

# CIFRADOS QUE PRESERVAN EL FORMATO

TRABAJO TERMINAL No. 2017-B008

DANIEL AYALA ZAMORANO

DAZ23AYALA@GMAIL.COM

LAURA NATALIA BORBOLLA PALACIOS

LN.BORBOLLA42@GMAIL.COM

RICARDO QUEZADA FIGUEROA

QF7.RICARDO@GMAIL.COM

ESCUELA SUPERIOR DE CÓMPUTO  
INSTITUTO POLITÉCNICO NACIONAL



FEBRERO DE 2018

# CONTENIDO

## Cifrados que preservan el formato

Introducción 3

Clasificación 6

# CONTENIDO

## Cifrados que preservan el formato

Introducción 3

Clasificación 6

## FFX

Redes Feistel 8

Definición de parámetros 10

# CONTENIDO

## Cifrados que preservan el formato

Introducción 3

Clasificación 6

## FFX

Redes Feistel 8

Definición de parámetros 10

## BPS

Introducción a BPS 11

Cifrador interno BC 12

Modo de operación 23

Conclusiones y recomendaciones 27

# CONTENIDO

## Cifrados que preservan el formato

Introducción 3

Clasificación 6

## FFX

Redes Feistel 8

Definición de parámetros 10

## BPS

Introducción a BPS 11

Cifrador interno BC 12

Modo de operación 23

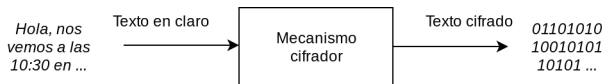
Conclusiones y recomendaciones 27

## Anatomía de un número de tarjeta

# INTRODUCCIÓN A FPE

## PLANTEAMIENTO DEL PROBLEMA

Los cifradores estándar (por ejemplo, AES) convierten un mensaje en una cadena binaria.





# Cifrados que preservan el formato

## └ Cifrados que preservan el formato

### └ Introducción

### └ Introducción a FPE

#### INTRODUCCIÓN A FPE

##### PLANTEAMIENTO DEL PROBLEMA

Los cifradores estándar (por ejemplo, AES) convierten un mensaje en una cadena binaria.



La cual, al ser interpretada, se compone principalmente de caracteres no imprimibles.



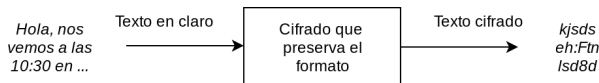
Generalización de ejemplo a pdf e imágenes: no se espera que un pdf cifrado siga siendo un pdf válido; o que una imagen cifrada se siga pudiendo ver con un visor de imágenes.



# INTRODUCCIÓN A FPE

## PLANTEAMIENTO DEL PROBLEMA

El objeto de los cifrados que preservan el formato (*Format-preserving Encryption*, FPE) es convertir un texto en claro con un formato dado en un texto cifrado con el mismo formato.



# Cifrados que preservan el formato

## └ Cifrados que preservan el formato

### └ Introducción

#### └ Introducción a FPE

#### INTRODUCCIÓN A FPE

##### PLANTEAMIENTO DEL PROBLEMA

El objeto de los cifrados que preservan el formato (*Format-preserving Encryption, FPE*) es convertir un texto en claro en un formato dado en un texto cifrado con el mismo formato.

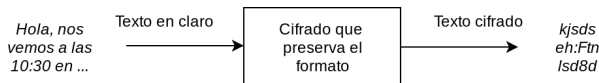


Para el ejemplo de la figura (un formato de los caracteres ASCII imprimibles), no existen muchas aplicaciones reales; es solo con fines ilustrativos.

# INTRODUCCIÓN A FPE

## PLANTEAMIENTO DEL PROBLEMA

El objeto de los cifrados que preservan el formato (*Format-preserving Encryption*, FPE) es convertir un texto en claro con un formato dado en un texto cifrado con el mismo formato.



Formalmente, se busca obtener una permutación

$$\mathcal{E} : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}$$

que sea difícil de invertir sin el conocimiento de la llave.

# Cifrados que preservan el formato

- └ Cifrados que preservan el formato
  - └ Introducción
    - └ Introducción a FPE

El objeto de los cifrados que preservan el formato (*Format-preserving Encryption, FPE*) es convertir un texto en claro con un formato dado en un texto cifrado con el mismo formato.



Formalmente, se busca obtener una permutación

$$\mathcal{E} : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}$$

que sea difícil de invertir sin el conocimiento de la clave.

En realidad la ecuación es casi la definición de cifrado común; lo único que hay que hacer notar es que *el formato* de  $\mathcal{X}$  se debe poder reproducir en el texto cifrado.

