

# GENERACIÓN DE TOKENS PARA PROTEGER LOS DATOS DE TARJETAS BANCARIAS

TRABAJO TERMINAL No. 2017-B008

PRESENTAN

DANIEL AYALA ZAMORANO

DAZ23AYALA@GMAIL.COM

LAURA NATALIA BORBOLLA PALACIOS

LN.BORBOLLA42@GMAIL.COM

RICARDO QUEZADA FIGUEROA

QF7.RICARDO@GMAIL.COM

DIRECTORA

DRA. SANDRA DÍAZ SANTIAGO

CIUDAD DE MÉXICO, 9 DE MAYO DE 2018

ESCUELA SUPERIOR DE CÓMPUTO  
INSTITUTO POLITÉCNICO NACIONAL



# CONTENIDO

Planteamiento del problema

# CONTENIDO

Planteamiento del problema

Planteamiento de la solución

# CONTENIDO

Planteamiento del problema

Planteamiento de la solución

Algoritmos generadores de *tokens*

# CONTENIDO

Planteamiento del problema

Planteamiento de la solución

Algoritmos generadores de *tokens*

Conclusiones

# CONTENIDO

Planteamiento del problema

Planteamiento de la solución

Algoritmos generadores de *tokens*

Conclusiones

# UN INICIO TORMENTOSO

- ▶ En la década de los 80 y 90, el comercio en línea comenzó a crecer y tomar importancia.
- ▶ Las empresas no estaban preparadas para el impacto que tuvieron y los fraudes relacionados con el comercio electrónico aumentaron rápidamente [1].
  - ▶ Visa y Mastercard reportaron, entre 1988 y 1998, pérdidas de 750 millones de dólares.
  - ▶ En 2001, se reportaron pérdidas de 1.7 miles de millones de dólares. y 2.1 miles de millones de dólares al año siguiente.

# UN ESTÁNDAR PARA GOBERNARLOS A TODOS

- ▶ A inicios del 2000, las grandes compañías (¿emisoras de tarjetas?) comenzaron a publicar, individualmente, *buenas prácticas* de seguridad.
- ▶ Las empresas intentaron adoptar las prácticas, pero era tremendamente complicado y costoso.
- ▶ Se aliaron las compañías y, en 2004, publicaron un estándar unificado: PCI-DSS<sup>1</sup> [2].
  - ▶ Se hizo obligatorio para quienes realizasen más de 20K transacciones al año.
  - ▶ Tiene un gran número de requerimientos (y subrequerimientos), por lo que es difícil de satisfacer.

---

<sup>1</sup>Payment Card Industry - Data Security Standard



# CAMBIO DE ESTRATEGIA

- ▶ Hasta ahora, el enfoque era proteger los datos sensibles donde sea que se encuentren y por donde sea que transiten.
- ▶ Surge un nuevo enfoque: cambiar la información valiosa, por *valores representativos* (tokens); es decir, la tokenización de la información.
- ▶ En 2011, el PCI-SSC<sup>2</sup> publicó las primeras guías para los procesos de tokenización [3].
  - ▶ Aunque indica lo que debe satisfacer el sistema tokenizador, no dice cómo generar los tokens.

---

<sup>2</sup>Payment Card Industry - Security Standards Council

# PERO ¿POR QUÉ?

A pesar de ser una práctica extendida, la tokenización sigue estando rodeada de desinformación y desconfianza.

- ▶ Se busca combatir la desinformación al estudiar e implementar cinco algoritmos tokenizadores, compararlos y mostrar los resultados.
- ▶ Hacer notar que la criptografía y la tokenización no están peleadas; pues la tokenización puede verse como una aplicación de la criptografía.

# CONTENIDO

Planteamiento del problema

Planteamiento de la solución

Objetivos del proyecto 9

Metodología del proyecto 10

Prototipos 11

Algoritmos generadores de *tokens*

Conclusiones

# OBJETIVOS DEL PROYECTO




Lo que se busca con este proyecto es implementar un programa generador de *tokens* que provea confidencialidad a los datos de las tarjetas bancarias.

