Computer and Network Security

Computer & Information Sciences & Engineering
University Of Florida
Gainesville, Florida 32611-6120
nemo@cise.ufl.edu

Basic Cryptography
(Pfleeger Ch. 2, KPS Ch. 1-2, etc.)

Why here? In computer security, crypto is but one of several ways of achieving isolation, and is not one of the more important ways. In network security, crypto is the main attraction - everything is done via message-passing (ultimately), so the only secure way to achieve confidentiality and the authentication needed for access control is through crypto.

3 Crypto types

3.1 Key Symmetry

- Symmetric $M = D(E(M, K), K)$
- Asymmetric $M = D(E(M, K), K^{-1})$, where $K \neq K^{-1}$ in general

3.2 Block vs. Stream

- Block
  Plaintext is broken into fixed-length blocks, which are fed to the cryptosystem for encryption one block at a time, and one block is output for each block input.
- Stream
  Plaintext is fed to the cryptosystem for encryption one symbol at a time, and a symbol is output for each symbol input.

4 Basics

4.1 Basic Goals

- Plaintext should not easily be obtained from ciphertext
- Key should not easily be obtained from ciphertext
- Keyspace should be large enough to resist brute-force attacks
- Confusion - effect of small change in plaintext on ciphertext should not be predictable
- Diffusion - small change in plaintext should affect large part of ciphertext (block ciphers, feed-forward)

4.2 Cryptanalysis

Attacks classified by amount of knowledge available to the cryptanalyst about P; it is assumed that the cryptanalyst knows $C = E(P, K)$ but does not know K. The cryptanalyst's goal may be to find just P or to find K also.

- Ciphertext-only - only C is known; quantity counts
- Recognizable plaintext - C is known, and there is a test to determine if $D(C, K')$ is valid plaintext for a candiate key K'
- Guessed plaintext - C is known, and (part or all) of the corresponding P may be guessed
- Partial plaintext - both C and some of P are known
- Known plaintext - both C and all of P are known
- Chosen plaintext - the attacker can submit any P for encryption and obtain $C = E(P, K)$

