USING MACHINE LEARNING TO ASSIST HARDWARE ATTACKS

Amelia Kwak, Soner Seckiner, Jacob Sepinuck

University of Rochester

ABSTRACT

Side-channel attacks remain a threat to cryptographic devices. Attacks such as power analyses have proven themselves capable of breaking even AES and other state of the art encryption. Chip designers know of this issue and implement masking techniques in order to obfuscate side-channel traces and complicate attacks. We set out to use machine learning in order to simplify masked side-channel measurements, thus making it easier for attackers to determine encryption keys of masked signals. We tested multiple machine learning networks including multi-level perceptron (MLP), autoencoders (AE), and convolutional neural networks (CNN) with generative adversarial networks (GAN). We found that our GAN-AE performed the best, successfully breaking a masked AES encryption.

1. SIDE-CHANNEL ATTACKS

All electronic devices emit side-channels. These can include power draw, temperature, sound, and many other measurable features of a device running. Such side-channels can give an attacker information about the device which may not be intended by the manufacturer. This is especially noticeable in cryptographic chips. Using side-channels (usually power traces due to their low noise and ease of measurement compared to other side-channels), attackers can use this extra information to break encryption (see *fig. 1)*. This can even affect cutting edge encryption algorithms and standards including AES [1]. Though a critical security risk, this does not affect most device users since the invasive nature of these attacks requires an attacker to have physical access to the device under attack. Even so, manufacturers recognize the risk side-channels pose to users, intellectual property, and more and thus implement countermeasures. Often, such countermeasures present as masking. Masking obfuscates the data an attacker can collect from a device, making power analyses much less successful. Though power traces can usually be correlated with the function and key of a cryptographic device, masking hides the correlation often by performing dummy functions to throw off an attacker.

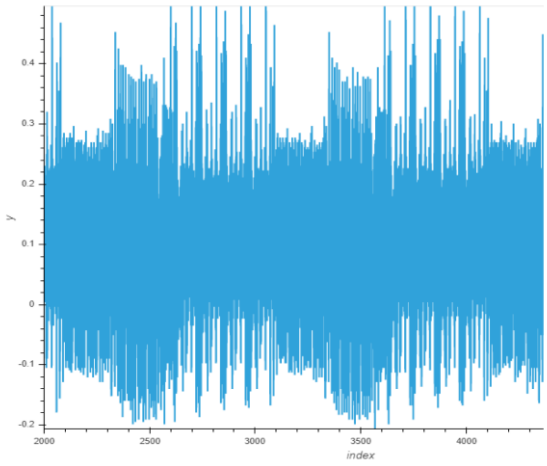

Figure 1 - Sample power traces collected from cryptographic device

2. USING MACHINE LEARNING TO AID ATTACKS

Machine learning has already proved itself useful for aiding side-channel attacks. Wang et. al. used a GAN to reduce the number of power traces needed to successfully attack a chip by nearly ½ [2]. Autoencoders have also been used to remove noise when preprocessing side-channel attacks with success [3]. We set out to use such techniques to combat masking in order to perform successful side-channel attacks. In our model, a discriminator evaluates traces from both a model and attacked device. It then uses back-propagation to update the attacked device's autoencoder to generate signals comparable to the model signals. This process is continued until the autoencoder can produce traces from the attacked signals that are indistinguishable from the model signals. With these signals, we should be able to perform an attack which we can use to break the encryption.

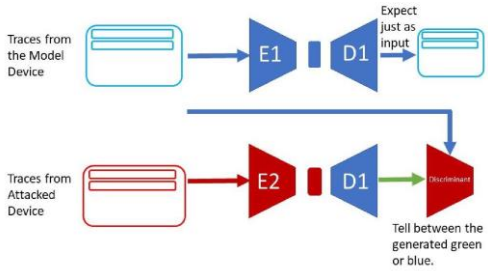

Figure 2 - Our model for enhancing side-channel attacks of masked signals

3. PREPARATION AND RESULTS

The ASCAD database provided us with a large number of masked 128-bit AES samples. We learned the secret key used on the device in the ASCAD database, and applied it to our own device which performed unmasked 128-bit AES encryption. Our device, a Chipwhisperer, is the same device as used in the ASCAD database. Once we applied the same key, we collected 100,000 unmasked AES power traces from our device. From this, we identified the signal-to-noise ratio of the 3rd S-Box (a subcomponent of the key). We also identified the range of traces which corresponded to the 3rd S-Box and trimmed our traces to this range due to the fact the ASCAD database published the same 3rd S-Box signals. We applied a hamming weight model to the correlation between hamming weight and the signal (see fig. 3). We then used this data to train our GAN. Examples of masked and unmasked AES traces can be seen below in figures 4 and 5 respectively.

