**Objetivo del prototipo**

Conocer el uso, funcionamiento e implementación de herramientas de cifrado, hashing y generación de CAPTCHAS, con el fin de conocer la integración de estos módulos en diferentes lenguajes de programación.

**Prototipo 1**

Se implementó un módulo de cifrado de mensajes de texto en lenguaje C++. Tratando de simular el proceso de cifrado del esquema que estamos usando.

La primera parte del proceso es abrir el mensaje para lo cual usamos las bibliotecas estándar stdio.h, posteriormente crearemos una palabra aleatoria de 10 caracteres usando una función Rand()%100 casteando el valor de salida a un char.

Al resultado se pasa por una función de hashing, esta función no es nativa de ninguna biblioteca estándar de C++ ni de C, por lo que se tuvo que conseguir una en internet y probar que efectivamente funcionara como lo necesitamos.

Posteriormente este hash se usara como llave para cifrar el mensaje que ya abrimos, para esto necesitaremos una función AES o DES, ninguna de estas es estándar de alguna biblioteca de C o C++, así que tendremos que buscarlas y verificar su funcionamiento.

**Conclusión**

Podemos ver que en C++ el proceso es simple pero se necesita buscar muy bien las bibliotecas externas que se usaran, ya que no siempre están funcionando correctamente, en algunos casos están ni siquiera compilan.

Este caso fue particularmente evidente al buscar una biblioteca de C o C++ que pudiera realizar el cifrado con AES o DES, encontrándonos con bibliotecas que cifraban mal ya que al meter la misma llave no descifraban e incluso bibliotecas que no logramos hacer que compilaran.

**Prototipo 2**

Se implementó un módulo de cifrado, descifrado y generación de CAPTCHAS en Python, simulando el proceso antes del envió del correo y el que se hace después de la recepción de los correos electrónicos.

Para este se usó el formato estándar del correo electrónico especificado en el RFC 822, también se usaron bibliotecas ya estandarizadas de python para la implementación de las funciones de hashing, funciones de cifrado y descifrado (AES o DES), funciones aleatorias y la generación de los CAPTCHAS.

**Conclusión**

Se logró generar todo el proceso de envió y parte de el de recibimiento de mensajes. En cuanto a el envió se logró leer el mensaje, crear el strig a partir de funciones random, crear la llave con esa string y cifrar el correo exitosamente, además de esto se logró leer el archivo de mensaje de correo electrónico y cifrar únicamente el mensaje que viene en este.

Por su parte el módulo de generación de CAPTCHAS mostró muchos problemas para generarlos, ya que no logramos hacer que el intérprete pudiera encontrar correctamente las funciones de la biblioteca de generación CAPTCHAS por lo que al no poder generar un CAPTCHA la recuperación no se puede realizar como se planteó, para verificar únicamente que las funciones sirven se implementó el descifrado del mensaje en el mismo método.