Universidad de Alcalá

Kevin van Liebergen Ávila

Práctica 1
Explotación de la herramienta JCrypTool (ii)

Máster en ciberseguridad

Asignatura: Criptografía aplicada

2020

Mediante el uso de la herramienta *JCrypTool*, se pretende que el alumno

- cifre y descifre e mediante el mecanismo de la libreta de usar y tirar *(one-time pad)*.
- genere una cadena cifrante mediante la función SHA.
- utilice gráficamente el algoritmo de Euclides extendido para obtener inversas modulares.
- utilice gráficamente el algoritmo derivado del teorema chino del resto para resolver congruencias modulares simultáneas.

1 Introducción

Para desarrollar la práctica, podemos seguir aproximadamente este guión. En primer lugar arrancamos la herramienta en las máquinas del laboratorio, después de presentarnos, tecleando dentro de un Terminal:

```
/usr/local/jcryptool/JCrypTool &
```

A continuación pasamos a desarrollar los distintos apartados propuestos.

2 Cifrado y descifrado con *one-time pad*

2.1 Abrir un fichero vacío con Text editor. Escribir algún texto sobre el que vayamos a trabajar.

El texto sobre el que vamos a trabajar se trata de un artículo de ciberseguridad, como se muestra en la imagen 1.

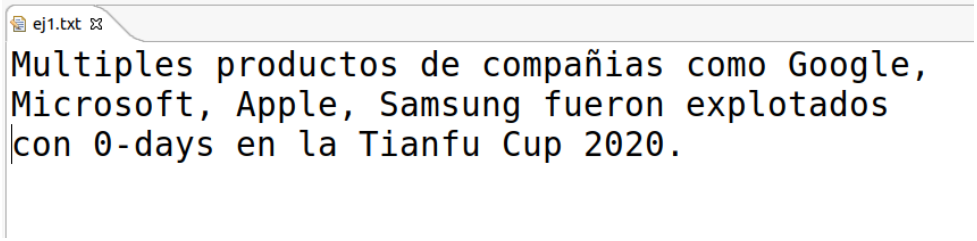

Figure 1: Texto sobre el que trabajaremos

2.2 Seleccionar la vista Default y, dentro de la ventana Crypto Explorer, la pestaña `Algorithms`.

Una vez se ha realizado lo ordenado, la ventana *Crypto Explorer* debe aparecer como en la imagen 2.

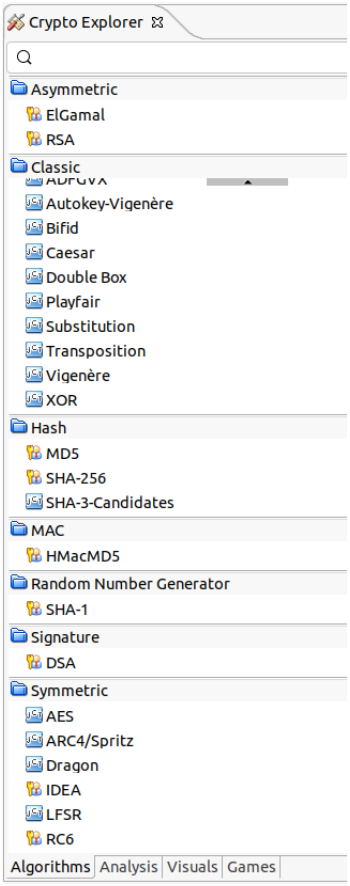

Figure 2: Ventana *Crypto Explorer*

2.3 Seleccionar Classic → XOR

En la ventana XOR como se puede oservar en la imagen 3 tenemos todas las opciones para cifrar y dscifrar nuestro mensaje, otorgando la posibilidad para insertar el tipo de abecedario que queremos, el tipo de clave o si deseamos importar desde un fichero y demás configuración de JCrypTool.

