Biuletyn WAT
Vol. LXXII, Nr 1, 2023

Systemy wbudowane oraz ich podatności na ataki sprzętowe

KONRAD SZCZEPANKIEWICZ[1], MARIAN WNUK[2]

[1]Wojskowa Akademia Techniczna, Szkoła Doktorska WAT, ul. gen. S. Kaliskiego 2B, konrad.szczepankiewicz@wat.edu.pl

[2]Wojskowa Akademia Techniczna, Wydział Elektroniki, Instytut Systemów Łączności, ul. gen. S. Kaliskiego 2, 00-908 Warszawa, marian.wnuk@wat.edu.pl

Streszczenie. Artykuł opisuje rodzaje ataków sprzętowych nakierunkowanych na systemy wbudowane, a także środki zapobiegawcze oraz metody modelowania zagrożenia bezpieczeństwa. Obecnie w urządzeniach elektronicznych poziom zabezpieczeń od strony oprogramowania jest zazwyczaj wysoki. Z kolei sprzętowe implementacje mogą pozostawiać luki, które atakujący mogą wykorzystać do ekstrahowania informacji lub zaburzania działania urządzenia w niezamierzony przez twórców sposób. Będąc użytkownikiem systemów wbudowanych, krytycznych dla bezpieczeństwa, należy być świadomym niebezpieczeństw spowodowanych lukami w oprogramowaniu, ale również znać zagadnienie analizy kanału pobocznego oraz iniekcji błędów.
Słowa kluczowe: ataki sprzętowe, analiza kanału pobocznego, podatności systemów wbudowanych, ochrona urządzeń przed atakami sprzętowymi
DOI: 10.5604/01.3001.0054.2897

1. Wprowadzenie

Systemy wbudowane to urządzenia bazujące na mikroprocesorze, przeznaczone do wykonywania wyspecjalizowanych funkcji, mogące stanowić część większego systemu. Należą do nich na przykład jednostki zarządzające pracą silnika w pojazdach, zegary cyfrowe, sterowniki ogrzewania centralnego, aparatura medyczna itp. Urządzenia elektroniczne zazwyczaj składają się z dwóch części: oprogramowania i sprzętu. Mówiąc o atakach sprzętowych, łatwo powiązać atak wykorzystujący sprzęt z atakiem ukierunkowanym na sprzęt, np. zakłócenie napięcia zasilania

mikrokontrolera (atak sprzętowy), aby wpłynąć na efekt wykonywanego przez niego programu (cel programowy). Dla odróżnienia atakiem sprzętowym nie będzie: spowodowanie przepełnienia bufora (atak programowy) w celu nieoczekiwanego zachowania się programu (cel programowy).

Ataki sprzętowe są trudniejsze do zrealizowania niż ataki programowe, ponieważ wymagają zazwyczaj bezpośredniego (fizycznego) dostępu do urządzenia oraz wiedzy w zakresie elektroniki. Niezbędna do przeprowadzenia ataku sprzętowego jest również aparatura. Atakujący nie zawsze musi dysponować laboratorium badawczym wyposażonym w drogi sprzęt, często może wystarczyć tylko komputer, oscyloskop i zestaw odpowiednich sond. Zatem z punktu widzenia producenta sztuką jest projektowanie urządzeń, które są tanie, a zarazem wystarczająco bezpieczne. W celu znalezienia kompromisu pomiędzy ceną a bezpieczeństwem podczas procesu opracowywania urządzenia należy wziąć pod uwagę między innymi następujące zagadnienia:

— Jaki cel może mieć potencjalny atakujący?
— Kim może być potencjalny atakujący i jakie ma umiejętności?
— Jakim sprzętem może dysponować?
— Ile czasu może zająć „złamanie zabezpieczeń"?

2. Analiza kanału pobocznego

Analiza kanału pobocznego (ang. *side channel analysis*) jest używana do ataków wykorzystujących podatności sprzętowe. To metoda korzystająca z monitorowania lub obsługi interfejsów niezamierzonych przez producenta, a wynikających z konstrukcji sprzętu. Prostym przykładem może być atak wykorzystujący kamerę termowizyjną przedstawiony na rysunku 1.

