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

Trabajo Terminal No. ＿＿＿＿-＿＿＿

Alumnos: Ayala Zamorano Daniel, Borbolla Palacios Laura Natalia, Quezada Figueroa Ricardo*

Director: Dra. Díaz Santiago Sandra

Turno para la presentación del TT: MATUTINO

e-mail: ln.borbolla.42@gmail.com

Resumen – Hoy en día, es bien conocido que los datos de tarjetas bancarias son datos sensibles y por tanto, garantizar su privacidad, es de suma importancia. Una solución que se ha vuelto muy popular es sustituir el dato sensible por un *token*, es decir, un valor que no revela la información original. Lamentablemente, la mayor parte de las soluciones que existen en el mercado son muy poco claras acerca de sus métodos de generación de *tokens*, ya que al no existir un estudio criptográfico formal, no hay ninguna certeza sobre la seguridad de sus métodos. En este trabajo se analizarán e implementarán diversos algoritmos para la generación de *tokens*.

Palabras clave – Criptografía, seguridad, tecnologías web, tarjetas bancarias.

1. Introducción

A finales de los noventas e inicios del 2000, los comerciantes comenzaron a expandir sus horizontes mediante el uso de internet y las tiendas en línea, por lo que el pago con tarjetas bancarias en la red comenzó a ser muy popular; sin embargo, los sitios de comercio electrónico y su conexión con los procesadores de pago estaban pobremente protegidos, de manera que los fraudes relacionados con tarjetas de crédito y el robo de información bancaria crecieron alarmantemente: Visa y MasterCard reportaron pérdidas por fraudes bancarios de 750 millones de dólares en 1988; en el 2000, una década después, esas pérdidas alcanzan los 1.5 mil millones de dólares [8].

Para disminuir los fraudes, en 1999 Visa desarrolló un estándar de seguridad para quienes realizan transacciones bancarias en línea, llamado CISP[1] (uno de los precursores del PCI[2] DSS[3]). Posteriormente, Visa, MasterCard y otras compañías comenzaron a presionar para que se utilizaran las políticas de seguridad, pero en vez de publicar un estándar unificado válido para todas las compañías, cada cual publicó su propia guía; provocando que muy pocas compañías fuesen capaces de satisfacer todos los requerimientos para todas las tarjetas.

Finalmente, en 2004, debuta el PCI DSS: un estándar de seguridad unificado y respaldado por las cinco compañías de tarjetas más importantes; además, hacen que el acatamiento del estándar sea obligatorio para cualquier comerciante y cualquier otra organización que se encuentre en el procesamiento de pagos. Sin embargo, el esquema que se proponía para mantener segura la información era el de protegerla en cualquier sitio, pero esto resultaba muy costoso, pues la información sensible, en especial la bancaria, podría estar en cualquier sección del sistema (ventas, clientes, pagos, etcétera). Es hasta mediados del 2011 cuando se propone cambiar el paradigma y utilizar *tokens* (valores representativos, sin relación al original, que reemplazan a la información valiosa) para aligerar la carga de la seguridad, pues ya no es necesario proteger la información en todos lados, sino solo en donde se tienen los valores originales [8].

Desde su primera publicación, el PCI DSS puso una gran carga en los bancos, procesadores de pagos y quienes aceptan pagos con tarjeta, pues es imperativo satisfacer alrededor de 200 requerimientos [1] para asegurar la robustez y seguridad de los sistemas que tienen contacto con datos bancarios. Varios sistemas surgieron con el propósito de aligerar la carga a los comerciantes y sitios web, pues se encargan de proteger los datos y, ahora, realizar el proceso de generación de *tokens*.

[1] Por sus siglas en inglés, *Cardholder Information Security Program*

[2] Por sus siglas en inglés, *Payment Card Industry.*

[3] Por sus siglas en inglés, *Data Security Standard*.

2. Objetivo

Objetivo general
Generar *tokens* para garantizar la confidencialidad de los datos de las tarjetas bancarias a través del uso de algoritmos criptográficos y no criptográficos.

