verifica la prueba.

Ejemplo s



Hashes

Veamos un ejemplo con [hashes](#).



Hashes

Dado un hash H , el protocolo sería:

- **Commit:**
 - Comparto $[j] = H(j)$ para mi jugada j .
- **Open:**
 - Publico mi jugada j .
- **Verify:**
 - Mis amigos verifican $[j] = H(j)$.

Veamos si cumple las propiedades...

Binding

¿Puedo cambiar mi jugada después de commitear $[j]=H(j)$?

- Depende de cosas como si H es resistente a colisiones.

Hiding

¿Podría obtener j a partir de $[j]=H(j)$?

Hiding

¿Podría obtener j a partir de $[j]=H(j)$?

- Hasheo todas las jugadas posibles j' hasta encontrar $[j]$.



Mejora



Mejora

Una solución es agregar un número random (**SALT**), en algunos contextos llamado **blinding**.

- **Commit:**
 - Sampleo random **r** y comparto $[j] = H(j + r)$.
- **Open:**
 - Publico (j, r)
- **Verify:**
 - El resto verifica $[j] = H(j + r)$.



Código

```
In [1]: from hashlib import sha256
```

```
In [2]: H = sha256()
```

```
In [3]: H.update(b"Kf3" + b"37")
```

```
In [4]: H.hexdigest()
```

```
Out[4]: '3ce172ff082732f446c5696cde62208a9b61ef1ab5a3f2d0b4da7cc9a799995c'
```

Garantías de seguridad

Garantías de seguridad

	Si resuelvo DLP (u otro)	Incondicional
Binding		
Hiding		


Garantías de seguridad

	Si resuelvo DLP (u otro)	Incondicional
Binding	Podés “cambiar de opinión”.	
Hiding		

Garantías de seguridad

	Si resuelvo DLP (u otro)	Incondicional
Binding	Podés “cambiar de opinión”.	
Hiding	Podés “espiar en el sobre”.	

Garantías de seguridad

	Si resuelvo DLP (u otro)	Incondicional
Binding	Podés “cambiar de opinión”.	
Hiding	Podés “espiar en el sobre”.	

Otro ejemplo

Veamos otro esquema basado en un grupo de orden primo.

- **Commit:**
 - Comparto $[m] = g^m$.
- **Open:**
 - Publico m .
- **Verify:**
 - Verifican $[m] = g^m$.

Este esquema es unconditionally binding y computationally hiding.

Otro ejemplo

Veamos otro esquema basado en un grupo de orden primo.

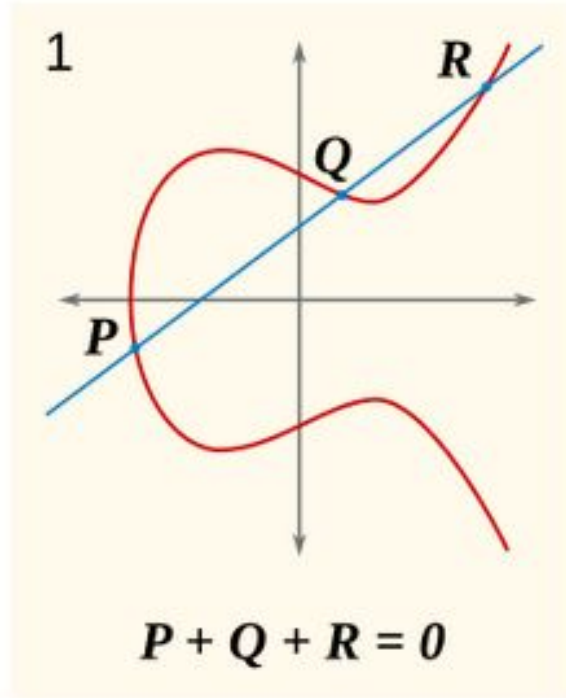
- **Commit:**
 - Comparto $[m] = g^m h^r$ para r random.
- **Open:**
 - Publico (m, r) .
- **Verify:**
 - Verifican $[m] = g^m h^r$

Este esquema es unconditionally hiding y computationally binding.

Garantías de seguridad

No existen commitment schemes “perfectos”.

Curvas elípticas



Curvas elípticas

Podemos crear una curva

```
[1]: F = GF(262147)
     E = EllipticCurve(F, (-1, 0))
```

Samplear puntos random

```
[2]: P = E.random_point()
     Q = E.random_point()
     P, Q
```

```
[2]: ((90328 : 254532 : 1), (29915 : 138117 : 1))
```

Curvas elípticas

Sumarlos:

```
[3]: P + Q
```

```
[3]: (122943 : 76558 : 1)
```

Sumar un punto con sí mismo k veces:

```
[4]: 100 * P
```

```
[4]: (57679 : 229693 : 1)
```


Curvas elípticas

Podemos hacer un commitment scheme:

- **Commit:**
 - Comparto $[m] = m G$.
- **Open:**
 - Publico m .
- **Verify:**
 - Verifican $[m] = m G$.

