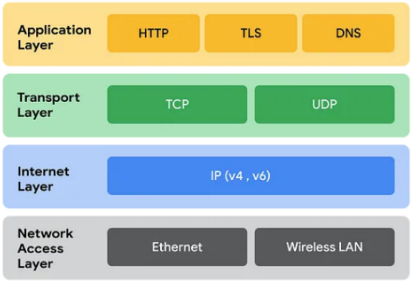
Licencjonowanie w sieciach komputerowych

inż. Mikołaj Nowak inż. Jakub Grzybowski inż. Wojciech Baranowski

15 maja 2025

Czym jest stos TCP/IP?

- Zestaw protokołów komunikacyjnych używanych w sieciach komputerowych.
- Obejmuje cztery główne warstwy:
  - Aplikacji
  - Transportową
  - Internetową
  - Dostępu do sieci
- Jest zaimplementowany jako część jądra systemu operacyjnego.
- Umożliwia komunikację w Internecie i sieciach lokalnych.



Rodzaje licencji oprogramowania

- **Otwarte licencje (Open Source):**
  - **GPL (GNU General Public License):** wymaga udostępnienia kodu źródłowego oraz wszelkich zmian – tzw. *copyleft*.
  - **BSD, MIT:** pozwalają na modyfikację i wykorzystanie komercyjne bez obowiązku udostępniania zmian.
- **Zamknięte licencje (Proprietary):**
  - Kod źródłowy nie jest publicznie dostępny.
  - Brak możliwości legalnej modyfikacji i redystrybucji.
  - Oprogramowanie objęte licencją końcowego użytkownika (*EULA*).
- Licencje wpływają na rozwój, bezpieczeństwo i elastyczność oprogramowania.

Znaczenie licencji dla stosu TCP/IP

- Implementacja stosu TCP/IP jest zwykle częścią jądra systemu operacyjnego.
- Licencja jądra decyduje o:
  - dostępie do kodu źródłowego stosu,
  - możliwości modyfikacji i ponownego rozpowszechniania,
  - integracji z innym oprogramowaniem (np. komercyjnym).
- **Otwarte jądra** (np. Linux, BSD) pozwalają na:
  - audyt bezpieczeństwa,
  - eksperymenty naukowe i edukacyjne,
  - tworzenie niestandardowych rozszerzeń.
- **Zamknięte jądra** (np. Windows, iOS) ograniczają kontrolę nad działaniem sieci.

- Jądro Linuxa jest licencjonowane na zasadach **GPLv2 (GNU General Public License)**.
- Stos TCP/IP jest jego integralną częścią – licencja obejmuje cały kod źródłowy.
- Użytkownicy mają pełen dostęp do kodu, mogą go:
  - analizować,
  - modyfikować,
  - dystrybuować dalej (z zachowaniem GPL).
- Bogate możliwości rozszerzania dzięki modułom takim jak:
  - `netfilter`, `nftables` – filtrowanie pakietów,
  - `eBPF` – dynamiczne programowanie zachowania stosu w jądrze.
- Wykorzystywany w wielu środowiskach: od serwerów i komputerów po Androida i IoT.

macOS / iOS

- Systemy Apple bazują na **Darwinie** – jądrze typu Unix, opartym częściowo na **FreeBSD**.
- Część komponentów (w tym fragmenty stosu TCP/IP) pochodzi z BSD i są dostępne na licencji **BSD**.
- Apple jednak wprowadza własne rozszerzenia i modyfikacje, które:
  - nie są publicznie dostępne,
  - objęte są licencjami zastrzeżonymi,
  - mogą być zamknięte mimo otwartego „rdzenia".
- Oficjalna licencja źródłowego Darwina: **APSL (Apple Public Source License)** – niezgodna z GPL, uważana za problematyczną.
- Stos TCP/IP w macOS/iOS to więc:
  - kombinacja komponentów BSD,
  - zamkniętych rozszerzeń Apple,
  - fragmentów o niejednoznacznym statusie licencyjnym.

