Wykład 6d
Temat: Układy cyfrowe symetrycznego szyfrowania blokowego.

6.1. Ogólna charakterystyka procesorów ochronnych

Zadanie procesorów ochronnych polega na szyfrowaniu, szybkim realizowaniu wielu złożonych algorytmów kryptograficznych, kompresji oraz uwierzytelnianiu danych. Pozwala to osiągnąć większy stopień bezpieczeństwa w systemach teleinformatycznych i umożliwia większą przepływność danych.

Ochrona przed nieautoryzowanym dostępem jest bardzo ważnym w przypadku współczesnych elementów sieci komputerowych, takich jak zapory sieciowe, jednostki VPN (Virtual Private Network), przełączniki sieciowe, serwery czy bazy danych. Wzrost zapotrzebowania na coraz bardziej zaawansowane techniki kryptograficzne jest związany z dynamiką rozwoju usług handlu elektronicznego, tzw. e-commerce. Firmy coraz częściej korzystają z usług wirtualnych sieci prywatnych rezygnując z drogich łączy dzierżawionych. W związku z udoskonalaniem metod podsłuchu elektronicznego użytkownicy dokonują szyfrowania większości swych danych. Według wstępnych analiz połowa transmisji w Internecie będzie zaszyfrowana przez w najbliższych latach.

W celu spełnienia oczekiwań użytkowników, firmy produkujące sprzęt sieciowy wdrażają standardy bezpieczeństwa bezpośrednio do swoich linii produkcyjnych, natomiast rzadziej korzystają z oddzielnych modułów VPN, implementujących te standardy. W wielu rozwiązaniach sprzętowych często się zdarza, że wymagania odnośnie szybkości transmisji danych znacznie przekraczają możliwości przetwarzania algorytmów kryptograficznych przez procesory ogólnego przeznaczenia. Dlatego producenci wprowadzają na rynek nowy rodzaj specjalizowanych układów pod nazwą „security processors”, czyli procesory ochronne.

Obszar zastosowań procesorów ochronnych obejmuje:

- akceleratory IPSec (Security Protocol) dla sprzętu VPN, czyli użycie pakietów tunelowych w trzeciej warstwie sieciowej;
- akceleratory warstwy transportowej SSL (Secure Socket Layer) dla przełączników i serwerów sieciowych.

Procesory ochronne odciążają znacznie rutery oraz serwery. Wymagania obliczeniowe są różne dla obu powyższych rodzajów akceleratorów, dlatego występują spore różnice w ich budowie.

Rynek procesorów ochronnych rozwija się niezwykle dynamicznie, jego wartość w 2001 r. była oceniana na ok. 80 mln dolarów. Prawie cały przychód w 2001 r. pochodził z czterech firm (Broad-com, Hifn, Motorola oraz SafeNet). Obecnie na rynku procesorów ochronnych pojawiły się nowe firmy, jak np. Cavium, Corrent, Layer N czy NetOctave, jednak pozycja dotychczasowych liderów wydaje się być niezagrożona. Analitycy uważają że wartość rynku przekroczyła 400 mln dolarów w 2005 r. przy średnim 50% rocznym wzroście przychodów wszystkich firm.

Według analityków firma Cavium za kilka lat będzie zdecydowanym liderem na rynku procesorów ochronnych. Pierwsze procesory tej firmy zostały wprowadzone na rynek w marcu 2002 r. W pierwszych dniach marca 2003 r. firma zaprezentowała nową linię procesorów Nitrox II, które umożliwiają jednoczesne przetwarzanie protokołów IPSec oraz SSL z przepływnością 2-10 Gbit/s. W ciągu niespełna dwóch lat szybkość przetwarzania IPSec w procesorach ochronnych wzrosła o ponad 900%, a kamieniem milowym stał się procesor Nitrox firmy Cavium, który jako pierwszy przekroczył granice 4 Gbit/s, Prędkość realizacji protokołu SSL wzrosła 80 razy w ciągu minionych 18 miesięcy.

Początek 2003 r. przyniósł przełamanie nowej bariery, tzn. procesor Nitrox II umożliwia jednoczesne przetwarzanie obydwu protokołów z szybkością dochodzącą do 10 Gbit/s i zapewnia kompleksowe zabezpieczenie urządzeń sieciowych.

Zwiększa się zapotrzebowanie na nowe układy, których zadaniem jest szyfrowanie, kompresja oraz uwierzytelnianie danych w celu lepszego zabezpieczenia sieci komputerowych. Takimi układami są właśnie procesory ochronne firmy Cavium, Hifn i Motorola, które dzięki inteligentnemu przetwarzaniu pakietów podnoszą wydajność całego systemu oraz zapewniają odciążenie innych rodzajów procesorów, w tym procesorów sieciowych od zadań związanych z szyfrowaniem danych.

W sieciach VPN oraz Gigabit Ethernet dzięki zastosowaniu procesorów ochronnych nastąpiło znaczne zwiększenie przepływności danych. Procesory sieciowe przy przetwarzaniu protokołów IP operują przede wszystkim na nagłówkach pakietów, podczas gdy szyfrowanie iPSec oraz tworzenie skrótów wiadomości (message digest) wymaga działania na użytecznej zawartości informacyjnej pakietów (payload). Tym samym nadmierne obciążenie procesorów sieciowych prowadzi do znacznego spadku przepływności danych szczególnie w sieciach VPN. Początkowo w celu przyspieszenia operacji szyfrowania danych stosowano tzw. akceleratory klucza publicznego (PK accelerators), dzięki którym można było wykonać o 200-300 operacji na sekundę więcej na strukturze danych klucza. Jednak dopiero wprowadzenie procesorów ochronnych przyniosło możliwość kompresji, szyfrowania oraz uwierzytelniania danych w jednym cyklu obliczeniowym - właściwość szczególnie pożądana np. w ruterach łączących sieci lokalne oraz rozlegle.

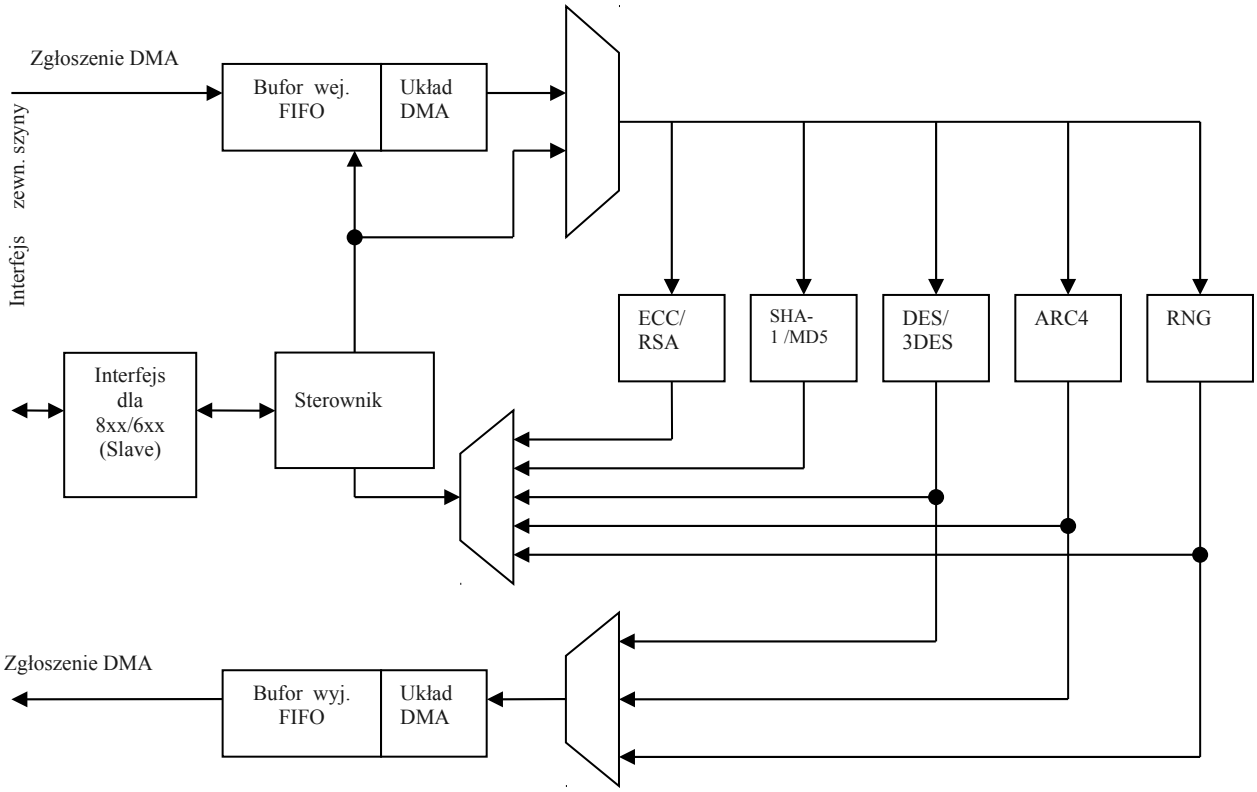
6.2. Budowa procesora ochronnego na przykładzie układu MPC180 firmy Motorola

