

Apéndice A: Imágenes de las simulaciones en la interfase gráfica de Matlab.

En este apéndice se muestran imágenes obtenidas de la interfase gráfica creada en Matlab al realizar las simulaciones del capítulo 6.

Simulación #1: En la Figura A.1 se muestra la ejecución del algoritmo WEP. Se ingresó la cadena “computadora” y una longitud de llave de 45 bytes. Al hacer clic sobre el botón “Encriptar” toma estos datos para generar el cifrado. Para realizar el proceso de decriptación se hace clic en el botón decriptar y se recupera el texto original.

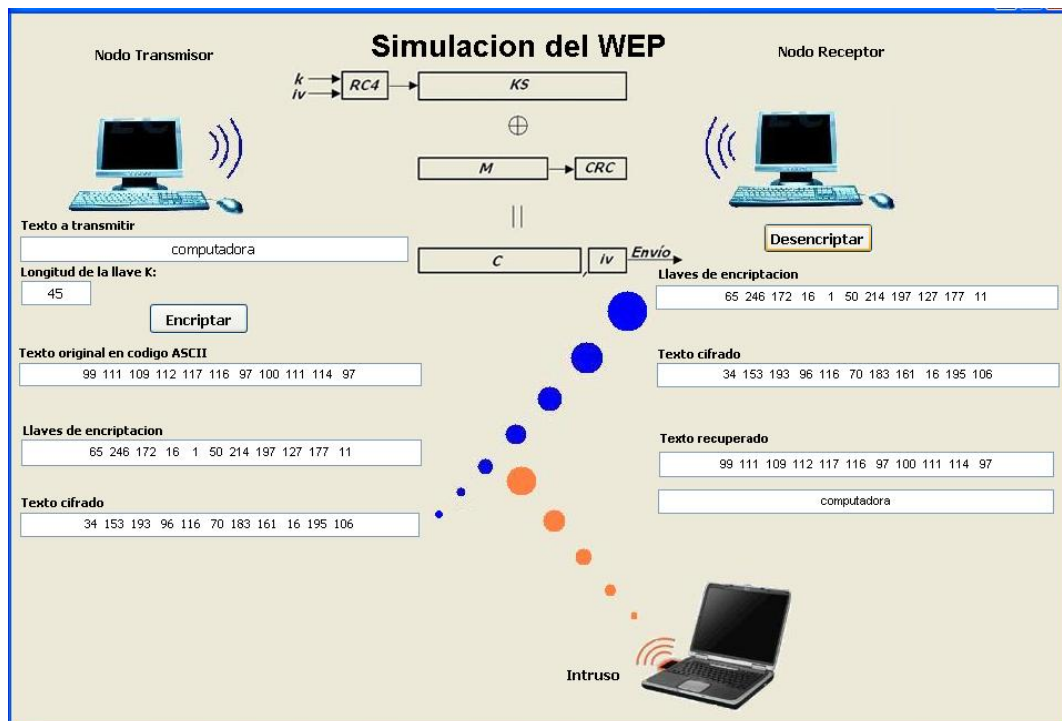


Figura A.1. Simulación # 1 en la interfase gráfica de Matlab.

Simulación # 2: En la Figura A.2 se muestra la ejecución del algoritmo FCICT con valor $m = 1$ y utilizando la técnica de reemplazo. Se ingresó la cadena “computadora”, una longitud de llave de 45 bytes, y una longitud de 160 bytes para el vector *fakekey*. Al hacer clic sobre el botón “Transmitir” el algoritmo recibe los datos y encripta los datos mostrándolos en la parte correspondiente al transmisor. En la parte correspondiente al receptor, el algoritmo decripta la información y recupera el texto original. En la parte que corresponde al intruso, se muestra el texto que recuperaría un intruso en caso de interceptar el cifrado y el vector de inicialización. Como se puede ver este texto no coincide con el recuperado por el receptor.

Tecnica FCICT con tecnica de reemplazo

Longitud de la llave K
Longitud de fakekey
Valor de m

Datos

Transmisor

Texto original en ASCII

Llaves de encriptacion

Cifrado RC4

Cifrado aplicando FCICT

Receptor

Llaves de encriptacion

Cifrado aplicando FCICT

Cifrado RC4

Texto recuperado

Intruso

Cifrado aplicando FCICT

Llaves de encriptacion

Texto recuperado por el intruso

```

    graph TD
      subgraph Transmisor
        K[k] --> RC4
        IV[iv] --> RC4
        RC4 --> KS[KS]
        M[M] -- XOR --> C[C]
        M --> CRC[CRC]
      end
      subgraph Receptor
        C --> RC4
        CRC --> RC4
        RC4 --> M_rec[M]
      end
      subgraph Intruso
        C --> RC4
        iv_rec[iv] --> RC4
        RC4 --> M_intr[M_intr]
      end
      C -- Envío --> Intruso
  
```

Figura A.2. Simulación # 2 en la interfase gráfica de Matlab.

Simulación # 3: En la Figura A.3 se muestra la ejecución del algoritmo FCICT con $m = 4$ utilizando la técnica de reemplazo. Se ingresaron los mismos datos que en el ejemplo anterior. De la misma forma se muestran los datos transmitidos, y recuperados por el recetor y por el intruso. Debido al valor diferente de m el algoritmo insertó menos caracteres falsos que en el ejemplo anterior.