- Systemy Windows korzystają z własnościowego stosu TCP/IP – zamkniętego i niedostępnego publicznie.
- Stos został zaimplementowany samodzielnie przez Microsoft – początkowo w Windows NT, dziś obecny we wszystkich wersjach.
- Brak dostępu do kodu źródłowego oznacza:
  - brak możliwości modyfikacji lub audytu,
  - pełną zależność od aktualizacji Microsoftu,
  - niemożność dostosowania do nietypowych zastosowań.
- Licencjonowanie odbywa się wraz z systemem – użytkownik akceptuje *EULA*, bez wpływu na wewnętrzne komponenty.
- Dla programistów dostępne są tylko wysokopoziomowe API (np. WinSock), ale nie kod źródłowy implementacji.

BSD (FreeBSD / OpenBSD / NetBSD)

- Rodzina systemów BSD korzysta z licencji **BSD** – bardziej liberalnej niż GPL.
- Licencja pozwala na:
  - modyfikację i dowolne wykorzystanie kodu,
  - zamknięcie kodu w produktach komercyjnych bez obowiązku publikacji zmian.
- **FreeBSD** – popularny w serwerach i systemach NAS (np. TrueNAS).
- **OpenBSD** – znany z nacisku na bezpieczeństwo i audyt kodu.
- **NetBSD** – ekstremalnie przenośny, działa na setkach architektur.
- Kod stosu TCP/IP z BSD jest wykorzystywany m.in. w:
  - MacOS i iOS (częściowo)
  - Juniper JunOS,
  - Sony PlayStation.

RouterOS / VxWorks / Cisco IOS

- Systemy te są ściśle powiązane ze sprzętem i mają wbudowany, zamknięty stos TCP/IP.
- **RouterOS (MikroTik):**
  - oparty częściowo na Linuksie, ale całość zamknięta,
  - brak dostępu do kodu źródłowego,
  - licencjonowany na zasadach komercyjnych – wg klucza lub poziomu.
- **Cisco IOS:**
  - zamknięty system operacyjny dla routerów i przełączników Cisco,
  - zintegrowany stos TCP/IP, brak możliwości modyfikacji,
  - licencja przypisana do urządzenia (hardware-locked).
- Wspólną cechą jest:
  - brak otwartości i modyfikowalności,
  - pełna kontrola producenta nad aktualizacjami i funkcjami.

Porównanie podejścia systemów operacyjnych

System operacyjny	Kod źródłowy	Licencja	Modyfikowalność	Typowe zastosowanie
Linux	Tak	GPLv2	Pełna	Serwery, IoT, Android
macOS iOS	Częściowo	Mieszana	Ograniczona	Komputery i urządzenia Apple
Windows	Nie	Komercyjna	Brak	Komputery osobiste, środowiska korporacyjne
FreeBSD OpenBSD NetBSD	Tak	BSD	Pełna	Routery, OS-y wbudowane, macOS (pośrednio)
RouterOS Cisco IOS VxWorks	Nie	Komercyjna sprzętowa	Brak	Routery, urządzenia sieciowe, systemy embedded

Lekkie stosy TCP/IP – lwIP, uIP

- Niektóre zastosowania (IoT, mikrokontrolery, embedded) wymagają lekkich implementacji stosu TCP/IP.
- **lwIP (lightweight IP):**
  - Licencja BSD – pełna dowolność wykorzystania,
  - zoptymalizowany pod wydajność i niskie zużycie zasobów,
  - używany w systemach z ograniczoną pamięcią (np. ESP32, STM32).
- **uIP (micro IP):**
  - Jeszcze mniejszy niż lwIP – działa na urządzeniach z ¡64KB RAM,
  - zintegrowany z systemem Contiki (IoT, sensory),
  - podstawowe wsparcie dla TCP, UDP, ICMP.
- Oba stosy są często używane w środowiskach, gdzie pełny OS byłby zbyt ciężki.