Procesor MPC-180 bazuje na procesorach rodziny Motoroli MPC8xx lub procesorach komunikacyjnych MPC826x rodziny POWERQUICC. MPC180 może obsługiwać skomplikowane aplikacje kryptograficzne i służy przede wszystkim do odciążenia procesora głównego. MPC180 jest zaprojektowany, by wspierać wszystkie algorytmy łącznie z IPSEC, IKE i SSL / TLS.

Firma Motorola ma ponad 30-letnie tradycje w dziedzinie technologii kryptograficznych, a swe produkty sprzedaje bardzo wymagającym klientom, np. instytucjom rządowym. Przełomem okazał się rok 2000, kiedy Motorola wprowadziła na rynek amerykański rodzinę security processors S1. Należą do niej procesory MPC180, MPC190, MPC 184 oraz MPC185. Najnowsze rozwiązania kryptograficzne zastosowane w procesorach ochronnych Motoroli umożliwiają istotne zwiększenie bezpieczeństwa sprzętu sieciowego.

Rodzina S1 jest przeznaczona przede wszystkim dla sprzętu CPE (**Customer Premise Equipment** – infrastruktura i sieciowy sprzęt konsumencki), dostępu szerokopasmowego (HDSL - czyli High Data Rate DSL, inaczej DSL - Digital Subscriber Line - cyfrowa linia abonencka) oraz skraju sieci szkieletowej (edge).

Procesory ochronne tej rodziny (S1) zostały zaprojektowane tak, aby łatwa była ich współpraca z tradycyjnymi procesorami Motoroli o architekturze PowerQUICC oraz PowerPC.



Rys. 6.1. Schemat blokowy procesora ochronnego MPC180

Tabela 6.1

Wyjaśnienie do rysunku 6.1	
	- Multiplekser. - Demultiplekser
1. RSA	Blok szyfrujący z algorytmem szyfrującym RSA
2.SHA-1/MD5	Blok szyfrujący z algorytmem haszującym SHA-1 lub MD5
3. DES / 3DES	Blok szyfrujący z algorytmem DES lub 3DES
4. ARC4	Blok szyfrujący z algorytmem ARC4
5. RNG	Generator liczb pseudolosowych
5. Zgłoszenie DMA	DMA (Direct Memory Access) – zgłoszenie danych z pamięci
6. Sterownik	Układ sterujący pracą procesora
7. Interfejs dla 8xx/6xx (Slave)	Interfejs (Glueless) służący do bezpośredniego podłączenia procesorów rodziny PowerPC i PowerQUICC do MPC180
8. Bufor FIFO/ Układ DMA	Bufor pamięci z kolejkowaniem FIFO (First In, First Out)

Algorytm SHA-1 (Secure Hash Algorithm)

Algorytm SHA-1 jest wymaganą funkcją haszującą (funkcja rozpraszająca, funkcja zwężająca, funkcja mieszająca) w algorytmie Digital Signature Algorithm (DSA) będącym składnikiem standardu Digital Signature Standard (DSS). Algorytm SHA-1 jest używany do obliczania skrótu (*digest*) dla dowolnej wiadomości lub pliku danych dostarczonego na wejściu.

Operacja Message padding - dopełnienie wiadomości

Celem operacji "Message padding" jest uczynienie aby wiadomość do haszowania miała długość będącą wielokrotnością 512. SHA-1 sekwencyjnie procesuje bloki 512 bitów podczas obliczania skrótu dla wiadomości.

Uzupełnianie polega na dopisaniu na końcu wiadomości:

1. Bitu "1"
2. m bitów "0"
3. 64 bitowej liczby mówiącej o długości wiadomości

Dopełniona wiadomość będzie zawierać 16 *n słów 32 bitowych (n>0). A zatem możemy wiadomość uważać za sekwencję n bloków M_1 , M_2, ... , M_n, gdzie każdy M_i zawiera 16 słów (32 bity).

SHA-1 jest to jednokierunkowy algorytm mieszający . Jest to również swoistego rodzaju wzór w wyniku którego z treści wiadomości lub dokumentu otrzymujemy tak zwany skrót o długości 160 bitów lub jak kto woli liczbę o maksymalnej wartości 2 do potęgi 160. Ten skrót jest reprezentacja treści np. dokumentu. Niezależnie od rodzaju sytemu operacyjnego, miejsca obliczania, wynik dla dokumentu o tej samej treści będzie zawsze taki sam.

Uwagi dot. rys. 6.1(2)

Algorytm MD4/MD5 (*Message-Digest Algorithm*)

MD4 (*Message Digest* - skrót wiadomości) jest jednokierunkową funkcją skrótu zaprojektowaną przez Rona Rivesta. Algorytm ten, dla danej wiadomości, wytwarza skrót wiadomości o długości 128 bitów. Rivest podał swoje następujące cele przy projektowaniu tego algorytmu:

- Bezpieczeństwo. Powinno być obliczeniowo niewykonalne znalezienie dwóch wiadomości, które po skróceniu dają tę samą wartość. Żaden atak nie powinien być bardziej efektywny niż, atak brutalny.
- Bezpośrednie bezpieczeństwo. Bezpieczeństwo algorytmu MD4 nie powinno zależeć od żadnego założenia podobnego do trudności faktoryzacji liczb.
- Szybkość. Algorytm MD4 powinien być przystosowany do bardzo szybkich implementacji programowych. Powinien bazować na zbiorze prostych operacji bitowych na 32-bitowych argumentach.
- Prostota i zwartość. Algorytm MD4 powinien być tak prosty, jak tylko jest to możliwe, bez dużych struktur danych lub skomplikowanego programu.
- Zalecana architektura *little-endian*. Algorytm MD4 powinien być zoptymalizowany pod kątem architektur mikroprocesorów (w szczególności mikroprocesorów firmy Intel); większe i szybsze komputery dokonają niezbędnych translacji.

Po tym jak algorytm został po raz pierwszy zaprezentowany, Bert den Boer i Antoon Bosselaers przeprowadzili skuteczną kryptoanalizę dwóch z trzech cykli tego algorytmu. Ralph Merkle skutecznie zaatakował pierwsze dwa cykle. Eli Biham przeprowadził dyskusję możliwych ataków, za pomocą kryptoanalizy różnicowej, przeciwko dwóm z trzech cykli MD4. Pomimo że ataki te nie mogły być rozszerzone na cały algorytm, Rivest wzmocnił swój algorytm. Wynikiem tego był algorytm MD5.

Algorytm haszujący **MD5** został opublikowany przez R.Rivesta bez podania argumentów, uzasadniających matematycznie, że może on pełnić funkcję dobrej jednokierunkowej funkcji skrótu. Wejściem algorytmu jest komunikat *M o* praktycznie dowolnej długości, a wyjściem 128-bitowy skrót, czyli wyciąg tego komunikatu. Algorytm MD5 jest następujący:

1. Doklejamy do haszowanego ciągu bit 1.
2. Doklejamy tyle zer ile trzeba żeby ciąg składał się z 512-bitowych bloków, i ostatniego niepełnego - 448-bitowego.
3. Doklejamy 64-bitowy (zaczynając od najmniej znaczącego bitu) licznik oznaczający rozmiar wiadomości. W ten sposób otrzymujemy wiadomość złożoną z 512-bitowych fragmentów.
4. Ustawiamy stan początkowy na **0123456789abcdeffedcba9876543210**.
5. Uruchamiamy na każdym bloku (jest przynajmniej jeden blok nawet dla pustego wejścia) funkcję zmieniającą stan.
6. Po przetworzeniu ostatniego bloku zwracamy stan jako wynik funkcji haszującej.