4.3 Systems Models

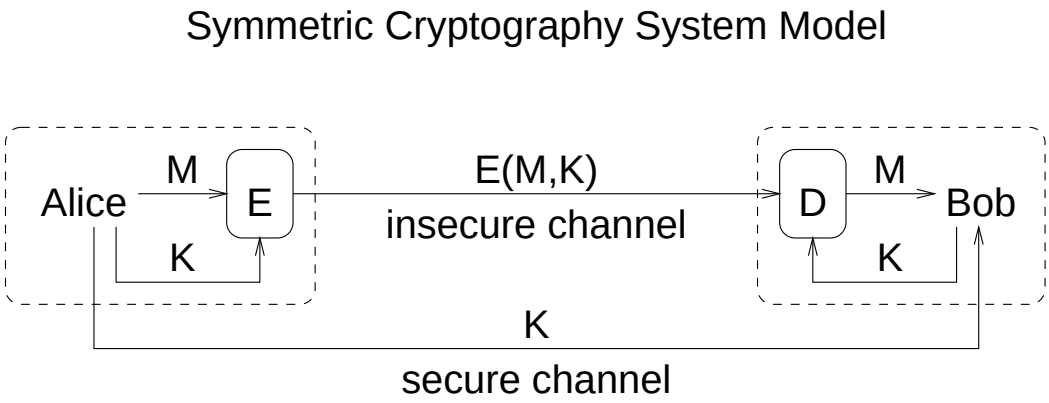

Figure 1: Symmetric cryptosystem model

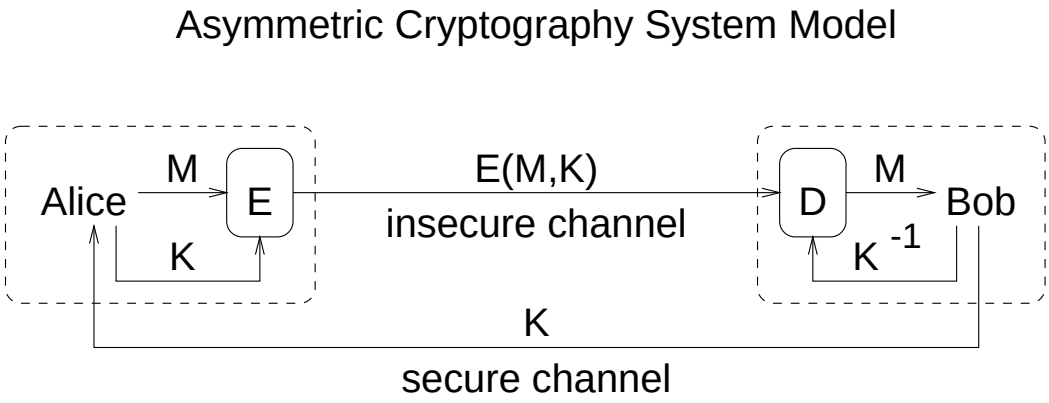

Figure 2: Asymmetric cryptosystem model

4.4 Functions

4.4.1 Converting text to numbers

- Number letters - A=0, B=1, ..., Z=25 (26 symbols)
- Both cases, numbers, '.' and ' ' (64 symbols)
- ASCII, EBCDIC, etc. (128 or 256 symbols)

4.4.2 Types of functions

Function $f : A \to B$ is

- total iff f is defined for every element $a \in A$ otherwise, f is a partial function;
- onto (surjection) iff
$$\forall\ b \in B, \exists\ a \in A \text{ such that } f(a) = b$$
- one-to-one (injection) iff
$$\forall\ b \in f(A), \exists!\ a \in A \text{ such that } f(a) = b$$
- a bijection iff f is total, one-to-one and onto;
- a permutation iff $A = B$ and f is a bijection.

Note that the image of A under f is

$$f(A) = \{b \in B \mid \exists\ a \in A \text{ such that } f(a) = b\}$$

4.4.3 Relevance to cryptography

- Only bijections are totally invertible.
- Cryptography is based on invertible functions (or else cannot decipher).
- In general, it uses keyed functions $f_k(x) = f(x, k)$ where f_k is a bijection
- An issue is the size of the key
- Another issues is the size of the key space

4.5 Groups

4.5.1 Definition

A group is a pair $G = \langle S, \diamond \rangle$, where S is a set and $\diamond$ is a binary function on S, that satisfies the four following requirements (CAIN).

1. (C) Closed:
$$\forall\ a, b \in S,\ a \diamond b \in S$$
2. (A) Associative:
$$\forall\ a, b, c \in S,\ (a \diamond b) \diamond c = a \diamond (b \diamond c)$$
3. (I) Identity:
$$\exists!\ e \in S \text{ such that } \forall\ a \in S,\ a \diamond e = e \diamond a = a$$
4. (N) Inverse:
$$\forall\ a \in S,\ \exists!\ a^{-1} \in S \text{ such that } a \diamond a^{-1} = a^{-1} \diamond a = e$$

Groups may be finite or infinite (we are interested in finite groups), and may also be commutative.

- $G = \langle S, \diamond \rangle$ is commutative iff
$$\forall\ a, b \in S,\ a \diamond b = b \diamond a$$

4.6 Monoalphabetic substitution cipher

4.6.1 Caeser cipher

$$c_i = p_i + 3\ modulo\ 26$$

4.6.2 Rotational cipher

$$c_i = p_i + k\ modulo\ N$$

where N is the size of the symbol set and key $k \in [0..N]$.

4.6.3 Multiplicative cipher

$$c_i = a \times p_i\ modulo\ N$$

where key $a \in [1..N]$ and N must be relatively prime, else a does not have an inverse.

