UNA VARIANTE DEL SISTEMA DE ROTORES

ERNESTO AZORIN y NELSON BALOIAN

Departamento de Ciencias de la Computación

Universidad de Chile

Casilla 2777

Santiago, Chile.

Resumen En este trabajo se describe una variante de un sistema criptográfico basado en el principio de los rotores. A la concepción original de este sistema se le han introducido modificaciones que hacen dependiente de la clave a la función de transformación; contempla, además, el uso de variables pseudoaleatorias en el algoritmo de rotación de los rotores. Esto vuelve inútiles los métodos tradicionales de ataque para rotores clásicos.

1- Introducción

Los rotores son un sistema criptográfico de sustitución polialfabética. Esto significa que cada letra x del texto originial se sustituye por otra c aplicándole una transformación $f(x) = c$. Además, a cada letra del texto original se le aplica una función distinta; es así como una misma letra se traduce, en general, de manera diferente en cada aparición.

En su concepción original, una máquina de rotores consiste en una batería de t rotores o ruedas. El perímetro de cada rotor R_i tiene$n$ contactos eléctricos en cada una de sus caras, uno por cada letra del alfabeto ($n = 26$ en el caso del alfabeto inglés). Cada contacto de la cara anterior está conectado por un cable a otro de la cara posterior haciendo corresponder uno a uno los contactos de ambas caras. Queda así definida una función de transformación f_i para cada rotor R_i . Cada rotor puede girar para colocarse en una cualquiera de n posiciones distintas y alterar la función f_i . Formalmente se puede escribir que cuando el rotor R_i está en la posición l_i la función de transformación resultante se redefine :

$$F_i = (f_i (x - l_i) \bmod n + l_i) \bmod n \qquad (1.1)$$

Los t rotores van colocados de modo que los contactos de la cara posterior del i-ésimo rotor estén en contacto con los de la cara anterior del rotor en la posición i+1. Para transformar una letra (por ejemplo la j-ésima del alfabeto) se envía una señal eléctrica por el contacto que le corresponde a su posición en la cara anterior del primer rotor. La señal viaja a través de los rotores en sucesión y emerge por alguno de los contactos de la cara posterior del último, lo cual da la letra del texto cifrado correspondiente. A la letra del texto original se le aplican, entonces, t transformaciones sucesivas :

$$c = F(x) = F_1(F_2 (\ldots F_{t-1}(F_t (x)))) \qquad (1.2)$$

Después de haber cifrado una letra, uno o más rotores cambian de posición originándose una nueva función de transformación F.

Un sistema de rotores famoso es el **Enigma** [Denn83] usado por los alemanes en la segunda Guerra Mundial. En el **Enigma**, después de toda transformación de una letra sigue un avance de una posición del primer rotor . Una vez que completa una rotación entera, el segundo avanza una posición. Cuando el segundo completa una vuelta, el siguiente avanza un lugar y así sucesivamente con los otros. La clave que mantiene el secreto del texto la constituye la posición inicial de los

rotores y las conexiones en cada uno de ellos. Formalmente, la clave de este sistema está constituida por las funciones de transformación de cada rotor y la posición inicial de ellos.

Para descifrar un texto hay que enviar los caracteres cifrados a través de los rotores pero ahora en sentido inverso, es decir, la señal que corresponde al carácter cifrado entra por un contacto de la cara posterior y sale por el de la cara anterior del rotor al cual está asociado. Esto corresponde a aplicarle la función inversa f_i^{-1} del rotor R_i. Si llamamos c_i al carácter cifrado por la función F_i,

$$c_i = F_i(x) \tag{1.3}$$

y F_i^{-1} se escribe como:

$$F_i^{-1}(c_i) = (f^{i-1}(c_i - l_i) \bmod n + l_i) \bmod n = x \tag{1.4}$$

Para obtener el carácter original es necesario que los rotores estén en la misma posición l_i en la que aquél fue cifrado. Si los caracteres se descifran en el mismo orden en que fueron cifrados basta generar las posiciones de la misma manera que en el proceso de cifrado.

En este trabajo se enfrenta el diseño de un sistema de rotores para proveer una herramienta que permita cifrar textos y programas con las siguientes características :

- Las funciones de transformación f_i de los rotores y la posición inicial de cada uno ellos se derivarán de la clave con que se cifre el texto.

