Universidad Complutense Madrid

En esta frase hay cinco letras e
En esta frase hay seis letras e

Arturo Acuaviva Huertos
arturacu@ucm.es

Antonio Acuaviva Huertos
antoacua@ucm.es

Inmaculada Pérez Garbín
inmape01@ucm.es

Modelización de problemas de empresas

Madrid, a 9 de diciembre de 2020

Language is a process of free creation.

Noam Chomsky

Introducción

El presente documento recoge la propuesta de solución del equipo e-motion al problema presentado por la empresa GMV en el Concurso de Modelización de Problemas de Empresa, edición 2020-21, organizado por la Facultad de Matemáticas de la Universidad Complutense de Madrid.

En las diferentes secciones aquí expuestas se desarrolla una breve teoría sobre las firmas con posdatas basadas en la frecuencia de aparición de letras en el mensaje y la posdata. De esta manera, se detallan la complejidad computacional de resolver el problema de generar estas firmas y formas de implementar algoritmos de cómputo de estas; comentando también algunas observaciones relevantes o de interés relacionadas con este problema. Finalmente, también se han añadido comentarios relativos a la seguridad y los sistemas de firmas digitales ya existentes, analizando la viabilidad del sistema propuesto y revisando brevemente alternativas a este modelo.

Por otro lado, como complemento a este trabajo escrito está disponible el sitio web https://team-e-motion.github.io/CMPE-GMV-2020 en el cual se encuentran implementados algunos de los algoritmos aquí propuestos, permitiendo realizar y validar firmas utilizando los resultados teóricos expuestos.

Contenido

1. Introducción al problema

En este trabajo estudiaremos el problema de firmar documentos utilizando para ello una posdata que hace referencia al mensaje y a sí misma. En particular, distinguiremos dos tipos de firmas que detallaremos más adelante: la firma utilizando una sola letra del alfabeto y la firma oficial.

1.1. Firmando con una letra

Dado un mensaje que queremos firmar, estamos interesados en construir una posdata que haga referencia a la frecuencia de aparición de letras de un cierto tipo. Sin embargo, como peculiaridad de esta firma, escribiremos la frecuencia con palabras de manera que haga referencia no solamente al número de letras del mensaje sin la posdata, sino que también incluirá en el conteo de las apariciones de la letra el número de veces que esta aparece en la posdata.

Por ejemplo, supongamos el siguiente mensaje:

Buenas tardes

En este caso, se verifica que la letra *e* aparece un total de dos veces. Pero, si le añadimos la siguiente posdata que hace referencia al número de veces que aparece la letra *e* obtendríamos:

Buenas tardes; en este mensaje aparece dos veces la letra e

Sin embargo, la frase anterior constituye una propisición que es falsa, ya que la letra *e* aparece más de dos veces. Si contamos las apariciones de la letra *e* incluyendo la posdata:

Buenas tardes; en este mensaje aparece dos veces la letra e

Es fácil obtener entonces que la frecuencia de aparición es 13, que se escribe como *trece*. Notemos que si en lugar de *dos* dijésemos *trece* obtendremos una frase que también es falsa:

Buenas tardes; en este mensaje aparece trece veces la letra e

El problema entonces es que la anterior proposición es también falsa, ya que no hay *trece* letras *e* si no *quince*. Esta variación ha sido generada porque el propio número 13 expresado con letras tiene dos letras *e* que han de sumarse al conteo. De hecho, si intentamos en vez de *trece* utilizar el número *quince* nos encontraremos de nuevo con una proposición igualmente falsa:

Buenas tardes; en este mensaje aparece quince veces la letra e

Sin embargo, el anterior mensaje contiene solamente 14 letras e. Si, finalmente, intentamos escribir *catorce* en vez de *quince*, nos encontramos que entre ambas palabras no varía el número de letras *e*. De esta manera, el siguiente mensaje sí estaría correctamente firmado al ser cierto lo que dice:

Buenas tardes; en este mensaje aparece catorce veces la letra e

Como se ha ido exponiendo, en general puede ser complicado el obtener una firma de manera directa a partir de un mensaje por la interdependencia existente entre la transcripción de la frecuencia de aparición y la posibilidad de que esta transcripción contenga a la misma letra. En este trabajo, estudiaremos en los primeros apartados la existencia de estas firmas, su unicidad, propiedades generales que verifican y algoritmos para su cómputo, así como estudiaremos formalmente la complejidad de resolución del problema de firmar con una letra.

1.2. Firmando oficialmente

La idea de firma oficial surge a partir de la construcción de la firma de una letra. Si en lugar de decidir firmar con una sola letra decidiésemos construir una posdata con todas las letras:

en este mensaje aparece • *veces la letra a,* • *veces la letra b,* • *veces la letra c, . . .,* • *veces la letra z*

Estaríamos generando una nueva firma que contiene todas las frecuencias de todas las letras utilizadas para construir las palabras del mensaje, dicha posdata la denominaremos como firma oficial. El problema de firmar con varias letras, de forma oficial, es una extensión del problema de firmar con una sola letra. La complejidad de este problema respecto al de firmar con una sola letra aumenta debido a las interdependencias que pueden formarse en la frecuencia de aparición de ciertas letras en función de la aparición de otras.

En los apartados posteriores al análisis de la firma de una letra cubriremos el estudio de la firma oficial. Para ello, estudiaremos propiedades de este tipo de firmas como la unicidad, tras esto discutiremos algunos aspectos formales relativos a la complejidad de resolver el problema y finalmente expondremos algoritmos de cómputo de firma oficial.

1.3. El problema de crear firmas

Tras analizar los problemas de firmar con una letra y firmar oficialmente, pasaremos a estudiar el problema de la creación de firmas para mensajes. Expondremos los problemas que existen en la construcción de firmas dependientes de la propia información que transmiten, y la diferencia de este tipo de construcciones con otras utilizadas en algoritmos de encriptación como RSA o las diferencias de esta construcción interdependientes con otras construcciones diagonales utilizadas para probar la no regularidad de los lenguajes.

Finalmente, comentaremos brevemente algunos modelos de firmas de mensajes de uso en la actualidad y sus diferencias con el modelo de firmas basado en la frecuencia de aparición de letras.

2. Analizando el problema de firmar con una letra

En este apartado estudiaremos algunos resultados teóricos relacionados con el modelo de firmas para una letra arbitraria del abecedario, de la manera en la que ya expusimos brevemente en la Sección 1.1. Comenzaremos por formalizar los conceptos utilizados para la construcción de mensajes y firmas, y posteriormente estudiaremos propiedades de este modelo. Al acabar este apartado incluimos un estudio sobre la complejidad del problema y algoritmos de resolución del mismo.

2.1. Formalizando los conceptos

Sin pérdida de generalidad, consideraremos que todos las letras utilizadas para la formalización de resultados en este documento estarán en minúsculas. Estas letras en minúsculas sirven para componer lo que denominaremos palabras.

Definición 2.1 (Palabra)**.** Definiremos una palabra como la concatenación de caracteres o letras del alfabeto latino[1] o del carácter *ñ*.

Por ejemplo, *bonjour*, *house* o *españa* son palabras. Utilizando el concepto de palabras, podemos definir lo que entendemos por mensaje:

Definición 2.2 (Mensaje)**.** Definiremos un mensaje m como un conjunto de palabras unidos por signos de puntuación y/o espacios en blanco.

Un ejemplo de mensaje sería:

"hola, esto es un ejemplo de mensaje."

De ahora en adelante, denotaremos al conjunto de mensajes posibles como M.

También, nos interesa conocer la frecuencia de aparición de una determinada letra λ en un cierto mensaje m. Para ello, definiremos la siguiente función:

Definición 2.3. Denominaremos F_λ a la función $F_\lambda : M \to \mathbb{N}$ que asigna a cada mensaje m en español la frecuencia de aparición de la letra λ en el mensaje.

Por ejemplo, $F_a(\text{“}saludos\text{”}) = 1$, $F_e(\text{“}mensaje\text{”}) = 2$ y $F_f(\text{“}mensaje\text{”}) = 0$.

Introduciremos también la noción natural de longitud de un mensaje, de modo que:

Definición 2.4. Dado un mensaje m, se define su longitud L como el número natural dado por $L = \sum_{\lambda, \lambda \text{ letra}} F_\lambda(m)$.

Nótese como esta definición se corresponde con la definición natural de longitud de un mensaje (ignorando los signos de puntuación y el espaciado).

Dado un mensaje, nos interesa saber el valor de su evaluación a través de la función F_λ para poder calcular su posdata. Por otro lado, estamos interesados en conocer además para un número la frecuencia de cierta letra λ en el conjunto de palabras que lo expresan. Para obtener esto primero definiremos una función que dado un número devuelva el conjunto de palabras que lo expresan:

[1]Nótese que la particularización de utilizar el alfabeto latino junto a la letra $\tilde{n}$ no resta valor a los resultados que aquí se presentan, el análisis de la firma se basará en la frecuencia de caracteres con lo cual es fácil extrapolar los resultados a otros alfabetos.

