Universidade Federal do Rio de Janeiro
Instituto de Matemática
Departamento de Ciência da Computação
Grupo de Resposta a Incidentes de Segurança

Rio de Janeiro, RJ – Brasil

Fundamentos da Criptologia
Parte II – Criptografia Simétrica

GRIS-2005-A-004

Breno Guimarães de Oliveira

A versão mais recente deste documento pode ser obtida na página oficial do GRIS

GRIS – Grupo de Resposta a Incidentes de Segurança
CCMN Bloco I 1° andar
Sala: I1021
Av. Brigadeiro Trompowski, s/nº
Cidade Universitária - Rio de Janeiro/RJ
CEP: 21949-900
Telefone: +55 (21) 2598-3309

Sumário:

1. Introdução

Como citado no artigo anterior, o processo de codificação e decodificação de uma mensagem ocorre basicamente da seguinte forma:

A chave de codificação é geralmente uma seqüência de operações matemáticas com a mensagem original, então para decodificar a mesma basta conhecermos essa chave e realizarmos as operações inversas. De fato, a chave de decodificação contém exatamente essas operações inversas, junto com quaisquer outras informações necessárias para a transformação, como valores preestabelecidos para variáveis, etc. Quando isso ocorre, estamos falando de *criptografia de chave simétrica*. O problema fundamental de tais métodos é que eles exigem um canal seguro para a transmissão das chaves – do contrário o "inimigo" poderia roubá-la e ler todos os seus segredos –, mas um canal só seria seguro com criptografia, e isso naturalmente leva a um círculo vicioso desconfortável.

2. Cifras de Substituição

As cifras de substituição são essencialmente tabelas ou funções de conversão das letras do alfabeto – separadas individualmente ou em grupos – para um ou mais caracteres diferentes, sejam letras, números ou símbolos. O processo de decodificação é realizado simplesmente desfazendo a conversão. Os métodos de substituição costumam ser separados em 3 tipos diferentes: monoalfabéticas (cada caractere da mensagem original é transformado em outro caractere, número ou símbolo), monofônicas (cada caractere da mensagem original pode ser transformado em um conjunto de outros caracteres, números ou símbolos) e polialfabéticas (combinações de várias substituições monoalfabéticas e monofônicas).

2.1 ROT-13 (e outras substituições monoalfabéticas e monofônicas)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M

Definição: O método ROT-X talvez seja um dos mais populares da criptografia amadora, devido a sua enorme facilidade de implementação. Consiste simplesmente na rotação do alfabeto X casas para a direita. Para decodificar, basta girar cada letra novamente, dessa vez X casas para a esquerda. Números e pontuações não são codificados. Conta-se que Júlio César usava o método ROT-3 para enviar mensagens

a suas tropas, e esse método ficou conhecido como a *Cifra de César*. A rotação de 13 casas tornou-se extremamente popular nos dias de hoje, devido justamente a sua facilidade de codificação e decodificação. Isso porque o nosso alfabeto possui 26 letras, então a mesma função de codificação pode ser usada para a decodificação (em notação matemática, M = ROT(ROT(M)), onde M é a mensagem e ROT() é a função ROT-13). Para rotações com qualquer outro valor que não o 13, é preciso uma função diferente para decodificar, movendo os caracteres para a esquerda X casas. O ROT-13 é usado em diversas páginas e grupos de discussão da Internet, essencialmente para evitar que pessoas acidentalmente vejam textos ofensivos – como piadas obscenas – ou que contenham o final de um filme ou livro. Assim, a menos que você deliberadamente opte por decodificar o texto, ele será apenas um monte de caracteres esquisitos. Diversos programas de e-mail já vêm com a opção de decodificar textos em ROT-13.

