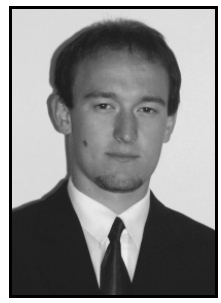
Piotr MAJKOWSKI, Tomasz WOJCIECHOWSKI, Maciej WOJTYŃSKI, Mariusz RAWSKI
POLITECHNIKA WARSZAWSKA, INSTYTUT TELEKOMUNIKACJI

Realizacja jednostki wspomagającej kryptoanalizę szyfrów opartych na krzywych eliptycznych w strukturach reprogramowalnych

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, kryptoanalizy, obliczeń rozproszonych oraz ich implementacji. Od października 2005 jest pracownikiem firmy Wincor-Nixdorf Polska, gdzie zajmuje się systemami płatniczymi.

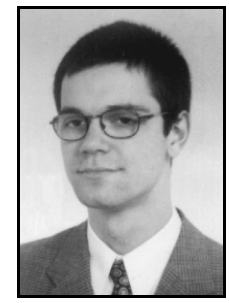
e-mail: P.Majkowski@elka.pw.edu.pl

Inż. Tomasz WOJCIECHOWSKI

Uzyskał tytuł zawodowy inżyniera telekomunikacji na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. Kontynuuje edukację na studiach magisterskich w Instytucie Telekomunikacji wyżej wymienionego wydziału. Równolegle studiuje matematykę na Uniwersytecie Warszawskim. Jego zainteresowania naukowe koncentrują się wokół tematyki projektowania układów cyfrowych, kryptografii oraz kryptoanalizy. Pracuje na stanowisku projektanta układów cyfrowych w dziale multimediów firmy Evatronix SA.

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

Inż. Maciej WOJTYŃSKI



We wrześniu 2006 roku uzyskał tytuł zawodowy inżyniera telekomunikacji Wydziału Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. Aktualnie uczęszcza na studia magisterskie w Instytucie Telekomunikacji wyżej wymienionego wydziału. Jego zainteresowania koncentrują się wokół projektowania układów cyfrowych z zastosowaniem w kryptografii i przetwarzaniu sygnałów. Od lutego 2006 roku pracuje w dziale multimediów firmy Evatronix SA na stanowisku projektanta układów cyfrowych.

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

Dr inż. Mariusz RAWSKI

Otrzymał tytuł 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 dodawania punktów na krzywej eliptycznej zdefiniowanej nad ciałem ***GF(2ⁿ).*** Układ zawiera moduł wykonujący operacje arytmetyczne w ciele bazowym, korzystający z właściwości optymalnych baz normalnych. Wyniki efektywności działania układu pozwoliły następnie na oszacowanie czasu potrzebnego na kryptoanalizę krzywej ECC2-89 (jednej z listy wyzwań firmy Certicom) za pomocą równoległej wersji algorytmu Rho Pollarda.

Słowa kluczowe: kryptoanaliza, krzywe eliptyczne, optymalne bazy normalne, Rho Pollard, ECDLP, ECC.

Implementation of module for cryptanalysis of elliptic curve ciphers in reprogrammable structures

Abstract

This paper presets the FPGA implementation of algorithm for addition of points on an elliptic curve defined over discrete field ***GF(2ⁿ)***. In proposed implementation a module was used that performs arithmetic operations in the base field, using characteristic features of optimal normal bases. The resulting FPGA core was used to estimate time necessary to cryptanalyze curve ECC2-89 (the one from the Certicom Challenge List) using parallel version of Pollard Rho algorithm.

Keywords: cryptanalysis, elliptic curves, optimal normal bases, Rho Pollard algorithm, ECDLP, ECC.

