Mecanismos de tokenizacin, un estudio comparativo

Daniel Ayala Zamorano, Laura Natalia Borbolla Palacios, Ricardo Quezada Figueroa, Sandra Daz Santiago
Escuela Superior de Cmputo, Instituto Politcnico Nacional
Ciudad de Mxico, Mxico
{daz23ayala, ln.borbolla.42, qf7.ricardo, sds.escom}@gmail.com

***Resumen*—La tokenizacin consiste en el reemplazo de informacin sensible por valores sustitutos, llamados tokens, donde el proceso inverso, ir del token a la informacin sensible, no es factible. En los ltimos aos, este proceso se ha vuelto muy popular entre los comercios en lnea, pues les permite liberarse de una parte de las responsabilidades de seguridad adquiridas al manejar nmeros de tarjetas de crdito en un tercero: el proveedor de servicios de tokenizacin. Lamentablemente, existe una gran cantidad de desinformacin alrededor de cmo generar los tokens, producida principalmente por las estrategias publicitarias de las empresas tokenizadoras, en donde cada una intenta convencer al comprador de que su sistema es el mejor sin explicar realmente qu es lo que hacen para generar tokens. Uno de los mensajes ms comunes entre la publicidad es que la criptografa y la tokenizacin son cosas distintas, y que la segunda es mucho ms segura. En este trabajo se explica a detalle en qu consiste la tokenizacin y cul es su relacin con la criptografa, adems, se revisa y compara el desempeo de los mtodos ms comunes para tokenizar.**

***Palabras clave*—tokenizacin, criptografa, seguridad web**

I. Introduccin

Cuando el comercio a travs de Internet comenz a popularizarse, los fraudes de tarjetas bancarias se volvieron un problema alarmante: segn [11], en 2001 se tuvieron prdidas de 1.7 miles de millones de dlares y para 2002 aumentaron a 2.1. Como una medida de proteccin, las principales compaas de tarjetas de crdito publicaron un estndar obligatorio para todos aquellos que procesaran ms de 20 000 transacciones anuales: el PCI DSS (en ingls, *Payment Card Industry Data Security Standard* [14]); este estndar cuenta con una gran cantidad de requerimientos [15], [18], por lo que resulta muy difcil satisfacerlo, especialmente para los negocios pequeos y medianos. A pesar de la publicacin del estndar en 2004, las grandes filtraciones de datos no han cesado: *TJX* en 2006, *Hannaford Bros.* en 2008, *Target* en 2013 y *Home Depot* en 2014, por mencionar algunos ejemplos.

Es en este contexto en el que surge la tokenizacin como alternativa, pues hasta ese momento, la idea era proteger la informacin sensible en donde quiera que se encontrase: si los nmeros de tarjetas de los usuarios se encontraban dispersos en diversas partes de un sistema, haba que proteger todas esas partes. La tokenizacin, en cambio, consiste en concentrar la informacin sensible en un solo lugar para hacer la tarea de proteccin ms sencilla; as, cuando se ingresa un nuevo valor sensible, por ejemplo, el nmero de tarjeta de un usuario, se genera un token ligado a esa informacin; el token se usa en todo el sistema y la informacin sensible se protege en un solo lugar. Un posible adversario con acceso a los tokens no debe poder obtener la informacin sensible a partir de estos.

Una de las ventajas de la tokenizacin es que puede verse como un sistema autnomo, independiente al sistema principal; de esta manera se establece una separacin de responsabilidades: el sistema principal se ocupa de la operacin del negocio (por ejemplo, una tienda en lnea) y el sistema tokenizador se dedica a la proteccin de la informacin sensible. Hoy en da, varias compaas ofrecen servicios de tokenizacin que permiten que los comerciantes se libren casi por completo de cumplir con el PCI DSS. En la Figura 1, se muestra una distribucin bastante comn para un comercio en lnea: el sistema tokenizador guarda la informacin sensible en su base de datos y se encarga de realizar las transacciones bancarias.

