L-010

マイコン実装した AES 暗号に対する CNN を用いた最終ラウンド鍵の解析
Analyzing Final Round Key of AES Implemented on Microcomputer using CNN

小杉 聡志 ‡　城市 翔 ‡　生田 健 ‡　日下 卓也 ‡　野上 保之 ‡　高橋 規一 ‡
Satoshi Kosugi　Sho Joichi　Ken Ikuta　Takuya Kusaka　Yasuyuki Nogami　Norikazu Takahashi

1. まえがき

暗号によって守られた IoT 機器に対する攻撃方法の 1 つとして，サイドチャネル攻撃がある．サイドチャネル攻撃は，ハードウェア上での暗号化処理時に発生する副次的な情報を観測する，あるいはハードウェアに直接手を加えることによって，外部からの鍵の推定を可能とする攻撃である．代表的なサイドチャネル攻撃の 1 つとして，電力解析攻撃という手法が知られている．電力解析攻撃は，暗号モジュールが暗号処理を行う際に発生する電源電圧波形であるサイドチャネル波形を測定し，その電圧値を解析することで秘密鍵を解析する攻撃手法である．

電力解析攻撃では，膨大な数のサイドチャネル波形を解析する必要がある．ディープラーニング[1]の分野は，こうした膨大なデータの解析に有用である．本研究では，ニューラルネットワーク[2]の一種である畳み込みニューラルネットワーク (CNN) の分類能力を用いて波形の解析を行い，その精度を求めた．

本稿では，暗号モジュールに搭載した AES[3]に対して，CNN を用いた新たな電力解析攻撃の手法を提案する．具体的には，8 ビットマイコンの一種である ATmega328p[4]に AES を実装し，PC オシロスコープを用いて暗号処理中の電圧波形を観測する．その後，観測した電圧波形の最終ラウンドの処理にあたる箇所を，CNN の学習モデルに学習させることにより最終ラウンド鍵の解析を行う．さらに，この提案手法を実装した場合の実験を行い，実際に鍵の推定を行った際の攻撃の成功率を求めることで，この提案手法の脅威を検証した．

2. 準備

本研究では，実際に AES (Advanced Encryption Standard) と呼ばれる暗号を実装したマイコンに対して，暗号処理中に発生する電圧波形を CNN により解析することで鍵を特定する．本章では，AES とサイドチャネル攻撃について述べた後，電圧波形の解析で用いたニューラルネットワークの基本的な構造について述べる．

2.1 AES (Advanced Encryption Standard)

AES とは，米国の旧国家暗号規格である DES (Data Encryption Standard) に代わる標準暗号として制定された暗号方式である．本節では，AES のアルゴリズムや各処理の内容について触れる．

2.1.1 AES のアルゴリズム

AES では，まず元のデータを 128 ビットごとに分割し，その 128 ビットをさらに 1 バイト (8 ビット) ごとに分割する．このようにして得られた 16 バイトは 4 × 4 の行列として扱われ，このまとまりごと SubBytes，ShiftRows，MixColumns，AddRoundKey の 4 つの処理が行われる．この 4 つの処理の流れは，図 1 に表す通りである．SubBytes，ShiftRows，MixColumns，AddRoundKey の一連の 4 つの処理を 1 ラウンドとして，何度か繰り返す．繰り返し回数は鍵長によって決定され，128 ビットなら 10，192 ビットなら 12，256 ビットなら 14 となっている．ただし，最終ラウンドのみ MixColumns の処理は行われない．

