Biuletyn WAT
Vol. LVII, Nr 4, 2008

Analiza bezpieczeństwa implementacji sprzętowych blokowych algorytmów szyfrowania informacji

JERZY GAWINECKI, PIOTR BORA

Wojskowa Akademia Techniczna,
Wydział Cybernetyki, Instytut Matematyki i Kryptologii,
00-908 Warszawa, ul. S. Kaliskiego 2

Streszczenie. W artykule przedstawiono skrótowo ataki na implementacje algorytmów blokowych szyfrowania informacji przez analizę ulotu elektromagnetycznego ze szczególnym uwzględnieniem ulotu przewodzonego. Odniesiono się w opisie do ogólnego sformułowania modułu szyfratora z uwzględnieniem rozwiązań zarówno w oparciu o karty procesorowe, jak i specjalizowane szyfratory. Na podstawie przedstawionych ataków odniesiono się do bezpieczeństwa i metod zabezpieczeń dla rozwiązań bazujących na układach FPGA.
Słowa kluczowe: algorytmy blokowe, kryptoanaliza algorytmów blokowych, implementacja sprzętowa
Symbole UKD: : 003.26, 512-12

1. Wprowadzenie

Systemy kryptograficzne mogą być realizowane z wykorzystaniem aplikacji działających w większym systemie (np. aplikacja w komputerze) lub w postaci dedykowanych modułów sprzętowych, dołączanych na czas pracy lub organizacji systemu do istniejących rozwiązań telekomunikacyjnych. To drugie rozwiązanie umożliwia zabezpieczenie stosowanego rozwiązania i izolację newralgicznych danych (np. kluczy) od otoczenia. Takie rozwiązania jednak wymagają analizy zagrożeń oraz możliwości zabezpieczeń. Do rozwiązań dedykowanych możemy zaliczyć systemy kart procesorowych oraz specjalizowane moduły szyfratorów. W niniejszym artykule autorzy

starają się przedstawić wybrane tylko metody ataku, możliwe do zastosowania bez wyspecjalizowanego sprzętu (no może poza oscyloskopem z pamięcią i modułem komunikacyjnym z komputerem i samym komputerem) oraz perspektywy możliwości pojawienia się nowych ataków na rozwiązania sprzętowe. Dla dalszych analiz przyjęto brać pod uwagę rozwiązania bazujące na kartach procesorowych oraz ściśle rozwiązana sprzętowe (np. w oparciu o struktury FPGA) z tego względu, że niektóre typy ataków dla jednych i drugich są podobne.

2. Techniki ataków

Podstawowym pytaniem, jakie należy sobie zadać projektując jakiekolwiek urządzenie kryptograficzne, jest czy napastnik może uzyskać nieograniczony i niekontrolowany dostęp do atakowanego urządzenia. Jeśli nie, to sposób zabezpieczenia przed wszelkimi atakami jest w miarę prosty, jednak w przypadku kart inteligentnych napastnik zwykle posiada niczym nieograniczony dostęp do atakowanej karty. W przypadku izolowanych modułów szyfratorów teoretycznie dostępu takiego nie ma, co nie znaczy, że nie można go uzyskać przez np. frezowanie obudowy modułu. Wszystkie ataki, jakimi mogą się posłużyć napastnicy do skompromitowania modułu kryptograficznego, można podzielić na cztery kategorie:

— atak fizyczny: polega na dostaniu się bezpośrednio do niezabezpieczonej powierzchni układu wbudowanego w kartę lub moduł i odczytaniu danych, będących przedmiotem ataku za pomocą mikroskopijnych sond,
— atak logiczny: polega na wykorzystaniu, w czasie normalnej komunikacji z kartą lub modułem, słabości algorytmów kryptograficznych i protokołów komunikacyjnych do nieautoryzowanego uzyskania danych,
— generacja błędów: ta technika ataku wykorzystuje anormalne warunki pracy układu do wymuszenia pomyłek w działaniu, powodujących przesłanie na zewnątrz danych, będących przedmiotem ataku,
— monitorowanie informacji: polega na monitorowaniu z dużą rozdzielczością czasową i wykorzystaniu do ataku analogowych charakterystyk napięcia, promieniowania elektromagnetycznego oraz czasu wykonywania operacji.

Ataki fizyczne są atakami inwazyjnymi. Ich przeprowadzenie wymaga często wielu godzin pracy w odpowiednio przygotowanym laboratorium, a w ich wyniku atakowane urządzenie ulega najczęściej zniszczeniu, a co najmniej pozostają na nim wyraźne ślady ingerencji z zewnątrz. Pozostałe trzy grupy są atakami nieinwazyjnymi, a po przygotowaniu ataku na dany

typ układu, można przeprowadzić ponownie taki atak na innym układzie tego samego typu w bardzo krótkim czasie i przy bardzo niskich nakładach finansowych. Ponadto wyposażenia służącego do przeprowadzenia ataków nieinwazyjnych zwykle nie można odróżnić od zwykłego czytnika lub modułu komunikacyjnego. Ataki nieinwazyjne w niektórych zastosowaniach np. kart mogą być znacznie poważniejsze w skutkach. Z drugiej jednak strony ataki nieinwazyjne wymagają zwykle większych umiejętności posiadanych przez napastników. Ataki fizyczne, łatwiejsze do wykrycia, zwykle wymagają bardzo szczątkowej wiedzy o atakowanym systemie, a ponadto do ataku na wiele różnych typów urządzeń napastnicy posługują się tym samym zbiorem technik. Dlatego też, najczęściej atak na system rozpoczyna się od ataku fizycznego na jeden egzemplarz, który pozwala na zdobycie niezbędnej wiedzy do przeprowadzenia powtarzalnych i mniej kosztownych ataków nieinwazyjnych na innych kartach lub modułach z tym samym rozwiązaniem.

