Eina per al disseny i correcció d'exàmens de tipus test

Miquel Dalmau**, Sebastià Martín* i Jordi Saludes**

(*) Dep. Matemàtica Aplicada IV. Universitat Politècnica de Catalunya
Campus Nord, c/Jordi Girona, 1-3, 08034 Barcelona.
(**) Dep. Matemàtica Aplicada II. Universitat Politècnica de Catalunya
c/Colom, 11, 08223 Terrassa, Barcelona.
e-mail: sebasm@mat.upc.es, jordi.saludes@upc.es, miquel.dalmau@upc.es

Resumen

En este trabajo presentamos una aplicación del clásico criptosistema knapsack al diseño de exámenes de tipo test. Una de las ventajas frente a un test tradicional es que posibilita la prevención y detección del fraude por copia, durante la realización del examen. El sistema permite al alumno obtener su calificación inmediatamente después del examen, así como liberar al profesor de la corrección del examen, y generar automáticamente una base de datos con las calificaciones del examen. Se comenta también su puesta en práctica, a modo de experiencia piloto, en la ETSEIT de la UPC.

1. Introducción

El clásico examen de tipo test ha sido y es una opción a considerar en la evaluación de estudiantes. Sin embargo, frente a los exámenes escritos tradicionales, presenta el inconveniente de disminuir las posibilidades de prevención y detección del eventual fraude por parte de algún estudiante, especialmente cuando hay masificación en las aulas. Una posible solución a este problema ha sido la permutación de preguntas y respuestas, de manera que cada estudiante reciba un modelo de examen de apariencia distinta.

En este trabajo, exponemos una aplicación del criptositema knapsack a la generación de exámenes de tipo test más eficaces frente al fraude por copia. Se ha elegido precisamente este criptosistema porque permite codificar las respuestas del test de manera sencilla y, a la vez, supone poco esfuerzo adicional para el estudiante, dado que el cifrado consiste en sumar números enteros, de manera que el resultado del examen queda codificado en un sólo número. Además, el estudiante puede obtener su nota inmediatamente después de la finalización del examen, consultando la página web de la asignatura. El sistema permite liberar al profesor de la corrección del examen, así como generar automáticamente una base de datos con las calificaciones del examen.

La propuesta ha sido implementada, usando software de libre distribución, en la asignatura Métodos Matemáticos de la Escuela Superior de Ingeniería Industrial de Terrassa (ETSEIT) de la UPC, y ha suscitado interés, por parte de los estudiantes, sobre las herramientas criptográficas utilizadas. La implementación está basada en un sistema *open-source* de administración de contenidos web, Zope [8], que permite una gran flexibilidad en la atribución de permisos (qué puede hacer cada persona con cada documento). Zope admite extensiones escritas en lenguaje Python [5], en el que se han desarrollado una serie de productos que ayudarán al profesor en la evaluación.

El artículo está organizado de la siguiente forma: en la sección 2 se expone el criptosistema knapsack, en la sección 3 se explica la realización de los exámenes de tipo test, en la sección 4

se muestra una adaptación del criptosistema knapsack que permite optimizar el diseño de tales exámenes, la seguridad se discute en la sección 5 y finalmente se exponen las conclusiones.

2. El criptosistema Knapsack

Los criptosistemas de tipo knapsack se basan en el problema *subset sum*, o problema de la mochila, descrito a continuación:

Dados S, $a_1, \ldots, a_s \in \mathbb{N}$ encontrar, si existen, $x_1, \ldots, x_s \in \{0, 1\}$ tales que $S = \sum_{i=1}^{s} x_i a_i$ (es decir, averiguar qué sumandos componen S). Podemos pensar que tenemos una mochila cuyo peso total es S, y el problema es determinar qué pesos a_i hay que meter en la mochila para llenarla. Si el número s de posibles pesos es grande, el problema es, en general, difícil (tradicionalmente, 160 se solía considerar una cantidad de pesos suficiente para garantizar la seguridad frente a ataques de fuerza bruta). Sin embargo, en el caso particular de que la sucesión $a_1, \ldots, a_s$ sea supercreciente, $a_i > \sum_{j=1}^{i-1} a_j \ \forall i$, entonces es fácil encontrar los pesos a_i cuya suma es S, comparando S con cada uno de los pesos, tomados en orden decreciente.