Podsumowanie i wnioski

- **Licencja ma realny wpływ** na to, jak stos TCP/IP może być używany, rozwijany i modyfikowany.
- **Otwarte systemy** (Linux, BSD):
  - umożliwiają pełny dostęp do stosu TCP/IP,
  - wspierają eksperymenty, badania i rozwój,
  - promują transparentność i bezpieczeństwo.
- **Zamknięte systemy** (Windows, macOS, RouterOS):
  - ograniczają kontrolę użytkownika,
  - wymagają zaufania do producenta,
  - są trudne lub niemożliwe do audytu.
- Wybór systemu to wybór między elastycznością a wygodą (i czasem – wsparciem komercyjnym).

Sieci komputerowe dzisiaj nieodzownie łączą się z kryptografią, która pozwala nam zapewnić atrybuty bezpieczeństwa informacji takie jak: poufność, integralność, uwierzytelnianie i niezaprzeczalność. Nie zawsze tak jednak było...

Historia eksportu kryptografii z USA

- Podczas zimnej wojny istniały silne regulacje ograniczające eksport technologii krytycznej z USA do Bloku Wschodniego, do której zaliczana była kryptografia
- Eksportem zarządzała organizacja CoCom (Coordinating Committee for Multilateral Export Controls)
- W latach 60 organizacje finansowe zaczynały zgłaszać zapotrzebowanie na opracowanie rozwiązań kryptograficznych celem zabezpieczenia rozwijających się przelewów elektronicznych

Data Encryption Standard - DES

- Na początku lat 70 NIST (National Institute of Standards and Technology) zdecydowało się na wprowadzenie rządowego standardu szyfrowania.
- IBM w odpowiedzi na zapotrzebowanie zaproponowało swój standard - znany później jako DES
- Oryginalny DES zakładał wykorzystanie kluczy 128-bitowych, NSA (National Security Agency) nalegało na skrócenie długości do 48 bitów. Jako "kompromisźastosowano klucze 56-bitowe.
- Ponadto NSA wprowadziło zmiany do algorytmu, których uzasadnienie nie zostało podane do wiadomości publicznej. Pojawiły się spekulacje o wprowadzenie tylnich drzwi
- DES został opublikowany w 1977 jako standard federalny FIPS PUB 46

Rivest–Shamir–Adleman Cryptosystem - RSA

- W 1977 trójka naukowców opracowała algorytm RSA
- Próba opublikowania algorytmu mogłaby skutkować karą więzienia dla autorów
- Algorytmy kryptograficzne z punktu widzenia prawa traktowane były jak broń
- Dokument opisujący RSA był kopiowany i przekazywany pocztą pantoflową
- Rząd USA ugiął się i pozwolił na publikację pracy w 1979
- W 1983 algorytm został opatentowany (w USA). Patent wygasł w roku 2000.

Pretty Good Privacy - PGP

- W 1991 Phil Zimmerman stworzył program PGP umożliwiający szyfrowanie oraz cyfrowe podpisywanie tekstu, maili, plików, katalogów a nawet całych partycji dyskowych
- Było to jedno z pierwszych narzędzi które umożliwiały każdemu stosowanie algorytmów kryptograficznych
- Opublikowany w domenie publicznej (public domain)
- Oprogramowanie trafiło też poza USA
- Rząd stanów zjednoczonych rozpoczął śledztwo przeciwko Zimmermanowi pod kątem nielicencjonowanego eksportu broni
- Wówczas wszystkie systemy kryptograficzne używające ponad 40 bitów były uważane za broń. PGP używało kluczy 128-bitowych

PGP - naruszenie patentu RSA

- Zimmerman naruszył także patent który od 1983 obejmował RSA.
- Organizacja, która zarządzała patentem na RSA zażądała od Zimmermana zaprzestania dystrybucji PGP
- Sam Zimmerman faktycznie zaprzestał...
- Co dało odwrotny efekt - oprogramowanie jeszcze bardziej zyskało na popularności i ludzie jeszcze chętniej je rozpowszechniali