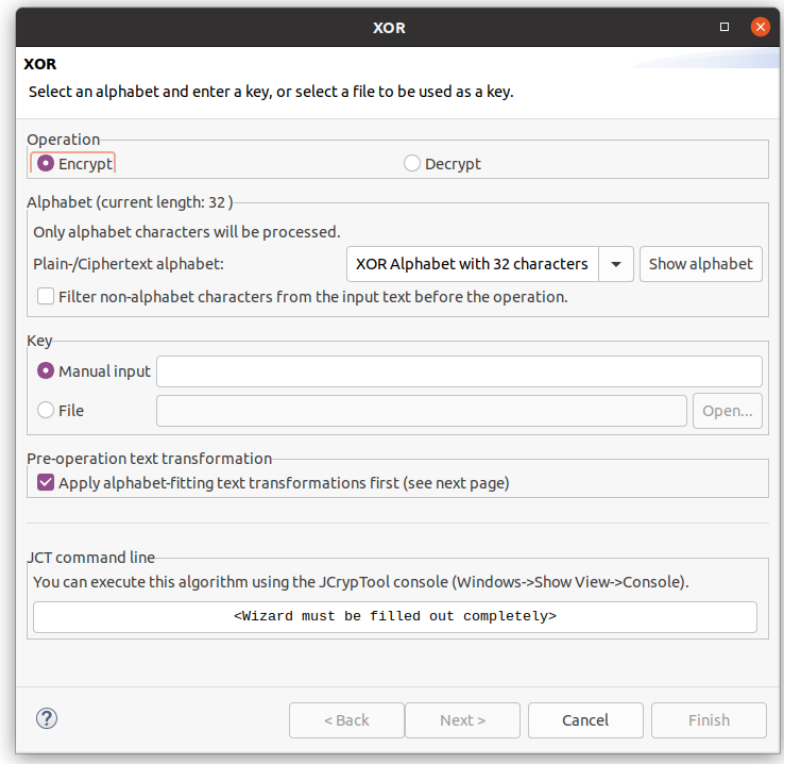

Figure 3: Ventana *XOR*

2.4 Seguir las indicaciones para crear una clave, cifrar y descifrar. Comprobar que los procesos se realizan correctamente y al descifrar recuperamos el texto claro original.

En primer lugar es necesario crear una clave del mismo tamaño que el mensaje original, por lo que se procede a buscar la longitud del texto y aparecen 18 caracteres, la clave para realizar un cifrado XOR es la siguiente (se han omitido las eñes y las tildes):

> EN UN LUGAR DE LA MANCHA, DE CUYO NOMBRE NO QUIERO ACORDARME, NO HA MUCHO TIEMPO QUE VIVIA

Se ha procedido a etiquetar cada letra con un número binario, comenzando con los dígitos 00000 perteneciente a la letra A hasta los dígitos 11111 asociado al número 5 como se muestra en la tabla siguiente.

A	B	C	D	E	F	G	H
00000	00001	00010	00011	00100	00101	00110	00111
I	J	K	L	M	N	O	P
01000	01001	01010	01011	01100	01101	01110	01111
Q	R	S	T	U	V	W	X
10000	10001	10010	10011	10100	10101	10110	10111
Y	Z	0	1	2	3	4	5
11000	11001	11010	11011	11100	11101	11110	11111

Table 1: Diccionario clave-valor con el abecedario y sus correspondiente dígitos

Realizando un cifrado `XOR` con la clave mostrada anteriormente se puede observar en la imagen 4 que se computa correctamente y la salida es la apropiada, asi mismo vamos a proceder a despiezarlo por partes.

