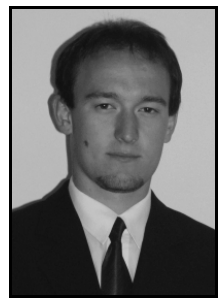
Piotr MAJKOWSKI, Tomasz WOJCIECHOWSKI, Maciej WOJTYŃSKI, Zbigniew KOTULSKI, Mariusz RAWSKI
POLITECHNIKA WARSZAWSKA, INSTYTUT TELEKOMUNIKACJI

Analiza możliwości sprzętowej kryptoanalizy szyfrów opartych na krzywych eliptycznych

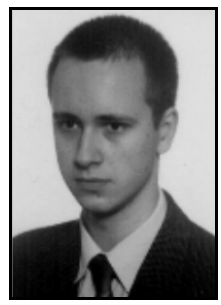
Inż. Piotr MAJKOWSKI



Uzyskał tytuł zawodowy inżyniera telekomunikacji na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej we wrześniu 2006 roku. Aktualnie kontynuuje edukację na studiach magisterskich w Instytucie Telekomunikacji wyżej wymienionego wydziału. Jego zainteresowania naukowe koncentrują się wokół zagadnień kryptografii (szczególnie asymetrycznej opartej o krzywe eliptyczne), kryptoanalizy, obliczeń rozproszonych oraz ich sprzętowych i programowych implementacji.

e-mail: P.Majkowski@elka.pw.edu.pl

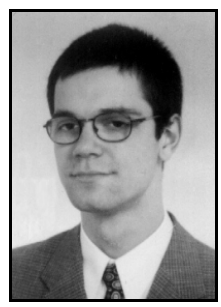
Inż. Tomasz WOJCIECHOWSKI



Uzyskał tytuł zawodowy inżyniera telekomunikacji na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej we wrześniu 2006 roku. Aktualnie kontynuuje edukację na studiach magisterskich w Instytucie Telekomunikacji wyżej wymienionego wydziału. Jego zainteresowania naukowe koncentrują się wokół zagadnień kryptografii (szczególnie asymetrycznej opartej o krzywe eliptyczne), kryptoanalizy, obliczeń rozproszonych oraz ich sprzętowych i programowych implementacji.

e-mail: T.Wojciechowski@elka.pw.edu.pl

Inż. Maciej WOJTYŃSKI



Uzyskał tytuł zawodowy inżyniera telekomunikacji na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej we wrześniu 2006 roku. Aktualnie kontynuuje edukację na studiach magisterskich w Instytucie Telekomunikacji wyżej wymienionego wydziału. Jego zainteresowania naukowe koncentrują się wokół zagadnień kryptografii (szczególnie asymetrycznej opartej o krzywe eliptyczne), kryptoanalizy, obliczeń rozproszonych oraz ich sprzętowych i programowych implementacji.

e-mail: M.Wojtynski@elka.pw.edu.pl

Prof. dr hab. inż. Zbigniew Adam KOTULSKI

Absolwent Wydziału Fizyki Technicznej i Matematyki Stosowanej Politechniki Warszawskiej, kierunku matematyka stosowana, pracownik Instytutu Podstawowych Problemów Techniki Polskiej Akademii Nauk i Wydziału Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. Autor lub współautor 3 monografii i ponad 100 publikacji naukowych z dziedziny zastosowań matematyki w zagadnieniach technicznych, bezpieczeństwa systemów informatycznych oraz kryptografii.

e-mail: zkotulsk@tele.pw.edu.pl

Dr inż. Mariusz RAWSKI

Otrzymał stopień magistra inżyniera na Wydziale Elektroniki Politechniki Warszawskiej w 1995 roku. Stopień doktora otrzymał na tym samym wydziale w 2000 roku. Obecnie jest adiunktem na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. Jego zainteresowania naukowe obejmują syntezę logiczną układów cyfrowych, narzędzia CAD dla syntezy i optymalizacji logicznej, projektowanie systemów cyfrowych z wykorzystaniem struktur programowalnych PLD.

