An example to demonstrate the correctness of the proposed algorithm for the secret value discovery attack against the AnonR^{2}AS protocol

Authors: Mahsa Ghasemi and Masoumeh Safkhani

Introduction

This appendix provides a detailed illustrative example of the secret value discovery attack against the AnonR^{2}AS protocol. The purpose of this example is to demonstrate the correctness of the proposed algorithm, showing step-by-step computations of secret values such as (k_i) and (TID), and verifying the algorithm's effectiveness in recovering these values from observed protocol messages.

Example / Illustrative Case Study

For the described algorithm and the explained steps, the following example can demonstrate the correctness of the proposed method.
For instance, according to Equations (1) and (2), we assume that the following values are the secret parameters in this protocol, and we aim to eventually recover them using the proposed method.

$$\text{TID}=1000000000100000 \tag{1}$$

$$\mathrm{k_i} = 1101000100001000 \tag{2}$$

Moreover, additional values, including (IDS), (n_3), (n_2), and (n_1), can be eavesdropped from the communication channel; therefore, according to Equation 3, we have:

$$\text{IDS}=1000100000000101 \tag{3}$$

Given that the above values can be eavesdropped from the communication channel, we assume that the adversary has collected these values in the form of a table through eavesdropping, as shown in Table 1. In this table, according to equation B, we have Y=n_2 and X=$f_h(IDS,n_1 \oplus k_i)$. Since k_i is unknown, the Hamming weight of X is also unknown; therefore, it is denoted by "?".

Table 1. Value of (B) for different values of wt(Y)

wt (X)	wt (Y)	section	value
?	0	n_2	0000000000000000
		B	0111011000000100
?	1	n_2	1000000000000000
		B	0011101100000101
?	2	n_2	1100000000000000
		B	1101110110000100
?	3	n_2	1110000000000000

		B	0111111011000100
?	4	n_2	1111000000000000
		B	0011101101100100
?	5	n_2	1111100000000000
		B	0001111110110010
?	6	n_2	1111110000000000
		B	1100110111011111
?	7	n_2	1111111000000000
		B	1111110011101001
?	8	n_2	1111111100000000
		B	1111111101110010
?	9	n_2	1111111110000000
		B	0011111110111100
?	10	n_2	1111111111000000
		B	0100111111011010
?	11	n_2	1111111111100000
		B	0011001111101001
?	12	n_2	1111111111110000
		B	1001110011110000
?	13	n_2	1111111111111000
		B	0100101000111100
?	14	n_2	1111111111111100
		B	0010011111001010
?	15	n_2	1111111111111110
		B	0001000111111101
?	16	n_2	1111111111111111
		B	1000101011111000

Now, the attacker selects one row from Table 1. For instance, the row corresponding to wt(Y) = 3 is considered. Since the value of wt(X) is unknown, the attacker constructs Table 2 based on the parametric table available at the same GitHub link, for wt(Y) = 3 and wt(X) ranging from 0 to 16, as shown below.

It should be noted that, as indicated in Table 1, for Hamming weight 3, according to Equation (4), we have:

$$n_2 = 1110000000000000 \tag{4}$$

After constructing all the rows, as shown in Table 2, some bits are represented as fixed numeric values, some as parametric values, and some as XOR combinations of numeric values with specific parameters. The attacker then independently compares the numeric bits obtained in each row with the actual bit values of (B) given in Table 1, which are equal to (B = 0111111011000100).

If any contradiction is observed during this comparison, the corresponding Hamming weight of (X) is eliminated. As can be seen, the rows whose numeric bit values are consistent with the original value of (B) are considered as candidates.

It should be noted that the rows of Table 2 can be generated either manually or automatically by using the Python code provided in Reference [13] of the paper, by appropriately setting the parameters (X) and (Y) for each row.

Table 2. Parametric computation of (B) for (wt(Y)=3) and different values of (wt(X))

wt (X)	wt (Y)	section	value	status
0	3	B	x0⊕0 x1⊕0 x2⊕0 x3⊕x8 x4⊕x9 x5⊕xa x6⊕xb x7⊕xc 0⊕xd 0⊕xe 0⊕xf 1 1 1 0 0	Mismatch – Rejected
1	3	B	0 xf⊕0 x0⊕yc x1⊕x7 x2⊕x8 x3⊕x9 x4⊕xa x5⊕xb x6⊕xc 0⊕xd 0⊕xe 0 1 1 1 0	Mismatch – Rejected
2	3	B	0 0 xe⊕0 xf⊕x6 x0⊕x7 x1⊕x8 x2⊕x9 x3⊕xa x4⊕xb x5⊕xc 0⊕xd 0 0 1 1 1	Mismatch – Rejected
3	3	B	1 0 0 xd⊕x5 xe⊕x6 xf⊕x7 x0⊕x8 x1⊕x9 x2⊕xa x3⊕xb x4⊕xc 0 0 0 1 1	Mismatch – Rejected
4	3	B	1 1 0 0⊕x4 xc⊕x5 xd⊕x6 x0⊕x9 x1⊕xa xe⊕x7 xf⊕x8 x2⊕xb x3⊕0 0 0 0 1	Mismatch – Rejected
5	3	B	1 1 1 0⊕x3 0⊕x4 xb⊕x5 xc⊕x6 xd⊕x7 xe⊕x8 xf⊕x9 x0⊕xa x1⊕y5 x2⊕0 0 0 0	Mismatch – Rejected
6	3	B	0 1 1 1⊕x2 0⊕x3 0⊕x4 xa⊕x5 xb⊕x6 xc⊕x7 xd⊕x8 xe⊕x9 xf⊕0 x0⊕0 x1⊕0 0 0	Mismatch – Rejected
7	3	B	0 0 1 1⊕x1 1⊕x2 0⊕x3 0⊕x4 x9⊕x5 xa⊕x6 xb⊕x7 xc⊕x8 xd⊕0 xe⊕0 xf⊕0 x0⊕0 0	Mismatch – Rejected
8	3	B	0 0 0 1⊕x0 1⊕x1 1⊕x2 0⊕x3 0⊕x4 x8⊕x5 x9⊕x6 xa⊕x7 xb⊕0 xc⊕0 xd⊕0 xe⊕0 xf⊕0	Mismatch – Rejected

9	3	B	xe⊕0 0 0 0⊕xf 1⊕x0 1⊕x1 1⊕x2 0⊕x3 0⊕x4 x7⊕x5 x8⊕x6 x9⊕0 xa⊕0 xb⊕0 xc⊕0 xd⊕0	Mismatch – Rejected
10	3	B	xc⊕0 xd⊕0 0 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1⊕x2 0⊕x3 0⊕x4 x6⊕x5 x7⊕0 x8⊕0 x9⊕0 xa⊕0 xb⊕0	Mismatch – Rejected
11	3	B	xa⊕0 xb⊕0 xc⊕0 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1⊕x2 0⊕x3 0⊕x4 x5⊕0 x6⊕0 x7⊕0 x8⊕0 x9⊕0	Match – Candidate
12	3	B	x8⊕0 x9⊕0 xa⊕0 xb⊕xc 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1⊕x2 0⊕x3 0 x4⊕0 x5⊕0 x6⊕0 x7⊕0	Match – Candidate
13	3	B	x6⊕0 x7⊕0 x8⊕0 x9⊕xb xa⊕xc 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1⊕x2 0 0 x3⊕0 x4⊕0 x5⊕0	Match – Candidate
14	3	B	x4⊕0 x5⊕0 x6⊕0 x7⊕0 x8⊕0 x9⊕0 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1 0 0 x2⊕0 x3⊕0	Mismatch – Rejected
15	3	B	x2⊕0 x3⊕0 x4⊕0 x5⊕x9 x6⊕xa x7⊕xb x8⊕xc 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1 1 0 0 x1⊕0	Mismatch – Rejected
16	3	B	x0⊕0 x1⊕0 x2⊕0 x3⊕x8 x4⊕x9 x5⊕xa x6⊕xb x7⊕xc 0⊕xd 0⊕xe 0⊕xf 1 1 1 0 0	Mismatch – Rejected

To examine the candidates, for each Hamming weight X under consideration, different weights Y are taken into account, and the corresponding value B is calculated. This value is then compared with the original B for that Hamming weight. The process continues until there is no discrepancy in the bits. If any bits are found that do not match the original bits, the corresponding candidate is discarded.

Since there are four candidates, each of the four is examined using this method until the original Hamming weight X is determined.

Table 3. Parametric computation of B for the selected candidates to eliminate invalid candidates.

wt (X)	wt (Y)	section	value	status
11	4	B	xa⊕0 xb⊕0 xc⊕0 0 0⊕xd 1⊕x1 0⊕xe 0⊕xf 1⊕x0	Mismatch – Rejected

			1⊕x2 1⊕x3 x5⊕x4 x6⊕0 x7⊕0 x8⊕0 x9⊕0	
12	5	B	x8⊕0 x9⊕0 xa⊕0 xb⊕0 0 0⊕xc 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1⊕x2 x4⊕x3 x5⊕1 x6⊕1 x7⊕0	Mismatch – Rejected
13	4	B	x6⊕0 x7⊕0 x8⊕0 x9⊕0 xa⊕xb 0⊕xc 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1⊕x2 1 x3⊕0 x4⊕0 x5⊕0	Mismatch – Rejected

According to Table 3, since Hamming weights 11, 12, and 13 were obtained for (Y), which can be used to verify their invalidity, (wt(X) = 6) is deemed acceptable. For further verification, a table (Table 4) can be constructed to calculate the corresponding (B) values for Hamming weights (Y) ranging from 0 to 16.