1. Wstęp

Termin kryptoanaliza niewtajemniczonemu czytelnikowi kojarzy się jedynie ze złamaniem Enigmy i Johnem Nashem – bohaterem filmu „Piękny umysł" - wykonującym w pamięci skomplikowane operacje na macierzach. Jednak rzeczywistość odbiega nieco od obrazu tworzonego przez ludzką wyobraźnię. Kryptoanaliza już dawno przestała być domeną sprytnych i spostrzegawczych językoznawców, którzy używając kartki papieru, gęsto wypełnionej krzyżówką permutacji słów kluczowych, potrafili z gąszczu znaków wydobyć użyteczną informację. Obecnie za kryptografią stoi rozbudowany i skomplikowany aparat matematyczny oparty na teorii liczb – dziale matematyki, jeszcze do niedawna uznawanym za całkowicie niepraktyczny. Wielkie liczby pierwsze, krzywe eliptyczne, trudne problemy obliczeniowe to podstawowe elementy, z których buduje się współczesne systemy kryptograficzne. Bezpieczeństwo zaszyfrowanej wiadomości nie opiera się już na braku wiedzy jak odszyfrować wiadomość. Metoda deszyfracji mimo, że jest powszechnie dostępna, to bez znajomości klucza jest bardzo kosztowna obliczeniowo.

Nowe metody szyfrowania wymuszają nowe narzędzia niezbędne do łamania szyfrów. Współcześni kryptoanalitycy, uzbrojeni w potężne superkomputery i klastry złożone z dużej liczby stacji roboczych, wciąż próbują rozkładać na czynniki pierwsze coraz to większe liczby, łamać coraz to dłuższe klucze. Wyzwania, rzucane przez czołowe firmy w branży kryptograficznej (np. RSA lub Certicom Challenge [7, 8]) podsycają atmosferę ciągłego wyścigu.

Autorzy pracy podjęli wyzwanie firmy Certicom – głównego specjalisty od krzywych eliptycznych. Wybrano krzywe eliptyczne, ponieważ stały się one w ostatnich latach bardzo poważnym konkurentem dla najpopularniejszego obecnie systemu z kluczem publicznym, czyli RSA. Zdecydowano się na podjęcie próby złamania krzywej ECC2-89. Zadanie to zostało już rozwiązane w roku 1998, z użyciem 70 komputerów połączonych w sieć pracujących około 16 dni. W niniejszym artykule została zaprezentowana jednostka dodawania punktów na krzywej eliptycznej, która umożliwia wykonanie tego zadania w porównywalnym czasie. Użyto jednej, niedrogiej płytki zawierającej układ FPGA typu EP2C35F672C6 z rodziny Cyclone II, która według producenta (Altera) kosztuje około 150$. Do realizacji wspomnianego celu wykorzystane zostały właściwości optymalnych baz normalnych, mieszane reprezentacje punktów na krzywej eliptycznej oraz algorytm Rho Pollarda.

2. Zarys teorii

Krzywe eliptyczne i problem logarytmu dyskretnego

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

Ciało *GF(2^n)* – ang. *Galois Field* (oznaczane także F_{2^n}); elementami ciała są binarne, *n* wymiarowe wektory współrzędnych w ustalonej bazie. Działaniem dodawania jest XOR wykonany na poszczególnych współrzędnych. Implementacja mnożenia i dzielenia zależy od wybranej bazy ([1] str. 23).

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

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

gdzie $a, b \in GF(2^n)$.

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].

Kryptosystemy asymetryczne bazują zwykle na trudnych obliczeniowo problemach matematycznych. Kluczowym zagadnieniem, jeśli chodzi o bezpieczeństwo ***ECC*** **(**ang. *Elliptic Curve Cryptography*) jest zagadnienie logarytmu dyskretnego na krzywej eliptycznej ***ECDLP*** (ang. *Elliptic Curve Discrete Logarithm Problem*) określone następująco.

Mamy daną krzywą eliptyczną ***E*** zdefiniowaną nad ciałem skończonym, punkt *P* rzędu *n* oraz punkt *Q* będący wielokrotnością punktu *P*. Należy odnaleźć liczbę całkowitą $l \in \langle 0, n-1 \rangle$ taką, że $Q = l \cdot P$. Liczbę l nazywamy ***dyskretnym logarytmem Q o podstawie P.***

Istnieje wiele metod, które można zastosować do rozwiązania tego problemu [1, 4]. Jednym z najlepszych obecnie znanych rozwiązań jest równoległa wersja algorytmu Rho Pollarda [4]. Z punktu widzenia niniejszego artykułu istotny jest fakt, że wykorzystuje on jedynie operację dodawania punktów na krzywej eliptycznej, dlatego problem obliczania wielokrotności punktu na krzywej został w tej pracy całkowicie pominięty.