Esta idea permitió a Merkle y Hellman definir una función unidireccional con puerta falsa, que dio lugar, en el año 1978, al criptosistem Knapsack [2]. Partiendo de una sucesión $a_1, \ldots, a_s$ supercreciente, se consideran $N > \sum_{i=1}^{s} a_i$, $m \in \mathbb{Z}_N^*$, y se calcula una nueva sucesión $c_i \equiv a_i m \pmod{N}$ $\forall i = 1, \ldots, s$. Veamos la descripción del criptosistema:

- *Clave pública*: $c_1, \ldots, c_s$.
- *Clave secreta*: m, N.
- *Mensajes*: $M = x_1 x_2 \cdots x_s$, donde $x_i \in \{0, 1\}$.
- *Cifrado*: $C = \sum_{i=1}^{s} x_i c_i$.
- *Descifrado*: se calcula $Cm^{-1} \equiv \sum_{i=1}^{s} x_i a_i \pmod{N}$, y se obtienen los x_i resolviendo un problema fácil, dado que ahora los sumandos $a_i = m^{-1} c_i \pmod{N}$ forman una sucesión supercreciente.

En 1982 encontramos el primer ataque al criptosistema de Merkle y Hellman, ideado por Shamir [7]. A partir de ese momento han surgido variantes del criptosistema original, que en muchos casos han sido atacadas. Para un resumen sobre los criptosistemas de tipo knapsack y los ataques que han sufrido, ver [4]. Más recientemente se han diseñado criptosistemas de tipo knapsack resistentes, como por ejemplo el de Naccache y Stern [3], de 1997.

Para nuestros propósitos, la versión original de Merckle y Hellman es suficientemente robusta, como veremos en la sección 5.

3. Exámenes de tipo test

3.1. Preparación

Para el diseño de p permutaciones distintas de un examen de tipo test con r preguntas, donde la pregunta i-ésima tenga n_i respuestas ($i = 1, \ldots, r$) se eligen:

- Una sucesión supercreciente de $s = \sum_{i=1}^{r} n_i$ elementos, $a_1, \ldots, a_s$.
- Un módulo $N > 2a_s$.
- Multiplicadores m_j, tal que $\gcd(m_j, N) = 1$, para $j = 1, \ldots p$.

Luego, siguiendo el método del criptosistema Knapsack [2] expuesto en la sección 2, se construye la sucesión $c_i = m_j a_i \pmod{N}$, con $m_j = \alpha^j \pmod{N}$ y $\alpha \in \mathbb{Z}_N^*$ de orden mayor que p. Los enteros $c_1, \ldots, c_s$ constituirán los códigos de las respuestas en la j-ésima permutación del examen (figura 1).

Primer Parcial **Mètodes Matemàtics I**
2 de Novembre de 2000

Per a cada pregunta sumeu el número que es troba a l'esquerra de la resposta correcta. Si no voleu contestar una pregunta no sumeu res. Escriviu el total rera el símbol **Suma**. Dividiu aquest total per 7. El resultat de la divisió ha de ser exacte, si no, us heu equivocat al sumar.

1. Si designem per e_r l'error relatiu i e_a l'error absolut, quina de les següents afirmacions és certa?

4194197 ◇ $e_r(x+y)$ és aproximadament $e_r(x) + e_r(y)$.
2211755 ◇ $e_r(xy) = e_r(x) + e_r(y)$.
229313 ◇ $e_a(xy) = e_a(x) + e_a(y)$.
4423510 ◇ $e_r(xy)$ és aproximadament $e_r(x)+e_r(y)$.

2. Sigui $p(x)$ el polinomi interpolador dels valors de la taula d'abscisses $x = \{-1, 1, 4\}$ i ordenades $f = \{a, b, c\}$ Llavors:

2441068 ◇ $p(0) > 0$, $\forall a > 0$, $b > 0$ i $c < 0$.
4882136 ◇ $p(0) < 0$, $\forall a > 0$, $b < 0$ i $c > 0$.
1146565 ◇ $p(1) = a$.
3587633 ◇ Cap de les anteriors no és certa.

