ỨNG DỤNG HỌC MÁY TRONG TẤN CÔNG KÊNH KỀ

Tấn công kênh kề là phương pháp tấn công thám mã nguy hiểm và phổ biến hiện nay. Trong các phương pháp tấn công kênh kề, tấn công mẫu là phương pháp đem lại hiệu quả cao, phổ biến nhất và được các kẻ tấn công sử dụng để khôi phục khóa thiết bị. Tuy nhiên, phương pháp tấn công mẫu lại tốn thời gian thực hiện và lưu trữ khối lượng bản mẫu cần thiết để thực hiện giai đoạn xử lý trước tấn công. Bài báo này cung cấp cho bạn đọc về tấn công mẫu, cách thức thực hiện và ưu, nhược điểm của phương pháp này, từ đó đưa ra các biện pháp khắc phục của tấn công kênh kề bằng cách sử dụng học máy.

TS. Phạm Văn Tới[1], Lê Thảo Uyên

[1] Phòng thí nghiệm trọng điểm An toàn thông tin, Bộ Tư lệnh 86

ĐÔI NÉT VỀ TẤN CÔNG KÊNH KẺ

Tấn công kênh kề (side-channel attack - SCA) là phương pháp tấn công mạnh mẽ và phổ biến hiện nay chống lại quá trình triển khai mã hóa. Mục đích của phương pháp tấn công này là phân tích các dữ liệu, nguyên tố, các giao thức, mô-đun và các thiết bị trong hệ thống [1]. Các tấn công kênh kề sử dụng thông tin bị rò rỉ thu được trong quá trình thiết bị hoạt động. Ví dụ, kẻ tấn công có thể giám sát năng lượng tiêu thụ hoặc bức xạ điện từ phát ra từ một thẻ thông minh trong khi nó thực hiện các hoạt động bảo mật như giải mã và tạo chữ ký. Kẻ tấn công cũng có thể đo thời gian cần thiết để thực hiện quá trình mã hóa, phân tích một thiết bị mật mã khi xảy ra lỗi. Trong thực tế, việc thu thập rò rỉ và các vết (trace) có thể tiến hành dễ dàng bằng nhiều biện pháp khác nhau. Tuy nhiên, công việc phân tích dữ liệu và trace lại tương đối phức tạp [2].

Các phương pháp tấn công kênh kề phổ biến như: Tấn công thời gian (timming attack); Tấn công tiêm lỗi (fault injection attack); Tấn công phân tích năng lượng; Tấn công phân tích điện từ (electromagnetic analysis); Tấn công mẫu (template attack).

Tấn công mẫu

Tấn công mẫu là tập hợp con của các phương pháp tấn công sử dụng bản mẫu (profiling attack), trong đó kẻ tấn công tạo một bản mẫu (profile) của một thiết bị bị tấn công và áp dụng các bản mẫu này để tìm ra khóa bí mật trong đó.

Để thực hiện một cuộc tấn công mẫu, kẻ tấn công phải có quyền truy cập vào một bản sao khác của thiết bị được bảo vệ. Sau đó kẻ tấn công cần thực hiện rất nhiều công việc xử lý để tạo ra mẫu đúng như mong muốn. Trong thực tế, điều này có thể mất rất nhiều trace năng lượng. Tuy nhiên, ưu điểm của tấn công mẫu là chỉ cần rất ít số lượng mẫu đã qua xử lý đã có thể hoàn thành cuộc tấn công và khôi phục khóa K, thậm chí là từ một trace duy nhất [3].

Có 4 bước để tấn công mẫu:

Bước 1: Sử dụng bản sao của thiết bị được bảo vệ, ghi lại số lượng lớn các trace bằng nhiều cách khác nhau (bao gồm bản rõ và khóa), đảm bảo rằng có đủ trace được ghi lại cung cấp thông tin về từng giá trị khóa con.

Bước 2: Tạo mẫu cho thiết bị. Mẫu này cần được xem xét kĩ lưỡng những điểm đặc biệt trong các trace năng lượng và sự phân bố đa biến của các trace đó tại mỗi điểm.

Bước 3: Trên thiết bị chuẩn bị tấn công, ghi lại số lượng nhỏ các trace năng lượng, lưu ý sử dụng nhiều bản rõ.

