Bezpieczeństwo informacji i kryptografia
- przegląd zagadnień

dr inż. Sławomir Samolej

email: ssamolej@kia.prz.edu.pl

WWW: ssamolej.kia.prz.edu.pl

Literatura

- Jason Andress, Podstawy bezpieczeństwa informacji – praktyczne wprowadzenie, Helion, 2022
- Kurs internetowy „Cryptography I” https://www.coursera.org,
- Książka w wolnym dostępie: Dan Boneh and Victor Shoup, „A Graduate Course in Applied Cryptography”, 2025, https://toc.cryptobook.us/
- Jean-Philippe Aumasson, Nowoczesna Kryptografia, Praktyczne wprowadzenie do szyfrowania, PWN 2018

Definicja

- **Bezpieczeństwo informacji** to:
  - Ochrona informacji i systemów informatycznych przed nieautoryzowanym
    - dostępem,
    - wykorzystaniem,
    - ujawnieniem,
    - zakłóceniem,
    - modyfikacją
    - zniszczeniem.

Bezpieczeństwo - produktywność

„ Jedyny naprawdę bezpieczny system to taki, który jest odłączony od zasilania, zalany w bloku betonu i zamknięty w wyłożonym ołowiem pokoju patrolowanym przez uzbrojonych strażników – ale nawet w takiej sytuacji mam pewne wątpliwości.”

Prof. Eugene Spafford

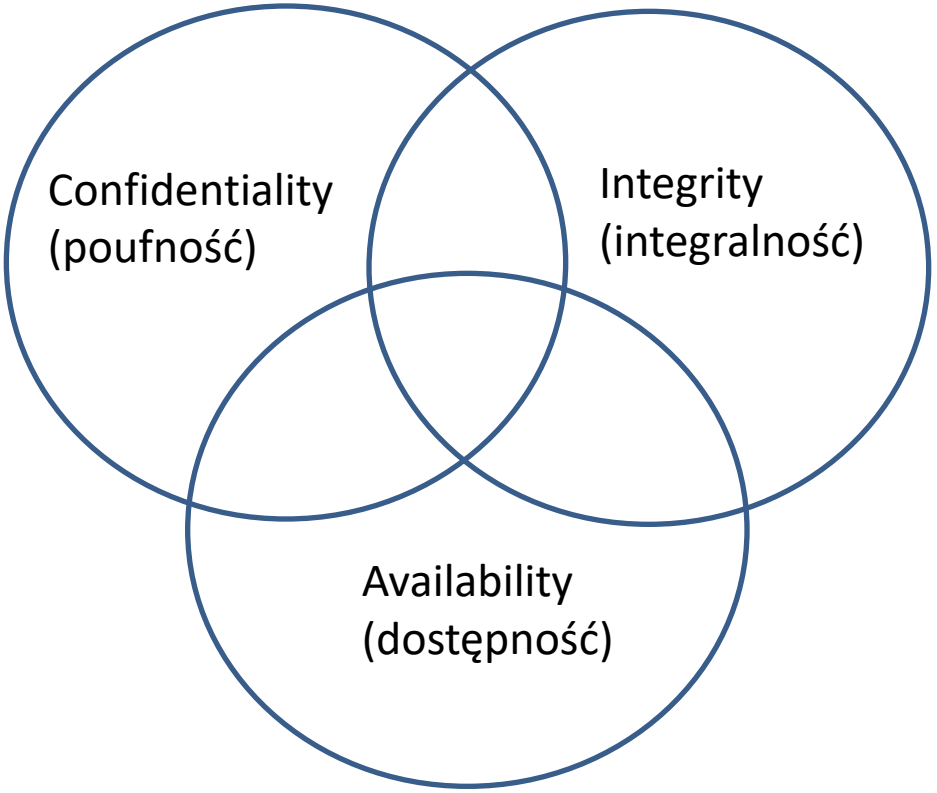
Zwiększenie poziomu bezpieczeństwa

->

obniżenie poziomu produktywności.

Triada CIA

- Jeden z modeli rzeczywistości pozwalający opisać rzeczywistość z punktu widzenia bezpieczeństwa
- Pomaga w opisywaniu ataków



Odwrotnie: Disclosure/Alteration/Denial
(ujawnianie/zmiana/odmowa)

Poufoność

- Zdolność do ochrony danych przed osobami, które nie są upoważnione do ich przeglądania

Integralność

- Zdolność do zapobiegania zamianom danych w nieuprawniony lub niepożądany sposób.
  - Zapobieganie zamianom
  - Wykrywanie zmian
  - Możliwość cofnięcia zmian

Dostępność

- Możliwość dostępu do Twoich danych, gdy ich potrzebujesz.
  - Awaria zasilania?
  - Uszkodzenie SO
  - Atak DoS

Odniesienie do bezpieczeństwa - przykład

- Przykład: transport niezaszyfrowanych dysków z jedynymi kopiami danych ginie podczas przejazdu.
  - Powstaje incydent bezpieczeństwa:
    - Naruszenie poufności (pliki były niezaszyfrowane)
    - Naruszenie integralności (nie wiemy, czy zostały zmodyfikowane)
    - Naruszenie dostępności (nie mieliśmy dostępu do tej informacji)