Objetivos específicos

- Analizar e implementar diversos algoritmos para la generación de *tokens*.
- Diseñar e implementar un servicio web que proporcione el servicio de generación de *tokens* de tarjetas bancarias a, al menos, una tienda en línea.
- Implementar una tienda web que use el servicio de generación de *tokens*.

3. Justificación

Actualmente, en el mercado, algunas de las soluciones existentes son: Shift 4Go, Merchant Link Payment Gateway, Bluepay ToknShield, Braintree, Jack Henry Card Processing Solution, Shift4 TrueTokenization y Thales Tokenization. Empero, ninguna de estas soluciones explica la manera en que realizan el proceso de generación de *tokens* o las pruebas que se hicieron para validar sus algoritmos; pues, aunque el PCI DSS es claro con lo que el sistema tokenizador debe cumplir, no hay un acuerdo sobre cómo obtener dichos *tokens*, pues no existe un estándar sobre su generación y, con algunas excepciones, este proceso no es analizado desde un punto de vista criptográfico [3].

Este es un gran problema, pues no existen puntos de comparación entre las distintas opciones, ya que cada uno publica distintas características. Estas empresas se aprovechan de la desinformación que hay alrededor del problema de la generación de *tokens*, para poder decir que sus métodos son mejores que los de los competidores, haciendo que los usuarios tomen decisiones muy importantes sin tener fundamentos.

Paralelo a esto, la PCI publica un conjunto de guías para la creación de sistema tokenizadores [2], las cuales dan una serie de requerimientos sin especificar nada acerca de cómo implementarlos, esto ha ocasionado que desarrolladores sin un perfil criptográfico, o de seguridad web, propongan soluciones arbitrarias.

En 2014, Díaz Santiago et al. [4], publicaron un estudio criptográfico sobre los sistemas tokenizadores, en donde proponen distintos algoritmos para solucionar el problema de la generación de *tokens*. Este trabajo terminal será de los primeros sistemas tokenizadores en implementar estos algoritmos, y otras opciones no criptográficas; para al final poder ofrecer una comparación de todos los implementados.

Los usuarios potenciales son todos aquellos que requieran proteger los datos bancarios de sus usuarios y realizar transacciones bancarias seguras. Se espera que al publicar las implementaciones, comiencen discusiones sobre la seguridad de los algoritmos utilizados para generar *tokens*.

Este proyecto hará uso de los conocimientos adquiridos en las asignaturas de criptografía, tecnologías web, bases de datos, desarrollo de aplicaciones en red, ingeniería de software, programación orientada a objetos, análisis y diseño orientado a objetos.

4. Productos o resultados esperados

En la Figura 1 se muestra a grandes rasgos la arquitectura del sistema. El sistema tokenizador funciona como intermediario entre el entorno del comerciante y la institución financiera. El usuario final (el propietario del PAN) no tiene ninguna interacción directa ni con el sistema tokenizador, ni con el entorno del banco.

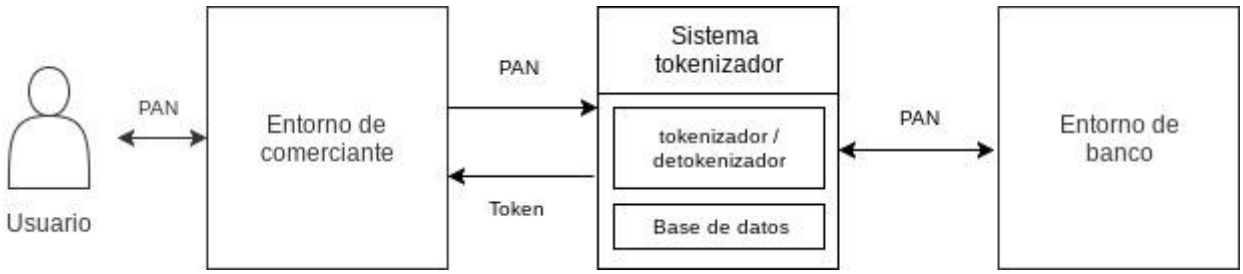

Figura 1. Arquitectura del sistema tokenizador.