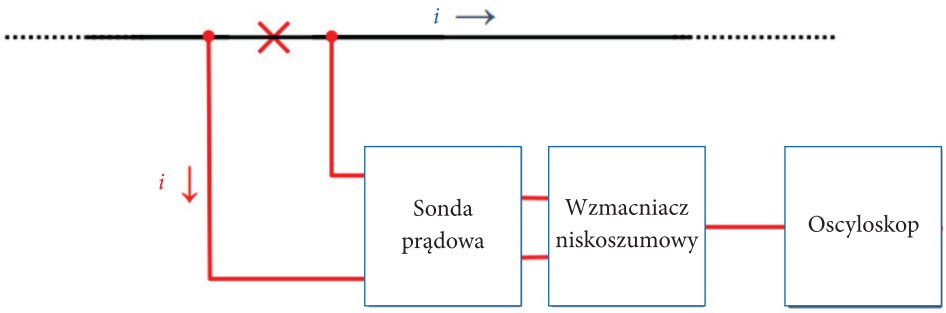
W wyjaśnieniu definicji interfejsów niezamierzonych może pomóc przedstawienie ich w opozycji do interfejsów standardowych na podstawie przykładów:

Przykłady standardowych interfejsów do komunikacji z urządzeniem:	Przykłady interfejsów niezamierzonych przez projektantów urządzeń:
— klawiatury, — monitory, — głośniki, — czytniki kart, — interfejsy USB, RS232 itp.	— pobór mocy, — promieniowanie EM, — temperatura, — czas, — dźwięk itp.

Ponadto w analizie kanału pobocznego można wyróżnić metody inwazyjne oraz bezinwazyjne. Aby przeprowadzić analizę poboru mocy przez urządzenie, niezbędne będzie włączenie sondy prądowej w obwód zasilania konkretnego układu poddanego badaniu. Ściślej mówiąc, jeżeli celem badania jest urządzenie szyfrujące,

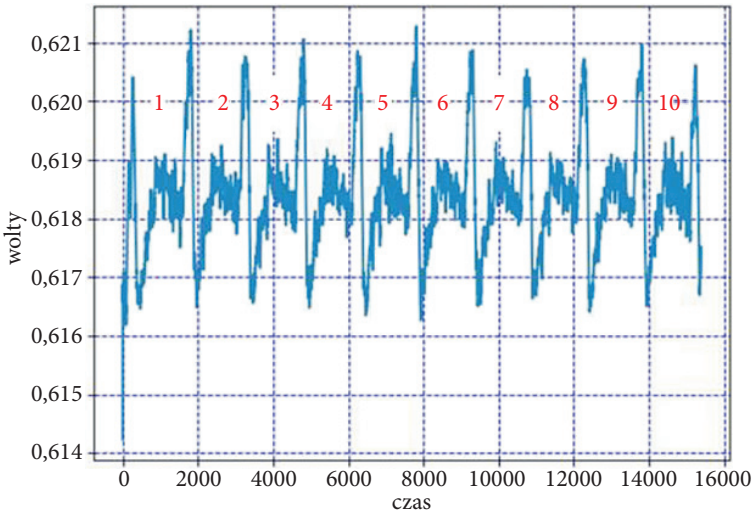
które dodatkowo wyposażone jest w peryferia (klawiatura, diody LED itd.) również pobierające prąd, to akwizycja poboru energii powinna dotyczyć wyłącznie układu scalonego odpowiedzialnego za szyfrowanie, co przeważnie wymaga przerwania linii zasilania układu scalonego na płytce PCB i wlutowania końcówek sondy prądowej.

Rys. 1. Analiza kanału pobocznego wykorzystująca temperaturę oraz podatność konstrukcji (przyciski z metalu, które dobrze przewodzą ciepło) do pozyskania wpisywanego przez użytkownika kodu PIN [3]



Rys. 2. Uproszczony schemat przedstawiający ideę podłączenia sondy do linii zasilającej

Kolejną ingerencją w układ, ułatwiającą analizę poboru mocy, może być pozbycie się kondensatorów odsprzęgających zasilanie. Są one umieszczane jak najbliżej układu scalonego, a ich zadaniem jest ustabilizowanie układu w przypadku wahań napięcia zasilania. Podczas analizowania poboru mocy to właśnie jak największe odchylenia od średniej są najbardziej wartościowe pod kątem analizy kanału pobocznego. To dzięki nim możliwe jest określenie, co układ robi w danej chwili. Dlatego też to właśnie sondy różnicowe są najczęściej stosowane.