Rodzaje ataków

- Przechwycenie (interception)
- Przerywanie (interruption)
- Modyfikowanie (modification)
- Podrabianie (fabrication)

C	Przechwycenie
I	Przerywanie Modyfikowanie Podrabianie
A	Przerywanie Modyfikowanie Podrabianie

Przechwycenie

- Umożliwienie nieautoryzowanym użytkownikom dostępu do danych, aplikacji lub środowiska celu.
- Są atakami przeciw poufności.
- Przykłady:
  - Nieautoryzowane przeglądanie lub kopiowanie plików
  - Podsłuchiwanie rozmów telefonicznych
  - Czytanie cudzych emaili
- Można je przeprowadzić na:
  - Danych w spoczynku
  - Danych w tranzycie
  - Danych w użyciu

C	Przechwycenie
I	Przerywanie Modyfikowanie Podrabianie
A	Przerywanie Modyfikowanie Podrabianie

Przerywanie

- Tymczasowe lub stałe zniszczenie lub zablokowane dostępu do zasobów
- Wpływają na:
  - Dostępność
  - Integralność (jeśli doszło do manipulacji na procesach w bazie danych)

C	Przechwycenie
I	Przerywanie Modyfikowanie Podrabianie
A	Przerywanie Modyfikowanie Podrabianie

Modyfikacja

- Manipulowanie zasobami
- Atak na:
  - Integralność
  - Dostępność

C	Przechwycenie
I	Przerywanie Modyfikowanie Podrabianie
A	Przerywanie Modyfikowanie Podrabianie

Podrabianie

- Generowanie danych, procesów, komunikacji.
- Typ ataku na:
  - Integralność
  - Dostępność

C	Przechwycenie
I	Przerywanie Modyfikowanie Podrabianie
A	Przerywanie Modyfikowanie Podrabianie

Przegląd najważniejszych obszarów bezpieczeństwa informacji

- Identyfikowanie i uwierzytelnianie
- Autoryzacja i kontrola dostępu
- Audytowanie i rozliczalność
- Kryptografia
- Zgodność, prawo i przepisy
- Bezpieczeństwo operacyjne
- Bezpieczeństwo czynniki ludzkiego
- Bezpieczeństwo fizyczne
- Bezpieczeństwo sieciowe
- Bezpieczeństwo systemu operacyjnego
- Bezpieczeństwo urządzeń mobilnych
- Bezpieczeństwo aplikacji

Identyfikacja i uwierzytelnienie

- **Identyfikacja** – pozwala stwierdzić, czym jest ktoś lub coś
- **Uwierzytelnienie** – pozwala stwierdzić, czy to stwierdzenie jest prawdziwe
- Przykład
  - Zbliżenie karty oznacza, że jesteś osobą wskazaną na karcie
  - Podanie PIN udowadnia, że jesteś prawidłowym posiadaczem karty

Identyfikacja

- Podanie tożsamości – za kogo się podajemy

- Weryfikacja tożsamości to nie uwierzytelnienie
  - Ktoś ogląda dowód, paszport i może na tej podstawie stwierdzić, że rzeczywiście to jest osoba o określonym imieniu, nazwisku i wyglądzie
- Tożsamość można łatwo sfałszować

Uwierzytelnienie

- To zbiór metod używanych do ustalenia, czy dane twierdzenie o tożsamości jest prawdziwe
- Nie należy go mylić z autoryzacją. Uwierzytelnienie mówi kim ktoś jest, autoryzacja – co ktoś może zrobić.

Składniki uwierzytelnienia

- Coś, co znasz – hasło, PIN (niepewne, jeśli zostaną ujawnione, to metoda traci wiarygodność)
- Coś, czym jesteś (biometria)
- Coś, co posiadasz (karty, tokeny, telefony)
- Coś, co robisz (jak chodzisz, jak uderzasz w klawiaturę – duży procent odrzuceń?)
- To, gdzie jesteś (zmiany PIN po zablokowaniu musisz dokonać w siedzibie banku, po potwierdzeniu twojej tożsamości)

- Do uwierzytelnienia będziemy się starać zastosować jak najwięcej z tych składników.

Autoryzacja i kontrola dostępu

- Po identyfikacji i uwierzytelnieniu można zdecydować, do czego dana osoba ma dostęp
  - Autoryzacja = określenie, co może dana osoba robić z naszym systemem i danymi
  - Kontrola dostępu = narzędzie do przydzielania dostępu

Czym są mechanizmy kontroli dostępu?

- Przyznawanie dostępu
  - Udzielenie osobie lub aplikacji możliwości korzystania z zasobów
- Odmawianie dostępu
  - Zabronienie osobie lub aplikacji możliwości korzystania z zasobów
- Ograniczenie dostępu
  - Określenie zbioru zasobów dostępnych (np. sandboxing)
- Cofnięcie dostępu
  - Zablokowanie osobie, aplikacji możliwości korzystania z zasobów np. z powodu zwolnienia itp.