Table 4. Parametric computation of B for wt(X)= 6 across different wt(Y) values.

wt (X)	wt (Y)	section	value	Comparison with Original B
6	0	B	0⊕x2 0⊕x3 0⊕x4 0⊕x5 0⊕x6 0⊕x7 xa⊕x8 xb⊕x9 xc⊕y8 xd⊕0 xe⊕0 xf⊕0 x0⊕0 x1⊕0 0 0	Match
6	1	B	0 0⊕x2 0⊕x3 0⊕x4 0⊕x5 0⊕x6 xa⊕x7 xb⊕x8 xc⊕x9 xd⊕0 xe⊕0 xf⊕0 x0⊕0 x1⊕0 0 1	Match
6	2	B	1 1 0⊕x2 0⊕x3 0⊕x4 0⊕x5 xa⊕x6 xb⊕x7 xc⊕x8 xd⊕x9 xe⊕0 xf⊕0 x0⊕0 x1⊕0 0 0	Match
6	3	B	0 1 1 1⊕x2 0⊕x3 0⊕x4 xa⊕x5 xb⊕x6 xc⊕x7 xd⊕x8 xe⊕x9 xf⊕0 x0⊕0 x1⊕0 0 0	Match
6	4	B	0 0 1 1 1⊕x2 1⊕x3 xa⊕x4 xb⊕x5 xc⊕x6 xd⊕x7 xe⊕x8 xf⊕x9 x0⊕y4 x1⊕0 0 0	Match
6	5	B	0 0 0 1 1 1⊕x2 xa⊕x3 xb⊕x4 xc⊕x5 xd⊕x6 xe⊕x7 xf⊕x8 x0⊕x9 x1⊕1 1 0	Match

6	6	B	1 1 0 0 1 1 xa⊕x2 xb⊕x3 xc⊕x4 xd⊕x5 xe⊕x6 xf⊕x7 x0⊕x8 x1⊕x9 1 1	Match
6	7	B	1 1 1 1 1 1 xa⊕y8 xb⊕x2 xc⊕x3 xd⊕x4 xe⊕x5 xf⊕x6 x0⊕x7 x1⊕x8 0⊕x9 1	Match
6	8	B	1 1 1 1 1 1 xa⊕1 xb⊕1 xc⊕x2 xd⊕x3 xe⊕x4 xf⊕x5 x0⊕x6 x1⊕x7 0⊕x8 0⊕x9	Match
6	9	B	0⊕x2 0 1 1 1 0⊕y3 xa⊕1 xb⊕1 xc⊕1 xd⊕x2 xe⊕x3 xf⊕x4 x0⊕x5 x1⊕x6 1⊕x7 1⊕x8	Match
6	10	B	1⊕x8 1⊕x9 0 0 1 1 xa⊕1 xb⊕1 xc⊕1 xd⊕1 xe⊕x2 xf⊕x3 x0⊕x4 x1⊕x5 1⊕x6 1⊕x7	Match
6	11	B	1⊕x7 1⊕x8 1⊕x9 1 0 0 xa⊕1 xb⊕1 xc⊕1 xd⊕1 xe⊕1 xf⊕x2 x0⊕x3 x1⊕x4 1⊕x5 1⊕x6	Match
6	12	B	1⊕x6 1⊕x7 1⊕x8 1⊕x9 1 1 xa⊕0 xb⊕0 xc⊕1 xd⊕1 xe⊕1 xf⊕1 x0⊕x2 x1⊕x3 1⊕x4 1⊕x5	Match
6	13	B	1⊕x5 1⊕x6 1⊕x7 1⊕x8 1⊕x9 0 xa⊕0 xb⊕0 xc⊕ye xd⊕yf xe⊕1 xf⊕1 x0⊕1 x1⊕x2 1⊕x3 1⊕x4	Match
6	14	B	1⊕x4 1⊕x5 1⊕x6 1⊕x7 1⊕x8 1⊕x9 xa⊕1 xb⊕1 xc⊕1 xd⊕1 xe⊕0 xf⊕0 x0⊕1 x1⊕1 1⊕x2 1⊕x3	Match
6	15	B	1⊕x3 1⊕x4 1⊕x5 1⊕x6 1⊕x7 1⊕x8 xa⊕x9 xb⊕1 xc⊕1 xd⊕1 xe⊕1 xf⊕1 x0⊕1 x1⊕1 0 1⊕x2	Match
6	16	B	1⊕x2 1⊕x3 1⊕x4 1⊕x5 1⊕x6 1⊕x7 xa⊕x8 xb⊕x9 xc⊕y8 xd⊕1 xe⊕1 xf⊕1 x0⊕1 x1⊕1 0 0	Match

Based on the calculations in Table 4 and the numerical values of the bits obtained, as well as their correspondence with the original (B) values, it can be concluded that (wt(X) = 6).

Therefore, the bits of (X), which correspond to the expression $X=f_h(\mathrm{IDS},n_1 \oplus k_i)$, can be computed as follows: for each bit resulting from the XOR of a number and a parameter, its value can be determined by equating it with the corresponding original value provided in Table 1. In this way, the value of the parameter, which is one of the bits of (X), can be obtained, and the remaining bits can be determined in a similar manner.

In the case where (wt(X) = 6) and (wt(Y) = 3), according to Equation 5, we have:

$$\begin{array}{ll} 1 \oplus x2 = 1 \Rightarrow x2 = 0 & xf \oplus 0 = 0 \Rightarrow xf = 0 \\ 0 \oplus x3 = 1 \Rightarrow x3 = 1 & x0 \oplus 0 = 0 \Rightarrow x0 = 0 \\ 0 \oplus x4 = 1 \Rightarrow x4 = 1 & x1 \oplus 0 = 0 \Rightarrow x1 = 0 \end{array} \tag{5}$$

In the case where (wt(X) = 6) and (wt(Y) = 5), according to Equation 6, we have:

$$\begin{array}{l} 1 \oplus x5 = 0 \Rightarrow x5 = 1 \\ 1 \oplus x6 = 1 \Rightarrow x6 = 0 \\ 1 \oplus x7 = 0 \Rightarrow x7 = 1 \\ 1 \oplus x8 = 0 \Rightarrow x8 = 1 \end{array} \tag{6}$$

In the case where (wt(X) = 6) and (wt(Y) = 13), according to Equation 7, we have:

$$\begin{array}{ll} xa \oplus 1 = 1 \Rightarrow xa = 0 & xd \oplus 0 = 0 \Rightarrow xd = 0 \\ xb \oplus 0 = 0 \Rightarrow xb = 0 & xe \oplus 1 = 1 \Rightarrow xe = 0 \\ xc \oplus 0 = 0 \Rightarrow x1 = 0 & xf \oplus 1 = 1 \Rightarrow xf = 0 \end{array} \tag{7}$$

As a result, based on the calculations performed, all bits of (X) have been determined. Therefore, according to Equation 8, we have:

$$X=f_h(IDS,n_1 \oplus k_i)=0101110110000000 \tag{8}$$

According to the relation for (B), now that the value of($f_h(\mathrm{IDS},n_1 \oplus k_i)$)has been obtained, ($k_i$) can be derived as follows. Since both (IDS) and (X) are known, and (n_1) can take different values, the corresponding row in the parametric table is examined based on the Hamming weight of (IDS). Then, for (wt(IDS) = 4) and various Hamming weights of ($n_1 \oplus k_i$), the corresponding parametric table is constructed as shown below, allowing some bits of ($n_1 \oplus k_i$) to be determined.

For the remaining bits, another value of (n_1) is considered, and the corresponding (B) values for the new (n_2) are calculated. Using the same method as before, $f_h(\mathrm{IDS},n_1 \oplus k_i)$is recalculated, and additional bits of ($\mathrm{f_h}(\mathrm{IDS},\mathrm{n}_1 \oplus \mathrm{k_i})$) are recovered. This process is repeated iteratively until all bits of this expression are discovered.

Therefore, by assuming ($\mathrm{Y}=\mathrm{n}_1 \oplus \mathrm{k_i}$) and (X = IDS), Table 5 is constructed.