Exemplo:
```
"O assassino é o mordomo! Eu já vi esse filme 17 vezes..."    (original)
"B nffnffvab r b zbeqbzb! Rh wn iv rffr svyzr 17 irmrf..."    (ROT-13)
"R dvvdvvlqr h r prugrpr! Hx md yl hvvh iloph 17 yhchv..."    (ROT-3)
"N zrrzrrhmn d n lnqcnln! Dt iz uh drrd ehkld 17 udydr..."    (ROT-25)
```

Código-fonte: o código abaixo escrito em C faz uma codificação simples em ROT-X, com as informações entradas pelo usuário. Note que ele não modifica números nem caracteres acentuados, mas discrimina letras maiúsculas de minúsculas.

```
------------------------------ rot-x.c (início) -------------------------------------
#include <stdio.h>
#include <ctype.h>

void main(void)
{
    int chave, entrada;
    printf("entre com a chave (0-25): ");
    scanf("%d", &chave);
    chave %= 26;
    printf("entre com o texto a ser codificado em ROT-%d (ctrl+c para sair):\n", chave);
    while( entrada = getchar( ) ) {
        if( islower( entrada ) )
            entrada = 'a' + (entrada - 'a' + chave) % 26;
        else if( isupper( entrada ) )
            entrada = 'A' + (entrada - 'A' + chave) % 26;
        putchar( entrada );
    }
}
------------------------------ rot-x.c (fim) ----------------------------------------
```

Variações: Como pôde ser observado, a codificação ROT-X é uma substituição monoalfabética bastante simples. No entanto, ela pode ser melhorada consideravelmente através de alguns recursos. O primeiro seria realizar substituições também para espaços e pontuações da mensagem, o que dificulta a separação de palavras pelo criptoanalista, especialmente se as mensagens forem pequenas. Além disso, é interessante usar números (de preferência em outras bases que não a decimal) ao invés de letras para o criptograma, também visando iludir o criptoanalista. No entanto, você verá na *Parte V* desse texto que decodificar cifras de substituição monoalfabéticas é extremamente simples.

Uma boa alternativa é a utilização de cifras monofônicas, que convertem cada caractere em outro qualquer de um conjunto específico de caracteres. Assim, a letra ‘a’ pode ser codificada nos valores ‘09’, ‘37’, ‘88’ ou ‘89’, por exemplo, a critério do remetente. No criptograma resultante, qualquer um desses valores será traduzido na letra ‘a’. Mas cuidado: se for usar símbolos que se repetem (como letras do alfabeto ou números) para a substituição, não se esqueça de usar o mesmo número de “dígitos” para cada letra da mensagem original (no exemplo acima, usamos dois dígitos). Se esse número for diferente, como por exemplo ‘a’ = 1 (um dígito), b = 2, ..., j = 10 (dois dígitos), k = 11, etc., o decodificador não poderá diferenciar o criptograma “11” de “aa” ou “k”, mesmo possuindo a tabela de conversão.