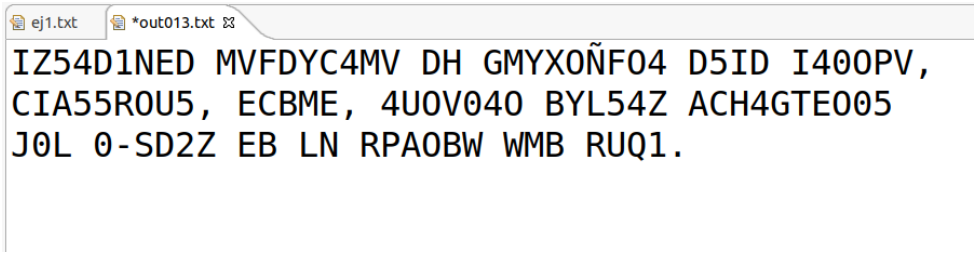

Figure 4: Output de la operación `XOR` con la clave

La primera palabra del texto plano es `Multiples`, en primer lugar el programa transforma todas las palabras minúsculas en mayúsculas por lo que se opera con la palabra `MULTIPLES`, que despiezado en binario es:

M	U	L	T	I	P	L	E	S
01100	10100	01011	10011	01000	01111	01011	00100	10010

Y la clave para los 9 primeros caracteres es:

M	U	L	T	I	P	L	E	S
01100	10100	01011	10011	01000	01111	01011	00100	10010
E	N	U	N	L	U	G	A	R
00100	01101	10100	1101	01011	10100	00110	0000	10001

Por lo que realizando una operación `XOR` de cada dígito del texto plano con el valor de la misma posición de la clave nos genera el mensaje cifrado tal y como se ve en la tabla siguiente, asimismo se puede observar la operación `XOR` es una suma en complemento a 2.

M	U	L	T	I	P	L	E	S
01100	10100	01011	10011	01000	01111	01011	00100	10010
E	N	U	N	L	U	G	A	R
00100	01101	10100	01101	01011	10100	00110	00000	10001
M $\oplus$ E	U $\oplus$ N	L $\oplus$ U	T $\oplus$ N	I $\oplus$ L	P $\oplus$ U	L $\oplus$ G	E $\oplus$ A	S $\oplus$ R
01000	11001	11111	11110	00011	11011	01101	00100	00011
I	Z	5	4	D	1	N	E	D

Como se puede observar, en color azul aparece la salida cifrada siendo este `IZ54D1NED` correspondiendo con la salida de la imagen 4

Cabe recalcar que el cifrado XOR sobre la letra A (00000), obtiene como salida el mismo caracter en texto plano, en el ejmplo de la tabla anterior el penúltimo caracter obtiene como salida la misma entrada.

Asimismo el abecedario que se ha escogido es el inglés, por lo que no se utiliza la *ñ*, y con los números del 1 al 5, por lo que como se muestra en la salida cifrada, no es psible operar con la letra *ñ* del texto plano (cuarta palabra de la slaida cifrada).

2.5 ¿Qué ocurre si la clave es más corta que el mensaje? Compruébese experimentalmente.

En esta sección se va a elegir como la clavela siguiente: `CLAVESECRETA`, cuya salida es la que se observa en la imagen 5.

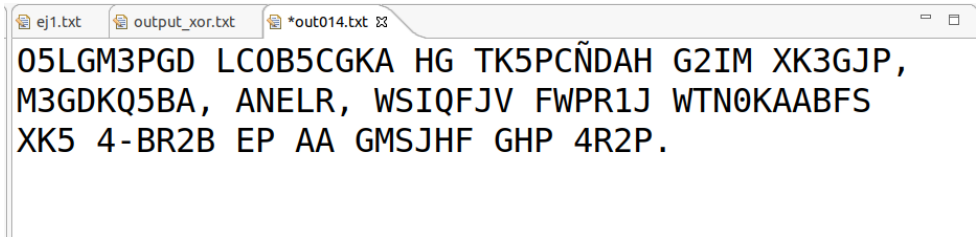

Figure 5: Output de la operación `XOR` con la clave `CLAVESECRETA`

Como la clave es demasiado corta se repite cíclicamente, esto puede suponer un problema debido a que es posible sacar correlaciones para textos grandes, y con ello sacar la clave y descifrar el mensaje. Para cifrados de tipo Vigenere existe una técnica denominada método Kasiski que se basa en la búsqueda de las palabras repetidas de un texto para determinar la clave.

