Please refer this paper as http://www.r-ef.com/research/publications/codecs_short.pdf

Block cyphers, keys, and abelian integer transformations

Pál Ruján

R-EF Research

Rujan Entwicklung und Forschung GmbH

Wintererstr. 47a, D-79104,

Freiburg, Germany *

(Dated: 27 Feb. 2015)

Abstract

We show that a block cypher operating on N-bit block can have at most $2^N - 1$ *distinct* keys, such that for any $n \in [0, 2^N - 1]$ the cypher texts generated by the key n are different from each other in at least one bit. We introduce a simple constructive method for parametrizing all distinct keys.

* research@r-ef.com

I. THE DISTINCT KEYS THEOREM

Consider a compact interval of non-negative integers $t \in \mathcal{N}$ with M elements. Without loss of generality we can shift this interval into the $[0, \ldots, M-1]$. Let us call t a plain text value. Using an integer key k, it is possible to encrypt t into another integer, c, $c \xleftarrow{\text{key}} t$. c is called the cypher text [1] and is assumed that there exists a decrypt operation such that $t \xleftarrow{\text{key}} c$. In most cryptographic systems the encrypt/decrypt operation is a bit-by-bit XOR operation. By "keys" we understand here the parameters of the encrypt/decrypt operators.

Let denote the encryption operation as $c = \hat{E}_k(t)$ and the decryption operation by $t = \hat{D}_k(c)$ for a given key k. The encrypt/decrypt operations have the following properties for any key k:

$$\hat{D}_k \hat{E}_k = \hat{E}_k \hat{D}_k = \hat{1}, \ \forall k \tag{1}$$

$$c = \hat{E}_k(t) \text{ are different for different } t \tag{2}$$

Note that as t runs over the interval $[0 \ldots M-1]$ so does c but following a different path. The two paths do not intersect: $c(t) \neq t, \ \forall t$.

We define two keys, $k_1 \neq k_2$ to be *distinct* iff

$$c_1 = \hat{E}_{k_1}(t) \neq c_2 = \hat{E}_{k_2}(t); \quad \forall t \in [0 \ldots M-1] \tag{3}$$

Two distinct keys will cover the full $[0 \ldots M-1]$ interval so that their respective paths, as parametrized by t, never intersect.

A set of distinct keys is a set of keys where all keys are pairwise distinct. The distinct key theorem says that **there can be at most $M-1$ distinct keys**. Proof: each distinct key k_i must generate according to (3) a distinct value $c_i = \hat{E}_{k_i}(t)$. There are at most M such values, so any further key will map t into an already existing value c_i and cannot be part of the distinct set. Taking into account that we also excluded the unity transformation via 2, one has $M-1$ distinct keys left.

Therefore, the number of distinct (encrypt:decrypt) operators can be parametrized by a number of keys equal to $M-1$. In practice, the encrypt/decrypt codecs are deployed in pairs, one codec per key or device. It is easy to generalize this result for *almost* distinct p-intersection keys, where the paths of keys in the set are allowed to intersect at most p-times. Following the same arguments as above, the maximal number of almost p-distinct keys is $M + p - 1$. In practical applications, when the keys are generated *i.e.e.*, the probability

that two keys generated paths intersect is $(\frac{p}{M})^2$. For $p << M$ the probability of failing the compatibility test is negligible.

II. ABELIAN PARAMETRIZATION OF DISTINCT KEYS

Integers can have different *representations*. The keys are related to the parametrization inherent in these representations. For instance, the compact interval $[0, \dots M-1]$ can be embedded into some high dimensional volume, where each dimension corresponds to a "digit" or figure. Consider first a N-dimensional rectangular grid of sides $\{q_1 \dots q_N\}$. Its volume equals $V = q_1 \times q_2 \times \cdots \times q_N$, such that $V \geq M$. Next, let us map uniquely any integer $t \in [0 \dots M-1]$ into a grid vertex of this volume with co-ordinates $x_1, x_2 \dots x_N$, where $x_k \in [0 \dots q_k - 1]$.