Rys. 3. Obraz z oscyloskopu ukazujący pomiar poboru mocy przez układ [4]

Na rysunku 3 pokazano przebieg poboru mocy przez układ szyfrujący algorytmem AES-128, na podstawie którego (bez dodatkowego przetwarzania sygnału) można wskazać momenty, w którym układ scalony wykonuje poszczególne rundy. Symetryczny szyfr blokowy AES (ang. *Advanced Encryption Standard*) w wersji ze 128-bitową długością klucza charakteryzuje się liczbą rund wynoszącą 10, co pokrywa się z pomiarem przedstawionym na rysunku 3. Na podstawie tak dokonanych pomiarów można prowadzić dalszą analizę, mającą na celu określenie klucza algorytmu szyfrującego.

Atakami wykorzystującymi kanały poboczne szczególnie zagrożone są urządzenia korzystające z kryptografii, ponieważ ich bezpieczeństwo zazwyczaj opiera się na tajemnicy klucza, podczas gdy implementacja samego algorytmu jest powszechnie znana. Dobrym przykładem takiego algorytmu jest przytoczony wcześniej AES, szeroko stosowany w różnych protokołach komunikacji i bezpieczeństwa takich jak: WPA2 (ang. *Wi-Fi Protected Access*), SSH (ang. *Secure Shell*), VoIP (ang. *Voice over Internet Protocol*) itp. W zależności od długości klucza (128, 192, 256 bitów) liczba kombinacji może wynosić nawet do 2^{256}, co sprawia, że algorytm jest odporny na próby wytypowania wszystkich możliwości w klasyczny sposób. Jednak analiza kanału pobocznego pozwala znacząco zawęzić obszar poszukiwań właściwego klucza.

W analizowaniu przebiegów poboru mocy przez urządzenia szyfrujące można wyróżnić co najmniej kilka metod. Najpowszechniejsze są:

— prosta analiza poboru mocy SPA (ang. *Simple Power Analysis*),
— różnicowa analiza poboru mocy DPA (ang. *Differential Power Analysis*).

Obie te metody zostały opisane przez Paula Kochera, Joshuę Jaffe'a i Benjamina Juna w publikacji *Differential Power Analysis* [6].

2.1. Prosta analiza poboru mocy (SPA)

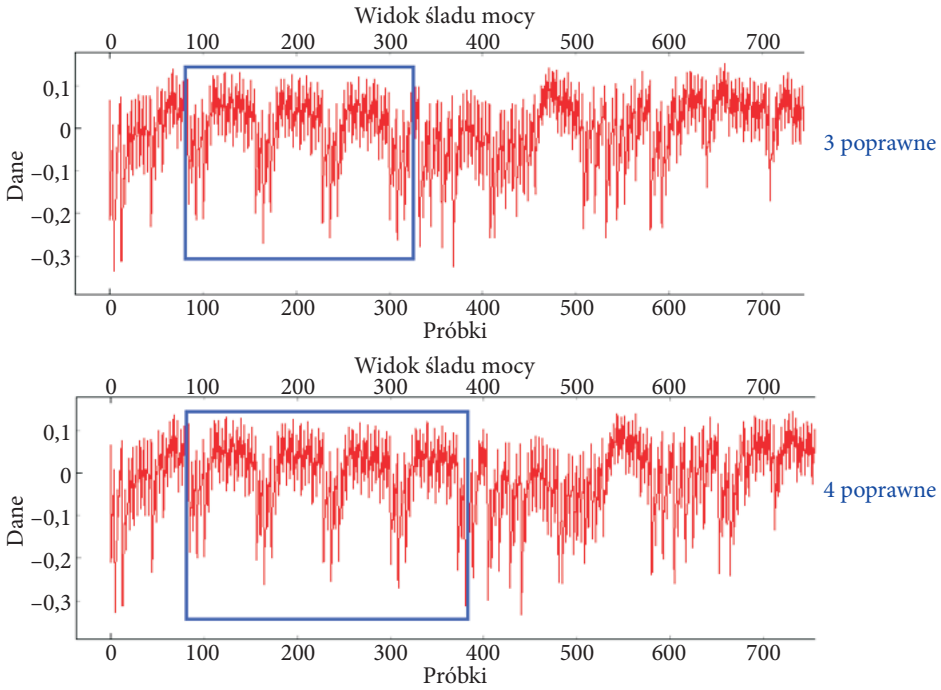
Prosta analiza poboru mocy może być przeprowadzona nawet na pojedynczym przebiegu lub niewielkiej ich liczbie. Przebiegi reprezentuje się w dziedzinie czasu. Klasycznym przykładem SPA jest wizualna inspekcja przebiegów (bez dodatkowych przekształceń czy obliczeń). Obecnie SPA służy w większości do rozpoznawania, z jakimi algorytmami mamy styczność, a nie do ich łamania (na rysunku 3 znajduje się przykład SPA, gdzie po liczbie rund można podejrzewać, z jakim algorytmem mamy do czynienia). Prosta analiza poboru mocy również może posłużyć do rozpoznania, jakie operacje urządzenie wykonuje w danej chwili, co może pomóc w precyzyjniejszym ustaleniu punktu wyzwalania. Sygnał wyzwalania to informacja dla oscyloskopu, kiedy ma rozpocząć rejestrowanie przebiegu. Wobec tego współczesne implementacje kryptograficzne są zbyt skomplikowane, aby były podatne na ataki SPA. Wyjątkiem mogą być słabo zrealizowane implementacje, takie jak sprawdzanie hasła czy PIN-u jak na poniższym przykładzie:

```
1   InputPassword=[a,l,a,b,d]
2   CorrectPassword =[a,l,a,m,a]
3
4   for i=1:5
5       if (InputPassword(i)~=CorrectPassword(i))
6           fprintf('Hasło niepoprawne!')
7           return
8       end
9   end
10
11  fprintf(Hasło poprawne!)
```