e-mail: rawski@tele.pw.edu.pl

Streszczenie

Artykuł opisuje jednostkę sprzętową służącą do efektywnego rozwiązywania zagadnienia logarytmu dyskretnego na krzywej eliptycznych zdefiniowanej nad ciałem $GF(2^n)$ za pomocą równoległej wersji algorytmu rho Pollard'a. Układ zawiera moduł sumatora punktów na krzywej eliptycznej wykorzystujący do przeprowadzania operacji w ciele bazowym podmoduł korzystający z właściwości baz normalnych. Artykuł opisuje także generator kodu VHDL pozwalający na uogólnienie rozwiązania na dowolne ciała charakterystyki dwa dla których występuje gaussowska baza normalna. Analizy efektywności działania układu pozwoliły na oszacowanie czasu potrzebnego na kryptoanalizę krzywych z listy wyzwań firmy Certicom.

Słowa kluczowe: kryptografia asymetryczna, kryptoanaliza, krzywe eliptyczne, bazy normalne, obliczenia rozproszone, rho Pollard, ECDLP, ECC.

Cryptanalysis of elliptic curve based ciphers in reprogrammable structures

Abstract

This paper presents the FPGA implementation of parallel version of the rho Pollard algorithm used for solving a discrete logarithm problem in the elliptic curve addition of points on an elliptic curve defined over discrete field $GF(2^n)$. In proposed implementation a hardware module has been developed that performs arithmetic operations in the base field, using characteristic features of optimal normal bases. A special generator of the VHDL source code that generalizes ~~ze~~ the solution is also presented in this paper. The resulting FPGA cores has been used to estimate time necessary for cryptanalysis of curves from the Certicom Challenge List.

Keywords: public key cryptography, cryptanalysis, elliptic curves, optimal normal bases, distributed calculations, rho Pollard algorithm, ECDLP, ECC.

1. Wstęp

Kryptosystemy oparte na krzywych eliptycznych ECC (*Elliptic Curve Cryptography*) są najpoważniejszym kandydatem do zastąpienia najpopularniejszego w tej chwili kryptosystemu z kluczem publicznym, czyli RSA. Z uwagi na fakt, że nie jest znany żaden atak na krzywe eliptyczne, o złożoności podwykładniczej, pozwalają one na używanie znacznie krótszych kluczy niż RSA. Sceptycy zwracają uwagę, że problem rozwiązywania zagadnienia logarytmu dyskretnego na krzywych eliptycznych ECDLP (*Elliptic Curve Discrete Logarithm Problem*), który stanowi o ich bezpieczeństwie, nie został tak dokładnie przebadany jak np. problem faktoryzacji. Postęp badań w tej dziedzinie jak i wprowadzanie usprawnień w implementacjach algorytmów kryptoanalitycznych pragnie stymulować firma Certicom – światowej sławy specjalista w dziedzinie krzywych eliptycznych. Firma ta w 1997 roku opublikowała listę *"krzywych-wyzwań"* i za ich złamanie wyznaczyła wysokie nagrody [9]. Obecnie część z umieszonych na liście zagadnień została już rozwiązana przy użyciu dużej liczby połączonych ze sobą komputerów. Celem autorów pracy było pokazanie możliwych alternatywnych rozwiązań, ze szczególnym uwzględnieniem metod sprzętowego wspomagania obliczeń z zastosowaniem układów reprogramowalnych. Zbadana została efektywność takiego podejścia oraz przeanalizowano możliwość ponownego złamania krzywych z listy Certicom. Podstawowymi elementami obliczeniowymi są układy FPGA z zaprogramowanym modułem *HardRho*. Moduł ten został zaprojektowany przez autorów pracy i realizuje równoległą wersję algorytmu rho Pollarda – najlepszego znanego obecnie algorytmu rozwiązującego ECDLP. Zastosowano także szereg optymalizacji na poziomie

arytmetyki krzywych eliptycznych, działań w ciele bazowym oraz w warstwie implementacyjnej projektu. Zaprezentowany został generator kodu źródłowego w języku VHDL, który służy to automatyzacji procesu tworzenia opisu układu HardRho dla różnych krzywych.

