*A. WDDL With No Precharge*

The table below shows the results of the attacks for a WDDL implementation with no precharge phase.

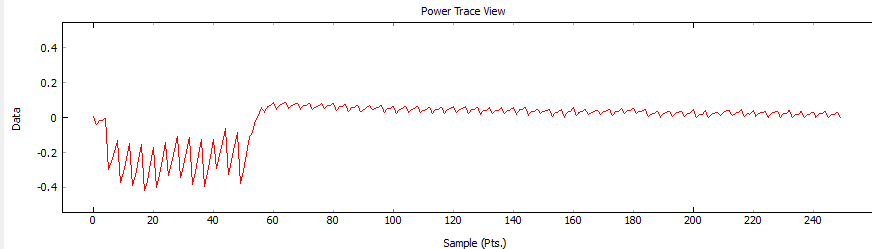
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AES Design** | **Area** | **Traces Avg.** | **Trace 1** | **Trace 2** | **Trace 3** | **Trace 4** | **Trace 5** |
| Basic | 2055 | 1526 | 1250 | 1430 | 1880 | 1640 | 1430 |
| AddRoundKey | 2224 | 2338 | 3060 | 1840 | 2550 | 2320 | 1920 |
| MixCol | 3510 | 2386 | 2040 | 2550 | 2030 | 2240 | 3070 |
| SubBytes | 8936 | 3374 | 2790 | 3940 | 3170 | 3000 | 3970 |
| Add+Mix | 3654 | 3000 | 2380 | 2720 | 3130 | 3290 | 3480 |
| Add+SB | 9780 | 3452 | 3360 | 4580 | 2520 | 3680 | 3120 |
| Mix+SB | 10487 | 6600 | 7540 | 5310 | 5130 | 8450 | 6570 |
| All | 11077 | 4450 | 5340 | 3190 | 4290 | 5330 | 4100 |

Table ??: Results of WDDL modification without precharge implementation.

One interesting aspect to observe in the data above is that there is a general trend whereas the area of the design increases so does the number of power traces needed to uncover the correct subkey. Figure ?? displays the data in a scatter plot. From this graph one can see this general. One explanation for this could possibly be with an increase in area there is an increase in critical paths. (Not really sure why these results are the way they are.)

Figure ??: Scatter plot shows the interesting trend of an increase in area correlating with an increase in security.

The power trace for the design with all three portions of AES implementing the logic of WDDL is below. This trace still looks very similar to the trace of the baseline attack done but the correlations of the subkey guesses are lower??



*B. WDDL With Precharge*

The table below shows the results of the attacks for a WDDL implementation with no precharge phase.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AES Design** | **Area** | **Traces Avg.** | **Trace 1** | **Trace 2** | **Trace 3** | **Trace 4** | **Trace 5** |
| Basic | 11912 | 1060 | 910 | 1150 | 1270 | 1000 | 970 |
| AddRoundKey | 12280 | 1010 | 780 | 880 | 1080 | 1130 | 1180 |
| MixCol | 12932 | 988 | 1020 | 1000 | 1020 | 1030 | 920 |
| SubBytes | 17222 | 38313 | 38860 | 52300 | 34310 | 28110 | 37985 |
| Add+Mix | 13128 | 1020 | 1020 | 980 | 1000 | 1110 | 990 |
| Add+SB | 16664 | 40658 | 43900 | 38790 | 38400 | 40130 | 42070 |
| Mix+SB | 18324 | 53050 | 44600 | 62100 | 55030 | 50430 | 53090 |
| All | 18618 | 70440 | 84230 | 69700 | 58400 | 65280 | 74590 |

Table ??: Results of WDDL modification without precharge implementation.

The results seem to vary in this table depending on what stage of the algorithm is protected by WDDL. Each of the designs which have WDDL implemented on the SubBytes stage have a significant increase in security. This is visible when only SubBytes is protected along with when it is protected in combination with AddRoundKey and MixColumn. In the design where only AddRoundKey was protected it only took an average of 1010 power traces to uncover the key. This is about half the traces needed for AddRoundKey when comparing it to WDDL with no precharge phase design. When AddRoundKey and SubBytes are protected together this number jumps all to 40,658. This is an increase of 40x.

One explanation for why designs that have a protected SubBytes stage have a significant increase in security is because this stage is the most computationally heavy. When WDDL is implemented in the SubBytes stage there are on average an addition of 5300 LUTs. This is compared to MixColumn, which only has an addition of about 1000 LUTs, and AddRoundKey, which only has an addition of about 400 LUTs. This means, by protecting the SubBytes stage, WDDL is actually protecting the majority of the logic in the design and they’re for providing the most security.

Another explanation of why protecting the SubBytes stage increases the security so much is because the output of the SBoxes used in the SubBytes stage is what is monitored during the attack. This means, if the power consumption during this stage of the algorithm is able to be consistent, an attacker targeting this point in the algorithm with a power analysis attack is going to have a more difficult time deciphering the key.

The other interesting aspect of this data that was collected is the basic design with a precharge phase actually has about half the security of just the baseline design with no added defense. This is also consistent with any design that does not protect the SubBytes stage. My hypothesis for why this is happening is because the precharge phase is actually making the variations in the power traces more significant. I believe this is the case because when the precharge phase is implemented, every time the evaluation phase occurs, every route that evaluates to a 1 has to be charged. Not only does this consume much more power than a design with no precharge phase but, because the loads of the LUTs are not balanced, it also varies the power traces as well. A collection of power traces with more significant variation allows for a faster CPA attack.

The graph below shows the optimal designs for WDDL protection. Any of the designs below the line exhibit an increase in area without an increase in security. A design that meets this condition is not an optimal AES design.

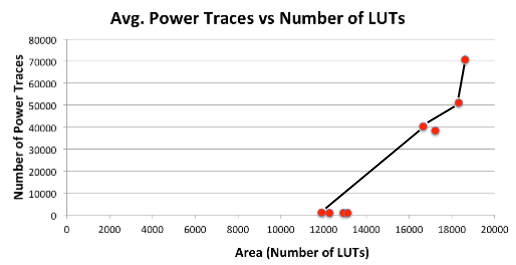


Figure ??: Graph displays optimal designs when protecting AES with WDDL.

A fully WDDL protected AES core took 70,440 power traces on average to uncover the subkey. This attack took about 4 hours to complete, which is a significant increase compared to the basic AES core which only took 20 minutes to break. Along with this, the design has 46 times the security as the baseline AES core. With this increase in security comes an increase in area as well. When compared to the baseline AES core, the fully protected AES core has 9 times the area.