Fizyczna kontrola dostępu

- Wchodzenie „na ogonie”
- Dobra praktyka – procedura wchodzenia na lotnisko i na pokład samolotu

Audytowanie i rozliczalność

- Audytowanie
  - sprawdzanie, jak użytkownicy zachowują się po uzyskaniu dostępu do danych
  - testy penetracyjne – wykrywanie podatności
- Rozliczalność – prowadzenie „dzienników” korzystania użytkowników z zasobów
  - Uruchomienie i pielęgnacja dzienników systemowych
    - Niezaprzeczalność (ktoś nie będzie mógł zaprzeczyć, że coś robił, jeśli będzie na to dowód)
    - Efekt odstraszania (użytkownicy świadomi, że są rozliczani będą mniej skłonni do wykroczeń)
    - Na podstawie logów można wdrożyć system alarmów i zapobiec incydentom lub je ograniczyć
    - Logi muszą być gromadzone i zabezpieczone zgodnie z wymogami formalno-prawnymi, wtedy można je wykorzystać jako materiał dowodowy

Kryptografia

- Ochrona danych w spoczynku
  - Zaszyfrowanie
- Ochrona danych w ruchu
  - Szyfrowanie
  - VPN
- Ochrona danych w użyciu
  - Są osoby, które mogą korzystać z danych i je ujawnić (Edward Snowden)

Bezpieczeństwo operacyjne

- wdrażanie mechanizmów bezpieczeństwa
- identyfikacja, przed czym się bronić

Bezpieczeństwo czynnika ludzkiego

- Człowiek to „najsłabsze ogniwo” w systemie zabezpieczeń
  - Możliwe jest przeprowadzenie na nim ataków socjotechnicznych
  - Ataki można **przygotować**:
    - Śledzenie celu
    - Biały wywiad
    - Życiorysy i oferty pracy
    - Media społecznościowe
    - Google Hacking https://www.exploit-db.com/google-hacking-database
    - Metadane plików (https://www.opentext.com/products/encase-forensic)
    - Shodan (https://www.shodan.io/) – wyszukiwanie urządzeń i systemów podłączonych do Internetu
    - Maltego (https://www.maltego.com) – wyszukiwanie relacji pomiędzy poszczególnymi punktami danych

Rodzaje ataków socjotechnicznych

- Atak pretekstowy (wcielenie się w rolę, wiarygodny scenariusz)
- Phishing (zebranie danych lub zainstalowanie złośliwego oprogramowania)
- Tailgating (wchodzenie na ogonie)

Budowanie świadomości użytkowników 1/2

- Zasady tworzenia i przechowywania haseł
- Szkolenia z zakresu inżynierii behawioralnej
- Szkolenia z korzystania z sieci
  - Publiczne sieci bezprzewodowe
  - Zasady podłączania urządzeń do sieci firmowej
  - VPN
- Złośliwe oprogramowanie
  - Załączniki do email od nieznanych nadawców
  - Załączniki do email potencjalnie wykonywalne
  - Skrócone łącza internetowe (weryfikować: https://linkexpander.com)
  - Łącza internetowe o nieznacznie różniących się nazwach
  - Aplikacje na smartfony do pobrania spoza oficjalnych sklepów
  - Pirackie oprogramowanie

Budowanie świadomości użytkowników 2/2

- Prywatny sprzęt komputerowy
  - Co do zasady żadne prywatnie urządzenie nie powinno być podłączone do sieci firmy
- Polityka czystego biurka
  - Żadne wrażliwe dane nie powinny być pozostawione „na biurku” jeśli opuszczamy miejsce pracy
  - Należy opracować sposób pozbywania się wrażliwych danych (niszczarki papieru i sprzętu)
- Jeśli jesteś osobą odpowiedzialną za bezpieczeństwo, to prowadź szkolenia i dostarcz z nich materiały. Mało prawdopodobne, że ktoś z własnej woli przeczyta przygotowany przez Ciebie materiał tekstowy i od razu zacznie stosować politykę bezpieczeństwa.

Bezpieczeństwo fizyczne

- Ochrona ludzi
- Ochrona danych

Identyfikacja zagrożeń fizycznych

- Kategorie zagrożeń fizycznych
  - Wibracje i przemieszczenia
  - Ogień i dym
  - Toksyny
  - Ludzie
  - Przerwy w dostawach energii
  - Ekstremalne temperatury
  - Gazy
  - Płyny
  - Organizmy żywe
  - Pociski

Ochrona danych

- Fizyczne zagrożenie dla danych
  - Nośniki magnetyczne
    - Zniszczenie może spowodować: silne pole magnetyczne lub drgania
  - Nośniki flash
    - Zniszczenie może spowodować: zgniecenie, impulsy i przepięcia elektryczne
  - Nośniki optyczne
    - Zniszczenie może spowodować: zadrapanie, temperatura
- Dostępność danych
  - Kopie zapasowe
  - RAID (Redundant Arrays of Independent Disks)
  - Replikacja danych
- Szczątkowe pozostałości danych
- Ochrona wyposażenia
  - Fizyczne zagrożenie dla sprzętu (temperatury, ciecze, org. żywe, wibracje, dym i ogień
  - Wybór lokalizacji
  - Zabezpieczenia dostępu (zamki, ogrodzenia, strażnicy)