Rys. 4. Źle zrealizowana implementacja sprawdzania hasła

Powyżej przedstawiony sposób weryfikacji hasła nie jest bezpieczny, ponieważ po napotkaniu pierwszego błędnego znaku zaprzestaje dalszego sprawdzania i wyświetla komunikat „Hasło niepoprawne". Tak zrealizowane sprawdzenie jest łatwym celem nawet dla SPA, ponieważ wystarczy obserwować przebiegi poboru mocy w zależności od czasu.

Sposobem na poprawę bezpieczeństwa mechanizmu weryfikacji hasła z rysunku 4 będzie refaktoryzacja kodu w taki sposób, aby sprawdzane było za każdym razem wszystkie pięć znaków i dopiero wtedy wyświetlanie komunikatu informującego o poprawności wprowadzonych danych.

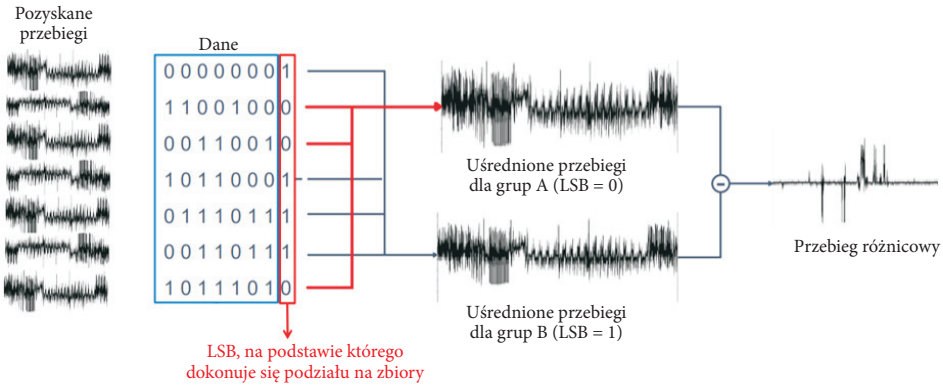


Rys. 5. Analiza poboru mocy urządzenia w zależności od liczby znaków poprawnie wprowadzanych do weryfikacji [5]