También hay que hacer notar cómo, si se ve al formato como una cadena binaria, entonces los cifradores estándar son por sí mismos cifrados que preservan el formato.

# INTRODUCCIÓN A FPE

## APLICACIONES

La utilidad de los cifrados que preservan el formato se centra principalmente en *agregar* seguridad a sistemas y protocolos que ya se encuentran en un entorno de producción. Estos son algunos ejemplos de dominios comunes en FPE:

La utilidad de los cifrados que preservan el formato se centra principalmente en agregar seguridad a sistemas y protocolos que ya se encuentran en un entorno de producción. Estos son algunos ejemplos de dominios comunes en FPE:

- Cifrados que preservan el formato
  - Cifrados que preservan el formato
    - Introducción
      - Introducción a FPE

Hablar de en qué contextos se quiere preservar el formato de este tipo de datos: bases de datos que los usan como índices, o aplicaciones en las que se usan como identificadores.

# INTRODUCCIÓN A FPE

## APLICACIONES

La utilidad de los cifrados que preservan el formato se centra principalmente en *agregar* seguridad a sistemas y protocolos que ya se encuentran en un entorno de producción. Estos son algunos ejemplos de dominios comunes en FPE:

- Números de tarjetas de crédito.

5827 5423 6584 2154  $\rightarrow$  6512 8417 6398 7423

# INTRODUCCIÓN A FPE

## APLICACIONES

La utilidad de los cifrados que preservan el formato se centra principalmente en *agregar* seguridad a sistemas y protocolos que ya se encuentran en un entorno de producción. Estos son algunos ejemplos de dominios comunes en FPE:

- ▶ Números de tarjetas de crédito.

5827 5423 6584 2154  $\rightarrow$  6512 8417 6398 7423

- ▶ Números de teléfono.

55 55 54 75 65  $\rightarrow$  55 55 12 36 98



# INTRODUCCIÓN A FPE

## APLICACIONES

La utilidad de los cifrados que preservan el formato se centra principalmente en *agregar* seguridad a sistemas y protocolos que ya se encuentran en un entorno de producción. Estos son algunos ejemplos de dominios comunes en FPE:

- ▶ Números de tarjetas de crédito.

5827 5423 6584 2154  $\rightarrow$  6512 8417 6398 7423

- ▶ Números de teléfono.

55 55 54 75 65  $\rightarrow$  55 55 12 36 98

- ▶ CURP.

GHUJ887565HGBTOK01  $\rightarrow$  QRGH874528JUH Y01

## Cifrados que preservan el formato

### └ Cifrados que preservan el formato

#### └ Introducción

#### └ Introducción a FPE

La utilidad de los cifrados que preservan el formato se centra principalmente en agregar seguridad a sistemas y protocolos que ya se encuentran en un entorno de producción. Estos son algunos ejemplos de dominios comunes en FPE:

- Números de tarjetas de crédito.  
5827 5423 6584 2154 → 6512 8417 6398 7423
- Números de teléfono.  
55 55 54 75 65 → 55 55 12 36 98
- CURP.  
QBJ887565H2GT001 → QNG887452BJUY01

En algunos casos, se debe mantener cierta parte del texto en claro en el texto cifrado (más adelante se hablará de las tarjetas de crédito), como en el ejemplo del teléfono.

# CLASIFICACIÓN DE FPE

En [1], Phillip Rogaway clasifica a los algoritmos que preservan el formato según el tamaño del espacio de mensajes ( $N = |X|$ ).

- ▶ Espacios minúsculos.
- ▶ Espacios pequeños.
- ▶ Espacios grandes.

# CLASIFICACIÓN DE FPE

En [1], Phillip Rogaway clasifica a los algoritmos que preservan el formato según el tamaño del espacio de mensajes ( $N = |X|$ ).

- ▶ Espacios minúsculos.
- ▶ Espacios pequeños.
- ▶ Espacios grandes.

**Espacios minúsculos:** el espacio es tan pequeño que es aceptable gastar  $O(N)$  en el proceso de cifrado.

Por ejemplo, se puede inicializar una tabla de  $N$  elementos, y realizar las operaciones de cifrado y descifrado con consultas. Para esto se pueden ocupar métodos como el *Knuth shuffle* o un cifrado con prefijo.

# Cifrados que preservan el formato

- └ Cifrados que preservan el formato
  - └ Clasificación
    - └ Clasificación de FPE

En [1], Phillip Rogaway clasifica a los algoritmos que preservan el formato según el tamaño del espacio de mensajes ( $N = |X|$ ).

- Espacios minúsculos.
- Espacios pequeños.
- Espacios grandes.