Bezpieczeństwo sieciowe

Ochrona sieci

- Projektowanie bezpiecznych sieci
  - Segmentacja (podział na podsieci i kontrola ruchu pomiędzy nimi)
  - Tworzenie nadmiarowości
- Zastosowanie zapór sieciowych
  - Filtrowanie pakietów
  - Pełnostanowa inspekcja pakietów (obserwacja ruchu w ramach danego połączenia)
  - Głęboka inspekcja pakietów (odczytywanie pakietów, przed przekazaniem dalej)
  - Strefa zdemilitaryzowana (kilka poziomów zapór sieciowych)
  - Sieciowe systemy wykrywania włamań

Ochrona ruchu sieciowego

- VPN (virtual private network)
  - Zdalny dostęp do wewnętrznych zasobów organizacji
  - Ochrona lub anonimizacja ruchu przesyłanego przez niezaufane połączenie (odłączenie dostawcy od śledzenia naszego ruchu)
- Sieci bezprzewodowe
  - Publiczne sieci bezprzewodowe (brak szyfrowania, brak haseł)
  - Wrogie punkty dostępu
- Dobór bezpiecznych protokołów komunikacyjnych
  - ftp -> SFTP
  - telnet -> SSH
- Narzędzia do ochrony sieci
  - Skanery (np. nmap)
  - Sniffery (np. wireshark)

Bezpieczeństwo systemu operacyjnego

- Utwardzanie SO
  - Usuwanie niepotrzebnego oprogramowania
  - Usuwanie niepotrzebnych usług
  - Modyfikacja domyślnych kont użytkowników
  - Zastosowanie reguły najmniejszych uprawnień
  - Aktualizacja
  - Implementacja logowania i audytowania
- Ochrona przed złośliwym oprogramowaniem
  - Antywirusy
  - Ochrona przestrzeni wykonywalnej
  - Programowe zapory sieciowe i wykrywania włamań
  - Skanery
  - Narzędzia do wykrywania podatności i luk

Bezpieczeństwo aplikacji

Luki w zabezpieczeniach oprogramowania

- Przepełnienie bufora (wprowadzenie na wejście większej liczby danych, niż przewidział autor oprogramowania -> nadpisanie pamięci własnej lub innych aplikacji)
- Warunki wyścigu (systemy współbieżne – dwa programy równocześnie modyfikują te same dane -> niespójność danych)
- Ataki na weryfikację danych wejściowych (nieprawidłowe dane)
- Ataki uwierzytelniające (uwierzytelnienie powinno się znaleźć po stronie serwera, a nie klienta, bo wtedy jest łatwiejsze do sfałszowania)
- Ataki autoryzacyjne (przejęcie zasobów, do których nie ma się uprawnień)
- Ataki kryptograficzne

Bezpieczeństwo sieci WEB

- Ataki po stronie klienta:
  - XXS (cross-site scripting) – umieszczenie dodatkowych skryptów na stronie, którą otwiera klient
  - CSRF (cross-site request forgery) – umieszcznie na stronie odsyłaczy do innych, które wykonają się automatycznie (my pracujemy na jednej zakładce, a na drugiej, gdzie też jesteśmy zalogowani, ktoś robi za nas zakupy za nasze pieniądze)
- Ataki po stronie serwera:
  - Brak weryfikacji danych po stronie serwera -> directory traversal
  - Przypisanie niewłaściwych uprawnień użytkownikom -> możliwość przejęcia wrażliwych danych
  - Zbędne pliki (po deweloperce)

Bezpieczeństwo baz danych

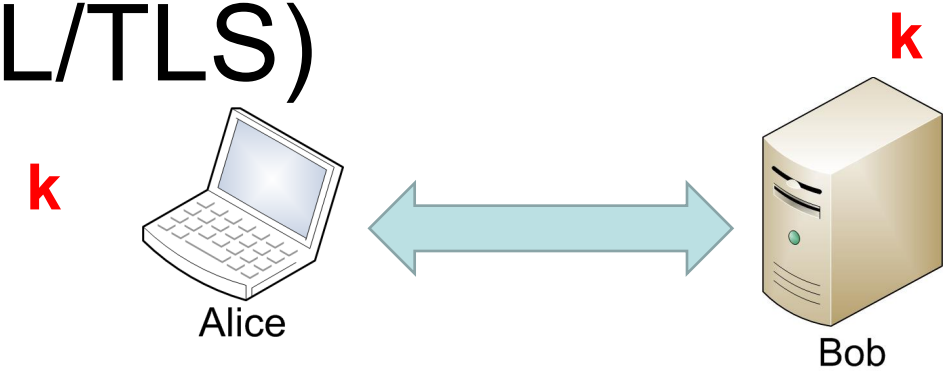
- Problemy z protokołami (połączenie z bazą przez sieć może mieć luki)
- Dostęp do funkcjonalności bez uwierzytelnienia (użytkownik nie jest uwierzytelniany i dostaje duże przywileje)
- Arbitralne wykonanie kodu (specyfika SQL)
- Eskalacja uprawnień (wstrzykiwanie kodu)

Kryptografia jest wszędzie…

- **Ochrona przesyłu danych**
  - Chroniony dostęp do stron: HTTPS
  - Sieci bezprzewodowe: 802.11i WPA2, GSM, Bluetooth
- **Szyfrowanie danych na dyskach:** EFS, TrueCrypt
- **Ochrona informacji:** (np. DVD, Blu-ray): CSS, AACS
- **Autentykacja użytkowników**

