دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

رمزنگاری

استاد درس: دکتر ملا

تیرماه ۱۴۰۳

کیمیا میرمقتدایی

۴۰۱۲۳۶۳۰۹۱

تسهیم راز شمیر

مقدمه

الگوریتم تسهیم راز شمیر توسط آدی شمیر طراحی شده‌است. این روش به ما اجازه می‌دهد که یک راز را بین n نفر تقسیم کنیم و به هر کس یک سهم بدهیم. به طوری که تنها با کنار هم گذاشتن حداقل t سهم بتوانیم راز را مجددا بسازیم. درواقع با داشتن کمتر از t سهم ساختن راز ممکن نیست. مقدار t همان آستانه‌ی مسئله است.

بررسی کد

برای این‌کار با ورودی گرفتن تعداد افراد، آستانه، راز و عدد اول که پیمانه‌ی محاسبات است، یک چندجمله‌ای خطی با درجه‌ی یکی کمتر از آستانه درست می‌کنیم. علت این‌کار این است که می‌دانیم با داشتن t نقطه می‌توانیم یک تابع چندجمله‌ای خطی یکتا به‌دست آوریم.

برای به‌دست آوردن این چند جمله‌ای، کافیست ضرایب چندجمله‌ای را به صورت رندوم تعیین کنیم (بین ۱ تا مقدار پیمانه تا ضریب هیچ کدام صفر نشود) و همچنین لازم به ذکر است که عرض از مبدأ این تابع همان مقدار راز است.

با داشتن ضرایب تابع خطی می‌توانیم خود چند جمله‌ای را بسازیم و سپس با استفاده از تابع calculate_F عرض نقاط را محاسبه می‌کنیم. ما می‌خواهیم در کل n نقطه روی این تابع داشته باشیم و برای این‌کار طول آن‌هارا از ۱ تا تعداد سهم‌ها و عرض آن‌هارا با جایگذاری طول نقاط در چندجمله‌ای به‌دست آمده محاسبه می‌کنیم.

```python
def generate_random_shares(secret: int, number_of_shares: int, threshold: int, prime: int) -> list:
    if number_of_shares < threshold:
        print("Number of shares MUST be greater than the threshold!")
        return None
    if prime <= number_of_shares:
        print("Number of shares MUST be smaller than the prime number!")
        return None

    coefficients = [random.randrange(1, prime) for _ in range(threshold - 1)]
    coefficients.append(secret)

    print_polynomial(coefficients, threshold)

    shares = [(x, calculate_F(coefficients, x, prime)) for x in range(1, number_of_shares + 1)]
    return shares
```

```python
def calculate_F(coefficients: list, x: int, prime: int) -> int:
    result = 0
    for coefficient in coefficients:
        result = (result * x + coefficient) % prime
    return result
```

در زیر یک مثال از چندجمله‌ای تولید شده و نقاط روی آن مشاهده می‌کنیم:

```
Secret: 123
Number of shares: 10
Threshold: 3
Prime: 127
The Polynomial is:
(33 * x ^ 2) + (126 * x ^ 1) + (123 * x ^ 0)
Shares:
(1, 28)
(2, 126)
(3, 36)
(4, 12)
(5, 54)
(6, 35)
(7, 82)
(8, 68)
(9, 120)
(10, 111)
```

با داشتن نقاط نوبت به بازسازی راز با استفاده از این نقاط می‌رسیم. برای این‌کار از فرمول درون‌یابی لاگرانژ با $x = 0$ استفاده می‌کنیم:

$$f(0) = \sum_{i=1}^{t} y_i \cdot \left(\prod_{j=1, j!=i}^{t} \frac{x_j}{x_j - x_i} \right)$$

این فرمول با دریافت t نقطه تابع خطی متناظر با آن را پیدا می‌کند و عرض از مبدأ آن را خروجی می‌دهد. این تابع را دقیقا با توجه به فرمول به شکل زیر پیاده‌سازی کرده‌ایم:

```python
def recover(shares: list[tuple[int, int]], prime: int, threshold: int) -> int:
    if len(shares) < threshold:
        print(f"We need at least {threshold} shares to recover the secret.")
        return None

    sum = 0
    for x, y in shares:
        result = y % prime
        for j in range(len(shares)):
            if x != shares[j][0]:
                result *= (shares[j][0] * pow(shares[j][0] - x, -1, prime)) % prime
        sum += result % prime

    return sum % prime
```

درنهایت خروجی این تابع خود مقدار راز خواهد بود.

چند مثال:

```
Secret: 65
Number of shares: 12
Threshold: 6
Prime: 83
The Polynomial is:
(66 * x ^ 5) + (9 * x ^ 4) + (52 * x ^ 3) + (15 * x ^ 2) + (45 * x ^ 1) + (65 * x ^ 0)
Shares:
(1, 3)
(2, 65)
(3, 80)
(4, 80)
(5, 3)
(6, 60)
(7, 23)
(8, 7)
(9, 7)
(10, 16)
(11, 60)
(12, 67)
Recovered secret: 65
```

```
Secret: 17452
Number of shares: 23
Threshold: 19
Prime: 18013
The Polynomial is:
(16376 * x ^ 18) + (7507 * x ^ 17) + (16376 * x ^ 16) + (14980 * x ^ 15) + (17712 * x ^ 14) + (5105 * x ^ 13) + (2259 * x ^ 12) + (9076 * x ^ 11) + (12632 * x ^ 10) + (8883 * x ^ 9) + (7226 * x ^ 8) +
(986 * x ^ 7) + (15428 * x ^ 6) + (304 * x ^ 5) + (2466 * x ^ 4) + (17687 * x ^ 3) + (2497 * x ^ 2) + (17598 * x ^ 1) + (17452 * x ^ 0)
Shares:
(1, 12420)
(2, 16214)
(3, 14353)
(4, 21)
(5, 16279)
(6, 16867)
(7, 6350)
(8, 16330)
(9, 7387)
(10, 315)
(11, 17148)
(12, 2879)
(13, 13738)
(14, 5410)
(15, 14032)
(16, 8309)
(17, 17025)
(18, 17206)
(19, 3404)
(20, 15139)
(21, 7301)
(22, 6977)
(23, 15313)
Recovered secret: 17452
```

همان‌طور که مشخص است راز به درستی ساخته می‌شود.