**Espacios minúsculos:** el espacio es tan pequeño que es aceptable gastar  $O(N)$  en el proceso de cifrado.

Por ejemplo, se puede inicializar una tabla de  $N$  elementos, y realizar las operaciones de cifrado y descifrado con consultas. Para esto se pueden ocupar métodos como el *Knuth shuffle* o un cifrado con prefijo.

*Knuth shuffle* (también *Fisher-Yates shuffle*) genera una permutación pseudoaleatoria de una secuencia finita.

También había un *permutation numbering*; no lo entiendo.

El cifrado con prefijo utiliza un cifrador estándar para cifrar todos los elementos y después utiliza un ordenamiento para determinar la permutación.

# CLASIFICACIÓN DE FPE

En [1], Phillip Rogaway clasifica a los algoritmos que preservan el formato según el tamaño del espacio de mensajes ( $N = |X|$ ).

- ▶ Espacios minúsculos.
- ▶ Espacios pequeños.
- ▶ Espacios grandes.

**Espacios pequeños:** el espacio no es más grande que  $2^w$ , en donde  $w$  es el tamaño de bloque del cifrado subyacente. Para AES, en donde  $w = 128$ ,  $N = 2^{128} \approx 10^{38}$ .

En este esquema, el mensaje se ve como una cadena de  $n$  elementos pertenecientes a un alfabeto de cardinalidad  $m$  (i. e.  $N = m^n$ ).

Por ejemplo, para números de tarjetas de crédito,  $n \approx 16$  y  $m = 10$ , por lo que  $N = 10^{16}$  (diez mil trillones); lo cual es aproximadamente  $2.93 \times 10^{-21} \%$  de  $2^{128}$ .

# Cifrados que preservan el formato

- └ Cifrados que preservan el formato
  - └ Clasificación
    - └ Clasificación de FPE

## CLASIFICACIÓN DE FPE

En [1], Phillip Rogaway clasifica a los algoritmos que preservan el formato según el tamaño del espacio de mensajes ( $N = |X|$ ).

- Espacios minúsculos.
- Espacios pequeños.
- Espacios grandes.

**Espacios pequeños:** el espacio no es más grande que  $2^w$ , en donde  $w$  es el tamaño de bloque del cifrado subyacente. Para AES, en donde  $w = 128$ ,  $N = 2^{128} \approx 10^{38}$ .

En este esquema, el mensaje se ve como una cadena de  $n$  elementos pertenecientes a un alfabeto de cardinalidad  $m$  (i. e.  $N = m^n$ ).

Por ejemplo, para números de tarjetas de crédito,  $n \approx 16$  y  $m \approx 10$ , por lo que  $N \approx 10^{16}$  (diez mil trillones); lo cual es aproximadamente  $2.93 \times 10^{-21} \%$  de  $2^{128}$ .

*Pequeño* en comparación con el tamaño del bloque, no con estándares humanos (e. g. el universo solo lleva  $\approx 2^{86}$  nanosegundos de existencia).

Las técnicas para los espacios pequeños pueden ser usadas en los espacios minúsculos, aunque con menos garantías de seguridad que con las técnicas propias.

# CLASIFICACIÓN DE FPE

En [1], Phillip Rogaway clasifica a los algoritmos que preservan el formato según el tamaño del espacio de mensajes ( $N = |X|$ ).

- ▶ Espacios minúsculos.
- ▶ Espacios pequeños.
- ▶ Espacios grandes.

**Espacios grandes:** el espacio es más grande que  $2^w$ .

Para estos casos, el mensaje se ve como una cadena binaria. Las técnicas utilizadas incluyen cualquier cifrado cuya salida sea *de la misma* longitud que la entrada (e. g. los TES: CMC, EME, HCH, etc.).



## Cifrados que preservan el formato

### └ Cifrados que preservan el formato

#### └ Clasificación

#### └ Clasificación de FPE

En [1], Phillip Rogaway clasifica a los algoritmos que preservan el formato según el tamaño del espacio de mensajes ( $N = |X|$ ).

- Espacios minúsculos.
- Espacios pequeños.
- Espacios grandes.

**Espacios grandes:** el espacio es más grande que  $2^n$ .

Para estos casos, el mensaje se ve como una cadena binaria. Las técnicas utilizadas incluyen cualquier cifrado cuya salida sea de la misma longitud que la entrada (e. g. los TES: CMC, EME, HCH, etc.).

Los ejemplos más clásicos que se usan aquí son los cifradores para sectores de discos duros.

# FFX

## INTRODUCCIÓN

FFX (*Format-preserving, Feistel-based encryption*) es un modo de operación para lograr FPE en espacios pequeños.

El mecanismo general usado en FFX son las redes Feistel, aplicadas sobre alfabetos arbitrarios.