La tokenizacin se ha visto una rodeada por una nube de confusin y desinformacin desde sus inicios; la falta de una definicin formal permiti que las campaas publicitarias de las empresas tokenizadoras esparcieran mensajes llenos de imprecisiones que, generalmente, dan a entender que la tokenizacin y la criptografa son dos cosas completamente aisladas la una de la otra, o que la primera es una alternativa (y no una aplicacin) de la segunda. Por ejemplo, lo nico que *Shift4* dice con claridad sobre sus tokens, es que se trata de valores aleatorios, nicos y alfanumricos [17]; para *Braintree*, la nica manera de generar tokens es por mtodos aleatorios [5]; finalmente, para *Securosis* los tokens son valores aleatorios que nada tienen que ver con la criptografa [16]; es fcil observar que la mayora de las soluciones trata a sus mtodos como secretos de compaa, adems, esperan que el trato entre cliente y proveedor est basado en la confianza y no en la comparacin de los propios mtodos tokenizadores.

Como se ver a continuacin por la exposicin de algunos mtodos tokenizadores, la tokenizacin es una aplicacin de la criptografa y, como tal, debe ser analizada con la misma formalidad que las dems herramientas criptogrficas; uno de los errores ms comunes entre las empresas tokenizadoras es considerar que la generacin de nmeros aleatorios no utiliza algoritmos criptogrficos.

En la Seccin II de este trabajo se revisan temas y nociones preliminares necesarias para entender mejor el trabajo; posteriormente, en la Seccin III, se exponen brevemente los mtodos de tokenizacin analizados. Finalmente, en la Seccin IV, se presentan los resultados de las comparaciones de desempeo realizadas y se concluye con una discusin alrededor del tema.

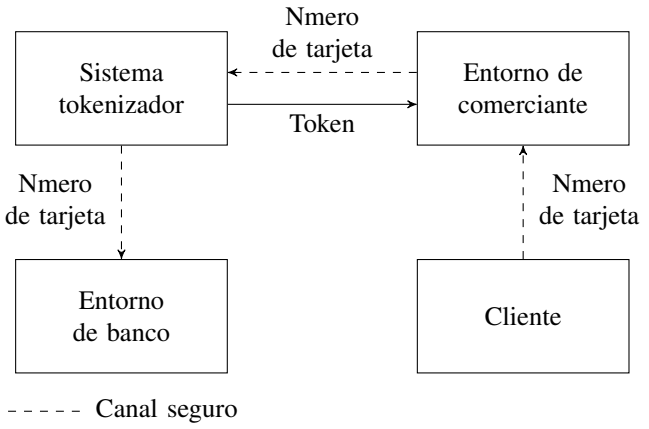


Figura 1. Arquitectura tpica de un sistema tokenizador.

II. Preliminares

II-A. Notacin

Se denotarn a todas las cadenas de bits de longitud n como $\{0,1\}^n$. Para una cadena de smbolos x sobre un alfabeto arbitrario, $|x|$ simboliza la longitud de la cadena. La expresin $f : \mathcal{X} \to \mathcal{Y}$ denota a una funcin f que asigna a cada elemento del conjunto $\mathcal{X}$ un elemento del conjunto $\mathcal{Y}$. La expresin $\mathcal{A} \times \mathcal{B}$, en donde $\mathcal{A}$ y $\mathcal{B}$ son conjuntos, denota al producto cartesiano, esto es, $\mathcal{A} \times \mathcal{B} = \{(a, b) \mid a \in \mathcal{A} \wedge b \in \mathcal{B}\}$.

II-B. Primitivas criptogrficas

Un cifrado por bloques es un cifrado simtrico que se define por la funcin $e : \mathcal{M} \times \mathcal{K} \to \mathcal{C} : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$, donde $\mathcal{M}$ es el espacio de textos en claro; $\mathcal{K}$, el espacio de llaves; $\mathcal{C}$, el espacio de mensajes cifrados; n, el tamao del bloque y k, la llave [12]. Uno de los cifrados por bloques ms usado en la actualidad es AES (*Advanced Encryption Standard*).

Los modos de operacin permiten extender la funcionalidad de los cifrados por bloque para poder operar sobre bloques de informacin de tamao arbitrario: $m : \mathcal{M} \times \mathcal{K} \to \mathcal{C} : \{0,1\}^* \times \{0,1\}^k \to \{0,1\}^*$. Algunos modos de operacin son CBC (*Code Block Chaining*) y CTR (*Counter Mode*) [**?**].