4.6.4 Affine cipher

$$c_i = a \times p_i + b\ modulo\ N$$

where key $(a, b) \in [1..N] \times [0..N]$, and a and N must be relatively prime

4.6.5 General MSC

Must specify general permutation - key length is $\lceil log_2 N! \rceil$ bits to indicate one of $N!$ permutations.

- Attack: size of keyspace

- Attack: frequency-based attack
  Sorted frequency histogram of ciphertext should match well the the expected sorted frequency histogram of plaintext

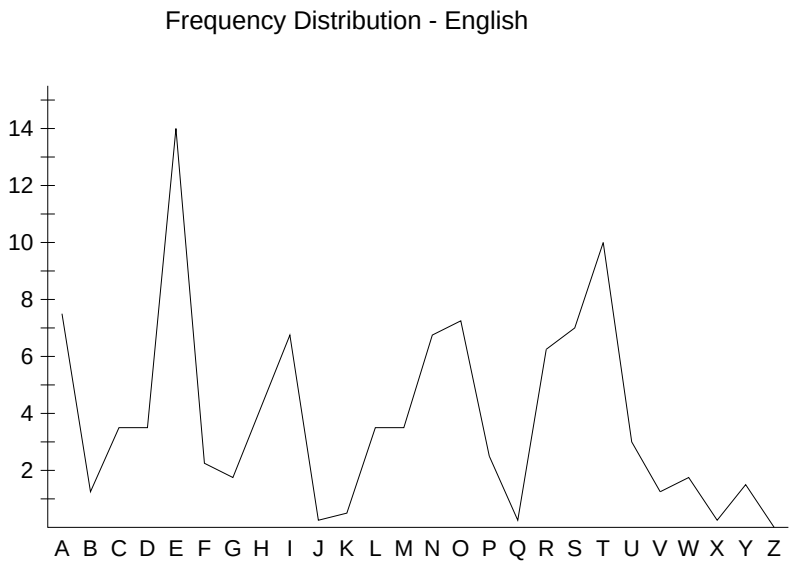

Figure 3: Relative frequency of characters in English text

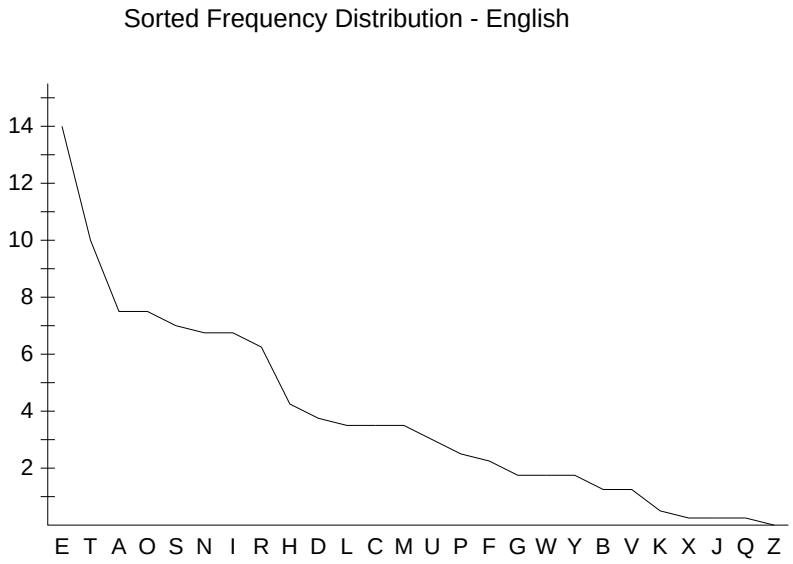

Figure 4: Sorted relative frequency of characters in English text

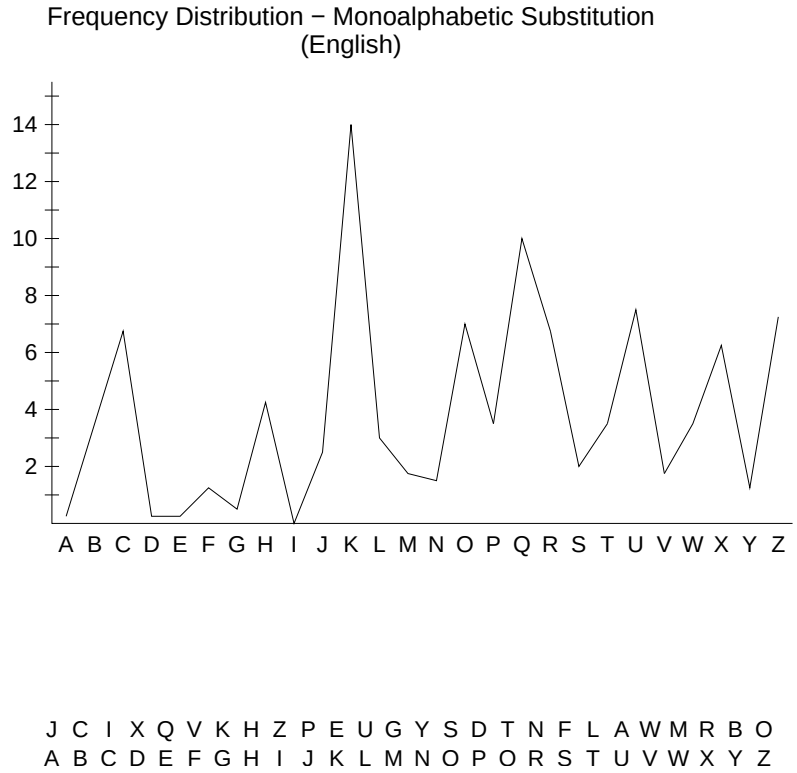

Figure 5: Relative frequencies of characters in MSC English

4.7 polyaphabetic cipher

flattens frequency histogram

- Viginere tableau
- Encryption keys
  - Single word/phrase repeated (GOGATORSGOGATORSGOGA...)
  - Word/phrase with
    - duplicate characters removed (GOATRSGOATRSGOA...)
    - rest of alphabet following in order (GOATRSBCDEFHIJKLMNPQUVWXYZGOATRSBCD...)
- Attack: index of coincidence
  The IOC is a measure of variance in the frequency histogram.

  Let $r(x)$ be the relative frequency of x in the data,

  $$r(x) = f(x)/n,$$