3. Monitorowanie informacji

Ponieważ układy modułów składają się z bardzo dużej liczby tranzystorów, które w czasie swego działania mają różne zapotrzebowanie na prąd, a okazuje się że tranzystor najwięcej prądu potrzebuje w czasie zmiany swojego stanu, więc w czasie pracy zapotrzebowanie układu na prąd się zmienia. Jako że wykonywanie różnych fragmentów kodu wymaga aktywności różnych części układu, a w szczególności powoduje zmianę stanu różnej liczby tranzystorów, to istnieje pewna korelacja pomiędzy wykonywanym kodem, a zapotrzebowaniem układu na prąd. Ponieważ ścieżka, którą przebiega wykonywanie programu, jest zwykle zależna od danych, które są w nim przetwarzane oraz fakt, że zapamiętanie w rejestrach układu, bądź pamięci, '1' wymaga więcej prądu niż zapamiętanie '0', to istnieje korelacja pomiędzy poborem prądu a danymi przetwarzanymi przez układ. W takiej sytuacji, jeśli jedną z danych, wykorzystywanych przez układ w czasie obliczeń, są dane kluczowe, to fakt istnienia korelacji pomiędzy nieznanym kluczem a poborem prądu można wykorzystać do ataku na moduł lub szczególnie kartę i uzyskania klucza. Ataki te noszą nazwę ataków poboru mocy, jako że zmienny prąd pobierany przez układ, przy stałym napięciu zasilania, powoduje zmianę faktycznej mocy pobieranej przez moduł. Oczywiście w praktycznej implementacji ataku łatwiej jest mierzyć wartość prądu niż mocy [6].

Ponieważ z przepływem prądu przez przewodnik nieodłącznie związane jest pole elektromagnetyczne indukowane wokół tego przewodnika, to istnieje podobna korelacja pomiędzy przetwarzanymi danymi, a w tym

przypadku wartością natężenia pola elektromagnetycznego, mierzonego w niewielkiej odległości nad układem np. karty. Do przeprowadzenia tego ataku używa się bardzo małych cewek, umieszczanych nad powierzchnią wybranego układu scalonego. Zmienne pole elektromagnetyczne powoduje indukowanie się w nich prądu elektrycznego. W efekcie dostajemy bardzo podobne przebiegi zmian prądu, jak przy przeprowadzaniu ataków poboru mocy. Atak ten ma jednak jedną zaletę, omawiane cewki można umieszczać w różnych miejscach nad układem, można używać ich także kilku na raz, co pozwala na uzyskanie większej korelacji z przetwarzanymi danymi, a więc i z danymi kluczowymi.

Jeszcze inną odmianą ataków, polegającą na monitorowaniu informacji, są ataki czasowe, okazuje się bowiem, że czas wykonywania operacji jest także skorelowany z przetwarzanymi danymi. Na przykład w czasie wykonywania potęgowania modularnego w algorytmie RSA w przypadku, gdy w nieznanym wykładniku występuje '1', są wykonywane inne operacje, niż gdy występuje '0'. Jeśli te operacje mają różny czas wykonywania, co jest bardo powszechne, to znajomość całkowitego czasu wykonania potęgowania oraz liczby bitów wykładnika daje natychmiast informację o jego wadze Hamminga, co redukuje nakład obliczeń przy ataku brutalnym [8].

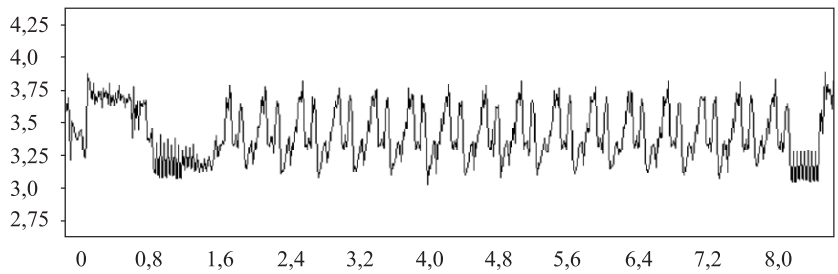
4. Ataki poboru mocy

Pierwsze doniesienia na temat ataków wykorzystujących monitorowanie poboru mocy oficjalnie pojawiły się w roku 1996. Wtedy to Paul Kocher w swojej pracy [8] przedstawił dokładny opis ataku na algorytm RSA pozwalający uzyskać nieznany wykładnik na podstawie czasu wykonywania potęgowania. W roku 1997 Ernst Bovenlander na konferencji Eurocrypt pokazał, że w sygnale poboru prądu, zebranym na przykład za pomocą oscyloskopu w czasie wykonywania szyfrowania algorytmem DES, widać wyraźną strukturę tego algorytmu, składającą się z 16 powtarzających się części. Jednak za ojca ataków poboru mocy na karty inteligentne uważa się Paula Kochera, który w roku 1998 razem z Jousha Jaffe i Benjaminem Yunem opublikował artykuł [6].

W artykule tym Kocher zaproponował trzy narzędzia do atakowania systemów na podstawie poboru prądu. Pierwsze z nich, SPA[1], polegało na prostej analizie przebiegu poboru prądu, na podstawie której można zidentyfikować poszczególne instrukcje, co może prowadzić do kompromitacji klucza, jeśli jest on używany jako warunek w instrukcjach skoków warunkowych. Przykładowy przebieg poboru prądu podczas operacji szyfrowania algorytmem DES na karcie procesorowej przedstawiono na rysunku 1.