2.2. Różnicowa analiza poboru mocy (DPA)

Różnicowa analiza poboru mocy DPA (ang. *Differential Power Analysis*) tak jak SPA została przedstawiona w publikacji *Differential Power Analysis* z 1999 r. [6], choć odtajnione dokumenty NSA (ang. *National Security Agency*) wskazują, że metoda DPA była znana już przed 1995 r. W metodzie SPA rozpoznawanie wzorców dokonywane jest bez dodatkowego przetwarzania sygnałów. Metoda DPA skupia się głównie na obserwowaniu zmian w osi amplitudy dla ustalonego punktu w czasie, lecz przy zmiennych danych (wejściowych/wyjściowych), korzystając przy tym ze statystyki. Z tego powodu DPA wymaga pozyskania nieporównywalnie większej liczby przebiegów poboru mocy (z różnymi wariantami danych wejściowych/wyjściowych). Pozyskane przebiegi z przypisanymi im danymi wejściowymi i wyjściowymi dzieli się na dwie grupy. Podział dokonywany jest na podstawie danych wyjściowych z pierwszego S-boxa algorytmu szyfrującego. Pierwszą grupę stanowią przebiegi skorelowane z danymi wyjściowymi, których najmniej znaczący bit to „0”, a drugą przebiegi skorelowane z danymi, których najmniej znaczący bit to „1”. S-box (ang. *Substitution box*), czyli skrzynka podstawień, to podstawowy element

algorytmów klucza symetrycznego. Pobiera on pewną określoną liczbę bitów wejściowych i przekształca je w określoną liczbę bitów wyjściowych. Przebiegi dla każdej z grup należy uśrednić w celu wyeliminowania szumów, a następnie odjąć wartości przebiegów A od B. Efektem będzie przebieg wskazujący, gdzie występuje największa korelacja przebiegów (wartość różna od zera).



Rys. 6. Etapy różnicowej analizy mocy

Taki zabieg pozwala na wyselekcjonowanie miejsc, gdzie dokonywane są operacje na danym bicie. Klucze powszechnie stosowanych algorytmów są 128- lub 256- bitowe, co w przypadku chęci łamania ich w klasyczny sposób daje 2^{128} lub 2^{256} możliwości. 128-bitowy klucz można podzielić na 16 bajtów, gdzie każdy bajt można rozwiązać indywidualnie w przedstawiony sposób DPA. Testowanie każdego bajtu to maksymalnie 256 prób, co daje 16 * 256 = 4096 = 2^{12} możliwości. Jest to znacznie mniej niż korzystanie z metody *brute force*.

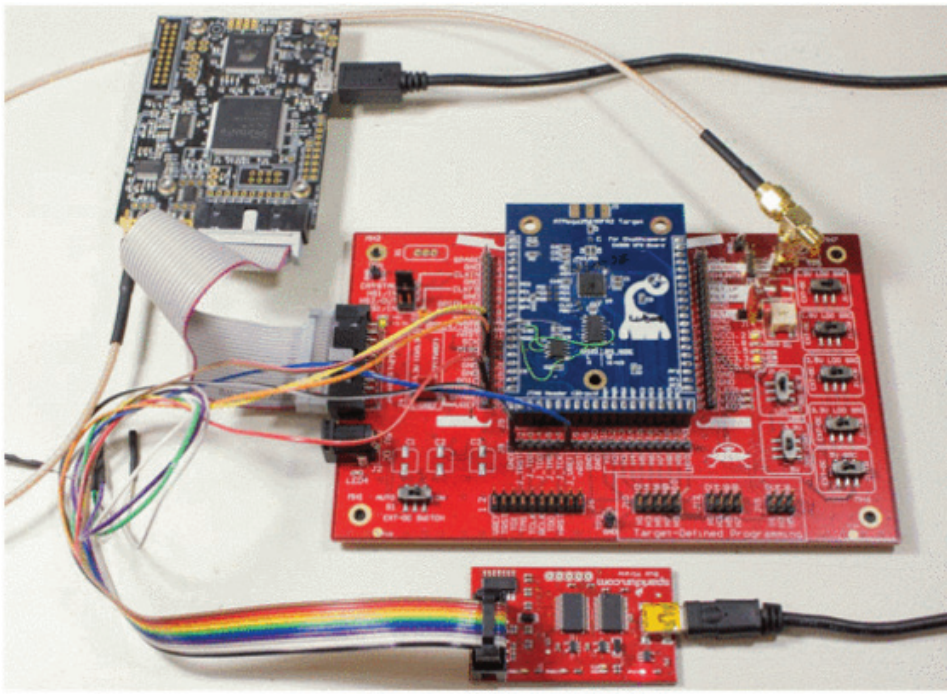
2.3. Przykład wykorzystania analizy kanału pobocznego w praktyce

W publikacji *IoT Goes Nuclear: Creating a ZigBee Chain Reaction* z 2017 r. [8] autorzy opisują sposób ataku na żarówki Philips Hue, czyli inteligentne źródła światła umożliwiające zdalne kontrolowanie ich ustawień przez właściciela. Lampki komunikują się ze sobą protokołem Zigbee Light Link, który działa w sieci bezprzewodowej (IEEE 802.15.4).