Table 5. Parametric computation of ($f_h(IDS,n_1 \oplus k_i)$) for (wt(X) = 4) and various (wt(Y)) values.

wt (X)	wt (Y)	$f_h(IDS,n_1 \oplus k_i)$	Comparison with $f_h(IDS,n_1 \oplus k_i)$
4	0	y4⊕1 y5⊕0 y6⊕0 y7⊕0 0 1 0 1 1⊕y8 0⊕y9 0⊕ya 0⊕yb y0⊕yc y1⊕yd y2⊕ye y3⊕yf	Mismatch – Rejected
4	1	y3⊕ye y4⊕1 y5⊕0 y6⊕0 0 1 0 1 1 0⊕y7 0⊕y8 0⊕y9 yf⊕ya y0⊕yb y1⊕yc y2⊕yd	Mismatch – Rejected
4	2	y2⊕yc y3⊕yd y4⊕1 y5⊕0 0 1 0 1 1 0 0⊕y6 0⊕y7 ye⊕y8 yf⊕y9 y0⊕ya y1⊕yb	Mismatch – Rejected
4	3	y1⊕ya y2⊕yb y3⊕yc y4⊕1 0 1 0 1 1 0 0 0⊕y5 yd⊕y6 ye⊕y7 yf⊕y8 y0⊕y9	Mismatch – Rejected
4	4	y0⊕y8 y1⊕y9 y2⊕ya y3⊕yb 1 1 0 1 1 0 0 0 yc⊕y4 yd⊕y5 ye⊕y6 yf⊕y7	Match – Candidate
4	5	yf⊕y6 y0⊕y7 y1⊕y8 y2⊕y9 0⊕ya 0 0 1 1 0 0 0 yb⊕0 yc⊕y3 yd⊕y4 ye⊕y5	Mismatch – Rejected
4	6	ye⊕y4 yf⊕y5 y0⊕y6 y1⊕y7 0⊕y8 1⊕y9 1 1 1 0 0 0 ya⊕0 yb⊕0 yc⊕y2 yd⊕y3	Mismatch – Rejected
4	7	yd⊕y2 ye⊕y3 yf⊕y4 y0⊕y5 0⊕y6 1⊕y7 0⊕y8 0 1 0 0 0 y9⊕0 ya⊕0 yb⊕0 yc⊕y1	Mismatch – Rejected
4	8	yc⊕y0 yd⊕y1 ye⊕y2 yf⊕y3 0⊕y4 1⊕y5 0⊕y6 1⊕y7 0 0 0 0 y8⊕0 y9⊕0 ya⊕0 yb⊕0	Mismatch – Rejected
4	9	yb⊕0 yc⊕yf yd⊕y0 ye⊕y1 0⊕y2 1⊕y3 0⊕y4 1⊕y5 1⊕y6 1 0 0 y7⊕0 y8⊕0 y9⊕0 ya⊕0	Mismatch – Rejected
4	10	ya⊕0 yb⊕0 yc⊕ye yd⊕yf 0⊕y0 1⊕y1 0⊕y2 1⊕y3	Mismatch – Rejected

		1⊕y4 0⊕y5 1 0 y6⊕0 y7⊕0 y8⊕0 y9⊕0	
4	11	y9⊕0 ya⊕0 yb⊕0 yc⊕yd 0⊕ye 1⊕yf 0⊕y0 1⊕y1 1⊕y2 0⊕y3 0⊕y4 1 y5⊕0 y6⊕0 y7⊕0 y8⊕0	Mismatch – Rejected
4	12	y8⊕0 y9⊕0 ya⊕0 yb⊕0 0⊕yc 1⊕yd 0⊕ye 1⊕yf 1⊕y0 0⊕y1 0⊕y2 0⊕y3 y4⊕1 y5⊕0 y6⊕0 y7⊕0	Match – Candidate
4	13	y7⊕0 y8⊕0 y9⊕0 ya⊕0 0 1⊕yb 0⊕yc 1⊕yd 1⊕ye 0⊕yf 0⊕y0 0⊕y1 y3⊕y2 y4⊕1 y5⊕0 y6⊕0	Mismatch – Rejected
4	14	y6⊕0 y7⊕0 y8⊕0 y9⊕0 0 1 1⊕yc 0⊕yd 0⊕ya 1⊕yb 0⊕ye 0⊕yf y2⊕y0 y3⊕y1 y4⊕1 y5⊕0	Mismatch – Rejected
4	15	y5⊕0 y6⊕0 y7⊕0 y8⊕0 0 1 1⊕ya 0⊕yb 0⊕yc 0 1⊕y9 0⊕yd y1⊕ye y2⊕yf y3⊕y0 y4⊕1	Mismatch – Rejected
4	16	y4⊕1 y5⊕0 y6⊕0 y7⊕0 0 1 1⊕y8 0⊕y9 0⊕ya 0⊕yb 0 1 y0⊕yc y1⊕yd y2⊕ye y3⊕yf	Mismatch – Rejected

Currently, there are two candidates. The first candidate has (wt(IDS) = 4) and (wt($n_1 \oplus k_i$) =4), while the second candidate has (wt(IDS) = 4) and (wt($n_1 \oplus k_i$) =12).

The first candidate is discarded because the parametric bits generated are either numerical values or the XOR of two parameters; therefore, none of the bits of (Y) can be determined from the two parameters. Consequently, we proceed with the second candidate, and the bits of (Y) are obtained according to Equation 9:

$$\begin{array}{lll}
y8 \oplus 0 = 0 \ \Rightarrow y8 = 0 & 1 \oplus yd = 1 \ \Rightarrow yd = 0 & 0 \oplus y2 = 0 \ \Rightarrow y2 = 0 \\
y9 \oplus 0 = 1 \ \Rightarrow y9 = 1 & 0 \oplus ye = 0 \ \Rightarrow ye = 0 & 0 \oplus y3 = 0 \ \Rightarrow y3 = 0 \\
ya \oplus 0 = 0 \ \Rightarrow ya = 0 & 1 \oplus yf = 1 \ \Rightarrow yf = 0 & y4 \oplus 1 = 0 \ \Rightarrow y4 = 1 \\
yb \oplus 0 = 1 \ \Rightarrow yb = 1 & 1 \oplus y0 = 1 \ \Rightarrow y0 = 0 & y5 \oplus 0 = 0 \ \Rightarrow y5 = 0 \\
0 \oplus yc = 1 \ \Rightarrow yc = 0 & 0 \oplus y1 = 0 \ \Rightarrow y1 = 0 & y6 \oplus 0 = 0 \ \Rightarrow y6 = 0 \\
 & & y7 \oplus 0 = 0 \ \Rightarrow y7 = 0
\end{array} \qquad (9)$$

Therefore, according to Equation 10, we have:

$$Y = n_1 \oplus k_i = 0000100001010000 \qquad (10)$$

Considering that, in one of the eavesdropping sessions, the values of (n_1), (n_2), (n_3), and (D) were captured according to Equations 11 to 14, and the value of (IDS) is also known.

$$n_1=1001000000011001 \qquad (11)$$

$$n_2=1110000000000000 \qquad (12)$$

$$n_3=1110010001000000 \qquad (13)$$

$$D=0010101000101001 \qquad (14)$$

Now, the value of (k_i) can be derived based on the known value of (n_1) and ($Y = n_1 \oplus k_i$), as expressed in Equation 15:

$$k_i = 0100100001010000 \qquad (15)$$

Using the obtained value of (k_i), the value of (D) is then computed and compared with the original (D) value captured from the eavesdropped channel in order to verify the validity of the selected candidate.

According to Equation 16:

$$D = f_h(Rot_l(Rot_r(f_h(IDS,n_1),k_i),n_2),n_3) \qquad (16)$$

Since all parameters involved in the computation of (D) are known, the value of (D) can be calculated. Based on Equation 17, we obtain:

$$\text{(computed) } D=0010110001010001 \qquad (17)$$

However, the original value of (D) eavesdropped from the communication channel, according to Equation 18, is:

$$\text{(eavesdropped) } D=0010101000101001 \qquad (18)$$

As the computed value of (D) does not match the original eavesdropped value, it can be concluded that the recovered (k_i) is invalid, and therefore the selected candidate is incorrect. Moreover, since the bits of (Y) cannot be derived from the other candidate either, a different pair of (n_1) and (n_2) values is considered.

Given that, through eavesdropping, the value of (B) corresponding to a fixed (n_1) and multiple values of (n_2) can be obtained, the value of ($f_h(IDS, n_1 \oplus k_i)$) can be derived accordingly. As these values are observable over the communication channel, it is assumed that the attacker collects a table of eavesdropped values, as shown in Table 6. In this table, based on the relation for (B), we have (Y = n_2) and ($X = f_h(IDS, n_1 \oplus k_i)$). Since ($k_i$) is unknown, the Hamming weight of (X) is also unknown and is therefore denoted by “?”.

Table 6. Values of (B) for different (wt(Y)) values.

wt (X)	wt (Y)	section	value
?	0	n_2	0000000000000000
		B	0000101001110111
?	1	n_2	1000000000000000
		B	0100010101110111
?	2	n_2	1100000000000000
		B	0011001011110111
?	3	n_2	1110000000000000
		B	0001110100110111
?	4	n_2	1111000000000000
		B	0000111111010111
?	5	n_2	1111100000000000
		B	0000011100100001
?	6	n_2	1111110000000000
		B	1100001101011100
?	7	n_2	1111111000000000
		B	1111100101100010
?	8	n_2	1111111100000000
		B	1111111101111101
?	9	n_2	1111111110000000
		B	1111111111110010
?	10	n_2	1111111111000000
		B	0111111110110101
?	11	n_2	1111111111100000
		B	1011111110010110
?	12	n_2	1111111111110000
		B	0101111110000111
?	13	n_2	1111111111111000
		B	1010100101001111
?	14	n_2	1111111111111100
		B	1101010010111011
?	15	n_2	1111111111111110
		B	1110101010001101
?	16	n_2	1111111111111111
		B	1111010110001000

Next, one row of Table 6 is selected; for example, the row corresponding to (wt(Y) = 5). Since (wt(X)) is unknown, Table 7 is constructed from the parametric table for (wt(Y) = 5) and for (wt(X)) ranging from 0 to 16, as shown below.