Bước 4: Áp dụng mẫu cho các trace tấn công. Đối với mỗi khóa con, cần chú ý đến giá trị mà có khả năng là khóa con chính xác và tiếp tục cho đến khi chìa khóa được khôi phục.

Khi thực hiện ghi lại các trace năng lượng, cần chú ý đến nhiễu ảnh hưởng đến năng lượng của thiết bị đó. Giả sử X_0 là giá trị công suất thực của thiết bị, N là giá trị nhiễu tuân theo phân bố Gaussian, như vậy giá trị trace năng lượng thu được sẽ là [4]:

$$X = X_0 + N \quad (1)$$

Giá trị X và N được lấy ngẫu nhiên theo hàm phân phối Gaussian:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad (2)$$

Trong đó, μ - giá trị trung bình, σ - độ lệch chuẩn.

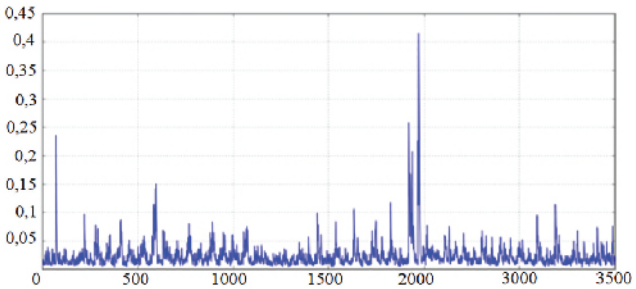
Như vậy, các mẫu trace thu được sẽ là tập hợp các phân phối xác suất mô tả các trace năng lượng với nhiều khóa khác nhau. Ví dụ, nếu một mẫu có khóa K_n, tập hợp các trace sẽ là hàm $f_{(Kn)}(X)^n$. Tập hợp các trace này sẽ được phân tích, sau đó lựa chọn ra những trace phù hợp để tiến hành tấn công mẫu. Tuy nhiên, tập hợp các trace này vẫn quá lớn để có thể lựa chọn, đưa ra được chính xác các điểm quan trọng trong mỗi tập hợp trace đó.

Có một số cách để chọn những điểm quan trọng nhất trong mỗi tập hợp trace. Phương pháp đơn giản nhất hay được sử dụng ở đây là tính tổng các điểm khác biệt. Đối với mỗi khóa K và mẫu i, thu được T_k trace trên mỗi khóa K, năng lượng trung bình sẽ được tính theo công thức:

$$M_{k,i} = \frac{1}{T_k} \sum_{j=1}^{T_k} t_{j,i} \quad (3)$$

Sau khi tìm được năng lượng trung bình, cần tính toán độ lệch tuyệt đối theo từng cặp, điều này sẽ tạo ra một trace có các đỉnh mà các mẫu thường khác nhau:

$$D_i = \sum_{k_1,k_2} |M_{k_1,i} - M_{k_2,i}| \quad (4)$$



Hình 1. Ví dụ về tổng số lệch các trace

Các đỉnh D_i hiển thị những điểm quan trọng nhất, từ đây sẽ được lựa chọn để xây dựng bộ mẫu cho các khóa, từ đó có thể khôi phục được khóa K.

Trong giai đoạn tấn công, giả sử chúng ta có A các trace năng lượng của mẫu thu thập được trong bước tiền xử lý ở trên. Các giá trị mẫu có giá trị

$$a_{j,si} \ (1 \le j \le A).$$

Đầu tiên, chúng ta sẽ tiến hành tấn công mẫu cho một trace duy nhất. Các bước tấn công như sau:

Bước 1: Đặt các giá trị theo dõi tại các điểm quan trọng vào một vector.

$$a_j = \begin{bmatrix} a_{j,1} \\ a_{j,2} \\ a_{j,3} \\ . \\ . \end{bmatrix}$$

Bước 2: Tính toán hàm mật độ xác xuất cho mỗi lần đoán chính xác giá trị của các trace này và lưu chúng lại để sử dụng các bước sau.

$$p_{k,j} = f_k(a_j)$$

Bước 3: Lặp lại hai bước này cho tất cả các trace.

Quá trình này sẽ cung cấp cho chúng ta một mảng các giá trị $p_{k,j}$, trong đó biểu thị các trace, khả năng thu được khóa K như thế nào.