Artykuł ten stanowi rozszerzenie referatu zaprezentowanego na konferencji RUC w 2007 roku [6].

2. ZARYS TEORII

Kompletne przedstawienie teorii krzywych eliptycznych wykracza poza ramy tego artykułu, dlatego też ograniczono się jedynie do zamieszczenia informacji niezbędnych do zrozumienia referatu. Pełne definicje, których uproszczone formy przedstawiono poniżej, można odnaleźć w pozycjach wymienionych w literaturze.

Operacje w ciele GF(2^n)

Skuteczna implementacja algorytmów operujących na krzywych eliptycznych wymaga efektywnej implementacji arytmetyki w ciele, nad którym zdefiniowana jest krzywa. W niniejszym artykule zakłada się, że krzywa została określona nad ciałem charakterystyki dwa, którego definicja znajduje się poniżej.

Ciało GF(2^n) (*Galois Field*) – elementami ciała są binarne, *n*-wymiarowe wektory współrzędnych w ustalonej bazie. Działaniem dodawania jest XOR wykonany na poszczególnych współrzędnych [1].

Sposób oraz szybkość przeprowadzania operacji mnożenia w ciele GF(2^n) zależy nie tylko od środowiska, w którym są przeprowadzane obliczenia (implementacja sprzętowa lub programowa), ale także od przyjętej reprezentacji (bazy) elementów ciała. Baza ta określa interpretację poszczególnych bitów w zapisie elementu ciała. W rozwiązaniach programowych zwykle używa się tzw. reprezentacji potęgowej (postaci $\left(1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^n\right)$) natomiast implementacje sprzętowe korzystają z tzw. baz normalnych (postaci $\left(\beta, \beta^2, \beta^{2^2}, \ldots, \beta^{2^n}\right)$). Użycie tej reprezentacji umożliwia bardzo efektywne wykonywanie operacji mnożenia, a podnoszenie do kwadratu wymaga jedynie cyklicznej rotacji wektora.

Niezwykle przydatną właściwością baz normalnych jest fakt, że struktura układu mnożącego nie zależy od elementów wejściowych, a jedynie od rozmiaru ciała i może być określona już w fazie projektowania układu. Przeprowadzenie operacji mnożenia elementów ciała bazowego jest wykonywane przy wykorzystaniu specjalnej binarnej macierzy (tzw. *macierzy mnożenia*) [4]. *Złożonością takiej macierzy nazywamy liczbę zawartych w niej jedynek. Im mniejsza złożoność, tym sprawniej może być przeprowadzona operacja mnożenia. Najlepsze wyniki można uzyskać dla optymalnych baz normalnych ONB (Optimal Normal Bases),* zwanych także bazami typu I i II, w których macierz mnożenia ma dokładnie 2*n* - 1 jedynek. Niestety nie dla wszystkich rozmiarów ciał istnieją bazy optymalne i w większości przypadków trzeba użyć macierzy mnożenia o większej złożoności (są to tzw. bazy wyższych typów). Do obliczenia macierzy mnożenia w przypadku implementacji opisywanej w niniejszym artykule służą funkcje zawarte w bibliotece ciała skończonego wchodzącej w skład generatora kodu HardRho. Szczegóły znajdują się w podpunkcie *Automatyczna generacja kodu źródłowego jednostki HardRho*.

Podstawy matematyczne optymalnych baz normalnych można odnaleźć w pracach Gao [2] oraz Gawineckiego i Szmidta [3], jednak z punktu widzenia projektanta systemu ciekawszą propozycją będzie na pewno norma IEEE [4], gdzie znajdują się wszystkie potrzebne algorytmy.