Un cdigo de autenticacin de mensaje (MAC, *Message Authenticathion Code*) es un medio para obtener autenticacin en la transmisin de datos. Se define como una funcin $mac : \mathcal{M} \times \mathcal{K} \to \mathcal{C} : \{0,1\}^* \times \{0,1\}^k \to \{0,1\}^n$, donde $\mathcal{C}$ es el espacio (de cardinalidad 2^n) de cdigos de autenticacin. AES-CBC-MAC es un algoritmo para obtener cdigos de autenticacin que utiliza el cifrado por bloques AES y el modo de operacin CBC.

Una funcin *hash* criptogrfica es un mapeo eficiente de cadenas de longitud arbitraria a cadenas de longitud fija: $h : \mathcal{M} \to \mathcal{E} : \{0,1\}^* \to \{0,1\}^n$ en donde $\mathcal{M}$ es el espacio de mensajes, de cardinalidad infinita, y $\mathcal{K}$ es el espacio de valores *hash*, de cardinalidad 2^n. No debe ser factible regresar al mensaje a partir de su valor *hash*, ni encontrar dos mensajes que produzcan el mismo *hash* [12].

Las redes Feistel son cifrados iterativos que transforman un texto en claro de $2t$ bits denominado (L_0, R_0), en donde L_0 y R_0 son bloques de t bits, en un texto cifrado (R_r, L_r) a travs de un proceso de r rondas. Existen dos generalizaciones de este concepto, las redes alternantes y las redes desbalanceadas; ambas permiten modificar el tamao de las mitades izquierda y derecha: $1 \geq |L_n| \leq 2t$ y $|R_n| = 2t - |L_n|$ [**?**], [**?**].

Un generador determinstico de bits aleatorios (DRBG, *Deterministic Random Bit Generator*) utiliza una primitiva criptogrfica interna, generalmente una funcin *hash* o un cifrado por bloques, y un valor inicial llamado semilla, para producir bits de aspecto aleatorio. Dada la naturaleza determinstica del proceso, a los bits generados se les conoce como *pseudoaleatorios*.

Un cifrado que preserva el formato (en ingls *Format-preserving Encryption*, FPE) es un cifrado simtrico en donde el mensaje en claro y el mensaje cifrado mantienen un formato en comn. Formalmente, de acuerdo a lo definido en [3], se trata de una funcin $E : \mathcal{K} \times \mathcal{N} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$, en donde los conjuntos $\mathcal{K}$, $\mathcal{N}$, $\mathcal{T}$, $\mathcal{X}$ corresponden al espacio de llaves, espacio de formatos, espacio de *tweaks* y el dominio, respectivamente. El proceso de cifrado de un elemento del dominio con respecto a una llave K, un formato N y un *tweak* T se escribe como $E_K^{N,T}(X)$. El proceso inverso es tambin una funcin $D : \mathcal{K} \times \mathcal{N} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$, en donde $D_K^{N,T}\left(E_K^{N,T}(X)\right) = X$.

III. Algoritmos tokenizadores

Como el enfoque de este artculo es ver a la generacin de tokens como un servicio (Figura 1), la interfaz para los procesos de tokenizacin y detokenizacin, desde el punto de vista de los usuarios del servicio, es sumamente simple: el proceso para generar tokens es una funcin $E : \mathcal{X} \to \mathcal{Y}$ y el proceso para regresar a los nmeros de tarjeta es simplemente la funcin inversa $D : \mathcal{Y} \to \mathcal{X}$, en donde $\mathcal{X}$ y $\mathcal{Y}$ son los espacios de nmeros de tarjetas y tokens, respectivamente.

Los nmeros de tarjetas bancarias cuentan con entre 12 y 19 dgitos, y se encuentran normados por el estndar ISO/IEC-7812 [10]. El ltimo dgito se trata de un cdigo de verificacin calculado mediante el algoritmo de Luhn; este dgito hace que ALGORITMODELUHN(x) = 0. Para poder diferenciar a los nmeros de tarjeta vlidos de los tokens se establece que, para los tokens, el dgito verificador haga que ALGORITMODELUHN(x) = 1 [7].