La primera versión fue presentada al NIST en [2], en noviembre de 2009; en la segunda versión [3], se agregó el perfil de parámetros FF2, para cadenas binarias.

## Cifrados que preservan el formato

└ FFX

└ FFX

FFX  
bernardocede

FFX (*Format-preserving, Feistel-based encryption*) es un modo de operación para lograr FPE en espacios pequeños.

El mecanismo general usado en FFX son las redes Feistel, aplicadas sobre alfabetos arbitrarios.

La primera versión fue presentada al NIST en [2], en noviembre de 2009; en la segunda versión [3], se agregó el perfil de parámetros FF2, para cadenas binarias.

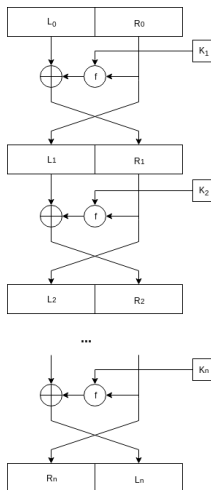
Fue creado por Rogaway, Bellare y Spies. El último, asociado a voltage security, había propuesto antes FFSEM. FFX es una generalización de FFSEM (las redes Feistel de este último solo funcionan en alfabetos binarios).

Terence Spies fue quien empezó a usar el término de cifrados que preservan el formato.

**TODO:** ¿Cómo estuvo lo de las vulnerabilidades encontradas?

# FFX

## REDES FEISTEL



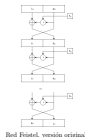
Red Feistel, versión original.

## Cifrados que preservan el formato

└ FFX

└┐ Redes Feistel

└┐ FFX



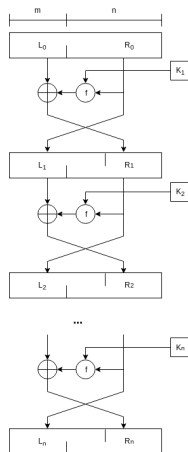
Una de las características más importantes es que no se necesita una función de ronda inversa. La operación de descifrado se obtiene despejando, solamente. Por ejemplo, en una última instancia solamente se conoce  $L_n$  y  $R_n$ :

$$R_{n-1} = L_n$$

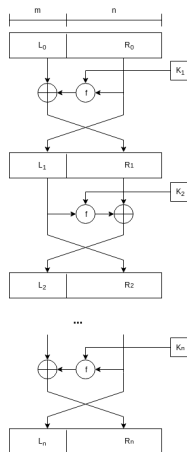
$$L_{n-1} = f_{k_n}(R_{n-1}) \oplus R_n$$

# FFX

## REDES FEISTEL



(a) Redes desbalanceadas.



(b) Redes alternantes.

Generalizaciones de las redes Feistel.

# Cifrados que preservan el formato

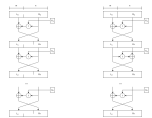
- FFX

- Redes Feistel

- FFX

FFX

Ramas Feistel.



(a) Redes desbalanceadas.

(b) Redes alternantes.

Generalizaciones de las redes Feistel.

Puede haber distintos grados de desbalanceo. En caso de que este sea igual a cero, la red está balanceada (versión original).

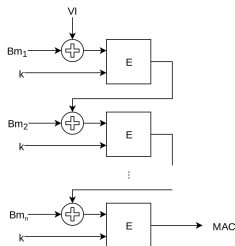
Las desbalanceadas llevan el costo extra de hacer la partición del bloque en cada ronda. Las alternantes necesitan de dos funciones de ronda.

Después de terminar explicación, regresar a 1 y poner ejemplo con alfabeto binario y con alfabeto decimal. FFSEM estaba pensado para alfabetos binarios solamente, por lo que había que utilizar caminata cíclica en la función de ronda.

# FFX

## FUNCIÓN DE RONDA

La función de ronda propuesta en [3] es AES CBC MAC, aunque también puede ser usado cualquier otro cifrador por bloques o función hash.



CBC MAC.

La salida de la primitiva utilizada determina el tamaño del espacio de mensajes aceptado.



## Cifrados que preservan el formato

└ FFX

└┐ Redes Feistel

└┐ FFX

## FFX

Función de ronda.

La función de ronda propuesta en [3] es AES CBC MAC, aunque también puede ser usado cualquier otro cifrador por bloques o función hash.



La salida de la primitiva utilizada determina el tamaño del espacio de mensajes aceptado.

Esta última característica es lo que permite que FFX funciones sobre cadenas de cualquier longitud. Por ejemplo, se puede poner un cifrador de flujo en el lugar de la función de ronda.

