Proyecto Criptografía

Integrantes:

Brancys Barrios

Nathalia De La Rans

Diego Linero

Docente:

Eduardo Angulo Madrid

Barranquilla

Universidad del Norte

26/11/2024

- **Funcionamiento esquemático firma digital.**

Para este proyecto, nosotros como grupo desarrollamos un esquema de firma digital basado en LUOV, un sistema criptográfico multivariante que se destaca por su resistencia a los ataques de computación cuántica. Este esquema de firma digital continúa la línea de trabajo que iniciamos previamente con el algoritmo 4 de LUOV, pero ahora avanzamos hacia la implementación del algoritmo 6, explorando sus particularidades y capacidades para garantizar una mayor seguridad y eficiencia en el manejo de firmas digitales.

El algoritmo 6 de LUOV se fundamenta en la complejidad de resolver sistemas de ecuaciones polinomiales multivariadas sobre campos finitos, característica que lo convierte en una solución robusta para un futuro post-cuántico. En este esquema, el proceso de generación de claves y firma se estructura en torno a una función matemática que mapea un espacio de n variables a otro de m variables, siendo $n=m+v$, donde v representa las variables de vinagre y m las variables de aceite.

La construcción de LUOV incluye una función cuadrática de un solo sentido definida mediante polinomios multivariados. Para que esta función sea invertible, el esquema introduce una estructura basada en una trampa matemática que combina una función cuadrática con una transformación lineal invertible (T), permitiendo generar firmas digitales únicas y verificables.

A lo largo de este proyecto, nos hemos centrado en comprender y aplicar los principios del algoritmo 6 para garantizar un esquema de firma digital eficiente y seguro. Este enfoque no solo refuerza los conceptos trabajados anteriormente, sino que también amplía nuestra capacidad para implementar soluciones criptográficas innovadoras que respondan a los desafíos emergentes en el ámbito de la seguridad digital.

Generación de llaves

En esta etapa, nosotros como equipo implementamos la generación de llaves utilizando el algoritmo 6 de LUOV. Este proceso comienza con la creación de una semilla privada, que se emplea como entrada en un Generador de Números Pseudoaleatorios. Este generador produce una serie de componentes que son esenciales tanto para la construcción de la clave pública como para la estructura interna del esquema de firma digital. En el código, esto se realiza mediante la función `GET_RANDOM_BYTES()` para generar los datos iniciales:

```
sk_seed = get_random_bytes(SEED_BYTES)
```

El PRNG utiliza esta semilla privada para producir una serie de componentes que son esenciales tanto para la construcción de la clave pública como para la estructura interna del esquema de firma digital.

Uno de los elementos generados es la matriz T, que desempeña un papel crucial en el ocultamiento de la estructura del mapa secreto F. A partir de esta semilla y la matriz T, se calculan los tres elementos principales de la llave pública:

1. **La Parte Constante (C):** Este término se define durante el proceso de construcción del mapa cuadrático.
2. **La Parte Lineal (L):** Se extrae junto con otros términos que componen el mapa secreto.
3. **El Mapa Cuadrático Representado por Q:** Compuesto por los componentes $Q1$ y $Q2$. En el código, se observa su definición mediante:

```
Q1 = [RANDOM_POLYNOMIAL() FOR _ IN RANGE(M)]
Q2 = [ADJUST_POLYNOMIAL(Q, T_INV) FOR Q IN Q1]
```

Aquí, $Q1$ se genera directamente mediante funciones aleatorias, mientras que $Q2$ se ajusta con la matriz T invertida (T_inv), garantizando que el mapa sea invertible y cumpla con los requerimientos de seguridad.

Finalmente, nosotros publicamos la semilla pública y los componentes de $Q2$ como parte de la llave pública, tal como se observa en el código:

```
PK = (PK_SEED, Q2)
```

Mientras tanto, la semilla privada, que fue la base para la generación de todos los elementos, permanece protegida como la clave privada. Este enfoque asegura que, aunque la clave pública sea ampliamente conocida, la clave privada no pueda inferirse, manteniendo la seguridad del sistema.