Código-fonte: o código abaixo codifica e decodifica mensagens usando uma substituição monofônica. Note que a escolha dos números da tabela é completamente arbitrária. Assim, enquanto a letra ‘a’ possui quatro números equivalentes, ‘e’ possui sete, e ‘z’ possui apenas um. Você pode modificar a tabela como desejar, adicionando ou removendo equivalências.

```
------------------------------ monofono.c (início) -----------------------------------
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <time.h>

void main( void )
{
    time_t rnd;
    int modo, entrada, codigo=10, i, j;
    /* no vetor abaixo, o primeiro valor indica o numero de elementos, e os zeros ao final sao ignorados. Exemplo:
    // a // 4,  9, 37, 88, 89,  0,  0,  0,  0,     (4 valores para letra ‘a’ (9, 37, 88, 89))
    // b // 4, 11, 14, 59, 70,  0,  0,  0,  0,     (4 valores para letra ‘b’ (11, 14, 59, 70))
    // c // 5, 25, 31, 55, 61, 90,  0,  0,  0,     (5 valores para letra ‘c’ (25, 31, 55, 61, 90))    */

    int alfabeto[ 26 ][ 9 ] = {
    /* a */  4, 9,37,88,89, 0, 0,0,0, /* b */ 4,11,14,59,70, 0, 0, 0,0, /* c */ 5,25,31,55,61,90, 0, 0, 0,
    /* d */  4,26,40,69,95, 0, 0,0,0, /* e */ 7, 5,16,44,60,76,94,99,0, /* f */ 3,19,24,67, 0, 0, 0, 0, 0,
    /* g */  3, 0,97,28, 0, 0, 0,0,0, /* h */ 4, 6,15,39,68, 0, 0, 0,0, /* i */ 6,29,33,47,50,57,81, 0, 0,
    /* j */  3,12,46,62, 0, 0, 0,0,0, /* k */ 1,83, 0, 0, 0, 0, 0, 0,0, /* l */ 4, 2, 8,23,58, 0, 0, 0, 0,
    /* m */  4,18,21,64,75, 0, 0,0,0, /* n */ 4,10,35,77,65, 0, 0, 0,0, /* o */ 8,20,27,53,73,74,80,84,98,
    /* p */  4,32,36,71,85, 0, 0,0,0, /* q */ 4,22,51,17,48, 0, 0, 0,0, /* r */ 6, 7,38,42,63,91,92, 0, 0,
    /* s */  6,34,43,54,72,87,96,0,0, /* t */ 4,13,56,66,78, 0, 0, 0,0, /* u */ 4, 4,45,49,82, 0, 0, 0, 0,
    /* v */  3, 1,30,79, 0, 0, 0,0,0, /* w */ 1,86, 0, 0, 0, 0, 0, 0,0, /* x */ 2, 3,41, 0, 0, 0, 0, 0, 0,
    /* y */  1,93, 0, 0, 0, 0, 0,0,0, /* z */ 1,52, 0, 0, 0, 0, 0, 0,0 };

    printf( "\nCodificacao monofonica:" );
    while( 1 )    {
        printf( "\n\n1-Codifica\n2-Decodifica\n(ctrl+c para sair)\n" );
        modo = getc( stdin );
        fflush( stdin );
        if( modo == '1' ) {   /* codificando */
            srand( ( unsigned )time( &rnd ) );
            puts( "digite mensagem a ser codificada:" );
            while( (entrada = tolower( getchar( ) ) ) != 10 ) {
                    if( isalpha( entrada ) ) {
                        entrada = alfabeto[ entrada - 'a' ][ 1 + rand() % alfabeto[ entrada - 'a' ][ 0 ] ];
                        printf( "%02d", entrada );
                    }
                    else
                        printf( "%c", entrada );
        }   }
        else {    /* decodificando */
            puts( "digite mensagem a ser decodificada:" );
            fflush( stdin );
```

```
            while( ( entrada = getchar( ) ) != 10 )
                if( isdigit( entrada ) ) {
                    entrada -= '0';
                    if( codigo < 10 ) {
                        codigo = codigo * 10 + entrada;
                        entrada = -1;

                        for( i = 0 ; i < 26 && entrada == -1 ; i++ )
                            for( j =alfabeto[ i ][ 0 ] ; j > 0 && entrada == -1 ; j-- )
                                if( codigo == alfabeto[ i ][ j ] ) {
                                    entrada = i + 'a';
                                    codigo = 10;
                                }
                        putchar( entrada );
                    }
                    else
                        codigo = entrada;
                }
                else
                    putchar( entrada );
}  }   }
---------------------------- monofono.c (fim) -------------------------------------
```

Adicionalmente, usar *nulos* (símbolos ou códigos incluídos no criptograma, mas que não significam nada e são ignorados na hora da decodificação) e *repertórios* (grupos de caracteres ou mesmo palavras inteiras que podem ser transformados em um ou mais símbolos codificados) dificulta ainda mais o trabalho do criptoanalista, justamente por aumentar as possibilidades de codificação. O uso de nulos pode ser incrementado pela inclusão de lixo (conjunto de caracteres quaisquer) no criptograma, adicionado (ou precedido) por um símbolo que signifique "ignore os últimos (ou próximos) X caracteres". Essa técnica foi usada na *Grande Cifra de Rossignol*, dos especialistas franceses (além de pai e filho) Antoine e Bonaventure Rossignol, no século XVII. Para exemplificar o aumento da dificuldade de decifração de métodos com repertório, imagine apenas que uma tabela que possua quatro equivalências para cada letra, e o repertório de quatro equivalências para o grupo de palavras "st" e quatro para o grupo "te", poderia transformar a palavra "teste" em 1920 criptogramas diferentes!