It should be noted that, as indicated in Table 6, for Hamming weight 3, according to Equation 19, we have:

$$n_2 = 1111100000000000 \tag{19}$$

After generating all rows, as illustrated in Table 7, some bits are obtained as numerical values, some as parametric values, and others as the XOR of a numerical value and a specific parameter. The attacker then separately compares the numerical bits obtained in each row with the actual (B) value in Table 6, which is: B = 0000011100100001
If any inconsistency is observed in this comparison, the corresponding Hamming weight for (X) is discarded. As can be seen, the rows whose numerical bit values match the original (B) are considered as valid candidates.

Table 7. Parametric computation of (B) for (wt(Y) = 5) and different (wt(X)) values.

wt(X)	wt(Y)	section	value	status
0	5	B	x0⨁0 x1⨁0 x2⨁0 x3⨁0 x4⨁0 x5⨁x8 x6⨁x9 x7⨁xa 0⨁xb 0⨁xc 0⨁xd 0⨁xe 0⨁xf 0 0 1	Match – Candidate
1	5	B	1 xf⨁0 x0⨁0 x1⨁0 x2⨁0 x3⨁x7 x4⨁x8 x5⨁x9 x6⨁xa 0⨁xb 0⨁xc 0⨁xd 0⨁xe 1 0 1	Mismatch – Rejected
2	5	B	1 1 xe⨁0 xf⨁0 x0⨁0 x1⨁x6 x2⨁x7 x3⨁x8 x4⨁x9 x5⨁xa 0⨁xb 0⨁xc 0⨁xd 1 1 1	Mismatch – Rejected
3	5	B	1 1 1 xd⨁0 xe⨁0 xf⨁x5 x0⨁x6 x1⨁x7 x2⨁x8 x3⨁x9 x4⨁xa 0⨁xb 0⨁xc 1 1 0	Mismatch – Rejected
4	5	B	0 1 1 1 xc⨁0 xd⨁x4 xe⨁x5 xf⨁x6 x0⨁x7 x1⨁x8 x2⨁x9 x3⨁xa 0⨁xb 1 1 0	Mismatch – Rejected
5	5	B	0 0 1 1 1 xb⨁x3 xc⨁x4 xd⨁x5 xe⨁x6 xf⨁x7 x0⨁x8 x1⨁x9 x2⨁xa 1 1 0	Mismatch – Rejected
6	5	B	0 0 0 1 1 1⨁x2 xa⨁x3 xb⨁x4 xc⨁x5 xd⨁x6 xe⨁x7 xf⨁x8 x0⨁x9 x1⨁1 1 0	Mismatch – Rejected
7	5	B	0 0 0 0 1 1⨁x1 1⨁x2 x9⨁x3 xa⨁x4 xb⨁x5 xc⨁x6 xd⨁x7 xe⨁x8 xf⨁1 x0⨁1 0	Mismatch – Rejected
8	5	B	0 0 0 0 0 1⨁x0 1⨁x1 1⨁x2x8 ⨁x3 x9⨁x4 xa⨁x5 xb⨁x6 xc⨁x7 xd⨁1 xe⨁1 xf⨁0	Match – Candidate

9	5	B	xe⊕0 0 0 0 0 0⊕xf 1⊕x0 1⊕x1 1⊕x2 x7⊕x3 x8⊕x4 x9⊕x5 xa⊕x6 xb⊕1 xc⊕1 xd⊕0	Match – Candidate
10	5	B	xc⊕0 xd⊕0 0 0 0 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1⊕x2 x6⊕x3 x7⊕x4 x8⊕x5 x9⊕1 xa⊕1 xb⊕0	Match – Candidate
11	5	B	xa⊕0 xb⊕0 xc⊕0 0 0 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1⊕x2 x5⊕x3 x6⊕x4 x7⊕1 x8⊕1 x9⊕0	Match – Candidate
12	5	B	x8⊕0 x9⊕0 xa⊕0 xb⊕0 0 0⊕xc 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1⊕x2 x4⊕x3 x5⊕1 x6⊕1 x7⊕0	Match – Candidate
13	5	B	x6⊕0 x7⊕0 x8⊕0 x9⊕0 xa⊕0 0⊕xb 0⊕xc 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1⊕x1 1⊕x2 x3⊕1 x4⊕1 x5⊕0	Match – Candidate
14	5	B	x4⊕0 x5⊕0 x6⊕0 x7⊕0 x8⊕0 x9⊕xa 0⊕xb 0⊕xc 0⊕xd 0⊕xe 0⊕xf 1⊕x0 1⊕x1 0 x2⊕1 x3⊕0	Match – Candidate
15	5	B	x2⊕0 x3⊕0 x4⊕0 x5⊕0 x6⊕0 x7⊕x9 x8⊕xa 0⊕xb 0⊕xc 0⊕xd 0⊕xe 0⊕xf 1⊕x0 0 0 x1⊕0	Match – Candidate
16	5	B	x0⊕0 x1⊕0 x2⊕0 x3⊕0 x4⊕0 x5⊕x8 x6⊕x9 x7⊕xa 0⊕xb 0⊕xc 0⊕xd 0⊕xe 0⊕xf 0 0 1	Match – Candidate

Next, the candidates are examined. For this purpose, for each candidate Hamming weight of (X), different Hamming weights of (Y) are considered, and the corresponding value of (B) is computed and compared with the original (B) value associated with those Hamming weights. This process is repeated until no bit mismatches are observed. If any bits are found that do not match the original bits, the corresponding candidate is discarded.

Since there are ten candidates, all ten are evaluated using this method until the original Hamming weight of (X) is identified.

Table 8. Parametric computation of (B) for eliminating invalid candidates.

wt (X)	wt (Y)	section	value	status

0	10	B	'x0⊕xe', 'x1⊕xf', 'x2⊕0', 'x3⊕0', 'x4⊕1', 'x5⊕1', 'x6⊕1', 'x7⊕1', 0, 0, '1⊕x8', '1⊕x9', '0⊕xa', '0⊕xb', '0⊕xc', '0⊕xd'	Mismatch – Rejected
9	4	B	'xe⊕0', 0, 0, 0, '0⊕xf', '1⊕x0', '1⊕x1', '1⊕x2', '1⊕x3', 'x7⊕x4', 'x8⊕x5', 'x9⊕x6', 'xa⊕0', 'xb⊕0', 'xc⊕0', 'xd⊕0'	Mismatch – Rejected
10	9	B	'xc⊕x5', 'xd⊕0', 0, 0, 1, 1, 1, 1, 1, '0⊕xe', 'x6⊕xf', 'x7⊕x0', 'x8⊕x1', 'x9⊕x2', 'xa⊕x3', 'xb⊕x4'	Mismatch – Rejected
11	9	B	'xa⊕x4', 'xb⊕0', 'xc⊕1', 0, 0, 1, 1, 1, 1, '0⊕xd', '0⊕xe', 'x5⊕xf', 'x6⊕x0', 'x7⊕x1', 'x8⊕x2', 'x9⊕x3'	Mismatch – Rejected
12	9	B	'x8⊕x3', 'x9⊕0', 'xa⊕1', 'xb⊕1', 0, 0, 1, 1, 1, '0⊕xc', '0⊕xd', '0⊕xe', 'x4⊕xf', 'x5⊕x0', 'x6⊕x1', 'x7⊕x2'	Mismatch – Rejected
13	9	B	'x6⊕x2', 'x7⊕0', 'x8⊕1', 'x9⊕1', 'xa⊕1', 0, 0, 1, 1, '0⊕xb', '0⊕xc', '0⊕xd', '0⊕xe', 'x3⊕xf', 'x4⊕x0', 'x5⊕x1'	Mismatch – Rejected
14	9	B	'x4⊕x1', 'x5⊕0', 'x6⊕1', 'x7⊕1', 'x8⊕1', 'x9⊕1', 0, 0, 1, '0⊕xa', '0⊕xb', '0⊕xc', '0⊕xd', '0⊕xe', 'x2⊕xf', 'x3⊕x0'	Mismatch – Rejected
15	9	B	'x2⊕x0', 'x3⊕0', 'x4⊕1', 'x5⊕1', 'x6⊕1', 'x7⊕1', 'x8⊕1', 0, 0, '0⊕x9', '0⊕xa', '0⊕xb', '0⊕xc', '0⊕xd', '0⊕xe', 'x1⊕xf'	Mismatch – Rejected

16	9	B	'x0⊕xf', 'x1⊕0', 'x2⊕1', 'x3⊕1', 'x4⊕1', 'x5⊕1', 'x6⊕1', 'x7⊕1', 0, '1⊕x8', '0⊕x9', '0⊕xa', '0⊕xb', '0⊕xc', '0⊕xd', '0⊕xe'	Mismatch – Rejected

Therefore, since for the candidates listed in Table 8 there exist Hamming weights of (Y) that can be used to demonstrate their invalidity, (wt(X) = 8) is considered acceptable. To provide further verification, Table 9 is constructed by considering different Hamming weights of (Y) ranging from 0 to 16.