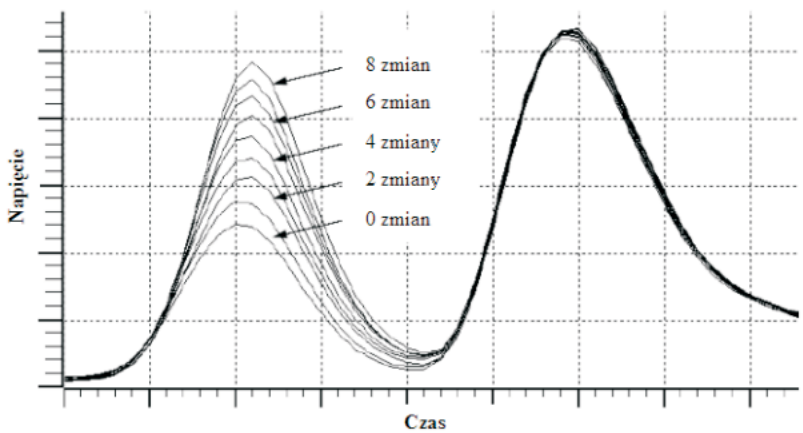
[1] Simple Power Analysis – prosta analiza mocy.

Podobne w charakterze obrazy przebiegów uzyskano dla wersji rundowej algorytmu implementowanej w układzie FPGA FLEX10K10. Widać na nim 16 powtarzających się wzorców realizacji rundy algorytmu. Różnice pomiędzy poszczególnymi fragmentami wynikają ze zmiennych elementów klucza, stosowanych w różnych rundach oraz danych.



Rys. 1. Pobór mocy w czasie szyfrowania algorytmem DES

Najczęściej w przypadku tego ataku wykorzystuje się analizę wag Hamminga obrazów przetwarzanych danych. Przykładem może być tu przebieg przedstawiony na rys. 2.



Rys. 2. Różnica w poborze prądu w czasie wykonywania tej samej instrukcji przy różnej liczbie zmieniających się bitów na szynie danych. Różnica pomiędzy zmianą i bitów a $i+1$ bitów jest na poziomie 6,5 mV

Analizując poszczególne fazy realizacji rundy, w przypadku kart procesorowych nie zabezpieczonych na ten typ ataku, oraz w pozostałych rundach otrzymujemy 128 (lub w gorszym przypadku 96) wag Hamminga dla 56 bitów klucza. W końcowym efekcie ilość równań w obu przypadkach

przewyższa ilość bitów klucza, dając jednoznaczny wynik (przy poprawnym zinterpretowaniu poziomów i zależności ich od ilości jedynek w czynniku). Można również ograniczyć zakres ataku do pierwszych rund lub niektórych S-boxów, natomiast resztę bitów klucza odzyskiwać z wykorzystaniem ataku brutalnego.

Jeśli napastnik z jakiegoś powodu nie może tego wykorzystać, można spróbować uzyskać klucz, atakując już sam algorytm szyfrowania. Dysponując wagą Hamminga bloku danych wychodzących po 15 rundzie algorytmu DES napastnik mógłby zgadnąć 48 bitów podklucza ostatniej rundy i sprawdzić poprawność przewidywań za pomocą kilku szyfrowań. Atak ten wymagałby około $48/\log_2 \sqrt{32} \simeq 15$ szyfrogramów i wykonanie już poza atakowaną kartą pracy równoważnej 2^{44} próbnych szyfrowań, ostatnią złożoność można zredukować do 2^{22} przez zastosowanie techniki „spotkanie w środku".

Jednak można zastosować inne podejście, wykorzystujące wagę Hamminga danych pośrednich z ostatniej rundy szyfrowania, bardziej skomplikowane, ale wymagające za to mniejszego nakładu obliczeń. Niech $F(R_{15})$ oznacza wyjście z funkcji F ostatniej, 16 rundy (R_{15} jest prawą połową wyjścia z 15. rundy), $W = H_w(F(R_{15}))$ oznacza wagę Hamminga wyjścia z funkcji F ostatniej rundy, które to może być zmierzone przez napastnika na podstawie poboru prądu. Ponadto, niech S będzie oznaczało wagę Hamminga czterobitowego wyjścia pierwszego S-boxa w czasie obliczania $F(R_{15})$, a N łączną wagę Hamminga wyjść pozostałych S-boxów w tej rundzie tak, że $W = S + N$. Zgadując wartość sześciu bitów wchodzących do pierwszego S-boxa oraz znając odpowiadający temu szyfrowaniu szyfrogram otrzymujemy S, jeśli nasze przewidywania co do bitów klucza były poprawne lub T w przeciwnym razie (zmienna T ma ten sam rozkład co S i jest od niej niezależna). Podstawową ideą tego ataku jest fakt, że S jest silnie skorelowane z $W = S + N$, a T nie, co pozwala na wybranie poprawnych wartości sześciu bitów spośród wszystkich 64 możliwości.

Ten sam scenariusz ataku można uogólnić do przypadku, gdy napastnik dysponuje jakąkolwiek daną skorelowaną z kluczem i o znanej wadze Hamminga. Dla przykładu niech napastnik dysponuje wagą Hamminga W, oznaczającą teraz całkowitą wagę Hamminga zsumowaną po wszystkich wyjściach S-boxów, pojawiających się we wszystkich rundach algorytmu. Okazuje się że w takiej sytuacji scenariusz ataku przedstawiony powyżej może być powtórzony w niezmienionej postaci. Jak poprzednio, niech S oznacza wagę Hamminga wyjścia pierwszego S-boxa, wtedy całkowitą wagę Hamminga W można wyrazić jako $W = S+N$, gdzie N oznacza tym razem sumę wag pozostałych S-boxów, biorących udział w obliczeniach w czasie wykonania całego algorytmu. W takim przypadku N jest zmienną losową