  where $f(x)$ is the frequency with which x appears in the n characters of the text. If $n = 26$ (for symbols from the Roman alphabet), and encryption makes the distribution perfectly smooth, each character x would have

  $$r(x) = 1/26 \approx 0.0384$$

Polyalphabetic Substitution

Plaintext:	**GOGAT ORSBE ATNOL ES**
ptxt - numerical:	**61601 11114 01111 41** **4 9 478 9341 8**
Key:	**SECSE CSECS ECSEC SE**
key - numerical:	**14214 21421 42142 14** **8 8 8 8 8 8**
ptxt + key:	**21812 13232 42311 22** **48 83 652 2 1183 22**
ctxt = ptxt + key modulo 26:	**21812 19232 42511 22** **48 83 6 2 2 1 83 22**
Ciphertext:	**YSISX QJWDW EVFSN WW**

Figure 6: Example of a polyalphabetic cipher

The nonuniformity of distribution for a single symbol x is just the difference between its observed relative frequency and the average, $1/n$ (in this case, $1/26$), or

$$d(x) = r(x) - 1/26.$$

Since these have positive and negative values (for peaks and valleys, resp.) summing these over all symbols would amount to zero - not very useful. (IOC con't)

Squaring these nonuniformities makes them all positive, and the greater the variation, the larger the resulting number, so one measure of variation is

$$Var = \Sigma_{x \in A}(r(x) - 1/|A|)^2 = \Sigma_{x \in A}(r(x))^2 - 1/|A|$$

(This really does work - try it.)

