Functional Description of SIGABA

Abstract

The SIGABA is an electromechanical encryption device used by the US during WWII and in the 1950s. Also known as ECM Mark II, Converter M-134, as well as CSP-888/889, the SIGABA was considered highly secure, and was employed for strategic communications, such as between Churchill and Roosevelt. The SIGABA encrypts and decrypts with a set of five rotors, and implements irregular stepping, with two additional sets of rotors generating a pseudo-random stepping sequence. Its full keyspace, as used during WWII, was in the order of $2^{95.6}$. It is believed that the German codebreaking services were not able to make any inroads into the cryptanalysis of SIGABA (Mucklow, 2015; Budiansky, 2000; Kelley, 2001).

1 The SIGABA Encryption Machine

In this section, a short functional description of the SIGABA is given, as well as an analysis of its keyspace size. There were two models, CSP-889 and CSP-2900. First, CSP-889 is described, and after that, the differences between CSP2900 and CSP-889 are listed. A complete description of the machine and its history may be found in the references (Savard and Pekelney, 1999; Sullivan, 2002b; Stamp and Chan, 2007; Mucklow, 2015; Kelley, 2001; Pekelney, 1998; Sullivan, 2002a).

1.1 Functional Description of SIGABA CSP-889

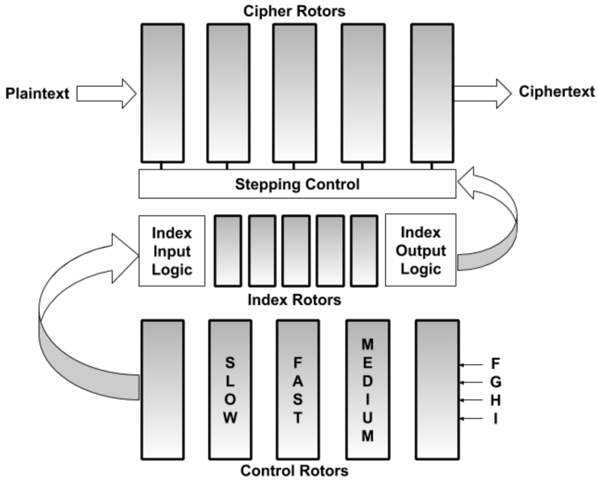


Figure 1: SIGABA – Functional Diagram

The SIGABA encryption and decryption mechanism consists of three banks of five rotors each, the *cipher* bank, the *control* bank, and the *index* bank, as depicted in Figure 1. Each rotor of the *cipher* bank has 26 inputs and 26 outputs (similar in concept to the Enigma rotors, but with different wirings). The *cipher* rotors implement encryption (from left to right), and decryption (from right to left). The rotors of the *cipher* bank step according to an irregular pseudo-random pattern generated by the *index* and the *control* rotor bank.

The *control* bank consists of five rotors, each with 26 inputs and 26 outputs. The *cipher* rotors and the *control* rotors are interchangeable, and are selected from a set of ten rotors. Furthermore, those rotors can be installed in two possible orientations – forward or reverse (thus increasing the size of the keyspace by a factor of $2^{10} = 1,024$). Interestingly, the *cipher* rotors and the *control* rotors move through the alphabet in reverse order (e.g., from D to C, or from C to B) when installed in forward orientation, and in alphabetical order (e.g., from D to E, or from E to F) when installed in reversed orientation. The leftmost and rightmost *control* rotors are stationary and do not rotate. The *fast* rotor always steps (interestingly, this rotor is located between the *slow* and the *medium* rotors). If the *fast* rotor steps from O to N (while in forward orientation) or from O to P (while in reversed orientation), the *medium* rotor also steps

(Pekelney, 1998; Sullivan, 2002a).[1] Similarly, if the *medium* rotor steps from O to N (while in forward orientation) or from O to P (while in reversed orientation), the *slow* rotor also steps. At each encryption step, the inputs F, G, H, and I of the rightmost (stationary) rotor are activated and fed with electrical current (and the 22 remaining input are always inactive). The 26 outputs of the leftmost *control* rotor enter the *index input logic*, described in Figure 2.