Además, con el afán de disminuir la desinformación existente sobre la tokenización, se busca obtener una comparativa de los algoritmos implementados.



# PROTOTIPOS

Este proyecto está dividido en 3 prototipos, los cuales son:

 <b>Prototipo de generación de tokens. ✓</b>	 <b>Prototipo de servicio Web.</b>	 <b>Prototipo de tienda en línea.</b>
<p>Revisar e implementar diversos algoritmos generadores de tokens para hacer un programa tokenizador, así como realizar pruebas comparativas entre estos algoritmos.</p>	<p>Diseñar e implementar una API web capaz de comunicar al programa tokenizador con al menos una tienda en línea con el fin ofrecer el servicio de tokenización.</p>	<p>Implementar una tienda en línea que utilice la API web para poder revisar el correcto funcionamiento del servicio.</p>

## Prototipos del trabajo terminal.

# CONTENIDO

Planteamiento del problema

Planteamiento de la solución

Algoritmos generadores de *tokens*

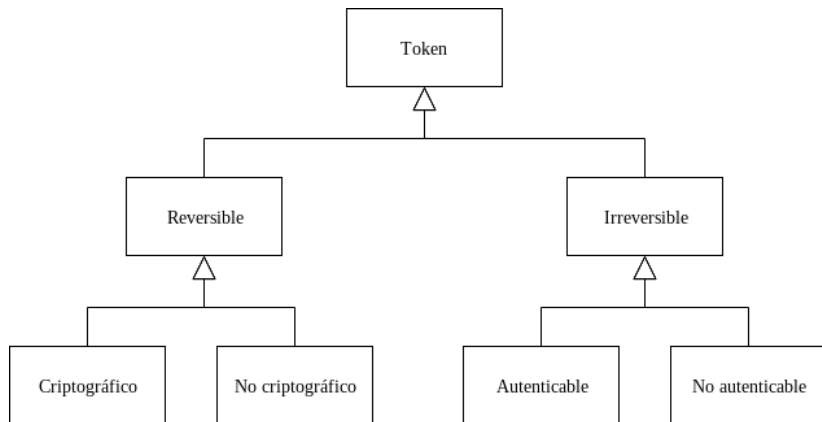
Clasificación 13

Implementaciones 16

Comparación de desempeño 18

Conclusiones

# CLASIFICACIÓN DEL PCI SSC



Clasificación de *tokens* [3].



## Generación de tokens para proteger los datos de tarjetas bancarias

- Algoritmos generadores de *tokens*

- Clasificación

- Clasificación del PCI SSC



Los irreversibles no pueden ser reconvertidos al PAN (de ninguna manera, mas que con fuerza bruta). Los autenticables funcionan como una función Hash: si tienes el PAN y el token, se puede validar que ese token es el par de ese PAN. Los no autenticables no pueden validar esto último.

Los reversibles permiten obtener el PAN a partir del token. Los no criptográficos ocupan funciones pseudoaleatorias y una base de datos para guardar las relaciones PAN-token. Los criptográficos ocupan un esquema de cifrado tradicional: un PAN mas una llave permiten obtener un token; la llave y el token pueden ser ocupados para obtener el PAN. No se ocupa una base de datos.

# CLASIFICACIÓN DEL PCI SSC

¿«No criptográficos»?

La clasificación anterior presenta los siguientes problemas:

- Los casos de uso que el PCI SSC prevé en [3] para los irreversibles resultan artificiosos.

# Generación de tokens para proteger los datos de tarjetas bancarias

- └ Algoritmos generadores de *tokens*
  - └ Clasificación
    - └ Clasificación del PCI SSC

La clasificación anterior presenta los siguientes problemas:

- Los casos de uso que el PCI SSC prevé en [3] para los irreversibles resultan artificiales.

Por ejemplo, la justificación para los no autenticables es para dar soporte a aplicaciones obsoletas que necesitan un formato de PAN válido. Esto se puede lograr con los no criptográficos sin guardar nada en la base; o pasando puros ceros en el campo del PAN.

El caso para los autenticables permite verificar la tarjeta del cliente en una compra cuando este perdió el comprobante. En est caso no resulta claro por qué la tienda (o el sistema tokenizador) no guardaría la transacción original.

