A UNIVERSAL ALGORITHM FOR HOMOPHONIC CODING

Christoph G. Günther
Asea Brown Boveri
Corporate Research
CH-5405 Baden, Switzerland

ABSTRACT

This contribution describes a coding technique which transforms a stream of message symbols with an arbitrary frequency distribution into a uniquely decodable stream of symbols which all have the same frequency.

I . INTRODUCTION

In a Caesar cipher each letter from the alphabet $\{a, b, \ldots, z\}$ is replaced by the successor of the successor of its successor, *i.e.* the alphabet is shifted by three: $\{a, b, \ldots, z\} \rightarrow \{d, e, \ldots, c\}$. In general, there are 26 possible shifts, and we say that the cipher defined by these shifts has a key size of $\log_2 26 \simeq 4.7$, which is very small. If we, however, consider the set of all permutations of the alphabet $\{a, b, \ldots, z\}$, we get a cipher with a key size $\log_2 26! \simeq 88$. This is more than one third larger than 56, which is the key size of todays most widely used cipher DES. Nevertheless, the cipher described is not secure for the encryption of English plaintext. In English the letters from the alphabet occur with the frequencies $p_e \simeq 0.13$, $p_t \simeq 0.09$, $p_a \simeq 0.08$, ... , and $p_z \simeq 0.001$ (see *e.g.* [1]), and therefore a frequency analysis of the cryptogram immediately reveals the chosen permutation.

In this respect, English is neither an exception amongst the natural languages nor amongst the technical data streams like ASCII codes or Δ-modulated speech. All of them show statistical irregularities through unequal probabilities of the symbols or correlations between the symbols. The above permutation cipher is also not exceptional, it is the most general block cipher defined on an alphabet of 26 symbols.

In order to describe more accurately the weakness discussed, we consider the uncertainty of the key, *i.e.* of the enciphering permutation, when n symbols or

C.G. Guenther (Ed.): Advances in Cryptology - EUROCRYPT '88, LNCS 330, pp. 405-414, 1988.

blocks of symbols of the cipher text are known. This uncertainty is quantified by the equivocation of the key $k \in \mathcal{K}$ given n cipher blocks $(c_0, c_{-1}, \ldots, c_{-(n-1)}) \in \mathcal{C}^n$ [2]:

$$H(k|c_0, c_{-1}, ..., c_{-(n-1)}). \tag{1}$$

The smallest n for which the key is completely determined is called the unicity distance d. According to Shannon [2] and Hellman [3], it is given by

$$d = \frac{H(k)}{1 - \frac{1}{\tau} H(c_0, c_{-1}, ..., c_{-(\tau-1)})}, \tag{2}$$

where τ is the length over which the blocks become statistically independent and where the basis of the logarithms involved in the definition of H is equal to the size C of the cipher alphabet $\mathcal{C}$. For English texts, Hellman [3] has estimated that

$$\frac{1}{\tau} H(c_0, c_{-1}, ..., c_{-(\tau-1)}) \simeq 0.32, \tag{3}$$

which implies

$$d \simeq 1.5\, H(k). \tag{4}$$

In the case of DES, the key is therefore completely specified by the redundancy in the text after two cipher blocks of 64 bits each. The only property that has prevented so far the design of efficient algorithms to break DES is the mismatch between the statistical information and the block structure of DES.

Even if cryptography is based to a large extent on the complexity of certain computations, unconditionally secure systems are preferable. In the present situation, unconditional security can be achieved by a suitable conditioning of the message either by reducing its redundancy with a data compression algorithm or by increasing its entropy in a randomisation process. The reduction of redundancy is more attractive from a theoretical point of view. The data compression algorithms known today, however, only imply a unicity distance proportional to the size of their encoding table, which makes them practically useless for the present purpose.