- Las posiciones siguientes de los rotores se darán de acuerdo a un algoritmo de generación de variables pseudoaleatorias, que dependerá también de la clave.

Es bien conocida la falta de aleatoriedad propia de los generadores congruenciales lineales **[Frei84]**. Por otro lado, el interés por construir sistemas criptográficos ideales ha suscitado la búsqueda de secuencias pseudoaleatorias criptográficamente seguras ([Blum84], **[Blum86]**, **[Gold86]**, **[Sham83]**). El uso de la clave para determinar el movimiento de los rotores provee una forma sencilla de modificar sucesivamente los ciclos propios de generadores congruenciales lineales, mejorando así las repeticiones "aleatorias" de dígitos. En este estudio se recurre a transformaciones de la clave original, análogas a las utilizadas por el DES [NBS77] , para calcular las claves parciales asociadas a los rotores.

2- La función de transformación

Llamemos A al alfabeto del texto que se quiere cifrar y sea n su cardinalidad. Sea x un número entre 1 y n que representa a la **x-ésima** letra del alfabeto. La función de transformación f_i de cada rotor debe ser tal que transforme x en un número $c = f_i(x)$ tal que c quede entre 1 y n para así poder asociarlo a otra letra del alfabeto. Además se debe cumplir que para un x' distinto a x, $f(x')$ debe ser distinto a $f_i(x)$ para todo x' ya que así se simulan las conexiones que asocian a cada contacto de la cara anterior de un rotor con otro único de la cara posterior. Para lograr esto se propone definir la función f_i como :

$$f_i(x) = (a_i * x) \bmod n + 1 \qquad (2.1)$$

en que : a_i es un número entero entre 1 y n que se deriva de la clave.

n es un número primo igual al número de símbolos que tiene el texto.

Es necesario que n sea primo ya que así cualquier a_i definido de la manera anterior cumplirá la condición de que el máximo común divisor entre a_i y n es 1. De aquí se desprende que para cada $x \in \{1,2,...,n\}$, $(a_i * x) \bmod n$ es un número distinto entre 0 y $n - 1$, lo que permite la asociación biunívoca x <--> $(a_i * x) \bmod n$.

Para abarcar la mayor cantidad posible de caracteres del código ASCII podemos utilizar un subconjunto de ellos de cardinalidad igual al primer número primo menor que 256, es decir 251.

Como en general, para dos rotores distintos R_i y R_j, se tiene que $a_i \neq a_j$, cada rotor aplicará una transformación distinta.

En este sistema se define la función de transformación **(2.1)** de cada rotor de modo que en el recorrido de ella nunca se encuentre el número cero (como consecuencia, tampoco hay un símbolo del alfabeto que sea representado por este número), ya que cuando apareciese, este valor podría propagarse hasta el final incluyendo información del texto original en el cifrado. Con la función de transformación así definida los valores del recorrido están en {1,...,n}.

3- El algoritmo de rotaciones

En un sistema de rotores $\{R_1, R_2, ..., R_t\}$ la cantidad de posiciones distintas que puede tomar el conjunto de ellos es n^t donde:

n es el número de posiciones posibles de cada rotor, es decir, el número de símbolos del alfabeto con que se trabaja (en este caso corresponde a 251).

t es el número de rotores en el sistema.

Como las funciones de transformación de cada rotor se mantienen fijas durante la codificación del texto, el número de sustituciones distintas que podemos generar con el sistema es precisamente el número de posiciones distintas del conjunto de rotores, es decir n^t .

Para dar mayor seguridad al cifrado del texto es deseable tener un número de sustituciones tan largo como el texto a codificar a fin de no repetir la misma sucesión de sustituciones. Podemos lograrlo haciendo lo siguiente:

- Se define un **estado de posiciones** como el conjunto de las posiciones de los rotores en un instante dado.
- Se enumeran todas los estados de posiciones posibles desde la posición 0 a la $n^t - 1$: cada estado de posiciones se representa por un número P de t dígitos en base n, es decir

$$P = p_1 p_2 \ \ldots \ p_{t-1} p_t \tag{3.1}$$

en el que cada p_i es un número entre 0 y $n - 1$ que representa la posición de cada rotor R_i en ese momento.