# CLASIFICACIÓN DEL PCI SSC

¿«No criptográficos»?

La clasificación anterior presenta los siguientes problemas:

- ▶ Los casos de uso que el PCI SSC prevé en [3] para los irreversibles resultan artificiosos.
- ▶ A pesar del nombre, los *no criptográficos* ocupan diversas aplicaciones de la criptografía para generar *tokens*.

# Generación de tokens para proteger los datos de tarjetas bancarias

- └ Algoritmos generadores de *tokens*

- └ Clasificación

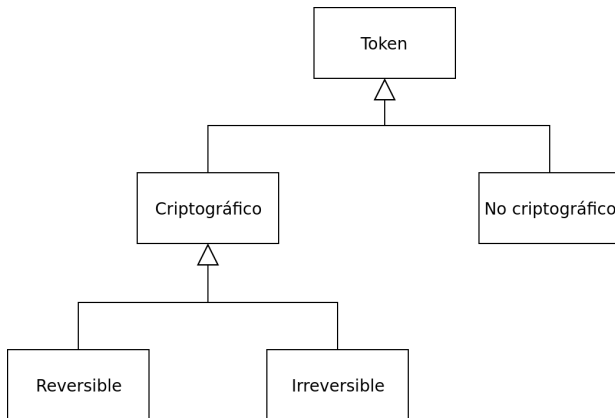
- └ Clasificación del PCI SSC

La clasificación anterior presenta los siguientes problemas:

- Los casos de uso que el PCI SSC prevé en [3] para los irreversibles resultan artificiales.
- A pesar del nombre, los no criptográficos ocupan diversas aplicaciones de la criptografía para generar tokens.

El problema con el PCI es que parecen pensar que la criptografía se limita a esquemas tradicionales, en donde hay una llave. La generación de números pseudoaleatorios seguros es también una aplicación de la criptografía.

# CLASIFICACIÓN PROPUESTA



Clasificación propuesta.

## Generación de tokens para proteger los datos de tarjetas bancarias

- └ Algoritmos generadores de *tokens*
  - └ Clasificación
    - └ Clasificación propuesta



Los únicos que se contemplan como «no criptográficos» son los que están basados en generadores realmente aleatorios. Todos los demás caen en la categoría de «criptográficos». Los reversibles son los que están basados en esquemas tradicionales (v. gr. los cifrados que preservan el formato). Los irreversibles necesitan de una base de datos para poder hacer el proceso inverso.

# ALGORITMOS IMPLEMENTADOS

- Reversibles:



# ALGORITMOS IMPLEMENTADOS

- ▶ Reversibles:
  - ▶ FFX (*Format-preserving Feistel-basen Encryption*).  
Publicado por Mihir Bellare, Phillip Rogaway y Terence Spies en [5].

# Generación de tokens para proteger los datos de tarjetas bancarias

- └ Algoritmos generadores de *tokens*
  - └ Implementaciones
    - └ Algoritmos implementados

- Reversibles:
  - FFX (Format-preserving Feistel-based Encryption)  
Publicado por Mikar Bodden, Phillip Rogaway y Tsvi  
Spies en [5].

Es una propuesta de estándar para el NIST. Los autores son los principales precursores de los cifrados que preservan el formato.

El método está basado en redes Feistel y una función de ronda que ocupa CBC-MAC-AES.

# ALGORITMOS IMPLEMENTADOS

- ▶ Reversibles:
  - ▶ FFX (*Format-preserving Feistel-basen Encryption*).  
Publicado por Mihir Bellare, Phillip Rogaway y Terence Spies en [5].
  - ▶ BPS. Publicado por Eric Brier, Thomas Peyrin y Jacques Stern en [6].

# Generación de tokens para proteger los datos de tarjetas bancarias

- └ Algoritmos generadores de *tokens*
  - └ Implementaciones
    - └ Algoritmos implementados

- Reversibles:
  - FFX (*Format-preserving Feistel-based Encryption*).  
Publicado por Mihir Bellare, Phillip Rogaway y Trent S. Spies en [5].
  - BPS. Publicado por Eric Brier, Thomas Peyrin y Jacques Stern en [6].