o rozkładzie 509 punktowym, o średniej 254 i jest niezależna od S. Atak należy przeprowadzić jak poprzednio, z tą jednak różnicą, że do uzyskania wysokiego prawdopodobieństwa poprawnego rozpoznania bitów klucza należy użyć $n = 2^{20}$ znanych tekstów jawnych i wykonać poza modułem obliczenia porównywalne z wykonaniem 2^{19} szyfrowań. Poprzez symultaniczne atakowanie po 24 bity klucza można zredukować wymaganą liczbę szyfrogramów do wartości 2^{19}, a nakład obliczeń do 2^{37} szyfrowań. W bardzo prosty sposób, atak ten może być zmieniony i zastosowany do uzyskania klucza w przypadku, gdy napastnik ma dostęp tylko do tekstów jawnych. Identyczne podejście można zastosować do ataku na implementację algorytmu MISTY1. Algorytm operuje na bloku danych o długości 64 bity oraz kluczu wejściowym 128 bitów. Klucz wejściowy co prawda rozszerzany jest do 256 bitów, ale dalej z tej bazy wybierane są bity do rund bez jakiegokolwiek przetwarzania.

Podobnie jest w przypadku algorytmu AES [9]. W algorytmie tym w każdej rundzie wykorzystywany jest inny klucz, ale tak naprawdę jest on zależny od klucza poprzedniej rundy i można w ten sposób uzyskać klucz źródłowy. W wielu przypadkach wśród funkcji, jakie zawierają np. karty oferujące szyfrowanie za pomocą algorytmu AES, jest także procedura generacji podkluczy, co z jednej strony pozwala przyśpieszyć proces szyfrowania i deszyfrowania informacji z użyciem tego samego klucza, ale z drugiej pozwala napastnikowi na łatwiejszą identyfikację odpowiednich części przebiegu. Jeśli napastnik ma zidentyfikowany fragment, w czasie którego realizowana jest generacja podkluczy, to przez odjęcie od siebie dwóch przebiegów uzyskanych w czasie generacji podkluczy dla różnych kluczy, uzyska on miejsca, w których układ wykonując te same instrukcje operował na innych danych, co w tym przypadku oznacza kolejne bity podkluczy. Sposoby lokalizacji w przebiegu jego części zależnych od klucza są opisane w pracach [2] i [5].

Kolejną metodą jest atak DPA[2], który pozwala na uzyskanie klucza np. w algorytmie DES bez jakiejkolwiek znajomości implementacji tego algorytmu w układzie karty procesorowej lub innego rozwiązania procesorowego. Atak wykorzystywał drobne różnice w poborze prądu w czasie wykonywania tych samych instrukcji lecz operujących na innych danych. Były one zbyt małe, aby można było je dostrzec analizując przebiegi za pomocą SPA, jednak obserwacja nie poziomów sygnałów ale ich różnicy pozwalała na poprawną identyfikację kluczy używanych w obliczeniach.

[2] Diferential Power Analysis – różnicowa analiza mocy.

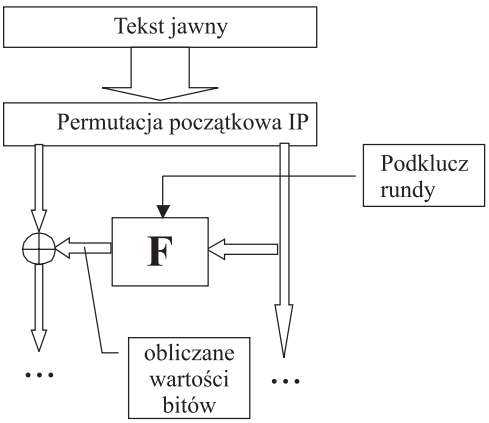
Ostatnim z pomysłów przedstawionych w artykule P. Kochera były ataki HO-DPA[3], będące rozwinięciem poprzedniego pomysłu.

Główna idea różnicowej analizy poboru mocy opiera się na grupowaniu przebiegów w dwa zbiory, wyliczeniu wartości średniej dla każdego zbioru i stworzeniu przebiegu, będącego różnicą tych dwóch. Oczywiście dzielenie na dwa zbiory nie odbywa się losowo, lecz na podstawie wagi Hamminga pośredniej danej, pojawiającej się w czasie wykonywania algorytmu. Jeden zbiór tworzą przebiegi, dla których jej wartość miała dużą wagę Hamminga, drugi tworzą te, dla których miała ona małą wagę. Ponieważ od poboru prądu zależy waga Hamminga przetwarzanych danych, to średni przebieg z jednego zbioru w miejscu przetwarzania omawianej danej będzie miał inną wartość niż drugi, utworzony na podstawie drugiego zbioru. Po odjęciu od siebie tych dwóch przebiegów powstanie ślad różnicowy w postaci prawie płaskiego przebiegu, z pikami w miejscu przetwarzania wspomnianej danej pośredniej. Głównym zadaniem dla napastnika jest dzielenie przebiegów na dwa zbiory na podstawie danych wejściowych do algorytmu ale także na podstawie atakowanego klucza, stąd pojawia się warunek, jaki musi spełniać algorytm, aby być podatny na ataki DPA: w czasie wykonywania algorytmu istnieje pośrednia zmienna taka, że znajomość kilku bitów klucza (w praktyce mniej niż 32) pozwala nam stwierdzić czy dwie wartości danych wejściowych (lub wyjściowych) do algorytmu dają lub nie tę samą wartość tej zmiennej. Na podstawie wartości tej zmiennej napastnik będzie dokonywał podziału przebiegów na rozłączne zbiory.