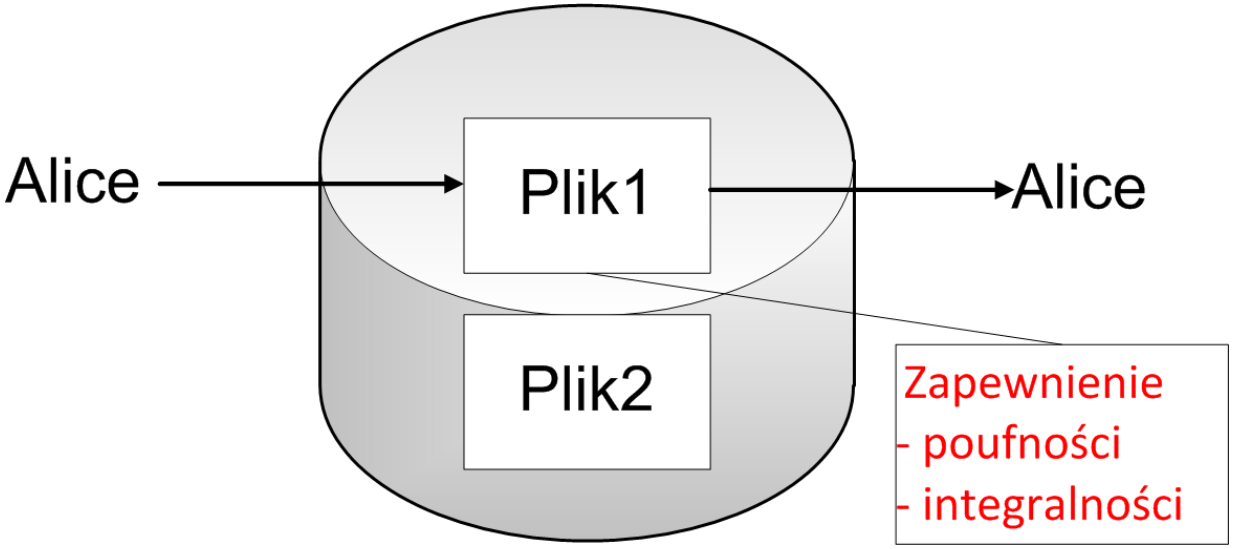
… i wiele innych

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



- Dwie podstawowe części:
  - Protokół „handshake” (uzgodnienie): **Ustanowienie współdzielonego tajnego klucza z zastosowaniem kryptografii klucza publicznego**
  - Warstwa „record” (ustalenie formatu przesyłu pakietów): **Transmisja danych z zastosowaniem tajnego klucza symetrycznego**
    Zapewnienie poufności i integralności danych

Ochrona danych dyskowych



Problem szyfrowania dysków jest analogiczny do chronionej komunikacji, kiedy Alice wysyła wiadomość do Alice

Tryby użycia klucza

Jednorazowe użycie klucza: (one time key)

- Klucz jest jednorazowo zastosowany do zaszyfrowania wiadomości
  - np. szyfrowanie poczty: nowy klucz jest generowany dla każdej wiadomości

Wielokrotne używanie klucza: (many time key)

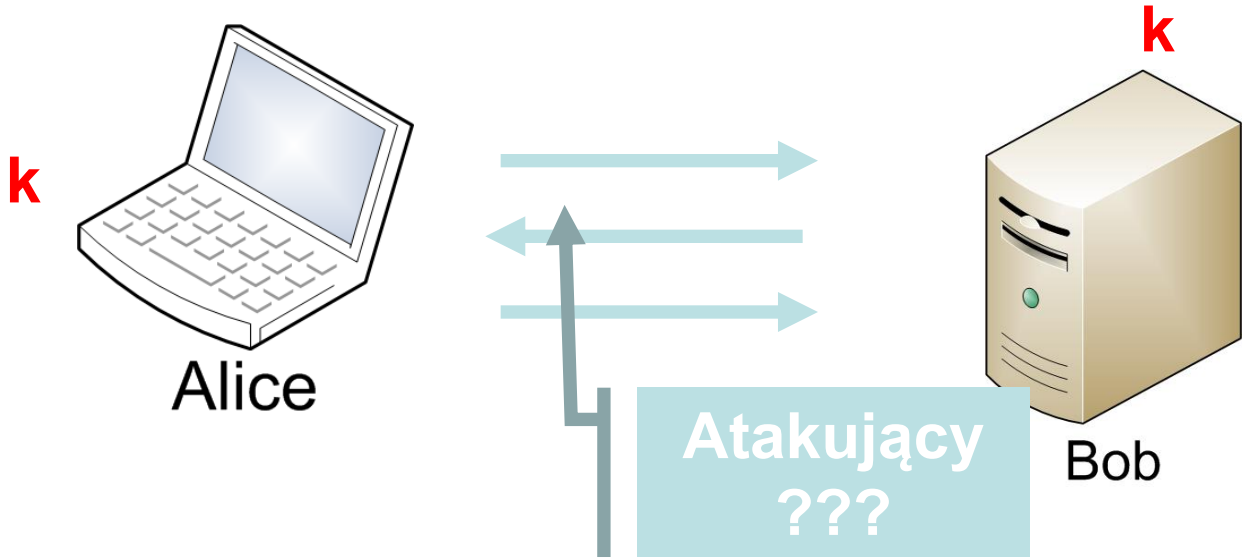
- Ten sam klucz jest stosowany do szyfrowania wielu wiadomości
  - np. szyfrowanie plików: ten sam klucz służy do szyfrowania wielu plików
- W takim przypadku będziemy potrzebowali dodatkowych zabiegów, żeby zapewnić bezpieczeństwo