Los primeros dos objetivos específicos competen únicamente al sistema tokenizador, mientras que el tercero es referente al entorno del comerciante. Un sistema tokenizador completo debe implementar una comunicación con el entorno del banco para poder realizar transacciones; hacer esto implica demostrar a las instituciones bancarias (Visa, MasterCard, American Express, etc.) que eres una empresa o institución real y confiable, para lo cual se requiere de comprobaciones personales y, en algunos casos, incluso se requiere pagar por el uso de las APIs[4]. Es por esto que nuestro trabajo sólo realizará una simulación del entorno bancario.

La parte central del trabajo es el estudio, implementación y comparación de distintas soluciones al problema de la tokenización. Uno de los productos es, por lo tanto, la publicación de los resultados y conclusiones de las comparaciones realizadas.

Definiremos una API para interactuar con el sistema generador de tokens y para realizar cualquier operación sobre la base de datos. Las operaciones sobre la API estarán disponibles como un servicio web, de manera que cualquier comerciante pueda crear una cuenta y usar el servicio. Además del propio servicio web, se generará un manual de usuario orientado a los desarrolladores de aplicaciones en los entornos de los comerciantes.

Como el sistema tokenizador guardará información sobre los propietarios de tarjetas bancarias, este debe de cumplir con los requerimientos del PCI DSS. Una evaluación real no es viable, pues, además del costo monetario que implica, depende bastante de la propia arquitectura de red utilizada por el sistema (no cubierto en el trabajo); sin embargo, argumentaremos el cumplimiento de los requerimientos pertinentes basándose en los cuestionarios de autoevaluación provistos por la misma organización.

Para poder probar el sistema tokenizador, desarrollaremos una tienda web (entorno de comerciante en figura 1) que haga uso de la API descrita en párrafos anteriores. De esta manera podremos simular varios casos de uso reales que impliquen el uso del sistema tokenizador.

5. Metodología

Para la elaboración de este proyecto, se utilizará una mezcla entre la metodología por prototipos y SDL[5] [9]. Los prototipos que se van a desarrollar se especifican a continuación:

1. **Prototipo 1. Módulo de algoritmos generadores de *tokens*.** Consiste en implementar distintos algoritmos, tanto criptográficos como no criptográficos, para la generación de tokens.
2. **Prototipo 2. Módulo de servicio web.** Consiste en desarrollar la interfaz web del sistema generador de *tokens* y los protocolos de comunicación que permiten a aplicaciones externas interactuar con la API generadora de *tokens*.
3. **Prototipo 3. Unión entre prototipo 1 y prototipo 2.** Consiste en desarrollar las conexiones entre el servicio web y el módulo de algoritmos generadores de *tokens*.
4. **Prototipo 4. Tienda en línea.** Consiste en desarrollar una tienda de comercio en línea.

[4] Por sus siglas en inglés, *Application programming interface.*
[5] Por sus siglas en inglés, *Microsoft Security Development Lifecycle.*

5. **Prototipo 5. Unión entre prototipo 3 y prototipo 4.** Consiste en juntar la tienda de comercio en línea con el módulo de servicio web realizado anteriormente.

El proceso de desarrollo que seguiremos para cada prototipo es el SDL[9] (ciclo de desarrollo de aplicaciones de seguridad, por sus siglas en inglés), el cual contempla las siguientes etapas: entrenamiento, requerimientos, diseño, implementación, verificación, lanzamiento y evaluación de respuesta. Al adecuar la metodología a nuestro trabajo encontramos que las dos últimas etapas (lanzamiento y evaluación de respuesta) se encuentran fuera del alcance del proyecto, por lo que no se llevarán a cabo.