W przypadku praktycznego ataku na algorytm DES do określania na podstawie danych wejściowych do algorytmu wartości zmiennej wspomnianej w powyższym warunku służy funkcja wyboru $D(P, b, K_S)$. Jest ona zdefiniowana jako obliczenie wartości jednego z bitów $-\leqslant b < 32$ danych wychodzących ze skrzynek podstawieniowych w pierwszej rundzie algorytmu, gdzie P jest tekstem jawnym, a $0 \leqslant K_S < 2^6$ to wartość sześciu bitów klucza wchodzących do odpowiedniego S-boxu. Miejsce, w którym funkcja D określa wartość bitów, jest pokazane na rys. 3. Jeśli wartość podklucza rundy, wchodząca do funkcji na dany S-box, nie jest poprawna, to wartość funkcji D pokrywa się z faktyczną wartością wyliczanego b-tego bitu z prawdopodobieństwem $\frac{1}{2}$. Jeśli natomiast sześć bitów klucza jest poprawne, to wartość funkcji D pokrywa się z wartością b-tego bitu z prawdopodobieństwem 1.

W celu przeprowadzenia ataku, napastnik wymusza na karcie wykonanie m szyfrowań, zapisując przy tym przebiegi poboru mocy ${}_{1,\ldots,N}[1,\ldots,p]$ próbkowane w p punktach każdy oraz odpowiadające im teksty jawne

[3] Higer Order DPA – różnicowa analiza mocy wyższych rzędów.



Rys. 3. Początek pierwszej rundy algorytmu DES i miejsce określenia wartości bitów przez funkcję D

(których rozkład powinien być jednostajny). Następnie rozpoczynając od pierwszego S-boxa, zgadując wartość sześciu bitów klucza K_S wchodzących na tę skrzynkę, oblicza wynik funkcji D, a na koniec wylicza ślad różnicowy wyrażony formułą.

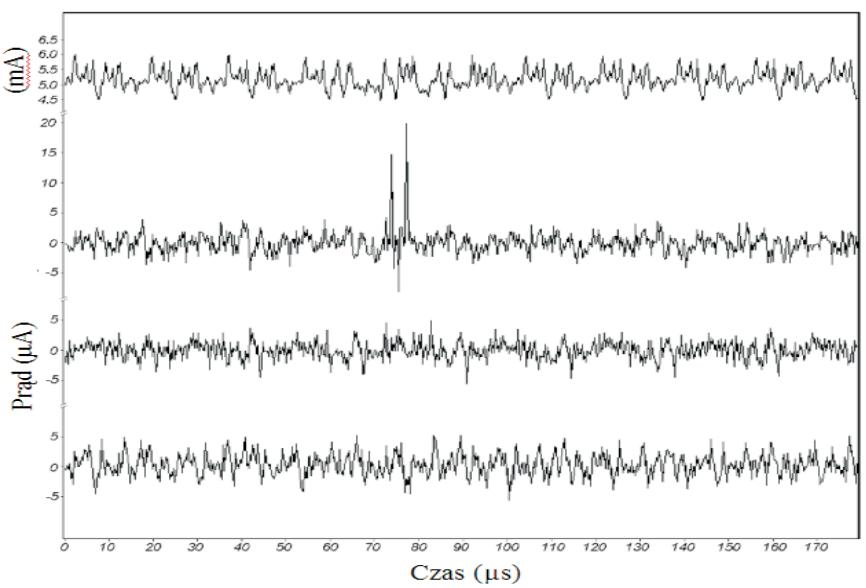
$$\Delta_{K_S}[j] = \frac{\sum_{i=1}^{N} D(P_i, b, K_S) T_i[j]}{\sum_{i=1}^{N} D(P_i, b, K_S)} - \frac{\sum_{i=1}^{N} (1-D)(P_i, b, K_S) T_i[j]}{\sum_{i=1}^{N} (1-D)(P_i, b, K_S)}$$

$$\approx \left(\frac{\sum_{i=1}^{N} D(P_i, b, K_S) T_i[j]}{\sum_{i=1}^{N} D(P_i, b, K_S)} - \frac{\sum_{i=1}^{N} T_i[j]}{N} \right)$$

Jeśli K_S jest zgadnięte prawidłowo, to wartość funkcji D jest zgodna z wartością b-tego bitu dla każdego tekstu jawnego. Wtedy dla przebiegów, dla których wartość funkcji D wynosiła 1, wartość oczekiwana wagi Hamminga czterobitowych danych wychodzących z pierwszego S-boxa wynosi $1 + \frac{3}{2} = \frac{5}{2}$, a dla przebiegów znajdujących się w drugim zbiorze wynosi odpowiednio $0 + \frac{3}{2} = \frac{3}{2}$. Pozostałe dane, pojawiające się w czasie wykonywania algorytmu i nie skorelowane z D można traktować jako dyskretne zmienne losowe o rozkładzie jednostajnym i o wartości oczekiwanej wagi Hamminga równej połowie szerokości tych danych w bitach. Wartość ta jest jednakowa dla przebiegów w obu zbiorach ($D = 0$ i $D = 1$). Obliczając ślad różnicowy $\Delta_{K_S}[j]$, uzyskujemy następujący efekt:

— Szum występujący w każdym przebiegu można traktować jako zmienną losową o rozkładzie normalnym i wartości średniej 0. Po wykonaniu uśredniania w każdym ze zbiorów nie pojawia się w śladzie różnicowym.

— Pobór prądu zależny od wykonywanych instrukcji jest taki sam w przypadku przebiegów z obu zbiorów, gdyż za każdym razem wykonywany był ten sam algorytm, więc po uśrednieniu i odjęciu od siebie, ten składnik poboru prądu nie pojawia się w śladzie różnicowym.
— Pobór prądu spowodowany przetwarzaniem danych nie skorelowanych z D jest dla obu zbiorów taki sam, jako że dane te mają w obu przypadkach te same wartości oczekiwane wag Hamminga. Po odjęciu ich od siebie, składnik ten nie występuje w ostatecznym przebiegu.
— Pobór prądu związany z przetwarzaniem danych skorelowanych z D jest różny dla obu zbiorów, jako że zależy od wagi Hamminga danych, również różnej dla obu zbiorów. Po odjęciu od siebie pozostają w śladzie różnicowym wartości różne od 0.