La fortaleza de `One-time pad` reside en que la clave es tan larga como el

texto plano y la utilización de una única vez para cifrar.

3 Generación de una secuencia cifrante con SHA

3.1 Seleccionar la vista Default y, dentro de la ventana Crypto Explorer, la pestaña Algorithms.

Al igual que en el punto anterior, seleccionando la opción que nos propone la ventana debe quedar algo como en la imagen 2

3.2 Seleccionar Random Number Generator → SHA1.

Seleccionando la opción que propone el enunciado, debe aparecer una pestaña como la de la imagen 6.

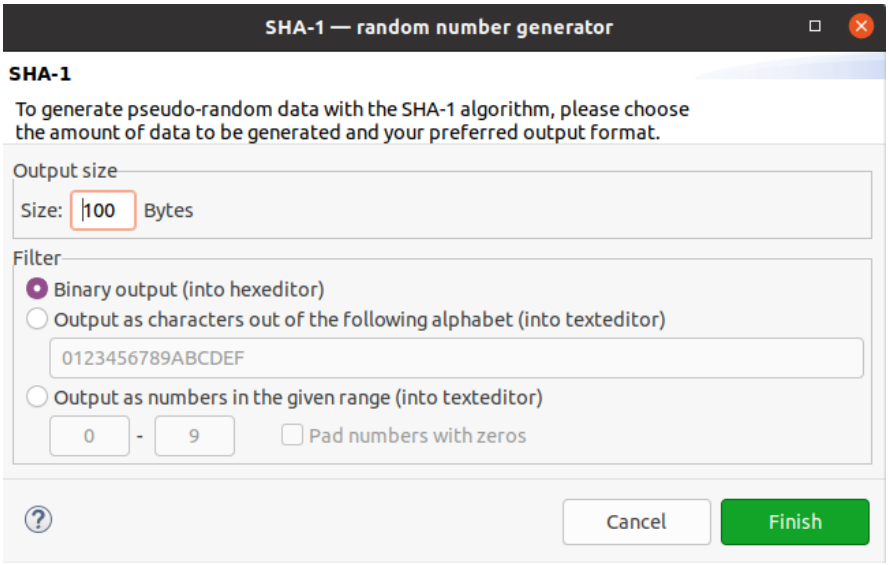

Figure 6: Pestaña SHA1

3.3 Seguir las instrucciones para generar la secuencia.

Generamos una secuencia de 50 bytes mediante este algoritmo de generación de números pseudoaleatorios, cuyo resultado es el que se muestra a continuación.

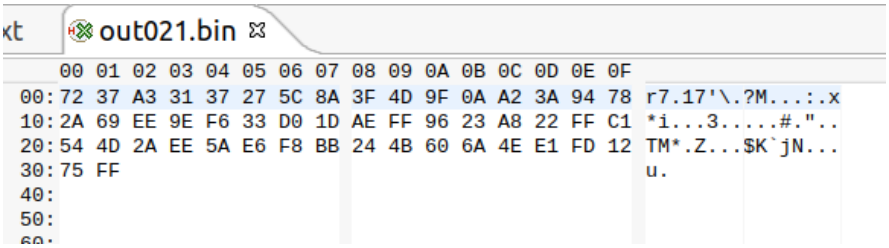

Figure 7: Generación pseudoaleatoria mediante SHA1

3.4 Se pide generar dos secuencias cifrantes, guardándolas en diferentes ficheros y, posteriormente, calcular en cuántos bits difieren tales secuencias. ¿Es coherente el resultado obtenido?

El resultado es coherente debido a que la generación de números pseudoaleatorios debe ser lo más diferente posible, de tal manera que cuando se nos presentan las dos secuencias cifrantes es prácticamente imposible diferenciar a cual pertenece un fichero. En la imagen 8 se muestra el segundo fichero generado mediante el generador de números pseudoaleatorios SHA1.