2.2 Cifra de Vigenère (e outras substituições polialfabéticas)

Definição: Talvez o método criptográfico de substituição polialfabética mais conhecido seja a chamada *Cifra de Vigenère*, esquematizada pelo diplomata francês Blaise de Vigenère no século XVI. Trata-se da combinação de 26 substituições ROT-X, realizadas de acordo com uma chave e com auxílio da tabela abaixo:

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
01:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
02:	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
03:	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
04:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
05:	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
06:	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
07:	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
08:	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
09:	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
10:	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
11:	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
12:	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
13:	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
14:	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
15:	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
16:	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
17:	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
18:	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
19:	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
20:	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
21:	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
22:	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
23:	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
24:	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
25:	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
26:	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Essa tabela nada mais é do que o conjunto das 26 possíveis rotações do alfabeto, dispostas em ordem crescente a partir da primeira linha, onde não há rotação. Para codificar uma mensagem, primeiro precisamos escolher a chave, que nos indicará o grau de rotação (chave monoalfabética) para cada caractere da mensagem. Agora é só procurar o criptograma equivalente de cada letra da mensagem original na coluna da linha que começa com cada letra da chave. Se a chave for menor que a mensagem, basta repeti-la até que tenham o mesmo tamanho. No exemplo a seguir, vamos escolher a palavra "*vigenere*" como chave. Queremos codificar a frase "*esses romanos sao uns neuroticos*".

Antes de mais nada, precisamos expandir a chave até que tenha o mesmo tamanho da mensagem:

Mensagem: *"esses romanos sao uns neuroticos"*
Chave expandida: *"vigen erevige ner evi generevige"*

Para codificarmos a primeira letra ("e"), procuramos na tabela pela linha que começa com a primeira letra da chave ("v"), que é a linha 22. Em seguida, procuramos a interseção dessa linha com a coluna que começa com a primeira letra da mensagem ("e"), que é a coluna 05. O valor encontrado nessa interseção ("z") é o criptograma equivalente para essa letra. Repetimos o processo agora para a segunda letra da mensagem, usando a segunda letra da chave, e assim por diante, até termos codificado toda a mensagem. Vejamos como fica a codificação da primeira palavra:

Chave (linha)	Letra original (coluna)	Letra codificada
V (22)	E (05)	Z
I (09)	S (19)	A
G (07)	S (19)	Y
E (05)	E (05)	I
N (14)	S (19)	F

O criptograma completo será:
"ZAYIF VFQVVUW FEF YIA TIHVFXDKUW"

Para decodificar, basta encontrar a letra atual do criptograma na linha da letra atual da chave. A coluna onde a letra se encontra indica a letra original.