Funkcja zmiany stanu ma 4 rundy (64 kroki). Stan jest traktowany jako 4 liczby 32-bitowe, i w każdym kroku do którejś z tych liczb dodawany jest jeden z 16 32-bitowych fragmentów bloku wejściowego, pewna stała zależna od numeru kroku oraz pewna prosta funkcja boolowska 3 pozostałych liczb. Następnie liczba ta jest przesuwana cyklicznie o liczbę bitów zależną od kroku, oraz jest dodawana do niej jedna z pozostałych liczb.

Funkcje te to:

- w krokach 1 do 16 (runda 1) funkcja F(x,y,z) = (x and y) or (neg x and z) (jeśli *x* to *y*, w przeciwnym wypadku *z*);
- w krokach 17 do 32 (runda 2) funkcja G(x,y,z) = (x and z) or (y and neg z) (jeśli *z* to *x*, w przeciwnym wypadku *y*);
- w krokach 33 do 48 (runda 3) funkcja H(x,y,z) = (x xor y xor z) (suma argumentów modulo 2, lub innymi słowy: czy występuje nieparzysta liczba jedynek w argumentach);
- w krokach 49 do 64 (runda 4) funkcja I(x,y,z) = (y xor (x or neg z)) (jeżeli (z=1 i x=0) wtedy y, w przeciwnym wypadku nie y).

Podobną budowę mają funkcje haszujące MD4, SHA0 i SHA1 – różnią się one jedynie postacią funkcji zmieniającej stan, oraz rozmiarem stanu (160 bitów, czyli 5 32-bitowych rejestrów w SHA i SHA1, wobec 128 w MD4 i MD5).

128 bitów jest uważane za zbyt mało, żeby zabezpieczyć przed kolizjami, dlatego do większości zastosowań lepiej jest używać funkcji zwracającej co najmniej 160 bitów.

Algorytm RC4 / Alleged RC4 (ARC4)

RC4 został zaprojektowany przez Rona Rivesta dla RSA Data Security. Jest to algorytm strumieniowy o zmiennym rozmiarze klucza. Pracuje w trybie OFB – ciąg klucza jest niezależny od tekstu jawnego. Algorytm używa osiem grup po osiem bloków typu S-blok: S_0, S_1, ..., S_{255}. Wartościami wejściowymi są permutacje liczb od 0 do 255, a permutacje te są funkcją klucza o zmiennej długości. W algorytmie używa się dwóch liczników: i oraz j, początkowo zapełnionych zerami.

Proces tworzenia losowego bajta można przedstawić następująco:

1. $i = (i + 1) \bmod 256$
2. $j = (j + S_j) \bmod 256$
3. *zamiana miejscami* S_i *i* S_j
4. $t = (S_i + S_j) \bmod 256$
5. $K = S_t$

Wynikowy bajt K jest sumowany modulo 2 albo z tekstem jawnym w procesie szyfrowania, albo z szyfrogramem w procesie deszyfrowania. Szyfrowanie algorytmem RC4 okazuje się dziesięciokrotnie szybsze niż przy użyciu algorytmu DES.

Proces wypełniania wartościami początkowymi S-bloków przebiega w prosty sposób. Na początku zapełnia się je kolejnymi liczbami całkowitymi począwszy od $S_0 = 0$ aż do $S_{255} = 255$. W następnym kroku zapełnia się inną tablicę 256-bajtową, używając klucza i powtarzając go tyle razy, ile jest to potrzebne do zapełnienia całej tablicy: K_0, K_1, ..., K_{255}. Indeks j ustawia się na 0 i wówczas:

For i = 0 to 255
$j = (j + S_i + K_j) \bmod 256$
zamiana miejscami S_i i S_j

Algorytm RC4 jest odporny na wszystkie znane ataki i jest stosowany w wielu komercyjnych programach. Algorytm ten stosują firmy Apple i Oracle Secure SQL w swych produktach, jest także częścią specyfikacji protokołu sieci komórkowych (Cellular Digital Packet Data).

Generator liczb pseudolosowych (RNG)

Generator liczb pseudolosowych (*Pseudo-Random Number Generator*, lub **PRNG(RNG)**) to program, który na podstawie niewielkiej ilości informacji (tzw. *seed*) generuje deterministycznie potencjalnie nieskończony ciąg bitów, który pod pewnymi względami jest nieodróżnialny od ciągu uzyskanego z prawdziwie losowego źródła. Generatory liczb pseudolosowych nie generują całkiem losowych ciągów – jeśli generator jako seed bierze *k* bitów informacji, to może wygenerować *n*-bitowy ciąg jedynie na 2^k sposobów spośród 2^n możliwych. Do bardzo wielu zastosowań taka pseudo-losowość jest zupełnie wystarczająca – w grach komputerowych, obliczeniach probablistycznych (takich jak np. całkowanie Monte Carlo) potrzebujemy jedynie liczb zachowujących się mniej więcej jak liczby losowe.

Zastosowanie w kryptografii

Zupełnie inna sytuacja ma miejsce w kryptografii – tutaj potrzeba bardzo silnych właściwości bezpieczeństwa. Generatory liczb pseudolosowych są używane przede wszystkim jako szyfry strumieniowe. Tajnym kluczem jest *seed*. Na jego podstawie nadawca generuje ciąg bitów, i XOR-uje te bity z bitami wiadomości. Odbiorca generuje ten sam ciąg bitów pseudolosowych i XOR-uje go z zaszyfrowaną wiadomością, otrzymując wiadomość oryginalną. Ważne właściwości, których oczekiwalibyśmy od generatora to:

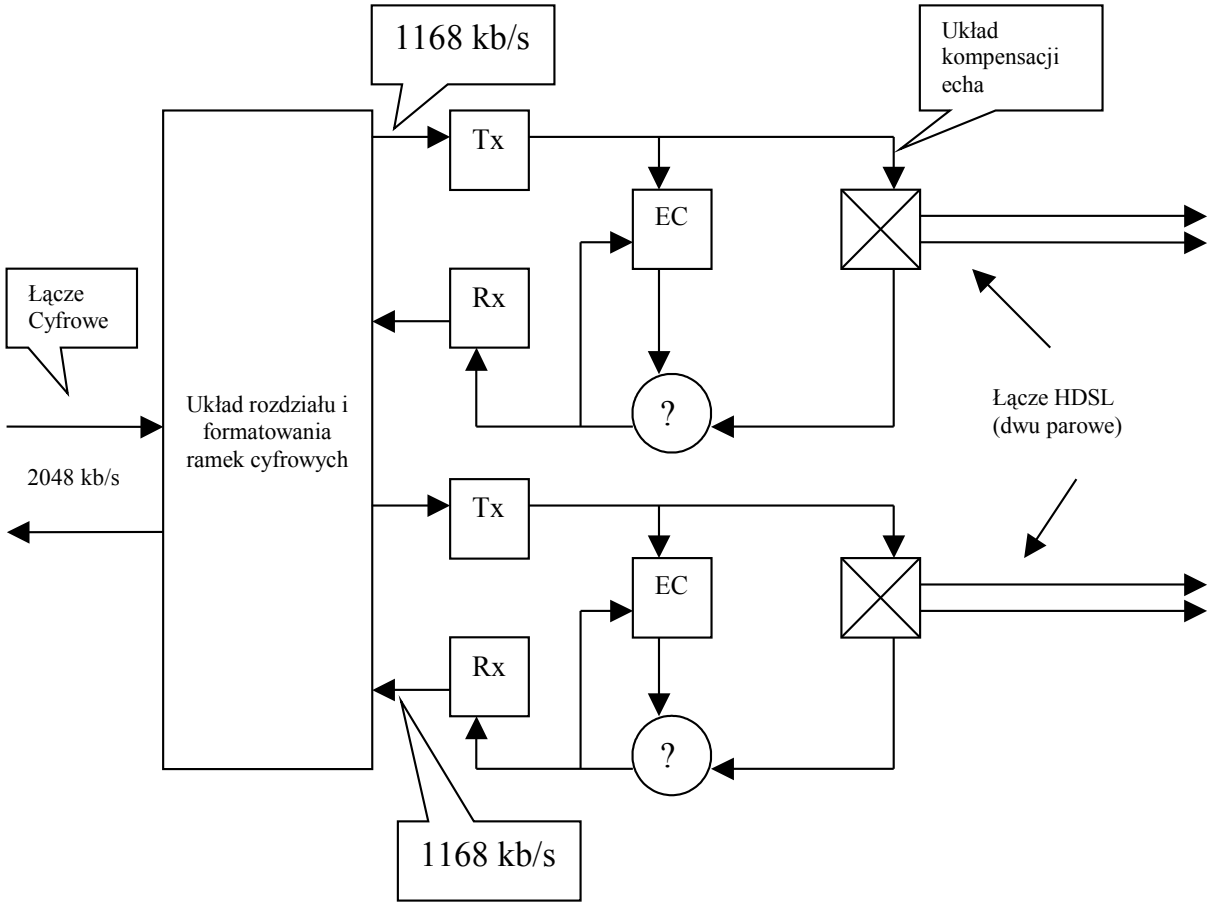
- znając ciąg wygenerowanych bitów nie da się w rozsądnej ilości obliczeń odzyskać seeda,
- znając ciąg wygenerowanych bitów nie da się w rozsądnej ilości obliczeń przewidzieć kolejnych z prawdopodobieństwem istotnie różnym od 1/2(ani zgadując seeda ani w żaden inny sposób).