Input (to index rotor bank)	Logic (A-Z are the outputs of the control rotor bank, $\vee$ indicates logical OR)
Input 1	(inactive)
Input 2	B
Input 3	C
Input 4	$D \vee E$
Input 5	$F \vee G \vee H$
Input 6	$I \vee J \vee K$
Input 7	$L \vee M \vee N \vee O$
Input 8	$P \vee Q \vee R \vee S \vee T$
Input 9	$U \vee V \vee W \vee X \vee Y \vee Z$
Input 10	A

Figure 2: Index Input Logic – CSP-889

Stepping Control (of cipher rotor)	Logic (I_1-I_{10} are the outputs of the index rotor bank, $\vee$ indicates logical OR)
Rotor 1	$I_1 \vee I_{10}$
Rotor 2	$I_8 \vee I_9$
Rotor 3	$I_6 \vee I_7$
Rotor 4	$I_4 \vee I_5$
Rotor 5	$I_2 \vee I_3$

Figure 3: Index Output Logic

The *index* bank consists of a set of five stationary rotors, which do not rotate during encryption or decryption, and they each have 10 inputs and 10 outputs. Those rotors are not interchangeable with the *cipher* and *control* rotors, and they can only be installed in a forward orientation. The *index input logic*, described in Figure 2, maps its 26 inputs into 10 outputs, which enter the leftmost *index* rotor. The *index output logic*, described in Figure 3, maps the 10 outputs of the *index* rightmost rotor into five *stepping control* signals, controlling the stepping of the five *cipher* rotors. The design of the *control* and *index* rotor banks, in conjunction with the *index input logic* and the *index output logic*, ensures that at least one of the five *cipher* rotors will step, but no more than four *cipher* rotors ever step (Stamp and Chan, 2007, p. 203).

Encryption is performed as follows, assuming that the 15 rotors have been installed. The machine must be set to the encryption mode. The operator selects the starting positions of the rotors, and types the plaintext on the SIGABA keyboard. The plaintext symbol is applied to the *cipher* rotors from left to right, producing the ciphertext symbol on a printing device. After encryption of a symbol, the *cipher* rotors step according to the state of the *stepping control* (see Figure 1). After the *cipher* rotors have stepped, some of the *control* rotors step, thus generating (via the *index* rotors) a new state for the *stepping control* of the *cipher* rotors. The process is repeated for the next plaintext symbols.

Decryption works similarly, except that the device must be set to the decryption mode, and the cipher symbols (typed on the keyboard) are applied to the *cipher* rotors from right to left, the resulting plaintext being printed.

1.2 Model CSP-2900

The differences between SIGABA model CSP-2900 and model CSP-889 are listed here:

1. With model CSP-889, there were four active inputs to the *control* rotor bank (F, G, H, and I). With model CSP-2900, there are instead six active inputs to the *control* rotor bank (D, E, F, G, H, and I).

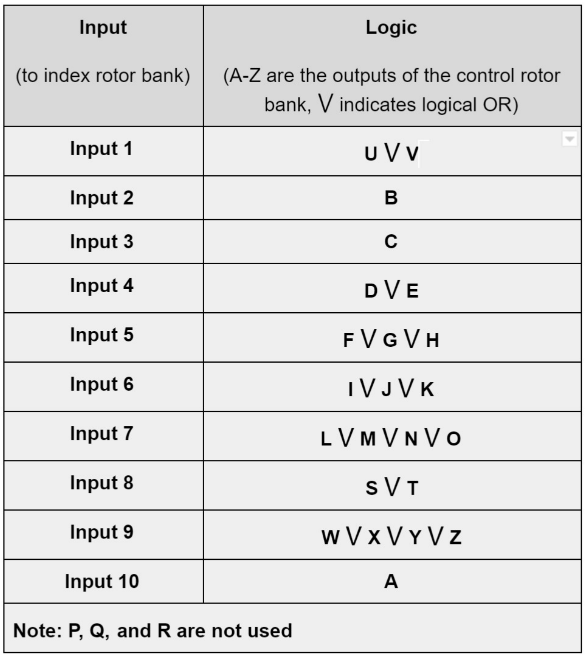
2. The logic of the input to the *index* rotor bank is different, as shown in Figure 4. The 10 inputs are connected (with CSP-889, Input 1 was always inactive). Also, the outputs P, Q, and R of the *control* rotors are not used.