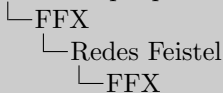
# FFX

## FUNCIÓN DE RONDA

La salida de la función de ronda se debe adaptar al alfabeto usado:

- ▶ En el caso de un alfabeto binario, tomar solamente el número de bits que la red Feistel requiere.
- ▶ En el caso de un alfabeto de caracteres, se debe interpretar de manera que produzca el número de caracteres necesarios. La forma más simple para hacer esto es tomar la salida de la primitiva módulo  $m^n$ , en donde  $m$  es la cardinalidad del alfabeto y  $n$  el número de caracteres ocupados por la red. En [3] se propone partir la salida de CBC MAC en dos: usar la primera mitad para producir  $n/2$  caracteres, y la segunda mitad para los restantes.

# Cifrados que preservan el formato



## FFX

Función de ronda

La salida de la función de ronda se debe adaptar al alfabeto usado:

- En el caso de un alfabeto binario, tomar solamente el número de bits que la red Feistel requiere.
- En el caso de un alfabeto de caracteres, se debe interpretar de manera que produzca el número de caracteres necesarios. La forma más simple para hacer esto es tomar la salida de la primitiva módulo  $m^n$ , en donde  $m$  es la cardinalidad del alfabeto y  $n$  el número de caracteres ocupados por la red. En [3] se propone partir la salida de CBC MAC en dos: usar la primera mitad para producir  $n/2$  caracteres, y la segunda mitad para los restantes.

**TODO:** En el caso de los alfabetos de caracteres, ¿cuál es la diferencia en cuestión de seguridad entre uno y otro esquema? Creo que, el segundo es mejor, dado que toma en cuenta más bits de la salida de la primitiva.

Antes de pasar a la siguiente sección, hablar de los *tweaks* y de los números de rondas recomendados.

# FFX

## PARÁMETROS

Los siguientes 9 parámetros hacen de FFX un esquema muy general, que puede ser utilizado para cifrar cadenas de *cualquier* longitud.

### 1. *Radix*

Número que determina el alfabeto usado.

$C = \{0, 1, \dots, \text{radix} - 1\}$ . Tanto el texto en claro como el texto cifrado pertenecen a este alfabeto.

### 2. Longitudes

El rango permitido para longitudes de mensaje.

### 3. Llaves

El conjunto que representa al espacio de llaves.

# Cifrados que preservan el formato

└ FFX

└ Definición de parámetros

└ FFX

## FFX

### Parámetros

Los siguientes 9 parámetros hacen de FFX un esquema muy general, que puede ser utilizado para cifrar cadenas de cualquier longitud.

#### 1. *Radix*

Número que determina el alfabeto usado.

$C = \{0, 1, \dots, \text{radix} - 1\}$ . Tanto el texto en claro como el texto cifrado pertenecen a este alfabeto.

#### 2. *Longitudes*

El rango permitido para longitudes de mensaje.

#### 3. *Llaves*

El conjunto que representa al espacio de llaves.

Los parámetros se condicionan unos a otros: el rango permitido depende de la función de ronda utilizada.

# FFX

## PARÁMETROS

Los siguientes 9 parámetros hacen de FFX un esquema muy general, que puede ser utilizado para cifrar cadenas de *cualquier* longitud.

### 4. *Tweaks*

El conjunto que representa al espacio de *tweaks*.

### 5. Suma

El operador utilizado en la red Feistel para combinar la parte izquierda con la salida de la función de ronda.

### 6. Método

El tipo de red Feistel a ocupar: desbalanceada o alternante.

## Cifrados que preservan el formato

└ FFX

└ Definición de parámetros

└ FFX

### FFX

#### Parámetros

Los siguientes 9 parámetros hacen de FFX un esquema muy general, que puede ser utilizado para cifrar cadenas de cualquier longitud.

#### 4. Tweaks

El conjunto que representa al espacio de tweaks.

#### 5. Suma

El operador utilizado en la red Feistel para combinar la parte izquierda con la salida de la función de ronda.

#### 6. Método

El tipo de red Feistel a ocupar: desbalanceada o alternante.

Explicar la diferencia entre una suma a nivel de caracter o una suma a nivel de bloque.

Recordar que las redes balanceadas son caso específico de cualquiera de los dos métodos.

# FFX

## PARÁMETROS

Los siguientes 9 parámetros hacen de FFX un esquema muy general, que puede ser utilizado para cifrar cadenas de *cualquier* longitud.

### 7. *Split*

El grado de desbalanceo de la red Feistel.

### 8. Rondas

El número de rondas de la red Feistel.

### 9. F

La función de ronda. Recibe la llave, el *tweak*, el número de ronda y un mensaje; regresa una cadena del alfabeto de la longitud apropiada.