En la primera etapa (entrenamiento) todos los miembros del equipo realizan una investigación sobre la teoría relacionada con el módulo en cuestión; paralelo a esto, se realiza una revisión de las buenas prácticas al momento de hacer software de seguridad. En la segunda etapa (requerimientos) se analizan cuáles son los requerimientos tanto funcionales como no funcionales (en especial los de seguridad y privacidad). La tercera etapa (diseño) contempla el establecimiento de los requerimientos de diseño, el análisis de riesgo, modelado de programa, etc. Finalmente, las fases de implementación y verificación se llevan a cabo de manera iterativa y paralela, de forma que los errores que se encuentren pueden ser corregidos de manera inmediata.

6. Cronogramas

Nombre del alumno:	**Daniel Ayala Zamorano**													
Título del trabajo terminal:	**Generación de *tokens* para proteger los datos de tarjetas bancarias**													
Id.	**Nombre de la tarea**	**2018**												
		Enero	**Febrero**	**Marzo**	**Abril**	**Mayo**	**Junio**	**Julio**	**Agosto**	**Septiembre**	**Octubre**	**Noviembre**	**Diciembre**
P1	**1° prototipo: algoritmos generadores de *tokens***												
1	**Fase de entrenamiento** Estudio de algoritmos criptográficos y no criptográficos.												
2	**Fase de análisis y diseño** Análisis de la implementación de los algoritmos generadores de *tokens.*												
3	**Fase de implementación** Implementación de los algoritmos criptográficos.												
4	**Fase de pruebas** Pruebas de los algoritmos criptográficos.												
5	**Fase de corrección** Correcciones de los algoritmos criptográficos.												
6	**Evaluación de trabajo terminal I**												
P2	**2° prototipo: servicio web**												
7	**Fase de entrenamiento** Estudio de métodos de cifrado para trasnmisiones y protección de datos en comunicación.												
8	**Fase de análisis y diseño** Análisis y diseño de funcionalidades para proteger los datos durante su transmisión.												
9	**Fase de implementación** Implementación de funcionalidades prara proteger los datos durante su transmisión.												
10	**Fase de pruebas** Pruebas de funcionalidades prara proteger los datos durante su transmisión.												
11	**Fase de corrección** Correcciones de funcionalidades prara proteger los datos durante su transmisión.												
P3	**3° prototipo: unión entre P1 y P2**												
12	**Fase de análisis y diseño** Análisis y diseño de unión de funcionalidades de los algoritmos de cifrado generadores de *tokens* con el servicio web.												
13	**Fase de implementación** Implementación de unión de funcionalidades de los algoritmos de cifrado generadores de *tokens* con el servicio web.												
14	**Fase de pruebas** Pruebas de unión de funcionalidades de los algoritmos de cifrado generadores de *tokens* con el servicio web.												
15	**Fase de corrección** Correcciones de unión de funcionalidades de los algoritmos de cifrado generadores de *tokens* con el servicio web.												
P5	**5° prototipo: unión entre P3 y P4**												
16	**Fase de análisis y diseño** Análisis y diseño de comunicación entre la tienda en línea y la API del sistema generador de *tokens*.												
17	**Fase de implementación** Implementación de comunicación entre la tienda en línea y la API del sistema generador de *tokens.*												
18	**Fase de pruebas** Pruebas de comunicación entre la tienda en línea y la API del sistema generador de *tokens.*												
19	**Fase de correción** Correcciones de comunicación entre la tienda en línea y la API del sistema generador de *tokens.*												
20	**Evaluación de trabajo terminal II**												
	Constantes												
21	Realizar manual técnico												
22	Realizar manuales de usuario												

Nombre del alumno:	Laura Natalia Borbolla Palacios
Título del trabajo terminal:	Generación de *tokens* para proteger los datos de tarjetas bancarias