Definición 2.5. Denominaremos G a la función $G : \mathbb{N} \to M$ que asigna a cada número $n \in \mathbb{N}$ el mensaje $G(n)$ que expresa n en palabras en español[2].

Por ejemplo, $G(24) = veinticuatro$ y $G(100) = cien$.

Combinando las funciones anteriores, podemos obtener una función que asigne a cada número la frecuencia de aparición de una letra λ:

Definición 2.6. Denominaremos Φ_λ a la función $\Phi_\lambda : \mathbb{N} \to \mathbb{N}$ que asigna a cada número $n \in \mathbb{N}$ la frecuencia de aparición de la letra λ en las palabras que describen al número, esto es, $\Phi_\lambda(n) = (F_\lambda \circ G)(n) = F_\lambda(G(n))$

Por ejemplo, tenemos que $\Phi_e(10) = 1$, $\Phi_n(10) = 0$ o $\Phi_e(13) = 2$ (observemos que *diez* contiene una e y ninguna n, y *trece* contiene dos letras e).

Si estudiamos ahora la curiosa forma de firmar los mensajes consistente en añadir al final del mensaje una posdata en la que se indica la frecuencia de aparición de ciertas letras, incluyendo en este recuento a la propia posdata, descubriremos que podemos plantear fácilmente este problema en función de los conceptos que hemos definido previamente. De hecho, el problema de firmar se puede formalizar como sigue:

Problema. *El problema de firmar un mensaje añadiendo una posdata donde se indica la frecuencia de aparición de cierta letra λ puede expresarse como*

$$F_\lambda(m) + \Phi_\lambda(n) = n$$

donde $n \in \mathbb{N}$ es la solución a la ecuación que verifica que el mensaje y la posdata anterior forman una frase cierta.

Nota. *Para que el anterior problema tenga sentido, fijada una letra λ añadiremos en el mensaje m' inicial también la parte de la posdata que se refiere a la frecuencia de aparición de λ, para obtener el mensaje a firmar m. Esto es, si fijamos el siguiente conjunto de palabras como posdata:*

"*en este mensaje aparece • veces la letra λ*"

entonces el mensaje m contiene a m' mensaje inicial y al conjunto de palabras "en este mensaje aparece • veces la letra λ".

En los próximos apartados estudiaremos si hay unicidad en la solución de la ecuación anterior, si existe siempre solución a la ecuación planteada y como obtener el valor de n fijados un mensaje m y una letra λ.

Finalmente, introduciremos el concepto de firma al que haremos referencia a la largo del documento:

Definición 2.7 (Firma)**.** Denominaremos firma de un mensaje m para la letra λ al valor $n \in \mathbb{N}$ tal que verifica la ecuación $F_\lambda(m) + \Phi_\lambda(n) = n$.

Esto es, en realidad firmar un mensaje es el proceso de encontrar la firma o solución a la ecuación $F_\lambda(m) + \Phi_\lambda(n) = n$ que hemos planteado.

[2]Podemos obtener una función equivalente para cualquier otro idioma.

2.2. El comportamiento de F_λ, G y Φ_λ

Previo estudio de la ecuación presentada en el apartado anterior, estamos interesados en estudiar el comportamiento de las funciones F_λ, G y Φ_λ que intervienen en la misma.

2.2.1. La función F_λ

Comenzaremos por estudiar F_λ. Fijado un mensaje m y una letra λ, es fácil computar el valor de F_λ simplemente contando la frecuencia de aparición de λ en el mensaje. Notemos que si estamos interesados en un mensaje con posdata, al cómputo anterior debemos añadir el conteo de la frecuencia de λ en el mensaje *"en este mensaje aparece • veces la letra λ"*[3]. En particular, esta última posdata que concatenamos al final del mensaje es constante fijado λ. La siguiente tabla recoge en función de la letra λ la frecuencia de aparición de la letra en esta postada que se añade al final:

λ	Frecuencia	λ	Frecuencia
a	6	n	2
b	1	ñ	1
c	3	o	1
d	1	p	2
e	11	q	1
f	1	r	3
g	1	s	4
h	1	t	1
i	1	u	1
j	2	v	2
k	1	w	1
l	3	x	1
m	2	y	1
-	-	z	1

Los valores de la tabla anterior junto con el de computar el número de letras λ en el mensaje original sin postadata son constantes, y su suma resulta ser el valor de F_λ buscado. Si denominamos m' al mensaje inicial sin añadir la posdata y denotamos k_λ el valor de la tabla correspondiente a la frecuencia para λ, entonces el mensaje a firmar m se puede evaluar como $F_\lambda(m) = F_\lambda(m') + k_\lambda$.

2.2.2. La función G

Para estudiar el comportamiento de G analizaremos como se construyen los números en español[4]. De acuerdo a la Real Academia Española (RAE), la construcción de los números puede resumirse como sigue [Esp70]:

[3]Nótese que, en el caso de que la frecuencia de aparición de la letra fuese una única vez, se debería cambiar "veces"por "vez", pero se puede seguir un razonamiento análogo. Notemos también que no añadimos en este caso β pues este valor vendrá dado por la solución a la ecuación y se calcula en el segundo sumando del primer miembro de la igualdad.

[4]Nótese que un análisis totalmente análogo puede realizarse para otros idiomas siempre y cuando se mantenga la regularidad en la construcción de los cardinales.

Hay cardinales simples —de cero a quince, todas las decenas (diez, veinte, treinta, etc.), cien(to), quinientos y mil— y cardinales compuestos, los formados por la fusión o suma de varios cardinales simples. De los compuestos, se escriben hoy en una sola palabra los correspondientes a los números 16 a 19 y 21 a 29, así como todas las centenas: dieciséis, dieciocho, veintiuno, veintidós, doscientos, cuatrocientos, etc. A partir de treinta, los cardinales compuestos que corresponden a cada serie se escriben en varias palabras y se forman, bien por coordinación, bien por yuxtaposición de cardinales simples; así, los correspondientes a la adición de unidades a las decenas se escriben interponiendo entre los cardinales simples la conjunción y: treinta y uno, cuarenta y cinco, noventa y ocho, etc.; el resto se forma por mera yuxtaposición: ciento dos, mil cuatrocientos treinta, trescientos mil veintiuno, etc.

Debemos entonces fijarnos primero en las excepciones y luego en las reglas generales que permiten construir números de forma regular. Las excepciones correspondientes a los primeros 29 números se recogen en la siguiente tabla:

1	uno	11	once	21	veintiuno
2	dos	12	doce	22	veintidos
3	tres	13	trece	23	veintitres
4	cuatro	14	catorce	24	veinticuatro
5	cinco	15	quince	25	veinticinco
6	seis	16	dieciséis	26	veintiseis
7	siete	17	diecisiete	27	veintisiete
8	ocho	18	dieciocho	28	veintiocho
9	nueve	19	diecinueve	29	veintinueve
10	diez	20	veinte	-	-

Los siguientes números a partir del treinta se obtienen por composición, por ejemplo *treinta y tres*, *cincuenta y cinco*. La siguiente tabla recoge las palabras que permiten construir estos números:

30	treinta	40	cuarenta	50	cincuenta	60	sesenta	70	setenta	80	ochenta	90	noventa

Con estas palabras se forman los números de dos cifras, como el *cuarenta y tres* o el *ochenta y cuatro*. A partir de las centenas es necesario otro conjunto de palabras que nos permitan construir el resto de números, la siguiente tabla recoge dichas palabras hasta el número mil:

100	cien	200	doscientos	300	trescientos	400	cuatrocientos	500	quinientos
1..	ciento	600	seiscientos	800	ochocientos	900	novecientos	1000	mil

Los siguientes números ya se construyen por composición, las únicas nuevas palabras que pueden aparecer entonces para expresar un número son las correspondientes a las nuevas potencias de diez que no pueden construirse directamente por coordinación. En particular, si queremos cubrir números hasta el trillón solo necesitamos introducir las palabras *millón*, *millones*, *billón*, *billones*, *trillón* y *trillones*[5].

Podemos estudiar entonces el comportamiento de G en función de las reglas de coordinación en español, de esta manera obtenemos el siguiente resultado:

[5]Por simplicidad, en los futuros apartados nos limitaremos a estudiar la construcción de números hasta el millón, notemos que las construcciones siguientes son regulares y se pueden obtener de forma análoga resultados para valores como el *cuatrillón*, *quintillón*...