Operacje w ciele *GF(2^n)*

Skuteczna implementacja algorytmów operujących na krzywych eliptycznych wymaga efektywnej implementacji działań w ciele, nad którym zdefiniowana jest krzywa. Szybkość ich wykonywania zależy nie tylko od środowiska, w którym są przeprowadzane obliczenia (implementacja sprzętowa lub programowa), ale także od przyjętej reprezentacji (bazy) elementów ciała i determinowanych przez nią algorytmów wykonywania operacji arytmetycznych. W rozwiązaniach programowych zwykle używa się tzw. reprezentacji potęgowej (postaci $(1, \alpha, \alpha^2, ..., \alpha^{n-1})$), natomiast implementacje sprzętowe korzystają z tzw. baz normalnych (postaci $(\beta, \beta^2, ..., \beta^{2^{n-1}})$). Użycie tej reprezentacji umożliwia bardzo efektywne wykonywanie operacji mnożenia, a podnoszenie do kwadratu wymaga jedynie cyklicznej rotacji w lewo.

Niezwykle przydatną właściwością baz normalnych jest fakt, że struktura układu mnożącego (określona przez tzw. macierz mnożenia) nie zależy od elementów wejściowych, a jedynie od rozmiaru ciała i może być określona już w fazie projektowania układu ([3] Algorytm A.6.2).

Najlepsze wyniki można uzyskać dla optymalnych baz normalnych ***ONB*** (ang. *Optimal Normal Bases*) typu *I* i *II,* które istnieją tylko dla niektórych rozmiarów ciała ***GF(2^n)*** ([3] Algorytm A.6.2). Dla interesującej nas krzywej ECC2-89 istnieje ***ONB*** typu *II*.

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

Wybór odpowiedniej reprezentacji punktów

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] str. 69). 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. Interesujący jest przypadek, gdy trzecia współrzędna jednego z punktów rzutowych jest równa 1. Sytuacja taka nazywana, *dodawaniem mieszanym* – ponieważ de facto jeden z punktów jest we współrzędnych afinicznych, pozwala na jeszcze efektywniejsze wykonywanie operacji dodawania punktów. Metoda ta nie zawsze może być wykorzystana z uwagi na koszty konwersji z reprezentacji rzutowej do afinicznej, które dla wynoszą 1O+4M.

W poniższej tabeli zamieszczono liczbę operacji, które należy wykonać dodając dwa punkty na krzywej (O-obliczanie odwrotności, M – wykonanie mnożenia, K – podniesienie do kwadratu, koszt operacji dodawania jest pomijalny).

Tab. 1. Koszt wykonania operacji dodawania punktów
Tab. 1. Cost of points addition

Współrzędne		
afiniczne	mieszane	Rzutowe
1O + 2M + 1K	11M + 4K	15M + 5K

3. Opis implementacji

W tym paragrafie zostaną szczegółowo opisane dwie jednostki wchodzące w skład układu dodawania punktów na krzywej eliptycznej: jednostka mnożąca elementy ciała ***GF(2^n)*** oraz korzystająca z poprzedniej jednostka dodająca punkty.

Jednostka mnożąca elementy ciała *GF(2^n)*

Mnożenie elementów ciała ***GF(2^n)*** odbywa się za pomocą wspomnianej macierzy mnożenia. Pojedyncza macierz umożliwia wygenerowanie jednego bitu wektora wynikowego, przy czym kolejne bity mogą być otrzymane za pomocą tej samej macierzy poprzez cykliczne przesunięcie wektorów wejściowych. Własność ta umożliwia tworzenie rozwiązań kompromisowych pomiędzy osiąganą szybkością, a wymaganą powierzchnią układu. W zależności od potrzeb i dostępnych zasobów możliwa jest implementacja jednej bądź wielu macierzy mnożenia w jednym układzie, a co za tym idzie, obliczanie wielu bitów wyniku w jednym cyklu zegarowym. Ponieważ dla celów kryptoanalizy pożądane jest osiągnięcie możliwie największej szybkości działania układu, to zaimplementowany został układ mnożący w pełni równoległy, czyli obliczający wszystkie 89 bitów wyniku w jednym cyklu zegara.