Krzywe eliptyczne

Krzywa eliptyczna *E* nad ciałem **GF(2^n)** jest zdefiniowana przez następujące równanie:

$$y^2 + xy = x^3 + ax^2 + b \tag{1}$$

gdzie *a* i *b* są elementami ciała ***GF(2^n).***

Z punktu widzenia kryptografii interesujący jest zbiór punktów spełniających powyższe równanie, uzupełniony dodatkowo o specjalny punkt ***O*** zwany *punktem w nieskończoności*. Zbiór ten, po zdefiniowaniu na nim działania dodawania, tworzy strukturę matematyczna zwaną grupą. Działanie dodawania punktów na krzywej można określić na dwa sposoby: analitycznie lub geometrycznie. Szczegóły można odnaleźć w [1]. Działanie obliczania wielokrotności punktu rozumiane jest jako wielokrotne dodanie do siebie danego punktu.

Liczba operacji w ciele, które należy wykonać, by dodać do siebie dwa punkty na krzywej eliptycznej, zależy od wyboru reprezentacji (współrzędnych) punktów. Najczęściej używa się *reprezentacji afinicznej* (punkt posiada dwie współrzędne *(X, Y)*) lub *rzutowej* (punkt przedstawiony jest jako trójka liczb *(X, Y, Z)*) [1]. Największą korzyścią wynikającą z użycia reprezentacji rzutowej jest możliwość wykonania dodawania punktów bez konieczności obliczania odwrotności w ciele, kosztem zwiększenia liczby mnożeń. Fakt ten jest niezwykle przydatny przy implementacjach sprzętowych, ponieważ oprócz zwiększenia szybkości pozwala także na redukcję rozmiaru układu.

Problem logarytmu dyskretnego

Kryptosystemy asymetryczne bazują zwykle na trudnych obliczeniowo problemach matematycznych. Jak to zostało wspomniane we wstępie kluczowym zagadnieniem, jeśli chodzi o bezpieczeństwo *ECC* jest zagadnienie logarytmu dyskretnego na krzywej eliptycznej *ECDLP* .

Niech dla danej krzywej eliptycznej ***E*** zdefiniowanej nad ciałem skończonym będą wybrane punkty *P* rzędu *n* oraz *Q* będący wielokrotnością punktu *P*. Należy odnaleźć nieujemną liczbę całkowitą *l* mniejszą od *n* taką, że *Q* = *l P*. Liczbę *l* nazywamy ***dyskretnym logarytmem Q o podstawie P***.

Algorytm rho Pollard'a

Najlepszym ze znanych obecnie algorytmów rozwiązujących zagadnienie logarytmu dyskretnego na ogólnej krzywej eliptycznej (niespełniającej np. kryteriów ataku MOV [4]) jest algorytm rho Pollard'a. Jest on oparty na błądzeniu losowym po punktach krzywej eliptycznej w oczekiwaniu na kolizję, czyli natrafienie ponownie na ten sam punkt. Błądzenie oparte jest na specjalnej funkcji, której podstawowym elementem jest obliczanie sumy dwóch punktów na krzywej eliptycznej. Jeden z punktów wchodzących w skład tej sumy jest tzw. punktem bieżącym, który jest modyfikowany w każdej rundzie algorytmu, drugi natomiast pochodzi z przygotowanej w fazie inicjalizacyjnej algorytmu tablicy (w tablicy tej znajdują się także powiązane z tym punktem dwie liczby całkowite). Szczegółowy opis algorytmu może zostać odnaleziony w [8].

Algorytm rho występuje zarówno w wersji sekwencyjnej jak i równoległej. W niniejszej pracy wykorzystano drugą z wymienionych metod. Uwzględniając zrównoleglenie zaproponowane przez van Oorschota i Wienera w [11] otrzymujemy oszacowanie na oczekiwaną liczbę kroków algorytmu, które należy wykonać by otrzymać kolizję, równą:

$$\frac{\sqrt{n \cdot \pi / 2}}{M}$$

gdzie *n* jest rzędem punktu, a *M* jest liczbą jednostek biorących udział w obliczeniach.