El valor en base diez del estado inicial de posiciones, P_0 , se calcula como:

$$P_0 = H * N_0 \ mod \ (n^t) \tag{3.2}$$

donde H, N_0 son números entre 0 y $n^t - 1$ derivados de la clave.

Los siguientes estados de posiciones , P_{i+1} , se calculan como :

$$P_{i+1} = H * N_{i+1} \ mod \ (n^t) \quad ; \quad N_{i+1} = N_i + 1 \tag{3.3}$$

Para encontrar las posiciones de cada rotor correspondiente al estado de posiciones P_i basta convertir este número a uno de base n y cada dígito dará la posición que le corresponde a cada rotor.

Como la divulgación o descubrimiento del período facilita el rompimiento del código se sugiere una modificación sencilla que esconde dicho período por medio de desfases sucesivos generados,

también, a partir de la clave. Para esto, cada D transformaciones se efectúa un salto de E posiciones en el valor de N_i :

$$N_i = (N_{i-1} + E) \bmod (n^t) \qquad (3.4)$$

donde D y E son números derivados de la clave de modo que D + E no sea divisor de n^t.

4- La clave

La clave K se ingresa como una palabra de $2k$ caracteres. Cada carácter se traduce a un número de tres dígitos correspondiente al valor en base diez de su representación binaria en ASCII, que da origen a una secuencia de $6k$ dígitos. Partiendo esta secuencia se construyen dos números, ci y cd, de $3k$ dígitos cada uno.

Obtención de los a_i :

Para derivar un a_i de los dos números ci y cd generados, seguimos la filosofía del sistema DES en el cual se efectúan varias permutaciones y desplazamientos circulares (indicados por RT en la figura 1) de los dígitos de ámbos números según una tabla predefinida. Luego, se escogen trés digitos de cada número ci, cd y, entremezclándolos, se construye un nuevo número m . Con éste se calcula

$$a_i = m \bmod (n-1) + 1 \qquad \text{(indicados por S en la figura 1)}$$

para obtener así un número entre 1 y $n-1$, que es la condición que deben cumplir los a_i.. Este proceso se vuelve a repetir las veces que sea necesario para obtener todos los a_i, $i = 1...t$ (uno para cada rotor).

Cada proceso de obtención de un a_i modifica los números generados a partir de la clave con lo cual a_{i+1} se obtendrá a partir de ci y cd distintos. Se asegura así que un mismo carácter se traduce, en general, de manera distinta en cada rotor como indica (2.1). Este procedimiento se esquematiza en la figura 1.