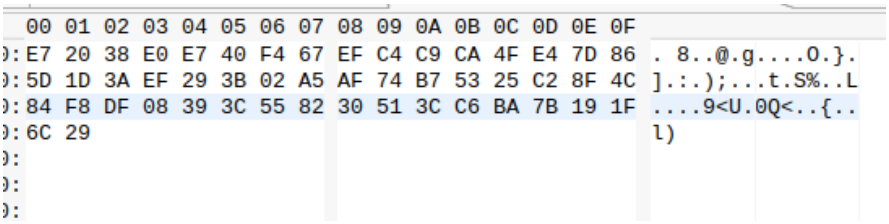

Figure 8: Otra generación pseudoaleatoria mediante SHA1

4 Algoritmo de Euclides extendido

4.1 Seleccionar la vista Default y, dentro de la ventana Crypto Explorer, la pestaña Visuals.

Seleccionando la opción que nos propone la ventana debe quedar algo como en la imagen 9.

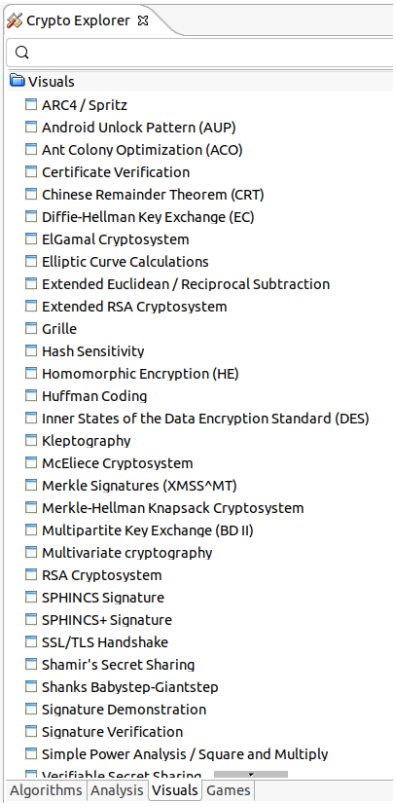

Figure 9: Ventana visuals

4.2 Seleccionar Extended Euclidean/Reciprocal Subtraction.

Seleccionando lo que se propone debe aparecer una pestaña como la de la imagen 10.

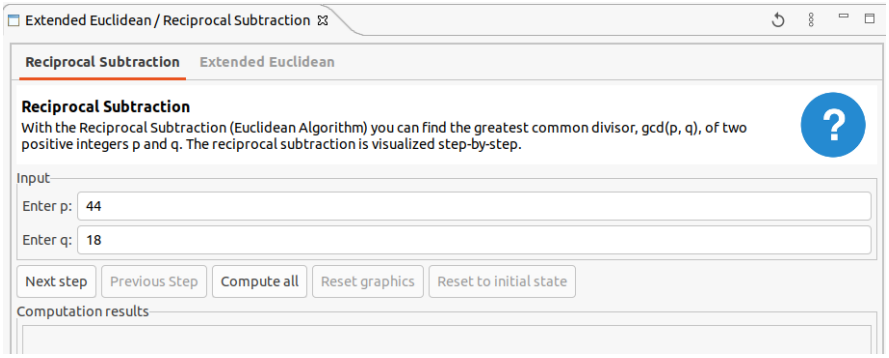

Figure 10: Pestaña extended euclidean

4.3 Ingresar dos valores numéricos y comprobar la ejecución paso a paso para obtener tanto el máximo común divisor como la solución a la ecuación de Bézout.

Se ha procedido a realizar el algoritmo de euclídes de manera ordenada, dividiendo el número mayor entre el menor, con el resto dividimos el divisor entre el resto obtenido y continuamos hasta que el resto sea cero. En este caso el resultado de la primera división, ha dado resto 2, de la segunda 1 y de la tercera división ya ha dado una división exacta.