3. Sigui $p(x)$ el polinomi interpolador de la funció $f(x) = \ln(4+x)$ en les abscisses $G = \{-1, 0, 2, 3\}$, i sigui $e(x)$ la fita de $|f(x) - p(x)|$ donada per la fórmula de l'error d'interpolació en un punt x. Llavors:

6028701 ◇ $e(1) < 0.02$.
5880763 ◇ $e(1) \geq 0.03$.
5732825 ◇ $e(0) > 10^{-5}$.
5584887 ◇ Cap de les anteriors no és certa.

4. Quina de les següents afirmacions sobre el mètode d'interpolació de Neville és certa?

5436949 ◇ Permet evitar el fenomen de Runge.
4697259 ◇ És útil per avaluar el polinomi interpolador en un punt prefixat.
3957569 ◇ Si les abscisses no són equiespaiades, té un cost computacional més baix que el mètode de les diferències de Newton.
3217879 ◇ Cap de les anteriors no és certa.

5. Si x és la representació en base 2 del número 1.6, llavors:

2478189 ◇ x té infinits dígits binaris i és periòdic.
4956378 ◇ $x > (1.101)_2$.
1257928 ◇ x té infinits dígits binaris i no és periòdic.
3736117 ◇ Cap de les anteriors no és certa.

6. Quan aproximem per mínims quadrats la funció $f(x) = 1/x$ a l'interval $[1, 2]$ pel polinomi $p(x) = x/2$, l'error és:

37667 ◇ 0.08330
75334 ◇ 0.2887
113001 ◇ 0.05685
150668 ◇ 0.2384

7. Considerem el producte escalar amb pesos $w = \{1, 2, 4\}$ sobre la xarxa $G = \{0, 1, 2\}$. Siguin $p(x) = 1$, $q(x) = x + a$ i $r(x) = x^2$. Llavors:

188335 ◇ q i r no són ortogonals per cap valor de a.
376670 ◇ q i r són ortogonals per algun valor positiu de a.
565005 ◇ p i q són ortogonals quan $a = -10/7$.
753340 ◇ Cap de les anteriors no és certa.

Codi: 97865 Versió: 599171 **DNI**: **Suma**: $\frac{\textbf{Suma}}{7}$:

Figura 1: Ejemplo de examen

En nuestra implementación práctica [6], a partir de un fichero que contiene las preguntas y respuestas (con indicación de la respuesta correcta), el sistema genera una copia del examen para el profesor y tantas copias distintas (obtenidas por permutación de preguntas y respuestas) como estudiantes.

3.2. En el aula

Al inicio del examen, cada estudiante recibe una hoja con las cuestiones como la de la figura 1. Elegir una respuesta consiste en seleccionar su código. El resultado final que ha de entregar cada estudiante es la suma $\sum$ de todos los números correspondientes a les respuestas escogidas. De esta manera, la obtención de la nota correspondiente a una permutación j (y, de hecho, de cada una de las respuestas elegidas por el estudiante), se reduce a un descifrado usando el criptosistema knapsack con clave secreta (m_j, N).

Es importante, a efectos prácticos, introducir algún sistema de detección de eventuales errores por parte del estudiante en el cálculo de la suma $\sum$. Por ejemplo, imponiendo que todos los códigos de respuesta sean múltiplos de 7, y pidiendo que verifiquen mediante una división que $\sum /7$ es entero. Por otra parte, para aumentar la seguridad contra el eventual fraude, se puede pedir al estudiante que no haga marca alguna en la hoja de examen, dado que el un número $\sum$ ya compacta toda la información necesaria sobre las respuestas seleccionadas.

En el caso de que la asignatura disponga de una página web, al final del examen el profesor proporciona al estudiante un número de control c, que servirá para evitar que el estudiante modifique $\sum$ en el momento de obtener su calificación al acceder a la página web de la asignatura (ver sección 5). En tal caso, el estudiante sale del aula de examen con un *recibo* formado por 3 números (ver parte inferior en la figura 1):

- El número de versión v, que permite determinar el modelo de examen que el estudiante ha realizado, es decir, tanto la permutación como el multiplicador m_j.
- La suma total $\sum$.
- El número de control c.