III-A. Clasificacin del PCI SSC

El PCI SSC (*Payment Card Industry Security Standard Council*) establece en sus guas de tokenizacin la siguiente clasificacin para los algoritmos tokenizadores [13]:

- Mtodos reversibles. Aquellos para los cuales es posible obtener el nmero de tarjeta a partir del token.
  - Criptogrficos. Utilizan un esquema de cifrado simtrico: el nmero de tarjeta y una llave entran al mecanismo de tokenizacin para obtener un token; el token y la misma llave entran al mecanismo de detokenizacin para obtener el nmero de tarjeta original.

- No criptogrficos. Ocupan una base de datos para guardar las relaciones entre nmeros de tarjetas y tokens; el proceso de detokenizacin simplemente es una consulta a la base de datos.

- Mtodos irreversibles. Aquellos en los que no es posible obtener el nmero de tarjeta original a partir del token.
  - Autenticable. Permiten validar cuando un token dado corresponde a un nmero de tarjeta dado.
  - No autenticable. No permiten hacer la validacin anterior.

III-B. Algoritmos implementados

FFX y BPS. Cifrados que preservan el formato. En [8] el NIST (*National Institute of Standards and Technology*) los estandariza, denominndolos FF1 y FF3, respectivamente.

FFX (*Format-preserving Feistel-based Encryption*) fue presentado en [4] por Mihir Bellare, Phillip Rogaway y Terence Spies. En su forma ms general, se compone de una serie de parmetros que permiten cifrar cadenas de cualquier longitud en cualquier alfabeto; los autores tambin proponen dos formas ms especficas (dos colecciones de parmetros) para alfabetos binarios y alfabetos decimales: A2 y A10, respectivamente. FFX A10 usa una red Feistel alternante junto con una adaptacin de AES-CBC-MAC (usada como funcin de ronda) para lograr preservar el formato.

BPS fue diseado por Eric Brier, Thomas Peyrin y Jacques Stern [6]. Se conforma de 2 partes: un cifrado interno BC que se encarga de cifrar bloques de longitud fija; y un modo de operacin especial, encargado de extender la funcionalidad de BC y permitir cifrar cadenas de mayor longitud.

TKR. En [7] se analiza formalmente el problema de la generacin de tokens y se propone un algoritmo que no est basado en cifrados que preservan el formato. Hasta antes de la publicacin de este documento, los nicos mtodos para generar tokens cuya seguridad estaba formalmente demostrada eran los basados en cifrados que preservan el formato.

El algoritmo propuesto usa un cifrado por bloques para generar tokens pseudoaleatorios y almacena en una base de datos la relacin original de estos con los nmeros de tarjetas. El proceso de detokenizacin es simplemente una consulta sobre la base de datos. El modo de generacin de tokens pseudoaleatorios es similar a la opracin de un DRBG: como semilla se mantiene un contador interno que es operado a lo largo de las distintas llamadas como el modo de operacin de contador; los bits generados pasan por una etapa de interpretacin que produce tokens vlidos.

AHR. En 2017, Longo, Aragona y Sala [1] propusieron un algoritmo que denominaron *hbrido reversible* que est basado en un cifrador por bloques y utiliza una base de datos para almacenar las relaciones entre nmero de tarjeta y token.

Las entradas del algoritmo son la parte del nmero de tarjeta a cifrar y una entrada adicional (por ejemplo, la fecha) que permite que se tengan varios tokens relacionados con la misma tarjeta. Como se desea obtener un token que tenga el mismo nmero de dgitos que la tarjeta ingresada, se utiliza un mtodo llamado *caminata cclica* para asegurarse de que el texto cifrado pertenezca al espacio del texto en claro.

Tabla I
COMPARACIN DE TIEMPOS DE TOKENIZACIN.

	Tokenizacin (μs)	Detokenizacin (μs)
FFX	83	61
BPS	573	315
TKR	4648	281
AHR	5554	657
DRBG	4649	431

DRBG. *Deterministic Random Bit Generator*. Probablemente este mtodo es el ms directo para generar tokens. La idea es producir una cadena binaria aleatoria con un DRBG e interpretarla para que tenga el formato de un token. Para este trabajo se hizo la implementacin de dos DRBG: uno basado en SHA (*Secure Hash Algorithm*) y el otro basado en un cifrado por bloques; ambos definidos en el estndar del NIST 800-90A [2].