Pierwszym krokiem wykonanym przez atakujących było pominięcie „testu zbliżeniowego" żarówki. Urządzenie jest tak zaprogramowane, aby kontroler znajdujący się nie dalej niż 20 cm mógł wyłączać źródło światła. Autorom udało się zdalne kontrolowanie urządzenia na dystansie nawet do 400 metrów w zależności od warunków. Kluczem do pominięcia „testu zbliżeniowego" była podatność

protokołu PAN (*Personal Area Network*). Urządzenie odbiorcze oczekiwało na niezerową 32-bitową ramkę z ID kontrolera, jednak badacze wysłali ramkę zawierającą wartość 0, co spowodowało, że żarówka zrestartowała się do ustawień fabrycznych i uruchomiła się w trybie wstecznej kompatybilności, gdzie „test zbliżeniowy" nie jest przeprowadzany.

Kolejną fazą był atak na mechanizm używany do szyfrowania i weryfikacji aktualizacji oprogramowania żarówek. Philips Hue pierwszej i drugiej generacji wykorzystuje mikrokontrolery ATMEGA2564RFR2. Autorzy zaprojektowali swoją własną płytkę PCB (koloru niebieskiego na rys. 7), z umieszczonym mikrokontrolerem żarówki wraz z niektórymi peryferiami (np. pamięcią flash), aby dokonać analizy poboru mocy. Pomiary odbywały się z użyciem *ChipWhisperer-lite* (platforma sprzętowa przeznaczona do badań kanału pobocznego).



Rys. 7. *ChipWhisperer-lite* (na górze po lewej) podłączony do niestandardowej płytki PCB z zamontowanym ATMEGA2564RFR2 (środkowa niebieska płytka)
Źródło: *IoT Goes Nuclear: Creating a ZigBee Chain Reaction* [8]

Metody badawcze wykorzystane przez autorów to wcześniej opisana różnicowa analiza mocy (DPA) oraz korelacyjna analiza mocy (CPA) nakierowana na algorytm szyfrujący pliki aktualizacyjne. Poważnym błędem ze strony producenta inteligentnych żarówek było stosowanie tego samego klucza algorytmu symetrycznego

AES dla każdego egzemplarza lampki konkretnego modelu. Badacze byli w stanie określić prawidłowy klucz, którym zaszyfrowano pliki aktualizacji, co pozwoliło im na napisanie własnej „aktualizacji". Łatka oprogramowania była tym bardziej niebezpieczna, że po zainfekowaniu pierwszej żarówki potrafiła sama się rozprzestrzeniać (z powodu tego samego klucza dla każdego egzemplarza żarówki) pomiędzy kolejnymi urządzeniami, co dało „efekt łańcuchowy".

Błędy w implementacji Philips Hue, które pozwoliły z powodzeniem przeprowadzić atak:

— Użycie jednego symetrycznego klucza szyfrowania współdzielonego przez wiele urządzeń w celu ochrony procesu aktualizacji oprogramowania sprzętowego.
— Sprzęt podatny na analizę kanału bocznego. Mikrokontrolery nie są tak bezpieczne jak np. układy FPGA, jeżeli chodzi o szyfrowanie blokowe, gdyż są dużo wolniejsze i zazwyczaj wykonują operacje szeregowo, a nie równolegle.

2.4. Środki utrudniające analizę kanału pobocznego

Metody prostej oraz różnicowej analizy mocy mogą być niebezpieczne, ponieważ umożliwiają obejście konwencjonalnych środków bezpieczeństwa polegających między innymi na limitach mocy obliczeniowej współczesnych urządzeń. Ponadto wymienione metody mogą być nieinwazyjne (np. obserwacja emisji promieniowanej), przez co atakujący może dokonać kradzieży poufnych informacji bez wykrycia. Dlatego należy pamiętać o środkach zapobiegających takim atakom.