Por su parte, el profesor se queda con la hoja de cuestiones, que contiene el recibo original.

3.3. Calificaciones

En nuestra implementación, una vez finalizado el examen, el profesor activa la página web donde cada estudiante introducirá la tripleta $\sum, v, c$, junto con su D.N.I. para identificarse. Si los datos son correctos, el estudiante obtiene:

- Su calificación.
- Cúales eran las respuestas correctas en su modelo de examen, y cúales había elegido.

Una vez consultada, la calificación es almacenada en una base de datos docente. Las notas se pueden publicar directamente a partir de la base de datos.

Si la asignatura no dispone de página web, puede usarse un ordenador portátil para descifrar, en el aula mismo, el resultado aportado por cada estudiante, de manera que los estudiantes pueden conocer su calificación en el momento de abandonar el aula.

4. Versión optimizada

Hay que tener en cuenta que los estudiantes utilizarán calculadoras (normalmente de 8 o 10 dígitos) para sumar los códigos de respuesta, lo cual constituye una limitación en relación al número de preguntas y respuestas que puede contener una hoja de examen. Una posible solución es confeccionar dos (o más) hojas de examen para un mismo test, cada una de ellas

cifrada usando una sucesión supercreciente distinta. En cualquier caso nos interesará diseñar problemas de la mochila en el que los pesos tengan el menor tamaño posible. La sucesión supercreciente de crecimiento más lento es: $a_i = 2^{i-1}$, $i = 1, \dots r$. Una condición para que cualquier número obtenido como suma de códigos de respuestas posibles (todos ellos múltiplos de 7, por la razón expuesta en la subsección 3.2) tenga, a lo sumo, 10 dígitos, es $7(2a_r - 1) < 10^{10}$, que cuando $a_i = 2^{i-1}$ $(i = 1, \dots r)$, equivale a $r \leq 30$ (10 preguntas con tres respuestas posibles cada una, por ejemplo). Ahora bien, si cada pregunta del test tiene una sola respuesta correcta, existe una solución mejor [6], que pasamos a exponer a continuación.

Definición 1 *Sea $\mathcal{B} = \{\mathcal{B}_1, \dots, \mathcal{B}_r\}$ una colección de subconjuntos disjuntos de enteros positivos. Dado $S \in \mathbb{N}$, llamaremos problema de la mochila $(\mathcal{B}, S)$ al problema de determinar, para todo $i \in \{1, \dots, r\}$, los enteros $b_i \in \mathcal{B}_i \cup \{0\}$ tales que $S = b_1 + \cdots + b_r$.*

Si pensamos en los subconjuntos $\mathcal{B}_i$ como bolsas, el problema de la mochila $(\mathcal{B}, S)$ se puede formular de la siguiente manera: determinar si hay que incluir algún peso de cada una de las bolsas $\mathcal{B}_i$ (y cúal), para que el peso total de la mochila sea S.

Proposición 2 *Consideremos el problema de la mochila $(\mathcal{B}, S)$, con $\mathcal{B} = \{\mathcal{B}_1, \dots, \mathcal{B}_r\}$. Sea $\mathcal{B}_r = \{b_{r,1}, \dots, b_{r,n_r}\}$, con $0 = b_{r,0} < b_{r,1} < \dots < b_{r,n_r}$. Supongamos que se satisface la siguiente condición:*

$$b_{i,j} - b_{i,j-1} > \sum_{k=1}^{i-1} \text{máx } \mathcal{B}_k \,, \ \forall\, j \in \{1, \dots r\}, \forall\, i \in \{1, \dots r\}. \tag{1}$$

Entonces, si el problema de la mochila $(\mathcal{B}, S)$ tiene solución, ésta es única, y puede hallarse comparando S con cada uno de los elementos de $\mathcal{B}_1 \cup \cdots \cup \mathcal{B}_r$ en orden decreciente.