# Cifrados que preservan el formato

└ FFX

└ Definición de parámetros

└ FFX

## FFX

### Parámetros

Los siguientes 9 parámetros hacen de FFX un esquema muy general, que puede ser utilizado para cifrar cadenas de cualquier longitud.

#### 7. *Split*

El grado de desbalanceo de la red Feistel.

#### 8. *Rondas*

El número de rondas de la red Feistel.

#### 9. *F*

La función de ronda. Recibe la llave, el *text*, el número de ronda y un mensaje; regresa una cadena del alfabeto de la longitud apropiada.

El número de rondas necesario depende de cada caso en particular.  
Por ejemplo, en FF2 van de 12 a 36 y en FF10, de 12 a 24.

# BPS

## INTRODUCCIÓN

*BPS* es un algoritmo de cifrado que preserva el formato.

Este es capaz de cifrar cadenas de longitudes casi arbitrarias que estén formadas por cualquier conjunto de caracteres.

Se conforma de 2 partes fundamentales:

- ▶ Un cifrado interno *BC*, que cifra bloques de longitud fija.
- ▶ Un modo de operación, que usando a *BC*, permite que *BPS* cifre cadenas de varias longitudes.

# BPS

## CIFRADOR INTERNO $BC$

Este cifrador se define como:

$$BC_{F,s,b,w}(X, K, T)$$

Donde:

- ▶  $F$  es un cifrador por bloques de  $f$  bits de salida.
- ▶  $s$  es la cardinalidad del conjunto de caracteres  $S$ .
- ▶  $b$  es la longitud del bloque. ( $b \leq 2 \cdot |\log_s(2^{f-32})|$ )
- ▶  $w$  es el número de rondas de la red Feistel interna. (par)
- ▶  $X$  es la cadena de longitud  $b$  a cifrar.
- ▶  $K$  es una llave acorde a  $F$ .
- ▶  $T$  es un *tweak* de 64 bits.

# BPS

## CIFRADOR INTERNO $BC$

El cifrador  $BC$  sigue el siguiente proceso para cifrar un bloque.

1. Dividir el *tweak*  $T$  en 2 *subtweaks*  $T_L$  y  $T_R$  de 32 bits.

$$T_R = T \quad \text{mód } 2^{32}$$

$$T_L = (T - T_R)/2^{32}$$

# BPS

## CIFRADOR INTERNO $BC$

2. Dividir la cadena  $X$  en 2 para obtener  $X_L$  y  $X_R$  con longitudes  $l$  y  $r$  respectivamente.

Si  $b$  es par:

$$l = r = b/2$$

Si  $b$  es impar:

$$l = (b + 1)/2$$

$$r = (b - 1)/2$$

3. Definir e inicializar  $L_0$  y  $R_0$ .

$$L_0 = \sum_{j=0}^{l-1} X_L[j] \cdot s^j$$

$$R_0 = \sum_{j=0}^{r-1} X_R[j] \cdot s^j$$

4. Partiendo de  $i = 1$  hasta  $i < w$ :

Si  $i$  es par:

$$L_{i+1} = L_i \boxplus F_K((T_R \oplus i) \cdot 2^{f-32} + R_i) \pmod{s^l}$$

$$R_{i+1} = R_i$$

Si  $i$  es impar:

$$R_{i+1} = R_i \boxplus F_K((T_L \oplus i) \cdot 2^{f-32} + L_i) \pmod{s^r}$$

$$L_{i+1} = L_i$$

# BPS

## CIFRADOR INTERNO $BC$

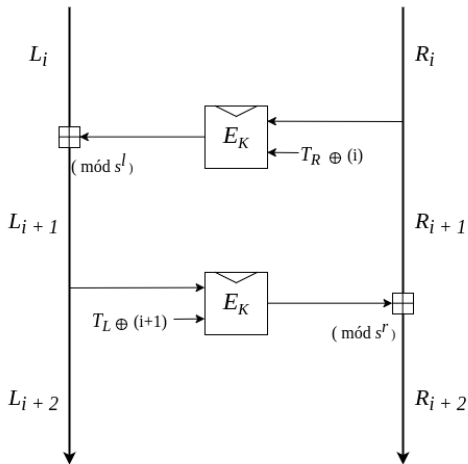


Diagrama de rondas del cifrado.