Encriptación homomórfica

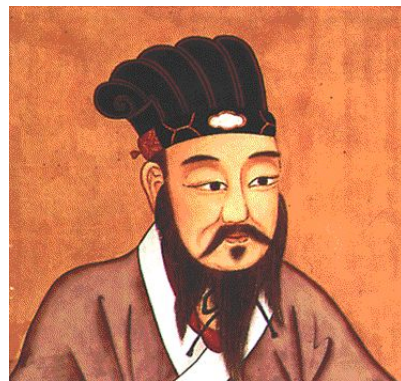
Los hashes no se suman, pero los puntos sí:

$$a G + b G = (a + b) G$$

$$[a] + [b] = [a + b]$$

“La suma de los commits es los commits de la suma”

Pero hay más...



Pairing

Es una operación **e**.

- Toma dos puntos de curva y devuelve un elemento de cuerpo.

Cumple que es bilineal:

$$e(a P, b Q) = e(P, Q)^{ab}$$

Hay curvas **pairing-friendly** y distintos **tipos** de pairing.

¿Para qué puede servir?

Vamos a tratar de usarlo



Ejemplo de juguete

Me quieren convencer de que resolvieron una ecuación.

$$x^2 = 10x + 8$$

Alguien conoce una solución r y me da $[r] = r \text{ G}$.

Puedo chequear:

$$e([r], [r]) = e(10[r] + 8 \text{ G}, \text{G})$$

Chequee que r cumplía la ecuación sin saber r . 🤖

Ejemplo de juguete

Prover:

```
p = X**2 - F(10) * X - F(8)
```

```
r = p.roots()[0][0]
```

```
R = r * G
```

Verifier:

```
pairing(R, R) == pairing(10 * R + 8 * G, G)
```

```
True
```

Más ejemplos

Restricción “sos un bit”:

$$x^2 - x = x(x - 1) = 0$$

Alguien conoce una solución a y me da $[a] = a \text{ G}$.

Puedo chequear:

$$e([a], [a]) = e([a], G)$$

Chequea x es un bit ¿sin verlo? 🤔

Más ejemplos

Restricción “sos un XOR”:

$$a - 2ab + b = c$$

Alguien conoce una solución a, b, c y me da $[a], [b], [c]$.

Puedo chequear:

$$e([a] + [b] - [c], G) = e(2[a], [b])$$

Chequea que a, b y c son un XOR.

Problemas

- Podría estar revelando **información**.
- Los pairings son muy **costosos**.
- Si chequeo ecuación por ecuación, no es **succinct**.

Pero hoy vamos a ver cómo se pueden usar de verdad!

Disclaimer

Disclaimer

```
[38]: pairing(P, Q)
```

```
[38]: 231308*i + 80030
```

Disclaimer



Disclaimer

```
[38]: pairing(P, Q)
```

```
[38]: 231308*i + 80030
```

Oráculo

Ya los venimos usando:

- Hay un polinomio p .
- Me dan un oráculo $[p]$.
- Sólo puedo preguntar $p(x_1)$, $p(x_2)$.
- Nunca miente y es mágico.



Opción 1

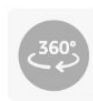
Hogar y Cocina > Decoración del Hogar > Decoraciones del Hogar > Accesorios Decorativos > Esferas Decorativas



Pasa el mouse encima de la imagen para aplicar zoom



2 VIDEOS



Seasons Bola de cristal mística con luces que cambian de color

Marca: Seasons

4.6 ★★★★★

403 calificaciones

Estilo: Bola de cristal

Bola de cristal

3 opciones desde US\$22.01

Hourglass

1 opción desde US\$29.99

Color	Multicolor
Marca	Seasons
Tema	Decoración para el hogar
Material	Cristal
Estilo	Bola de cristal

Sobre este artículo

- Decoración de bola de cristal de 7.5 pulgadas de alto
- La bola de vidrio cuenta con un aspecto único de vidrio agrietado con luces que cambian de color
- La bola está unida al soporte. Utiliza 3 pilas AAA (incluidas)
- Perfecto para cualquier escena o exhibición de Halloween
- Ideal para usar como accesorio de teatro

Informar de un problema con este producto o vendedor

Opción realista

Usar un polynomial commitment scheme (PCS).

- **Commit(p):**
 - Crea un commitment $[p]$ para un polinomio p .
- **Open(p, z):**
 - Calcula $y = p(z)$ y genera una prueba π .
- **Verify([p], π , z, y):**
 - Verifica que $y = p(z)$ usando π .

Revela información parcial sobre el polinomio.