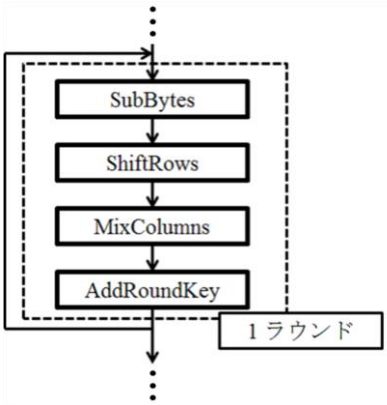

図 1　AES の処理の流れ

2.1.2 AES の各処理の説明

まず，SubBytes ではデータの各バイトに S-Box と呼ばれる LUT を用いた変換を適用する．ShiftRows では，4× 4 の行列とみなされているデータを，行ごとに 0〜3 バイトほど巡回シフトする．MixColumns では，行列とみなされているデータの各行ごとに演算を行う．AddRoundKey では，データとラウンド鍵ごとの XOR を取り，出力を得る．

2.1.3 AES のラウンド鍵

AES はラウンドごとに異なる鍵の値をもつ．この各ラウンドがもつ鍵のことをラウンド鍵といい，あらかじめ決められた鍵を親として鍵拡張を行うことで，ラウンドの数だけ生成される．

2.1.4 AES の暗号利用モード

AES のようなブロック暗号に対しては，複数の暗号利用モードが規格化されている．各モードでは，各ブロックの暗号化にその前のブロックの暗号文を用いるなどの機密性を保つための工夫がなされている．本研究では，こういった工夫はなされていない，各ブロックが独立して暗号化されるモード (ECB モード) に対して攻撃を行った．

2.2 サイドチャネル攻撃

暗号攻撃法には，考えうるすべての可能性を試す総当たり攻撃，何かしらのアルゴリズムを使用し数学的に解読を行うショートカット方式などがある．これらのほかに，暗号実装が行われているハードウェアの様子を観察したり，直接働きかけたりなどするサイドチャネル攻撃と呼ばれる攻撃法が存在する．

もっとも有名なサイドチャネル攻撃の一つに，電力解析攻撃といわれる攻撃法がある．これは，ハードウェア上で

暗号化や復号処理をする際の消費電力を観測し，それを解析することで暗号鍵を解析するものである．このように，サイドチャネル攻撃ではハードウェア上で暗号処理をする際に発生する消費電力，電磁波，時間などといった副次的な情報を解析することで鍵を特定する．

2.3 ニューラルネットワークの概要

ニューラルネットワークとは，いくつかある人工知能技術の 1 つであり，脳機能に見られる特性に類似した数理的モデルである．ニューラルネットワークとして設計されたモデルは一連のスカラ積による処理と，活性化関数と呼ばれる非線形変換で構成され，複数の処理層をもつ．

数年前まで，コンピュータの計算能力の不足や学習の際の勾配消失といった問題のために，深層なニューラルネットワークを訓練することは困難だった．しかし，GPU（Graphics Processing Unit）の活用や新たな活性化関数の提案によって，ニューラルネットワークは多くの層をもつことができるようになり，それに伴ってより多くの訓練データセットを学習することが可能となった．

これらは深層なニューラルネットワークと呼ばれる，深層学習の技術として知られている．

本節では，各処理層についての概要と，複数の層によって得られる深層なニューラルネットワークについて述べる．

2.3.1 各処理層

ニューラルネットワークは，一般に入力層，出力層と，複数の中間層によって構成されており，ニューロン数の設計方法は各層により異なる．

入力層は，ニューラルネットワークにおいて，入力データとネットワークのほかの部分との間をつなぐ層に過ぎない．したがって，この層から出力される情報は，単に入力データそのものである．そのため，ニューロンの数は入力データのベクトルの数と同一に設計する必要がある．

中間層は，ニューラルネットワークのモデルが非線形に分離する必要のあるデータセットに適合するために，モデルに何らかの非線形性を導入することを目的として設計される．中間層のもつ非線形性は，後述する活性化関数により表現が異なる．中間層の数は，入力となるデータセットの非線形性や複雑さに応じて増やすことができる．また，中間層内のニューロンの数も入力データの複雑さによって任意の値に設計することができる．ただし，中間層内のニューロンの数に関しては，学習しなければならない入力データが線形のものに近い場合，膨大な数のニューロンを使用すると，モデルの予測精度の低下を招きうることが分かっている． 一方で，複雑な入力データの場合，中間層に設計したニューロンの数が少なすぎると，正確な解を得られないことがある．