Amongst the randomisation techniques, homophonic coding seems by far the most adequate, as was pointed out by Massey [4]. The basic idea of such a coding is to improve the distribution of the symbols in the cipher text alphabet $\mathcal{C}$ towards equidistribution by introducing a suitable number of representations for each letter from the message alphabet $\mathcal{M}$ and by randomly choosing one of the representations at each step. Such a coding was already used in 1401 by the Duke of Mantua in his correspondence with Simeone de Crema [5] and is also well known through the

Beale ciphers [6]. An example which by its simplicity is particularly suitable to explain this type of encoding was proposed by Massey [4]. In this example, the message stream consists of independent, identically distributed (*i.i.d.*) random variables from the alphabet $\mathcal{M} = \{a, b\}$ with the letter frequencies $p_a = \frac{3}{4}$ and $p_b = \frac{1}{4}$. A homophonic code for this example is defined on the image alphabet $\mathcal{C} = \{0,1\}^2$ by

$$\begin{aligned} a &\longrightarrow \begin{cases} 00 \\ 01 \\ 10 \end{cases} \quad \textit{with probability } 1/3 \textit{ each}, \\ b &\longrightarrow \{\ 11 \quad \textit{with probability } 1, \end{aligned} \tag{5}$$

i.e. the message $m = a$ is encoded at random into $00, 01, 10$, with equal probabilties. As a consequence of this encoding, the message source stays *i.i.d.* and becomes equidistributed, and the unicity distance skips from $d = 5.3\ H(k)$ to infinity if at least two keys are used.

A similar approach can in principle be chosen for every rational frequency distribution. In general, this will however lead to an enormous data expansion. Furthermore, the frequency distribution completely specifies the cipher text alphabet in this scheme. Both disadvantages are avoided in the systematic approach we shall adopt now.

II . DESCRIPTION OF THE ALGORITHM

The homophonic code defined in equation (5) contains two essential elements, an encoding table, *i.e.* the association of the symbols $00, 01$ and 10 with the letter a and the association of the symbol 11 with the letter b, and an encoding rule which states that each representation of a letter has to be chosen with equal probability. The construction of these two elements are the main steps in the universal algorithm. In order to get an idea of the general form of these elements, we observe that the following mapping also defines a homophonic code for the above example:

$$\begin{aligned} a &\longrightarrow \begin{cases} 0 & \textit{with probability } 2/3, \\ 10 & \textit{with probability } 1/3, \end{cases} \\ b &\longrightarrow \{\, 11 \quad \textit{with probability } 1. \end{aligned} \tag{6}$$

This mapping causes a smaller data expansion than the previous one. The mapping itself is obtained by noting that the second bit in the strings 00 and 01 of equation (5) does neither carry information nor contribute to the equidistribution. The mapping can be interpreted as follows: if a 0 is transmitted it is to represent an a, if a 1 is transmitted it is not to represent any letter but just to tell the decoder to

wait for the next symbol in order to determine the information transmitted. With this interpretation the encoding table can be rewritten as two tables (see Figure 1) with σ denoting the prefix symbol, *i.e.* the symbol which tells the decoder to wait and to decode the next symbol according to table $T^{(2)}$:

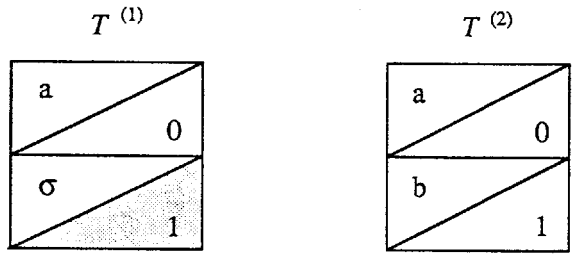

Figure 1: *The encoding tables for the example* $\mathcal{M} = \{a, b\}$, $C = \{0, 1\}$, $p_a = \frac{3}{4}$ and $p_b = \frac{1}{4}$.