W metodzie równoległej oprócz jednostek obliczeniowych występuje także serwer centralny, którego zadaniem jest kolekcjonowanie punktów odwiedzonych w trakcie błądzenia przez poszcze-

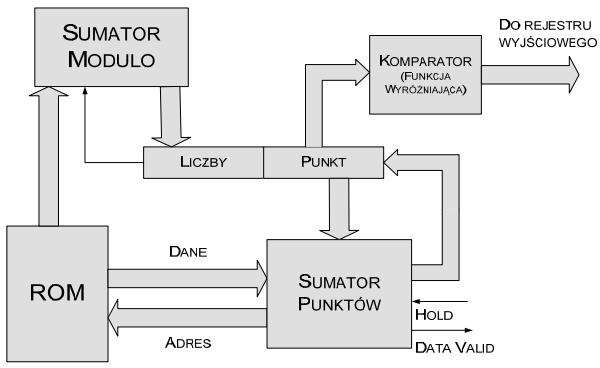
gólne jednostki obliczeniowe oraz wykrywanie kolizji. Przesyłanie wszystkich punktów odwiedzonych w trakcie działania algorytmu jest oczywiście niemożliwe, z uwagi na zasoby komunikacyjne oraz niezbędną do przechowania tych punktów pamięć. Z tego powodu stosuje się tzw. *kryterium wyróżniające*, czyli funkcję która z całego zbioru punktów wybiera te które mają pewną, ściśle określoną cechę, np. współrzędną X mniejszą od zadanego progu. Tylko takie punkty wysyłane są do serwera.

3. Opis implementacji

Układ HardRho

W niniejszym paragrafie przedstawiona zostanie budowa jednostki HardRho. Przedstawiona została ona jedynie w sposób poglądowy – szczegóły można znaleźć w [7].

Struktura jednostki obliczeniowej opisanej w języku VHDL została zobrazowana na rysunku 1. Wyróżniono 5 bloków: sumator punktów na krzywej eliptycznej, sumator liczb całkowitych modulo rząd punktu, wewnętrzna pamięć ROM oraz rejestry przechowujące wartość aktualnie przetwarzanego punktu i odpowiadających mu liczb całkowitych. Jednostka HardRho działa pod nadzorem zewnętrznego kontrolera odpowiedzialnego za komunikację z serwerem głównym. Punktem styku pomiędzy kontrolerem a jednostką obliczeniową jest bufor FIFO w którym przechowywane są punkty wyróżnione przed wysłaniem do serwera.



Rys. 1. Architektura układu HardRho
Fig. 1. Architecture of HardRho module

Rejestry przechowujące wartość aktualnie przetwarzanego punktu (punkt bieżący) oraz powiązanych z nim liczb całkowitych są uaktualniane podczas kolejnych iteracji algorytmu. Równocześnie stanowią one dane wejściowe do iteracji następnych, co umożliwia realizację błądzenia przypadkowego po punktach krzywej eliptycznej. Z rejestrami powiązany jest komparator realizujący *kryterium wyróżniające* algorytmu i sygnalizujący wystąpienie konieczności odebrania danych po znalezieniu punktu wyróżnionego. Próg funkcji wyróżniającej może być dowolnie określony w fazie automatycznego generowania opisu jednostki HardRho, aby osiągnąć pożądany kompromis pomiędzy opóźnieniem obliczeń, a rozmiarem dostępnej pamięci serwera głównego.

Wewnętrzna pamięć ROM układu zawiera tablicę z obliczonymi wcześniej liczbami całkowitymi oraz punktami na krzywej eliptycznej. Ta wspólna dla wszystkich jednostek HardRho tablica jest tworzona automatycznie przez generator. Pamięć zawiera 16 słów, każde cztery razy większe niż rząd ciała (oznaczony n). Na każde słowo składają się dwie współrzędne afiniczne punktu na krzywej eliptycznej ($2n$) oraz dwie liczby naturalne ($2n$). Każda iteracja algorytmu rho Pollard'a wymaga odczytu jednego słowa z pamięci. Adres dla kolejnej iteracji stanowią 4 najmłodsze bity współrzędnej X punktu obliczonego w iteracji poprzedniej, jest to tzw. *funkcja pozycjonująca* algorytmu.