出力層はネットワークの最後の層に位置する．後述する多クラス分類において，この層の出力は，設計者の用意したクラスに直接マップされる．

2.3.2 ニューロン

ニューロンとは，シナプスの結合により情報を伝達する脳の機能を模して設計されたニューラルネットワークにおいて，基本単位となる要素である．ニューロンは 1 つ以上の情報を入力として受け取る．各入力には重みが掛けられ，その総和を活性化関数と呼ばれる非線形関数に通すことにより出力を生成する．活性化関数については次節で述べる．ニューロンは前節の処理層の中に含まれており，各層内のニューロンが，その次の層のニューロンに情報を受け渡すことで，情報の伝達が行われる．

n 個の入力をもつニューロンを考える．入力信号を x_1 から x_n とし，ニューロン間の結合の強さを表す重みを w_1 から w_n，バイアス項を b，活性化関数を ϕ，出力を y とすると，以下の 式で示される

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1}$$

2.3.3 重みとバイアス

重みとは，情報の伝達が行われるニューロン間の，結合の強さを表すものである．結合が強いほど重みは大きくなり，より多くの情報を伝達することができる．n 個のニューロンの出力を x_1 から x_n とし，次のニューロンにその値が受け渡される送電線の重みをそれぞれ w_1 から w_n とするモデルを考える．

受け取った入力から計算された値が閾値 θ を超えると発火するので，出力は(2)式のように示すことができる．

$$y = \begin{cases} 1 (w_1 x_1 + \cdots + w_n x_n \geq 0) \\ 0 (w_1 x_1 + \cdots + w_n x_n < 0) \end{cases} \tag{2}$$

ここで，ネットワークの出力 y は (3) 式のように示すことが できる．

$$y = \varphi(w_1 x_1 + \cdots + w_n x_n - \theta) \tag{3}$$

この $\varphi(x)$ はステップ関数とよばれる活性化関数である．式 を和で統合して扱いやすくするために，$b = -\theta$ とおく．また， 重み $w_1, w_2, \ldots, w_n$ と入力 $x_1, x_2, \ldots, x_n$ の積和の部分はベク トルの内積を用いて表すことができるので，列ベクトル X と W として (4) 式のように考えることができる．

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad W = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \tag{4}$$

これらを用いて，最終的な出力を (5) 式のように表すことができる．

$$y = \varphi(W^T X + b) \tag{5}$$

これはネットワークを一般形で示した式であり，式内の W を重みベクトル，b をバイアスとよぶ．

2.3.4 活性化関数

活性化関数とは，ニューラルネットワークにおいて，ニューロンの線形結合後の非線形変換を行う関数の総称である．ニューロン間の伝送線には重みづけがされ，その入力と重みの積和が活性化関数に渡される．

シグモイド関数は，後述する多クラスの分類を行うニューラルネットワークを構成するための非線形な関数として，もっとも初期の時期から用いられていた活性化関数である．

$$\varphi(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

ReLU(Rectified Linear Unit, Rectifie)はランプ関数とも呼ばれる，区分線形関数の一種である．ReLU は指数関数のない単純な式で表されるため，シグモイド関数やその他の関数に比べて計算が高速化され，学習が速く進みやすくなるという利点がある．