Proposición 2.1 (Construcción de los cardinales hasta el millón)**.** *Dado un número $n \in \mathbb{N}$ que podemos expresar en función de sus cifras c como $n = n_{c-1}n_{c-2}\dots n_0$ siendo n_0 la cifra menos significativa, con $c \leq 7$ entonces G puede computarse como:*

- *Si $c = 1$, entonces $G(n) = G(n_0)$ y la imagen de la función será: "uno", "dos", "tres", "cuatro", "cinco", "seis", "siete", "ocho" o "nueve".*
- *Si $c = 2$, entonces se tienen dos casos:*
  - *$n_1 \geq 3$, entonces el número se construye como "treinta", "cuarenta", "cincuenta", "sesenta", "setenta", "ochenta" o "noventa", seguido de la letra "y" junto a la imagen de $G(n_0)$.*
  - *$n_1 < 3$, entonces el número se corresponde con uno de los cardinales compuestos que puede dividirse en prefijo seguido del número. De esta manera, tenemos los dos siguientes casos:*
    - *$n_1 = 2$, entonces el número se construye como el prefijo "veinti"sobre la imagen de $G(n_0)$.*
    - *$n_1 = 1$, entonces si $n_2 \geq 5$ el número se construye como el prefijo "dieci"sobre la imagen de $G(n_0)$, en caso contrario se tiene un valor fijo: "diez", "once", "doce", "trece", "catorce" o "quince".*
- *Si $c = 3$ entonces los números se expresan como "cien" o "cientos"(si existen más cifras no nulas detrás), "doscientos", "trescientos", "cuatrocientos", "quinientos", "seiscientos", "setecientos", "ochocientos" o "novecientos"seguidos de la imagen de $G(n_1n_0)$.*
- *Si $6 \geq c \geq 4$, se tienen dos casos:*
  - *Si $n_3 = 1$ y $c = 4$, entonces el número se expresa como "mil"seguido de la imagen de $G(n_2n_1n_0)$*
  - *En caso contrario, entonces el número se expresa como la imagen de $G(n_{c-1}\dots n_3)$ seguido de la palabra "mil" y posteriormente seguido de la imagen de $G(n_2n_1n_0)$*
- *Si $c = 7$ se tienen los siguientes dos casos:*
  - *Si $n_6 = 1$, entonces el número se construye como "un millón"seguido de la imagen de $G(n_5n_4n_3n_2n_1n_0)$.*
  - *En caso contrario, entonces el número se construye como la imagen de $G(n_6)$ seguida de la palabra "millones"seguida de la imagen de $G(n_5n_4n_3n_2n_1n_0)$.*

Este resultado se sigue directamente de las reglas de construcción de cardinales en español. En el Anexo A se incluye una tabla con las frecuencias de las letras en la construcción de cardinales.

2.2.3. La función Φ_λ

Notemos ahora, conocidas las excepciones y palabras que intervienen en la construcción de los números, que es fácil establecer reglas para la función Φ_λ atendiendo a las reglas de coordinación que permiten generar los números con G. Podemos estudiar, entonces, algunas propiedades relevantes de esta función. En particular, podemos acotar el número de letras λ que computará la función Φ_λ considerando los dígitos de n.

Presentamos primero un lema que será de particular interés para hallar cotas sobre la función Φ_λ.

Lema 2.1. *La imagen de cualquier número $n \in \mathbb{N}$, que podemos expresar como $n = n_{c-1} \dots n_0$, a través de ϕ_λ coincide con la evaluación de ϕ_λ sobre todas las cifras salvo un valor $K_\lambda(c)$ que depende del número de cifras c de n. Esto es,*

$$\phi_\lambda(n) \leq K_\lambda(c) + \sum_{i=0}^{c-1} \phi_\lambda(n_i)$$

Demostración. Observemos que por la Proposición 2.1 la construcción de los cardinales sigue ciertas reglas que permiten descomponer al número en la suma de las expresiones de sus cifras. La diferencia entre $\phi_\lambda(n)$ y $\sum_{i=0}^{c-1} \phi_\lambda(n_i)$ depende claramente entonces del número de cifras c. Para $c = 1$ el resultado es trivial, estudiemos ahora los casos para mayor número de cifras.

- Si $c = 2$ los números se construyen utilizando las cifras con sufijos o con variaciones. Por ejemplo, tenemos "treinta y dos"modificando la palabra "tres"por "treinta". Además se ha añadido la conjunción "y"para concatenar el valor de la segunda cifra. Se puede comprobar que para todos los casos la variación en el número de letras de algún tipo que surge al expresar directamente el número como concatenación de las letras de las cifras en vez de expresar el número con su sufijo y con la conjunción no es mayor que 1. Esto es, se verifica:

$$\phi_\lambda(n) \leq (1 + \phi_\lambda(n_1)) + \phi_\lambda(n_0)$$

- Si $c = 3$ podemos omitir el cálculo de las dos últimas cifras que ya conocemos su constante de acotación del caso anterior; en general, la variación para en la frecuencia de la centena no será mayor de 1, al igual que el caso anterior.

$$\phi_\lambda(n) \leq (1 + \phi_\lambda(n_2)) + (1 + \phi_\lambda(n_1)) + \phi_\lambda(n_0)$$

- Si $6 \geq c \geq 4$, nos centraremos en las milésimas. Por construcción se tiene la palabra "mil"que a lo sumo añade una letra, con lo cual tenemos que el máximo en este caso es el máximo de los posibles valores anteriores más uno (nótese que la letra "i"puede aparecer una vez extra en las centésimas, "seiscientos", con lo cual en "seiscientos mil" aparecerá dos veces extra). Esto es, el máximo para el caso de la milésima es 2.

$$\phi_\lambda(n) \leq (2 + \phi_\lambda(n_5)) + (2 + \phi_\lambda(n_4)) + (2 + \phi_\lambda(n_3))$$
$$+(1 + \phi_\lambda(n_2)) + (1 + \phi_\lambda(n_1)) + (\phi_\lambda(n_0)$$

- Finalmente, si $c = 7$ obtenemos el número por combinación de los anteriores con la palabra "millones"(o apareciendo simplemente "un millón"para el caso en que $n_6 = 1$). Nótese que en este caso aunque aparece dos veces la letra "l", en realidad esta letra no puede aparecer tres veces en los millares con lo cual sólo puede aumentar en uno (por contener la "i"). Por ello, la cota en este caso es 3, como solo consideraremos las unidades de millones tomaremos la cota en 2:

$$\phi_\lambda(n) \leq (2 + \phi_\lambda(n_6)) + (2 + \phi_\lambda(n_5)) + (2 + \phi_\lambda(n_4)) + (1 + \phi_\lambda(n_3))$$
$$+(1 + \phi_\lambda(n_2)) + (1 + \phi_\lambda(n_1)) + \phi_\lambda(n_0)$$

- Para valores de $c > 7$ la construcción es análoga ajustando la constante $K_{\lambda,i}$ en función de las variaciones en el prefijo o sufijo utilizado para expresar el número en el orden de magnitud que fuese.

Reagrupando las sumas anteriores y renombrando los términos constantes como $K_{\lambda,i}$, donde i indica el número de cifra al que acompañan, obtenemos

$$\phi_\lambda(n) \leq \sum_{i=0}^{c-1} K_{\lambda,i} + \sum_{i=0}^{c-1} \phi_\lambda(n_i) = K_\lambda(c) + \sum_{i=0}^{c-1} \phi_\lambda(n_i)$$

En particular el valor de $K(c)$ será en función de las cifras $c = \lceil \log_{10} n \rceil$, □

Con ayuda de este lema, podemos probar el siguiente resultado.

Corolario 2.1. *Existe una función $K(c)$ creciente, de modo que $\forall \lambda$ se tiene que*

$$\phi_\lambda(n) \leq K(c) + \sum_{i=0}^{c-1} \phi_\lambda(n_i)$$

donde $c = \lceil \log_{10} n \rceil$ es el número de cifras de n.

Demostración. Basta tomar $K(c) = \text{máx}\{K_\lambda(c) : \lambda \text{ letra}\}$. Nótese que con el razonamiento del teorema anterior uno puede probar que $K(c)$ alcanza los siguientes valores $K(c)$: $K(1) = 0, K(2) = 1, K(3) = 2, K(4) = 4, K(5) = 6, K(6) = 8, K(7) = 10$. En particular, tendremos $K(c) \leq C\lceil \log_{10} n \rceil$, para cierta constante C. □

Por otro lado, tenemos el siguiente resultado:

Lema 2.2. *La frecuencia de aparición de cualquier letra en los primeros 9 números es inferior a 2.*

Demostración. Basta comprobar de forma exhaustiva la siguiente lista: “uno”, “dos”, “tres”, “cuatro”, “cinco”, “seis”, “siete”, “ocho”, “nueve”. □

Combinando los lemas y el colario anterior podemos obtener el siguiente resultado que acota la función:

Corolario 2.2. *Existe una constante $C > 0$ tal que para todo $n \in \mathbb{N}$, se tiene*

$$\phi_\lambda(n) \leq C\lceil \log_{10} n \rceil.$$

Demostración. Utilizando el Lema 2.1, el Lema 2.2 y el Corolario 2.1 se sigue que

$$\phi_\lambda(n) \leq K_\lambda(c) + \sum_{i=0}^{c-1} \phi_\lambda(n_i) \leq C_1\lceil \log_{10} n \rceil + \sum_{i=0}^{c-1} 2 = (2 + C_1)\lceil \log_{10} n \rceil = C\lceil \log_{10} n \rceil.$$

□

Este resultado de acotación nos será de vital importancia en secciones posteriores, cuando estemos estudiando el comportamiento en complejidad de la firma de documentos. Por otro lado, también tenemos el siguiente teorema de acotación que permite establecer cotas más exactas en función del número de dígitos:

Teorema 2.1 (Acotación de Φ_λ)**.** *El valor de $\Phi_\lambda(n)$ para $n \in \mathbb{N}$ verifica la desigualdad*

$$\Phi_\lambda(n) \leq C \cdot \lceil \log_{10}(n) \rceil + K(c)$$

donde $C = 2$ y $K(c)$ es la expuesta en el Lema 2.1.