Código-fonte: O código abaixo codifica e decodifica mensagens usando a Cifra de Vigenère. Nesse programa, o tamanho da chave está limitado a 80 caracteres. Você pode trocar esse valor por qualquer outro maior que zero, bastando apenas modificar o valor de CHAVE_MAX, definida no inicio do código.

```
------------------------------ polialfa.c (inicio) -------------------------------------
#include <stdio.h>
#include <ctype.h>
#include <string.h>

#define CHAVE_MAX   80       /* Maximo de caracteres da chave */

void main(void)
{
    int modo, letra, i, j, nChave;
    char chave[CHAVE_MAX];

    printf( "\nCifra de Vigenere (codificacao polialfabetica):" );
    while( 1 ) {
        printf( "\n\n1-Codifica\n2-Decodifica\n(ctrl+c para sair)\n" );
        modo = getc( stdin );
        fflush( stdin );
        printf("digite a chave (max. %d caracteres):", CHAVE_MAX);
        fgets(chave, CHAVE_MAX, stdin);
        nChave = strlen(chave);
        for(i=j=0; i < nChave; i++) { /* converte chave em deslocamentos validos */
            if(isalpha(chave[i])) {
                chave[j] = toupper(chave[i]) - 'A';
                j++;
            }
        }
        chave[j] = 0;
        if(j==0) {
            puts("chave precisa ter no minimo um caractere entre 'a' e 'z'.");
            return;
        }
        i = 0;
        puts("digite a mensagem:");
        fflush( stdin );
        while ((letra = toupper(getchar())) != 10) {
            if(isalpha(letra)) {
                if(modo == 1)
                    letra = ( (chave[i] + letra - 'A') % 26 ) + 'A';        /* codificando */
                else
                    letra = ((letra - 'A' - chave[i] + 26) % 26 ) + 'A';    /* decodificando */
                i++;
            }
            if(i == j) /* repete a chave, caso tenha chegado ao fim */
                i = 0;
            putchar(letra);
}   }   }
------------------------------ polialfa.c (fim) -------------------------------------
```

Variações: A própria definição de codificação polialfabética permite infinitas variações para a mesma. Por tratar-se de uma coletânea de substituições monoalfabéticas, todas as recomendações anteriores também valem para tornar a cifra mais segura. Convém ainda, ao contrário do que a Cifra de Vigenère propõe, usar diferentes métodos de substituição monoalfabética, ao invés de apenas diferentes chaves para um mesmo método.

Apêndice – A Cifra da Garrafa de Cerveja

Definição: Essa é uma cifra no mínimo interessante. Conta a história que, na universidade de Yale, existe uma sociedade secreta de beberrões chamada "Skull and Bones" (Crânio e Ossos). Em 1999, um estudante do MIT (Instituto de Tecnologia de Massachussets, EUA) supostamente descobriu no cofre dessa sociedade secreta a surpreendente explicação para a origem e significado da conhecida música de bar "99 Bottles of Beer on the Wall" (99 garrafas de cerveja na parede). A música, cantada pelos bêbados mais chatos, começa com o seguinte verso:

"99 bottles of beer on the wall,	(99 garrafas de cerveja na parede)
99 bottles of beeeeeeeeeeer…	(99 garrafas de cerveeeeeja)
Take one down,	(pegue uma)
Pass it around,	(passe adiante)
98 bottles of beer on the wall."	(98 garrafas de cerveja na parede)

Os versos seguintes têm a mesma forma, mas sempre com uma cerveja a menos. A música termina com "No bottles of beer on the wall" (nenhuma garrafa de cerveja na parede), embora os cantores embriagados costumem adormecer no balcão após o 12º ou 13º verso, e dificilmente consigam chegar ao final da canção...

Segundo o estudante, a música aparentemente inofensiva – exceto aos ouvidos dos que têm que agüentar esse pessoal no bar – descreve o procedimento criptográfico usado pelos membros dessa sociedade secreta para proteger informações confidenciais. Esse procedimento, denominado a "Cifra da Garrafa de Cerveja", foi desenvolvido no início do século XVIII por um membro do "Skull and Bones" com aptidões matemáticas, e a música teria sido criada como um mnemônico para o mesmo. Vamos ao procedimento (que, nas palavras de Ronald Rivest, "é complicado o bastante de modo que você provavelmente não deve estar bebendo cerveja quando for tentar implementá-lo"):

A mensagem que desejamos criptografar é primeiramente transformada num número, usando dois dígitos para cada letra. Assim, A=01, B=02, C=03, etc., e 00 para "espaço". A frase: "AGUA FAZ MAL A SAUDE", por exemplo, se transformaria no número 01072101000601260013011200010019012104 05, que tem 40 dígitos – correspondendo aos 20 caracteres da frase. Na cifra da garrafa de cerveja, cada dígito é representado por uma garrafa, e no exemplo acima usaríamos 40 garrafas. As garrafas são alinhadas "no muro" da esquerda para a direita, com a garrafa mais a esquerda contendo o dígito mais significativo. Caso o número obtido resulte em menos de 99 garrafas – como foi o caso – complete-o com garrafas contendo 0 (zero), sempre a esquerda do número. Para segredos mais longos, comece com mais garrafas, e cante mais versos.