Ejemplo 3 *Una colección de subconjuntos que verifican la condición (1) es:*

$$\mathcal{B}_1 = \{1, 2, 3, 4\}, \quad \mathcal{B}_2 = \{5, 10, 15\}, \quad \mathcal{B}_3 = \{20, 40, 60, 80, 100\},$$
$$\mathcal{B}_4 = \{120, 240\}, \quad \mathcal{B}_5 = \{360, 720, 1080, 1440\}.$$

Volviendo al diseño de tests, $\mathcal{B}_i$ será el conjunto de códigos de respuesta para la pregunta i, antes del cifrado. Es decir, podemos usar los elementos de $\mathcal{B}_1 \cup \cdots \cup \mathcal{B}_r$ en sustitución de las sucesiones supercrecientes.

Veamos ahora una manera de construir los subconjuntos $\mathcal{B}_1, \dots, \mathcal{B}_r$, dado sus cardinales $n_1, \dots, n_r$, de manera que sus elementos sean lo más pequeños posible.

Proposición 4 *Sea w_i el primer elemento de la bolsa $\mathcal{B}_i$, $i = 1, \dots, r$. Elegimos estos elementos de la siguiente forma:*

$$w_1 = 1,$$
$$w_i = (1 + n_{i-1})w_{i-1} \quad \textit{for } i = 2, \dots, r$$

y definimos $\mathcal{B}_i = \{jw_i \mid j = 1, \dots, n_i\}$ para $i = 1, \dots, r$. Entonces, la colección de subconjuntos $\mathcal{B}_1, \cdots, \mathcal{B}_r$ satisface de manera óptima la condición (1).

Observemos que el ejemplo 3 satisface las condiciones de la anterior proposición.

En la siguiente tabla podemos ver el número máximo de preguntas que pueden incluirse en una sola hoja de examen, en el caso que el número n de respuestas posibles a cada pregunta sea fijo. La primera columna corresponde al uso de una sucesión supercreciente, mientras en la segunda figura el caso optimizado.

n	Supercreciente	Optimizada
2	13	16
3	8	12
4	6	11
5	5	10

5. Consideraciones de seguridad

5.1. Escenarios

Hay varias situaciones de utilización del sistema propuesto, respecto al procedimiento de corrección, que consiste en calcular la nota n a partir del vector de sumas S (cuyas coordenadas son las sumas $\sum$ correspondientes a cada una de las hojas de examen), el número de versión v y el identificador id del alumno (normalmente su NIF).

$$n = \text{nota}(S, v)$$

Por lo tanto distinguimos los escenarios siguientes:

Directo El alumno entrega el examen y el profesor proporciona la nota n a partir de (S, v) usando una calculadora o un ordenador portàtil. En este último caso, la información (id, S, v, n) queda anotada en la base de datos del examen.

Diferido El profesor recoge las hojas de examen y en su despacho prepara un fichero con (id, S, v) para cada alumno. Un programa calcula la nota n asociada a cada tripleta y la añade a la base de datos.

Autoservicio El alumno entrega la hoja de examen, el profesor anota en el recibo el número de control c asociado a la tripleta (id, S, v). El alumno se retira, conecta con la página web de corrección y mediante el recibo, rellena el formulario de corrección. En la pantalla aparece la nota n, las respuestas del alumno y las correctas. La información (id, S, v, n) queda automáticamente registrada en la base de datos del examen.

Los objetivos de seguridad del sistema dependen de qué situación se utilice. Consideramos los siguientes riesgos:

1. Copia de respuestas de un compañero (consentida o no)
2. Apropiación de los datos (S, v) de un compañero que tenga una mejor calificación con la intención de conseguir la misma nota que éste.
3. Sustitución de S por otra suma S' construida a posteriori a partir de v y de los c_i correspondientes a las respuestas correctas.

La correcciones directa y diferida son la más seguras (sólo se ven afectadas por el riesgo 1) mientras que la corrección en régimen de autoservicio es la más cómoda pero adolece además de los riesgos 2 y 3.

5.2. Protección contra el riesgo de copia

El sistema se ha diseñado para reducir este tipo de riesgo. Consideremos los números de versión v y v' asociados a dos alumnos distintos id y id'. Sean $(c_i)_{i=1,\ldots,r}$ y $(c'_i)_{i=1,\ldots,r}$ los códigos de respuesta en sus respectivas hojas de examen. El hecho de que dichos códigos numéricos sean función del número de versión impide al alumno id' compartir respuestas o sumas con el alumno id, a no ser que $v = v'$, o bien pueda obtener $(c'_i)_{i=1,\ldots,r}$ a partir de $(c_i)_{i=1,\ldots,r}$, para lo cual se requiere el módulo N (que es secreto). A pesar de que este valor se pueda conseguir por criptoanálisis con todos los datos, es improbable que se pueda realizar en el aula, con las restricciones de tiempo del examen y dada la dificultad de obtener todos los códigos de respuesta del compañero.