# BPS

DESCIFRADOR  $BC^{-1}$

Ahora, el proceso para descifrar la cadena  $Y$  es:

1. Dividir  $Y$  para obtener las subcadenas  $Y_L$  y  $Y_R$  con una longitud  $l$  y  $r$  respectivamente, de igual forma que se hizo con la cadena  $X$  en el proceso de cifrado.
2. Definir e inicializar  $L_w$  y  $R_w$  en:

$$L_w = \sum_{j=0}^{l-1} Y_L[j] \cdot s^j$$

$$R_w = \sum_{j=0}^{r-1} Y_R[j] \cdot s^j$$

# BPS

DESCIFRADOR  $BC^{-1}$

3. Comenzando con  $i = w - 1$ , para cada ronda  $i \geq 0$ .

Si  $i$  es par:

$$L_i = L_{i+1} \boxminus E_K((T_R \oplus i) \cdot 2^{f-32} + R_{i+1}) \pmod{s^l}$$

$$R_i = R_{i+1}$$

Si  $i$  es impar:

$$R_i = R_{i+1} \boxminus E_K((T_L \oplus i) \cdot 2^{f-32} + L_{i+1}) \pmod{s^r}$$

$$L_i = L_{i+1}$$

# BPS

DESCIFRADOR  $BC^{-1}$

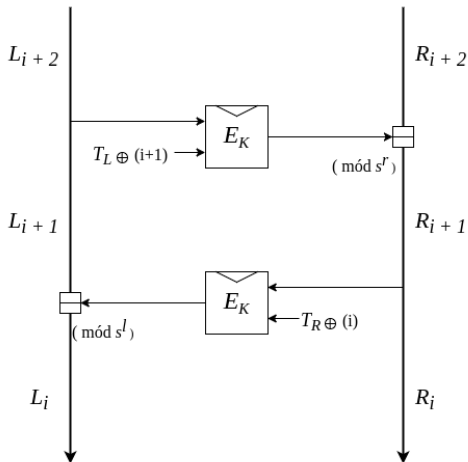


Diagrama de rondas del descifrado.

# BPS

DESCIFRADOR  $BC^{-1}$

4. Descomponer  $L_0$  y  $R_0$  (con el mismo proceso del cifrado) para obtener a  $X_L$  y  $X_R$ , las cuales concatenadas ( $X_L \parallel X_R$ ) dan la cadena de salida  $X$ .

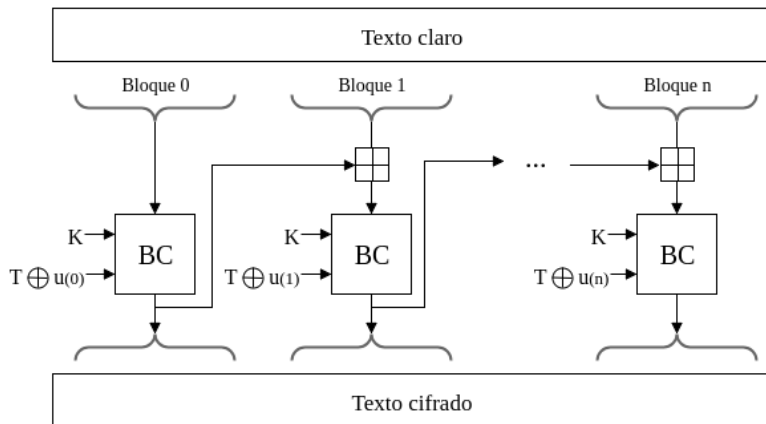
# BPS

## MODO DE OPERACIÓN

El modo de operación usado por *BPS* es equivalente al modo de operación CBC, ya que el bloque  $BC_n$  utiliza el texto cifrado de la salida del bloque  $BC_{n-1}$ , con la distinción de que en lugar de aplicar operaciones *xor* usa sumas modulares caracter por caracter, y de que no utiliza un vector de inicialización.

# BPS

## MODO DE OPERACIÓN



Modo de operación de *BPS*.

# BPS

## MODO DE OPERACIÓN

Como se observó en la figura anterior, se utiliza un contador  $u$  de 16 bits para aplicar una  $xor$  al *tweak*  $T$  en la entrada de cada  $BC$ .

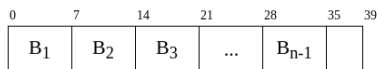
El  $xor$  se aplica a los 16 bits más significativos de ambas mitades de *tweak*, debido a cada mitad de *tweak* funciona de manera independiente en  $BC$ , y a que no se desea un traslape entre el contador externo e interno.



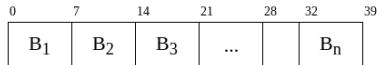
# BPS

## MODULO DE OPERACIÓN

Con este modo de operación, cuando el texto en claro a cifrar no tenga una longitud total que sea múltiplo de la longitud de bloque, el último bloque recorre su cursor de inicio hasta que su longitud concuerde.



b = 7



b = 7

Corrimiento de cursor de selección del ultimo bloque.