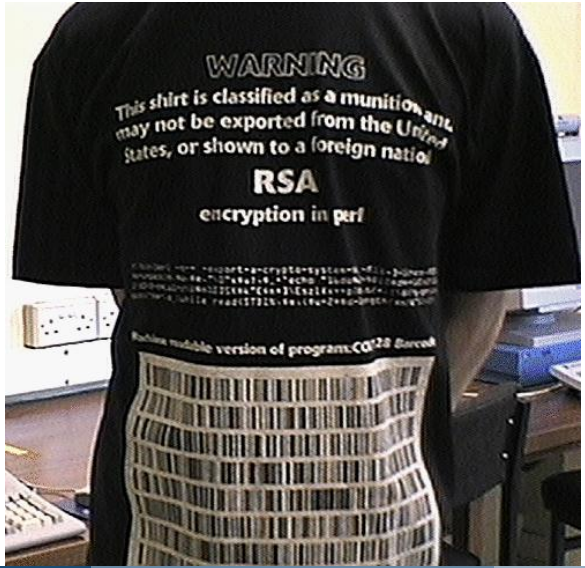
PGP - pierwsze zastosowania i obawy

- PGP świetnie sprawdziło się u ludzi żyjących w krajach rządzonych przez opresyjne rządy
- NSA argumentowało, że oprogramowanie będzie używane przez pedofili i przestępców
- W praktyce takie argumenty można zastosować do wielu innych wynalazków

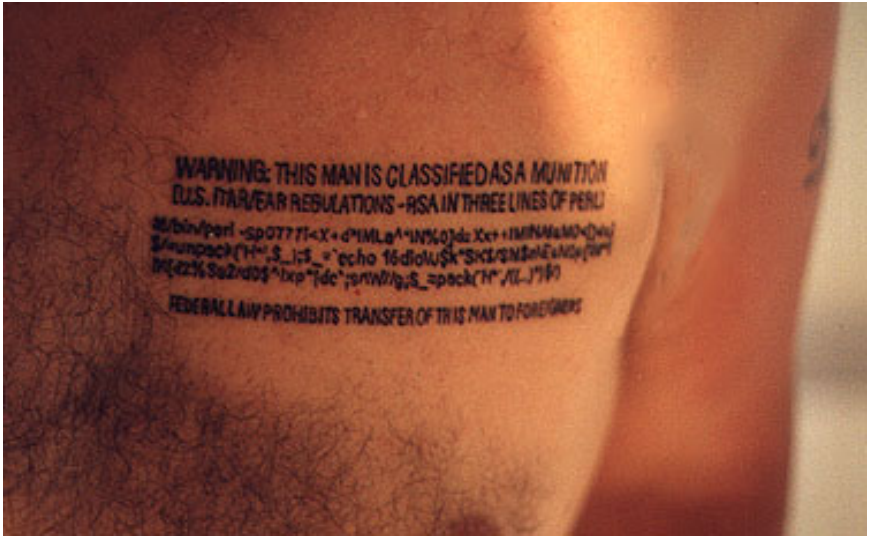
PGP vs USA - ciąg dalszy

- Rząd USA naciskał na konieczność wdrażania tylnych drzwi dla służb przez firmy tworzące oprogramowanie
- Administracja Billa Clintona powiedziała kongresowi „Americans have no Constitutional right to choose their own method of encryption”
- Ludzie zwracali uwagę na podobieństwo kodu i innych form wolności słowa
- Zimmerman przekonał MIT do wydrukowania kodu, oprawienia go w książkę i wysłania do europejskich bibliotek
- Rząd wiedział, że gdyby zablokowali wydanie książki, skutkowałoby to sprawą sądową którą by przegrał ze względu na naruszenie pierwszej poprawki (wolność religii, słowa, prasy, zgromadzeń i petycji)
- Stosowano też inne, niecodzienne formy eksportu

Eksport kryptografii - obejścia



Eksport kryptografii - obejścia



Eksport kryptografii - obejścia

PGP vs USA - ciąg dalszy

- Sąd po przeanalizowaniu sprawy Zimmermana uznał, że oprogramowanie szyfrujące jest chronione przez pierwszą poprawkę i nie powinno się ograniczać jego rozpowszechniania
- Wszelkie akcje prawne przeciwko Zimmermanowi zostały wstrzymane
- Do dnia dzisiejszego rządy nie są zadowolone z kryptografii i często naciskają na producentów oprogramowania na wprowadzanie do oprogramowania tylnej furtki