Bước 4: Kết hợp các giá trị $p_{k,j}$ để chọn ra khóa K nào là phù hợp nhất.

$$P_k = \prod_{j=1}^{A} p_{k,j}$$

Ưu, nhược điểm của tấn công mẫu

Tấn công mẫu là phương pháp thám mã đang được sử dụng phổ biến hiện nay. Ưu điểm của tấn công mẫu là: Nó là phương pháp chính để đánh giá độ an toàn của thiết bị đối với tấn công kênh kề; Cần ít số trace để tìm được khóa của thiết bị tấn công.

Tuy nhiên, tấn công mẫu vẫn có nhược điểm, đó là cần có một thiết bị là bản sao của thiết bị tấn công. Với thiết bị này người tấn công có toàn quyền kiểm soát và sử dụng để xây dựng bộ mẫu.

ỨNG DỤNG HỌC MÁY TRONG TẤN CÔNG KÊNH KỀ

Với sự phát triển của trí tuệ nhân tạo (AI) và học máy (machine learning) trong những năm gần đây đã giúp cho cách tiếp cận phương pháp tấn công này trở nên đơn giản và dễ thực hiện hơn. Bài toán tấn công mẫu có thể xem như một bài toán phân loại do đó nó có thể được thực hiện thông qua học máy. Pha lập mẫu tương ứng với pha Huấn luyện, Pha tấn công tương ứng với pha Kiểm tra trong học máy.

Khi sử dụng học máy cho tấn công mẫu thì không cần giả định về đặc điểm phân phối năng lượng của thiết bị (thường là giả định là Gaussian đa biến).

Tấn công kênh kề sử dụng học máy

Một số nghiên cứu gần đây tập trung vào tấn công sử dụng bản mẫu dựa trên học máy để chống lại việc triển khai mật mã. Những đóng góp này chủ yếu tập trung vào hai thuật toán là Support Vector Machine (SVM) và Random Forest (RF). Kết quả thực tế trên một số tập dữ liệu đã chứng minh được khả năng khôi phục khóa thành công của các tấn công này. Bên cạnh đó, các tác giả [7] đã chỉ ra rằng tấn công dựa trên SVM hoạt động tốt hơn so với tấn công mẫu thông thường đối với các trace có độ nhiễu cao.

Các kỹ thuật dựa trên học máy thường dựa vào việc xử lý trước tập dữ liệu để xác định các đặc tính có liên quan, không giống như tấn công mẫu và các dạng khác của tấn công bản mẫu. Chủ yếu, tấn công kênh kề dựa trên học máy khai thác cùng một tiêu chí phân biệt (tức là sự phụ thuộc giữa dữ liệu và thời điểm thống kê khi rò rỉ) như tấn công mẫu. Có các điểm khác biệt chính giữa hai loại tấn công này như sau:

- Tấn công mẫu đưa ra xấp xỉ phân phối dữ liệu theo phân phối Gaussian đa biến (hay còn gọi là giả định rò rỉ Gaussian) mà có tham số (tức là vectơ trung bình và ma trận hiệp phương sai) được ước lượng trong giai đoạn lập bản mẫu. Điều này có nghĩa là các thời điểm thống kê của phân phối rò rỉ có bậc lớn hơn 2 là không được khai thác, điều này có thể làm cho tấn công trở nên kém tối ưu và thậm chí không hiệu quả trong một số trường hợp.

- Tấn công kênh kề dựa trên học máy không đưa ra giả định về phân phối dữ liệu và xây dựng phân loại trực tiếp từ tập dữ liệu thô. Mặc dù rò rỉ Gaussian là một giả định khá thực tế đối với kênh kề, nhưng việc áp dụng các kỹ thuật thống kê đối với những phân phối không thể biết, dường như là một cách tiếp cận hợp lý.

- Những lợi thế của học máy so với tấn công mẫu là: Giảm độ phức tạp tính toán của việc phân loại và tránh huấn luyện các đặc tính không liên quan; Việc lựa chọn đặc tính có thể được thực hiện bằng nhiều cách khác nhau với tỷ lệ thành công khác nhau trong các trường hợp khác nhau.

