IMT4124 Cryptology

Siegenthaler's correlation attack

Individual project task 2:

Siegenthaler's correlation attack is a cryptanalytic ciphertext-only attack against stream ciphers using non-linear combiners and was first presented in the paper (Siegenthaler, 1985). This report will present what the attack is and how it could be conducted, with the help of an example.

Enciphering a plaintext with the use of a stream cipher is commonly done by using the XOR-operation bitwise between the plaintext and the secret keystream. One of the requirements for this to be considered secure is that the secret keystream is random, or at least appears to be so. Creating long sequences of true random numbers is too computationally demanding for use in practical cryptographic applications, so the solution is often to create pseudo-random sequences which fulfills Golomb's postulates. These sequences can be created by Linear Feedback Shift Registers (LFSRa). These registers are given an initial state and a feedback polynomial which enables the registers to create pseudo-random sequences (often called PN-sequences) of sufficient length. It is important to keep the initial state away from a cryptanalyst, as the initial state can be used to recreate the secret keystream in its entirety.

A downside of this scheme is that LFSRs are linear and are thus vulnerable to algebraic cryptanalytic attacks. To avoid this, it is common to add non-linearity by non-linear combiners which combines the results from several LFSRs in a non-linear manner to create the key stream. One such combination is found in Geffe's generator, as shown in (Siegenthaler, 1985). However, some PN-sequences created using non-linear combiners are suspect to correlation attacks and Geffe's generator is one of them.

There are many other non-linear combiners that are suspect to Siegenthaler's correlational attack, (all where the output from one of the LFSRs correlates with the output from the non-linear combiner), but these will not be of focus in this report. Attacks on these other non-linear combiners can nevertheless be conducted in roughly the same way.

A correlation attack is possible if one or more of the LFSRs' output correlates with the output of the non-linear combiner. This is best shown by inspecting the truth table as seen below:

x_1	x_2	x_3	Output from Geffe's generator (y)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

The three x-tabs represent the output bit from each LFSR respectively, and the final tab (y) displays the final output after the output bits from the three LFSRs have gone through the non-linear combiner from Geffe's generator. The Boolean function of the non-linear combiner is $x_1 \neg x_2 + x_2 x_3$, as seen in (Siegenthaler, 1985). All calculations in this context is performed modulo 2.

By counting the number of times the different LFSRs correlates with the output from Geffe's generator, you can find the correlation by calculating the probability that the output bit from each LFSR matches the output bit from the non-linear combiner.

In this case, you can see that:

$$P(x_1 = y) = \frac{6}{8} = 0.75$$

$$P(x_2 = y) = \frac{4}{8} = 0.5$$

$$P(x_3 = y) = \frac{6}{8} = 0.75$$

We want to find LFSRs with correlation different from 0.5, the further away the better. In this case, you can see that the output of the first and the third register have a correlation of 0.75 with the output from Geffe's generator. This means that the Boolean function in Geffe's generator is not correlation immune (of order 1), which makes a correlation attack possible.

The motivation behind correlation attacks is that when guessing the initial state (which is what we want to obtain), you do not have to brute-force the initial states for all the registers at the same time, but you can isolate the attempts for the LFSRs with correlation other than 0.5. In this case, it means that the first or the third can be attacked by themselves instead of breaking all three at the same time. This reduces the complexity of the attack tremendously.

In broad terms, the attack consists of determining a threshold T, which a correlation measure α is compared against for each guessed initial state. The correlation measure α measures the correlation between the intercepted ciphertext and the output from the attacked LFSR. Based on how α compares to T, we can determine whether the tested state could be considered a candidate initial state of that LFSR.

The best way to display the attack is by the means of an example, following the procedure as described in (Siegenthaler, 1985):

Assume that want to perform attack on LFSR 1 in Geffe's generator (x_1 the table mentioned previously). We know that a correlation attack is possible because $P(x_1 = y) = \frac{6}{8} = 0.75$, as stated before. Another assumption we have to make is that the plaintext source must not be random. This holds for most meaningful text written in natural languages and is thus not a problem for us in this context.