PGP dziś

- PGP zostało przejęte przez firmę Network Associates Inc. (NAI) i przekształciło się w rozwiązanie komercyjne
- Oprócz rozwiązania komercyjnego powstał też otwarty standard OpenPGP
- Najbardziej znaną implementacją standardu OpenPGP jest GnuPG (GPG), działającą w latach 1997-2007 na licencji GPL-2.0-or-later a od 2007 na licencji GPL-3.0-or-later
- GPG nie może używać opatentowanych algorytmów, PGP (komercyjne) może używać tych, na które wykupiło licencje
- Jako że patenty działają w obrębie danego kraju, istnieją wtyczki umożliwiające GPG korzystanie z algorytmów zastrzeżonych np. w USA. Oczywiście nie powinny być one stosowane przez mieszkańców USA.

Inne rozwiązania

Do innych rozwiązań kryptograficznych działających w sieciach komputerowych możemy zaliczyć:

- OpenSSL - wieloplatformowa implementacja protokołów SSL i TLS i innych algorytmów kryptograficznych. Wcześniej udostępniana była na licencji OpenSSL License / SSLeay license, zbliżonej do licencji Apache, od wersji 3.0 wydanej w 2018 roku obowiązuje licencja Apache-2.0
- OpenVPN - oprogramowanie do tworzenia sieci VPN udostępniane na licencji GPLv2
- WireGuard - oprogramowanie do tworzenia VPN, nowsze, prostsze i szybsze niż OpenVPN, licencje różnią się w zależności od implementacji (MIT/Apache 2.0/GPLv2/GPLv3)
- OpenSSH - oprogramowanie bazujące na protokole SSH umożliwiające zestawienie bezpiecznego połączenia w niezabezpieczonej sieci. Udostępnione na licencji BSD.

Wirtualizacja sieci komputerowych

W ostatnich dwóch dekadach, rozwinęła się znacząco koncepcja przeniesienia sieci komputerowych z fizycznych urządzeń (przełączników, routerów) do postaci oprogramowania

- IETF bada wyniesienie płaszczyzny sterowania z urządzeń sieciowych (Forwarding and Control Element Separation – ForCES, 2004)
- Projekt Ethane – podwaliny pod kolejne rozwiązania (Stanford, 2007)
- Paradygmat sieci definiowanych programowo – SDN – oraz powstanie protokołu OpenFlow (Stanford, 2008)
- Powstanie organizacji Open Network Foundation, finansowanej przez największe spółki cyfrowe (Facebook, Google czy Microsfot) w celu rozpromowania koncepcji SDN i rozwoju OpenFlow (2011)
- Implementacja SDN w centrach danych (m.in. przez Google) przez autorskie rozwiązania (2012)

Czym jest Open Compute Project (OCP)?

- Inicjatywa zapoczątkowana przez Facebooka w 2011 roku.
- Celem jest otwarte projektowanie sprzętu dla centrów danych: serwerów, przełączników, pamięci.
- Promuje open hardware – otwarte specyfikacje i współdzielone projekty.
- Partnerzy: Facebook, Intel, Microsoft, Google, Dell, IBM i inni.
- Zalety: niższe koszty, większa elastyczność, lepsza efektywność energetyczna.

Microsoft dołącza do OCP (2014)

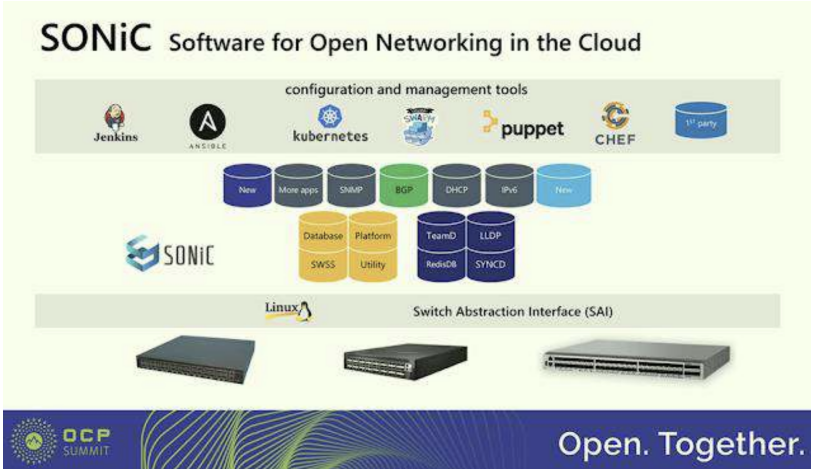
- Microsoft dołącza do OCP jako contributing member w 2014 r.
- Udostępnia specyfikacje sprzętowe używane w chmurze Azure.
- Cele:
  - Standaryzacja i interoperacyjność infrastruktury.
  - Przejście z zamkniętych do otwartych rozwiązań.
  - Stworzenie podstaw pod otwarte projekty sieciowe (np. SONiC).
- Źródło: `azure.microsoft.com/blog/microsoft-joins-open-compute-project`