Sumator liczb całkowitych modulo rząd punktu na krzywej eliptycznej realizuje obliczenia niezbędne do rozwiązania ECDLP, ale związane jedynie w sposób pośredni z samym sumowaniem punktów. Dane wejściowe do sumatora dostarczane są przez pamięć ROM adresowaną przez moduł sumatora punktów krzywej eliptycznej. Każdej iteracji algorytmu rho Pollard'a odpowiadają dwa sumowania liczb całkowitych modulo liczba reprezentująca rząd punktu na krzywej eliptycznej. Liczba ta jest powiązana z warunkami początkowymi algorytmu i jest znana przed rozpoczęciem obliczeń, zatem stanowi ona jeden z parametrów automatycznego generatora kodu VHDL. Zarówno sumowanie, jak i redukcja modulo są dokonywane przy wykorzystaniu tych samych zasobów, co pozwala na zmniejszenie powierzchni układu.

Sumator punktów krzywej eliptycznej składa się z jednostki mnożącej (generowanej dla danego rozmiaru ciała przez wspomnianą aplikację), zestawu rejestrów tymczasowych, modułów dodawania (logiczny XOR), podnoszenia do kwadratu (cykliczna rotacja) oraz automatu kontrolnego. Uzyskanie pierwszej sumy punktów trwa 13 cykli zegara, każda kolejna otrzymywana jest co 11 cykli zegara (efekt zastosowania mechanizmu potoku). Jest to minimalna liczba cykli zegara, potrzebnych na obliczenie sumy punktów przy obecnej architekturze układu z uwagi na fakt, że dodawanie mieszane wymaga dokładnie 11 mnożeń w ciele bazowym. Jednostka mnożąca jest zatem wykorzystywana w sposób maksymalnie efektywny przeprowadzając działania w każdym cyklu obliczeń. Dokładny opis struktury i algorytmu działania sumatora punktów krzywej eliptycznej został przedstawiony w [6].

Automatyczna generacja kodu źródłowego jednostki HardRho

Generator kodu VHDL układu HardRho służy do zwiększenia ogólności przedstawionego rozwiązania poprzez automatyzację procesu tworzeniu opisu układu. Elementami, które należy każdorazowo generować osobno dla poszczególnych krzywych eliptycznych są: opis jednostki mnożącej w ciele bazowym, plik z zapisem zawartości pamięci ROM, plik zawierający parametry wykorzystywane następnie w dalszej części opisu układu.

Aplikacja generatora powstała w ramach pracy magisterskiej [5] na podstawie zaimplementowanej wcześniej biblioteki programistycznej ciała skończonego. Pakiet ten udostępnia programiście, oprócz metod służących do przeprowadzania arytmetyki w ciele skończonym z wykorzystaniem baz normalnych, także funkcje takie jak: odnajdowanie typu bazy o najmniejszej złożoności dla danego ciała oraz obliczanie macierzy mnożenia dla znalezionej bazy, która następnie przekładana jest na równania logiczne zapisane w języku VHDL. Dodatkowo w celu wygenerowania zawartości ROM niezbędne było zaimplementowanie funkcji wymaganych w fazie inicjalizacyjnej algorytmu rho Pollard'a, czyli arytmetyki na krzywych eliptycznych (sumowanie punktów oraz obliczanie wielokrotności punktu). Kolejnym zastosowaniem prezentowanej biblioteki jest możliwość obliczania *macierzy konwersji* pomiędzy bazami potęgowymi i normalnymi. Jest to niezbędna funkcjonalność wykorzystywana w środowiskach, w których część obliczeń jest realizowana sprzętowo, a część programowo.

