|  |
| --- |
|  |
| Voto Electrónico |
| Etapa 2 |
|  |
| **Organización de Datos – Cátedra Servetto 2do cuatrimestre 2011** |
| **3/12/2011** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Alfredo Scoppa | 89149 | | Cecilia khalil | 87996 | | Fernando Romera Ferrio | 88406 | | Juan Manuel Romera Ferrio | 88405 | |

Tabla de contenidos

[Diseño 3](#_Toc310619175)

[Introducción 3](#_Toc310619176)

[Diagrama de clases 3](#_Toc310619177)

[Algoritmo asimétrico 4](#_Toc310619178)

[RSA 4](#_Toc310619179)

[Diagrama de clases 4](#_Toc310619180)

[Vigenere 5](#_Toc310619181)

[Diagrama de clases 5](#_Toc310619182)

[Ataque al criptosistema Vigenere 6](#_Toc310619183)

[Método de Kasiski 6](#_Toc310619184)

[Ataque al criptosistema RSA 7](#_Toc310619185)

[Método de factorización de números primos 7](#_Toc310619186)

[Manual de Usuario 8](#_Toc310619187)

# Diseño

## Introducción

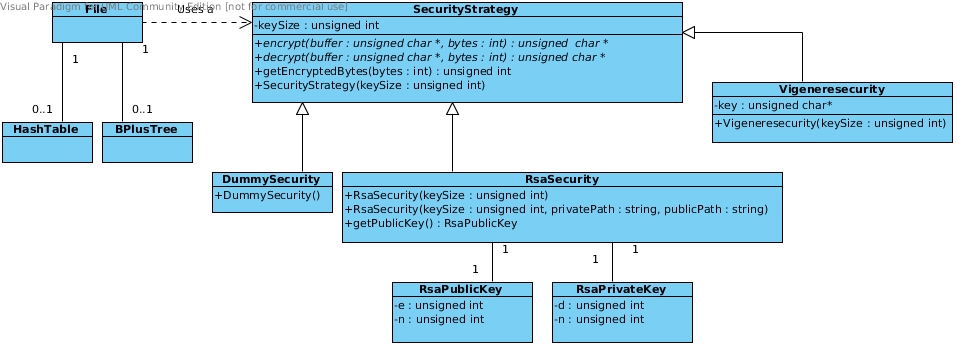
Se decidió optar por una estrategia que impacte lo menor posible en el modelo existente y que sea casi transparente para el mismo.

En definitiva, las estructuras de archivos existentes siguen dependiendo de las mismas clases pero son los archivos (clase File) los que ahora delegan en una estrategia de seguridad el resguardo de los datos.

Para tal objetivo se optó por usar una solución similar a la propuesta por el patrón Strategy, implementando las siguientes estrategias de seguridad:

* Simétrica: VigenereSecurity, para los archivos de reportes.
* Asimétrica: RsaSecurity, para los archivos que almacenan la información de votantes y administradores.
* Texto Plano: DummySecurity, para archivos de texto plano.

## Diagrama de clases



# Algoritmo asimétrico

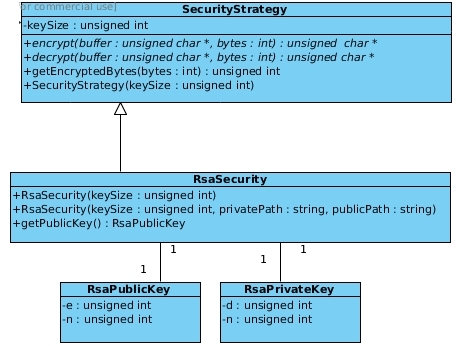
## RSA

Este algoritmo se implementó según lo visto en clase con algunas restricciones e hipótesis simplificadoras aplicadas para que su implementación sea más sencilla.

Primero está el factor del resguardo de las claves, las mismas se almacenan en el sistema de archivos del usuario y sin ninguna protección alguna. Sin embargo, se supone que la carpeta en la cual se almacena la clave privada (.forbiddenAccessFolder) es solo accesible por el dueño de la clave.

Finalmente, aunque parametrizable, el tamaño de las claves está limitado hasta los 26 bits.

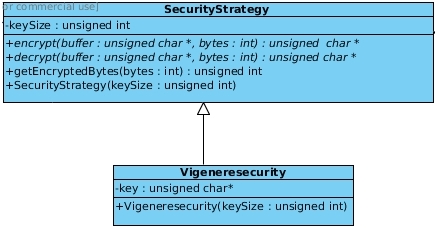
## Diagrama de clases

Algoritmo simétrico

## Vigenere

Este algoritmo se implementó según lo visto en clase.   
  
Las claves se generan de manera aleatoria tomando cualquiera de los 256 caracteres disponibles en ASCII, su tamaño es parametrizable y luego de generadas se almacenan en un archivo para ser recuperadas más tarde.

## Diagrama de clases



# Ataque al criptosistema Vigenere

## Método de Kasiski

Este método intenta vulnerar el algoritmo de Vigenere mediante la busqueda de strings repetidos, aprovechando que la distancia entre ocurrencias es probablemente un múltiplo de la longitud de la clave, se factorizan todas las distancias encontradas, los factores más frecuentes son candidatos a ser la longitud de la clave.

Teniendo longitudes candidatas para la clave, se trata cada columna como un cifrador monoalfabético (ROT(n)), muy fácil de romper, y si el correspondiente análisis de frecuencia tiene éxito para alguna de las longitudes candidatas, se ha roto el cifrador.

# Ataque al criptosistema RSA

## Método de factorización de números primos

Este método intenta vulnerar RSA mediante la factorización de números primos.   
Se van probando todos los números primos hasta encontrar un divisor de N. Al encontrarlo, se lo asigna a p o a q.

Luego, despejando de la fórmula n = p x q se obtiene el otro valor.

Se busca D con el algoritmo de Euclides Extendido, obteniendo la clave privada.

Finalmente, se aplica el algoritmo de desencripción sobre el archivo que se desee vulnerar.

# Manual de Usuario

Ver documento adjunto: ManualDeUsuario.pdf