$$\varphi(x) = max(1, x) \tag{7}$$

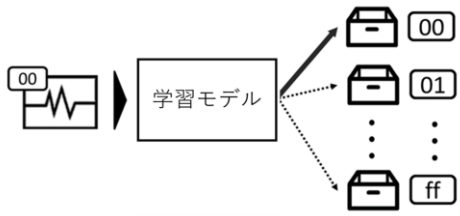

図 2　学習モデルが波形を分類し，鍵を推定する

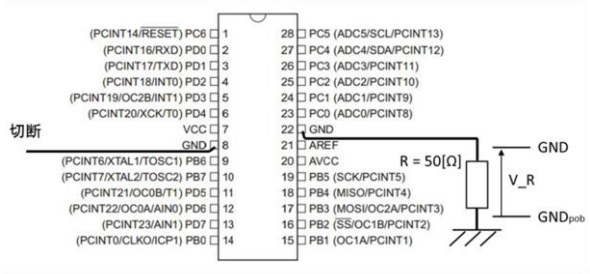

図 3　ATmega328p における測定箇所

2.3.5 多クラス分類

出力層にステップ関数を用いたモデルは，ニューロンが発火する，しないの 2 パターンのみで分類を行う．この分類可能なパターンをクラスという．この場合，出力は 2 クラスに分類されるため，これを指して 2 値分類とよぶ．これに加え，出力を 2 クラス以上のクラスに分類する場合，そのモデルは多クラス分類と呼ばれ，出力層に非線形な関数を用いた場合に実現できる．その関数のことをソフトマックス関数といい，シグモイド関数より導くことができる．多クラス分類を目的とするモデルにおいて，ソフトマックス関数は出力層の活性化関数として用いられる．ソフトマックス関数は，n 次元ベクトル $X = (x_1, x_2, \ldots, x_n)^T$ に対して，以下の式で与えられる．

$$\mathrm{softmax}(x)_i = \frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_j}} \quad (8)$$

2.3.6 畳み込みニューラルネットワーク (CNN)

CNN では，前節までの全結合層と呼ばれる層のほかに，畳み込み層とプーリング層を用いている．畳み込み層では，入力データに対してカーネルと呼ばれるフィルタ適用することで，フィルタ演算が行われる．フィルタ演算では，入力データに対して一定の間隔でカーネルのウィンドウをスライドさせながら適用し，それぞれの箇所でのカーネルの要素と入力の対応する要素の積和演算の値を得る．その値に対し，バイアス項を加算することで出力を得る．プーリング層では，Max プーリングや Average プーリングと呼ばれる，入力データの対象領域から最大値や平均値をとる演算が行われる．これは入力データの縦横の空間を縮小するための演算であり，これによりプーリング層は畳み込み層と違って学習するパラメータを持たない．

3. CNN を用いた攻撃の提案

ニューラルネットワークは，学習元となる訓練データを学習させることによって，訓練データには含まれていない未知のデータを推定できる．本章では，AES の最終ラウンド処理時の電圧波形を取得し，それを CNN のモデルに学習させることで，未知の電圧波形から鍵の推定を行う攻撃を提案する．

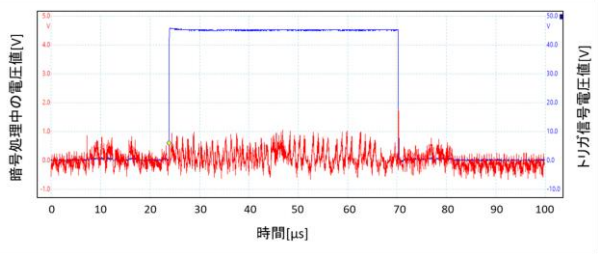

図 4　観測された波形の例

表 1　学習モデルが波形を分類し，鍵を推定する

	Type number	Configuration
PC Osilloscope	PicoScope 5224B	Resolution : 8, 12, 14, 15, 16 bits Bandwidth : 200 MHz Memory : 512 MS
Passive Probe	TA131	Bandwidth : 250MHz Input Voltage Max : 600V
Cryptgraphic module	Arduino Uno 8 bit Microcomputer	Clock frequency : 16MHz Harvard architecture