Demostración. Tomemos $n \in \mathbb{N}$ tal que su expresión en dígitos viene dada por $n = n_{c-1}n_{c-2}\dots c_0$ siendo c el número de dígitos. Por otro lado, y en virtud del Lema 2.1, si estudiamos la expresión general de Φ_λ obtendremos:

$$\Phi_\lambda(n) = (F_\lambda \circ G)(n) \leq \sum_{i=0}^{c-1}(F_\lambda \circ G)(n_i) + K(c)$$

Pero, sin embargo, utilizando el Lema 2.2 podemos acotar la expresión anterior como:

$$\Phi_\lambda(n) = (F_\lambda \circ G)(n) \leq \sum_{i=0}^{c-1} 2 + K(c) \leq 2 \cdot c + K(c)$$

Sin embargo, el número de dígitos c verifica que $c \leq \lceil \log_{10}(n) \rceil$, y tomando $C \geq 2$ obtenemos:

$$\Phi_\lambda(n) \leq c \cdot 4 \leq C \cdot \lceil \log_{10}(n) \rceil + K$$

□

Finalmente, comentarmeos que existen una serie de letras cuyo tratamiento será especialmente sencillo.

Teorema 2.2. *Sea $\lambda \in \{b, f, g, j, k, p, w, x\}$, entonces se tiene que $\phi_\lambda(n) = 0$ para $0 \leq n < 10^{12}$.*

Demostración. Basta revisar la tabla del Anexo A y comprobar que las letras no aparecen en ningún prefijo o palabra usada para la construcción de los cardinales. □

2.3. Propiedades de la ecuación $F_\lambda(m) + \Phi_\lambda(n) = n$

Fijado un mensaje $m \in M$ y una letra λ, el número $n \in \mathbb{N}$ que verifique $F_\lambda(m) + \Phi_\lambda(n) = n$ es el número buscado para poder firmar un mensaje. En las siguientes secciones estudiaremos la existencia, unicidad y cómputo de la firma.

2.3.1. Unicidad de la firma

Si nos atenemos al concepto de firma expuesto en 2.7, nos encontramos que el mismo mensaje para la misma letra tiene varias firmas válidas posibles. Esto es, no podemos garantizar la unicidad de la firma fijado el mensaje y la letra.

Proposición 2.2 (No unicidad de la firma)**.** *La definición de firma dada en 2.7 no es única. Esto es, dado un mensaje pueden existir más de una firma para el mismo.*

Demostración. Consideremos el mensaje $m =$ "*hola*". Observamos que si queremos firmar el mensaje anterior con la letra e, nos encontramos que los siguientes mensajes con firmas cumplen las definiciones expuestas hasta ahora:

"*hola, en este mensaje aparece doce veces la letra e*"
"*hola, en este mensaje aparece trece veces la letra e*"

De hecho, ambos mensajes verifican la ecuación $F_e(m) + \Phi_e(n) = n$ pues:

$$11 + \Phi_e(12) = 12$$

$$11 + \Phi_e(13) = 13$$

Esto es, para un mismo mensaje m hemos encontrado dos firmas distintas $n = 12$ y $n' = 13$. □

La unicidad no es un requisito para poder firmar mensajes, sin embargo, para el desarrollo de algoritmos deterministas que puedan computar firmas simplifica la tarea el trabajar con soluciones únicas. Motivados por este aspecto, introducimos a continuación el concepto de firma mínima válida que garantiza la unicidad:

Definición 2.8 (Firma mínima válida)**.** Definiremos la firma mínima válida n_* de un mensaje m como aquella firma que verifica que para cualquier otra firma n del mensaje m entonces $n_* \leq n$.

La unicidad de la definición anterior se obtiene entonces por propia construcción, la firma mínima válida es el mínimo valor de entre todas las firmas posibles. Nótese que la firma mínima válida existe si y solamente si existe al menos una firma para el mensaje.

El concepto de firma mínima válida lo utilizaremos posteriormente en el desarrollo de algoritmos de cómputo deterministas para hallar el valor de la firma para un cierto mensaje, fijada una letra.

2.3.2. Existencia de la firma

Por otro lado, desafortunadamente, no podemos garantizar la existencia de firma para cualquier mensaje dado, el siguiente resultado recoge esta afirmación:

Proposición 2.3 (Existencia de la firma no garantizada)**.** *La firma definida en 2.7 no siempre existe. Esto es, dado un mensaje puede no existir una firma para el mismo.*

Demostración. Si tomamos el mensaje original $m' =$ "*buenas*" podemos comprobar que no existe firma con la letra e. En el mensaje m' tenemos que la frecuencia de aparición de e es 1, además, en el mensaje con posdata sin contar la la expresión de la firma es 11. Esto significa que como mínimo el valor de n deberá expresar ser mayor o igual a $11 + 1 = 12$. Esto es, se cumple la siguiente acotación inferior:

$$F_e(m) + \Phi_e(n) = n \iff 12 + \Phi_e(n) = n \geq 12$$

Por otro lado, por el Teorema 2.1 sabemos que $\Phi_e(n) \leq 2 \cdot \lceil log_{10}(n) \rceil + K_e(c)$. Esto es, tenemos las siguientes acotaciones:

$$12 + 2 \cdot \lceil log_{10}(n) \rceil + K_e(c) \geq 12 + \Phi_e(n) = n \geq 12$$

Esto es,

$$12 + 2 \cdot \lceil log_{10}(n) \rceil + K_e(c) \geq n \geq 12$$

Pero, por ser $12+2\cdot\lceil log_{10}(n)\rceil+K_e(c)$ una función monótona creciente, si tomamos por ejemplo $n = 100$, entonces obtenemos $12 + 4 \cdot \lceil log_{10}(100) \rceil + (2 + 2) = 12 + 2 \cdot 2 + 4 << 100$, con lo cual si existe firma esta tiene que ser estrictamente inferior a 100. Es más, cualquier valor v entre 11 y 100 resultará en $\lceil log_{10}(v) \rceil = 2$ con lo cual sabemos que de existir n debería verificar

$$12 + 2 \cdot 2 + 4 \geq n$$

Esto significa que los valores de n candidatos a ser firmas son $\{12, 13, 14, 15, 16, 17, 18, 19, 20\}$. Sin embargo, ninguno de estos valores es válido ya que ninguno de los siguientes mensajes es correcto:

$m_{12} =$"*en este mensaje aparece doce veces la letra e*"
$m_{13} =$"*en este mensaje aparece trece veces la letra e*"
$m_{14} =$"*en este mensaje aparece catorce veces la letra e*"
$m_{15} =$"*en este mensaje aparece quince veces la letra e*"
$m_{16} =$"*en este mensaje aparece dieciséis veces la letra e*"
$m_{17} =$"*en este mensaje aparece diecisiete veces la letra e*"
$m_{18} =$"*en este mensaje aparece dieciocho veces la letra e*"
$m_{19} =$"*en este mensaje aparece diecinueve veces la letra e*"
$m_{20} =$"*en este mensaje aparece veinte veces la letra e*"

Con lo cual, como n tiene que verificar que $n \in \{12, 13, 14, 15, 16, 17, 18, 19, 20\}$ pero ninguno de estos valores es una firma válida. En particular, no existe firma para el mensaje $m' =$ "*buenas*". Es más, en general, podemos afirmar que dado un mensaje m puede no existir su firma. □

2.4. Algoritmos para la firma de una letra

Una vez expuestos los resultados teóricos anteriores, en esta sección presentamos un análisis general del problema en cuanto a complejidad y a posibles algoritmos que lo resuelvan. Notemos que el problema de firmar con una letra un mensaje en general puede no tener solución, por lo expuesto en la Proposición 2.3, y puede no existir siempre solución única, Proposición 2.2. Por este motivo, el problema que aquí analizamos se corresponde con el problema de encontrar el valor más pequeño $n \in \mathbb{N}$ que sea solución para la ecuación

$$F_\lambda(m) + \Phi_\lambda(n) = n$$

siempre que exista, y si no existe determinar que no se puede encontrar el valor. Esto es, buscaremos decidir si existe la Firma Mínima Válida expuesta en la Definición 2.8 y si existe, dar su valor.

2.4.1. Cómputo de la firma mínima válida de un mensaje

En este apartado expondremos un algoritmo de cómputo de la firma mínima válida, así como probaremos formalmente su corrección y estudiaremos su complejidad computacional. Este algoritmo se basa en los resultados de acotación expuestos en los apartados anteriores que garantizan que se puede encontrar solución si existe.