Technologia HDSL (transmisja szerokopasmowa)

Stosowana coraz częściej w telekomunikacji technologia HDSL umożliwia uzyskanie przepływności 2 Mb/s za pomocą zwykłej dwuprzewodowej linii telefonicznej. Dedykowany odcinek symetrycznej linii telefonicznej może być wykorzystany jako szerokopasmowy trakt cyfrowy 2 Mb/s bądź traktowany jako medium transmisyjne do jednoczesnego przekazu 30 zwykłych rozmów telefonicznych za pomocą jednej pary przewodów miedzianych. Technologia cyfrowego łącza abonenckiego o dużej przepływności HDSL (*High bit rate Digital Subscriber Line)* umożliwia przesyłanie danych linią dedykowaną (bez komutacji) z szybkością 2 Mb/s (2048 kb/s) lub udostępnienie 30 kanałów telefonicznych, każdy o przepływności 64 kb/s, początkowo za pomocą trzech, następnie dwóch, a ostatnio już tylko jednej pary skręconych przewodów miedzianych. Zwykły kabel telefoniczny do tej pory stosowany do przyłączenia jednego lub dwóch pojedynczych abonentów telefonicznych lub połączenia lokalnej centralki abonenckiej PABX *(Private Automated Branch Exchange)* z centralą miejską może być teraz wykorzystywany w technologii HDSL na dystansie od kilku do kilkunastu kilometrów, bez konieczności używania wzmacniaczy pośrednich (regeneratorów sygnału).

Minimalna konfiguracja systemu transmisji w technologii HDSL obejmuje dwa identyczne pod względem funkcji urządzenia, z których jedno jest instalowane po stronie użytkownika, a drugie u operatora sieci.
Rozwiązania konstrukcyjne obydwu urządzeń są zwykle odmienne:
centralowe - obsługujące zwykle wielu użytkowników od strony systemu komutacji i zdalne - dla niewielkiej grupy lub pojedynczego abonenta.

Zasady transmisji - Pierwsze instalacje urządzeń wykonanych w technologii HDSL wymagały jeszcze trzech par linii symetrycznych niezbędnych do transmisji sygnałów z pełną przepływnością 2 Mb/s, jednak największą popularność uzyskały systemy działające na dwóch parach linii telefonicznej. Niezależnie od tego, ile par przewodów jest wykorzystanych do transmisji informacji o przepływności 2 Mb/s - co związane jest z wdrażaniem coraz nowszych rozwiązań technicznych - zasada działania łącza w technologii HDSL jest podobna (rys. 6.2).



Rys. 6.2 Zasada działania łącza HDSL (dwie pary)

W systemie opartym na dwóch symetrycznych liniach strumień informacji cyfrowej o przepływności 2,048 Mb/s jest dzielony dla każdego z kierunków na dwa strumienie - zawierające po 1024 kb/s informacji użytkownika - przesyłane równolegle i równocześnie w obu kierunkach przy użyciu dwóch par przewodów. Zastosowana po obydwu stronach łącza technika kompensacji echa umożliwia prowadzenie w pełni dupleksową transmisję cyfrową dla każdej z par oddzielnie. W układzie formatowania dla każdej pary przewodów jest tworzona własna ramka, zawierająca oprócz transmitowanej informacji użytkownika również dodatkową przepływność sygnalizacyjną (128 kb/s lub 144 kb/s), pozwalającą na monitorowanie transmisji w trakcie normalnej pracy oraz utworzenie dodatkowego kanału do utrzymania i lokalizacji uszkodzeń. W związku z tym łączna przepływność bitowa pojedynczej linii symetrycznej wynosi 1024+144=1168 kb/s, z możliwością wykorzystania tego kanału do szybkości 1152 kb/s (sygnalizacja 128 kb/s). W trybie pracy z ramkowaniem (G.704) sygnały są przesyłane na każdej z dwóch par przewodów w 15 kanałach, każdy po 64 kb/s danych, jak również szczeliny czasowe o numerach 0 i 16 (nadmiarowa) oraz tworzy się kanał sterujący EQC o pojemności 16 kb/s. Niezależnie od pracy z ramkowaniem możliwy jest tryb pracy bez ramkowania (G.703). W starszych modelach urządzeń HDSL, transmitujących dane przez trzy linie symetryczne, łączna przepływność bitowa każdego toru transmisyjnego jest (lub była) odpowiednio niższa i wynosi 784 kb/s.

- **Sieć szkieletowa**

Do szkieletu sieci zalicza się wszystkie urządzenia których funkcjonowanie może mieć wpływ na łączność danej sieci LAN ze światem. Umownie oznacza to ostatnie urządzenie, które posiada interfejs o adresie IP nie należącym do sieci lokalnej. Odpowiednio do sieci lokalnej zalicza się wszystkie urządzenia danej instytucji które nie zawierają adresów IP innych niż właściwe dla tej jednostki. Do szkieletu sieci zalicza się również urządzenia ethernetowe z interfejsami (*uplinkami*) ATM.

- **PowerPC**

Handlowa nazwa mikroprocesora produkowanego przez firmy Motorola, IBM i Apple. Stosowany w kilku generacjach komputerów Apple Macintosh, komputerach Pegasos, AmigaOne (uprzednio w kartach procesorowych dla Amigi - modele 603 oraz 604) a także w wielu innych urządzeniach różnych firm, np. w drukarkach laserowych. Pierwszy model tego procesora został wyprodukowany w 1994 roku pod numerem 601, stopniowo powstawały nowe: 602, 603, 604, potem PowerPC G3 - procesor o symbolu 750 pracujący początkowo z częstotliwością 350 MHz, który był przełomem na rynku komputerów Macintosh. Jego następcą był G4, z taktowanym zegarem od 1 GHz do 1,5 GHz. Obecnie najnowszym procesorem z linii PPC jest 64-bitowy PowerPC G5. PowerPC jest procesorem RISC (*Reduced Instruction Set Processor*). Jego podstawowe cechy:

1. Zredukowana liczba rozkazów, ich liczba wynosi kilkadziesiąt, podczas gdy w procesorach CISC (*Complex Instruction Set Processor*) sięga setek. Upraszcza to znacznie dekoder rozkazów.
2. Redukcja trybów adresowania, dzięki czemu kody rozkazów są prostsze, bardziej zunifikowane, co dodatkowo upraszcza wspomniany wcześniej dekoder rozkazów. Ponadto wprowadzono tryb adresowania, który ogranicza ilość przesłań, większość operacji wykonuje się wg schematu:

$$rejestr_C = rejestr_A \quad operacja \quad rejestr_B .$$

3. Ograniczenie komunikacji pomiędzy pamięcią, a procesorem. Przede wszystkim do przesyłania danych pomiędzy pamięcią, a rejestrami służą dedykowane instrukcje, które zwykle nazywają się **load** (załaduj z pamięci), oraz **store** (zapisz do pamięci); pozostałe instrukcje mogą operować wyłącznie na rejestrach. Schemat działania na liczbach znajdujących się w pamięci jest następujący: załaduj daną z pamięci do rejestru, na zawartości rejestru wykonaj działanie, przepisz wynik z rejestru do pamięci.
4. Zwiększenie liczby rejestrów (np. 32, 192, 256, podczas gdy np. w architekturze x86 jest zaledwie 8 rejestrów), co również ma wpływ na zmniejszenie liczby odwołań do pamięci.
5. Wszystkie rozkazy wykonują się w jednym cyklu maszynowym, co pozwala na znaczne uproszczenie bloku wykonawczego, a także na zrównoleglenie wykonywania rozkazów poprzez **przetwarzanie potokowe** (ang. *pipelining*). Czas reakcji na przerwania jest także krótszy.

- **PowerQUICC**

Oznacza zintegrowaną rodzinę procesorów komunikacyjnych z wbudowanym rdzeniem PowerPC. Stanowią one nową generacje procesorów stanowiących poważne zagrożenie dla procesorów rodziny Intela. Podstawowa ich cecha to zastosowanie architektury RISC.

Rodzina S1 wykorzystuje tzw. szyfrowanie skojarzone. Polega ono na tym, że główny procesor analizuje najpierw przesyłany pakiet. Jeżeli pakiet wymaga zaszyfrowania oraz uwierzytelnienia, jest on przekazywany do procesora ochronnego. Stosowany rodzaj szyfrowania w procesorach rodziny S1 może wkrótce nie spełnić wymagań rynku, ponieważ aplikacje kryptograficzne wymagają zapewnienia coraz większych szybkości transmisji danych, OC-12[1] lub wyższej.

Obecnie trwają prace nad procesorami ochronnymi rodziny S2. Dąży się do konstrukcji procesora, który miałby wbudowany podsystem bezpieczeństwa dla wysoko wydajnych aplikacji. Zakłada się, iż przyszłe procesory komunikacyjne firmy Motorola będą miały wiele cech wspólnych z procesorami rodziny S2. Kolejnym etapem rozwoju procesorów ochronnych ma być rodzina S3, która jako następca S2 będzie stanowiła w pełni bezpieczną platformę. Procesory ochronne tej rodziny mają zapewniać niezawodne zabezpieczenie przed atakami, których nie zdołały powstrzymać urządzenia kontroli dostępu i osiągnąć pozycję gwaranta bezpiecznych usług w sieciach komputerowych. Procesor MPC180 może współpracować z procesorami rodziny MPC8xx lub MPC826x należącymi do grupy procesorów komunikacyjnych PowerQUICC (Procesory w technologii RISC). Procesor ochronny MPC180 realizuje generację kluczy i ich wymianę, uwierzytelnianie wiadomości oraz szyfrowanie dużej liczby danych rzędu 1 Gbit/s. Procesor ochronny MPC180 może przetwarzać algorytmy związane z protokołami transportowymi takimi jak IPSec, IKE (Internet Key Exchange), SSL oraz TLS (Transport Layer Security). Dużą zaletą procesorów rodziny S1 w tym MPC180, jest zastosowanie do szyfrowania danych algorytmów kryptograficznych korzystających z metod krzywych eliptycznych.

[1] **OC-12**
Jest to indeks z tabeli hierarchii prędkości transmisji przez interfejsy (takie jak SDH i PDH), w naszym przypadku oznacza że potrzebujemy zapewnić prędkość 622.08Mb/s.

Tabela 6.2

Tabela prędkości dla poszczególnych interfejsów

Typ **Mb/s** OC-1 51.84OC-3 155.52OC-9 466.5 OC-12 622.08OC-18 933.12OC-24 1244.16OC-36 1866.24OC-48 2488.32

IPSec (Internet Protocol Security)

IPSec jest protokołem, który tworzy architekturę przeznaczoną do bezpiecznego przesyłania przez sieć pakietów IP. Obsługa IPSec jest opcjonalna w IPv4 i obowiązkowa w IPv6. Protokół ten działa bezpośrednio ponad IP, co pozwala mu chronić wszystkie protokoły z wyższych warstw modelu TCP/IP. Zastosowanie IPSec pozwala zapobiec wielu typom ataków, w tym podsłuchiwaniu pakietów, fałszowaniu adresów IP oraz przejmowaniu połączeń. Usługi zapewniane przez IPSec to:

- poufność (ang. confidentiality) – szyfrowanie danych uniemożliwia ich podglądanie przez osoby trzecie,
- integralność (ang. integrity) – gwarancja, że dane nie zostały zmodyfikowane podczas transmisji,
- uwierzytelnianie (ang. authenticity) – kryptograficzne podpisanie danych zapewnia, że pochodzą one od nadawcy,
- ochrona przed odtwarzaniem (ang. reply protection) – przechwycony pakiet, który zostanie wysłany ponownie nie zostanie zaakceptowany,
- kontrola dostępu (ang. access control) – odmowa wynegocjowania parametrów bezpieczeństwa uniemożliwia nawiązanie połączenia.

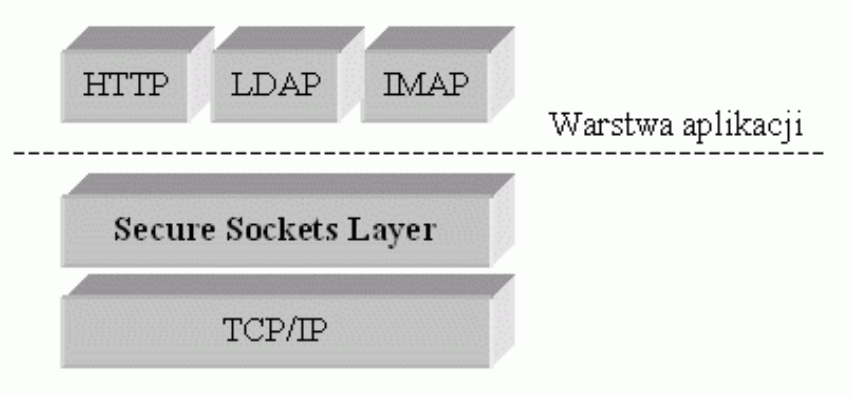
Internet Key Exchange (IKE)

Jest to protokół automatycznej wymiany klucza na potrzeby IPSec. Po ustaleniu pożądanej konfiguracji połączeń VPN w ramach firewalla, protokół IKE realizuje automatycznie proces ustalenia indywidualnych parametrów Security Associations ze zdalnym urządzeniem połączonym za pomocą VPN. Protokół IKE jest obecnie najczęściej stosowanym sposobem wymiany klucza. Opiera się on na wcześniejszych standardach ISAKMP/Oakley. Dzięki temu, Firewall Server może współpracować z szeroką gamą urządzeń zgodnych ze standardem IKE.

IKE generuje dwa typy danych Security Associations, służących do szyfrowania. Najpierw wymieniane są dane SA służące do bezpiecznego przekazania klucza. Gdy dane IKE SA są przekazane, ustalana jest sesja SA do normalnej bezpiecznej transmisji danych VPN. Te dwa etapy są określane odpowiednio jako IKE Faza-1 oraz Faza-2. Sesje wykorzystujące dane SA są krótkotrwałe i są wielokrotnie powtarzane w regularnych odstępach czasu. Dzięki temu klucze są regularnie zmieniane i dane klucze są wykorzystywane do przesyłania tylko ograniczonej ilości danych.