También es propuesta de estándar para el NIST. Representa la principal competencia de FFX.

Al igual que FFX, ocupa redes Feistel de forma interna; se diferencian en algunos detalles de instanciación y en que BPS está diseñado para cadenas de longitud arbitraria.

# ALGORITMOS IMPLEMENTADOS

- ▶ Reversibles:
  - ▶ FFX (*Format-preserving Feistel-basen Encryption*).  
Publicado por Mihir Bellare, Phillip Rogaway y Terence Spies en [5].
  - ▶ BPS. Publicado por Eric Brier, Thomas Peyrin y Jacques Stern en [6].
- ▶ Irreversibles:

# ALGORITMOS IMPLEMENTADOS

- ▶ Reversibles:
  - ▶ FFX (*Format-preserving Feistel-basen Encryption*).  
Publicado por Mihir Bellare, Phillip Rogaway y Terence Spies en [5].
  - ▶ BPS. Publicado por Eric Brier, Thomas Peyrin y Jacques Stern en [6].
- ▶ Irreversibles:
  - ▶ TKR. Publicado por Sandra Díaz-Santiago, Lil María Rodríguez-Henríquez y Debrup Chakraborty en [7].

# Generación de tokens para proteger los datos de tarjetas bancarias

- └ Algoritmos generadores de *tokens*
  - └ Implementaciones
    - └ Algoritmos implementados

- Reversibles:
  - FFX (*Format-preserving Fixed-Secret Encryption*).  
Publicado por Mihir Bellare, Phillip Rogaway y Trent S. Spies en [5].
  - RPS. Publicado por Eric Brier, Thomas Peyrin y Jacques Stern en [6].
- Irreversibles:
  - TKR. Publicado por Sandra Díaz-Santiago, Lil María Rodríguez-Henríquez y Debapriya Chakraborty en [7].

El documento es el primer análisis formal sobre la generación de tokens. TKR es el primer método propuesto (cuya seguridad está formalmente probada) que no es un cifrado que preserva el formato.

# ALGORITMOS IMPLEMENTADOS

- ▶ Reversibles:
  - ▶ FFX (*Format-preserving Feistel-basen Encryption*). Publicado por Mihir Bellare, Phillip Rogaway y Terence Spies en [5].
  - ▶ BPS. Publicado por Eric Brier, Thomas Peyrin y Jacques Stern en [6].
- ▶ Irreversibles:
  - ▶ TKR. Publicado por Sandra Díaz-Santiago, Lil María Rodríguez-Henríquez y Debrup Chakraborty en [7].
  - ▶ AHR (Algoritmo Híbrido Reversible). Longo, Aragona y Sala en [8].



# ALGORITMOS IMPLEMENTADOS

- ▶ Reversibles:
  - ▶ FFX (*Format-preserving Feistel-basen Encryption*). Publicado por Mihir Bellare, Phillip Rogaway y Terence Spies en [5].
  - ▶ BPS. Publicado por Eric Brier, Thomas Peyrin y Jacques Stern en [6].
- ▶ Irreversibles:
  - ▶ TKR. Publicado por Sandra Díaz-Santiago, Lil María Rodríguez-Henríquez y Debrup Chakraborty en [7].
  - ▶ AHR (Algoritmo Híbrido Reversible). Longo, Aragona y Sala en [8].
  - ▶ DRBG (*Deterministic Random Bit Generator*). Adaptación a partir de estándar del NIST (*National Institute of Standards and Technology*) [9].

# Generación de tokens para proteger los datos de tarjetas bancarias

- └ Algoritmos generadores de *tokens*
  - └ Implementaciones
    - └ Algoritmos implementados

- Reversibles:
  - FFX (*Fastmat-preserving Feistel-based Encryption*). Publicado por Mihir Bellare, Phillip Rogaway y Trent S. Spies en [5].
  - RPS. Publicado por Eric Brier, Thomas Peyrin y Jacques Stern en [6].
- Irreversibles:
  - TKR. Publicado por Sandra Díaz-Santiago, Lil María Rodríguez-Henríquez y Delarup Chakraborty en [7].
  - AHR (Algoritmo Híbrido Reversible). Longo, Aragona y Sala en [8].
  - DRBG (*Deterministic Random Bit Generator*). Adaptación a partir de estándar del NIST (*National Institute of Standards and Technology*) [9].