La identidad de Bezóut establece que si existen dos numeros enteros distintos de 0 (a y b) y con un máximo común divisor d, se puede establecer que :

$$ax + by = d \tag{1}$$

Por lo que basándonos en la ecuación anterior se puede establecer que los coeficientes de la identidad de Bézout son los siguientes:

$$-11 \times 1 + 3 \times 4 = 1 \tag{2}$$

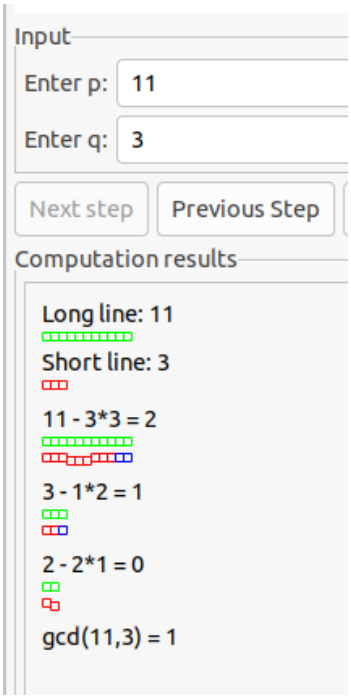

Figure 11: Algoritmo extendido de euclídes extendido

4.4 El algoritmo, ¿necesita el mismo número de pasos en todos los casos? Si la respuesta es no, ¿considera que podría haber valores para los que el algoritmo no terminara?

El algoritmo no siempre utiliza el mismo número de pasos, eso depende de la velocidad de conseguir una división exacta, puede ser en un paso, dos.

Si los números son muy grandes es muy probable que el algoritmo no termine, la fuerza de HTTPS radica en que actualmente no tenemos computación suficiente para factorizar un número compuesto por dos números primos grandes.

Si los números son muy grandes es posible que computacionalmente no termine nunca debido al alto rendimiento que exige las operaciones.

5 Algoritmo derivado del teorema chino del resto

5.1 Seleccionar la vista Default y, dentro de la ventana Crypto Explorer, la pestaña Visuals.

Seleccionando la opción que nos propone la ventana debe quedar algo como en la imagen 9.

5.2 Seleccionar Chinese Remainder Theorem.

Seleccionando la opción elegida nos aparece un desplegable como la de la figura 12.