In the usual number representations all sides of the grid are equal, so that the grid is a hypercube with side $q_i = q$ for $i = 1 \dots N$. Depending on the size $q = 2, 10, 16, ..$ the representation is called binary, decimal, hexadecimal, etc. We will call the case when the q_i might have different values depending on i a "mixed number representation".

The integers are mapped into a grid vertex by a linear scan along the sides $x_1, x_2, \dots x_N$, leading to $t = x_1 + x_2 \times q_1 + x_3 \times q_1 \times q_2 + ...$ This is not the only possible choice of visiting all grid points in N-dimensions. To distinguish between different realizations, call such a scan "encoding path". Therefore, the parametrization of keys will depend also on the set of parameters defined by the encoding path and the specific form of the used volume V.

In order to express in a simple form the encrypt/decrypt operations we will extend here the notion of a digit by associating a full $Z(q_i)$ group variable to it, so that $Z(t) = Z(q_1) \bigotimes Z(q_2) \bigotimes \cdots \bigotimes Z(q_N)$. In other words, the $i-th$ digit looks like a clock's main arm, jumping around a circle with q_i markings only.

In this interpretation the **key** K is an integer number whose coordinates are the $k_1, k_2 \dots k_N$ shifts. The encrypt operation corresponds to a right cyclic shift performed on each co-ordinate:

$$c = [(x_1 + k_1) \mod q_1] + [(x_2 + k_2) \mod q_2] \times q_1 + ... \tag{4}$$

while the decrypt operation is the corresponding left cyclic shift. The effect of shift 1 is to move the origin one marking to the right, of 2 to move the origin 2 markings to the right,

etc. Hence, if one disregards the unity operation $K = 0$, the total number of such right shifts is $M - 1$ and all the requirements of Eqs.(1 - 3) are fulfilled. Hence, this form of direct Abelian $Z(q_i)$ group products is representative for all cypher systems with distinct keys. The XOR operation is a special case of the cyclic shifts discussed above for the case when all digits are in binary representation.

Note that encrypting the number 0 (zero) will output a cypher block equal to the key K. That suggests that anyone able to access the (hardwired) encrypter will discover its key. However, if one adds to the key the encoding path, encrypting 0 returns only some permutation of the different digits of K. If the numbers of digits N is large enough, reconstructing the original key should be of order N^N. If an encrypter falls in bad hands and can be used for tests, a binary search approach might be used to find the volume sides and then the key shifts. In fact, only non-local, strongly distributed encrypt transformation can provide a format secure against attacks based on local plain text input changes.

III. CONCLUSIONS

We have shown that the number of all possible encryp-decrypt operations with distinct keys can be at most of the size of the cypher block. Given a block of 128 bits, there are $2^{128} - 1$ different keys such that the cypher texts generated by these keys are all different, for an arbitrary value of the text block. A simple constructive realization in terms of cyclic shifts can be easily implemented in hardware.

Using the Abelian representation of distinct keys provides a simple but universal way of encoding with shift-like keys. The keys are not secured against local text changes attacks. To make such an attack against the device codec more difficult one should use additional methods [1], like block-chaining or self-synchronizing stream ciphers. Furthermore, building additional error-correcting codes into the encoder/decoder can be used for both error-correcting - or in case of massive failures - for checking that the encoder used in the actual application is the the one corresponding to the used decoder. We call this test the **compatibility** test. It helps keep distinct the many applications which might use the same technology (DVD, matrix codes, etc.). Hence, a given DVD, marked product, transaction, or matrix code can be encoded with its own keys and only those decoders possessing the keys can decode it correctly. Such hybrid codes are used in visual reference tags, like eg.

the [2].

---

[1] B. Schneier: *Applied Cryptography*, John Wiley and Sons, Second Edition, 1996

[2] P. Rujan: *Producing, Capturing, and Using Visual Identification Tags for Moving Objects*, provisional patent application dated 10.08.2010 US 13/206977, EP 11757789.0
`http://www.r-ef.com/research/publications/reftags_patent.pdf`