Algoritmo 1: Generación de la firma mínima válida basada de una letra

Datos: Un mensaje m con su posdata sin la frecuencia indicada, una letra λ usada para firmar y la función $K(c)$ definida en 2.1.
Resultado: La firma mínima válida n_* tal que $n_* \leq n$ para cualquier otra n que sea firma o -1 en caso de no existir

```
total_λ = cuentaFrequenciaAparición(m, λ) + 1;
dígitos = ⌈log10(total_λ)⌉ ;
n = total_λ;
cota = 2 × dígitos + K(dígitos) + total_λ;
mientras n ≤ cota y ¬ esSolucion(n) hacer
    n = n + 1 ;
fin
si n > cota entonces
    resultado = -1;
en otro caso
    resultado = n;
fin
```

Notemos que en el anterior algoritmo las funciones *cuentaFrequenciaAparición* y *comprobarSolucion* realizan las funciones que sus propios nombres indican. En el primer caso simplemente se limita a devolver la frecuencia de aparición de la letra λ en el mensaje m y en el segundo caso se comprueba si se cumple la ecuación que verifican las firmas.

A efectos de comprobar que el algoritmo es correcto añadimos el siguiente resultado:

Proposición 2.4. *El Algoritmo 1 calcula, si existe, el valor de la firma mínima válida.*

Demostración. Basta observar que cualquier firma válida verifica la ecuación:

$$F_\lambda(m) + \Phi_\lambda(n) = n$$

Y además, por el Teorema 2.1 de Acotación de $\Phi_\lambda(n)$ tenemos que:

$$\Phi_\lambda(n) \leq 2 \cdot c + K(c)$$

Donde c es el número de dígitos. En particular, esto quiere decir que se cumple la siguiente desigualdad:

$$F_\lambda(m) + 2 \cdot c + K(c) \geq n$$

Además, trivialmente se tiene que:

$$F_\lambda(m) \leq n$$

Esto es, se tiene la siguiente desigualdad:

$$F_\lambda(m) \leq n \leq F_\lambda(m) + 2 \cdot c + K(c)$$

Esta desigualdad es precisamente el rango de valores que explora el bucle while del algoritmo, empezando desde el valor más pequeño y garantizando entonces encontrar el mínimo valor de n válido que satisface la ecuación, esto es, la firma mínima válida. □

Por otro lado, podemos extraer también la complejidad computacional del algoritmo haciendo un análisis formal del mismo. Para este análisis no tendremos en cuenta el coste de calcular la frecuencia de aparición de la letra λ en el mensaje m, y nos limitaremos a expresar el coste de la parte de generación de la firma conocida la frecuencia de aparición inicial:

Proposición 2.5. *El Algoritmo 1 que calcula, si existe, el valor de la firma mínima válida pertenece a* $O(\log(L))$ *donde* L *es la longitud del mensaje inicial* m *dada en la Definición 2.4.*

Demostración. Notemos simplemente que la complejidad del algoritmo viene dada por el bucle que itera desde un valor n inicial hasta la cota dada por el Teorema 2.1 de Acotación de $\Phi_\lambda(n)$. En particular, en el peor caso en el que no exista firma tendremos que el bucle iterará un total de $2 \cdot c + K(c) + F_\lambda(m) - F_\lambda(m) = 2 \cdot c + K(c)$ veces, siendo c el número de dígitos que expresan la frecuencia de aparición de la letra λ en el mensaje. En particular, dado que $c \propto \log(L)$ se tiene que la complejidad del algoritmo está superiormente acotada por $O(\log(L))$. □

2.4.2. Cómputo de todas las firmas válidas de un mensaje

Utilizando los resultados obtenidos hasta ahora podemos definir un algoritmo que además calcule todas las firmas válidas, para esto observemos que por el Teorema 2.1 de Acotación de $\Phi_\lambda(n)$ se tiene el siguiente resultado:

Lema 2.3. *Si existe más de una firma válida para un mensaje* m*, todas las firmas válidas se encuentran a distancia a lo sumo* $c + K(c)$ *donde* c *es el número de dígitos de* $F_\lambda(m)$*.*

Es más, podemos modificar el algoritmo presentado en el apartado anterior para calcular todas las

firmas válidas de un mensaje dado:

Algoritmo 2: Generación de la firma mínima válida basada de una letra

Datos: Un mensaje m con su posdata sin la frecuencia indicada, una letra λ usada para firmar y la función $K(c)$ definida en 2.1.
Resultado: Todas las firmas válidas o el conjunto vacío en caso de no existir ninguna

```
total_λ = cuentaFrequenciaAparición(m, λ) + 1;
dígitos = ⌈log10(total_λ)⌉ ;
n = total_λ;
cota = 2 × dígitos + K(dígitos) + total_λ;
resultado = ∅;
mientras n ≤ cota hacer
    si esSolucion(n) entonces
        resultado = resultado ∪ { n };
    fin
    n = n + 1 ;
fin
```

De forma totalmente análoga a los resultados del apartado anterior se tienen los mismos resultados de corrección y complejidad.

Proposición 2.6. *El Algoritmo 2 obtiene, si existe, el conjunto de todas las firmas válidas.*

Este algoritmo es de hecho equivalente al anterior exceptuando que en este caso siempre se completan todas las iteraciones del bucle buscando alguna solución, sin embargo el razonamiento en cuanto a su corrección es totalmente análogo.

Proposición 2.7. *El Algoritmo 2 que obtiene, si existe, el conjunto de todas las firmas válidas pertenece a $O(\log(L))$ donde L es la longitud del mensaje dada en la Definición 2.4.*

De igual manera, el razonamiento en cuanto a la complejidad es totalmente análogo, simplemente observando que las iteraciones del bucle son las que acotan la complejidad del mismo.

3. Analizando el problema de la firma oficial

En apartados anteriores estudiamos el concepto de la firma generada por la frecuencia de aparición de una palabra en el mensaje y en la propia posdata que actuaba de firma. En este apartado nos centraremos en una generalización de este problema.

3.1. Formalizando los conceptos

En esta sección expandiremos el problema de la firma a un ambiente más general, cuando se utilice más de una letra para firmar. Es decir, nos preguntamos cuando, dado un mensaje y una familia de letras $\lambda_1, \ldots, \lambda_k$, es posible firmar el mensaje con estas letras. Siguiendo un razonamiento análogo al realizado en la Sección 2, podemos formalizar este problema como[6]:

Problema. *El problema de firmar un mensaje añadiendo una posdata donde se indica la frecuencia de aparición de cierta familia de letras $\lambda_1, \ldots, \lambda_k$ puede expresarse como*

$$\begin{cases} F_{\lambda_1}(m) + \sum_{i=1}^{k} \phi_{\lambda_1}(n_i) = n_1 \\ F_{\lambda_2}(m) + \sum_{i=1}^{k} \phi_{\lambda_2}(n_i) = n_2 \\ \vdots \\ F_{\lambda_k}(m) + \sum_{i=1}^{k} \phi_{\lambda_2}(n_i) = n_k \end{cases}$$

donde $N = (n_1, \ldots, n_k)$ es una solución a la ecuación que verifica que el mensaje y la posdata anterior forman una frase cierta.

Nota. *Al igual que en el caso de una letra, para que el anterior problema tenga sentido, fijada una familia de letras $\lambda_1, \ldots, \lambda_k$ añadiremos en el mensaje m' inicial también la parte de la posdata que se refiere a la frecuencia de aparición de la familia $\lambda_1, \ldots, \lambda_k$, para obtener el mensaje a firmar m. Esto es, si fijamos el siguiente conjunto de palabras como posdata:*

"en este mensaje aparecen • veces la letra λ_1, • veces la letra λ_2, ..., • veces la letra λ_k"

entonces el mensaje m contiene a m' mensaje inicial y al conjunto de palabras añadidas en la posdata.

De esta manera, surge el concepto de firma oficial que detallamos a continuación:

Definición 3.1. Dado un mensaje, llamamos firma oficial del mensaje al vector (en caso de que exista) N que es solución al problema de la firma para la familia de letras $a, b, \ldots, z$.

[6]Como en el caso de una letra, hay que hacer la distinción entre "veces" y "vez" en ciertos casos, pero, como dijimos, esto es fácilmente tratable ad hoc y no resta generalidad a ninguno de los resultados.

A priori, con la definición, se podría pensar que, para resolver el problema de la firma oficial, hemos de resolver un sistema con 27 incógnitas. No obstante, utilizando el Teorema 2.2, podemos simplificar un poco nuestro problema.

Lema 3.1. *Dado un mensaje m y una firma oficial N asociada al mismo, necesariamente se ha de satisfacer $n_i = F_\lambda(m)$, donde n_i es la coordenada asociada a $\lambda \in \{b, f, g, j, k, p, w, x\}$.*

Demostración. Basta aplicar el Teorema 2.2 y la ecuación del problema de firma. □

Por tanto, de ahora en adelante, cuando nos preocupemos del problema de la firma oficial, prefijaremos los valores de n_i asociadas a las letras de la proposición anterior, trabajando sobre un nuevo mensaje que incluirá, las posdatas y los valores de las letras ya fijadas.

3.2. Propiedades del sistema

De forma análoga a como se planteó el estudio de la ecuación de la firma, en el caso general nos centraremos en estudiar el sistema de ecuaciones que da lugar a la firma oficial.