Adicionalmente, existem a chave de codificação – conhecida como "crânio" (skull) – e a "mesa", para onde as garrafas vão. O crânio nada mais é do que um número grande, que vai servir para a codificação. Agora que temos as garrafas contendo a mensagem original na parede, o crânio na mão, e a mesa vazia, é só começar a cantar.

"k garrafas de cerveja"

Primeiro, pegue a garrafa de cerveja mais a esquerda da parede e coloque-a no canto direito da parede, depois de todas as outras. O número k indica quantas garrafas de cerveja ainda estão na parede. Multiplique esse número por (10k + 1) e, do valor resultante, descarte aqueles da extremidade mais significativa, se necessário, para formar o novo número com k dígitos. Vejamos o exemplo:

Mensagem: "CERVEJA"
Número na parede: 03051822051001 (ocupa 14 garrafas)
Cante: "14 garrafas de cerveja"
Novo número na parede: 30518220510010
*Multiplique por 141 (que é 10*14+1)*
Resultado: 4303069091911410 (16 garrafas, descartar duas da extrema esquerda)
Novo número na parede: 03069091911410

"na parede"

Ao cantar esse trecho, adicione o crânio ao número na parede, mantendo somente as k garrafas mais a esquerda do resultado.

Número na parede: 03069091911410 (ocupa 14 garrafas)
Cante: "na parede"
Crânio: 975648116853487300645420094647 72
Soma: 97564811685348730067611101376182 (32 garrafas, descartar 18 da esquerda)
Novo número na parede: 67611101376182

"pegue uma"

Remova a garrafa mais a direita da parede. Chame o dígito dentro da garrafa de "osso", mas não o coloque na mesa ainda...

Número na parede: 67611101376182
Cante: "pegue uma"
Novo número na parede: 6761110137618
Osso: 2

"passe adiante"

Ao cantar esse trecho, o seguinte procedimento deve ser seguido. Imagine que começamos com g garrafas na mesa, representando um número de g dígitos. Seja o "grande osso" um número de g+1 dígitos, onde cada dígito é o osso. Adicione o grande osso a dez vezes o número que está na mesa, mantendo apenas os g+1 dígitos mais a direita do resultado.

Número na mesa: 844291 (g = 6)
Cante: "passe adiante"
Osso: 2
Grande osso: 2222222 (g+1 dígitos)
*Mesa * 10: 8442910*
Soma: 10665132 (g+2 dígitos, retirar o mais a esquerda)
Novo número na mesa: 0665132 (agora g=7)

Num primeiro momento, a mesa estará vazia, mas o procedimento permanece o mesmo. No exemplo que estamos fazendo, teríamos:

Número na mesa: nenhum (g=0)
Cante: "passe adiante"
Osso: 2
Grande osso: 2
*Mesa * 10: 0*
Soma: 2
Novo número na mesa: 2 (agora g=1)

Quando não houver mais nenhuma garrafa de cerveja na parede, e a música tiver acabado, o número resultante na mesa será o criptograma desejado.

Michael Neve e Eric Peeters escreveram um programa de codificação e decodificação da *Cifra da Garrafa de Cerveja* para o Matlab, disponíveis em *http://www.cs.virginia.edu/~evans/cs588/manifests/* (arquivos "*beer2.m*" (codificação) e "*unbeer2.m*" (decodificação))

Bibliografia:

"Codes, Ciphers & Codebreaking" Goebel, G. http://www.vectorsite.net/ttcode.html (06/05/03)
"The Beer Bottle Cipher" – Rivest, R. L. http://theory.lcs.mit.edu/~rivest/publications.html (02/04/03)