Tecnica FCICT con tecnica de reemplazo

Longitud de la llave K: 45 Longitud de fakekey: 160 Valor de m: 4 **Transmitir**

Datos

Transmisor

Texto original en ASCII

Llaves de encriptacion

Cifrado RC4

Cifrado aplicando FCICT

Receptor

Llaves de encriptacion

Cifrado aplicando FCICT

Cifrado RC4

Texto recuperado

Intruso

Cifrado aplicando FCICT

Llaves de encriptacion

Texto recuperado por el intruso

Diagrama de flujo:
 $k, iv \rightarrow RC4 \rightarrow KS$
 $M \oplus KS \rightarrow CRC$
 Inserción de caracteres falsos $\rightarrow R$
 $C, iv \rightarrow Envío$

Figura A.3. Simulación # 3 en la interfase gráfica de Matlab.

Simulación # 4: En la Figura A.4 se muestra la ejecución del algoritmo FCICT con $m = 8$ utilizando la técnica de reemplazo. Se ingresaron los mismos datos que en el ejemplo anterior. De la misma forma se muestran los datos transmitidos, y recuperados por el recetor y por el intruso. Debido al valor de m el algoritmo no insertó ningún carácter falso en el cifrado RC4, por lo que el intruso recupera el mismo texto que el receptor.

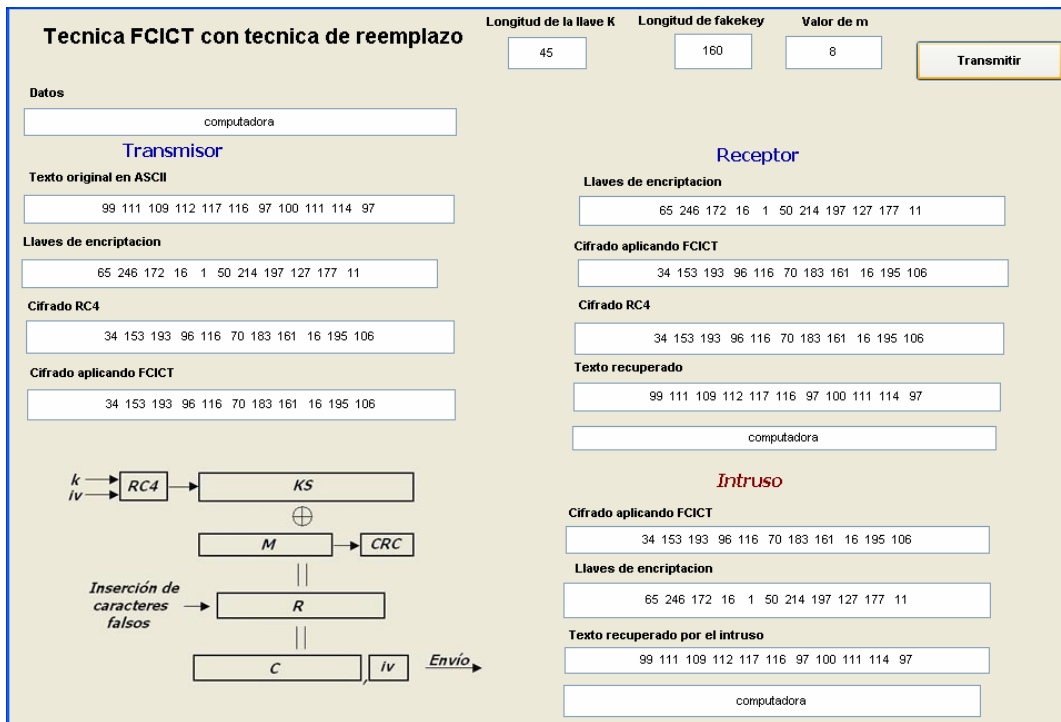


Figura A.4. Simulación # 4 en la interfase gráfica de Matlab.

Simulación # 5: En la Figura A.5 se muestra la ejecución del algoritmo FCICT con $m = 1$ y utilizando el algoritmo de compresión de Huffman . Se ingresaron los mismos datos que en el ejemplo anterior. De la misma forma se muestran los datos transmitidos, y recuperados por el recetor y por el intruso. En este caso se observa como el cifrado FCICT es mayor en longitud al cifrado RC4 debido a la inserción de caracteres falsos. El receptor descarta los caracteres falsos para obtener el cifrado RC4 y decriptar la información. El intruso interceptaría el texto cifrado y tendría que realizar cálculos para determinar y desechar los caracteres falsos, por lo que no recupera ningún texto.

Técnica FCICT con algoritmo de compresión de Huffman

Longitud de la llave K

Longitud de fakekey

Valor de m

Datos

Transmisor

Texto original en ASCII

Llaves de encriptacion

Cifrado RC4

Cifrado aplicando FCICT

Receptor

Llaves de encriptacion

Cifrado aplicando FCICT

Cifrado RC4

Texto recuperado

Intruso

Cifrado aplicando FCICT

Llaves de encriptacion

Texto recuperado por el intruso:

?

```

graph TD
    k --> RC4
    iv --> RC4
    RC4 --> KS
    M -- XOR --> KS
    KS --> CRC
    M -- "Inserción de caracteres falsos" --> R
    R -- "||" --> C
    CRC -- "||" --> C
    C -- "Envío" --> Out
  
```

Figura A.5. Simulación # 5 en la interfase gráfica de Matlab.