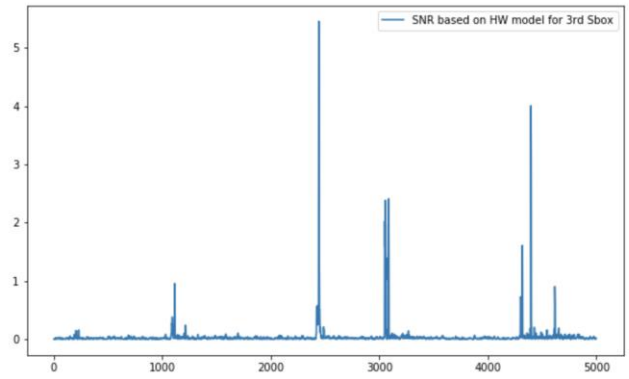

Figure 3 - Signal-to-noise ratio based on hamming weight model of 3rd S-Box

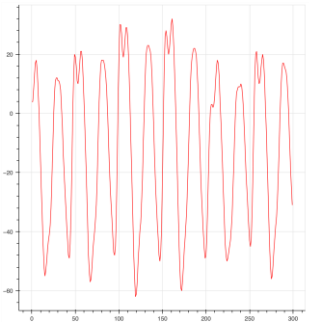

Figure 4 - Masked Traces

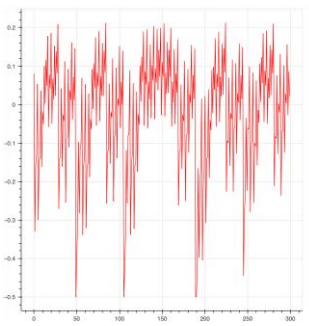

Figure 5 - Unmasked traces

Using the hamming weight model discussed above, we used a correlation power analysis (CPA) attack to validate the key stored in the ASCAD database with the traces in the database. We confirmed that both keys matched, verifying that the data was good to use. For the rest of our tests, we attacked specifically the 3rd S-Box location (subkey 2 shown in fig. 6). We initially trained the GAN-AE described above, and successfully found the correct key using the traces created by our technique. We decided to test other networks as well as shown in fig. 7. We calculated the ranking of each network, with the rank value corresponding to how close the predicted key was to the correct subkey value. As shown in fig. 7, the GAN-AE was the best network, with the others getting close, but not finding the key correctly.

```
leak_model = cwa.leakage_models.sbox_output

attack = cwa.cpa(proj, leak_model)

results = attack.run()
```

```
print(results)
```

```
Subkey KGuess Correlation
  00    0x4D    0.26777
  01    0xFB    0.29003
  02    0xE0    0.25682
  03    0xF2    0.25862
  04    0x72    0.25381
  05    0x21    0.30222
  06    0xFE    0.24523
  07    0x10    0.25612
  08    0xA7    0.25570
  09    0x8D    0.29639
  10    0x4A    0.26180
  11    0xDC    0.29614
  12    0x8E    0.25300
  13    0x49    0.25942
  14    0x04    0.26382
  15    0x69    0.29467
```

Figure 6 - CPA Attack results confirming the same key as in the database

Network	Best Ranking – Lower is better
GAN – MLP	10
GAN – AE	1
GAN – CNN	6
CNN	10

Figure 7 – Rank of different networks when attacking 3rd S-Box

Examples of the transformed traces can be seen below in figure 8.

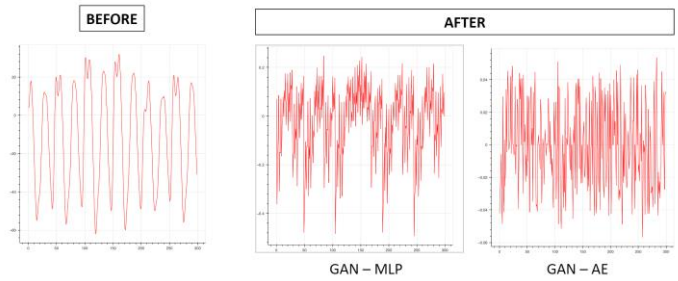

Figure 8 - Masked traces transformed by GAN - MLP and GAN - AE into 'unmaksed' traces

As we can see in fig. 8, our networks were able to transform the masked signals from the ASCAD database into those which resembled unmasked traces gathered from our model device. This is apparent when comparing the traces in fig. 8 to those in fig. 4 and 5. And, as stated above, in the case of our GAN – AE, we were able to use these transformed, 'unmasked' traces in order to successfully guess the encryption key.

4. FUTURE PLANS

Although our GAN – AE network was able to produce traces which allowed us to guess the key correctly, there was substantial variation in our key prediction depending on how many traces we used to run the attack. For example, when using 2,000 traces to run the attack, we achieved generally better results than with higher numbers of traces, which is unexpected (usually a higher number of traces results in better key predictions). This, we wish to adjust our network to be more stable in this respect. We also wish to expand our framework to work with other devices and different types of countermeasures. Finally, we wish to propose a robust denoising framework (also utilizing machine learning techniques) which may produce higher quality traces and improve the accuracy of our attacks.

5. CONCLUSION

Overall, we were able to successfully implement machine learning techniques to attack masked traces from at least one encryption device. Our network was able to take masked power traces and transform them into traces which resemble those which are unmasked, allowing power analysis attacks to succeed. This shows promise for attacks on other devices as well. Although our technique was, in the end, successful, it did have a high level of variation depending on the number of traces used, which we would like to improve going forward.

6. REFERENCES

[1] Owen Lo, “Power Analysis Attacks on the AES-128 S-box Using Differential Power Analysis (DPA) and Correlation Power Analysis (CPA),” *Journal of Cyber Security Technology*, 1:2, 88-107, 19 Sep 2016

[2] Wang, Ping, et al. "Enhancing the performance of practical profiling side-channel attacks using conditional generative adversarial networks." arXiv preprint arXiv:2007.05285 (2020).

[3] Wu, Lichao, and Stjepan Picek. "Remove some noise: On pre-processing of side-channel measurements with autoencoders." *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2020): 389-415.

[4] Zhang, Ziyue, A. Adam Ding, and Yunsi Fei. "A fast and accurate guessing entropy estimation algorithm for full-key recovery." *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2020): 26-48.