Tab. 2. Wyniki implementacji jednostki mnożącej (układ EP2C35F672C6 - CycloneII)
Tab. 2. Implementation results of multiplication unit (device EP2C35F672C6 - CycloneII)

Wykorzystane komórki logiczne	9212
Całkowite zużycie zasobów	28%
Częstotliwość zegara	133.58 MHz
Efektywność obliczeń	133.58 mln mnożeń / sekundę

Pojedyncza macierz jest opisywana jako równanie logiczne o 178 (2*89) zmiennych wejściowych i jednym bicie wyjściowym. Równanie takie jest odwzorowywane w układzie w formie drzewa komórek logicznych, którego liczba warstw (poziomów) wynosi $[\log_4 178]+1=4$ (gdzie [.] oznacza część całkowitą). Oszacowanie takie wynika z budowy pojedynczej komórki układu FPGA, która umożliwia realizację dowolnej funkcji logicznej o 4 wejściach i jednym wyjściu. Podstawą logarytmu

jest więc liczba wejść do pojedynczej komórki logicznej. Liczba poziomów nie jest duża, co zapewnia niewielką długość ścieżki krytycznej i możliwość uzyskania stosunkowo wysokiej częstotliwości zegara.

Jednostka sumująca punkty krzywej eliptycznej

Obliczenie sumy dwóch różnych punktów na krzywej eliptycznej $y^2 + xy = x^3 + ax^2 + b$ nad ciałem ***GF (2^n)*** we współrzędnych rzutowych wymaga następującej serii działań:

Dane wejściowe:

$(X_0, Y_0, Z_0), (X_1, Y_1, Z_1)$, krzywa $y^2 + xy = x^3 + ax^2 + b$

Wynik:

$$(X_0, Y_0, Z_0) + (X_1, Y_1, Z_1) = (X_2, Y_2, Z_2)$$

$$\begin{aligned} U_0 &= X_0 \cdot Z_1^2 \\ S_0 &= Y_0 \cdot Z_1^3 \\ U_1 &= X_1 \cdot Z_0^2 \\ W &= U_0 + U_1 \\ S_1 &= Y_1 \cdot Z_0^3 \\ R &= S_0 + S_1 \\ L &= Z_0 \cdot W \\ V &= R \cdot X_1 + L \cdot Y_1 \\ Z_2 &= L \cdot Z_1 \\ T &= R + Z_2 \\ X_2 &= a \cdot Z_2^2 + T \cdot R + W^3 \\ Y_2 &= T \cdot X_2 + V \cdot L^2 \end{aligned} \qquad (2)$$

Struktura algorytmu Rho Pollarda pozwala na zastosowanie reprezentacji mieszanej bez konieczności częstego przeprowadzania konwersji. Eliminuje to część obliczeń, ponieważ trzecia współrzędna Z_1 jest stale równe 1. Po pominięciu mnożenia przez 1 do obliczenia sumy wymagane jest 11 mnożeń, 3 podniesienia do kwadratu oraz 7 dodawań w ciele ***GF(2^n)***.

Sumator punktów krzywej został wyposażony w jedną jednostkę mnożącą i jeden sumator dwuargumentowy. Taka architektura podyktowana była strukturą zależności pomiędzy kolejnymi etapami obliczeń. Zdublowanie jednostki mnożącej zwiększyłoby objętość układu prawie dwukrotnie, nie zapewniając przy tym proporcjonalnej redukcji czasu sumowania punktów. Z podobnych powodów zrezygnowano ze skracania ścieżki krytycznej poprzez implementację dodatkowych rejestrów potokowych; zwiększenie częstotliwości zegara byłoby okupione zbyt dużą liczbą cykli jałowych (*idle cycles*). Sumator dwuargumentowy wystarcza do przeprowadzenia obliczeń w krótkim czasie, nie wprowadzając jednocześnie nadmiarowej logiki koniecznej do multipleksacji zbyt dużej liczby sygnałów.