- # Códigos desarrollados y criptosistemas utilizados

Nosotros continuamos el desarrollo del esquema de firma digital avanzando al algoritmo 6, basándonos en el trabajo previo con el algoritmo 4. Esto nos permitió optimizar y ampliar la funcionalidad implementada, manteniendo la base criptográfica robusta. Utilizamos Python para estructurar el código de manera eficiente, aprovechando sus bibliotecas para manejar operaciones matemáticas complejas y criptográficas, fundamentales en LUOV.

Estrategias y diseño del código

Para esta etapa, comenzamos con la implementación del algoritmo de generación de llaves (KEYGEN.PY), siguiendo las especificaciones oficiales del algoritmo 6 de LUOV. La estrategia central fue la reutilización y mejora de los principios establecidos en el algoritmo 4, con ajustes específicos para manejar las complejidades adicionales. Este proceso inicia con la generación de una semilla privada, que es utilizada para derivar tanto la semilla pública como la matriz T, componente clave para ocultar la estructura del mapa secreto F.

En el código, esto se refleja en la creación de la semilla privada utilizando un generador de números pseudoaleatorios y el algoritmo SHAKE256:

$$\text{PRIVATE\_SEED} = \text{RANDOMBYTES}(32)$$

$$\text{SHAKE} = \text{SHAKE256}(\text{PRIVATE\_SEED})$$

A partir de esta semilla, generamos los polinomios $Q1$ y $Q2$, así como la matriz T. Estos forman el núcleo de la llave pública, que, junto con la semilla pública derivada, aseguran la resistencia del esquema ante ataques de reconstrucción de estructura. Por ejemplo, en el código, los componentes del mapa cuadrático son calculados de la siguiente manera:

$$Q1, Q2 = \text{GENERATE\_POLYNOMIALS}(\text{PUBLIC\_SEED})$$

$$T = \text{GENERATE\_MATRIX\_T}()$$

Criptosistemas utilizados

Para garantizar la seguridad y eficiencia, seguimos utilizando SHAKE256, que nos permite derivar valores de longitud variable necesarios para los cálculos. Este hash ha demostrado ser una herramienta flexible y segura, especialmente adecuada para las necesidades de LUOV. Además, ajustamos los parámetros en función de las especificaciones del algoritmo 6, permitiendo una mayor optimización en comparación con el esquema del algoritmo 4.

Desarrollo de la firma digital

Para el algoritmo 6, nosotros seguimos un enfoque iterativo para generar las firmas digitales. El proceso comienza fijando las variables de vinagre v, lo que transforma el sistema de ecuaciones cuadráticas en uno lineal. Este sistema se resuelve utilizando técnicas de álgebra matricial, lo que nos permite calcular las variables de aceite o y, finalmente, generar la firma s. Una vez obtenida, aplicamos la matriz T inversa para garantizar que la firma sea válida respecto al mensaje original.

En nuestro código, la resolución del sistema lineal se realiza mediante eliminación gaussiana, aprovechando librerías como NumPy para gestionar las operaciones matriciales de manera eficiente.

Bibliotecas utilizadas

Como se ha mencionado anteriormente, utilizamos SHAKE256 de la biblioteca pycryptodome, que es clave en todo el esquema LUOV por su capacidad para generar hashes seguros de longitud variable. Además, empleamos otras librerías como binascii, base64, json, os y numpy para la manipulación de datos binarios, generación de valores aleatorios y operaciones matemáticas necesarias para resolver los sistemas de ecuaciones.

Por ejemplo, para generar la semilla privada utilizamos:

```
PRIVATE_SEED = OS.URANDOM(32)

SHAKE = SHAKE256.NEW(PRIVATE_SEED)
```

Desafíos y aprendizajes

Un desafío significativo fue mantener la eficiencia al manejar sistemas complejos. Para abordar esto, optimizamos las operaciones relacionadas con las matrices y utilizamos estrategias de precomputación para reducir el tiempo necesario en etapas críticas como la firma y verificación. Estos ajustes resultaron en un proceso más rápido y eficiente, lo que valida la aplicabilidad práctica del esquema.

En resumen, nosotros desarrollamos un sistema robusto y modular que sigue los principios del algoritmo LUOV, garantizando su seguridad en un entorno post-cuántico. Este proyecto reafirmó la importancia de combinar un diseño criptográfico sólido con una implementación optimizada.