El mtodo que utiliza como mecanismo interno a una funcin *hash* consiste en ir concatenando de forma consecutiva los valores *hash* derivados de la semilla e ir incrementando el valor de esta. El mtodo basado en un cifrador por bloques utiliza el modo de operacin CTR, en donde la semilla juega el papel de vector de inicializacin. En ambos casos, la seguridad se basa en que la semilla sea un valor secreto.

Segn la clasificacin del PCI (Seccin III-A), FFX y BPS son algoritmos reversibles criptogrficos, ya que al ser cifrados que preservan el formato, funcionan como esquema de cifrado simtrico. TKR, AHR y DRBG son, contradictoriamente, reversibles no criptogrficos, pues necesitan de una base de datos para guardar las relaciones entre nmeros de tarjeta y tokens.

IV. RESULTADOS Y CONCLUSIONES

En la Tabla I y la Figura 2a se muestran los resultados en tiempo de las ejecuciones de los algoritmos presentados en la Seccin III-B. Estos se llevaron a cabo en una computadora con las siguientes caractersticas:

- **Procesador:** Intel i5-7200U (2.5 GHz) de 4 ncleos.
- **Sistema operativo:** Arch Linux, kernel 4.17.
- **Base de datos:** MariaDB 10.1.
- **Compilador:** GCC 8.1.1.

El procesador ocupado soporta los conjuntos de intrucciones de Intel AES-NI y RD-SEED [9]. Todas los algoritmos tokenizadores que usan un cifrado por bloques utilizan una implementacin de AES con las instrucciones a nivel de hardware. El DRBG implementado ocupa la instruccin RD-RAND como fuente de entropa.

La comparacin de la Figura 2a muestra como los dos algoritmos reversibles, FFX y BPS, son considerablemente ms rpidos que los tres irreversibles: TKR, AHR y DRBG. Los reversibles estn en el rango de 60 a 170 microsegundos, mientras que los irreversibles estn en alrededor de los 5500 microsegundos. Tambin es posible observar cmo, para los

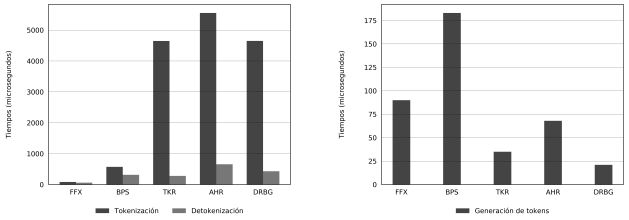
mtodos irreversibles, el proceso de detokenizacin es mucho ms rpido que la generacin de tokens. Estos dos resultados dejan ver un poco la carga de las operaciones en los mtodos irreversibles: la tokenizacin involucra una consulta a la base, la generacin del token y una insercin; la detokenizacin solamente es una consulta.

El que FFX y BPS sean ms rpidos puede resultar un poco contraintuitivo, pues la generacin de tokens reversibles involucra ms operaciones; es por esto que en la Figura 2b se muestran los tiempos de la generacin de tokens solamente, sin tomar en cuenta tiempos de acceso a base de datos. En este caso, el ms veloz es DRBG, seguido de ceca por TKR y AHR; los dos reversibles van al ltimo.

Adems de los tiempos de ejecucin, tambin es importante sealar que los irreversibles, al operar como funciones de un solo sentido, son ms seguros que los reversibles: un atacante con acceso a la llave de cifrado puede obtener el nmero de tarjeta correspondiente si se trata de un mtodo reversible, mientras que con un mtodo irreversible necesita tambin acceso a la base de datos.