Table 9. Parametric computation of (B) for (wt(X) = 9) and different (wt(Y)) values.

wt(X)	wt(Y)	section	value	status
8	0	B	'0⊕x0', '0⊕x1', '0⊕x2', '0⊕x3', '0⊕x4', '0⊕x5', '0⊕x6', '0⊕x7', 'x8⊕0', 'x9⊕0', 'xa⊕0', 'xb⊕0', 'xc⊕0', 'xd⊕0', 'xe⊕0', 'xf⊕0'	Match
8	1	B	0, '1⊕x0', '0⊕x1', '0⊕x2', '0⊕x3', '0⊕x4', '0⊕x5', '0⊕x6', 'x8⊕x7', 'x9⊕0', 'xa⊕0', 'xb⊕0', 'xc⊕0', 'xd⊕0', 'xe⊕0', 'xf⊕0'	Match
8	2	B	0, 0, '1⊕x0', '1⊕x1', '0⊕x2', '0⊕x3', '0⊕x4', '0⊕x5', 'x8⊕x6', 'x9⊕x7', 'xa⊕0', 'xb⊕0', 'xc⊕0', 'xd⊕0', 'xe⊕0', 'xf⊕0'	Match
8	3	B	0, 0, 0, '1⊕x0', '1⊕x1', '1⊕x2', '0⊕x3', '0⊕x4', 'x8⊕x5', 'x9⊕x6', 'xa⊕x7', 'xb⊕0', 'xc⊕0', 'xd⊕0', 'xe⊕0', 'xf⊕0'	Match
8	4	B	0, 0, 0, 0, '1⊕x0', '1⊕x1', '1⊕x2', '1⊕x3', 'x8⊕x4', 'x9⊕x5', 'xa⊕x6', 'xb⊕x7', 'xc⊕0', 'xd⊕0', 'xe⊕0', 'xf⊕0'	Match
8	5	B	0, 0, 0, 0, 0, '1⊕x0', '1⊕x1', '1⊕x2', 'x8⊕x3', 'x9⊕x4', 'xa⊕x5', 'xb⊕x6', 'xc⊕x7', 'xd⊕1', 'xe⊕1', 'xf⊕0'	Match
8	6	B	1, 1, 0, 0, 0, 0, '1⊕x0', '1⊕x1', 'x8⊕x2', 'x9⊕x3', 'xa⊕x4', 'xb⊕x5', 'xc⊕x6', 'xd⊕x7', 'xe⊕1', 'xf⊕1'	Match

8	7	B	1, 1, 1, 1, 1, 0, 0, '1⊕x0', 'x8⊕x1', 'x9⊕x2', 'xa⊕x3', 'xb⊕x4', 'xc⊕x5', 'xd⊕x6', 'xe⊕x7', 'xf⊕1'	Match
8	8	B	1, 1, 1, 1, 1, 1, 1, 1, 'x8⊕x0', 'x9⊕x1', 'xa⊕x2', 'xb⊕x3', 'xc⊕x4', 'xd⊕x5', 'xe⊕x6', 'xf⊕x7'	Match
8	9	B	'1⊕x7', 1, 1, 1, 1, 1, 1, 1, 'x8⊕1', 'x9⊕x0', 'xa⊕x1', 'xb⊕x2', 'xc⊕x3', 'xd⊕x4', 'xe⊕x5', 'xf⊕x6'	Match
8	10	B	'1⊕x6', '1⊕x7', 1, 1, 1, 1, 1, 1, 'x8⊕1', 'x9⊕1', 'xa⊕x0', 'xb⊕x1', 'xc⊕x2', 'xd⊕x3', 'xe⊕x4', 'xf⊕x5'	Match
8	11	B	'1⊕x5', '1⊕x6', '1⊕x7', 1, 1, 1, 1, 1, 'x8⊕1', 'x9⊕1', 'xa⊕1', 'xb⊕x0', 'xc⊕x1', 'xd⊕x2', 'xe⊕x3', 'xf⊕x4'	Match
8	12	B	'1⊕x4', '1⊕x5', '1⊕x6', '1⊕x7', 1, 1, 1, 1, 'x8⊕1', 'x9⊕1', 'xa⊕1', 'xb⊕1', 'xc⊕x0', 'xd⊕x1', 'xe⊕x2', 'xf⊕x3'	Match
8	13	B	'1⊕x3', '1⊕x4', '1⊕x5', '1⊕x6', '1⊕x7', 0, 0, 1, 'x8⊕0', 'x9⊕0', 'xa⊕1', 'xb⊕1', 'xc⊕1', 'xd⊕x0', 'xe⊕x1', 'xf⊕x2'	Match
8	14	B	'1⊕x2', '1⊕x3', '1⊕x4', '1⊕x5', '1⊕x6', '1⊕x7', 0, 0, 'x8⊕1', 'x9⊕1', 'xa⊕0', 'xb⊕0', 'xc⊕1', 'xd⊕1', 'xe⊕x0', 'xf⊕x1'	Match
8	15	B	'1⊕x1', '1⊕x2', '1⊕x3', '1⊕x4', '1⊕x5', '1⊕x6', '1⊕x7', 0, 'x8⊕1', 'x9⊕1', 'xa⊕1', 'xb⊕1', 'xc⊕1', 'xd⊕0', 'xe⊕1', 'xf⊕x0'	Match
8	16	B	'1⊕x0', '1⊕x1', '1⊕x2', '1⊕x3', '1⊕x4', '1⊕x5', '1⊕x6', '1⊕x7', 'x8⊕1', 'x9⊕1', 'xa⊕1', 'xb⊕1', 'xc⊕1', 'xd⊕1', 'xe⊕1', 'xf⊕1'	Match

Based on the calculations in Table 9 and the numerically obtained bits, as well as their consistency with the original (B) values, it can be concluded that (wt(X) = 8). Therefore, the bits of (X), which correspond to the expression (X=$f_h(IDS,n_1 \oplus k_i)$), can be derived as follows: for each bit that results from the XOR of a numerical value and a parameter, its value can be determined by equating it to the corresponding original value provided in Table 6. In this manner, the parameter value, which represents one of the bits of (X), can be recovered, and the remaining bits can be obtained in the same way.

In the case where (wt(X) = 8) and (wt(Y) = 0), according to Equation 20, we have:

$$\begin{array}{lll}
0 \oplus x0 = 0 \Rightarrow x0 = 0 & 0 \oplus x5 = 0 \Rightarrow x5 = 0 & xa \oplus 0 = 1 \Rightarrow xa = 1 \\
0 \oplus x1 = 0 \Rightarrow x1 = 0 & 0 \oplus x6 = 1 \Rightarrow x6 = 1 & xb \oplus 0 = 1 \Rightarrow xb = 1 \\
0 \oplus x2 = 0 \Rightarrow x2 = 0 & 0 \oplus x7 = 0 \Rightarrow x7 = 0 & xc \oplus 0 = 0 \Rightarrow xc = 0 \\
0 \oplus x3 = 0 \Rightarrow x3 = 0 & x8 \oplus 0 = 0 \Rightarrow x8 = 0 & xd \oplus 0 = 1 \Rightarrow xd = 1 \\
0 \oplus x4 = 1 \Rightarrow x4 = 1 & x9 \oplus 0 = 1 \Rightarrow x9 = 1 & xe \oplus 0 = 1 \Rightarrow xe = 1 \\
 & & xf \oplus 0 = 1 \Rightarrow xf = 1
\end{array} \qquad (20)$$

As a result, based on the calculations performed, all bits of (X) have been obtained. Therefore, according to Equation 21, we have:

$$X = f_h(IDS, n_1 \oplus k_i) = 0000101001110111 \qquad (21)$$

Now, according to the relation for (B), since the value of the expression ($f_h(IDS, n_1 \oplus k_i)$has been obtained, (k_i) can be derived as follows. (IDS) and (X) are known, and (n_1) can take different values. Since the Hamming weight of (IDS) is specified, the corresponding row in the parametric table is examined. Then, for (wt(IDS) = 4) and different Hamming weights of ($n_1 \oplus k_i$), the corresponding parametric table is constructed, as shown in Table 10, allowing some bits of ($n_1 \oplus k_i$) to be obtained.

For the remaining bits, another (n_1) is considered according to Equation 22:

$$n_1 = 0010111000000111 \qquad (22)$$

Using the same method as before, the value of ($f_h(IDS, n_1 \oplus k_i)$) is recalculated, and the remaining bits of ($n_1 \oplus k_i$) are recovered. This process continues iteratively until all bits of this expression are discovered. At each stage, in order to verify which candidate is valid, (D) can be computed using the obtained (k_i) and compared with the original (D) to determine the correctness or invalidity of the candidate.