Id.	Nombre de la tarea	2018											
		Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
P1	**1° prototipo: algoritmos generadores de tokens**												
1	**Fase de entrenamiento** Estudio de algoritmos criptográficos y no criptográficos.												
2	**Fase de análisis y diseño** Diseño de la implementación de los algoritmos generadores de *tokens*.												
3	**Fase de implementación** Implementación de los algoritmos de cifrado que preservan el formato.												
4	**Fase de pruebas** Pruebas de los algoritmos de cifrado que preservan el formato.												
5	**Fase de corrección** Correcciones de los algoritmos de cifrado que preservan el formato												
6	**Evaluación de trabajo terminal I**												
P2	**2° prototipo: servicio web**												
7	**Fase de entrenamiento** Estudio de métodos de cifrado de bases de datos y protección de datos en almacenamiento.												
8	**Fase de análisis y diseño** Análisis y diseño del módulo de la base de datos.												
9	**Fase de implementación** Implementación del módulo de la base de datos.												
10	**Fase de pruebas** Pruebas del módulo de la base de datos.												
11	**Fase de corrección** Correcciones del módulo de la base de datos.												
P4	**4° prototipo: tienda en línea**												
12	**Fase de entrenamiento** Estudio de alternativas de implementación de tienda en línea.												
13	**Fase de análisis y diseño** Análisis y diseño del módulo *backend* de la tienda en línea.												
14	**Fase de implementación** Implementación del módulo *backend* de la tienda en línea.												
15	**Fase de pruebas** Pruebas del módulo *backend* de la tienda en línea.												
16	**Fase de corrección** Correcciones del módulo *backend* de la tienda en línea.												
P5	**5° prototipo: unión entre P3 y P4**												
17	**Fase de análisis y diseño** Análisis y diseño de comunicación entre la tienda en línea y la API del sistema generador de *tokens*.												
18	**Fase de implementación** Implementación de comunicación entre la tienda en línea y la API del sistema generador de *tokens.*												
19	**Fase de pruebas** Pruebas de comunicación entre la tienda en línea y la API del sistema generador de *tokens.*												
20	**Fase de correción** Correcciones de comunicación entre la tienda en línea y la API del sistema generador de *tokens.*												
21	**Evaluación de trabajo terminal II**												
	Constantes												
22	Realizar manual técnico												
23	Realizar manuales de usuario												

Nombre del alumno:	**Ricardo Quezada Figueroa**	
Título del trabajo terminal:	**Generación de *tokens* para proteger los datos de tarjetas bancarias**	

Id.	Nombre de la tarea	2018											
		Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
P1	**1° prototipo: algoritmos generadores de *tokens***												
1	**Fase de entrenamiento** Estudio de algoritmos criptográficos y no criptográficos.												
2	**Fase de análisis y diseño** Diseño de la implementación de los algoritmos generadores de *tokens*.												
3	**Fase de implementación** Implementación de los algoritmos no criptográficos.												
4	**Fase de pruebas** Pruebas de los algoritmos no criptográficos.												
5	**Fase de corrección** Correcciones de los algoritmos no criptográficos.												
6	**Evaluación de trabajo terminal I**												
P2	**2° prototipo: servicio web**												
7	**Fase de entrenamiento** Estudio de protocolos de comunicacion para la API web.												
8	**Fase de análisis y diseño** Análisis y diseño de la API web de sistemas generadores de *tokens.*												
9	**Fase de implementación** Implementación de la API web de sistemas generadores de *tokens.*												
10	**Fase de pruebas** Pruebas de la API web de sistemas generadores de *tokens*.												
11	**Fase de corrección** Correcciones de la API web de sistemas generadores de *tokens*.												
P4	**4° prototipo: tienda en línea**												
12	**Fase de entrenamiento** Estudio de alternativas de implementación de tienda en línea.												
13	**Fase de análisis y diseño** Análisis y diseño de módulo *frontend* de la tienda en línea.												
14	**Fase de implementación** Implementación de módulo *frontend* de la tienda en línea.												
15	**Fase de pruebas** Pruebas de módulo *frontend* de la tienda en línea.												
16	**Fase de corrección** Correcciones de módulo *frontend* de la tienda en línea.												
P5	**5° prototipo: unión entre P3 y P4**												
17	**Fase de análisis y diseño** Análisis y diseño de comunicación entre la tienda en línea y la API del sistema generador de *tokens*.												
18	**Fase de implementación** Implementación de comunicación entre la tienda en línea y la API del sistema generador de *tokens.*												
19	**Fase de pruebas** Pruebas de comunicación entre la tienda en línea y la API del sistema generador de *tokens.*												
20	**Fase de correción** Correcciones de comunicación entre la tienda en línea y la API del sistema generador de *tokens.*												
21	**Evaluación de trabajo terminal II**												
	Constantes												
22	Realizar manual técnico												
23	Realizar manuales de usuario												