Rys. 4. Wygląd przebiegów uzyskanych podczas ataku DPA. Górny przedstawia średni przebieg poboru mocy, poniższy zaś ślad różnicowy dla poprawnie odgadniętych bitów klucza. Dwa dolne to ślady różnicowe dla błędnych kluczy

W przypadku gdy nieprawidłowo została przewidziana wartość bitów klucza wchodzących na pierwszy S-box, wartość funkcji D zgadza się z wartością b-tego bitu z prawdopodobieństwem $\frac{1}{2}$. Czyli rozdzielenie przebiegów na dwa zbiory następuje zupełnie losowo, stąd wartość oczekiwana wagi Hamminga czterobitowych danych wychodzących z pierwszego S-boxa wynosi 2 dla obu zbiorów i pobór prądu związany z przetwarzaniem tych danych jest średnio taki sam. Powoduje to, że żadna z wartości mających

wpływ na pobór prądu nie jest skorelowana ze sposobem, w jaki następuje podział przebiegów na dwa zbiory w efekcie czego ślad różnicowy jest bliski 0.

$$\lim_{N\to\infty} \Delta_{K_S}[j] \approx 0$$

W związku z tym napastnik, sprawdzając wszystkie 64 możliwe wartości klucza wchodzące na pierwszy S-box, otrzymuje 64 ślady różnicowe, z których większość jest płaska, a tylko jeden, dla którego wartość klucza jest prawidłowa, zawiera piki wszędzie tam, gdzie były przetwarzane dane wychodzące z pierwszego S-boxa lub dane z nimi skorelowane. Sytuacja ta jest przedstawiona na rys. 4, gdzie górny ślad przedstawia uśredniony przebieg dla jednego ze zbiorów, drugi od góry jest śladem różnicowym dla poprawnie przewidzianych wartości bitów klucza, a dwa dolne odpowiadają sytuacji, gdy klucz był zgadnięty nieprawidłowo.

W celu uzyskania całego podklucza rundy napastnik wykonuje atak dla kolejnych części wchodzących na pozostałe S-boxy, uzyskując w ten sposób 48 bitów klucza. Pozostałe bity uzyskuje się przez przegląd 2^8 możliwych wartości lub aplikując powyższy atak do kolejnej rundy, jako że mając pierwszy podklucz można obliczyć wartości wchodzące do funkcji F w drugiej rundzie algorytmu.

T. S. Messerges, E. A. Dabbish, R. H. Sloan w pracy [11] przedstawili dokładną analizę ataku zaproponowanego przez Kochera rok wcześniej oraz podali sposób zwiększenia jego efektywności, pozwalający w efekcie skrócić czas ataku. W tym samym roku pojawił się także nowy rodzaj ataku na algorytm DES zaproponowany przez Paula N. Fahna i Petera K. Pearsona w pracy [5]. Polegał on na zlokalizowaniu, w czasie wstępnego etapu ataku, w przebiegu poboru prądu miejsc, w których jest on zależny w głównej mierze od poszczególnych bitów klucza, i na podstawie wartości poboru prądu w tych miejscach w czasie szyfrowania danych przez atakowane urządzenie wnioskowaniu o poszczególnych bitach nieznanego klucza. Thomas S. Messerges przedstawił na konferencji CHES2000 swoją pracę [12], w której pokazał, jak stosować ataki HO-DPA do kompromitacji algorytmów zabezpieczonych przed atakami DPA. Na tej samej konferencji pojawiły się jeszcze dwie inne prace o atakach mocy: Rity Mayer-Sommer [10], w której przedstawia ona wyniki świadczące o możliwościach ataku SPA przedstawionego dość ogólnie przez Kochera w roku 1998 oraz praca Christophea Claviera, Jean-Sebastiana Corona i Nory Dabbous [3]. Zespół ten zaproponował zastosowanie innych funkcji statystycznych, używanych w czasie przeprowadzania ataku i sposobu przeprowadzania samego ataku DPA na urządzenia, które zostały „zabezpieczone" przeciwko tym atakom.

Praca Jean-Jacquesa Quisquatera i Davida Samydea [15], zawierająca ogólne wskazówki dotyczące przeprowadzenia takiego ataku i zabezpieczenia się przed nim oraz praca [13] Karine Gandolf, Christophea Mourtela i Francisa Olivera, w której przedstawiono wyniki z prób przeprowadzenia tego ataku na różne algorytmy oraz porównanie z atakiem DPA, stanowią rozszerzenie prac wcześniejszych ze wskazaniem metod zabezpieczeń w przypadku rozwiązań dla kart procesorowych. Rozwinięcie ataku DPA na DES przez zmianę funkcji statystycznych, będących sercem tego ataku, prowadzące do jeszcze większego zmniejszenia nakładów czasowych niż poprzednie modyfikacje, przedstawiają Regis Bevan i Erik Knudsen w pracy [1].

Propozycja zupełnie nowego ataku (ataku szablonowego – TA), autorstwa Suresh Chari, Josyula R. Rao i Pankaj Rohatki, zawarta jest w pracy [4]. Zaproponowali stworzenie „szablonów" przez szyfrowanie danych na innym egzemplarzu urządzenia tego samego typu co atakowane, odpowiadających różnym wartościom klucza, a następnie sprawdzenie, do którego z nich pasuje przebieg uzyskany z atakowanego urządzenia. Jak podają autorzy, jest to najmocniejsza forma ataku, wykorzystująca do swego działania wszystkie informacje, jakie można uzyskać z przebiegu poboru prądu.