If x is chosen at random as the first character, then in the remaining text, its relative frequency (and probability that x will again be chosen as the second character, since the first one selected is no longer available) is

$$(f(x) - 1)/(n - 1),$$

so the probability of picking two different characters out of the text and having both of them be x is

$$f(x)(f(x) - 1)/n(n - 1)$$

The Index of Coincidence is just the sum of these probabilities over all symbols,

$$IC = \Sigma_{x \in A} f(x)(f(x) - 1)/n(n - 1).$$

Another way of looking at it is that the variation measure above has a useless constant term, $1/26$, that is always subtracted from the interesting part. IC just eliminates that constant term and specifically accounts for the difference that a finite amount of text makes in the computation.

IC varies from .068 for 26-symbol English prose enciphered with a monoalphabetic substitution, to .038 for encipherment with a large number of alphabets.

Its usefulness declines as the number of alphabets used for encipherment increases, as the curve becomes pretty flat after about 4 alphabets.

However, it can be used to gauge whether a small number of alphabets were used, and to test a subtext to see if it may have been enciphered with a monoalphabetic substitution.

- Attack: guessing number of ciphers (in round robin) This approach can use the SFH or the IC approach to test a series of hypotheses regarding the number of alphabets used (or the periodicity) to encrypt the plaintext.
- Attack: Kasiski attack - uses repeated polygrams (multiple symbol sequences) in the ciphertext to guide guessing of the period of a periodic encipherment. The assumption is that two repeated texts will be the result of two identical plaintext polygrams that were enciphered using the same sequence of keys, so distances between these pairs should be a multiple of the period of the cipher.