3.1 暗号処理中の波形観測と推定する鍵の詳細

本研究では，AES の暗号処理中の最終ラウンド鍵にあたる部分の電圧波形を観測する．観測波形をニューラルネットワークに解析させることで，どれほど正確に鍵を推定できるかを示す正答率の値を確率として求める．実験に用いるマイコンとして ATmega328p を使用した．ATmega328p を搭載した基盤である Arduino Uno にオシロスコープをつないで電圧波形を観測し，最終ラウンドの暗号処理にあたる電圧波形を取得する．

最終ラウンド鍵は前述のとおり 128 ビットあるが，ニューラルネットワークに学習させる際，学習のコストと推定のために用意するクラスを削減することで，学習に必要となる波形データ数をなるべく減らす工夫が必要である．そこで，本実験では，学習の際に最終ラウンド鍵を 1 バイトごとに 16 分割し，各バイトについて推定を行う．

この手法では，暗号化処理時に最終ラウンド鍵の先頭 1 バイトを固定する際，他の 2〜16 バイトについて考慮する必要がある．ここでは，2〜16 バイトと，鍵に対応する平文はすべてランダムに設定する．こうすることで，観測できる波形を現実的な環境で得られる波形とほぼ同一と考えることができる．

なお，1 つの平文を 1 つの鍵を用いて暗号化する．

この手法では，1 バイトのとりうる値を網羅するために，00〜ff までの 256 通りの鍵の値のパターンを考慮する必要がある．各パターンごとに，それぞれ 1000 波形ずつを十分な波形数として設定し，その合計として 256000 波形をサイドチャネル波形として用意する．そのうちの 8 割にあたる 204800 波形をニューラルネットワークの学習の訓練データとして用い，残りの 2 割に当たる 51200 波形を検証データとして学習の正確性の検証に用いる．

3.2 観測した波形を用いた CNN による鍵の解析方法

本研究では，AES の最終ラウンドの処理波形を CNN に学習させ，正しい鍵の推定を行うことを目的とする．

CNN の訓練データはそれぞれクラスラベルと関連付けたうえで学習させる必要がある．

前節で述べた通り，最終ラウンド鍵の先頭 1 バイトを 256 通り考慮して電圧波形を取得するため，対応付けのためのラベルもそれぞれ 256 通り用意する．これらを電圧波形と関連させて，訓練データセットとして学習させる．

この提案手法では，学習の検証用に用いられるデータを CNN がどの程度正しく推定できるかを検証する．電圧波形と鍵に何らかの相関があれば，ニューラルネットワークは電圧波形から正しい特徴を抽出して学習することができる．学習がうまくいくと，正しい鍵を推定できた試行は多くなるため，正答率も高い値を示す．以上のように，電圧波形と鍵の間に相関を読み取ることで正答率の値を求める．

4. 提案手法による実験

ニューラルネットワークは，事前に訓練データを学習させることによって，未知のデータに含まれる特徴を検出する教師あり学習である．本稿では，CNN に AES の最終ラウンド処理中の電圧波形を学習させることで，未知の電圧波形のデータから最終ラウンドの鍵を推定する手法を提案している．本章では，Arduino 上に実装した AES を用い，この手法による鍵の正答率を求める．

4.1 実験環境

本節では，実験に使用した機器や測定方法，測定機器の設定などを示す．

4.1.1 使用機器

本研究では 128 ビットの AES を実装した Arduino Uno の基板上にある ATmega328p の電源とグラウンドの間を測定する．ATmega328p が暗号処理をしている際に発生する電源電圧の測定箇所を図 3 に示す．この図では電源とグラウンド間に 50Ω の抵抗を接続し，PC オシロスコープを用いて観測されるサイドチャネル波形の大きさを制御している．

また，測定に使用した機器の詳細を表 1 に示す．

4.1.2 プログラム作成時の注意と観測されたサイドチャネル波形

鍵長 128 ビットの AES (ECB モード) の暗号化プログラムの作成時には Arduino 言語を使用し，最終ラウンドである 10 ラウンド目の暗号化処理中にトリガを立てることで，電圧波形の取得を容易にしたものを実装した．観測された波形の例を図 4 に示す．プログラム内で最終ラウンド鍵を直接指定することで，決められた鍵ごとの電圧波形の取得を可能としている．なお，本プログラムにおける SubBytes は LUT を使用し出力を求める手法を採用している．

