Báo cáo Kiểm thử

INT2208 5

Trần Thùy Dung

1 Giới thiệu

Báo cáo kiểm thử này mô tả quá trình kiểm thử và kết quả cho hàm thực hiện phép cộng giữa 2 điểm trên Elliptic Curve $y^2 = x^3 + 7 \bmod p = 10^9 + 7$. Mục tiêu của quá trình kiểm thử là để đảm bảo hàm này hoạt động chính xác cho nhiều dạng điểm khác nhau và các tổ hợp điểm đầu vào.

2 Cách tiếp cận

Những phương pháp sau được sử dụng để kiểm thử hàm cộng 2 điểm trên đường $y^2 = x^3 + 7 \bmod 10\texttt{e}9 + 7$:

- **Kiểm thử giá trị biên.** Kỹ thuật này được áp dụng để kiểm thử với các giá trị biên của đầu vào để kiểm tra đầu ra có chính xác hay không.

  Ta định nghĩa các giá trị `norm` và biên như sau:

  - $\texttt{norm} = \frac{1}{2}(p-1), f^{-1}(\frac{1}{2}(p-1)) = (500000003, 390770053)$
  - $\texttt{b}_{x0} = -\sqrt[3]{7} = -1.9129312$, $\texttt{b}_{x1} = 1\texttt{e}9 + 7$

- **Phân hoạch giá trị tương đương.** Kỹ thuật này được áp dụng để xác định các nhóm tổ hợp 2 điểm đầu vào và chọn đầu vào đại diện từ mỗi nhóm để kiểm tra.

  - **TH1.** Cộng một điểm khác điểm gốc với chính nó.
  - **TH2.** Cộng điểm gốc với chính nó.
  - **TH3.** Cộng hai điểm nghịch đảo (đường thẳng song song trục tung).
  - **TH4.** Cộng hai điểm khác tung độ thông thường.
  - **TH5.** Cộng một điểm với điểm vô cùng.
  - **TH6.** Đầu vào điểm không nằm trên đường cong.

Bộ ca kiểm thử được áp dụng với 2 phiên bản mã nguồn, bao gồm phiên bản gốc và phiên bản biến đổi có lỗi để kiểm tra chất lượng mã nguồn cũng như bộ ca kiểm thử.

3 Môi trường Kiểm thử

4 Kết quả Riểm thử

4.1 Phiên bảo mã nguồn gốc

Bảng 1: Báo cáo Ca kiểm thử

Test ID	(x_1, y_1)	(x_2, y_2)	Expected Output	Result	Note
TC01	(-1.91, 0)	(500000003. 390770053)	(500000003, 390770053)	Pass	
TC02	(500000003, 390770053)	(-1.91, 0)	(500000003, 390770053)	Pass	
TC03	(500000003, 390770053)	(1000000006, 959647287)		Pass	
TC04	(-1.91, 0)	(1000000006, 959647287)		Pass	
TC05	(500000003, 390770053)	(500000003, 390770053)		Pass	
TC06	(-1.91, 0)	(-1.91, 0)	`inf`	Pass	
TC07	(500000003, 390770053)	(500000003, -390770053)	`inf`	Pass	
TC08	(500000003, 390770053)	(994243306, 627647763)		Pass	
TC09	(500000003, 390770053)	`inf`	(500000003, 390770053)	Pass	
TC10	(500000003, 390770054)	(994243306, 627647763)	Invalid input	Pass	

4.2 Phiên bản mã nguồn lỗi

```python
class Point:
    def __add__(self, other):
       # Tampering handling inverse point addition
        if self.x == other.x and self.y == (-1 * other.y):
            return self

        # Tampering tangent line calcutation
        if self == other:
            x1, y1, a = self.x, self.y, self.curve.a

            s = (2 * x1 ** 2 + a) / (2 * y1)
            x3 = s ** 2 - 2 * x1
            y3 = s * (x1 - x3) - y1

            return self.__class__(
                x=x3.value,
                y=y3.value,
                curve=secp256k1
            )
```

Bảng 2: Báo cáo Ca kiểm thử

Test ID	(x_1, y_1)	(x_2, y_2)	Expected Output	Result	Note
TC01	(-1.91, 0)	(500000003. 390770053)	(500000003, 390770053)	Pass	
TC02	(500000003, 390770053)	(-1.91, 0)	(500000003, 390770053)	Pass	
TC03	(500000003, 390770053)	(1000000006, 959647287)		Pass	
TC04	(-1.91, 0)	(1000000006, 959647287)		Pass	
TC05	(500000003, 390770053)	(500000003, 390770053)		Fail	
TC06	(-1.91, 0)	(-1.91, 0)	`inf`	Pass	
TC07	(500000003, 390770053)	(500000003, -390770053)	`inf`	Pass	
TC08	(500000003, 390770053)	(994243306, 627647763)		Pass	
TC09	(500000003, 390770053)	`inf`	(500000003, 390770053)	Fail	
TC10	(500000003, 390770054)	(994243306, 627647763)	Invalid input	Pass	

5 Kết luận