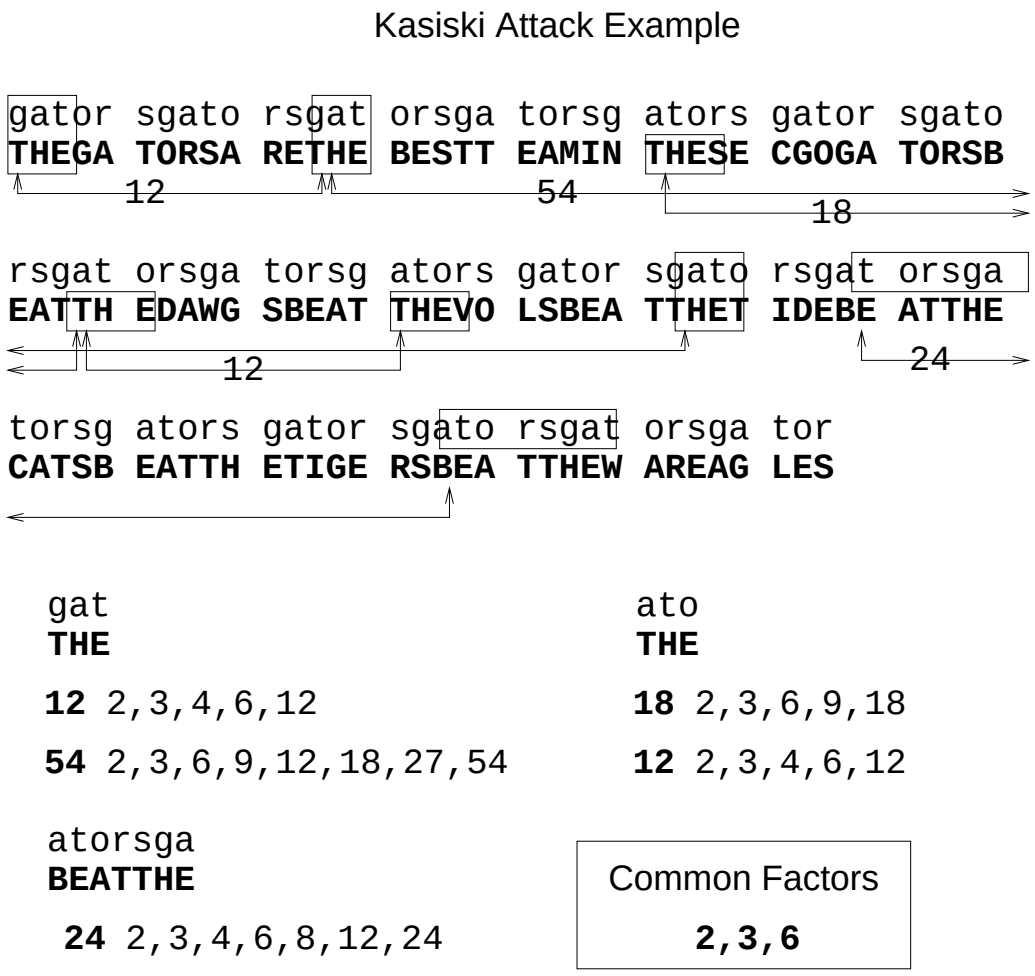

Figure 7: Example Kasiski Attack

General methodology for polyalphabetic substitution

1. Use IC to estimate number of alphabets; if more than a few

2. Use Kasiski to get candiate key lengths,
3. Based on the candidate key lengths, separate the ciphertext into l pieces (c_i goes into piece i *modulo* l)
4. Take the IC for each piece to verify that only one alphabet was used to encrypt it, or try another candidate length
5. Use sorted frequency histogram to break monoalphabetic substition in each piece

4.8 One-time pad

- Stream cipher
- Truly secure (confidentiality - iff used only once)
- Issue: Random number generation
- Vernam cipher
- MDSR (multidimensional spatial rotation - Casio)
- General OTP - any invertible binary function may be used
  - $\oplus$ for bits
  - $+$ modulo N
  - $\times$ modulo N if N is prime
- Use block cipher as PRNG to make stream cipher

