



Trabajo Terminal 2015-A010



Aplicación de cifrado contra adversarios clasificadores, para el correo electrónico

Presenta:

Jonathan Arcos Ayala

Allan Ulises Zepeda Ibarra

Dirige:

Sandra Díaz Santiago

Manuel Alejandro Soto Ramos

2 de junio de 2016

Contenido

- 1 **Introducción**
 - Información
 - Datos relevantes
 - Tipos de Adversarios
- 2 **Problemática**
 - Adversario Clasificador
 - Esquema de Díaz-Chakraborty
- 3 **Propuesta de solución**
 - Objetivos
 - Arquitectura propuesta
 - Tecnologías
- 4 **Trabajo terminal I**
 - Prototipos
- 5 **Trabajo terminal II**
 - Complemento para el cliente de correo Thunderbird
 - Complemento para el cliente de correo Nylas-N1
 - Implementación de un cliente propio

Introducción

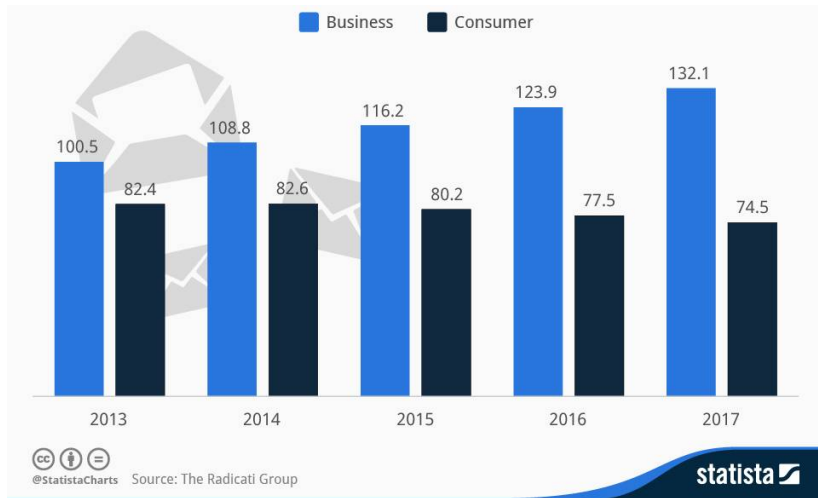


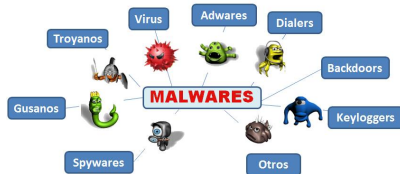
Figura: Estimación de mensajes enviados y recibidos en un día en todo el mundo (en billones)

Datos relevantes.

Datos relevantes

- 3.9 billones de cuentas de correo electrónico
- 3 mil millones de usuario en internet aproximadamente
- 1.55 mil millones de usuarios en facebook aproximadamente

Tipos de Adversarios.



Software



Personas o grupos de personas

Adversario Clasificador.



Adversarios Clasificadores

On July 11, 2014 Lang Lang performed together with Plácido Domingo, Ana Maria Martinez, Maestro Eugene Kohn, the Orchestra Sinfonica Brasileira, Paula Fernandez and other musicians for World Cup Concert at HSBC Arena in Rio de Janeiro Brasil

No



Si



Esquema de Díaz-Chakraborty

Protocolo P



Protocolo P'