3.2.1. Existencia de la firma

En general la existencia de la firma oficial no está garantizada para un mensaje cualquiera m. La siguiente proposición recoge esta afirmación:

Proposición 3.1 (Existencia de la firma oficial no garantizada)**.** *La firma oficial dada en la Definición 3.1 no siempre existe. Esto es, dado un mensaje puede no existir una firma oficial para el mismo.*

Demostración. Para ello, lo afrontaremos de un modo puramente teórico, y no nos centraremos en un ejemplo concreto que puede ser encontrado realizando el proceso inverso (fijando los números y buscando un mensaje tales que esa elección de números sea una firma oficial).

Supongamos, por tanto, que tenemos un mensaje original m' tales que al sumarle la posdata y adjuntarle los números de las letras fijas, dadas por el Teorema 2.2, cumple que $F_z(m) = 10$, por lo tanto sabemos que si N es una solución a la firma, $n_z \neq 10$, dado que $\phi_z(10) = 1$ y no se podría cumplir la condición de firma, por lo que en caso de ser N un vector solución a nuestro problema de la firma $n_z \geq 11$.

Veamos que, bajo cierta elección de $F_\lambda(m)$, es decir, para cierta elección de ocurrencia de letras en el mensaje, esto no es posible (nótese que, dado que hemos añadido la postada y los números de las letras fijas, las condiciones que impongamos sobre los valores de $F_\lambda(m)$ han de ser posibles, por ejemplo, no podemos pedir que $F_e(m) < 20$, pues esto sería incompatible con la inclusión de las posdata).

Al igual que hicimos con las letras fijas, tenemos que m y l sólo aparecen para números a partir del mil, por lo tanto, para un mensaje lo suficientemente corto, podemos dejarlas fijas y meterlas dentro de nuestro mensaje. Con ello, nos quedarán únicamente 15 letras para mover, donde suponemos que $\phi_z(n_l) = 0$ y $\phi_z(n_m) = 0$, para no perturbar el valor de la z.

Con todo esto en mente, supongamos que tenemos un mensaje que satisface las relaciones mostradas en la siguiente tabla, que uno puede comprobar con facilidad que cumplen las condiciones de ser compatibles con la posdata (donde ya hemos eliminado las letras fijas).

$111 \leq F_a(m) \leq 134$	$30 \leq F_c(m) \leq 34$	$11 \leq F_d(m) \leq 34$	$111 \leq F_e(m) \leq 134$	$11 \leq F_h(m) \leq 34$
$11 \leq F_i(m) \leq 34$	$11 \leq F_n(m) \leq 34$	$11 \leq F_o(m) \leq 34$	$11 \leq F_q(m) \leq 34$	$30 \leq F_r(m) \leq 34$
$30 \leq F_s(m) \leq 34$	$30 \leq F_t(m) \leq 34$	$11 \leq F_u(m) \leq 34$	$11 \leq F_y(m) \leq 34$	$F_z(m) = 10$

Bajo esta elección de mensaje inicial m', que da lugar al mensaje m con las condiciones descritas, afirmamos que no existe ninguna posible firma inicial. En efecto, dado que $n_z \geq 11$, hemos de tener que alguno de los valores $\phi_z(n_\lambda) \geq 1$, pero eso implicaría que tenemos alguna letra, λ tal que[7] $\sum_{i=1}^{15} \phi_\lambda(n_i) \geq 76$, dado que la única forma de añadir una z al número es mediante la terminación "*diez*" y todas las condiciones iniciales se encuentran a almenos esa distancia de algún número con dicha terminación.

Por otro lado, bajo observación de la tabla del Anexo A vemos que para todos los números $n \leq 350$ y todas las letras λ se satisface que $\phi_\lambda(n) \leq 5$, por lo tanto, suponiendo que los n_i son menores que 350, se sigue que $\sum_{i=1}^{15} \phi_\lambda(n_i) \leq \sum_{i=1}^{15} 5 = 5 \times 15 = 75 < 76$, lo cuál contradice la posibilidad de que $\phi_z(n_\lambda) \geq 1$ para cierto λ.

Por tanto, sólo basta demostrar que ningún vector con alguna de sus coordenadas mayor que 350 puede ser solución par este mensaje, pero esto es elemental, dado que si alguna de sus coordenadas fuera mayor que 350, se debería tener que para cierto λ se tiene que $\sum_{i=1}^{15} \phi_\lambda(n_i) \geq 200$, no obstante, por el mismo razonamiento que antes mirando la tabla, sabemos que para todo n menor que 999 y para todo λ se tiene que $\phi_\lambda(n) < 7$, por lo que se sigue que si $n_i < 999\ \forall i$, entonces $\sum_{i=1}^{15} \phi_\lambda(n_i) < 7 \times 15 = 105$, y por tanto n_λ no puede ser mayor o igual que 350.

Uno puede repetir el argumento todas las veces que considere conveniente, notando que cada vez necesita que las coordenadas sean más grandes, y, por tanto, es imposible que se den dichas condiciones, en otras palabras, el vector solución no puede existir.

□

Nótese cómo, las elecciones tan restrictivas sobre algunas letras para el mensaje sólo son necesarias para hacer la escritura formal de la prueba más sencilla, cuando uno puede elegir unas condiciones mucho menos restrictivas (por ejemplo, q aparece poco en los números, y por tanto la cota de $\phi_q(n) \leq 5$ para valores pequeños de n es una cota bastante mala), por lo que, en realidad, encontrar un mensaje que los restricciones impuestas no es tán complicado. Nótese, que, al final, tódo se reduce al hecho clave de que z aparece únicamente en números que terminen en "*diez*" y que el tamaño de $\phi_\lambda(n)$ no puede crecer demasiado rápido y, por tanto, los valores de n_lambda de una firma válida no se pueden desviar demasiado de los valores iniciales $F_\lambda(m)$

3.2.2. Unicidad de la firma.

Al igual que en el caso de una sola letra, nos preguntamos si en el caso de la firma oficial, en caso de existir, tenemos unicidad en la firma. No obstante, veremos que al igual que antes, la respuesta es en general negativa.

Proposición 3.2. *Sea m un mensaje y $N = (n_1, \ldots, n_{27}) \equiv (n_a, \ldots, n_z)$ su firma oficial. Si se cumple que $n_u = u_{c-1}u_{c-2}\ldots u_3u_1 3$ con $u_1 \neq 1$ y $n_s = s_{c-1}s_{c-2}\ldots s_2 14$, entonces la firma no es única.*
En particular el vector $\overline{N}$ dado por $\overline{n}_\lambda = n_\lambda\ \forall \lambda \neq u, s$ y $\overline{n}_u = u_{c-1}u_{c-2}\ldots u_3u_1 4$, $\overline{n}_s = s_{c-1}s_{c-2}\ldots s_2 13$ es también una firma válida.

[7]Donde hemos ordenada las coordenadas n_i de modo que solo sean los n_λ que no se correspondan con letras fijas.

Demostración. Basta notar que al hacer este cambio en la solución, por la formación de los números y el hecho de que $u_1 \neq 1$ y $s_1 = 1$ se sigue que, el cambio neto producido en las letras es un cambio equivalente al de cambiar "tres"por "trece" y "catorce"por "cuatro". En el primer cambio, ganamos una c, una e y perdemos una s, mientras que en el segundo cambio perdemos una c, una e y ganamos una u. Por tanto, el cambio neto en el mensaje es ganar una u y perder una s, que corresponde en incrementar en uno el valor de n_u y disminuir en uno el valor de n_s, que es exactamente lo que está sucediendo.
Por tanto, si N era una firma válida, se sigue que $\overline{N}$ también será una firma válida. En particular, la firma válida no es única. □

En un principio, uno podría preguntarse si se puede dar esta situación, es decir, si semejante mensaje con una firma. Para ello, uno puede seguir el procedimiento a la inversa, es decir, prefijar la firma oficial deseada y elegir un mensaje de modo que dicha firma sea solución, que es un procedimiento relativamente sencillo de realizar.

3.3. Complejidad del problema de firmar oficialmente

En este apartado cubrimos la complejidad del problema de firmar, resolviendo la incógnita planteada por los organizadores sobre la naturaleza de este problema. Probaremos en apartados posteriores que existe un algoritmo que pertenece a la clase de complejidad polinómica y resuelve este problema. Esto es, el problema no es *NP difícil* y de hecho pertenece a *P*.
En una primera aproximación un poco burda, uno puede encontrar una cota para la complejidad del problema.