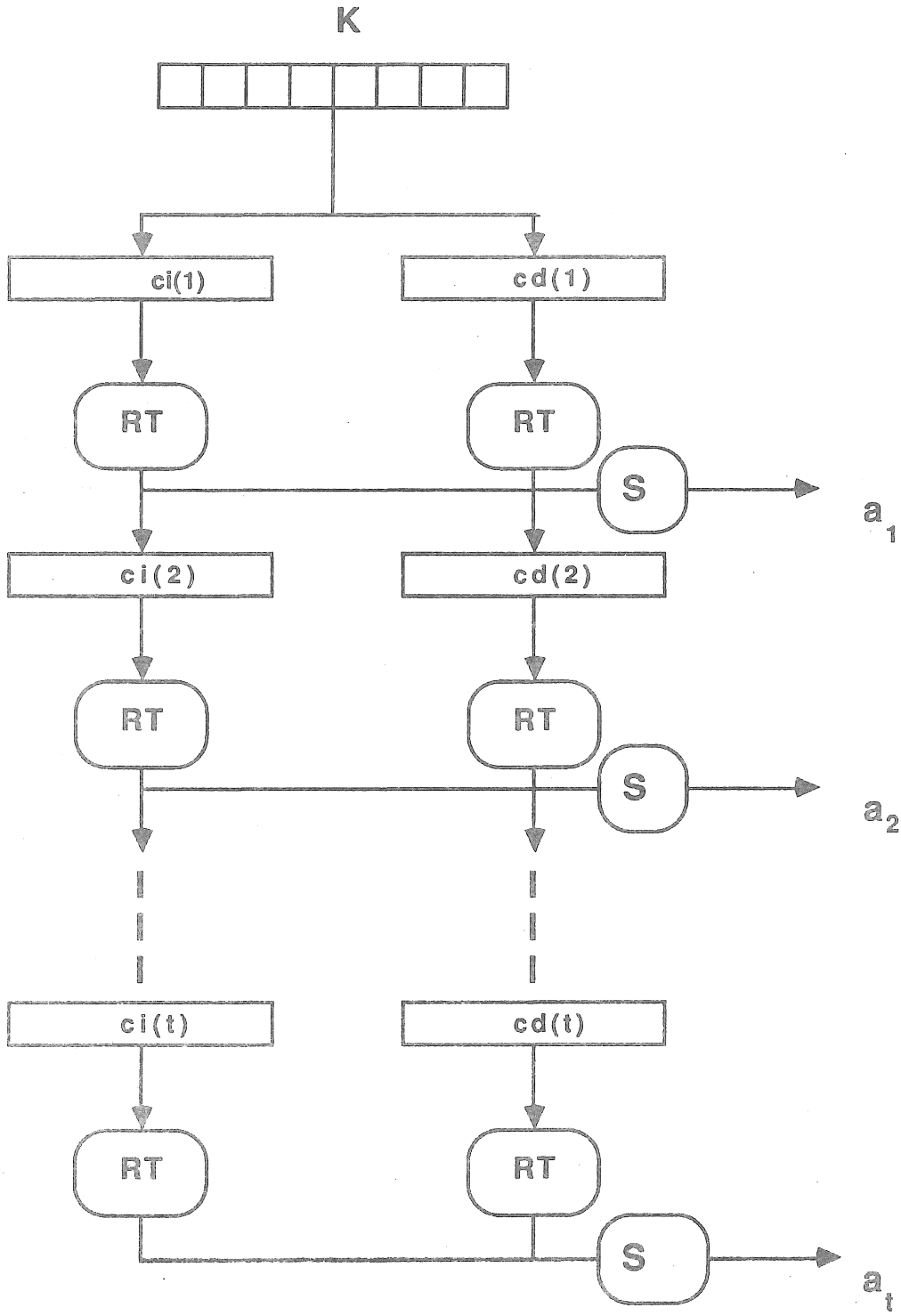

figura 1: Obtención de los a_i

Obtención de H :

H debe cumplir la condición

$$Mcd\ (\ H\ ,\ n^t\ ) = 1 \qquad (4.1)$$

Por esto, el proceso que la genera obtiene un número m de manera semejante a la usada para generar los a_i a partir de los dígitos de ci y cd,
y calcula iterativamente valores Hi de la siguiente forma :
Se calcula primero H_1 :

$$H_1 = m\ mod\ (\ n^t\ ) \qquad (4.2)$$

Luego se comprueba si este número H_1 cumple la condición **(4.1)**. De no ser así, se busca el primero entre los siguientes mayores que él con el algoritmo :

$i = 1$
while $Mcd\ (\ H_i,\ n^t\ ) \neq 1$ *do*
begin
$H_i + 1 = (\ H_i + 1)\ mod\ n^t$;
$i = i + 1$;
end
$H = H_i$;

El número N_O que da los desplazamientos (posiciones) iniciales de los rotores se obtiene de la misma manera que H con la salvedad que no necesita cumplir la condición :

$$Mcd(\ N_O,\ n^t\ ) = 1 \qquad (4.3)$$

Finalmente, los números D y E se pueden obtener de manera similar verificando que se cumpla:

$$Mcd(\ D + E,\ n^t\ ) = 1 \qquad (4.4)$$

5 - Cifrado y descifrado

El procedimiento que cifra un archivo usa 3 parámetros : nombre de un archivo de entrada a cifrar, nombre de un archivo de salida donde quedará el archivo transformado, y una palabra clave.