SSL (Secure Sockets Layer)/ TLS (Transport Layer Security)

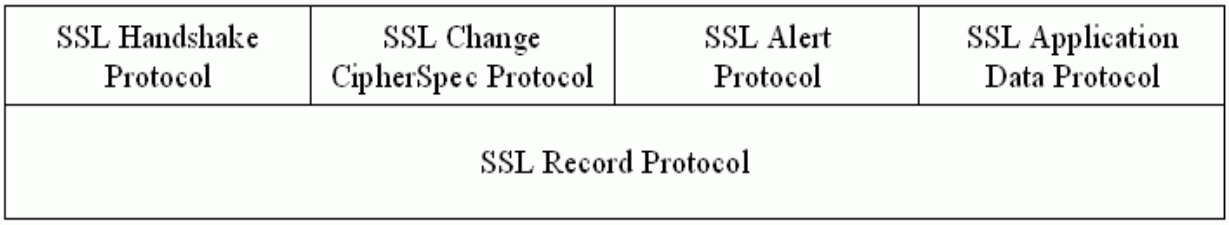
Protokół SSL został zaprojektowany przez firmę Netscape. Konkurentem dla SSL był początkowo podobny protokół nazywany S-HTTP. Przeglądarki obsługujące S-HTTP nie były jednak darmowe i to zadecydowało o zdobyciu dominującej pozycji przez SSL. Pierwotnie jego zadaniem była więc ochrona sesji HTTP, ale obecnie jest stosowany w szerszym zakresie. TLS (ang. Transport Layer Security) to standard IETF (Internet Engineering Task Force www.ietf.org) oparty na SSL. Warstwa bezpiecznych gniazd działa ponad TCP/IP chroniąc w ten sposób protokoły takie jak HTTP, czy IMAP (rys. 6.3).



Rys. 6.3. SSL zabezpiecza protokoły z warstwy aplikacji

Szyfrowanie połączeń (ang. encrypted SSL connection) pomiędzy klientem i serwerem zapewnia tajność komunikacji. Do szyfrowania danych używane są symetryczne metody kryptograficzne. Jednocześnie sprawdzana jest integralność przesyłanych danych za pomocą MAC. Do wyliczania MAC wykorzystywane są bezpieczne funkcje mieszające.

Architekturę protokołu SSL przedstawia rysunek 6.4.



Rys. 6.4 Dwie warstwy protokołu SSL

Usługi SSL realizowane są za pomocą następujących protokołów:

- SSL Handshake Protocol – uwierzytelnianie i negocjacja parametrów,
- SSL Change CipherSpec Protocol – negocjowanie parametrów dotyczących szyfrowania,
- SSL Alert Protocol – sygnalizacja o wystąpieniach błędów,
- SSL Application Data Protocol – interfejs pozwalający na dostęp do SSL Record Protocol,
- SSL Record Protocol – używany do wymiany danych warstwy aplikacji.

Protokół uwierzytelniania tekstu wiadomości MAC/HMAC – (*Key-Hash*) *Message Authentication Code*

Pojęcie z zakresu kryptografii. Posiada własności kryptograficznej funkcji skrótu (czyli ochrona integralności) z dodatkowym uwierzytelnieniem tekstu wiadomości. Jest to zrealizowane przez wprowadzenie tajnego klucza, wymaganego do obliczenia (i zweryfikowania) wartości MAC.

HMAC - *Keyed-Hash Message Authentication Code*

HMAC jest nowoczesną wersją MAC, o identycznej funkcjonalności - jego zadaniem jest weryfikacja integralności oraz autentyczności wiadomości. HMAC wykorzystuje klucz tajny znajdujący się w rękach nadawcy i odbiorcy, przy czym ten drugi zakłada że wiadomość jest autentyczna, to jest pochodzi od nadawcy, bo tylko oni dwaj znają klucz użyty do wygenerowania HMAC. Istotną różnicą w stosunku do MAC (który wykorzystywał DES), jest wykorzystanie jednokierunkowych funkcji skrótu przez HMAC. Podstawą HMAC może być dowolna funkcja skrótu, na przykład MD5 lub SHA1, do której w odpowiedni sposób "domieszany" zostaje tajny klucz. Standardowo odbywa się to w następujący sposób:

1) na końcu bloku jawnego dołączamy tajny klucz i obliczamy skrót z całości,
2) do wynikowego skrótu znów dołączamy klucz i obliczamy skrót z całości,
3) wynik jest kodem HMAC dla danego bloku.

W powyższym opisie dla uproszczenia zostały pominięte dodatkowe dwie operacje - w pierwszym kroku każdy bajt klucza jest xorowany z bajtem 0x36 (tzw. *inner pad*), w drugim kroku klucz jest xorowany z bajtem 0x5c (tzw. *outer pad*). Jedną z zalet HMAC w stosunku do MAC jest brak ograniczenia długości bloku tekstu poddawanego ochronie - większość stosowanych obecnie funkcji skrótu może w praktycznych implementacjach sumować tekst jawny o dowolnej długości.

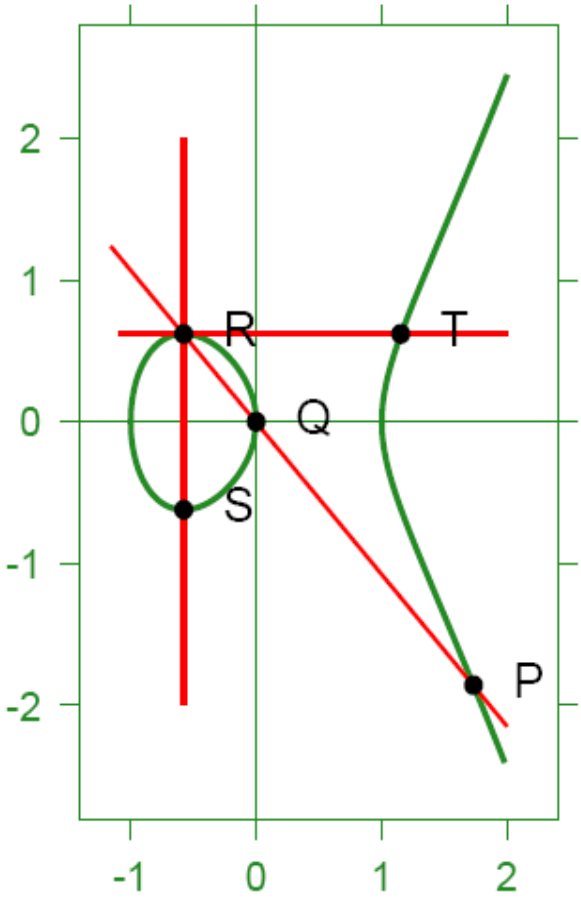
Oficjalnie protokół **TLS** (ang. *Transport Layer Security*) powstał na podstawie SSL, SSH i PTC (ang. Private Communication Technology). Po opublikowaniu specyfikacji TLS 1.0 okazało się jednak, że protokół ten jest bardzo podobny do SSL 3.0. Wprowadzono niewielkie zmiany, ale sposób działania TLS i SSL jest taki sam.

Krzywe eliptyczne

Jak dobrze wiadomo, nowoczesne metody szyfrowania skupiają się wokół **arytmetyki modularnej.** W połowie lat osiemdziesiątych różni badacze stwierdzili, że innym źródłem trudnych problemów mogą być **krzywe eliptyczne.** Krzywe eliptyczne są matematycznymi strukturami, które okazały się bardzo użyteczne w wielu dziedzinach, w tym w testowaniu pierwszości liczb (czyli w jakim stopniu liczba jest „pierwsza"), a także w znajdowaniu dzielników liczb całkowitych. Jedną z potencjalnych możliwości zastosowania krzywych eliptycznych jest tworzenie systemów z kluczem publicznym, podobnych do istniejących systemów szyfrowania.