Koncepcja SONiC

W 2016 roku, firma Microsoft w ramach członkostwa w OCP postanowiła utworzyć w pełni niezależną od sprzętu platformę sieciową, która byłaby uniwersalnym frameworkiem dla centrum danych (w MS na potrzeby Azure). Cechy rozwiązania:

- **Przykład NOS** – Network Operating System, czyli sieciowego systemu operacyjnego
- **Modularność i konteneryzacja** – komponenty SONiC działają jako niezależne kontenery Docker, co ułatwia zarządzanie i aktualizacje.
- **Niezależność od sprzętu** – dzięki SAI (Switch Abstraction Interface), SONiC działa na przełącznikach wielu dostawców, eliminując vendor lock-in.
- **Skalowalność** – zaprojektowany do dużych centrów danych, obsługuje funkcje klasy operatorskiej jak BGP, RDMA, ACL, ECMP.

SONiC – schemat



Rysunek: Architektura rozwiązania SONiC

Licencjonowanie SONiC – przegląd ogólny

- **SONiC** jest oprogramowaniem open-source, aktualnie rozwijanym przez Linux Foundation
- Zbudowany z wielu modułów – większość udostępniona na licencji **Apache 2.0**.
- Część komponentów pomocniczych pochodzi ze świata Linuksa i korzysta z licencji **GPLv2**.
- Komercyjni dostawcy (np. Dell, Broadcom) mogą oferować rozszerzenia na licencjach zamkniętych.
- SONiC zachęca do modularności – dzięki temu łatwiej kontrolować zgodność licencyjną.

Licencjonowanie SONiC – warstwa po warstwie

- **Kernel i narzędzia systemowe:** bazują na Debianie/Linux – licencja **GPLv2**.
- **Platforma NOS:** główne komponenty SONiC (orchestrator, systemd services, syncd) – licencja **Apache 2.0**.
- **SAI (Switch Abstraction Interface):** udostępniany przez OCP, najczęściej na **Apache 2.0**, ale konkretna implementacja zależy od dostawcy ASIC.
- **Interfejsy API i narzędzia zarządzania:** CLI, REST API, gRPC – przeważnie Apache 2.0.
- **Dodatki i pluginy:** mogą być na licencjach mieszanych (np. MIT, BSD, własne licencje dostawców).

POLITECHNIKA GDAŃSKA | Bibliografia

- Thomas R. Johnson Ąmerican Cryptology during the Cold War, 1945-1989. Book III: Retrenchment and Reform, 1972-1980"
- Film dokumentalny Ćypherpunks Write Code"produkcji magazynu Reason
- philzimmermann.com
- Stallings, W.: Cryptography and network security: principles and practice.
- Casado M. et al. Ęthane: Taking Control of the Enterprise"(Stanford, 2007)
- McKeown N. et al. ÓpenFlow: Enabling Innovation in Campus Networks"(Stanford, 2008)
- https://sonicfoundation.dev/
- https://en.wikipedia.org/wiki/SONiC_(operating_system)
- Tanenbaum A. S. "Modern Operating Systems", Prentice Hall
- https://www.netburner.com/learn/lwip-stack-overview/
- https://contiki-os.org/
- Licencje GPL i BSD – https://opensource.org/licenses

POLITECHNIKA
GDAŃSKA