3. As a result, up to six out of the 10 outputs of the *index* rotor bank might be active at each step. Since those are ORed by pairs by the *index output logic* (unchanged for CSP-2900),

[1] (Stamp and Chan, 2007; Savard and Pekelney, 1999) describe different implementations for the stepping mechanism.

this means that it is possible that all five *cipher* rotors step (instead of no more than four with CSP-889). As with model CSP-889, at least one *cipher* rotor steps.

4. While with model CSP-889, the five *cipher* rotors, while in forward orientation, step so that the displayed letters progress in inversed alphabetical order (and in alphabetical order while in reverse orientation), with model CSP-2900 *cipher* rotors 2 and 4 step in reversed direction, so that the displayed letters progress in alphabetical order while in forward orientation (and in inverse alphabetical order while in forward orientation). The other *cipher* rotors step in the same direction as the *cipher* rotors of model CSP-889.



Input (to index rotor bank)	**Logic** (A-Z are the outputs of the control rotor bank, ⋁ indicates logical OR)
Input 1	**U ⋁ V**
Input 2	**B**
Input 3	**C**
Input 4	**D ⋁ E**
Input 5	**F ⋁ G ⋁ H**
Input 6	**I ⋁ J ⋁ K**
Input 7	**L ⋁ M ⋁ N ⋁ O**
Input 8	**S ⋁ T**
Input 9	**W ⋁ X ⋁ Y ⋁ Z**
Input 10	**A**
Note: P, Q, and R are not used	

Figure 4: Index Input Logic – CSP-2900

1.3 Analysis of the Keyspace

Assuming that there is a set of ten rotors from which the *cipher* and *control* rotors are selected, there are 10! possible selections for those rotors. Each one of those rotors may be installed in either a forward or reverse orientation. The size of the keyspace for the settings of the ten rotors of the *cipher* and *control* banks is therefore $10! \cdot 2^{10} \cdot 26^{10} = 2^{78.8}$.

There are 5! possible ordering of the *index* rotors. The size of the keyspace for the settings of the rotors of the *index* bank is therefore $5! \cdot 10^5 = 2^{23.5}$. The combined size of the SIGABA keyspace is $2^{78.8+23.5} = 2^{102.3}$.

However, the size of the keyspace for the *index* bank is limited by the fact that the five rotors implement a (stationary) permutation of the ten inputs and the ten outputs are mapped by the *index output logic* into only five outputs. Therefore, the size of the practical keyspace for the *index* rotors including the *index output logic* is only $10!/2^5 = 113,400 = 2^{16.8}$, and the combined size of the practical keyspace of SIGABA is as follows:

$$2^{78.8+16.8} = 2^{95.6} \tag{1}$$

For comparison, the size of the keyspace the German Enigma I was in the order of 2^{77} (Stamp and Low, 2007, p. 31), and it is 2^{56} for the more recent DES.

References

Stephen Budiansky. 2000. *Battle of wits: the complete story of codebreaking in World War II*. Simon and Schuster.

Stephen J. Kelley. 2001. Big Machines: Cipher Machines of World War II.

Timothy Jones Mucklow. 2015. *The SIGABA/ECM II Cipher Machine: "a Beautiful Idea"*. National Security Agency, Center for Cryptologic History.

Richard S. Pekelney. 1998. ECMApp - Emulation of ECM Mark II. `https://maritime.org/tech/ecmapp.txt`, [Accessed: January, 18th, 2019].

John J. G. Savard and Richard S. Pekelney. 1999. The ECM Mark II: Design, History, and Cryptology. *Cryptologia*, 23(3):211–228.

Mark Stamp and Wing On Chan. 2007. SIGABA: Cryptanalysis of the Full Keyspace. *Cryptologia*, 31(3):201–222.

Mark Stamp and Richard M. Low. 2007. *Applied Cryptanalysis: Breaking Ciphers in the Real World*. John Wiley & Sons.

Geoff Sullivan. 2002a. CSG Sigaba (ECM Mark II) Simulator for Windows. `http://cryptocellar.org/simula/sigaba/index.html`, [Accessed: January, 18th, 2019].

Geoff Sullivan. 2002b. The ECM Mark II: some observations on the rotor stepping. *Cryptologia*, 26(2):97–100.