دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

درس رمزنگاری و امنیت شبکه

گزارش پروژه طرح تسهیم راز شمیر

تاریخ:

۱۴۰۳/۰۴/۷

ارشیا شفیعی ۴۰۰۳۶۲۳۰۱۹

لینک پروژه در گیت‌هاب:

https://github.com/arshiashafiei/shamir-secret-sharing

دو فایل برای اجرا وجود دارد.

قسمت اول پروژه در فایل calculate_shares.py وجود دارد که کار آن درست کردن n نقطه (x_i, y_i) در فضا است که بر روی یک چند جمله‌ای خاص قرار دارند.

تابع درست کردن نقاط:

```python
def make_shares(secret: int, t: int, n: int, prime: int) -> list[int]:
    """
    Generates a random shamir pool for a given secret, returns share points.
    """
    # TODO if p is a prime raise error
    if t > n:
        raise ValueError("Error: t is bigger than n!")
    if prime <= n:  # Because you give the secret to someone
        raise ValueError("Error: the number of shares is bigger than prime!")
      You, 2 hours ago • Add shares calculation
    coefficients = make_polynomial(secret, prime, t)

    print("=== The polynomial ===")
    for i, coeff in enumerate(coefficients):
        print(f"[({coeff}) * x^({t - i - 1})]", end=" + ")
    print()
    print("======")

    shares = [(i, evaluate_polynomial(coefficients, i, prime))
              for i in range(1, n + 1)]  # Y_i's set
    return shares
```

تابع ساخت چندجمله‌ای از درجه t-1 در GF عدد اول prime:

```python
def make_polynomial(secret: int, prime: int, t: int) -> list[int]:
    """Evaluates polynomial coefficients"""
    # The last coefficient is S = f(0)
    others = random.sample(range(prime), k=t-1)
    coefficients = others + [secret]
    return coefficients

```

تابع تعیین مقدار چند جمله‌ای P با استفاده از ضرایب آن:

```python
def evaluate_polynomial(coefficients: list[int], x: int, prime: int) -> int:
    """Evaluates polynomial at x,
    (A(t-1) * x + A(t-2)) mod p * x + A(t-3) mod p
    and so on..."""
    accum = 0
    for coeff in coefficients:
        accum = ((accum * x) + coeff) % prime
    return accum
```

به عنوان ورودی یک راز، یک عدد اول که پیمانه محاسبات است و t و n را می‌گیرد و برای خروجی نقاط مورد نظر را می‌دهد.

خروجی و اجرای برنامه:

```
--Please enter your Secret: 55
--Please enter minimum threshold(t): 3
--Please enter number of shares(n): 6
--Please enter the field number(p): 13
=== The polynomial ===
[(7) * x^(2)] + [(2) * x^(1)] + [(55) * x^(0)] +
======
(Xi=1, Yi=12)
(Xi=2, Yi=9)
(Xi=3, Yi=7)
(Xi=4, Yi=6)
(Xi=5, Yi=6)
(Xi=6, Yi=7)
```

در فایل دوم recover_secret.py قسمت دوم پیاده‌سازی شده است.

با اجرای آن با گرفتن نقاط yiها و p و تعداد نقاط t، به عنوان خروجی راز مورد نظر آن‌ها داده می‌شود.

تابع recover_secret با استفاده از فرمول درون‌یابی لاگرانژ:

```python
def recover_secret(shares: list[tuple[int, int]], prime: int) -> int:
    """Recover the secret from share points
    (points (x,y) on the polynomial).
    """
    if len(shares) < 3:
        raise ValueError("need at least three shares")
    sigma = 0
    for x, y in shares:
        pi = y % prime  # Y_i * ...
        # print(f"--Y= {y}")
        for j in range(len(shares)):
            if x != shares[j][0]:  # j != i
                # X_j / (X_j - X_i)
                # print(f"X_j= {shares[j][0]}")
                # print(f"(X_j - X_i)= {shares[j][0] - x}")
                # print(mod_inverse(shares[j][0] - x, prime))
                pi *= (shares[j][0]
                       * mod_inverse((shares[j][0] - x), prime)) % prime
                # print(pi % prime)
        sigma += pi % prime
    return sigma % prime
```

تابع محاسبه وارون ضربی طبق الگوریتم اقلیدس پیشرفته:

```python

def mod_inverse(a: int, prime: int) -> int:
    """
    Inverse of A can be computed via the extended Euclidean algorithm.
    """
    x = 0
    last_x = 1
    old_prime = prime
    while prime != 0:
        quot = a // prime
        a, prime = prime, a % prime
        x, last_x = last_x - quot * x, x
        # r_i+1 = r_i-1 - r_i * q_i not exactly this but you get the point.
    return last_x % old_prime
```

خروجی و اجرای برنامه:

```
--Please enter number of shares(t): 3
--Please enter the field number(p): 13
--Please enter X_i: 1
--Please enter Y_i: 4
--Please enter X_i: 2
--Please enter Y_i: 8
--Please enter X_i: 3
--Please enter Y_i: 1
Recovered Secret:
2
```

```
--Please enter number of shares(t): 3
--Please enter the field number(p): 13
--Please enter X_i: 3
--Please enter Y_i: 1
--Please enter X_i: 4
--Please enter Y_i: 9
--Please enter X_i: 5
--Please enter Y_i: 6
Recovered Secret:
2
```