This form of the encoding table immediatly suggests the association with a binary, or more generally with a C-ary representation of the frequency distribution $\{p_\alpha\}_{\alpha \in \mathcal{M}}$. And the two objectives of having a number of representations of the letters in the encoding tables which is proportional to the probability of that letter and of having at least one letter represented in each table together with the above association lead to the following general construction of the encoding tables:

Initialisation:

$i = 1$, $p_\alpha^{(0)} = p_\alpha$

Construction of the i-th table $T^{(i)}$*:*

a) The dimension κ_i of the table $T^{(i)}$ is determined by

$$\kappa_i := \lceil -\log_C(\max_{\alpha \in \mathcal{M}} p_\alpha^{(i-1)}) \rceil. \tag{7}$$

b) A number $n_\alpha^{(i)} := \lfloor C^{\kappa_i} p_\alpha^{(i-1)} \rfloor$ of symbols $\beta_\alpha^{(i,1)}, \ldots, \beta_\alpha^{(i,n_\alpha^{(i)})} \in \mathcal{C}^{\kappa_i}$ is chosen to represent the letter α in table $T^{(i)}$.

c) The remaining $n^{(i)} := C^{\kappa_i} - \sum_{\alpha \in \mathcal{M}} n_\alpha^{(i)}$ symbols $\sigma^{(i,1)}, \ldots, \sigma^{(i,n^{(i)})} \in \mathcal{C}^{\kappa_i}$ are chosen as prefix symbols.

Computation of $p_\alpha^{(i)}$ and loop control:

If $n^{(i)} = 0$, the construction is completed. If $n^{(i)} \neq 0$, the new probability distribution is determined by

$$p_\alpha^{(i)} := \frac{C^{\kappa_i} p_\alpha^{(i-1)} - n_\alpha^{(i)}}{n^{(i)}}, \tag{8}$$

i is incremented by one and the next table is constructed.

The encoding tables for the slightly more complex example $\mathcal{M} = \{a, b, c\}$, $C = \{0, 1\}$ and $p_a = \frac{5}{12}$, $p_b = \frac{1}{3}$ and $p_c = \frac{1}{4}$ are shown in Figure 2.

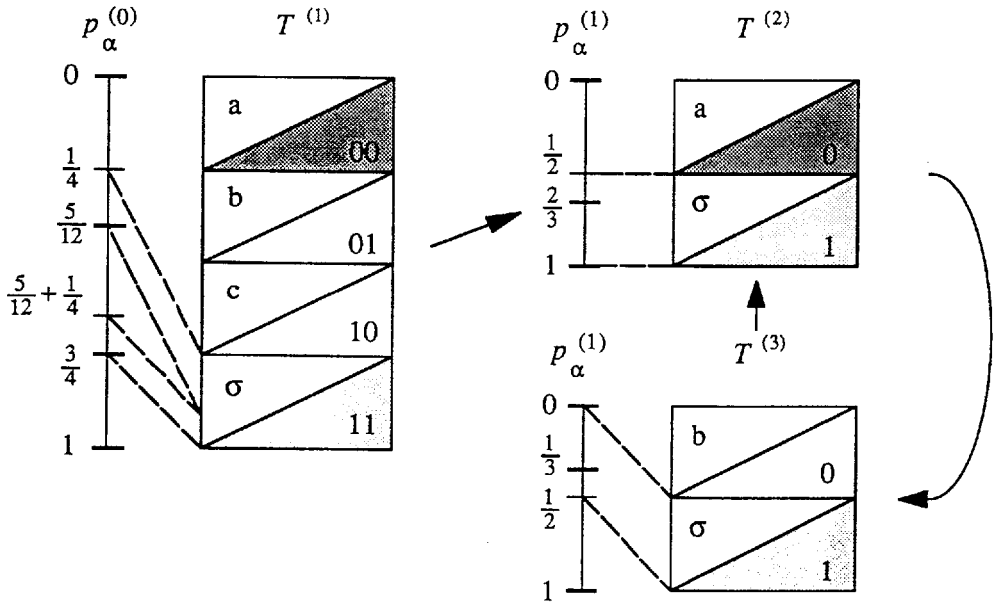

Figure 2: *The encoding tables for the example $\mathcal{M} = \{a, b, c\}$, $C = \{0, 1\}$ and $p_a = \frac{5}{12}$, $p_b = \frac{1}{3}$, $p_c = \frac{1}{4}$. The first table has size $C^2 = 4$ as, due to $p_\alpha < \frac{1}{2}$, $\forall \alpha \in \mathcal{M}$, no letter can be represented in a table of size $C = 2$. The symbols in the dark areas represent the letter a. The symbols in the pale areas are prefix symbols which are used in the representation of several letters. The codewords* 00, 110, 11110, 1111110,... *all represent the letter a.*