Krzywa eliptyczna to krzywa o równaniu ogólnie rzecz biorąc:

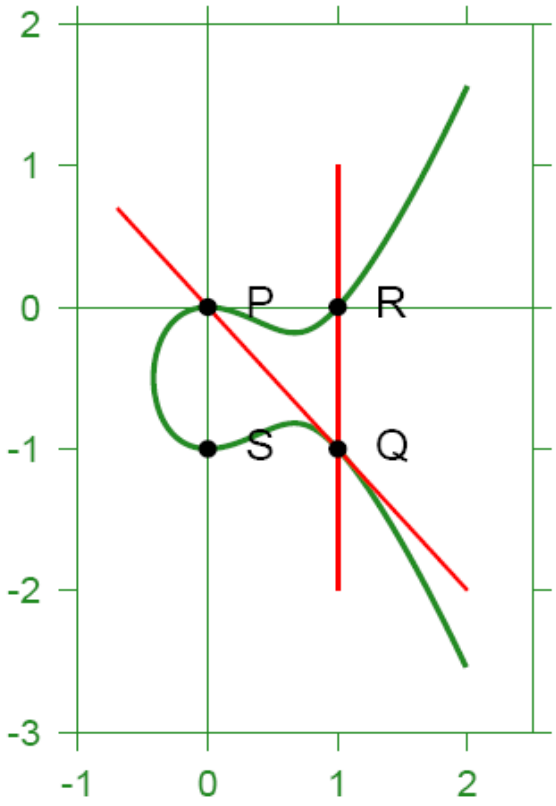
$$y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6$$



Rys. 6.5. Krzywa eliptyczna o równaniu $y^2 = x^3 - x$

W pierwszym przybliżeniu możemy myśleć, że chodzi o liczby rzeczywiste. Wówczas można narysować wykres tej krzywej. Na rys. 6.5 i 6.6 przedstawiamy dwie krzywe, o równaniu $y^2=x^3-x$ oraz $y^2+y=x^3-x^2$. Krzywa eliptyczna w pobliżu nieskończoności, tzn. dla dużych x, jest zbliżona do krzywej o równaniu $y=x^{3/2}$, w szczególności przecina każdą prostą, która nie jest pionowa. Do krzywej zaliczymy też punkt urojony w nieskończoności, będziemy go oznaczać przez O. Zakładamy, że punkt ten należy do wszystkich prostych pionowych. Widoczne jest teraz, np. patrząc na wykres, że wiele prostych przecina krzywą dokładnie w trzech punktach. Jeśli prosta jest styczna do krzywej, oś y-ów na pierwszym czy oś x-ów na drugim rysunku, to widoczne są tylko dwa punkty przecięcia (punkcie urojony należy do osi y-ów), ale punkt styczności będziemy wyznaczać podwójnie, więc też są trzy.

Punkty leżące na krzywej eliptycznej będziemy do siebie "dodawać". Uznajemy punkt w nieskończoności O za element neutralny dodawania, czyli zero, a dla dowolnych trzech punktów krzywej eliptycznej P, Q, R, leżących na jednej prostej postulujemy, że $P+Q+R=O$. Jeśli prosta jest pionowa, to jednym z punktów jest zero i otrzymujemy $R+S+O=O$. Punkty R i S są punktami przeciwnymi, $R = -S$ i $S = -R$. W ogólnym przypadku, jeśli $P+Q+R=O$, to $P+Q = -R$ i wiemy już gdzie szukać wyniku dodawania dwóch punktów. Z pierwszym rysunku widzimy, że $P+Q=S=R+T$. W tej drugiej równości wykorzystujemy fakt, że prosta pozioma jest styczna i punkt R liczy się podwójnie. Inną styczną jest oś y-ów, widoczne jest więc, że $Q+Q=O$. Możliwe teraz jest sprawdzenie, że punkt O jest elementem neutralnym dodawania. $P+O+(-P)=O$, na rysunku jest to prosta pionowa przechodząca przez punkty P i $-P$, więc $P+O=P$. Nie ma problemu ze sprawdzeniem, że $P+Q=Q+P$. Zachodzi też równość $(P+Q)+R=P+(Q+R)$.. Wszystkie przedstawione fakty dowodzą, że punkty krzywej eliptycznej stanowią *grupę abelową.*



Rys. 6.6. Krzywa eliptyczna o równaniu $y^2 + y = x^3 - x^2$

Podano jest geometryczną definicję dodawania punktów krzywej. Wyliczamy wzór na sumę dwóch punktów: Jeśli *P=(x',y')*, *Q=(x'',y'')*, a wynik ma współrzędne *P+Q=(x,y)*, to:

$$x = -x' - x'' + ((y' - y'')/(x' - x''))^2$$
$$y = -y' + (x' - x'')(y' - y'')/(x' - x'')$$

Wzór jest zupełnie inny, gdy *P=Q*. Zamiast siecznej przechodzącej przez dwa punkty trzeba rozpatrzyć styczną do jednego punktu. Jeśli zaś punkty są różne, ale *x'=x''*, to sieczna jest pionowa.

Na rysunku 6.6 możemy łatwo zbadać kolejne wielokrotności punktu). Widzimy, że *2P=Q* (oś *x*-ów jest styczna w *P*), *3P=R* (prosta na ukos jest styczna w *Q*), *4P=S* (oś *x*-ów) i *5P=O* (oś *y*-ów).

Załóżmy teraz, że chcemy obliczyć *nP*, gdzie *n* jest dużą liczbą całkowitą, np. 160 bitową. Można wykonać ok. 2^{160} dodawań. Prościej jednak 160 razy wykonać operację "podwój liczbę na wejściu i dodaj *P*, o ile kolejny bit w *n* jest jedynką". Tak więc złożoność operacji mnożenia przez liczbę całkowitą *n* względem liczby bitów w *n* jest liniowa. A co z operacją odwrotną? Ta właśnie zależno jest wykorzystywana w kryptografii.

- **Zastosowanie krzywych eliptycznych w kryptografii**

Wtedy nie mamy do czynienia z liczbami rzeczywistymi, lecz z liczbami całkowitymi. Współrzędne *x* i *y* punktów przebiegają zbiór liczb całkowitych, ale liczonych modulo pewna bardzo duża liczba, może być to liczba pierwsza, może być to potęga dwójki. Jest ich skończenie wiele. Krzywa też nie wygląda tak jak na rysunkach. Ponieważ liczymy reszty z dzielenia, jest więc tak jakbyśmy zwinęli płaszczyznę z wykresem najpierw w „tulejkę", a potem w „obwarzanek". Dokładniej nazywa się to *torus*. Rozpatrujemy jedynie punkty o współrzędnych całkowitych. Po tym zwinięciu zbiór punktów w ogóle nie przypomina krzywej. Gdyby rozciąć *torus* i rozpłaszczyć go, zbiór punktów krzywej eliptycznej będzie wyglądał całkiem losowo. W szczególności dodawanie nie będzie niczego przypominać. Ale wzory na obliczanie iloczynu *nP* są nadal prawdziwe, trzeba tylko pamiętać, że obliczenia dokonywane są na resztach z dzielenia. Przy odpowiednim doborze krzywej (a jest kilka krzywych, których należy unikać), liczba punktów na krzywej jest zbliżona do podstawy arytmetyki modularnej, tzn. jest ich naprawdę bardzo dużo. Tworzą one grupę ze względu na dodawanie. Musimy jeszcze pamiętać o zapewnieniu, że nie ma w niej podgrup, o krótkich cyklach.

Jednym z zastosowań zagadnienia logarytmu dyskretnego jest protokół Diffiego-Hellmana wymiany klucza. W przypadku kryptografii krzywych eliptycznych zakładamy, że Obiekt1 i Obiekt2 uzgodnili krzywą eliptyczną i sposób w jaki punkty krzywej będą odpowiadać ciągom bitów. Wybrali też punkt *Q* na niej. Zrobili to jawnie, a może jest to część powszechnie znanego oprogramowania. Teraz uzgadniają klucz tajny, sesyjny. W tym celu generują swoje liczby losowe i przesyłają następujące komunikaty:

1. $A \longrightarrow B : k_A Q$
2. $B \longrightarrow A : k_B Q$

Teraz i Obiekt1 i Obiekt2 są w stanie obliczyć punkty $k_A k_B Q$, każde z nich mnoży otrzymany punkt przez wybraną przez siebie, a nieznaną innym, liczbę. Będzie to ich klucz na użytek jednej sesji. Użytkownicy znający jedynie przesyłane komunikaty, tzn. $k_A Q$ oraz $k_B Q$, nie potrafią, przynajmniej według dzisiejszego stanu wiedzy, obliczyć $k_A k_B Q$.