We then define the following parameters:

$$q_1 = 0.75$$

$$p_0 = 0.65$$

$$L = 1200$$

Where q_1 is the correlation between the output from the LFSR and the output from the non-linear combiner, p_0 is the probability that a symbol from the plaintext source is 0 (The value of 0.65 is chosen arbitrarily as all values except 0.5 are acceptable.) and L is the length of the intercepted ciphertext.

We can then calculate:

$$p_e = 1 - (p_0 + q_1) + 2p_0q_1 = 1 - (0.65 + 0.75) + 2 \cdot 0.65 \cdot 0.75 = 0.575$$

Where p_e is the probability that the sum of the intercepted ciphertext and the LFSR output is 0. For a correlation attack to be possible this cannot be 0.5, because then the two Gaussian distributions (soon to be defined) will be the same, which makes it impossible to determine which distribution the guessed initial state belongs to. p_e will only be 0.5 if either p_0 or q_i is 0.5, which confirms the two limitations mentioned before.

We then create two hypotheses:

H_0: The intercepted sequence was not generated from the given initial state.

H_1: The intercepted sequence was generated from the initial given state.

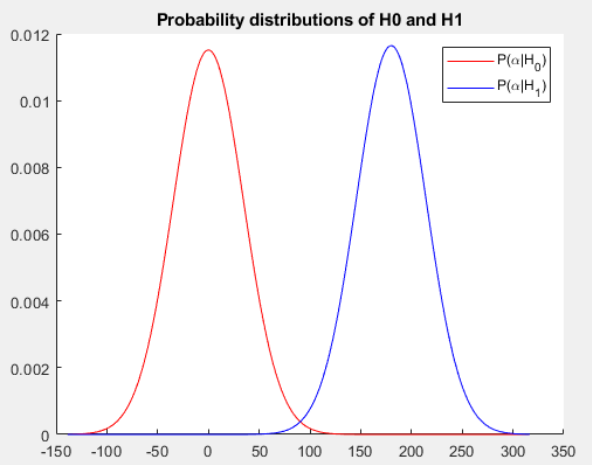
These hypotheses are given their own Gaussian distributions with the following parameters:

$$P(\alpha|H_0): \mu_0 = 0, \sigma_0 = \sqrt{L} = \sqrt{1200} = 34.64$$

$$P(\alpha|H_1): \mu_1 = L(2p_e - 1) = 1200(2 \cdot 0.575 - 1) = 180,$$
$$\sigma_1 = 2\sqrt{L}\sqrt{p_e(1-p_e)} = 2 \cdot 34.64 \cdot \sqrt{0.575 \cdot 0.425} = 34.25$$

Where α is the correlation measure between the intercepted sequence and the LFSR output generated by the tested initial state. Based on the value of α, you can thus determine which probability distribution it most likely belongs to.

In the distributions' parameters, you can see that the distance between the distributions' mean is 180, with standard deviations around 34 for both. This means that the two distributions have a relatively small overlap, which should make it easy to place the correlation measure α for the guessed initial state in the correct distribution. Visually, the distributions would look like this:



The next thing we need to do is to calculate the threshold T, which is used to separate the two distributions. The threshold will therefore be placed somewhere between the two distributions.

To do that, we need to first set our target value for probability of false positives (p_f) or the probability of missing the event (p_m). These values represent the probability that a guessed initial state is placed in the wrong distribution. In this example, they would be defined as follows:

$p_f = P(\alpha \geq T|H_0)$, $p_m = P(\alpha < T|H_1)$

In this example, we will assume that we want to achieve p_m=0.05 (one could also determine T by using p_f). This implies that there is a 5% chance that the correct initial state will falsely be considered to not be a candidate.

The following formula defines p_m:

$$p_m = Q\left(\left|\frac{L(2p_e - 1) - T}{2\sqrt{L}\sqrt{p_e(1-p_e)}}\right|\right)$$

Where

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-y^2/2}dy$$

Q(x) is known as the formula for the cumulative standard normal distribution, which means that we could look up in its table directly rather than calculating the formula.