The number of tables generated in this example is infinite. However, only three of these tables are truly different ($T^{(2n)} = T^{(2)}$, $T^{(2n+1)} = T^{(3)}$, $\forall n \geq 1$). The partition of the interval $[0,1)$ induced by the probability distribution $\{p_a, p_b, p_c\}$, which is represented in Figure 2, is useful for the construction of the tables themselves and also for the formulation of the encoding rule. If an a is to be encoded, the rule for the first symbol reads: choose at random a number r in the interval $[0, \frac{5}{12})$, if $r < \frac{1}{4}$ transmit the symbol 00 if $r \geq \frac{1}{4}$ transmit the symbol 11 and encode a using the next table. This rule is symbolically represented in Figure 3:

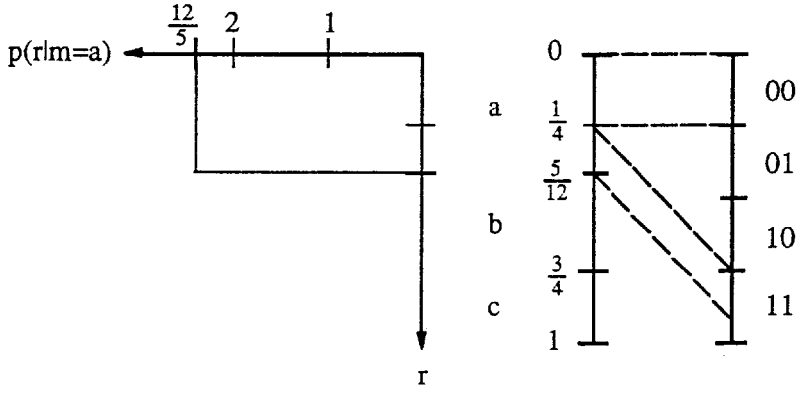

Figure 3: *Symbolic representation of the first step in the encoding of a. A number r is chosen randomly according to the distribution $p(r|m = a)$. If $r < \frac{1}{4}$ the symbol 00 is transmitted and the encoding ends, else the symbol 11 is transmitted and further steps are needed to transmit the letter a to the receiver.*

With these considerations in mind, it is no longer difficult to derive the general encoding algorithm:

a) Read a new symbol $\alpha \in \mathcal{M}$ from the data stream.

b) Set $i = 1$.

c) Choose a random number $r \in [0, p_\alpha^{(i-1)})$.

d) If $C^{\kappa_i} r \leq n_\alpha^{(i)}$, transmit $\beta_\alpha^{(i, \lceil C^{\kappa_i} r \rceil)}$ and go to a),

if $C^{\kappa_i} r > n_\alpha^{(i)}$, transmit $\sigma^{(i, \lceil \frac{1}{p_\alpha^{(i)}} (C^{\kappa_i} r - n_\alpha^{(i)}) \rceil)}$, increment i by one and go to c).

The effect of this algorithm is to combine the message source and the randomness from homophonic coding such that all symbols 00, 01, 10 and 11, and a fortiori 0 and 1, become equally likely. This does not only hold for the first step but for

every one, which immediately implies the statistical independence of the output stream if the symbols from the source are statistically independent. With these remarks, the proof of the following theorem is easy:

Theorem 1: If a message source generates a sequence of *i.i.d.* variables but with unequal letter probabilities, then the sequence obtained by applying the universal homophonic coding algorithm is *i.i.d.* and has equal letter probabilities.

Many sources are modelled more accurately by a Markovian process with finite memory. For them the following theorem applies:

Theorem 2: If the message source can be described by a Markovian process with finite memory τ, then the sequence obtained by applying the universal homophonic coding algorithm, with the probability distribution $\{p_\alpha\}_{\alpha\in\mathcal{M}}$ replaced by the conditional probability distribution $\{p_{\alpha|\alpha_{-1},\ldots,\alpha_{-\tau}}\}_{\alpha;\alpha_{-1},\ldots,\alpha_{-\tau}\in\mathcal{M}}$, is *i.i.d.* and has equal letter probabilities.