Therefore, by assuming ($Y = n_1 \oplus k_i$) and (X = IDS), Table 10 is constructed.

Table 10. Parametric computation of ($f_h(IDS, n_1 \oplus k_i)$) for (wt(X) = 4) and different (wt(Y)) values.

Wt(X)	Wt(Y)	$f_h(IDS, n_1 \oplus k_i)$	Comparison with $f_h(IDS, n_1 \oplus k_i)$
4	0	y4⊕1 y5⊕0 y6⊕0 y7⊕0 0 1 0 1 1⊕y8 0⊕y9 0⊕ya 0⊕yb y0⊕yc y1⊕yd y2⊕ye y3⊕yf	Mismatch – Rejected

4	1	y3⊕ye y4⊕1 y5⊕0 y6⊕0 0 1 0 1 1 0⊕y7 0⊕y8 0⊕y9 yf⊕ya y0⊕yb y1⊕yc y2⊕yd	Mismatch – Rejected
4	2	y2⊕yc y3⊕yd y4⊕1 y5⊕0 0 1 0 1 1 0 0⊕y6 0⊕y7 ye⊕y8 yf⊕y9 y0⊕ya y1⊕yb	Mismatch – Rejected
4	3	y1⊕ya y2⊕yb y3⊕yc y4⊕1 0 1 0 1 1 0 0 0⊕y5 yd⊕y6 ye⊕y7 yf⊕y8 y0⊕y9	Mismatch – Rejected
4	4	y0⊕y8 y1⊕y9 y2⊕ya y3⊕yb 1 1 0 1 1 0 0 0 yc⊕y4 yd⊕y5 ye⊕y6 yf⊕y7	Mismatch – Rejected
4	5	yf⊕y6 y0⊕y7 y1⊕y8 y2⊕y9 0⊕ya 0 0 1 1 0 0 0 yb⊕0 yc⊕y3 yd⊕y4 ye⊕y5	Mismatch – Rejected
4	6	ye⊕y4 yf⊕y5 y0⊕y6 y1⊕y7 0⊕y8 1⊕y9 1 1 1 0 0 0 ya⊕0 yb⊕0 yc⊕y2 yd⊕y3	Mismatch – Rejected
4	7	yd⊕y2 ye⊕y3 yf⊕y4 y0⊕y5 0⊕y6 1⊕y7 0⊕y8 0 1 0 0 0 y9⊕0 ya⊕0 yb⊕0 yc⊕y1	Mismatch – Rejected
4	8	yc⊕y0 yd⊕y1 ye⊕y2 yf⊕y3 0⊕y4 1⊕y5 0⊕y6 1⊕y7 0 0 0 0 y8⊕0 y9⊕0 ya⊕0 yb⊕0	Mismatch – Rejected
4	9	yb⊕0 yc⊕yf yd⊕y0 ye⊕y1 0⊕y2 1⊕y3 0⊕y4 1⊕y5 1⊕y6 1 0 0 y7⊕0 y8⊕0 y9⊕0 ya⊕0	Mismatch – Rejected
4	10	ya⊕0 yb⊕0 yc⊕ye yd⊕yf 0⊕y0 1⊕y1 0⊕y2 1⊕y3 1⊕y4 0⊕y5 1 0 y6⊕0 y7⊕0 y8⊕0 y9⊕0	Mismatch – Rejected
4	11	y9⊕0 ya⊕0 yb⊕0 yc⊕yd 0⊕ye 1⊕yf 0⊕y0 1⊕y1 1⊕y2 0⊕y3 0⊕y4 1 y5⊕0 y6⊕0 y7⊕0 y8⊕0	Match – Candidate
4	12	y8⊕0 y9⊕0 ya⊕0 yb⊕0 0⊕yc 1⊕yd 0⊕ye 1⊕yf 1⊕y0 0⊕y1 0⊕y2 0⊕y3 y4⊕1 y5⊕0 y6⊕0 y7⊕0	Match – Candidate
4	13	y7⊕0 y8⊕0 y9⊕0 ya⊕0 0 1⊕yb 0⊕yc 1⊕yd 1⊕ye 0⊕yf 0⊕y0 0⊕y1 y3⊕y2 y4⊕1 y5⊕0 y6⊕0	Mismatch – Rejected

4	14	y6⊕0 y7⊕0 y8⊕0 y9⊕0 0 1 0⊕ya 1⊕yb 1⊕yc 0⊕yd 0⊕ye 0⊕yf y2⊕y0 y3⊕y1 y4⊕1 y5⊕0	Mismatch – Rejected
4	15	y5⊕0 y6⊕0 y7⊕0 y8⊕0 0 1 0 1⊕y9 1⊕ya 0⊕yb 0⊕yc 0⊕yd y1⊕ye y2⊕yf y3⊕y0 y4⊕1	Mismatch – Rejected
4	16	y4⊕1 y5⊕0 y6⊕0 y7⊕0 0 1 0 1 1⊕y8 0⊕y9 0⊕ya 0⊕yb y0⊕yc y1⊕yd y2⊕ye y3⊕yf	Mismatch – Rejected

Now, there are two candidates. The first candidate has (wt(IDS) = 4) and (wt($n_1 \oplus k_i$)=11), while the second candidate has (wt(IDS) = 4) and (wt($n_1 \oplus k_i$)=12). For the first candidate, some of its bits can be obtained according to Equation 23:

$$\begin{array}{lll}
y9 \oplus 0 = 0 \ \Rightarrow y9 = 0 & 0 \oplus y0 = 1 \ \Rightarrow y0 = 1 & y5 \oplus 0 = 0 \ \Rightarrow y5 = 0 \\
ya \oplus 0 = 0 \ \Rightarrow ya = 0 & 1 \oplus y1 = 0 \ \Rightarrow y1 = 1 & y6 \oplus 0 = 1 \ \Rightarrow y6 = 1 \\
yb \oplus 0 = 0 \ \Rightarrow yb = 0 & 1 \oplus y2 = 0 \ \Rightarrow y2 = 0 & y7 \oplus 0 = 1 \ \Rightarrow y7 = 1 \\
0 \oplus ye = 1 \ \Rightarrow ye = 1 & 0 \oplus y3 = 1 \ \Rightarrow y3 = 1 & y8 \oplus 0 = 1 \ \Rightarrow y8 = 1 \\
1 \oplus yf = 0 \ \Rightarrow yf = 1 & 0 \oplus y4 = 1 \ \Rightarrow y4 = 1 &
\end{array} \tag{23}$$

Therefore, according to Equation 24, we have:

$$Y = n_1 \oplus k_i = 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ yc\ yd\ 1\ 1 \tag{24}$$

Considering that, in one of the eavesdropping sessions, the values of (n_1), (n_2), (n_3), and (D) were captured according to Equations 25 to 28, and the value of (IDS) is also known.

$$n_1{=}0010111000000111 \tag{25}$$

$$n_2{=}1110000000000000 \tag{26}$$

$$n_3{=}1110010001000000 \tag{27}$$

$$\mathrm{D}{=}0010101000101001 \tag{28}$$

Now, the value of (k_i) can be obtained based on the known value of (n_1) and ($Y = n_1 \oplus k_i$), according to Equation 29:

$$k_i = 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ kc\ kd\ 0\ 0 \tag{29}$$

According to Equation 30:

$$D = f_h(Rot_l(Rot_r(f_h(IDS,n_1),k_i),n_2),n_3) \tag{30}$$

However, with the obtained (k_i), it is not possible to compute the value of (D) for comparison with the original (D) and thereby verify the correctness or invalidity of the candidate.

Moreover, the remaining bits cannot be obtained, because if a different (n_1) is assumed, the resulting expression ($f_h(IDS,n_1 \oplus k_i)$) generally changes, and (Y) changes as well. Similarly, if (n_2) is changed while (n_1) remains the same, no change occurs, and the value of ($f_h(IDS,n_1 \oplus k_i)$) remains constant.

The value of (Y) can only be fully determined when (wt(IDS) = 4) and ($\mathrm{wt}(n_1 \oplus k_i)$=12), because in that case all bits can be calculated. In the other rows, even if there is a candidate, it is not possible to accept or reject it by changing (n_1) and (n_2), since the remaining bits cannot be recovered.

Candidates in other rows can only be examined by considering different (IDS) values corresponding to the same (n_1) and (n_2) that were obtained through eavesdropping. In this way, the validity or invalidity of candidates in rows other than 4 and 12 can be easily verified.