La denominacin *no criptogrficos*, de la clasificacin del PCI SSC (Seccin III-A), resulta totalmente confusa, pues en realidad todos los mtodos conocidos que caen en esa categora utilizan primitivas criptogrficas. La segunda categora (irreversibles) carece de utilidad para aplicaciones que procesan pagos con tarjetas de crdito, pues la habilidad de regresar al nmero de tarjeta a partir de su token es uno de los requerimientos principales para los sistemas tokenizadores. Por lo anterior, en este trabajo se propone una clasificacin distinta:

- Mtodos criptogrficos. Todos aquellos que ocupan herramientas critogrficas.
  - Reversibles. Usan un esquema de cifrado simtrico: el nmero de tarjeta y una llave entran al mecanismo de tokenizacin para obtener un token; el token y la misma llave entran al mecanismo de detokenizacin para obtener el nmero de tarjeta original.
  - Irreversibles. Hacen uso de herramientas criptogrficas para generar el token de un nmero de tarjeta y herramientas externas, como una base de datos, para guardar las relaciones entre tokens y nmeros de tarjetas.
- Mtodos no criptogrficos. Aquellos posibles mtodos que no necesitan herramientas relacionadas con la criptografa; por ejemplo, un generador de nmeros realmente aleatorio (TRNG, *True Random Number Generator*).



(a) Tokenizacin y detokenizacin (b) Generacin de tokens

Figura 2. Comparaciones de tiempos.

REFERENCIAS

[1] R. Aragona, R. Longo, and M. Sala. Several proofs of security for a tokenization algorithm. *Appl. Algebra Eng. Commun. Comput.*, 28(5):425–436, 2017.

[2] E. Barker and J. Kelsey. NIST special publication 800-90a - recommendation for random number generation using deterministic random bit generators, 2015.

[3] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format-preserving encryption. In M. J. J. Jr., V. Rijmen, and R. Safavi-Naini, editors, *Selected Areas in Cryptography, 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers*, volume 5867 of *Lecture Notes in Computer Science*, pages 295–312. Springer, 2009.

[4] M. Bellare, P. Rogaway, and T. Spies. The FFX mode of operation for format-preserving encryption. 2009. Presentado al NIST para estandarizacin.

[5] Braintree. Tokenization secures CC data and meet PCI compliance requirements. https://www.braintreepayments.com/blog/using-tokenization-to-secure-credit-card-data-and-meet-pci-compliance-requirements/. Consultado en marzo de 2018.

[6] E. Brier, T. Peyrin, and J. Stern. BPS: a format-preserving encryption proposal. 2010. Presentado al NIST para estandarizacin.

[7] S. Diaz-Santiago, L. M. Rodrguez-Henrquez, and D. Chakraborty. A cryptographic study of tokenization systems. *Int. J. Inf. Sec.*, 15(4):413–432, 2016.

[8] M. Dworkin. NIST special publication 800-38g - recommendation for block cipher modes of operation: Methods for format-preserving encryption, 2016.

[9] G. Hofemeier and R. Chesebrough. Introduction to intel AES-NI and intel secure key instructions. https://software.intel.com/sites/default/files/m/d/4/1/d/8/Introduction_to_Intel_Secure_Key_Instructions.pdf. Consultado en abril de 2018.

[10] International Organization for Standarization. *ISO/IEC 7812*. 5 edition, 2017.

[11] J. S. Kiernan. Credit card and debit card fraud statistics. https://wallethub.com/edu/credit-debit-card-fraud-statistics/25725/. Consultado en marzo de 2018.

[12] A. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[13] Payment Card Industry Security Standards Council. Tokenization product security guidelines irreversible and reversible tokens, 2015.

[14] Payment Card Industry Security Standards Council. Data security standard - version 3.2, 2016.

[15] SearchSecurity Staff. The history of the PCI DSS standard: A visual timeline. https://searchsecurity.techtarget.com/feature/The-history-of-the-PCI-DSS-standard-A-visual-timeline. Consultado en marzo de 2018.

[16] Securosis. Understanding and selecting a tokenization solution. https://securosis.com/assets/library/reports/Securosis_Understanding_Tokenization_V.1_.0_.pdf. Consultado en febrero de 2018.

[17] Shift4 Payments. The history of truetokenization. https://www.shift4.com/dotn/4tify/trueTokenization.cfm. Consultado en agosto de 2018.

[18] UK Cards Association. What is PCI DSS? http://www.theukcardsassociation.org.uk/security/What_is_PCI\%20DSS.asp. Consultado en febrero de 2018.