Aby zapobiec SPA, układ może generować dodatkowo szum, wykonywać przypadkowe procesy równolegle lub kluczowe operacje sprawdzania przeprowadzać niedeterministycznie. Natomiast zapobieganie DPA jest trudniejsze. Jedną z metod może być zmniejszenie stosunku sygnału do szumu (ekranowanie itd.) — im niższy stosunek, tym więcej śladów wymaganych do przeprowadzenia ataku. Można również wprowadzać fikcyjne operacje w celu zaciemnienia działania systemu. Przede wszystkim algorytmy kryptograficzne powinny możliwie często mieć zmieniane klucze.

3. Podsumowanie

Analiza kanału pobocznego to narzędzie, które pozwala atakującym korzystać z luk w implementacji sprzętu elektronicznego, obchodzą w ten sposób problemy z ograniczoną mocą obliczeniową. Należy być świadomym, że bezpieczeństwo systemu elektronicznego to nie tylko oprogramowanie, lecz także słabości sprzętu. Co ważne, przy pewnej przezorności i odpowiednich środkach zaradczych wskazanych w poprzednim rozdziale można zapobiegać takim atakom. Odnosząc się

do przykładu z podrozdziału 2.4, ochrona przed analizą kanału bocznego, która uniemożliwiłaby odzyskanie kluczy szyfrujących, jest zwykle zbyt kosztowna dla tanich urządzeń konsumenckich. Zamiast tego prostszym rozwiązaniem byłoby zapewnienie, że wyciek klucza z pojedynczego produktu nie uszkodzi całego ekosystemu składającego się z wielu urządzeń.

Źródło finansowania pracy — środki własne autorów.

Artykuł wpłynął do redakcji 17.04.2023. Zatwierdzono do publikacji 19.09.2023.

Konrad Szczepankiewicz https://orcid.org/0000-0003-3292-8113

Marian Wnuk https://orcid.org/0000-0003-4576-4023

LITERATURA

[1] Woudenberg J. van, O'Flynn C., *The Hardware Hacking Handbook: Breaking Embedded Security with Hardware Attacks*, No Starch Press, San Francisco, 2022, 265-322.

[2] Kocher P., Jaffe J., Jun B., Rohatgi P., *Introduction to differential power analysis*, J. Cryptogr. Eng., 1, 2011, https://doi.org/10.1007/s13389-011-0006-y [dostęp: 17.03.2023].

[3] *ATM card pin hacked with help of thermal cameras*, Hello engineers blog, http://hello-engineers.blogspot.com/2011/08/atm-card-pins-hacked-with-help-of.html [dostęp: 17.03.2023].

[4] Randolph M., Diehl W., *Power Side-Channel Attack Analysis: A Review of 20 Years of Study for the Layman*, Cryptography, 4, 2, 2020, https://doi.org/10.3390/cryptography4020015 [dostęp: 17.03.2023].

[5] *V4: Tutorial B3-1 Timing Analysis with Power for Password Bypass*, https://wiki.newae.com/V4:Tutorial_B3-1_Timing_Analysis_with_Power_for_Password_Bypass [dostęp: 17.03.2023].

[6] Kocher P., Jaffe J., Jun B., *Differential Power Analysis*, [w:] M. Wiener (ed.), Advances in Cryptology – CRYPTO '99, Lecture Notes in Computer Science, vol. 1666, 1999, 388-397.

[7] Bucci M., Luzzi R., Guglielmo M., Trifiletti A., *A countermeasure against differential power analysis based on random delay insertion*, IEEE International Symposium on Circuits and Systems, Kobe, Japan, 2005.

[8] Ronen E., O'Flynn C., Shamir A., Weingarten A.-O., *IoT Goes Nuclear: Creating a ZigBee Chain Reaction*, IEEE Symposium on Security and Privacy, San Jose, CA, USA, 2017.

K. SZCZEPANKIEWICZ, M. WNUK

Embedded Systems and their Vulnerabilities to Hardware Attacks

Abstract. The article describes the types of hardware attacks targeting embedded systems, countermeasures, and methods of modelling security threats. In currently used electronic devices, the software usually ensures very high level of security. On the other hand, hardware implementations, often leave vulnerabilities that attackers can use to extract information or disrupt the operation of the device. Being a user of embedded systems, critical for safety, you should be aware of what dangers they may be exposed to from the hardware side and to know the analysis of the side channel and fault injection.
Keywords: hardware attacks, side channel analysis, vulnerabilities of embedded systems, protection of devices against hardware attacks.
DOI: 10.5604/01.3001.0054.2897