Innym zastosowaniem zagadnienia logarytmu dyskretnego jest protokół ElGamala do tajnej wymiany wiadomości. Po uzgodnieniu klucza jak wyżej Obiekt1 wysyła do Obiektu2 tajną wiadomość *M*:

1. $A \longrightarrow B : (nQ, M + nk_B Q)$

Liczba *n* jest liczbą jednorazową, przesłaną tylko w tej wiadomości. Obiekt2 oblicza *M*, bo zna swój tajny klucz i może wykonać proste obliczenie. Inne użytkownicy nie mają dostępu do przesyłanej wiadomości.

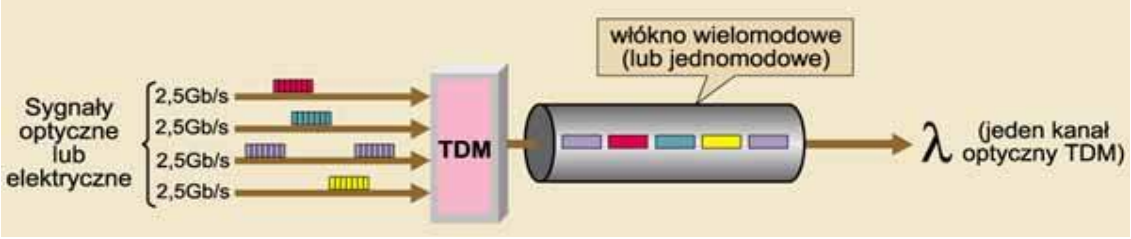
MPC180 doskonale współpracuje z systemami przechowującymi dane, takimi jak systemy bazo danowe, MRPII/ERP, CRM (systemy wspomagania finansowego), kadrowo-płacowe, wspierania sprzedaży. Procesor może przetwarzać kod programu korzystając z pamięci ROM oraz styków sieci lokalnych i rozległych. Poniżej opisano przykład zastosowania procesora w ruterze regionalnym.

Ruter regionalny łączy oddalone od siebie sieci lokalne oddziałów przedsiębiorstwa, jak np. 10/1000 Ethernet. W układzie rutera wykorzystano zwielokrotnianie z podziałem czasu TDM (Time Division Multiplexing)[2], dzięki któremu można przesyłać dane z dużą prędkością między wieloma hostami

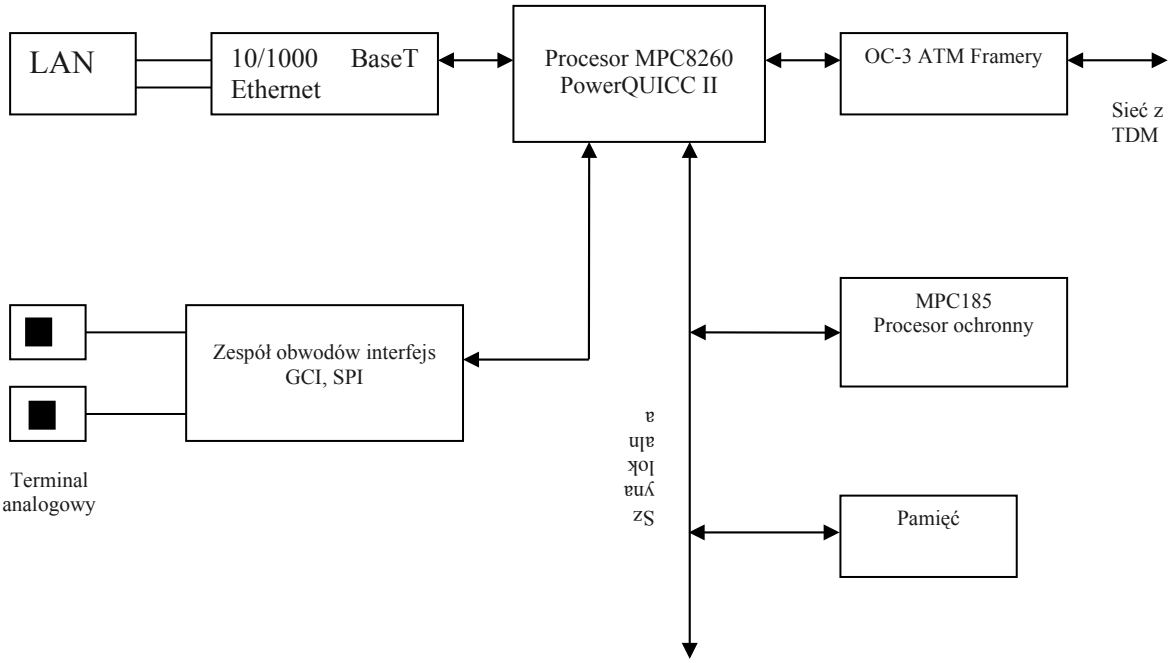
Ruter gwarantuje bezpieczne połączenia pomiędzy biurami regionalnymi a biurem centralnym z wykorzystaniem tunelowania VPN (technologia VPN stosuje tzw. tunele pomiędzy bramami, by chronić prywatne dane w czasie ich przesyłania przez Internet - tunelowanie jest procesem szyfrowania pakietów z danymi, tak by uczynić je niemożliwymi do przeczytania, gdy przechodzą przez sieć publiczną. Taki tunel VPN zestawiony przez Internet chroni wszystkie dane przechodzące przez niego, niezależnie od aplikacji). Podobnie jak w ruterze łączącym sieci lokalne oraz rozlegle, bardzo ważna rolę odgrywa procesor komunikacyjny MPC826O PowerQUICC II który obsługuje interfejsy Ethernetu 10/100 Base-T, framery T1/E1 i T3/E3 po stronie sieci z TDM i przepływności rzędu 155 Mbit/s dla sieci ATM (Asynchronous Transfer Modę).

Procesor komunikacyjny zapewnia bezpośredni interfejs dla większości urządzeń działających w warstwie fizycznej. Procesor ochronny MPC185 jest łatwo integrowany z systemami PowerQUICC II przez magistralę 60x (jest to szyna pamięci do której podłączony jest SDRAM, EEPROM i SDRAM DIMM's). MPC185 idealnie nadaje się do sieci VPN, gdyż używa bezpośredniego dostępu do pamięci systemu i nie obsługuje transferu danych przez mosty oraz dodatkowe magistrale. Bezpieczna wymiana danych między oddalonymi od siebie biurami jest możliwa dzięki procesorom ochronnym, które obsługują protokół IPSec. Przetwarzanie protokołu IPSec przez główny procesor zmniejsza wydajność systemu, dlatego zastosowanie procesora ochronnego jest konieczne w celu utrzymania transferu danych o wysokiej przepływności np.: 44 Gigabitów (procesor sieciowy Radware Fireproof oparty na technologii Motoroli z procesorem MPC7410 Power PC).

[2] TDM (Time Division Multiplexing) – zwielokrotnianie z podziałem czasu, zachowanie stałych odstępów pomiędzy kolejnymi paczkami informacji, dzięki któremu można przesyłać dane z dużą prędkością między wieloma hostami (rys. 6.7).



Rys. 6.7. Zasada zwielokrotnienia z podziałem czasu TDM.



Rys. 6.8. Schemat blokowy rutera regionalnego

Tabela 6.3

Wyjaśnienie do rysunku 6.8	
LAN	podłączenie do sieci lokalnej
10/1000 BaseT Ethernet	karta sieciowa lub interfejs dostępowy umożliwiający podłączenie do sieci lokalnej
Procesor MPC8260 PowerQUICC II	procesor sieciowy oparty na technologii Motoroli – procesory tego typu charakteryzują się równoległym wykonywaniem wielu operacji, szerokim pasmem łączącym komponenty oraz szerokim stopniem integracji
MPC185 Procesor ochronny	procesor ochronny służy do szybkiego realizowania wielu złożonych algorytmów kryptograficznych, kompresji oraz uwierzytelniania danych
Pamięć	tutaj chodzi o RAM – Random Access Memory
OC-3 ATM Framery	wbudowany port, styk fizyczny SC o przepływności rzędu 155.52 Mb/s
Zespół obwodów interfejs GCI, SPI	GCI – General Communications Interface, SPI – System Packet Interface - są to interfejsy służące do łączenia w segmenty podobnych urządzeń sieciowych