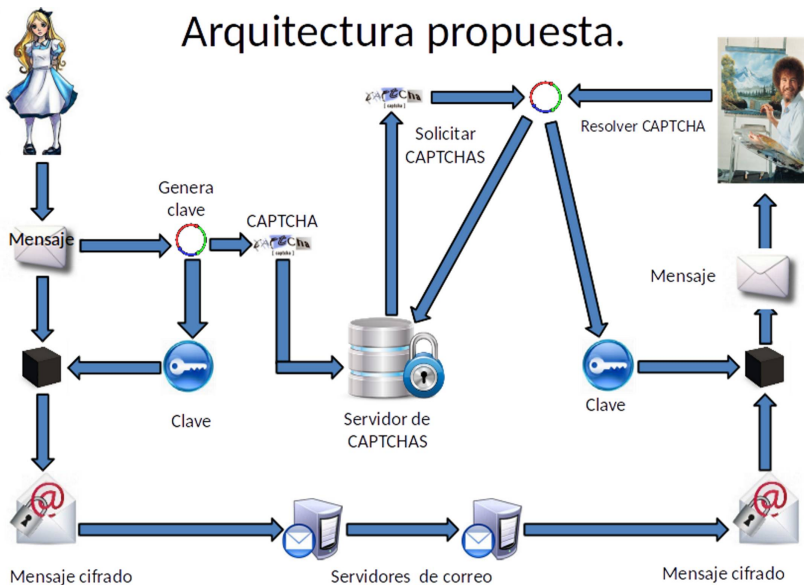


Objetivos

Objetivos

- 1 Desarrollar una herramienta en un cliente de correo electrónico para el envío y recepción de los correos cifrados y la generación, envío y recepción de CAPTCHAS.
- 2 Desarrollar un servidor de llaves que reciba, aloje y envíe los CAPTCHAS a los usuarios para descifrar los correos electrónicos.
- 3 Desarrollar un algoritmo de cifrado y descifrado basado en el envío y recepción de CAPTCHAS.

Arquitectura propuesta.



Tecnologías



Trabajo desarrollado en TT I

Prototipos

- 1 Prototipo de generación de CAPTCHAS en C++.
- 2 Prototipo de generación de CAPTCHAS en PYTHON.
- 3 Instalación de un cliente de correo electrónico web y un servidor DNS.
- 4 Prototipo de generación de CAPTCHAS a partir de un mensaje de correo electrónico recuperado del cliente de correo web.

Complemento para el cliente de correo Thunderbird

The screenshot shows the MDN website with a dark blue header. The header includes the MDN logo, navigation links for 'WEB PLATFORM', 'MOZILLA DOCS', 'DEVELOPER TOOLS', and 'FEEDBACK', and a search bar. Below the header, the article title 'Creating custom Firefox extensions with the Mozilla build system' is displayed. A yellow box highlights a note: 'This article needs a technical review. How you can help.' The article text begins with 'There is a wealth of material on creating extensions for Firefox. All of these documents currently assume, however, that you are developing your extension using XUL and JavaScript only. For complex extensions, it may be necessary to create components in C++ that provide additional functionality. Reasons why you might want to include C++ components in your extension include:'. A list of reasons follows: 'Need for high-performance beyond what can be delivered by JavaScript code.' and 'Use of third-party libraries written in C or C++.' The right sidebar contains a section 'IN THIS ARTICLE' with links to 'Bambi Meets Mozilla', 'On Windows Platforms', 'On Other Platforms', 'Structuring Your Project', 'Anatomy of a Simple C++ Extension', 'Public Interfaces', and 'Source File'.

MDN > Mozilla > Add-ons > Creating custom Firefox extensions with the Mozilla build system

Creating custom Firefox extensions with the Mozilla build system

SEE ALSO

- WebExtensions
 - Getting started
 - Guides
 - JavaScript APIs
 - Manifest keys
- Add-on SDK
 - Getting started

This article needs a technical review. [How you can help.](#)

There is a [wealth of material](#) on creating extensions for Firefox. All of these documents currently assume, however, that you are developing your extension using [XUL](#) and [JavaScript](#) only. For complex extensions, it may be necessary to create components in C++ that provide additional functionality. Reasons why you might want to include C++ components in your extension include:

- Need for high-performance beyond what can be delivered by JavaScript code.
- Use of third-party libraries written in C or C++.

IN THIS ARTICLE

- [Bambi Meets Mozilla](#)
- [On Windows Platforms](#)
- [On Other Platforms](#)
- [Structuring Your Project](#)
- [Anatomy of a Simple C++ Extension](#)
- [Public Interfaces](#)
- [Source File](#)

Figura: Página web de Mozilla Developer Network.