Now, the next candidate with (wt(IDS) = 4) and ($\mathrm{wt}(n_1 \oplus k_i)$=12) is considered, and (Y) is obtained according to Equation 31:

$$\begin{array}{lll}
y8 \oplus 0 = 0 \Rightarrow y8 = 0 & 1 \oplus yd = 0 \Rightarrow yd = 1 & 0 \oplus y1 = 1 \Rightarrow y1 = 1 \\
y9 \oplus 0 = 0 \Rightarrow y9 = 0 & 0 \oplus ye = 1 \Rightarrow ye = 1 & 0 \oplus y2 = 1 \Rightarrow y2 = 1 \\
ya \oplus 0 = 0 \Rightarrow ya = 0 & 1 \oplus yf = 0 \Rightarrow yf = 1 & 0 \oplus y3 = 1 \Rightarrow y3 = 1 \\
yb \oplus 0 = 0 \Rightarrow yb = 0 & 1 \oplus y0 = 0 \Rightarrow y0 = 1 & y4 \oplus 1 = 0 \Rightarrow y4 = 1 \\
0 \oplus yc = 1 \Rightarrow yc = 1 & y6 \oplus 0 = 1 \Rightarrow y6 = 1 & y5 \oplus 0 = 1 \Rightarrow y5 = 1 \\
 & & y7 \oplus 0 = 1 \Rightarrow y7 = 1
\end{array} \tag{31}$$

Therefore, according to Equation 32, we have:

$$n_1 \oplus k_i = Y = 1111111100001111 \tag{32}$$

Considering that (n_1) can be eavesdropped and, according to Equation 33, it is equal to:

$$n_1 = 0010111000000111 \tag{33}$$

(k_i) can thus be obtained, which according to Equation 34 is:

$$k_i = 1101000100001000 \tag{34}$$

To verify (k_i), (D) can be computed using the obtained (k_i) and compared with the original eavesdropped (D). If they match, it confirms the validity of (k_i).

The computation of (D) with the obtained (k_i) is given according to Equation 35:

$$D = f_h(Rot_l(Rot_r(f_h(IDS,n_1),k_i),n_2),n_3) \tag{35}$$

Since all the parameters involved in the computation of (D) are known, (D) can be calculated as follows. In the eavesdropped session, the values of (n_1), (n_2), (n_3), and (D) were captured according to Equations 36 to 39, and (IDS) is also known:

$$n_1=0010111000000111 \tag{36}$$

$$n_2=1110000000000000 \tag{37}$$

$$n_3=1110010001000000 \tag{38}$$

$$D=0010101000101001 \tag{39}$$

Now, using the obtained (k_i), (D) is calculated, resulting in Equation 40. Comparing this with the original eavesdropped (D) verifies the correctness or invalidity of the selected candidate:

$$\text{(computed) } D=0010101000101001 \tag{40}$$

The original (D) captured from the channel, according to Equation 41, is:

$$\text{(eavesdropped) } D=0010101000101001 \tag{41}$$

Since the computed (D) matches the original eavesdropped value, it can be concluded that the recovered (k_i) is valid, making the selected candidate valid, while the previous candidate is invalid.

Now that (k_i) has been recovered and (n_1) and (n_2) are also known via eavesdropping, (TID) can be obtained using Equation 42:

$$A = Rot_r(f_h(Rot_l(TID,k_i),n_1 \oplus n_2),n_2) \tag{42}$$

According to this relation, and considering that the values of (A), (n_1), and (n_2) are observable from the channel, the attacker can perform eavesdropping with different values of (n_1) and (n_2) and obtain different values of (A). Therefore, according to Equation 43:

$$Rot_l(A,n_2) \ = \ f_h(Rot_l(TID,k_i),n_1 \oplus n_2) \tag{43}$$

If we denote (X=$Rot_l(TID,k_i)$), (Y=$n_1 \oplus n_2$), and (Z=$Rot_l(A,n_2)$), the attacker can construct a table similar to Table 11:

Table 11. Eavesdropped values of (Z) for different (n_1) and (n_2) values.

wt (X)	wt (Y)	n_1	n_2	section	value
?	0	0000000000000000	0000000000000000	Y	0000000000000000
				Z	0000010001000000
?	1	1000000000000000	1000000000000000	Y	1000000000000000
				Z	0000001001010000
?	2	0000000000000000	1000000000000000	Y	1000000000000001
				Z	0000000101011000
?	3	1000000000000000	1111000000000000	Y	0111000000000000
				Z	1000000011000011
?	4	1110000000000000	1111111000000000	Y	0001111000000000
				Z	0100000000001110
?	5	0001110000000000	1111111100000000	Y	1110001100000000
				Z	0000000001100001
?	6	0000001110000000	1111111110000000	Y	1111110000000000
				Z	0000000001010011
?	7	1000000111000000	1111111111100000	Y	0111111000100000
				Z	1111100001011001
?	8	0000001101011100	1111111111111100	Y	1111110010100000
				Z	1111110001101100
?	10	1001000000011001	1111111111111000	Y	0110111111100001
				Z	0001011010111111
?	11	0010111000000111	1111111111110000	Y	1101000111110111
				Z	0001111100001111

Next, one of the rows of Table 11 is selected, and for the corresponding value of (Y), different Hamming weights are considered for (X) and computed in a parametric manner.

For (wt(Y) = 2), Table 12 can be constructed.

Table 12. Parametric computation for (wt(Y) = 2) and different (wt(X)) values.

wt (X)	wt (Y)	section	value	status
0	2	Z	x0⊕0', 'x1⊕0', 'x2⊕x8', 'x3⊕x9', 'x4⊕xa', 'x5⊕xb', 'x6⊕xc', 'x7⊕xd', '0⊕xe', '1⊕xf', 1, 0, 0, 0, 0, 0	Mismatch – Rejected

1	2	Z	0, 'xf⊕0', 'x0⊕x7', 'x1⊕x8', 'x2⊕x9', 'x3⊕xa', 'x4⊕xb', 'x5⊕xc', 'x6⊕xd', '0⊕xe', 1, 1, 0, 0, 0, 0	Mismatch – Rejected
2	2	Z	0, 0, 'xe⊕x6', 'xf⊕x7', 'x0⊕x8', 'x1⊕x9', 'x2⊕xa', 'x3⊕xb', 'x4⊕xc', 'x5⊕xd', 0, 1, 1, 0, 0, 0	Match – Candidate
3	2	Z	0, 0, '0⊕x5', 'xd⊕x6', 'xe⊕x7', 'xf⊕x8', 'x0⊕x9', 'x1⊕xa', 'x2⊕xb', 'x3⊕xc', 'x4⊕0', 0, 1, 1, 0, 0	Mismatch – Rejected
4	2	Z	0, 0, '0⊕x4', '0⊕x5', 'xc⊕x6', 'xd⊕x7', 'xe⊕x8', 'xf⊕x9', 'x0⊕xa', 'x1⊕xb', 'x2⊕0', 'x3⊕0', 0, 1, 1, 0	Mismatch – Rejected
5	2	Z	0, 0, '0⊕x3', '0⊕x4', '0⊕x5', 'xb⊕x6', 'xc⊕x7', 'xd⊕x8', 'xe⊕x9', 'xf⊕xa', 'x0⊕0', 'x1⊕0', 'x2⊕0', 0, 1, 1	Mismatch – Rejected
6	2	Z	1, 0, '0⊕x2', '0⊕x3', '0⊕x4', '0⊕x5', 'xa⊕x6', 'xb⊕x7', 'xc⊕x8', 'xd⊕x9', 'xe⊕0', 'xf⊕0', 'x0⊕0', 'x1⊕0', 0, 1	Mismatch – Rejected
7	2	Z	1, 1, '0⊕x1', '0⊕x2', '0⊕x3', '0⊕x4', '0⊕x5', 'x9⊕x6', 'xa⊕x7', 'xb⊕x8', 'xc⊕0', 'xd⊕0', 'xe⊕0', 'xf⊕0', 'x0⊕0', 0	Mismatch – Rejected
8	2	Z	0, 1, '1⊕x0', '0⊕x1', '0⊕x2', '0⊕x3', '0⊕x4', '0⊕x5', 'x8⊕x6', 'x9⊕x7', 'xa⊕0', 'xb⊕0', 'xc⊕0', 'xd⊕0', 'xe⊕0', 'xf⊕0'	Mismatch – Rejected
9	2	Z	'xe⊕0', 0, '1⊕xf', '1⊕x0', '0⊕x1', '0⊕x2', '0⊕x3', '0⊕x4', '0⊕x5', 'x7⊕x6', 'x8⊕0', 'x9⊕0', 'xa⊕0', 'xb⊕0', 'xc⊕0', 'xd⊕0'	Match – Candidate
10	2	Z	'xc⊕0', 'xd⊕0', '0⊕xe', '1⊕xf', '1⊕x0', '0⊕x1', '0⊕x2', '0⊕x3', '0⊕x4', '0⊕x5', 'x6⊕0', 'x7⊕0', 'x8⊕0', 'x9⊕0', 'xa⊕0', 'xb⊕0'	Match – Candidate
11	2	Z	'xa⊕0', 'xb⊕0', 'xc⊕xd', '0⊕xe', '1⊕xf', '1⊕x0', '0⊕x1', '0⊕x2', '0⊕x3', '0⊕x4', 0, 'x5⊕0', 'x6⊕0', 'x7⊕0', 'x8⊕0', 'x9⊕0'	Match – Candidate
12	2	Z	'x8⊕0', 'x9⊕0', 'xa⊕xc', 'xb⊕xd', '0⊕xe', '1⊕xf', '1⊕x0', '0⊕x1', '0⊕x2', '0⊕x3', 0, 0, 'x4⊕0', 'x5⊕0', 'x6⊕0', 'x7⊕0'	Mismatch – Rejected