Szczegóły dotyczące generatora oraz biblioteki ciała skończonego można odnaleźć we wspomnianej pracy magisterskiej.

4. Wyniki implementacji

Korzystając z generatora kodu VHDL uzyskano kody układu HardRho dostosowane do pierwszych sześciu krzywych z listy Certicom (zastosowano oryginalne nazewnictwo; cyfra w nazwie oznacza rozmiar ciała nad którym zdefiniowana jest krzywa). Wynik syntezy poszczególnych projektów wykonanych w systemie Quartus II dla układu FPGA typu STRATIX II firmy Altera o symbolu EP2S90F1508C3 zamieszczono w tabeli 1.

Dzięki znajomości częstotliwości pracy układu oraz korzystając z zaprezentowanego w podpunkcie *algorytm rho Pollard'a* wzoru na złożoność czasową można oszacować czas potrzebny na złamanie poszczególnych krzywych z listy Certicom.

Tab. 1. Wyniki syntezy logicznej dla krzywych z listy Certicom
Tab. 1. Results of logic synthesis for curves from Certicom list

Nazwa	Zajętość ALUT [komórki]	Zajętość rejestrów [komórki]	Zajętość układu [%]	Częstotliwość pracy [MHz]
ECC2_79	7938	2096	12	118,71
ECC2_89*	8286	2383	13	141,36
ECC2_97	14508	2551	21	58.71
ECC2_109	19582	2865	29	59,6
ECC2_131*	16732	3499	25	98.97
ECC2_163	31002	4284	23	48.92

* dla krzywej występuje optymalna baza normalna (typu II). Przejawia się to m.in. mniejszą zajętością komórek logicznych oraz większą częstotliwością działania układu.

W tabeli 2 zostały przedstawione czasy łamania krzywej uzyskane w pionierskim eksperymencie – wg danych firmy Certicom (*Czas*), szacowane czasy działania pojedynczej jednostki HardRho (*Czas HardRho*) oraz ten sam czas przeskalowany o czynnik równy liczbie jednostek obliczeniowych biorących udział w pionierskim łamaniu danej krzywej (*Czas HardRho znormalizowany*). Krzywe ECC2_131 oraz ECC2_163, które nie zostały jeszcze nigdy złamane, przeskalowano o czynnik użyty także dla krzywej ECC2_109, czyli o 10000.

Tab. 2. Czasy potrzebne na złamanie krzywych z listy Certicom
Tab. 2. Times need to successful attacks on curves from Certicom list

Krzywa	ECC2_79	ECC2_89	ECC2_97	ECC2_109	ECC2_131	ECC2_163
Liczba jednostek	20	70	195	10000	nie złamana, przyjęto 10000	nie złamana, przyjęto 10000
Czas	6 dni	16 dni	31 dni	549 dni	nie złamana	nie złamana
Czas HardRho	18 godzin	20 dni	32 miesiące	135 lat	137 000 lat	$2{,}2 \cdot 10^{10}$ lat
Czas HardRho znorm.	1 godzina	7 godzin	94 godziny (ok. 4 dni)	116 godzin (ok. 5 dni)	13,7 lat	$2{,}2 \cdot 10^{6}$ lat

Warto zwrócić uwagę, że zaprezentowane powyżej wyniki wykazują, że dysponując liczbą układów FPGA równą liczbie komputerów użytych do pierwszego łamania poszczególnych krzywych z listy Cericom, można powtórzyć te obliczenia uzyskując znacznie krótszy czas wymaganych obliczeń.

5. Podsumowanie

W artykule zaprezentowano jednostkę sprzętową HardRho wspomagającą obliczenia niezbędne do rozwiązania problemu logarytmu dyskretnego w grupie punktów krzywej eliptycznej (ECDLP). Problem ten leży u podstaw bezpieczeństwa kryptosystemów opartych na krzywych eliptycznych, obecnie najpoważniejszego konkurenta dla kryptosystemu RSA.