Complemento para el cliente de correo Nylas-N1

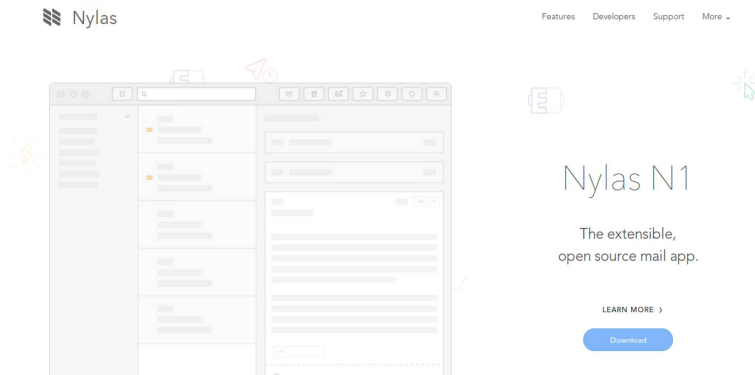


Figura: Página web de Nylas N1.

Implementación de un cliente propio

Conclusiones

Conclusiones

- 1 El esquema Díaz - Chakraborty es posible implementarse en los esquemas actuales de comunicación por correo electrónico.
- 2 El ataque de los agentes clasificadores es en los servidores de correo y no en las comunicaciones.

Trabajo Futuro

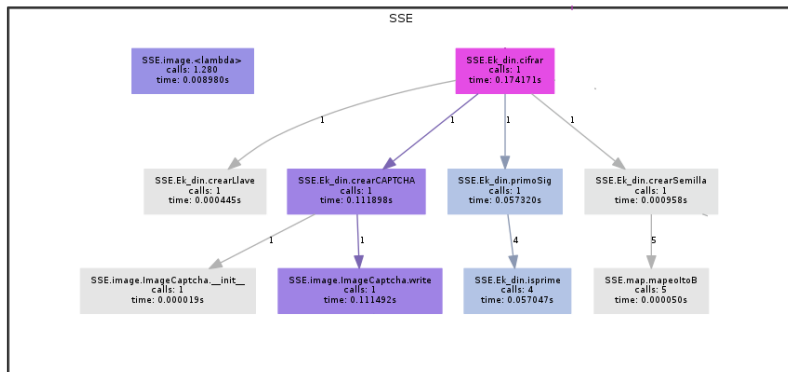
Conclusiones

- 1 Complemento para un otro tipo de clientes de correo electrónico.
- 2 Esquema de intercambio de claves.
- 3 Implementar un cifrado semántico.
- 4 Biblioteca de creación de CAPCHAS en el lenguaje PYTHON.

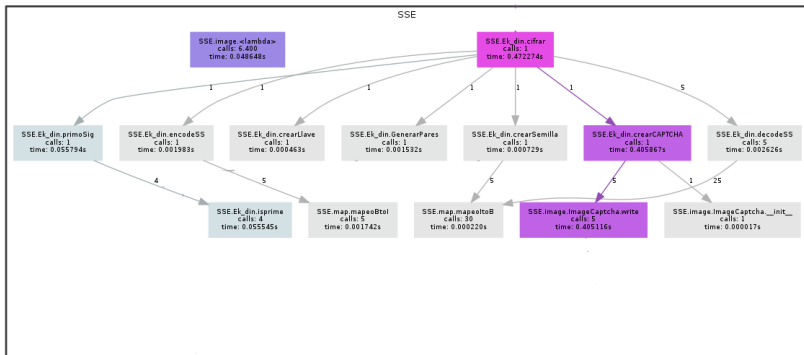
Sección de preguntas



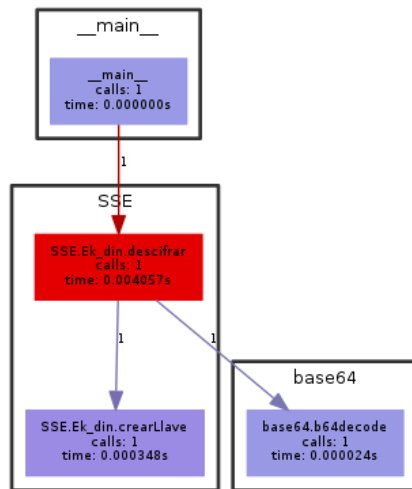
Prueba de cifrado unicaptchas



Prueba de cifrado multicaptchas



Prueba de descarga de CAPTCHAS



Generated by Python Call Graph v1.0.1
<http://pycallgraph.slowchop.com>

Prueba de descarga de multi-CAPTCHAS