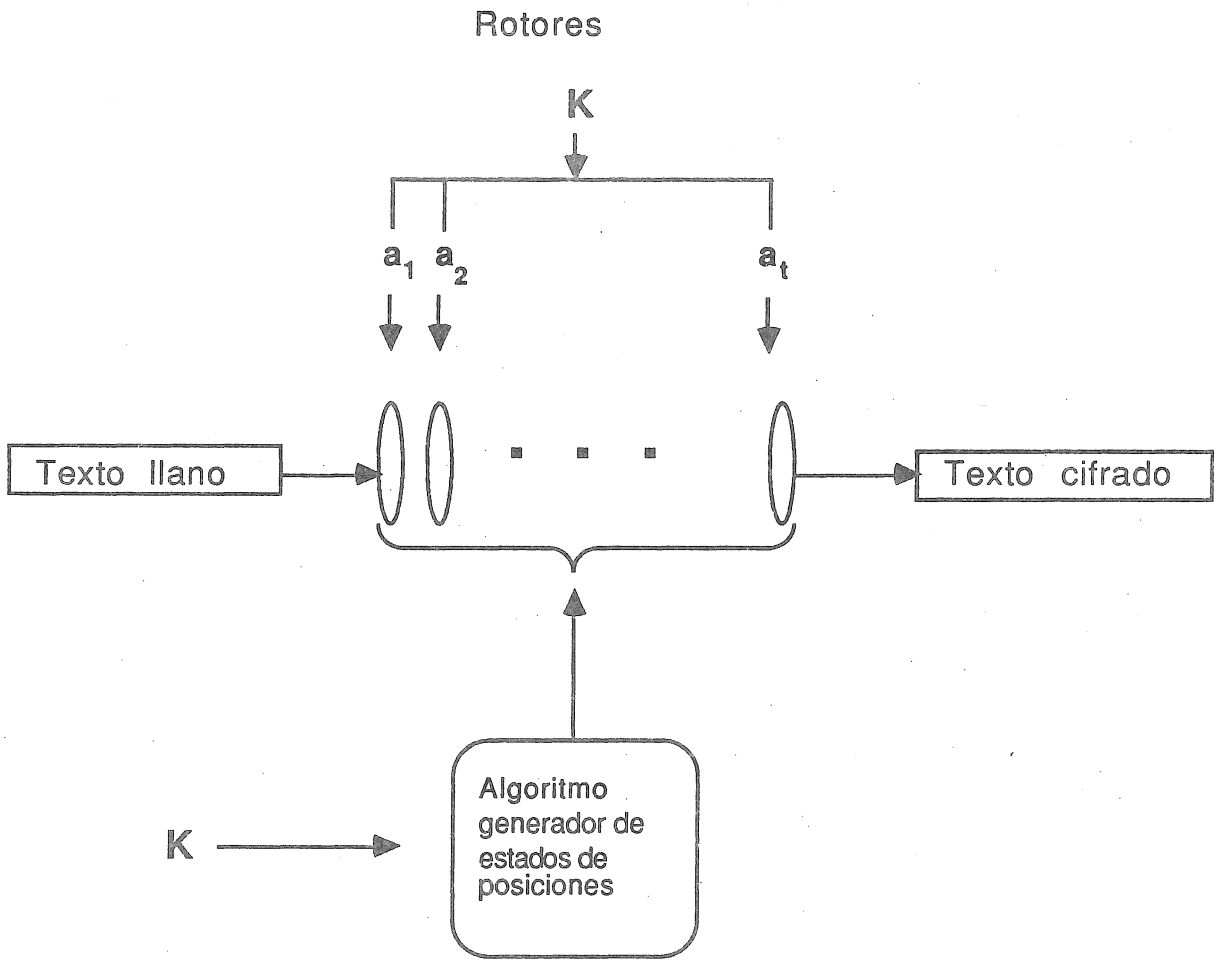

fig 2 Esquema del cifrado

Un subprograma se encarga de recibir la clave y generar los números que a partir de ella se derivan: a_i, $i = 1 \ldots t$; H ; N_O ; D y E . Luego, otro procedimiento aplica a cada carácter las t transformaciones correspondientes cada una al paso por un rotor. Una vez finalizadas estas transformaciones se escribe el cáracter resultante en el archivo de salida y se invoca un tercer subprograma que genera un nuevo estado de posiciones para el sitema de rotores por medio del algoritmo descrito anteriormente.