CNN のモデルの設計には，Keras[5]と呼ばれるライブラリを用い，Python3 を使用した．Keras は，Python で書かれた，TensorFlow または CNTK，Theano 上で実行可能なニューラルネットワークライブラリである．

4.2 実験とその結果

本節では，最終ラウンド鍵の先頭 1 バイトについて，学習に用いた CNN の学習モデルとその結果について述べる．使用した学習モデルの構造は，「Conv (512) - MaxPooling (3) - Conv (512) - MaxPooling (2) - Affine (500) - Affine (400) - Affine (300)」となっている．ここで，Conv は畳み込み層とカーネルサイズの値，MaxPooling は Max プーリング層とプーリングサイズの値，Affine は全結合層とそのニューロン数の値を指す．これらの構造とパラメータの値は，入力層の次元数が 1000，出力層の次元数が 256 であることを加味して試行錯誤の末に得られたものである．条件下において取得した波形をもとに，本実験で得られた CNN の推定の正答率の変遷を図 5 に示す．

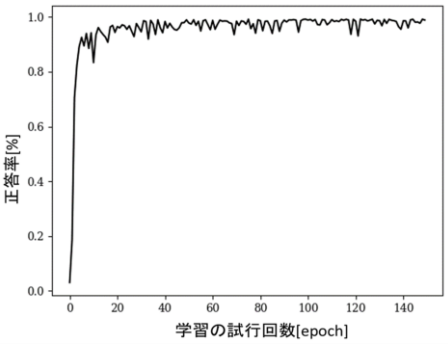

図 5　先頭 1 バイトに対する CNN の正答率

5. まとめ

本稿では，暗号モジュールに搭載した AES に対して，畳み込みニューラルネットワーク (CNN) を用いた新たな電力解析攻撃の手法を提案した．その手法とは，AES の暗号化処理中の電圧波形データを CNN のモデルに学習させることで，未知の電圧波形 データから最終ラウンド鍵を推定するものである．ここで，最終ラウンド鍵の値を分割して学習させることで，学習に必要な 訓練データ数をなるべく減らしている．実験では，128 ビット ある最終ラウンド鍵の値を 1 バイトずつに 16 分割して CNN に学習させることにより，その正答率を求めた．

実験結果の図 5 を見ると，学習の終盤ではほとんど頭打ちになっているため，学習が停滞した段階で学習を停止させ，その値を最終的な正答率の値とした．その結果，98.91%の正答率で鍵を推定できることが分かった．これに対して，畳み込み手法を用いないニューラルネットワークによる既存の手法[6]では正答率は 97.26%であったため，提案手法では既存の手法を上回る精度の学習モデルを作ることができたといえる．以上より，マイコン実装した AES に対する CNN を用いたサイドチャネル攻撃は有効であることが確認できた．

参考文献

[1] Deep learning website, http://deeplearning.net/tutorial/.
[2] 巣籠 悠輔, 『詳解 ディープラーニング TensorFlow・Keras による時系列データ処理』, マイナビ出版,2017.
[3] National Institute of Standards and Technology (NIST), “Advanced Encryption Standard (AES)”, FIPS publication 197, http://csrc.nist.gov/encryption/aes/index.htm, 2001.
[4] Atmel,“8AVR”,http://avr.jp/user/DS/PDF/mega88A.pdf, 5/2/2018
[5] Keras library, https://keras.io/.
[6] 小杉 聡志, 城市 翔,生田 健,日下 卓也,野上 保之,高橋 則一, “マイコン実装した AES 暗号に対するニューラルネットワークを用いた 最終ラウンド鍵の解析”, 電子情報通信学会技術研究報告, 3/7~8, 2019.