Perceptron

Perceptron là mô hình mạng nơ-ron đơn giản nhất. Nó là bộ phân loại tuyến tính sử dụng thuật toán học máy để điều chỉnh trọng số (weight), nhằm giảm thiểu hàm mất mát (loss function). Trọng số perceptron được khởi tạo ở số không hoặc giá trị ngẫu nhiên nhỏ, sau đó trong giai đoạn huấn luyện được học và điều chỉnh theo tập dữ liệu bản mẫu. Ví dụ, áp dụng thuật toán suy giảm độ dốc (gradient descent) nhằm mục tiêu tìm hoặc học các trọng số kết nối tối ưu, để di chuyển các đầu ra perceptron càng gần các nhãn/điểm chính xác càng tốt.

Perceptron nhiều lớp (Multilayer Perceptron - MLP) là sự kết hợp các perceptron nhằm xây dựng bộ phân loại cho các tập dữ liệu phức tạp hơn. MLP được tạo ra từ ba loại lớp khác nhau: lớp đầu vào, lớp ẩn và lớp đầu ra.

Trong một số công trình gần đây, MLP đã được áp dụng để thực hiện thành công khôi phục khóa trong tấn công kênh kề. Trong đó, một nghiên cứu đã trình bày tấn công kênh kề dựa trên mạng nơ-ron để phá vỡ triển khai thuật toán AES có mặt nạ trong cuộc thi DPA V4. Các tác giả [11] cũng giả định rằng đối thủ có quyền truy cập vào các giá trị mặt nạ trong giai đoạn lập bản mẫu. Theo giả định này, tấn công bao gồm xác định mặt nạ bằng cách áp dụng khôi phục mặt nạ dựa trên mạng nơ-ron. Sau đó, tấn công thứ hai cũng dựa trên mạng nơ-ron được thực hiện để khôi phục khóa bí mật với một trace. Mặc dù nghiên cứu này đưa ra

nhiều kết quả tốt, nhưng giả định của nghiên cứu này là không phải lúc nào cũng đáp ứng được trong thực tế.

Support Vector Machine (SVM)

SVM là một thuật toán phổ biến được sử dụng để thay thế cho tấn công mẫu. Nó là bộ phân loại tuyến tính thiết lập một mặt phẳng để tách các lớp dữ liệu theo cách tùy chọn, nhằm mục đích tìm đường tách biệt tối ưu giữa các lớp trong dữ liệu. Điểm quan trọng nhất của SVM là phương pháp này làm giảm số lượng trace cần có để phân tích trước khi tấn công. Điều này rất hữu ích khi phải xử lý số lượng lớn dữ liệu đầu vào.

Đối với tấn công kênh kề, một số công trình đã nghiên cứu việc sử dụng SVM nhằm thực hiện các cuộc tấn công phá vỡ các triển khai mật mã. Theo tài liệu [7], có nghiên cứu đã chứng minh được rằng khi tỷ lệ tín hiệu trên nhiễu (Signal-to-Noise Ratio) của tập dữ liệu được nhắm mục tiêu là rất thấp, thì cuộc tấn công dựa trên SVM vượt trội hơn so với cuộc tấn công mẫu.

Random Forest (RF)

RF bao gồm nhiều cây quyết định, mỗi cây làm việc với một tập con khác nhau của tập dữ liệu huấn luyện. Trên đỉnh của tất cả các cây, đầu ra toàn cục được tính toán thông qua đa số phiếu bình chọn (vote) giữa các kết quả đầu ra của các cây phân loại này. RF cũng đã được áp dụng thành công trong tấn công kênh kề để phá vỡ triển khai mật mã.

Tấn công kênh kề sử dụng học sâu

Học sâu (Deep learning) là kỹ thuật thuộc phương pháp học máy dựa trên một tập hợp các thuật toán để cố gắng mô hình dữ liệu trừu tượng hóa ở mức cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp, hoặc bằng cách khác bao gồm nhiều biến đổi phi tuyến.

Một trong những điều hứa hẹn của học sâu là thay thế các tính năng thủ công bằng các thuật toán hiệu quả đối với học không có giám sát hoặc bán giám sát và tính năng phân cấp [9]. Các nghiên cứu trong lĩnh vực này cố gắng thực hiện các đại diện tốt hơn và tạo ra các mô hình để tìm hiểu các đại diện này từ dữ liệu không dán nhãn quy mô lớn.