En la gran mayoría de los casos se buscó no hacer implementaciones propias de primitivas criptográficas, sin embargo, en el caso del generador, se hizo un excepción, para darle un poco más de contenido al trabajo. Esto último dado que hacer un generador implica también validarlo con pruebas de aleatoriedad del NIST.

# DISEÑO DE PROGRAMA

## COMPONENTES

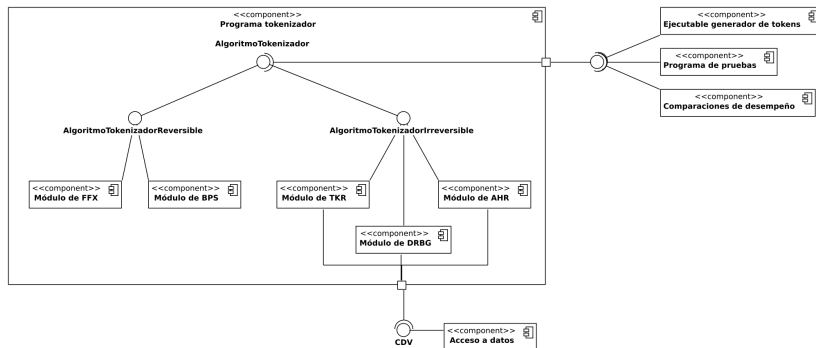


Diagrama de componentes del programa.

# Generación de tokens para proteger los datos de tarjetas bancarias

- └ Algoritmos generadores de *tokens*
  - └ Implementaciones
    - └ Diseño de programa



Diagrama de componentes del programa.

Se muestra la estructura interna del componente del programa tokenizador. Aunque adentro de este hay varios módulos, a las entidades externas solo les interesa comunicarse a través de la interfaz. El acceso a datos se maneja también a través de una interfaz externa al componente; los métodos irreversibles deben tener acceso a esta.

# CONTENIDO

Planteamiento del problema

Planteamiento de la solución

Algoritmos generadores de *tokens*

Conclusiones

Avances logrados 19

Trabajo a futuro 20

# AVANCES LOGRADOS

En TT1 se planeó terminar el primer prototipo dado que es la parte central del proyecto; situación que se consiguió, llegando a implementar 5 algoritmos distintos y realizar pruebas comparativas entre estos. Además, en los algoritmos pertinentes, se realizaron pruebas de aleatoriedad con el fin de respaldar la seguridad de la implementación.

# TRABAJO A FUTURO







# BIBLIOGRAFÍA III



Riccardo Aragona, Riccardo Longo y Massimiliano Sala. “Several proofs of security for a tokenization algorithm”. En: *Appl. Algebra Eng. Commun. Comput.* 28.5 (2017), págs. 425-436. DOI: 10.1007/s00200-017-0313-3. URL: <https://doi.org/10.1007/s00200-017-0313-3>.



Elaine Barker y John Kelsey. *NIST Special Publication 800-90A - Recommendation for Random Number Generation Using Deterministic Random Bit Generators*. 2015. URL: <http://dx.doi.org/10.6028/NIST.SP.800-90Ar1>.

# GENERACIÓN DE TOKENS PARA PROTEGER LOS DATOS DE TARJETAS BANCARIAS

TRABAJO TERMINAL No. 2017-B008

PRESENTAN

DANIEL AYALA ZAMORANO

DAZ23AYALA@GMAIL.COM

LAURA NATALIA BORBOLLA PALACIOS

LN.BORBOLLA42@GMAIL.COM

RICARDO QUEZADA FIGUEROA

QF7.RICARDO@GMAIL.COM

DIRECTORA

DRA. SANDRA DÍAZ SANTIAGO

CIUDAD DE MÉXICO, 9 DE MAYO DE 2018

ESCUELA SUPERIOR DE CÓMPUTO  
INSTITUTO POLITÉCNICO NACIONAL