Zoptymalizowana architektura układu umożliwia obliczanie wyniku kolejnej sumy punktów co 11 cykli zegara, przy czym pierwszy punkt podawany jest na wyjście po 13 cyklach (2 cykle jałowe). Jest to minimalna możliwa liczba cykli potrzebna na obliczenie sumy punktów w przypadku sumatora punktów zawierającego jedną jednostkę mnożącą.

Układ zawiera 9 rejestrów przechowujących częściowe wyniki obliczeń. Wykonywane działania zostały uszeregowane w kolejności zapewniającej możliwie najlepsze wykorzystanie zasobów jednostki. Priorytetem była minimalizacja logiki potrzebnej na multipleksację sygnałów. Cel ten został osiągnięty poprzez odpowiednią adresację przy operacjach zapisu/odczytu w czasie trwania obliczeń, tak aby jednostki mnożąca i sumująca elementy ciała ***GF(2^n)*** odwoływały się do jak najmniejszej liczby różnych rejestrów układu. Dodatkowy wzrost szybkości działania umożliwiło skrócenie ścieżki krytycznej układu, dzięki odpowiedniemu rozmieszczeniu rejestrów roboczych (m.in. na wejściu i wyjściu jednostki mnożącej).

Tab. 3. Wyniki implementacji jednostki sumującej (układ EP2C35F672C6 - CycloneII)
Tab. 3. Implementation results of point addition unit (device EP2C35F672C6 - CycloneII)

Wykorzystane komórki logiczne	11873
Całkowite zużycie zasobów	36%
Częstotliwość zegara	137.25 MHz
Efektywność obliczeń	12.48 mln sumowań / sekundę

Długość ścieżki krytycznej całego układu jest zbliżona do długości ścieżki krytycznej jednostki mnożącej wchodzącej w jego skład, co świadczy o poprawnym rozmieszczeniu rejestrów tymczasowych.

4. Podsumowanie i plany przyszłych prac

Oczekiwana liczba kroków równoległej wersji algorytmu Rho Pollarda wyraża się wzorem:

$$\frac{\sqrt{\pi n/2}}{M}, \qquad (3)$$

gdzie n jest rzędem punktu, a M liczbą jednostek uczestniczących w obliczeniach.

A zatem biorąc pod uwagę efektywność działania opisanej jednostki dodającej punkty na krzywej eliptycznej oraz fakt, iż to właśnie dodawanie jest najbardziej kosztowną operacją w każdym kroku algorytmu Rho Pollarda, można oszacować czas potrzebny na złamanie krzywej ECC2-89 na:

$$\sqrt{\pi 2^{88}/2} \,/\, 12.48 \cdot 10^6 \approx 20{,}5 \; dni.$$

Według specjalistów z firmy Certicom efektywność dodawania punktów wspomnianej krzywej eliptycznej, korzystając z implementacji programowej, wykonywanej na stacji roboczej Digital Alpha z procesorem taktowanym zegarem 500 MHz, wynosi około 187 tysięcy sumowań na sekundę ([6]). A zatem, w stosunku do stacji Alpha, opisywany w tym artykule układ jest ponad 66 razy szybszy.

Autorzy artykułu nie zamierzają poprzestać na samym oszacowaniu możliwości złamania opisywanej krzywej. Będą kontynuować prace nad rozbudową jednostki, poprzez dodanie do niej pełnej funkcjonalności algorytmu Rho Pollarda oraz przebadanie możliwości zwielokrotnienia jednostki w jednym układzie FPGA. Docelowo opisywana jednostka ma stać się częścią większego, rozproszonego systemu kryptoanalitycznego, składającej się z wielu struktur reprogramowalnych oraz klastra komputerowego.

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

5. 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] IEEE P1363. Standard Specyfications for Public Key Cryptography. Draft 13. 1999
[4] Menezes Alfred , Hankerson Darrel, Vanstone Scott. Guide to elliptic curve cryptography. Springer 2004
[5] Menezes Alfred, van Oorschot P.C., Vanstone Scott. Handbook of applied cryptography. CRC Press 1997
[6] Certicom. ECC Challenge. www.certicom.com/download/aid-111/cert_ecc_challenge.pdf
[7] www.rsa.com/rsalabs/node.asp?id=2093
[8] www.certicom.com/index.php?action=ecc,ecc_challange

Artykuł recenzowany