In both cases we thus have perfect statistical properties and therefore an infinite unicity distance.

So far the homophonic coding algorithm has been described without taking its practical aspects into consideration. Amongst these, the two most important ones are the termination conditions for the table construction and the data expansion.

III . TERMINATION OF THE TABLE CONSTRUCTION

Two simple conditions for the termination of the table construction are obtained from the observation that the algorithm induces the following representation of the probabilities p_α :

$$p_\alpha = \frac{1}{C^{\kappa_1}}\left(m_\alpha^{(1)} + \frac{1}{C^{\kappa_2}}\left(m_\alpha^{(2)} + \cdots\right)\right) \tag{9}$$

with

$$m_\alpha^{(i)} = n_\alpha^{(i)} \prod_{j=1}^{i-1} n^{(j)}. \tag{10}$$

This is a special form of a C-ary expansion and therefore easily implies:

Lemma 3 : a. If all probabilities have a finite C-ary expansion, the table construction stops.

b. If all probabilities are rational, the sequence of constructed tables becomes ultimately periodic.

Condition b is a termination condition as only a finite number of tables needs to be determined and stored. So in all practical situations the table construction terminates, but eventually after a very large number of tables.

In applications, a given key is only used for a finite message length and correspondingly the unicity distance does not need to be larger than this length. Therefore, we can tolerate a deviation of the probabilities q_γ of the cipher symbol γ from its ideal value $\frac{1}{C}$ and restrict the algorithm to a maximum of say $l+1$ tables. If this is done by constructing l tables according to the algorithm of Section II and by adding one table, which contains a representation for every symbol $\alpha \in \mathcal{M}$ with $p_\alpha^{(l)} > 0$, the probability q_γ of the symbol $\gamma \in \mathcal{C}$ is given by:

$$q_\gamma = \frac{1}{C} + (\tilde{q}_\gamma - \frac{1}{C}) \frac{\log_C M \cdot \kappa_{l+1}}{\lambda_l} \prod_{i=1}^{l} \frac{n^{(i)}}{C^{\kappa_i}}, \tag{11}$$

where $\tilde{q}_\gamma$ is the frequency of the symbol γ in table $T^{(l+1)}$, where M is the size of the alphabet $\mathcal{M}$, where κ_{l+1} is the dimension of that table, and where λ_l is given by

$$\lambda_l = \frac{1}{\log_C M} \sum_{i=1}^{l+1} \kappa_i \prod_{j=1}^{i-1} \frac{n^{(j)}}{C^{\kappa_j}}. \tag{12}$$

In this expression, the error $q_\gamma - \frac{1}{C}$ converges exponentially to zero for $l \longrightarrow \infty$ and the Taylor expansion of the entropy

$$\begin{aligned} H(p) &= H(\frac{1}{C}) - \frac{C}{\log_e C} \sum_{\gamma \in \mathcal{C}} (q_\gamma - \frac{1}{C})^2 + \cdots \\ &= 1 - \frac{C}{\log_e C} \sum_{\gamma \in \mathcal{C}} (q_\gamma - \frac{1}{C})^2 + \cdots , \end{aligned} \tag{13}$$

therefore implies an *exponential* increase of the unicity distance with the table size l.

IV . DATA EXPANSION

From the description in Section II it is rather obvious that the algorithm will change the data rate. In some singular cases in which the distribution is concentrated on a few symbols, this change can be a lowering of the rate. In the example $\mathcal{M} = \{a, b, c, d\}$, $p_a = \frac{3}{4}$, $p_b = \frac{1}{8}$, $p_c = \frac{1}{16}$, $p_d = \frac{1}{16}$, and $\mathcal{C} = \{0, 1\}$ the compression factor is $\frac{15}{16}$. In the generic case this change will, however, be an expansion and it is very important to have some information on how large this expansion will be.

