1

Deep Learning Techniques for Side Channel Analysis

Nimar Blume

***Abstract*—Put the abstract here. The abstract should be 200-300 words long and must answer the following questions:**

- **Why is this topic important and interesting?**
- **What is the topic?**

It's a good writing style to answer the "why" question first by putting the topic in a broader context.

***Index Terms*—deep learning, side channel analysis, side channel attack, encryption, AES**

I. Introduction

Data encryption is commonly employed to restrict data access to selected people or endpoints. Proper encryption provides privacy, integrity and authenticity of data exchanged between two or more parties over an insecure connection. An encryption key or a pair of keys is generated with which the chosen data is encrypted, so that the data can only be read or modified by an endpoint who possesses the appropriate encryption key. For symmetric encryption, where the same secret key is used to encrypt and decrypt data, the key has to be shared over a secure connection beforehand, whereas asymmetric encryption uses key pairs, one of which is used to encrypt and one is used to decrypt data. Therefore, asymmetric encryption enables two parties to exchange data securely over an insecure connection by publishing the key used for encryption and keeping the key used for decryption private. This paper will only consider symmetric encryption, although techniques discussed could also partly be applied to asymmetric encryption algorithms.

A. The Advanced Encryption Standard

While there are many encryption algorithms available, this paper focuses on the most popular symmetric block cipher: the Advanced Encryption Standard (henceforth AES) with a $128\,\mathrm{bit}$ key. The AES algorithm encrypts data in blocks of $128\,\mathrm{bit}$ or $16\,\mathrm{B}$. Should a block be smaller than $16\,\mathrm{B}$, padding will be appended until the block size reaches $16\,\mathrm{B}$. Furthermore, AES expands a single $128\,\mathrm{bit}$ key to 11 round keys which are then used in the subseqent 11 rounds to encrypt the given data. It is important to note, that a compromise of any of the 11 round keys enables an attacker to reconstruct the initial $128\,\mathrm{bit}$ key and thus decrypt the whole encrypted data set.

B. Side Channel Analysis

Side Channel Analysis (henceforth SCA) refers to a technique used to break encryption schemes by using data indirectly generated by the implementation of the cryptographic algorithm, instead of attacking the algorithm itself. Even a theoretically perfectly safe cryptographic algorithm can be subject to SCA and be broken by it. SCA uses side channels such as the power consumption of an microprocessor, electro magnetic emissions or even emitted sound to reconstruct the entire or parts of the secret key used to encrypt the data. It can be possible to infer the secret key form side channel data due to data dependant program flows. That means, that the code contains conditional statements acting on the secret key data. For example a certain loop will only execute if the currently procressed key bit is zero, therefore the power consumption of the microprocessor will measurably increase. Furthermore, power modelling is used to predict an microprocessor's power consumption based on the secret key data. Power models such as Hamming Weight (a "1" consumes more energy to be processed than a "0") or Hamming Distance (bitwise comparison resulting in the number of different bits) are popular choices ... Therefore, to protect against SCA it is vital to pay attention not only on the theoretical safety of an encryption algorithm but also on its implementation.

1) Defenses against Side Channel Attacks - should it be included???:

II. State of the Art

State of the art of crypto attacks

A. The state of Side Channel Analysis

Side channel attacks were first developed in 1996 by Paul Kocher [1] in the form of timing attacks. Timing attacks exploit an implementation's execution flow dependence on secret key data, which results in a different execution time if a key bit is for example 0 rather than 1. Later, SCAs have been extended to infer a secret key by analysing a processor's power consumption and its dependence on secret key data.

1) Simple Power Analysis: Simple Power Analysis (henceforth SPA) is a SCA based on the power consumption of the target device. The target device (e.g. a smart card) performs multiple encryption or decryption operations during which the attacker measures the device's power consumption. The attacker does not know the plaintext, which makes it a ciphertext only attack (COA). Then, the attacker calculates a hypothetical power consumption for each possible key combination using for example the Hamming Weight model. Finally, he correlates the hypothetical power consumption to the measured power

Fabrizio Desantis, Chair of Security in Information Technology of the Technical University of Munich

traces and then ranks the hypothetical keys accordingly. The correct key should now be among the top ranked hypothetical keys. However, SPA is very reliant on clean power traces. Modern computers perform many operations in parallel and thus generate a lot of noise, which is why SPA is only used on simple devices like smart cards.

2) Differential Power Analysis: Differential Power Analysis (henceforth DPA) is similar to SPA because it also tries to extract the secret key from a device by using power traces taken during a operation. However, in contrast to the SPA, multiple power traces are taken during different device states. Power traces of the target device are also taken while the device is idle to determine the background noise. The noise level is then subtracted from the power traces taken during the cryptographic operation, resulting in a more distinct signal. Still, concurrent operations running parallel to the cryptographic operations worsen the quality of the power traces, but that can be averaged out if enough traces are taken. Thus, DPA can also be used on more complex devices than smart cards.

3) High-order Differential Power Analysis: High-order Differential Power Analysis (henceforth HO-DPA) is similar to DPA but additionally, further power traces are taken at multiple steps of the cryptographic operation. HO-DPA allows to attack implementations using masking to hide cryptographic operations, by taking power traces at the mask generation, thereby demasking the final encryption operation. However, due to taking many different traces and correlating them with one another, HO-DPAs are highly complex and not used when a DPA or a SPA could also viably attack an implementation.

4) Template Attacks: A Template Attack is a very targeted attack on an implementation of a cryptographic algorithm. The premise is, that the attacker only has restricted access to the target machine, but ubiquitous access to similar machines containing a varying secret key. The attacker can now execute encryption operations while varying the secret key and measuring the machine's power consumption. The goal is to acquire a large labeled data set, mapping power consumption to secret key of this specific machine. Afterwards, the attacker trains a machine learning model, such as a Support Vectore Machine (SVM) with the model to be able to determine the secret key by using the power consumption. In the end, the attacker obtains one or two powertraces of the target machine during an encryption or decryption operation and feeds the powertrace into the SVM for analysis. The SVM will now output a potential secret key.

III. Theory

A. Convolutional Neural Networks

B. Stacked Auto-Encoders

C. Recurrent Neural Networks

D. Long and Short Term Memory (RNN)

Regular research papers need at least two additional sections here. One section for contributions and methods and one section for the results. For seminar papers these sections can be omitted.

IV. Experimental results

Show the results from the paper which covered the same topic.

V. Conclusion

Put the conclusions of the work here. The conclusion is like the abstract with an additional discussion of open points.

References

[1] "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," Cryptography Research, Inc., 607 Market Street, 5th Floor, San Francisco, CA 94105, USA., Tech. Rep., 1996.