Atak ten wymaga od napastnika dostępu do urządzenia identycznego z atakowanym, na podstawie którego budowana jest dokładna charakterystyka poboru prądu, włącznie z zamodelowaniem szumu za pomocą wielowymiarowej zmiennej losowej o rozkładzie Gaussa. Cała idea tego ataku opiera się na budowaniu szablonów poboru prądu, odpowiadających poszczególnym wartościom klucza i dopasowywaniu do nich pojedynczego przebiegu uzyskanego z atakowanego urządzenia. Ponieważ wartości klucza jest zbyt wiele, by mogły być analizowane jednocześnie, szablony buduje się na podstawie jego fragmentów. Następnie po wybraniu najbardziej prawdopodobnych, buduje się kolejne, uwzględniające większy fragment klucza i tak aż do uzyskania jego pełnej wartości. Ze względu na sposób przeprowadzania, atak ten wymaga, aby napastnik miał możliwość wykonywania atakowanego algorytmu z dowolnie wybranym przez siebie kluczem.

Dedukcyjna analiza poboru mocy (IPA) jest atakiem składającym się z dwóch etapów, profilowania i ekstrahowania klucza. Pierwszy etap, zajmujący większość czasu, polega na porównaniu ze sobą powtarzających się fragmentów algorytmu, jak na przykład rundy w algorytmie DES. Etap ten należy przeprowadzić na układzie identycznym z atakowanym i wymaga on zwykle dużej ilości powtórzeń algorytmu oraz zgromadzenia wielu przebiegów poboru mocy. Jego celem jest uzyskanie profilu atakowanego urządzenia, który może posłużyć do uzyskania klucza z innego układu na podstawie bardzo małej ilości przebiegów poboru mocy. Atak ten, na podstawie

przeprowadzenia przez napastnika złożonych czasowo operacji związanych z profilowaniem układu pozwala mu później na atakowanie układów tego samego typu przy bardzo małych nakładach czasowych, więc koszt przeprowadzenia takiego ataku może się bardzo szybko zwrócić. W przeciwieństwie do ataków DPA napastnik nie musi znać żadnego z tekstów jawnych ani szyfrogramów przetwarzanych przez atakowaną kartę, ani kartę, na której dokonuje profilowania.

W celu przeprowadzenia ataku, napastnik wymusza na karcie wielokrotne wykonanie atakowanego algorytmu, w celu otrzymania wielu przebiegów poboru mocy. Zwykle do przeprowadzenia tego etapu wystarczy od 100 do 1000 przebiegów. Wykonania algorytmu odbywają się z tym samym kluczem, nie musi on być znany napastnikowi, ale przy różnych danych wejściowych, których rozkład powinien być jednostajny.

Tak uzyskane przebiegi uśrednia się w celu wyeliminowania zmian poboru prądu spowodowanych szumem i zmieniającymi się danymi, a pozostawiając efekty wynikające z wykonywania instrukcji i przetwarzania tego samego klucza. Aby poprawnie przeprowadzić uśrednianie, napastnik musi być pewny, że czas wykonywania tych samych instrukcji odpowiada dokładnie tym samym punktom na przebiegu poboru mocy. Po wykonaniu tej operacji napastnik dysponuje uśrednionym przebiegiem, odpowiadającym średniej po wszystkich tekstach jawnych przebiegów zebranych w czasie szyfrowania z użyciem tego samego klucza.

W ramach prac realizowanych w Laboratorium Badawczym Kryptologii IMiK WAT zaproponowano własną wersję ataku poboru mocy [14], będącego modyfikacją zaprezentowanego ataku IPA i wykorzystującego lepiej własności algorytmu DES. Atak ten składa się także z dwóch etapów, profilowania i ekstrahowania klucza, jednak obie części różnią się od proponowanych wcześniej. W pracy [5] autorzy kładli duży nacisk na poprawne dopasowanie do siebie fragmentów przebiegu reprezentującego poszczególne rundy. W przypadku gdy to zadanie było wykonane błędnie, napastnik w wyniku etapu profilowania mógł otrzymać inną liczbę pików niż jest bitów podklucza w rundzie, co mogło uniemożliwić dalszą analizę. Niestety autorzy nie podali żadnego sposobu na wykonanie takiego dopasowania twierdząc tylko, że w przypadku niektórych algorytmów zadanie to może być dość trudne. Opracowana wersja ataku pozwala, przez zwiększenie czasu profilowania, pozbycie się konieczności dokładnego dopasowania każdej rundy. Ponadto przez zmianę sposobu przeprowadzania drugiego etapu ataku, polegającego na ekstrahowaniu klucza, będzie mógł być przeprowadzony już na podstawie jednego przebiegu uzyskanego z atakowanego urządzenia. Etap ten będzie podobny do ataku szablonowego, jednak w przypadku pracy [4] budowanie szablonów odbywało się po zebraniu przebiegu z atakowanego urządzenia, gdyż to on decydował, jakie

mają być kolejne szablony. Wersja opracowana w IMiK pozwala wyliczyć wszystkie szablony jeszcze przed atakiem, co pozwoli napastnikowi na szybsze przeprowadzenie samego etapu uzyskiwania klucza.

Założenia co do ataku są połączeniem wymagań stawianych przez atak szablonowy i atak dedukcyjny. Napastnik dysponuje testowym egzemplarzem urządzenia, identycznym z egzemplarzem atakowanym. Posiada nad nim nieograniczoną kontrolę, może wykonywać wielokrotnie szyfrowanie algorytmem DES, z własnoręcznie wybranymi kluczami i tekstami jawnymi. Ostatnim założeniem jest, że atakowany algorytm przetwarza wszystkie bity klucza pojedynczo, jest więc to taka sama sytuacja co zaprezentowana w pracy [5], jednak w przypadku tego ataku warunek ten jest konieczny, by był on realizowalny.