Uwagi na tym etapie

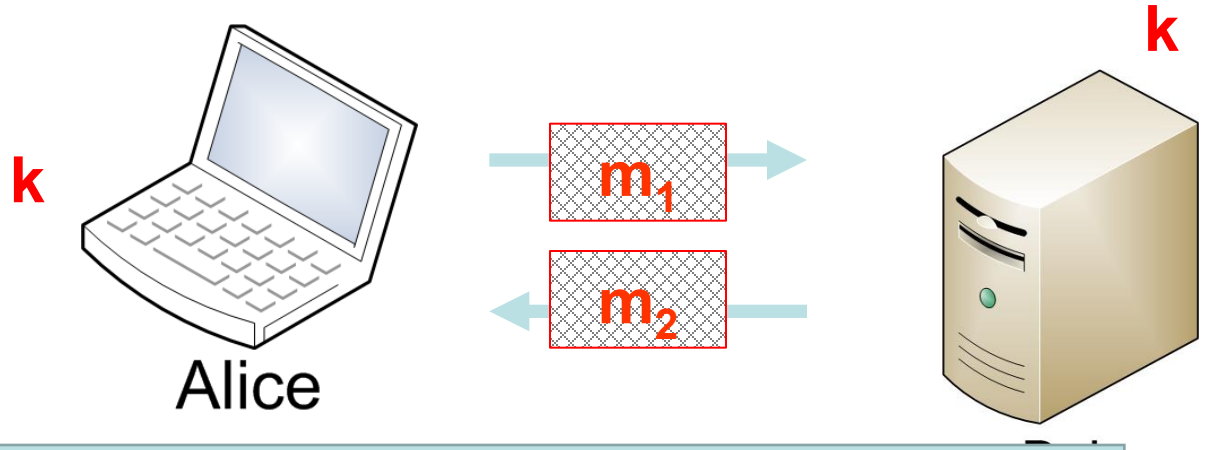
- Kryptografia jest:
  - Doskonałym narzędziem do ochrony informacji w komputerach
  - Podstawą do stworzenia wielu mechanizmów ochrony przesyłu informacji i przechowywania danych
- Kryptografia nie jest:
  - Rozwiązaniem wszystkich problemów ochrony (np. błędy oprogramowania, ataki stosujące soscjotechniki)
  - Zabezpieczeniem, jeśli jest stosowana w niewłaściwy sposób (np. WEP)
  - Czymś, co trzeba wymyślić samemu
    - Jest bardzo wiele przykładów złamanych systemów szyfrowania powstałych at-hoc

Jądro kryptografii

Ustalenie tajnego klucza:



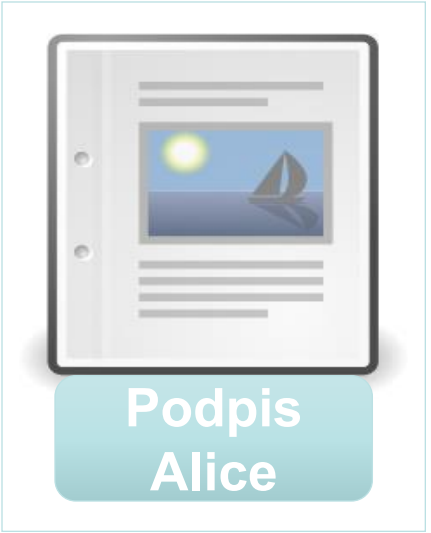
Chroniona komunikacja



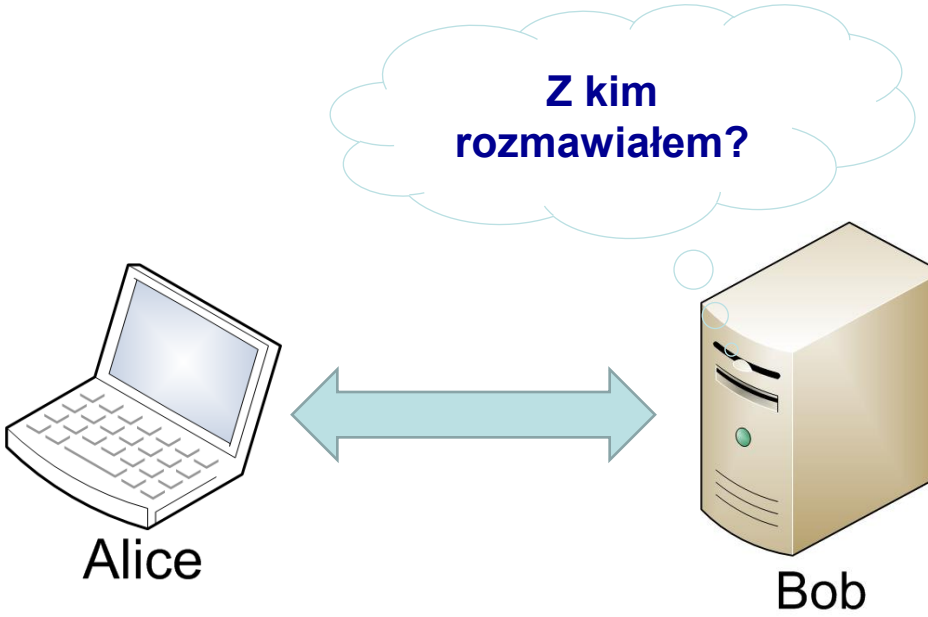
Zachowanie poufności i integralności

Kryptografia umożliwia więcej… (1)

- Podpis elektroniczny



- Anonimowa komunikacja (mix net)



Kryptografia umożliwia więcej… (2)

- Anonimowa **cyfrowa** waluta
  - Czy mogę wydać „cyfrową monetę” z zachowaniem anonimowości?
  - Jak zapobiec podwójnemu płaceniu tą samą monetą?

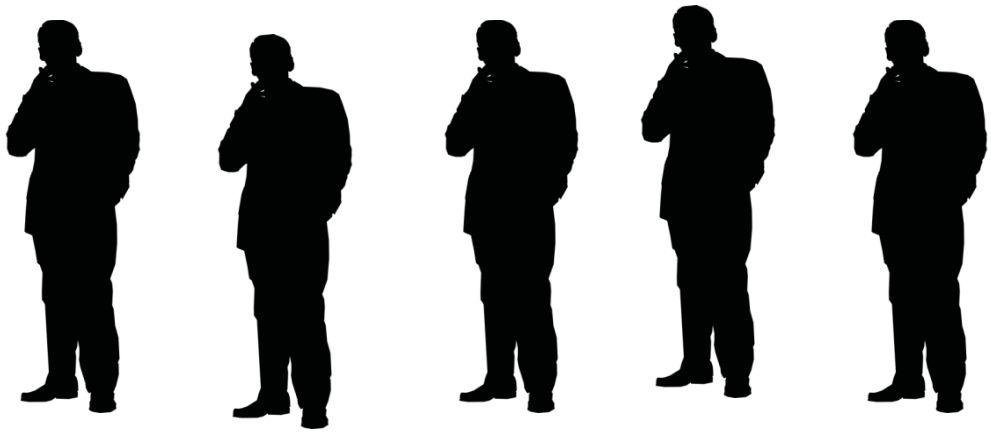
Alice

Internet
anonimowa komunikacja

Kto to był?