7. Referencias

[1] PCI Security Standards Council, *PCI DSS: Requirements and Security Assessment Procedures version 3.2*, 2016. Disponible en https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2.pdf?agreement=true&time=1502343652729.

[2] PCI Security Standards Council, *Tokenization Product Security Guidelines version 1.0*, 2015 Disponible en https://www.pcisecuritystandards.org/documents/Tokenization_Product_Security_Guidelines.pdf?agreement=true&time=1502343740439.

[3] R. Krikken, *Towards secure tokenization algorithms and architecture*, Inglaterra, 2011.

[4] S. Diaz-Santiago, L. Rodriguez-Henriquez y D. Chakraborty, *A Cryptographic Study of Tokenization Systems*. International Journal of Information Security, vol. 15, no. 4, pp. 413-432, 2016.

[5] E. Brier, T. Peyrin y J. Stern, *BPS: a format-preserving encryption proposal*. NIST submission, 2010. Disponible en http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/bps/bps-spec.pdf.

[6] M. Bellare, P. Rogaway y T. Spies, *The FFX mode of operation for format-preserving encryption*. NIST Submission, 2010. Disponible en http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ffx/ffx-spec.pdf.

[7] CardHub, *Number of credit cards holders*, 2012. Disponible en http://www.cardhub.com/edu/number-of-credit-cards/.

[8] SearchSecurity, *The history of the PCI DSS standard: A visual timeline*, [online] SeachSecurity, 2013 Disponible en http://searchsecurity.techtarget.com/feature/The-history-of-the-PCI-DSS-standard-A-visual-timeline

[9] Microsoft, *Security Development Lifecycle* [online]. Disponible en: https://www.microsoft.com/en-us/sdl/default.aspx.

8. Alumnos y directora

CARÁCTER: Confidencial
FUNDAMENTO LEGAL: Art. 3, fracc. II, Art. 18, fracc. II y Art. 21, lineamiento 32, fracc. XVII de la L.F.T.A.I.P.G.
PARTES CONFIDENCIALES: No. de boleta y Teléfono.

Daniel Ayala Zamorano.- Alumno de la carrera de Ingeniería en Sistemas Computacionales en ESCOM. Boleta 2015630041, teléfono: 55 35295706, correo electrónico: daz23ayala@gmail.com

Firma:_____

Laura Natalia Borbolla Palacios.- Alumna de la carrera de Ingeniería en Sistemas Computacionales en ESCOM. Boleta 2015630056, teléfono: 55 69656957, correo electrónico: ln.borbolla.42@gmail.com

TURNO PARA LA PRESENTACIÓN DEL TRABAJO TERMINAL:

MATUTINO

Firma:_____

Ricardo Quezada Figueroa.- Alumno de la carrera de Ingeniería en Sistemas Computacionales en ESCOM. Boleta 2015630389, teléfono: 76 26233530, correo electrónico: qf7.ricardo@gmail.com

Firma:_____

Sandra Díaz Santiago.- Doctorado en Ciencias en Computación (CINVESTAV-IPN, 2014). Maestría en Ciencias (Matemáticas) (UAM-Iztapalapa, 2005). Licenciatura en Computación (UAM-Iztapalapa, 1998). Profesor titular en ESCOM (Departamento de Ciencias e Ingeniería de la Computación) desde 2004. Áreas de interés: criptografía, pseudoaleatoriedad, seguridad demostrable. Extensión: 52022, correo electrónico: sdiazs@gmail.com, sdiazsa@ipn.mx

Firma:_____