Para descifrar el texto, los parámertos necesarios son el nombre del archivo cifrado, el nombre del archivo de salida en el que se dejará el texto descifrado y la clave. El mismo procedimiento

empleado en el proceso de cifrado genera nuevamente los valores a_i , H , N_O , E y D. Con N_O se obtiene nuevamente la posición inicial de los rotores y con H se genera la misma secuencia de estados de posiciones usando el mismo algoritmo empleado en el proceso de cifrado.

En el proceso de descifrado la transformación que debe aplicársele a un carácter en cada rotor R_i es la inversa de la aplicada en el proceso de cifrado, es decir f_i^{-1} . Para obtener esta función inversa basta despejar x de la expresión **(2.1)**. Si llamamos c_i al carácter cifrado por $f_i(x)$ tenemos que :

$$f_i^{-1}(c_i) = ((c_i - 1) * inv(a_i, n)) \bmod n \qquad (5.1)$$

en que $inv(a_i, n)$ es un número d_i tal que $(a_i * d_i) \bmod n = 1$. Por esto, en el proceso de descifrado es necesario además calcular los valores d_i por medio de $d_i = inv(a_i, n)$. Esto puede hacerse recurriendo al algoritmo descrito en **[Denn83]**. El esquema del proceso de descifrado es análogo al de la figura 1 pero ahora el texto de entrada se introduce por el extremo derecho del sistema de rotores.

6 - Comentarios finales

A pesar que los rotores clásicos generan secuencias de transformaciones de período largo, se sabe que éstas son menos aleatorias de lo que podrían parecer. La variante presentada apunta a mejorar esta propiedad de aleatoriedad.

La clave, por su parte, se compone de un máximo de 8 caracteres y no se aceptan menos de 4, siguiendo de este modo, las características de las claves de entrada en UNIX **[Fili86]**. Esto da un total de claves posibles igual a 256^8 si se contemplan todos los símbolos ASCII para formar la clave. Podemos suponer, con mucha seguridad, que no se usarán más de 100 símbolos (26 mayusculas, 26 minúsculas, 10 dígitos, signos de puntuación y matemáticos). Por esto, una estimacíon más realista del número de claves posibles es $(100)^8 = (10)^{16}$. Esta cantidad dificulta suficientemente un ataque contra la clave. (Suponiendo que la elección de los caracteres es equiprobable -no hay preferencias por secuencias de caracteres con significado propio, por ejemplo-, la entropía para la clave es de $H_k = log_2 (10)^{16} \approx 53{,}152$).

El sistema de rotores descrito se encuentra funcionando en un computador NCR-Tower 32/600 bajo el sistema operativo UNIX, con un número de 4 rotores (t =4) con lo cual, el número de estados de posiciones distintas es 251^4 ; por lo tanto se tiene igual número de funciones de transformación F distintas.

Existe además una versión de este sistema, con 6 rotores, para PCs.

Referencias

Blum84 Blum, M. and Micali, S. "How to Generate Cryptographically Strong Sequences of Pseudo-random bits", SIAM J. of Comp., 13, pp. 850-864 (Nov. 1984).

Blum86 Blum, L., Blum M. and Schub, M. "A Simple Unpredictable Pseudo-random number Generator" SIAM J. of Comp., 15, pp. 364-383 (May 1986).

Denn83 Denning, D. E. R. "Cryptography and Data Security", Addison-Wesley (1983).

Fili86 Filipsky, A. and Hanko, J. "Making Unix Secure", Byte pp. 113-128 (April 1986).

Frei84 Freize, A.M.; Kannan, R. and Lagarias, J. C. "Linear Congruential Generators do not Produce Random Sequences", Proc. of the 25th. Symp. on Foundations of Comp. Sci., IEEE, N.Y., pp. 480-484 (1984).

Gold85 Goldreich, O.; Goldwasser, S. and Micali, S. "How to Construct Random Functions", J. of the Assoc. for Comp. Mach, 33, 4, pp. 792-807 (Oct. 1986).

NBS77 "Data Encryption Standard", Federal Information Processing Standard Publication 46, National Bureau of Standards, US Departament of Commerce, Washington, D.C (Jan. 1977).

Sham83 Shamir, A. "On the Generation of Criptographically Strong Pseudorandom Sequences", ACM Trans. Comp. Syst. 1,1 pp. 38-44 (Feb. 1983).