Teorema 3.1. *Dado el un mensaje m de longitud L (de acuerdo a la Definición 2.4), y una familia de letras distintas $\lambda_1, \ldots, \lambda_k$, el problema de encontrar una firma válida para el mensaje en esa familia de letras tiene complejidad, a lo sumo, $O(\log^k(L))$.*

Demostración. El problema de encontrar una firma para la familia de letras, es equivalente a encontrar una solución $N = (n_1, \ldots, n_k)$ del sistema

$$\begin{cases} F_{\lambda_1}(m) + \sum_{i=1}^{k} \phi_{\lambda_1}(n_i) = n_1 \\ F_{\lambda_2}(m) + \sum_{i=1}^{k} \phi_{\lambda_2}(n_i) = n_2 \\ \vdots \\ F_{\lambda_k}(m) + \sum_{i=1}^{k} \phi_{\lambda_k}(n_i) = n_k \end{cases}$$

Si suponemos que $\overline{N}$ es una solución del sistema, sabemos que ha de satisfacer que, aplicando el Corolario 2.2

$$F_{\lambda_j}(m) \leq \overline{n_j} \leq F_{\lambda_j}(m) + C_1 \sum_{i=1}^{k} \lceil \log_{10}(\overline{n}_i) \rceil \leq F_{\lambda_j}(m) + kC_1 \lceil \log_{10}(\text{máx}\{\overline{n}_i, i = 1, \ldots, k\}) \rceil.$$

Sin pérdida de generalidad, y cambiando el orden de los índices si fuera necesario, podemos suponer que $\text{máx}\{\overline{n}_i, i = 1, \ldots, k\} = \overline{n}_1$, se tiene por tanto que

$$F_{\lambda_i}(m) \leq \overline{n}_i \leq F_{\lambda_i}(m) + C_2 \lceil \log_{10}(\overline{n}_1) \rceil.$$

donde $C_2 = kC_1$ es una nueva constante. Es decir, que para coordenada tenemos que comprobar a a lo sumo del orden de $O(\log(\overline{n}_1))$ valores[8], por lo que a lo sumo hemos de probar $O(\log^k(\overline{n}_1))$ para probar todos los posibles vectores que podrían ser soluciones. Por tanto, sólo hemos de estimar el tamaño de $\overline{n}_1$ para acabar nuestra prueba. Sabemos que

$$F_{\lambda_1}(m) \leq \overline{n}_1 \leq F_{\lambda_1}(m) + C_2\lceil \log_{10}(\overline{n}_1)\rceil.$$

Veamos que $F_{\lambda_1}(m) \leq \overline{n_1} \leq (C_2+1)F_{\lambda_1}(m)$. La primera desigualdad está ya demostrada, para la segunda basta notar que la función $\overline{n}_1 - C_2\lceil \log_{10}(\overline{n}_1)\rceil$ es una función creciente en $\overline{n}_1$, y dado que, obviamente $(C_2+1)F_{\lambda_1}(m) > F_{\lambda_1}(m) + C_2\lceil \log_{10}(F_{\lambda_1}(m))\rceil$ se sigue la desigualdad. Por tanto, se tiene que $\overline{n}_1 = O(F_{\lambda_1}(m)) = O(L)$, de lo que se sigue que $O(\log^k(\overline{n}_1)) = O(\log^k(L))$.

□

A lo largo de la prueba, no hemos sido demasiado cuidadoso con el tratamiento de las constantes, sin preocuparnos necesariamente si estas son las óptimas y utilizando una constante global para todas las letras en lugar de una constante para cada una de ellas, en función de la naturaleza de las mismas. Si bien esto funciona bien en el marco teórico, como forma práctica de asegurarse el rango de búsqueda necesario para encontrar la firma, no es eficiente, dado que se terminan obteniéndo rangos de búsqueda demasiado grandes. Por tanto, para la implementación del algoritmo, hemos de ser más cuidadosos a la hora de tratar con nuestras cotas.

[8]Nótese que se cambia de base del logaritmo a la base natural, como es convención, esto solo produce un cambio en las constantes multiplicativas y no cambia el orden de complejidad del problema.

4. El problema de crear firmas

En esta sección estudiaremos la validez del modelo de firmas presentado hasta el momento. En particular, detallaremos al principio los aspectos teóricos de las firmas digitales, así como las propiedades que el modelo utilizado en este documento aspiraba a satisfacer; comentando sus limitaciones. De esta manera, en posteriores secciones se presentan alternativas actuales que son utilizadas para la generación de firmas digitales, abordando las propiedades de estos modelos y sus diferencias con el propuesto.

4.1. Firmas manuscritas y firmas digitales

Las firmas manuscritas han constituido un método que históricamente vinculaba al firmante, identificándolo y comprometiéndolo, con el objeto firmado. De esta manera, se ha tratado de resolver el problema del repudio, esto es, que el receptor del mensaje pueda negar la participación del emisor en la comunicación recibida y en última instancia no pueda fiarse de la información que recibe. En este documento no discutiremos aspectos relativos al proceso de creación de la firma manuscrita que, en general, se considera un proceso físico intrínseco a la persona que firma; existiendo personas capaces de verificar la autenticidad del mismo.

Sin embargo, es importante estudiar cuáles son los objetivos que satisfacía la firma manuscrita históricamente para poder hacer frente a los principales retos que una firma digital debe solventar. Por ello, abordamos ahora dos propiedades fundamentales de la firma manuscrita:

- Cada firmante produce una firma única, que es difícil de reproducir por otros agentes.
- La firma producida por el firmante replica una misma grafía para los diferentes objetos firmados.

No obstante, en la firma digital, estas dos propiedades son difíciles de reproducir. La firma digital, en general, es producida por una máquina, de manera que esta genera una secuencia de bits asociadas al objeto firmado que debería de no ser reproducible por otras máquinas. Por otro lado, si el estado de los bits fuese el mismo para diferentes objetos firmados se diluiría completamente el valor de la firma, al ser fácilmente reproducible.

En general, y al igual que ocurría en las firmas manuscritas, la firma no debe depender únicamente del mensaje (si es que depende) pero siempre debe depender, de alguna manera, del emisor o firmante. Esto es, en el caso de la firma manuscrita es el firmante el único agente capacitado para generar la firma válida; en el caso de las firmas digitales, debemos asegurar con más información que la contenida en el objeto a firmar la autoría del firmante. Si esto no se cumpliese, el modelo de firma sería en sí fácilmente falsificable al depender únicamente del mensaje. Esto implicaría que la seguridad del sistema de firmas se encontraría justificado únicamente en el desconocimiento del proceso de generación de firmas, cuando para poder firmar mensajes es necesario el acceso a un sistema de generación de firmas. De esta manera, la seguridad del sistema se desplaza a dos hechos: un atacante no conoce el proceso de generación de firmas y no tiene acceso a un generador de firmas.

Por otro lado, la estrategia propia de los sistemas de seguridad por oscuridad puede verse comprometida fácilmente en cualquier momento en el que se libere o se descubra el secreto, y su seguridad no reside en una prueba teórica de su solidez como generador de firmas seguras sino en el desconocimiento probable del atacante. Relacionado con esta idea se encuentra el más conocido de los Principios de Kerckhoffs sobre los sistemas criptográficos [Pet11]:

II Principio de Kerckhoffs. *La seguridad de un criptosistema debe medirse suponiendo que el atacante conoce completamente el proceso de cifrado y descifrado. En general, la efectividad del sistema no debe depender de que su diseño permanezca en secreto.*

Con el objetivo de respetar el Segundo Principio de Kerckhoffs en los modelos de firmas digitales, buscaremos sistemas cuya justificación de su solidez y seguridad se cimienten en desarrollos teóricos y demostraciones formales y no en el desconocimiento del modo de funcionamiento del sistema por parte de los atacantes.

Por otro lado, hemos visto que las firmas digitales no pueden satisfacer las propiedades de las firmas manuscritas y conseguir con esto validar la seguridad del sistema. En el caso de las firmas digitales buscamos propiedades distintas que puedan asegurar la seguridad del sistema frente a ataques de agentes externos que quieran falsificar los mensajes. Recogemos a continuación algunas de las propiedades más importantes que sirven para justificar la seguridad de un sistema de firmas digitales.

Definición 4.1 (Propiedad de autenticación)**.** La firma digital debe de ser fácil de autenticar, cualquier receptor tiene que poder establecer la autenticidad del mensaje.

Esta propiedad nos asegurará que podemos conocer al emisor del mensaje y asegurar que es quien dice ser; además compromete al emisor con el mensaje emitido sin que este pueda negar su autoría.

Definición 4.2 (Propiedad de univocidad)**.** La firma digital debe identificar unívocamente al emisor.

Esto es, una firma digital debe hacer referencia siempre a un emisor de forma que este no pueda variar y quedando totalmente determinado.

Definición 4.3 (Propiedad de no falsificabilidad)**.** La firma digital debe de ser fácil de generar para el emisor, pero difícil de replicar para atacantes externos.

Estas propiedades no recogen explícitamente la necesidad que hemos resaltado anteriormente de que el sistema no dependa únicamente del propio mensaje para generar la firma, sin embargo, es fácil ver que se cumple el siguiente resultado:

Proposición 4.1. *Suponiendo que se verifica el II Principio de Kerckhoffs, si un sistema de firmas depende únicamente del mensaje para generar la firma, entonces incumple las propiedades de autenticación, univocidad y no falsificabilidad.*

Demostración. Trivialmente se incumplen las propiedades de autenticación y univocidad, al no depender del firmante y solo del mensaje a firmar de manera que cualquiera puede firmar un mensaje y no puede verificarse ni conocerse el autor. Por otro lado, si el sistema depende únicamente del mensaje cualquier agente externo puede generar igualmente la firma pudiendo falsificar mensajes. □

4.2. El modelo de firmar mensajes con frecuencias de letras

El modelo de firma de mensajes con frecuencias de letras estudiado a lo largo de este documento busca ser un modelo de firma digital. De esta manera, busca ser un sistema que certifique la veracidad e integridad del contenido del mensaje generando una posdata sensible a cambios en el mensaje. Con todo, este modelo de firmas aspira a cumplir la Propiedad de no falsificabilidad, aunque no las propiedades expuestas de univocidad y autenticación. Sin embargo, a partir de la Proposición 4.1 podemos obtener el siguiente corolario:

Corolario 4.1. *El modelo de firmar construyendo una posdata con información sobre las frecuencias de las letras en el mensaje y la posdata no cumple la propiedad de no falsificabilidad.*

El resultado anterior es debido a que el modelo de firmar basándose en la frecuencias de letras en el mensaje y la posdata constituye un modelo de firmas dependiente exclusivamente del mensaje.