Theorem 4 : The ratio λ of the output rate divided by the input rate of the homophonic coding algorithm is

$$\lambda = \frac{1}{\log_C M} \sum_{i=1}^{\infty} \kappa_i \prod_{j=1}^{i-1} \frac{n^{(j)}}{C^{\kappa_j}} \tag{14}$$

In this theorem we have taken to our disadvantage the value $\log_C M$ for the input rate (instead of $\lceil \log_C M \rceil$) in order not to overestimate the mismatch between the usual alphabet $\{a, b, \ldots, z\}$ and the technically relevant binary alphabet. For $M \leq C$ we have the following general result:

Lemma 5 : a. If $M \leq C$, the data expansion λ is bounded by

$$\lambda \leq C \cdot \log_M C. \tag{15}$$

b. For $M = C = 2$ or 3, the distribution

$$p_j := \frac{(\frac{C-1}{C})^{j-1}}{1 - (\frac{C-1}{C})^C} \tag{16}$$

has a data expansion $\lambda = C \cdot \log_M C$.

The proof of this lemma follows easily from the observation that $\kappa_i = 1$ and $n^{(i)} \leq C-1$ if $M \leq C$. Unfortunately, the lemma is too weak for most applications.

Therefore, we have estimated the average value of λ, with the average taken over all probability distributions $\{p_\alpha\}_{\alpha \in \mathcal{M}}$. For $M \leq C$ we have obtained

$$\langle \lambda \rangle \lesssim 2 \cdot \log_M C. \tag{17}$$

A Monte Carlo simulation has confirmed this estimate and has provided the following results for the relevant cases $M = 27$ (usual alphabet with blank) and $C = 2, 4, 8, 16, 32, 64, 128, 256$: (the error of λ is ≤ 0.1)

C	=	2	4	8	16	32	64	128	256
$\langle \lambda \rangle$	=	2.7	2.4	1.9	2.4	1.7	1.7	1.6	1.8

Finally, we have also computed λ for the frequency distribution of letters in English texts, as taken from Beker and Piper [1]: (the error of λ is ≤ 0.1)

C	=	2	4	8	16	32	64	128	256
λ	=	2.7	2.3	2.0	2.3	1.6	1.5	1.6	1.8

If we compare this with the above results we see that English is quite typical. Furthermore, we note that a suitable choice of the alphabet size C can considerably reduce the data expansion. This indicates that our simple rule for the choice of the dimension κ_i of table $T^{(i)}$ was not optimal and that it can be further improved.

V . CONCLUSION

In the present contribution we have shown that homophonic coding is an efficient precoding, suitable to increase the unicity distance of a cipher to any required length. Furthermore, even if only the lower order correlations are smoothed out, attacks on the higher order dependencies become practically infeasible due to the variable length of the codewords. The additional random data transmitted causes a data expansion by a factor of roughly two. It can, however, be used to further strengthen the system by suitably randomising the cipher applied to the precoded data. Finally, we note that the described precoding can, after some small modifications, be run in an adaptive way. Homophonic coding is thus highly adequate to substantially increase the strength of ciphers in most applications.

ACKNOWLEDGMENT

I would like to thank Professor James L. Massey for his continuous interest and support.

REFERENCES

[1] H. Beker, F. Piper, *Cipher Systems, The protection of Communications,* Northwood Books, London (1982).

[2] C. E. Shannon, "Communication theory and secrecy systems," *Bell System Tech. J.*, vol. **28**, pp. 656–715 (1949).

[3] M. E. Hellman, "An extension of the Shannon theory approach to cryptography," *IEEE Trans. on Inform. Theory,* vol. **IT-23**, pp. 289–294 (May 1977).

[4] J. L. Massey, "On probabilistic encipherment," *1987 IEEE Information Theory Workshop*, Bellagio (Italy).

[5] D. Kahn, *The Codebreakers, The Story of Secret Writing,* Weidenfeld and Nicolson, London (1966).

[6] "The Beale Ciphers", The Beale Cipher Assoc., Medfield, Mass. (1978).