Nhiều kiến trúc học sâu khác nhau như mạng nơ-ron sâu (Deep neural network), mạng nơ-ron tích chập sâu (Convolutional neural network - CNN), mạng niềm tin sâu (deep belief networks) và mạng nơ-ron hồi quy (Recurrent neural network - RNN) đã được áp dụng cho các lĩnh vực như thị giác máy tính, tự động nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên, nhận dạng âm thanh,... Chúng đã được chứng minh là tạo ra các kết quả rất tốt đối với nhiều nhiệm vụ khác nhau.

Mạng nơ-ron tích chập sâu (CNN)

CNN là một loại mạng nơ-ron được xây dựng bằng cách xếp chồng các lớp sau: lớp tích chập, lớp Max Pooling và lớp SoftMax. Việc học các bộ lọc cho phép trích xuất các đặc tính cấp cao cụ thể từ dữ liệu. Do đó, bước này có thể được sử dụng như một kỹ thuật giảm kích thước chiều dữ liệu hoặc lựa chọn Points of Interest (ví dụ thuật toán PCA). Dựa trên điều này, có thể đánh giá hiệu quả của tính năng trích xuất đặc tính của CNN trong việc lựa chọn các điểm có chứa nhiều thông tin nhất, để thực hiện tấn công khôi phục khóa thành công.

Việc học các bộ lọc cho phép trích xuất các đặc tính cấp cao cụ thể từ dữ liệu. Do đó, bước này có thể được sử dụng như một kỹ thuật giảm kích thước chiều dữ liệu hoặc lựa chọn Points of Interest (ví dụ thuật toán PCA). Dựa trên điều này, có thể đánh giá hiệu quả của tính năng trích xuất đặc tính của CNN trong việc lựa chọn các điểm có chứa nhiều thông tin nhất, để thực hiện tấn công khôi phục khóa thành công.

Bộ tự mã hóa xếp chồng

Bộ tự mã hóa xếp chồng (Stacked Auto-Encoders) là mạng nơ-ron với nhiều lớp được huấn luyện bằng cách tuân theo một quy trình cụ thể. Quy trình này bao gồm huấn luyện từng lớp một cách độc lập, sử dụng đầu ra của lớp trước làm đầu vào cho lớp hiện tại. Mỗi lớp bao gồm một bộ mã hóa và một bộ giải mã, cả hai đều là một lớp dày đặc (tức là lớp được kết nối đầy đủ). Vai trò của bộ mã hóa là tạo ra các đặc tính cấp cao hơn từ các đầu vào. Trong khi đó, vai trò của bộ giải mã là tái tạo lại các đầu vào từ các đặc tính trung gian mà bộ mã hóa đã học.

Ở cuối các lớp của bộ tự mã hóa xếp chồng, một bộ phân loại SoftMax thường được thêm vào để dự đoán lớp của đầu vào bằng cách sử dụng các đặc tính cấp cao được trích xuất của lớp cuối đó. Mỗi lớp trong số này (bao gồm cả lớp SoftMax) được huấn luyện tuần

tự. Nhưng khi lớp cuối cùng được huấn luyện, một quá trình huấn luyện toàn cục sử dụng thuật toán lan truyền ngược (back-propagation) sẽ được thực hiện. Kỹ thuật này được gọi là tinh chỉnh (fine tuning).

Như CNN, bộ tự mã hóa là bộ trích xuất các đặc tính. Vai trò của nó là xây dựng các đặc tính cấp cao dễ sử dụng hơn trong tác vụ lập bản mẫu. Nhiệm vụ này đặc biệt có ý nghĩa đối với tấn công kênh kề, khi phương pháp lựa chọn đặc tính là rất quan trọng.

Mạng nơ-ron hồi quy (RNN)

RNN được sử dụng cho những dữ liệu mà thông tin giống nhau được trải rộng trên nhiều mẫu thời gian. Do đó, thay vì giả định rằng các thành phần của vectơ đầu vào là độc lập, thì mỗi nơ-ron sẽ suy ra đầu ra của nó từ cả đầu vào và đầu ra hiện tại của các đơn vị trước đó. Kỹ thuật RNN có thể được áp dụng đối với tấn công kênh kề vì sự rò rỉ có thể lan rộng trong một số mẫu thời gian.