As we defined p_m=0.05, we need to find 1-0.05=0.95 in the cumulative standard normal distribution table, which is the z-value 1.65. To find our threshold, we then need to solve the following equation for T:

$$\left|\frac{L(2p_e - 1) - T}{2\sqrt{L}\sqrt{p_e(1-p_e)}}\right| = 1.65$$

$$\left|\frac{1200(2\cdot 0.575 - 1) - T}{2\sqrt{1200}\sqrt{0.575(1-0.575)}}\right| = 1.65$$

$$\left|\frac{180 - T}{34.25}\right| = 1.65$$

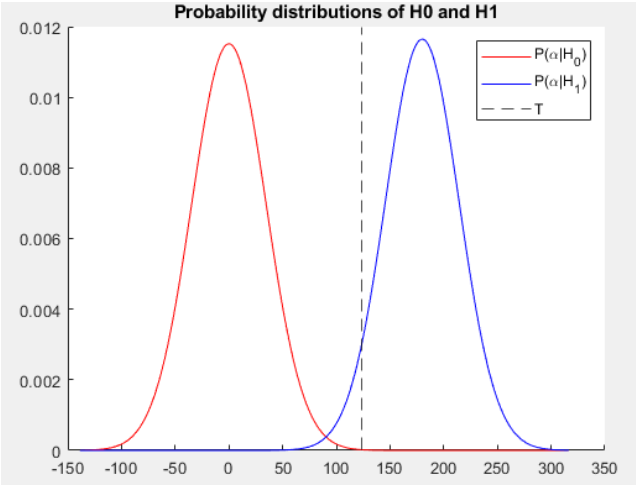
$$T = 123.49 \; and \; T = 236.51$$

We then obtain that the threshold T=123.49 (T=236.51 cannot be used as this value is not placed between our two distributions and would therefore be a poor separator), which we can use to easily find p_f:

$$p_f = 1 - Q\left(\left|\frac{T}{\sqrt{L}}\right|\right) = 1 - Q\left(\left|\frac{123.49}{34.64}\right|\right) = 1 - Q(3.56) = 1 - 0.999815 = 0.000185$$

The probability that a guessed initial state results in a false alarm is thus 0.0185%.

Visually, the distributions and threshold look like this:



The next step is then to obtain the correlation measure α with the following formula:

$$\alpha = L - 2\sum_{n=1}^{L}(C_n \oplus X_n^1)$$

Where C is the intercepted ciphertext and X^1 is the output from the LFSR (with our guessed initial state) we try to attack, in our case LFSR 1.

The final stage of the attack involves determining which of the probability distributions, as defined earlier, our correlation measure belongs to, and thus which hypothesis to accept. Thanks to our threshold T, as seen in the figure above, the evaluation is trivial and goes like this:

> If α≥T, we accept the hypothesis H_1 and the tested initial state is a candidate for being the correct one. The probability that is was falsely selected as a candidate is expressed in p_f.
>
> If α<T, we accept the hypothesis H_0, and the tested initial state is not a candidate for being the correct one. The probability that it was falsely selected as not a candidate is expressed in p_m.

The correlation measure α needs to be calculated for each possible initial state for the selected LFSR, which might sound like much work, but as mentioned earlier in the paper, the alternative is worse.

As an example, assume we obtained α=145 with the initial state β=10011011010 for an 11-bit LFSR. Recall that we obtained t=123.49. This would imply that the initial state β would be a candidate for being the correct initial state, since 145≥123.49.

After this operation is done on all possible initial states, the result should be a sufficiently small list of candidates where hopefully one of them would be the correct one. This could be determined by further testing.

This concludes the discussion of what the Siegenthaler's correlation attack is and how it can be performed.

References:

Siegenthaler, T. (1985) Decrypting a class of stream ciphers using ciphertext only, *IEEE Transactions on computers,* (1), pp. 81-85