# BPS

## CONCLUSIONES

- ▶ *BPS* está basado en las redes Feistel y primitivas criptográficas estandarizadas, lo cual puede verse como una ventaja, debido al amplio estudio que tienen, y a que hacen más comprensible y fácil su implementación.
- ▶ *BPS* es un cifrado que preserva el formato capaz de cifrar cadenas formadas por cualquier conjunto y de un longitud de 2 hasta  $\max(b) \cdot 2^b$ .

# BPS

## CONCLUSIONES

- ▶ Se puede considerar que *BPS* es eficiente, debido a que la llave  $K$  usada en cada bloque  $BC$  es constante, y a que usa un número reducido de operaciones internas.
- ▶ El uso de *tweaks* protege a *BPS* de ataques de diccionario, los cuales son fáciles de cometer cuando el dominio de la cadena a cifrar es muy pequeño.

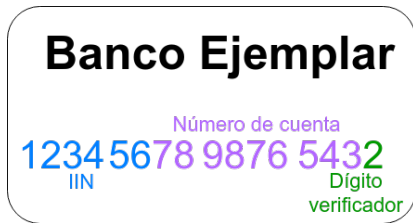
# BPS

## RECOMENDACIONES

- ▶ Se recomienda que el número de rondas  $w$  de la red Feistel sea 8, dado que es un número de rondas eficiente, y se ha estudiado la seguridad de *BPS* con este  $w$ .
- ▶ Es recomendable que como *tweak* se use la salida truncada de una función hash, en donde la entrada de la función puede ser cualquier información relacionada a los datos que se deseen proteger; por ejemplo, fechas, lugares, o parte de los datos que no se deseen cifrar.

## SOBRE EL PAN

Un número de tarjeta (PAN, por sus siglas en inglés), se compone por tres partes:



Los números están regidos por el ISO/IEC-7812. La longitud del número de tarjeta puede ir desde 12 hasta 19 dígitos.

# ANATOMÍA DE UN NÚMERO DE TARJETA

## SOBRE EL MII

El primer dígito de la tarjeta se refiere al *Major Industry Identifier* (MII). La relación entre dígitos e industrias es la siguiente:

- ▶ 1, 2: Aerolíneas
- ▶ 3: Viajes y entretenimiento (American Express, JBC)
- ▶ 4, 5: Bancos e industria financiera (Visa, Electron, Mastercard)
- ▶ 6: Comercio (Discover, Laser, China UnionPay)
- ▶ 7: Industria petrolera
- ▶ 8: Telecomunicaciones
- ▶ 9: Asignación nacional

# ANATOMÍA DE UN NÚMERO DE TARJETA

## SOBRE EL IIN

El *Issuer Identification Number* (IIN) comprende los primeros 6 dígitos, incluyendo el MII. El IIN puede proveer los siguientes datos:

- ▶ Banco emisor de la tarjeta
- ▶ Tipo de la tarjeta (crédito o débito)
- ▶ Marca de la tarjeta (Visa, MasterCard, Discover)
- ▶ Nivel de la tarjeta (Clásica, Gold, Black)

# ANATOMÍA DE UN NÚMERO DE TARJETA

## SOBRE EL IIN

La base de datos BINDB provee información cuando se ingresa un BIN<sup>1</sup> válido. Permite solo 10 consultas gratuitas por computadora.

Bin:	522130
Card Brand:	MASTERCARD
Issuing Bank:	TARJETAS BANAMEX SA DE CV SOFOM E.R.
Card Type:	CREDIT
Card Level:	STANDARD
Iso Country Name:	MEXICO
Iso Country A2:	MX
Iso Country A3:	MEX
Iso Country Number:	484

---

<sup>1</sup>*Bank Identifier Number*



# ANATOMÍA DE UN NÚMERO DE TARJETA

## SOBRE EL NÚMERO DE CUENTA

Los dígitos que siguen al IIN, excepto el último, son el número de cuenta. El número de cuenta puede variar, pero máximo comprende 12 dígitos, por lo que cada emisor tiene  $10^{12}$  posibles números de cuenta.



# BIBLIOGRAFÍA I



Phillip Rogaway. *A Synopsis of Format-Preserving Encryption*. 2010. URL: <http://web.cs.ucdavis.edu/~rogaway/papers/synopsis.pdf> (vid. págs. 19, 20, 22, 24).



Terence Spies Mihir Bellare Phillip Rogaway. “The FFX Mode of Operation for Format-Preserving Encryption”. Ver. 1.0. En: (2009) (vid. pág. 26).



Terence Spies Mihir Bellare Phillip Rogaway. “The FFX Mode of Operation for Format-Preserving Encryption”. Ver. 1.1. En: (2010) (vid. págs. 26, 32, 34).