Mạng bộ nhớ dài - ngắn hạn (LSTM)

LSTM là một đơn vị dựa trên RNN. Ban đầu nó được giới thiệu để giải quyết các vấn đề khi sử dụng RNN, chủ yếu là độ dốc bị biến mất hoặc lên quá cao. Nó cho phép mạng xử lý độ trễ thời gian dài giữa các chuỗi thời gian liên quan của tập dữ liệu đã xử lý. Để làm như vậy, một trạng thái của ô (hay còn gọi là ô nhớ) được thêm vào bên trong mỗi đơn vị. Nó chứa một số thông tin thống kê (ví dụ: giá trị trung bình, phương sai) được tính toán dựa trên chuỗi thời gian dữ liệu đã được xử lý trước đó. Ô này có thể được ghi lên hoặc xóa tùy thuộc vào mức độ liên quan của thông tin được lưu trữ. Quyết định ghi lên ô hoặc xóa được thực hiện bởi một mạng nơ-ron nhỏ.

Đối với tấn công kênh kề, tính năng này có thể được sử dụng khi muốn thực hiện cuộc tấn công bậc cao hơn, chẳng hạn như khi đối thủ phải kết hợp nhiều mẫu thời gian trễ để phá vỡ các triển khai mặt nạ.

KẾT LUẬN

Bài báo đã trình bày phương pháp tấn công mẫu trong tấn công kênh kề, phân tích và đánh giá ưu nhược điểm của phương pháp tấn công mẫu. Bài báo còn đưa ra các bước tiến hành tấn công mẫu và chuẩn bị mẫu trước khi thực hiện tấn công.

Từ những ưu nhược điểm đã nêu, bài báo đã giới thiệu cách tiếp cận mới đối với phương pháp tấn công mẫu, đó là sử dụng học máy cho việc phân loại các mẫu và loại bỏ được các mẫu không liên quan. Điều này rất quan trọng trong bước tiền xử lý trước khi tấn công vì nó sẽ làm giảm thời gian thực hiện và tối ưu hóa lưu trữ dữ liệu.❖

TÀI LIỆU THAM KHẢO

1. E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen. A Side-Channel Analysis Resistant Description of the AES S-box. In FSE 2005, vol. 3557 of LNCS, pages 413–423. Springer, 2005.
2. S. Chari, J. R. Rao, and P. Rohatgi. Template Attacks. In CHES, volume 2523 of LNCS, pages 13–28. Springer, August 2002. San Francisco Bay (Redwood City), USA.
3. Z. Chen and Y. Zhou. Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage. In CHES, volume 4249 of LNCS, pages 242–254. Springer, October 10-13 2006. Yokohama, Japan, http://dx.doi.org/10.1007/11894063_20.
4. O. Choudary and M. G. Kuhn. Efficient Template Attacks. Cryptology ePrint Archive, Report 2013/770, 2013.
5. F.-X. Standaert, C. Archambeau, Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages, in the proceedings of CHES 2008, LNCS, vol 5154, Washington DC, USA, August 2008.
6. G. Hospodar, B. Gierlichs, E. De Mulder, I. Verbauwhede, and J. Vandewalle. Machine learning in side-channel analysis: a first study. Journal of Cryptographic Engineering, 1(4):293–302, 2011.
7. A. Heuser and M. Zohner. Intelligent Machine Homicide - Breaking Cryptographic Devices Using Support Vector Machines. In W. Schindler and S. A. Huss, editors, COSADE, volume 7275 of LNCS, pages 249–264. Springer, 2012.
8. L. Deng and D. Yu. Deep learning: Methods and applications. Found. Trends Signal Process., 7(3–4):197–387, June 2014.
9. M. Hermans and B. Schrauwen. Training and analysing deep recurrent neural networks. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 190–198. Curran Associates, Inc., 2013.
10. L. Lerman, R. Poussier, G. Bontempi, O. Markowitch, and F. Standaert. Template attacks vs. machine learning revisited (and the curse of dimensionality in sidechannel analysis). In S. Mangard and A. Y. Poschmann, editors, Constructive Side-Channel Analysis and Secure Design - 6th International Workshop, COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers, volume 9064 of Lecture Notes in Computer Science, pages 20–33. Springer, 2015.
11. R. Gilmore, N. Hanley, and M. O'Neill. Neural network based attack on a masked implementation of aes. In Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on, pages 106–111, May 2015.