13	2	Z	'x6⊕0', 'x7⊕0', 'x8⊕xb', 'x9⊕xc', 'xa⊕xd', '0⊕xe', '1⊕xf', '1⊕x0', '0⊕x1', '0⊕x2', 0, 0, 0, 'x3⊕0', 'x4⊕0', 'x5⊕0'	Mismatch – Rejected
14	2	Z	'x4⊕0', 'x5⊕0', 'x6⊕xa', 'x7⊕xb', 'x8⊕xc', 'x9⊕xd', '0⊕xe', '1⊕xf', '1⊕x0', '0⊕x1', 0, 0, 0, 0, 'x2⊕0', 'x3⊕0'	Mismatch – Rejected
15	2	Z	'x2⊕0', 'x3⊕0', 'x4⊕x9', 'x5⊕xa', 'x6⊕xb', 'x7⊕xc', 'x8⊕xd', '0⊕xe', '1⊕xf', '1⊕x0', 0, 0, 0, 0, 0, 'x1⊕0'	Mismatch – Rejected
16	2	Z	'x0⊕0', 'x1⊕0', 'x2⊕x8', 'x3⊕x9', 'x4⊕xa', 'x5⊕xb', 'x6⊕xc', 'x7⊕xd', '0⊕xe', '1⊕xf', 1, 0, 0, 0, 0, 0	Mismatch – Rejected

According to Table 12, there are four candidates. Therefore, it is necessary to examine these candidates for different (wt(Y)) values in order to identify the valid candidate and recover the bits of (X). Accordingly, Table 13 is constructed.

Table 13. Elimination of some candidates.

wt(X)	wt(Y)	section	value	status
9	7	Z	'xe⊕1', 1, 0, 1, 1, 0, 0, '0⊕xf', '0⊕x0', 'x7⊕x1', 'x8⊕x2', 'x9⊕x3', 'xa⊕x4', 'xb⊕x5', 'xc⊕x6', 'xd⊕1'	Mismatch – Rejected
10	0	Z	'xc⊕xe', 'xd⊕xf', '0⊕x0', '0⊕x1', '0⊕x2', '0⊕x3', '0⊕x4', '0⊕x5', 0, 0, 'x6⊕0', 'x7⊕0', 'x8⊕0', 'x9⊕0', 'xa⊕0', 'xb⊕0'	Mismatch – Rejected
11	0	Z	'xa⊕xd', 'xb⊕xe', 'xc⊕xf', '0⊕x0', '0⊕x1', '0⊕x2', '0⊕x3', '0⊕x4', 0, 0, 0, 'x5⊕0', 'x6⊕0', 'x7⊕0', 'x8⊕0', 'x9⊕0'	Mismatch – Rejected

Since, for the candidates in Table 13, a Hamming weight for (Y) was found that can prove their invalidity, (wt(X)=2) is acceptable. For further verification and in order to recover the bits, Table 14 can be constructed for different Hamming weights.

Table 14. Parametric computation of (Z) for (wt(X)=2) and different values of (wt(Y)).

wt (X)	wt (Y)	sectoin	value	status

2	0	Z	'0⊕x6', '0⊕x7', 'xe⊕x8', 'xf⊕x9', 'x0⊕xa', 'x1⊕xb', 'x2⊕xc', 'x3⊕xd', 'x4⊕0', 'x5⊕0', 0, 0, 0, 0, 0, 0	Match
2	1	Z	0, '0⊕x6', 'xe⊕x7', 'xf⊕x8', 'x0⊕x9', 'x1⊕xa', 'x2⊕xb', 'x3⊕xc', 'x4⊕xd', 'x5⊕0', 0, 1, 0, 0, 0, 0	Match
2	2	B	0, 0, 'xe⊕x6', 'xf⊕x7', 'x0⊕x8', 'x1⊕x9', 'x2⊕xa', 'x3⊕xb', 'x4⊕xc', 'x5⊕xd', 0, 1, 1, 0, 0, 0	Match
2	3	B	1, 0, 'xe⊕0', 'xf⊕x6', 'x0⊕x7', 'x1⊕x8', 'x2⊕x9', 'x3⊕xa', 'x4⊕xb', 'x5⊕xc', '0⊕xd', 0, 0, 0, 1, 1	Match
2	4	B	0, 1, 'xe⊕0', 'xf⊕0', 'x0⊕x6', 'x1⊕x7', 'x2⊕x8', 'x3⊕x9', 'x4⊕xa', 'x5⊕xb', '0⊕xc', '0⊕xd', 1, 1, 1, 0	Match
2	5	B	0, 0, 'xe⊕0', 'xf⊕0', 'x0⊕0', 'x1⊕x6', 'x2⊕x7', 'x3⊕x8', 'x4⊕x9', 'x5⊕xa', '0⊕xb', '0⊕xc', '0⊕xd', 0, 0, 1	Match
2	6	B	0, 0, 'xe⊕0', 'xf⊕0', 'x0⊕0', 'x1⊕0', 'x2⊕x6', 'x3⊕x7', 'x4⊕x8', 'x5⊕x9', '0⊕xa', '0⊕xb', '0⊕xc', '0⊕xd', 1, 1	Match

14 and the numerically obtained bits, as well as their consistency with the original (Z) values, it can be concluded that (wt(X) = 2). Therefore, the bits of (X), which correspond to the expression (X=$Rot_l(TID,k_i)$), can be determined as follows: for each bit that results from the XOR of a numerical value and a parameter, its value can be obtained by equating it to the corresponding original value provided in Table 11. In this way, the parameter value, which represents one of the bits of (X), is recovered, and the remaining bits are computed in the same manner.

In the case where (wt(X) = 2) and (wt(Y) = 0), according to Equations 44, we have:

$$\begin{aligned} 0 \oplus x6 = 0 \ \Rightarrow x6 = 0 \qquad & x5 \oplus 0 = 1 \ \Rightarrow x5 = 1 \\ 0 \oplus x7 = 0 \ \Rightarrow x7 = 0 \qquad & 0 \oplus x4 = 1 \ \Rightarrow x4 = 1 \\ x4 \oplus 0 = 0 \ \Rightarrow x4 = 0 \qquad & 0 \oplus x5 = 0 \ \Rightarrow x5 = 0 \end{aligned} \tag{44}$$

In the case where (wt(X) = 2) and (wt(Y) = 3), according to Equation 45:

$$0 \oplus xd = 0 \ \Rightarrow xd = 0 \tag{45}$$

In the case where (wt(X) = 2) and (wt(Y) = 4), according to Equations 46:

$$\begin{aligned} &xe \oplus 0 = 0 \ \Rightarrow xe = 0 \\ &xf \oplus 0 = 0 \ \Rightarrow xf = 0 \\ &0 \oplus xc = 0 \ \Rightarrow xc = 0 \\ &0 \oplus xd = 0 \ \Rightarrow xd = 0 \end{aligned} \tag{46}$$

In the case where (wt(X) = 2) and (wt(Y) = 5), according to Equations 47:

$$\begin{aligned} &x0 \oplus 0 = 0 \ \Rightarrow x0 = 0 \\ &x1 \oplus x6 = 0 \ \Rightarrow x1 \oplus 0 = 0 \Rightarrow x1 = 0 \\ &x2 \oplus x7 = 0 \ \Rightarrow x2 \oplus 0 = 0 \Rightarrow x2 = 0 \\ &x4 \oplus x9 = 0 \ \Rightarrow 0 \oplus x9 = 0 \Rightarrow x9 = 0 \\ &x5 \oplus xa = 1 \ \Rightarrow 1 \oplus xa = 1 \Rightarrow xa = 0 \\ &0 \oplus xb = 1 \ \Rightarrow xb = 1 \end{aligned} \tag{47}$$

In the case where (wt(X) = 2) and (wt(Y) = 6), according to Equations 48:

$$\begin{aligned} &x3 \oplus x7 = 0 \ \Rightarrow x3 \oplus 0 = 0 \Rightarrow x3 = 0 \\ &x4 \oplus x8 = 0 \ \Rightarrow 0 \oplus x8 = 0 \Rightarrow x8 = 0 \end{aligned} \tag{48}$$

As a result, (X) is obtained according to Equation 49:

$$X{=}Rot_l(TID{,}k_i) = 0000010000010000 \tag{49}$$

Since (k_i) was previously recovered, (TID) can be obtained using Equation 50:

$$TID = Rot_r(X{,}k_i) = 1000000000100000 \tag{50}$$

Therefore, by following the above algorithm, the secret values (k_i) and (TID) have been successfully recovered.

Consequently, based on this example, it can be concluded that the function (f_h) used is **not secure against the secret value discovery attack**.

Conclusion

The example demonstrates that the proposed algorithm successfully recovers the secret values (k_i) and (TID) by analyzing the observed protocol messages. The results indicate that the function (f_h) used in AnonR2AS is not secure against the secret value discovery attack, highlighting the vulnerability of the protocol.