Proyecto Criptografía

Integrantes:

Brancys Barrios

Nathalia De La Rans

Brayan Gutierrez Pupo

Docente:

Eduardo Angulo Madrid

Barranquilla

Universidad del Norte

22/09/2024

- **Funcionamiento esquemático firma digital.**

Nosotros desarrollamos un esquema de firma digital utilizando LUOV, un sistema criptográfico multivariante resistente a ataques cuánticos. Este esquema basa su seguridad en la dificultad de resolver sistemas de ecuaciones polinomiales multivariadas sobre campos finitos, lo que lo convierte en una opción atractiva para un mundo post-cuántico.

El proceso de generación de llaves sigue el algoritmo LUOV, que involucra una función de un solo sentido $F_2{}^n{}_r \rightarrow F_2{}^m{}_r$, donde $n=m+v$ son variables, con v como variables de vinagre y m como variables de aceite. La función se define a través de polinomios cuadráticos, pero lo interesante del esquema es que su construcción permite que sea invertible gracias a una trampa basada en una función cuadrática combinada con una función lineal invertible T .

Generación de llaves

Primero, generamos una semilla privada que alimenta un Generador de Números Pseudoaleatorios, el cual genera la semilla pública y la matriz T , que es crucial para ocultar la estructura del mapa secreto F . A partir de la semilla pública, se generan tres componentes importantes para la llave pública: la parte constante C , la parte lineal L , y una parte del mapa cuadrático representada por Q_1. Luego, calculamos Q_2, la segunda parte del mapa cuadrático, que junto con Q_1 y T , forma el núcleo de nuestra llave pública.

En este punto, publicamos la semilla pública y Q_2 como la clave pública, mientras que la clave privada simplemente es la semilla utilizada inicialmente para generar todos los demás componentes.

• Códigos desarrollados y criptosistemas utilizados

Nosotros desarrollamos el código en Python, aquí se nos permitió manejar de manera eficiente las operaciones matemáticas y criptográficas necesarias para implementar el esquema LUOV. A la hora de estructurar el código, pensamos que era fundamental dividir las tareas en módulos claros que representaran cada parte del proceso, desde la generación de las llaves hasta la firma de mensajes.

Estrategias y diseño del código

El núcleo de nuestra solución está basado en el algoritmo de generación de llaves (KeyGen), que es donde comienza todo el proceso criptográfico. Decidimos implementar este algoritmo siguiendo los pasos descritos en la especificación oficial del LUOV, asegurándonos de optimizar el código para manejar eficientemente grandes volúmenes de datos. El algoritmo se basa en la idea de generar una semilla privada, que luego utilizamos para derivar tanto la semilla pública como la matriz T , que oculta la estructura interna del sistema F .

Un punto importante que discutimos fue cómo generar la matriz T . Para esto, utilizamos el generador de números pseudoaleatorios junto con el algoritmo SHAKE256. Esta función hash tiene la capacidad de generar flujos de bits de longitud variable, lo que fue ideal para nuestro propósito. Esto nos permitió no solo generar la matriz T , sino también derivar los polinomios Q_1 y Q_2, que forman parte del mapa cuadrático de la llave pública.

Criptosistemas utilizados

Uno de los puntos críticos de nuestro desarrollo fue seleccionar las funciones hash y criptosistemas adecuados para garantizar la seguridad y eficiencia del algoritmo. Optamos por **SHAKE1256** como el mecanismo principal para generar semillas y derivar componentes clave. Pensamos que **SHAKE256** ofrecía la combinación perfecta entre seguridad y eficiencia, ya que es resistente a ataques de colisión y su capacidad de generar salidas de longitud variable lo hace versátil para nuestras necesidades.

Desarrollo de la firma digital

Para la firma digital, seguimos una estrategia iterativa. El algoritmo que implementamos primero genera una firma parcial probando diferentes asignaciones para las variables de vinagre v . Lo interesante es que, una vez que las variables de

vinagre se fijan, el sistema de ecuaciones cuadráticas se reduce a un sistema lineal, que podemos resolver utilizando álgebra matricial.

Nosotros implementamos una función que construye una **matriz aumentada**, utilizando las ecuaciones lineales que resultan al fijar v . Esta matriz se resuelve mediante **eliminación gaussiana** para obtener las soluciones del sistema, es decir, las variables de aceite o . Este paso es clave porque nos permite obtener la firma s , que luego transformamos aplicando la matriz T inversa para recuperar el mensaje original.

Bibliotecas utilizadas

Nosotros decidimos usar **SHAKE256** de la biblioteca *pycryptodome* porque sabíamos que era una opción ideal para generar hashes criptográficamente seguros y flexibles. Lo que hace que SHAKE256 sea tan útil en este proyecto es que nos permite generar salidas de longitud variable, lo que es importante en diferentes partes del esquema LUOV, especialmente en la generación de claves y firmas.

Además, queríamos asegurarnos de que cualquier dato binario que necesitáramos procesar o almacenar pudiera ser manejado de manera eficiente. Por eso decidimos que usar *binascii* y *base64* era útil para transformar los datos en representaciones manejables cuando era necesario exportarlos o compartirlos entre sistemas.

Nosotros usamos *os.urandom()* para generar la semilla privada, es algo que debe ser impredecible, y para asegurarnos de esto, dependimos de *os* para obtener la aleatoriedad que necesitábamos.

Desafíos y aprendizajes

Uno de los principales desafíos que enfrentamos fue la optimización del tiempo de ejecución. Sabíamos que la complejidad del sistema podía aumentar significativamente, especialmente cuando se trataba de resolver los sistemas de ecuaciones. Para mitigar este problema, decidimos implementar un esquema de **precomputación** para acelerar el proceso de firma. Esto implicaba almacenar parte de los cálculos relacionados con las llaves públicas y privadas, lo que reducía considerablemente el tiempo de cálculo durante la verificación de firmas.

En resumen, nosotros desarrollamos un sistema que es seguro, eficiente y modular, basado en los principios del criptosistema LUOV. Cada paso del proceso fue cuidadosamente analizado y optimizado, asegurando que cumpliera con los requisitos de seguridad post-cuántica y que fuera factible implementarlo en un entorno práctico.