Przedstawiono architekturę jednostki wraz z zastosowanymi rozwiązaniami zwiększającymi efektywność obliczeń i ogólność rozwiązania. Zaprezentowano wykorzystanie baz normalnych dla przyspieszenia obliczeń w ciele bazowym, optymalizację na poziomie współrzędnych punktów dla efektywniejszej realizacji arytmetyki na krzywej eliptycznej oraz usprawnienia w warstwie implementacyjnej.

Zaprezentowany został system umożliwiający automatyzację procesu tworzenia jednostki sprzętowej. W skład systemu weszły: automatyczny generator kodu źródłowego w języku VHDL oraz towarzyszące mu oprogramowanie (w tym biblioteki funkcji związanych z obliczeniami na krzywych eliptycznych).

Przedstawione zostały osiągnięte wyniki syntezy układu dla różnych krzywych eliptycznych z listy „wyzwań" firmy Certicom. W szczególności ukazane zostały różnice w efektywności rozwiązania w zależności od typu bazy normalnej dla danego rozmiaru ciała bazowego. Osiągnięte wyniki porównano ze znanymi atakami kryptoanalitycznymi na krzywe z listy firmy Certicom.

Przeprowadzone porównania wskazują na wysoką efektywność sprzętowego wspomagania obliczeń kryptoanalitycznych. Zastosowanie technologii układów reprogramowalnych umożliwia ścisłe dopasowanie architektury jednostki sprzętowej nie tylko do rozmiaru ciała bazowego, ale również do konkretnej krzywej eliptycznej, na którą kierowany jest atak. Autorzy zamierzają kontynuować prace nad zaprezentowaną jednostką w celu dalszego zwiększania efektywności sprzętowej kryptoanalizy systemów opartych na krzywych eliptycznych.

Praca naukowa finansowana ze środków na naukę w latach 2007-2010 jako projekt badawczy nr N517 003 32/0583.

6. Literatura

[1] Blake Ian, Seroussi Gadiel, Smart Nigel. Krzywe eliptyczne w kryptografii. WNT 2004
[2] Gao Shuhong, Lenstra W. Hendrik. Optimal Normal Bases.1992
[3] Gawinecki Jerzy, Szmidt Janusz. Zastosowanie ciał skończonych i krzywych eliptycznych w kryptografii. Wojskowa Akademia Techniczna 1999.
[4] IEEE P1363. Standard Specyfications for Public Key Cryptography. Draft 13. 1999
[5] Majkowski Piotr. Automatyzacja procesu tworzenia sprzętowego narzędzia służącego do rozwiązywania zagadnienia logarytmu dyskretnego na krzywych eliptycznych. praca magisterska napisana pod opieką prof. Zbigniewa Kotulskiego. Politechnika Warszawska 2008
[6] Majkowski Piotr, Wojciechowski Tomasz. Wojtyński Maciej. Rawski Mariusz. Realizacja jednostki wspomagającej kryptoanalizę szyfrów opartych na krzywych eliptycznych w strukturach reprogramowalnych. RUC 2007
[7] Majkowski Piotr, Wojciechowski Tomasz. Wojtyński Maciej. Rawski Mariusz. System sprzętowo-programowy służacy do rozproszonej kryptoanalizy szyfrów opartych na krzywych eliptycznych. ENIGMA 2007
[8] Menezes Alfred , Hankerson Darrel, Vanstone Scott. Guide to elliptic curve cryptography. Springer 2004
[9] Certicom. ECC Challenge. www.certicom.com/download/aid-111/cert_ecc_challenge.pdf
[10] Pollard J.M. Monte Carlo methods for index computation (mod p). Math. Comp. 32, 918-924. 1978
[11] van Oorschot P.C., Wiener M.J. Parrallel collision search with cryptoanalytic applications. Journal of Cryptography, volume 12, number 1, 1-12, 1999

Artykuł recenzowany