5.3. Proteccion contra los riesgos adicionales

En el caso de autoservicio, hay que evitar que S o v sean sustituidos, fuera del aula, por otros datos con mejor calificación. Para esto se propone usar una función de *hash* para calcular un número de control c, que debe introducirse juntamente con S y v en el formulario web. Para impedir el riesgo 3, c debe ser como mínimo función de S. Para prevenir el riesgo 2 hay dos posibilidades:

- Que v sea inyectiva como función de id: a cada alumno se le asigna una versión distinta, con lo cual la apropiación de la S de un compañero resulta inviable.
  Se requiere que el sistema corrector compruebe que en la base de datos no hay dos entradas (id, S, v, n) y (id', S, v, n) con $id' \neq id$. En este caso se alerta al profesor y éste comprueba las hojas de examen para ver si hay fraude.
- Construir c como función de id y S. Una posibilidad es usar una función de *hash* h de dos argumentos para encadenar id y cada uno de las componentes de $S = (S_1, \ldots S_L)$, donde L es el número de páginas del examen. Así,

$$c_1 = h(id, S_1),\ c_2 = h(c_1, S_2),\ \ldots\ c = h(c_{L-1}, S_L)$$

En cualquier caso, recordemos que los exámenes restan en posesión del profesor, para eventuales comprobaciones manuales.

6. Conclusiones

Hemos presentado una aplicación del criptositema knapsack al diseño de exámenes de tipo test, con las siguientes características:

- Las respuestas se codifican de manera sencilla y compacta, con poco esfuerzo de cálculo para el estudiante.
- Se trata de exámenes más eficaces frente al fraude por copia que los tests tradicionales.
- Los estudiantes pueden conocer su nota en el momento de finalizar el examen.
- Permite liberar al profesor de la corrección del examen.
- Permite generar automáticamente una base de datos con las calificaciones del examen.

Se ha adaptado convenientemente el criptosistema knapsack original [2] para optimizarlo de cara al diseño de tales exámenes. En concreto, para poder incluir en una sola hoja de exámenes el máximo número de preguntas, teniendo en cuenta las limitaciones computacionales de los estudiantes en el aula de examen.

Se ha estudiado la seguridad de nuestro trabajo, y se han proporcionado soluciones para aumentarla, según los diversos escenarios de aplicación.

Se ha detallado cómo llevar a cabo nuestra propuesta, a partir de una experiencia de implementación real en una asignatura de matemáticas de la Escuela Superior de Ingeniería Industrial de Terrassa (ETSEIT) de la UPC, usando software de libre distribución.

Referencias

[1] B. Chor and R. Rivest. A Knapsack-type public key cryptosystem based on arithmetic in finite fields. *Advances in cryptology: Proceedings of Crypto'84*. LNCS Springer-Verlag 1985 Berlin-Heidelberg. 54–65. Revised version in *IEEE Trans. Information Theory* (1988), **IT 34**, 901–909

[2] R. Merkle and M. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory* (1978), vol 5 **IT** 24, 525–530.

[3] D. Naccache and J. Stern. A new public key cryptosystem. *Advances in cryptology: Proceedings of Eurocrypt'97*. LNCS 1233 Ed. W. Fumy Springer-Verlag (1997), 27–36

[4] A. M. Odlyzko. The rise and fall of knapsack cryptosystems. *Cryptology and Computational Number Theory*. Proc. Symp. Appl. Math.(1990), **42** Ed. C. Pomerance, Am. Math. Soc. 75–88.

[5] `http://www.python.org`

[6] J. Saludes. *Fast and secure multiple-option tests*. TUGboat 17 (1996), **3**.

[7] A. Shamir. A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Transactions on Information Theory* (1984), **IT** 30, 699–704.

[8] `http://www.zope.org`