Trabajo Práctico. Circuitos Aritméticos en Blockchain

Elias Sebastian Gill Quintana
Johana Mabel Bareiro

"Análisis de un circuito aritmético para pruebas zk-SNARK"

Dr. Marcos Daniel Villagra Riquelme

Introducción

Este documento describe el diseño, implementación y uso de un circuito aritmético en el lenguaje Circom, utilizado para generar pruebas zk-SNARK en aplicaciones de blockchain. El circuito calcula la suma de los cuadrados de dos números privados y luego toma el módulo con respecto a un número primo p, asegurando que el resultado sea menor que p. Esta versión incluye mejoras significativas en la implementación del circuito.

El documento está estructurado de la siguiente manera:

(1) Descripción detallada del circuito.

(2) Proceso de generación de pruebas.

(3) Proceso de verificación de pruebas.

(4) Ejemplos de uso con valores concretos.

Estructura del Circuito

El circuito implementado en Circom sigue la siguiente lógica mejorada:

1. *Entradas*:
   - Dos entradas privadas: *a* y *b*.
   - Una entrada pública: *p* (un número primo).
2. *Salida*:
   - Una salida pública: *c*, que representa el residuo de la suma de los cuadrados de *a* y *b* módulo *p*.
3. *Novedades en la implementación*:
   - Uso de un componente *LessThan* independiente para verificar que el residuo sea menor que p.
   - Eliminación de variables intermedias redundantes.
   - Implementación más segura del cálculo del módulo.

El **código** del circuito en Circom es el siguiente:

```
pragma circom 2.0.0;

template LessThan(n) {
    signal input in[2];
    signal output out;

    signal diff <== in[1] - in[0] - 1;

    signal bits[n];
    var lc = 0;
    for (var i = 0; i < n; i++) {
        bits[i] <-- (diff >> i) & 1;
        bits[i] * (bits[i] - 1) === 0;
        lc += bits[i] * (1 << i);
    }
```

```
    lc === diff;
    out <== 1 - bits[n-1]; }

template Main() {
    signal input a;
    signal input b;
    signal input p;
    signal output c;

    signal sum <== a*a + b*b;
    c <-- sum % p;

    signal k;
    k <-- (sum - c) / p;
    sum === k * p + c;

    component lt = LessThan(252);
    lt.in[0] <== c;
    lt.in[1] <== p;
    lt.out === 1;

}

component main = Main();
```

Generación de Pruebas

El proceso de generación de pruebas con el circuito mejorado implica:

(1) Compilar el circuito en Circom para generar los archivos necesarios (r1cs, wasm, etc.).

(2) Realizar un *trusted setup* para generar las claves de prueba y verificación.

(3) Calcular el *witness* (testigo) utilizando las entradas proporcionadas.

(4) Generar la prueba zk-SNARK utilizando las claves y el *witness*.

Para el ejemplo concreto:

$$\{ \text{"a"}: 3, \text{"b"}: 4, \text{"p"}: 17 \}$$

El cálculo del residuo es:

$$c = (3^2 + 4^{2)} mod 17 = (9 + 16) mod 17 = 25 mod 17 = 8$$

Verificación de Pruebas

El proceso de verificación con el circuito mejorado:

(1) Usar la clave de verificación generada durante el *trusted setup*.

(2) Verificar que la prueba es válida para las entradas públicas y la salida.

En el ejemplo, la verificación confirmará que el residuo $c = 8$ es correcto y cumple $c < p$.

1. Ejemplos

Ejemplo completo con los valores proporcionados:

(1) Entradas:

$$\{ \text{"a"}: 3, \text{"b"}: 4, \text{"p"}: 17 \}$$

(2) Salida esperada:

```
{ "c": 8 }
```

Este ejemplo demuestra cómo el circuito mejorado calcula correctamente el residuo de la suma de los cuadrados módulo p.

Conclusión

Este documento ha presentado una versión mejorada de un circuito aritmético en Circom para calcular la suma de los cuadrados de dos números privados y tomar el módulo con respecto a un número primo. Las principales mejoras incluyen:

(1) Implementación segura de la comparación mediante *LessThan*.

(2) Reducción de complejidad y variables intermedias.

(3) Mayor claridad en las restricciones matemáticas.

Este tipo de circuitos mejorados son esenciales en aplicaciones de blockchain que requieren privacidad y verificabilidad.

Referencias

1. Circom documentation: https://docs.circom.io/

2. SnarkJS documentation: https://github.com/iden3/snarkjs

3. zk-SNARKs: https://z.cash/technology/zksnarks/