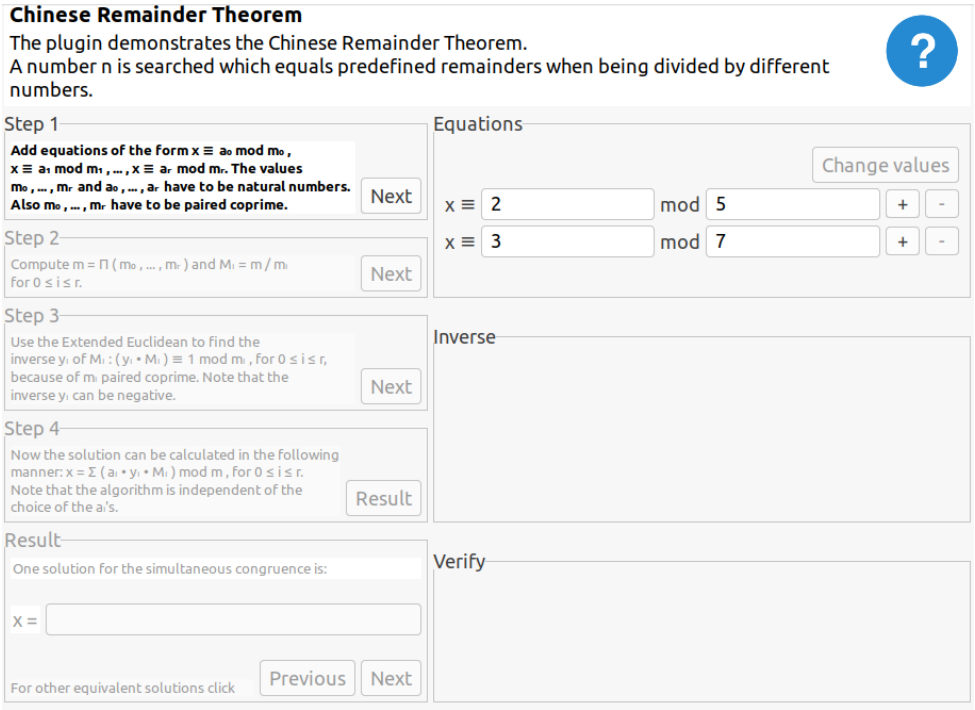

Figure 12: Pestaña teorema chino del resto

5.3 Ingresar tantas ecuaciones como queramos, con la precaución de que todos los módulos sean primos entre sí.

Se han elegido los números que aparecen en la imagen 13, estos son 5, 7 y 11, y como requisito que es, son coprimos entre sí.

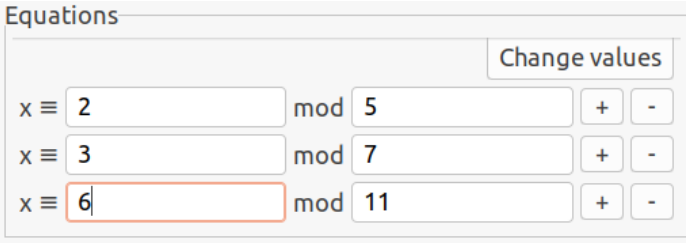

Figure 13: Valores elegidos para el teorema chino del resto

5.4 Compruebe que la solución es única módulo el producto de los módulos

Según el teorema Chino del resto, sean $m_0, m_1, ..., m_k$ enteros y coprimos entre sí (máximo común divisor entre sí es 1), y $a_0, a_1, ..., a_k$ siendo enteros

cualesquiera, se cumple que

$$\begin{aligned} x &\equiv a_0 (mod\ m_0) \\ x &\equiv a_1 (mod\ m_1) \\ &... \\ x &\equiv a_k (mod\ m_k) \end{aligned} \tag{3}$$

De tal manera que posee solución única modulo $m = m_0 \times m_1 \times ... \times m_k$, quiere decir que todas las soluciones x son congruentres módulo el producto (tienen el mismo resto al dividirlos por m).

Desglosando el teorema chino del resto con los números de la imagen 13, en primer lugar se halla m como el producto de $m_0 = 5,\ m_1 = 7,\ m_2 = 11 = 385$.

Seguidamente se procede a realizar el cálculo de M_0, M_1, M_2 como

$$M_i = m/m_i\ para\ 0 \leq i \leq 2 \tag{4}$$

Es decir:

$$\begin{aligned} M_0 &= 385/5 = 77 \\ M_1 &= 385/7 = 55 \\ M_2 &= 385/11 = 35 \end{aligned} \tag{5}$$

Ahora utilizando el algoritmo extendido de Euclídes se halla la inversa y de M como:

$$\begin{aligned} y_0 \cdot M_0 \equiv 1\ mod\ m_0 &\Longrightarrow y_0 \cdot 77 \equiv 1\ mod\ 5 \Longrightarrow y_0 = -2 \\ y_1 \cdot M_1 \equiv 1\ mod\ m_1 &\Longrightarrow y_1 \cdot 55 \equiv 1\ mod\ 7 \Longrightarrow y_1 = -1 \\ y_2 \cdot M_2 \equiv 1\ mod\ m_2 &\Longrightarrow y_2 \cdot 35 \equiv 1\ mod\ 11 \Longrightarrow y_2 = -5 \end{aligned} \tag{6}$$

Y por último y para hallar X, se halla el sumatorio de la ecuación de debajo y como resultado final nos aparece 17 como resultado, y como aparece en la figura 14

$$\begin{aligned} X &= \sum (a_i \cdot y_i \cdot M_i)\ mod\ m,\ for\ 0 \leq i \leq r \\ X &= (a_0 \cdot y_0 \cdot M_0) + (a_1 \cdot y_1 \cdot M_1) + (a_2 \cdot y_2 \cdot M_2)\ mod\ m \\ X &= (2 \cdot -2 \cdot 77) + (3 \cdot -1 \cdot 55) + (6 \cdot -5 \cdot 35)\ mod\ 385 \\ X &= 17 \end{aligned} \tag{7}$$