Por otro lado, si la seguridad del sistema anterior se cimienta en incumplir el Segundo Principio de Kerckhoffs nos encontraríamos en un sistema de seguridad por oscuridad donde no existe seguridad real del sistema, sino seguridad práctica en la medida en la que no se descubra el método de funcionamiento del algoritmo generador de firmas. Sin embargo, este no es el único problema del que adolece este sistema de firma de mensajes.

Un problema también relavante de este modelo de firmas es que no siempre es posible firmar un mensaje dado, esto es, como ya se probó en apartados anteriores no se puede asegurar la existencia de firmas para cualquier mensaje. Un problema menor, pero también interesante, es que existen además múltiples firmas para un mismo mensaje. Por ello, este sistema no es ni siquiera posible de implementar en la prácticar por no ser siempre generable.

Aparte de los argumentos anteriormente expuestos, existen argumentos contrarios a este modelo de firmas por motivos de autenticación y univocidad. El sistema de firmas con frecuencias no permite identificar al emisor del mensaje, solamente justifica (de forma susceptible a ataques de falsificaciones) la integridad del mensaje. Esto es, no permite conocer al emisor y comprometer a este con el mensaje emitido, de manera que cualquier atacante podría suplantar cualquier identidad simplemente conociendo la manera de generar firmas.

4.3. Firmando mensajes de forma segura

Una vez expuestas las bases de los sistemas de firmas digitales y las propiedades más relevantes que deben verificar, así como argumentada una crítica al sistema de firmas con frecuencias de letras, pasaremos a exponer un ejemplo de un sistema actual de firmas digitales.

En general, existen dos tipos de modelos de firmas digitales basados en la intervención o no de una tercera persona: los esquemas arbitrados y los no arbitrados. Como sus propios nombres indican, el primero requiere de una tercera parte (árbitro) que decide la autenticidad del mensaje y el emisor, mientras que el segundo tipo no necesita de la intervención del árbitro para que el receptor pueda autenticar el mensaje. A su vez, también podemos dividir los esquemas de firmas digital en los basados en criptografía simétrica y los basados en sistemas asimétricos.

Centrándonos en los modelos no arbitrados de firma digital asimétricos, de clave pública, podemos extraer un esquema de funcionamiento general [Gil03]:

1. Se seleccionan una pareja de claves privada/secreta y pública utilizando un algoritmo de generación de claves.

2. Tomando como entrada el mensaje y la clave secreta, un algoritmo de firma genera una firma digital.

3. Un algoritmo verificador recibe la firma y la clave pública, decidiendo si la firma es válida o no.

Un ejemplo de sistema de firma digital basado en el esquema anterior es la firma RSA, que utiliza como adaptación del conocido algoritmo de encriptación RSA. En su versión original, la clave generada era independiente del mensaje, aunque esto hacía vulnerable al sistema. En implementaciones actuales la clave privada se genera también haciendo uso de información del mensaje, evitando posibles vulnerabilidades que permitían cifrar mensajes sin conocer la clave privada pero conociendo las firmas de diferentes mensajes [JC86].

Este tipo de sistemas sí que verifican las propiedades que expusimos con anterioridad de autenticación, univocidad y no falsificabilidad. Además, todos estos cumplen el segundo Principio de Kerckhoffs, basando su seguridad no en el desconocimiento del sistema por parte del atacante si no en la dificultad de resolver un problema matemático para generar las claves o descodificar las firmas.

5. Conclusiones y futuros pasos

A lo largo de este documento hemos cubierto un estudio del modelo de firmas basado en la frecuencia de aparición de las letras en el mensaje y en la propia firma. De esta manera, hemos formalizado el problema de manera análiticа y expuesto algoritmos de cálculo de firmas para el caso simple de una sola letra, y hemos estudiado su complejidad. De forma parecida hemos procedido con modelo de firma oficial, con todas las letras, estudiando la complejidad de este problema y demostrando que no es *NP difícil* sino que pertenece a la clase *P*.

Por otro lado, hemos expuesto algunas de las debilidades de este modelo como sistema de firmas digitales. Entre ellas, podemos destacar la no existencia de firma en todos los casos o la ausencia de univocidad entre firmas y mensajes. Es más, en los últimos apartados hemos comentado problemas relacionados con la falsificabilidad y la incapacidad de autenticar al emisor del mensaje con firma.

A lo largo del documento se han ido comentando aspectos relativos a la generabilidad del modelo teórico que hemos presentado, sin embargo, queremos destacar nuevamente en este apartado que todos los resultados propuestos son independientes del lenguaje siempre y cuando este lenguaje posea un patrón regular en la construcción de los cardinales. Esta propiedad se cumple para todos los idiomas más hablados del mundo: español, inglés, chino, francés... De esta manera, cualquier resultado aquí expuesto puede replicarse ajustando la función G, variando probablemente las cotas pero manteniendo la naturaleza logarítmica de la complejidad del problema.

Finalmente, nos parece oportuno mencionar algunas posibles áreas a estudiar que podrían ampliar los resultados comentados en este documento. De esta manera, los siguientes apartados recogen propuestas de ampliación del contenido de este trabajo.

5.1. Algoritmo de generación de firmas oficiales eficiente

Si bien el Apartado 3.3 ofrece una demostración constructiva para acotar la complejidad del problema, replicar en un ordenador actual la metodología es impracticable. Los valores de las constantes multiplicativas y del logaritmo exponencial impiden una implementación eficiente del algoritmo. Sin embargo, reutilizando resultados como el Lema 3.1 pueden encontrarse formas de reducir el coste de implementación real del algoritmo. En particular, puede existir una forma eficiente de implementar una solución que fuese fuertemente dependiente del lenguaje, por ejemplo aprovechándose del conocimiento de la frecuencia de aparición en todas las palabras y prefijos que construyen los cardinales en el español (véase el Anexo A).

5.2. Un sistema de firma digital seguro

De forma parecida a como se ha cubierto en el Apartado 4.3 un ejemplo de firma digital segura. Un problema del que adolecen la mayoría de sistemas de firmas digitales es el tamaño de los mensajes, para lo que se suele recurrir a funciones de Hash que reducen el contenido a un resumen del mismo. De esta manera, podría estudiarse el esquema propuesto basado en la firma oficial como función de Hash. Para ello, habría que estudiar como construir una envoltura que complete los casos para los que no existen firmas oficiales.

Anexo A: Frecuencias de letras al construir cardinales

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
una	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
dos	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
tres	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
cuatro	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0
cinco	0	0	2	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
seis	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
siete	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
ocho	0	0	1	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
nueve	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0
diez	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
once	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
doce	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
trece	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
catorce	1	0	2	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0
quince	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0
dieci	0	0	1	1	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
veinti	0	0	0	0	1	0	0	0	2	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0
treinta	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	2	0	0	0	0	0	0
cuarenta	2	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0
cincuenta	1	0	2	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	1	0	0	0	0	0
sesenta	1	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	2	1	0	0	0	0	0	0
setenta	1	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	1	2	0	0	0	0	0	0
ochenta	1	0	1	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0
noventa	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0
cien	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
ciento	0	0	1	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0
cientos	0	0	1	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0
mil	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
millon	0	0	0	0	0	0	0	0	1	0	0	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0
millones	0	0	0	0	1	0	0	0	1	0	0	2	1	1	1	0	0	0	1	0	0	0	0	0	0	0

Referencias

[Esp70] Real Academia Española. *Real Academia Espanola: Diccionario de la* Lengua Española*. Espasa-Calpe, 1970.

[JC86] Wiebren de Jonge y David Chaum. "Some Variations on RSA Signatures and Their Security". En: *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*. Vol. 263. Lecture Notes in Computer Science. Springer, 1986, págs. 49-59. DOI: 10.1007/3-540-47721-7_4.

[Gil03] Pino Caballero Gil. *Introducción a la criptografía*. Ra-Ma, 2003.

[Pet11] Fabien A. P. Petitcolas. "Kerckhoffs' Principle". En: *Encyclopedia of Cryptography and Security*. Ed. por Henk C. A. van Tilborg y Sushil Jajodia. Boston, MA: Springer US, 2011. ISBN: 978-1-4419-5906-5. DOI: 10.1007/978-1-4419-5906-5_487. URL: https://doi.org/10.1007/978-1-4419-5906-5_487.