Apéndice C: Código en Matlab de la simulación de la FCICT con técnica de reemplazo.

En este apéndice se muestra el código implementado en Matlab para simular la técnica FCICT con técnica de reemplazo en la interfase visual.

```
% este código se crea automáticamente para inicializar los valores de la interfase
% gráfica

function varargout = fakefake(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @fakefake_OpeningFcn, ...
                   'gui_OutputFcn',  @fakefake_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% finaliza el código de inicialización


% el código siguiente llama a las imágenes JPG que se muestran en la interfase
set(hObject, 'Units', 'pixels');
handles.FCICT_GUI = imread(['FCICT_GUI.jpg']); % Read the image file banner.jpg
info = imfinfo(['FCICT_GUI.jpg']); % Determine the size of the image file
position = get(hObject, 'Position');

axes(handles.axes1);
image(handles.FCICT_GUI)
set(handles.axes1, ...
   'Visible', 'off', ...
   'Units', 'pixels', ...
   'Position', [50 30 info.Width info.Height]);


% funciónes que crea el cuadro de texto que recibe del usuario la longitud de del
% vector K
function long_K_Callback(hObject, eventdata, handles)

handles.long_K = str2double(get(hObject,'String')); %guarda el valor del tamaño del vector K en la
variable long_K

function long_K_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
```

```
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%funciones que crean el cuadro de texto donde se recibe la cadena de
% texto a encriptar

function datos_Callback(hObject, eventdata, handles)

handles.plain_text = get(hObject,'String') % guarda la cadena ingresada por
% el usuario en la variable plain_text

function datos_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% las siguientes funciones crean el cuadro de texto donde se mostrará
% la cadena de texto ingresada por el usuario en texto ASCII
function textorigASCII_Callback(hObject, eventdata, handles)


function textorigASCII_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% las siguientes funciones crean el cuadro de texto donde se mostrarán las
% llaves de encriptación generadas
function llaves_enc_Callback(hObject, eventdata, handles)

function llaves_enc_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% estas funciones crean el cuadro de texto donde se mostrará el texto cifrado RC4
function cifrado_normal_Callback(hObject, eventdata, handles)

function cifrado_normal_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% estas funciones crean el cuadro de texto que mostrará el cifrado FCICT
function cifrado_fake_Callback(hObject, eventdata, handles)

function cifrado_fake_CreateFcn(hObject, eventdata, handles)
if ispc
```

```
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% estas funciones crean el cuadro de texto que mostrará las llaves de decriptación
function llaves_decript_Callback(hObject, eventdata, handles)

function llaves_decript_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% estas funciones crean el cuadro de texto que mostrará el cifrado FCICT
% en el receptor
function ciffcict_Callback(hObject, eventdata, handles)

function ciffcict_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% estas funciones crean el cuadro de texto que mostrará el cifrado RC4
% en el receptor
function cifradonormal_dec_Callback(hObject, eventdata, handles)

function cifradonormal_dec_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



% estas funciones crean el cuadro de texto que mostrará el texto ASCII
% recuperado en el receptor
function textrecuperado_ASCII_Callback(hObject, eventdata, handles)

function textrecuperado_ASCII_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% estas funciones crean el cuadro de texto que mostrará la cadena texto
% recuperada en el receptor
function Texto_recuperado_Callback(hObject, eventdata, handles)

function Texto_recuperado_CreateFcn(hObject, eventdata, handles)
if ispc
```

```
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% estas funciones crean el cuadro de texto que mostrará el texto FCICT en
% la parte del intruso
function cifradoFCICT_int_Callback(hObject, eventdata, handles)

function cifradoFCICT_int_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% estas funciones crean el cuadro de texto que mostrará las llaves de encriptación obtenidas por el
intruso
function llaves_intruso_Callback(hObject, eventdata, handles)

function llaves_intruso_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% estas funciones crean el cuadro de texto que mostrará el texto ASCII recuperado por el intruso
function rec_ascii_intruso_Callback(hObject, eventdata, handles)

function rec_ascii_intruso_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end




% esta función contiene el algoritmo FCICT, que se ejecuta al hacer clic en el
% boton Transmitir
function pushbutton1_Callback(hObject, eventdata, handles)

handles.plainASCII = double(handles.plain_text)  %convierte la cadena ingresada
% por el usuario en caracteres ASCII


%inicialización de la llave S, RC4

% long_K es el número de bytes con el que se forma el vector K
% mismo que se repetirá tantas veces como sea necesario
% para llenar el vector temporal T

for i = 0:(handles.long_K-1)
   K(i+1) = i;            %generación del vector K
end
```

```
S = [0:255];

for j=0:255                % se inicializa S con valores de 0 a 255
   y = mod(j,handles.long_K);    % se repite el vector K tantas veces como sea