5. Wpływ ataków na implementacje algorytmów blokowych w strukturach FPGA

Po zapoznaniu się z przedstawionymi skrótowo w tym artykule atakami na moduły kryptograficzne widać, że omawiane urządzenia, w przypadku dostępu do elementów modułu, zdecydowanie nie są odporne na rozpoznanie. Co więcej w przypadku niektórych ataków nakłady finansowe i czasowe ich przeprowadzenia są tak małe, a poziom ich skomplikowania tak niski, że praktycznie każdy może je przeprowadzić. Co prawda w przypadku rozwiązań realizowanych w oparciu o struktury FPGA i w dodatku według zasady „wszystko w jednym układzie", można opierać się co najwyżej na wagach Hamminga całych słów (np. 64 lub 128 bitów danych z rundy), jednak istnieją ataki wykorzystujące ten fakt. Jedną z metod zabezpieczenia się przed tymi atakami jest tu zaszumienie przez wykonywanie w ramach danej rundy obliczeń podkluczy do rundy następnej (np. możliwe jest to dla algorytmu AES) i zapisywanie obu wyników jednocześnie. Inną metodą jest realizowanie całości algorytmu jako funkcji kombinacyjnej danych i klucza, jednak wymaga to w przypadku implementacji algorytmów często dużych zasobów struktury FPGA.

Najnowszymi metodami analizy implementacji algorytmów w modułach jest nasłuch elektromagnetyczny. Wykonuje się go wykorzystując układy czujników wraz z precyzyjnym pozycjonowaniem ich nad układem. Wyróżnić należy tu dwa podejścia: ze znajomością rozkładu funkcjonalnego elementów w układzie (np. bitstreamu w strukturze FPGA) oraz bez tej znajomości. W skrócie opisując ten atak podczas jego realizacji odwzorowywana jest struktura pola elektromagnetycznego zarówno w przestrzeni XY jak i w czasie. Na podstawie zmian i analiz statystycznych wyników nasłuchu wielokrotnych szyfrowań można próbować odtworzyć wartości tajne.

Warunkiem powodzenia jest tu jednak w obu przypadkach znajomość zaimplementowanego algorytmu. Pewnym zabezpieczeniem przed tym atakiem w przypadku struktur FPGA jest implementacja różnicowa z uwzględnieniem rozkładu elementów wewnątrz struktury. W skrócie, jeśli dana komórka logiczna realizuje operację w zależności od bitów klucza, to sąsiednia musi wykonywać postać komplementarną tej operacji.

Ostatecznym wnioskiem, jaki się nasuwa jest taki, że moduły z danymi kluczowymi zarówno w postaci kart procesorowych, jak i układów FPGA muszą być projektowane z uwzględnieniem ataków poboru mocy i nasłuchu elektromagnetycznego. Nie powinno się ich zostawiać byle gdzie, ani oddawać „na chwilę" obcym [6].

Praca naukowa finansowana ze środków na naukę w latach 2008–2010 jako projekt rozwojowy Nr O R00 0031 06.

Artykuł wpłynął do redakcji w dniu 27.06.2008 r. Zweryfikowaną wersję po recenzji otrzymano w grudniu 2008 r.

LITERATURA

[1] R. Bevan, E. Knudsen, *Ways to Enhance Differential Power Analysis*, 2002.

[2] sc E. Biham, A. Shamir, *Power analysis of the Key Scheduling of the AES Candidates*, Second Advanced Encryption Standard Candidate Conference, 1999.

[3] Ch. Clavier, J. S. Coron, N. Dabbous, *Differential Power Analysis in the Presence of Hardware Countermeasures*, 2000.

[4] S. Chari, J. R. Rao, P. Rohatgi, *Template Attacks*, 2002.

[5] P. N. Fahn, P. K. Pearson, *IPA: A New Class of Power Attacks*, 1999.

[6] P. Kocher, J. Jaffe, B. Yun, *Introduction to Differential Power Analysis and related attacks*, 1998.

[7] P. Kocher, J. Jaffe, B. Yun, *Differential Power Analysis*, 1999.

[8] P. Kocher, *Timing attacks on implementation of Diffie-Hellman, RSA, DSS and other system*, 1996.

[9] S. Mangard, *A Simple Power Analysis (SPA) Attacks on Implementation of the AES Key Expansion*, 2002.

[10] R. Mayer-Sommer, *Smartly Analyzing the Simplicity and the Power of Simple Power Analysis on Smartcards*, 2000.

[11] T. S. Messerges, E. A. Dabbish, R. H. Sloan, *Investigation of Power Analysis Attacks on Smartcards*, 1999.

[12] T. S. Messerges, *Using Second-Order Power Analysis to Attacks DPA Resistant Software*, 2000.

[13] K. Gandolf, Ch. Mourtel, F. Oliver, *Electromagnetic Analysis: Concrete Results*, 2001.

[14] R. Parzych, *Analiza ataków mocy na wybrane algorytmy kryptograficzne*, 2004.

[15] J. J. Quisquater, D. Samyde, *Electromagnetic Analysis (EMA):Measures and Countermeasures for Smart Cards*, 2001.

J. GAWINECKI, P. BORA

Security analyssis of the hardware implementation of the block algorithms for information encryption

Abstract. In the paper attacks on implementation of block algorithms for information encryption were briefly described. These attacks are based on analysis of electromagnetic emanation especially of conducted emanation. In the description we address to general construction of encryption model and take into account solutions based on smart cards and specialized encryptors as well. On the base of presented attacks we addressed to security and protection methods for solutions based on FPGA circuits.

Keywords: block algorithms, cryptoanalysis of block algorithms, hardware implementation

2000 Mathematics Subject Classification: (primary) 94A60 (secondary) 94C99