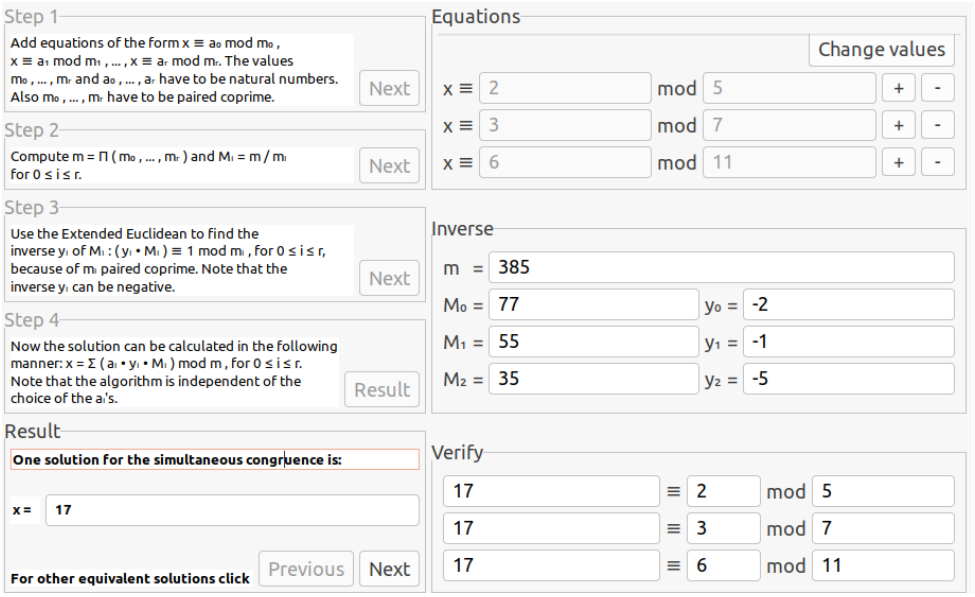

Figure 14: Teorema desglosado

Como se ha comentado anteriormente el producto de los módulos $5, 7, 11$ es 385, y su solución, $X = 17, 402, 787$, etc (figuras 14 15 16) cumplen que todos ellos son el mismo número en módulo 385.

$$\begin{aligned} 17 &\equiv 17 \; mod \; 385 \\ 402 &\equiv 17 \; mod \; 385 \\ 787 &\equiv 17 \; mod \; 385 \end{aligned} \tag{8}$$

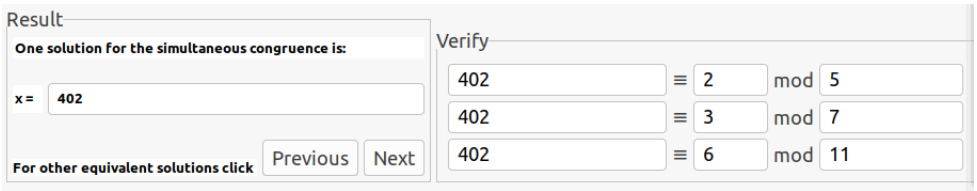

Figure 15: Uno de los resultados de X

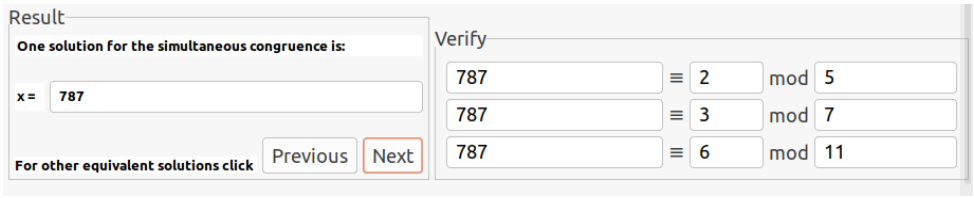

Figure 16: Uno de los resultados de X