The background is a dark blue gradient. It features several abstract geometric elements: a cluster of small blue squares in the top-left corner, a large blue circle on the left side, a large blue ring on the right side, and a blue gear-like shape at the bottom center. There are also some pixelated patterns in the bottom-right corner.

eryx

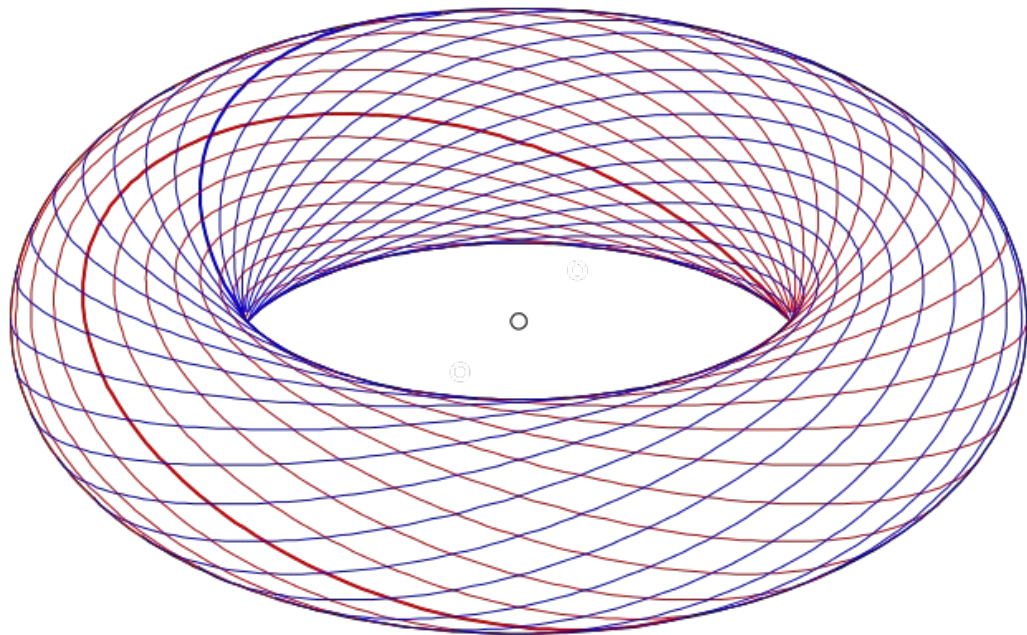
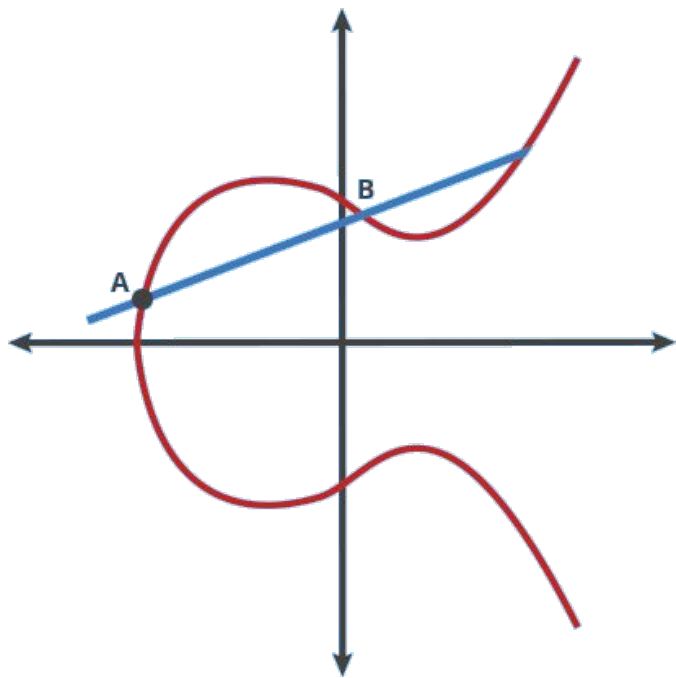
KZG



eryx

Commit

¿Por qué curvas elípticas?



¿Que me gustaría esconder?

(Tiene que estar relacionado al polinomio)



¿Que necesito para evaluar un polinomio?

Ahora puedo commitear.





eryx

Prove

**Me piden un z ,
quiero probar que**

$$p(z) = y$$

**Me piden un z ,
quiero probar que**

$$p(z) = y$$

quiero probar que

$$p(x) - y = (x - z) \cdot q(x)$$

**Me piden un z ,
quiero probar que**

$$p(z) = y$$

quiero probar que

$$p(x) - y = (x - z) \cdot q(x)$$

quiero probar que

$$p(\tau) - y = (\tau - z) \cdot q(\tau)$$

La prueba termina siendo

$$q(\tau)G$$



eryx

Verify

Quiero verificar que

$$p(\tau) - y = (\tau - z) \cdot q(\tau)$$

¿Que necesito?

**Aca entran en juego los Pairings.
:)**



**Cuando tenés que
trabajar con
curvas elípticas**

**Pero sabes lo que
es un pairing**



(En realidad sería algo como esto):

**Sabes lo que es un
pairing**



**La verificación termina siendo
chequear que**

$$e(p(\tau)G - yG, 1 \cdot G_2) = e(q(\tau)G, (\tau - z)G_2)$$



eryx

Batch opening

**Si me piden varios valores al mismo tiempo,
¿puedo hacer algo mejor?**





eryx

Cositas extra

Trusted setup

¿En qué parte se rompería el protocolo si alguien conociera τ ?

¿Y si el prover conoce τ ?

¿Y si el verifier conoce τ ?

¿Cuántas veces se debe hacer el setup?

Encriptación homomórfica

¿Que se puede hacer con el commitment $[p]$?

¿Y si tengo otro polinomio r comprometido?

¿Que puedo hacer con $[p]$ y $[r]$?