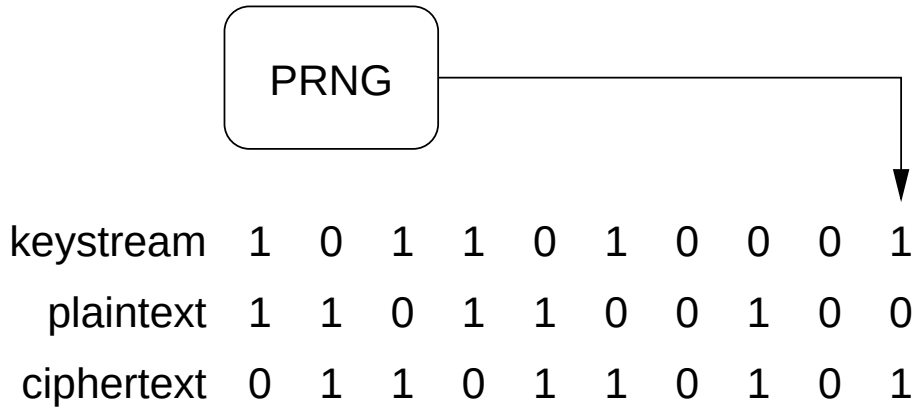

Figure 8: Example of a Vernam cipher

4.9 Transposition ciphers

- Columnar Transpositions
- General Transpositions

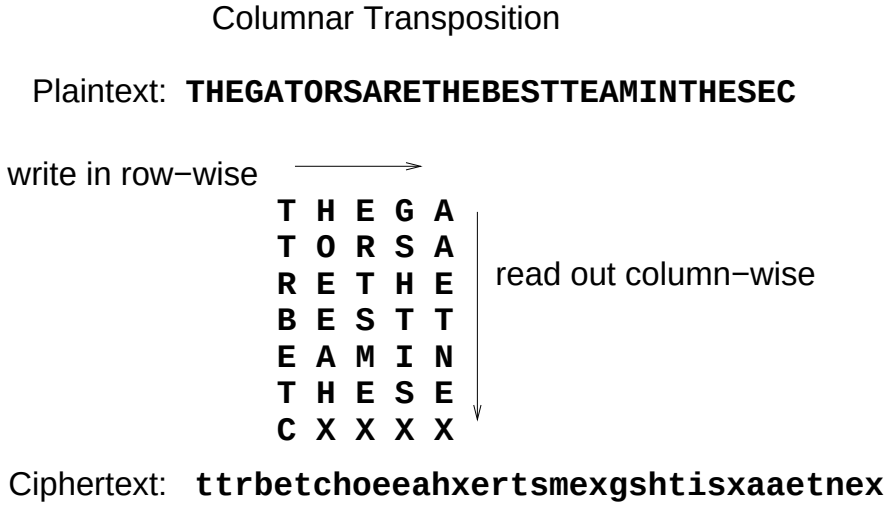

Figure 9: Columnar transposition cipher

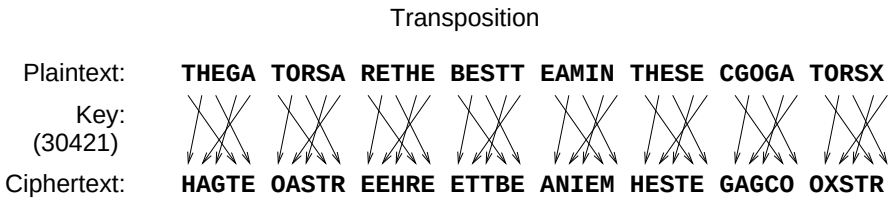

Figure 10: Example of a transposition cipher

4.10 Product cipher (concatenated cipher)

- Can increase period
- Can increase confusion and diffusion

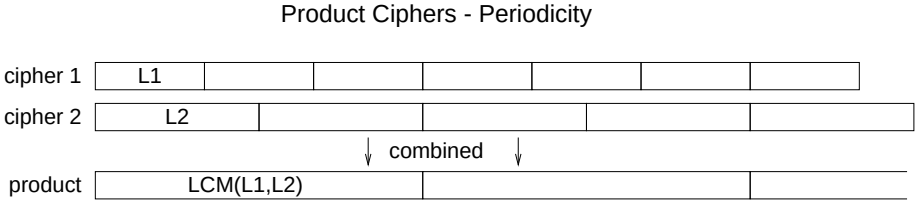

Figure 11: Product cipher period expansion

5 Theory

5.1 Shannon Characteristics

1. Secrecy required should determine level of effort
2. Keys and algorithm should be low in complexity - easy to generate good keys and apply them
3. Simple implementation
4. Errors should not propagate
5. Size of ciphertext should be no larger than plaintext

5.2 Redundancy

Absolute rate of a language is

$$A = \lceil lg(N) \rceil,$$

where N = size of alphabet, $\lceil \rceil$ = ceiling.

If the number of meaningful n-letter messages is 2^{Rn}, then the rate of the language is $R \leq A$, and the redundancy of the language is

$$D = A - R,$$

i.e., D is the number of extra bits per symbol used.

5.3 Information theoretic security

let $h(C)$ be the set of possible plaintexts for ciphertext C. An encryption is effectively secure if

$$Prob(h(C) = P) < \epsilon$$

for some arbitrarily small ϵ

Dual message entrapment - ϵ is never $> 1/2$

Ideally, knowing that the ciphertext is some particular C_1 should not narrow down the possible plaintexts,

$$Prob(h(C_1) = P|C_1) = Prob(h(C_1) = P) = Prob(P)$$

5.4 Unicity Distance

Unicity distance is a measure of the amount of ciphertext needed to break a cipher. Let

$$H(P|C) = \Sigma_{all\ P} Prob(P|C) lg(1/Prob(P|C))$$

Unicity distance is the length n of the smallest message for which $H(P|C)$ is close to 0.

With $2^{H(P)}$ keys, a cryptosystem has unicity distance

$$N = H(P)/D$$

Probability of spurious decryption is $p = (1 - q)$, where

$$q = 2^{n(R-A)} = 2^{-Dn}$$