   T(j+1) = K(y+1);          % necesario para genera el vector T
end

% se inicializa el vector S

k = 0;
for m=0:255
   z = (k + S(m+1) + T(m+1));
   k = mod(z,256);
   temp = S(m+1);
   S(m+1) = S(k+1);
   S(k+1) = temp;
end

% estas instrucciones obtienen la longitud del texto ingresado
size_plain = size(handles.plainASCII);
handles.size_plain = size_plain(2);


% el siguiente algoritmo genera la secuencia de llaves de encriptación

i=0;
k=0;

format compact

for j = 1:handles.size_plain

   q = i+1;
   i = mod(q,256);
   p = k + S(q);
   k = mod(p,256);
   temp = S(i+1);
   S(i+1) = S(k+1);
   S(k+1) = temp;
   r = (S(i+1) + S(k+1));
   t = mod(r,256);
   handles.Keys(j) = S(t+1);
   cipher(j) = bitxor(handles.Keys(j), handles.plainASCII(j)); %esta sentencia
%realiza la operación XOR entre las llaves generadas y el texto a encriptar


end


% las siguientes instrucciones envían las cadenas de texto ASCII y las llaves
% de encriptación, a los cuadros de texto del transmisor, receptor e intruso
set(handles.textorigASCII,'String',num2str(handles.plainASCII))
set(handles.llaves_enc,'String',num2str(handles.Keys))
set(handles.llaves_decript,'String',num2str(handles.Keys))
set(handles.llaves_intruso,'String',num2str(handles.Keys))


%FCICT encriptacion
```

```
%inicialización de la llave S, RC4

% long_K es el número de bytes con el que se forma el vector K
% mismo que se repetirá tantas veces como sea necesario
% para llenar el vector temporal T

for i = 0:(handles.long_K_fakekey-1)
   K(i+1) = i;            %generación del vector K
end

fakekey = [0:255];

for j=0:255                % se inicializa S con valores de 0 a 255
   y = mod(j,handles.long_K_fakekey);    % se repite el vector K tantas veces como sea necesario
   T(j+1) = K(y+1);          % necesario para genera el vector T
end


% generación de la llave fakekey que servirá para determinar la posición en la
% que deben ser insertados los caracteres falsos
k = 0;
for m=0:255
   z = (k + fakekey(m+1) + T(m+1));
   k = mod(z,256);
   temp = fakekey(m+1);
   fakekey(m+1) = fakekey(k+1);
   fakekey(k+1) = temp;
end

% este ciclo inicializa el cifrado FCICT con los valores del cifrado RC4
for j = 1:handles.size_plain
   FAKEcipher(j) = cipher(j);
end

j=0;

% el siguiente ciclo inserta los caracteres falsos en el cifrado RC4
for j = 2:handles.size_plain-1
   w = mod(j,256);
   fakecheck = mod(cipher(j) + cipher(j-1) + fakekey(w),2^handles.m_prob);
   if (fakecheck == 0)
      xor_uno = bitxor(cipher(j), cipher(j-1));
      fakechar = bitxor(xor_uno, fakekey(w));
      q = fakechar + cipher(j);
      FAKEcipher(j) = mod(q,256);
   end
end




% el siguiente ciclo convierte el cifrado FCICT a cifrado RC4, detectando los caracteres falsos y
reemplazandolos

for j = 2:handles.size_plain-1
    w = mod(j,256);
    fakecheck = mod(cipher_dec(j) + cipher_dec(j-1) + fakekey(w),2^handles.m_prob);
    if (fakecheck == 0);
       xor_uno = bitxor(cipher_dec(j), cipher_dec(j-1));
       fakechar = bitxor(xor_uno, fakekey(w));
```

```
        q = 2^8 + FAKEcipher_dec(j) - fakechar;
        FAKEcipher_dec(j) = mod(q,256);
    end
end


% el siguiente conjunto de instrucciones envía las cadenas de cifrado
% FCICT y cifrado RC4 a los cuadros de texto correspondientes en la interfase

set(handles.cifrado_normal,'String',num2str(cipher))
set(handles.cifrado_fake,'String',num2str(FAKEcipher))

set(handles.ciffcict,'String',num2str(FAKEcipher))
set(handles.cifradonormal_dec,'String',num2str(cipher_dec))

set(handles.cifradoFCICT_int,'String',num2str(FAKEcipher))


% el siguiente ciclo realiza la operación XOR entre las llaves y el texto
% cifrado RC4
for j = 1:handles.size_plain

   recuperado_ascii_rx(j) = bitxor(handles.Keys(j), cipher(j));

end

text_recuperado_rx = char(recuperado_ascii_rx);

%  el texto recuperado por el receptor se envía al cuadro de texto
% del receptor en la interfase
set(handles.textrecuperado_ASCII,'String',num2str(recuperado_ascii_rx))
set(handles.Texto_recuperado,'String',text_recuperado_rx)



% el siguiente ciclo realiza la operación XOR entre las llaves y el cifrado RC4
% en la parte del intruso
for j = 1:handles.size_plain

   recuperado_ascii_intruso(j) = bitxor(handles.Keys(j), FAKEcipher(j));

end

text_recuperado_intruso = char(recuperado_ascii_intruso);

%  el texto recuperado por el receptor se envía al cuadro de texto
% del intruso en la interfase

set(handles.rec_ascii_intruso,'String',num2str(recuperado_ascii_intruso))
set(handles.rec_text_intruso,'String',text_recuperado_intruso)


% estas funciones crean el cuadro de texto en la interfase que recibe la
% longitud de la llave fakekey

function long_fakekey_Callback(hObject, eventdata, handles)

handles.long_K_fakekey = str2double(get(hObject,'String'));

function long_fakekey_CreateFcn(hObject, eventdata, handles)
```

```
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
% estas funciones crean el cuadro de texto en la interfase que muestran
% el texto recuperado por el intruso
function rec_text_intruso_Callback(hObject, eventdata, handles)

function rec_text_intruso_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
%las siguientes funciones crean el cuadro de texto en la interfase donde
%se recibe el valor de m dado por el usuario
function valordem_Callback(hObject, eventdata, handles)

handles.m_prob = str2double(get(hObject,'String'))
guidata(hObject, handles);

function valordem_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```