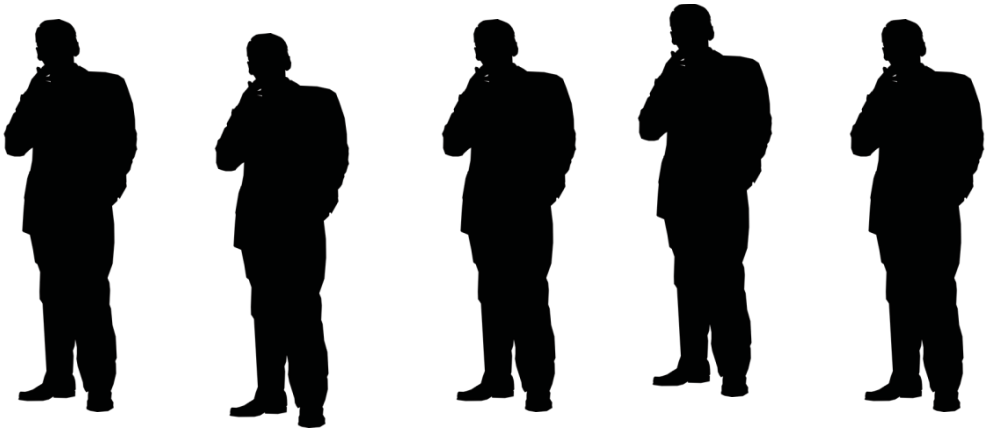
Protokoły

- Wybory



Protokoły

- Bezpieczne obliczenia wielostronne



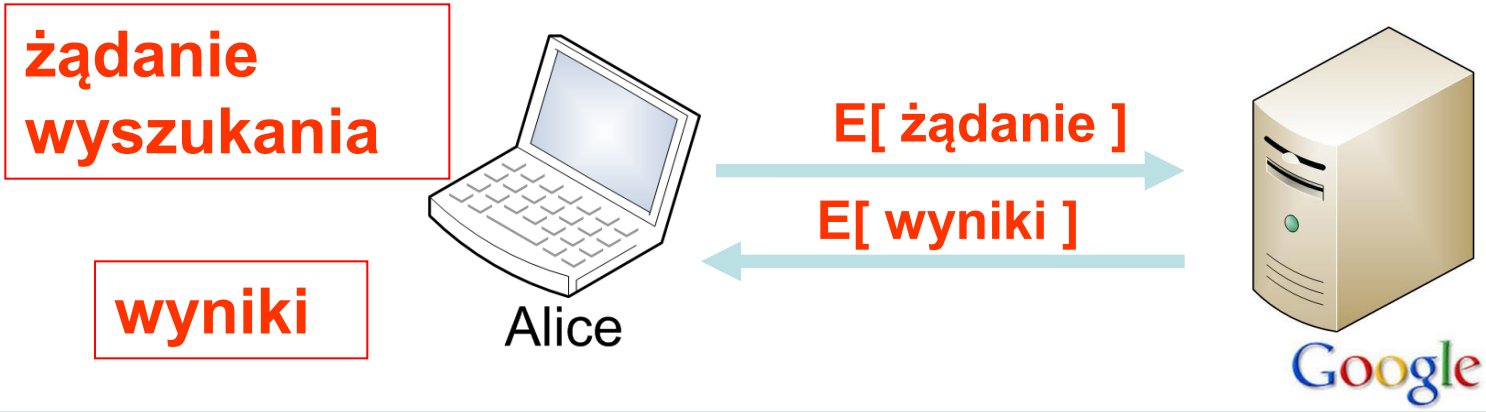
Zaufany urząd

Cel: obliczyć $f(x_1, x_2, x_3, x_4)$

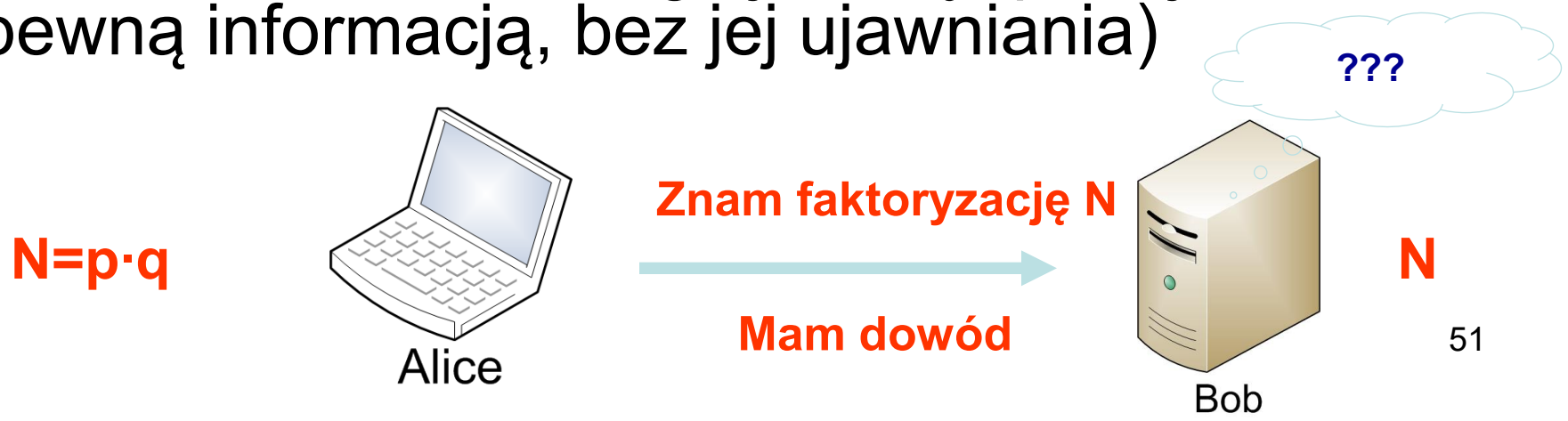
“Twierdzenie:” cokolwiek może być wykonane z zastosowaniem zaufanego urzędu może być wykonane bez niego

Magia kryptografii

- Prywatne zlecenia zdalnych obliczeń



- Dowód z wiedzą zerową (jedna ze stron potrafi udowodnić drugiej, że dysponuje pewną informacją, bez jej ujawniania)



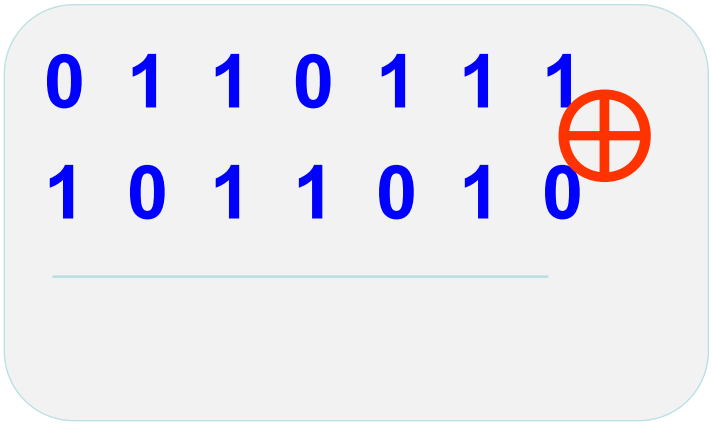
Precyzyjna nauka

Trzy kroki w kryptografii:

- Precyzyjny model zagrożenia
- Propozycja konstrukcji
- Udowodnienie, że złamanie konstrukcji pod względem zagrożenia jest bardzo trudnym problemem

Podstawowy aparat matematyczny

Operacja bitowa XOR

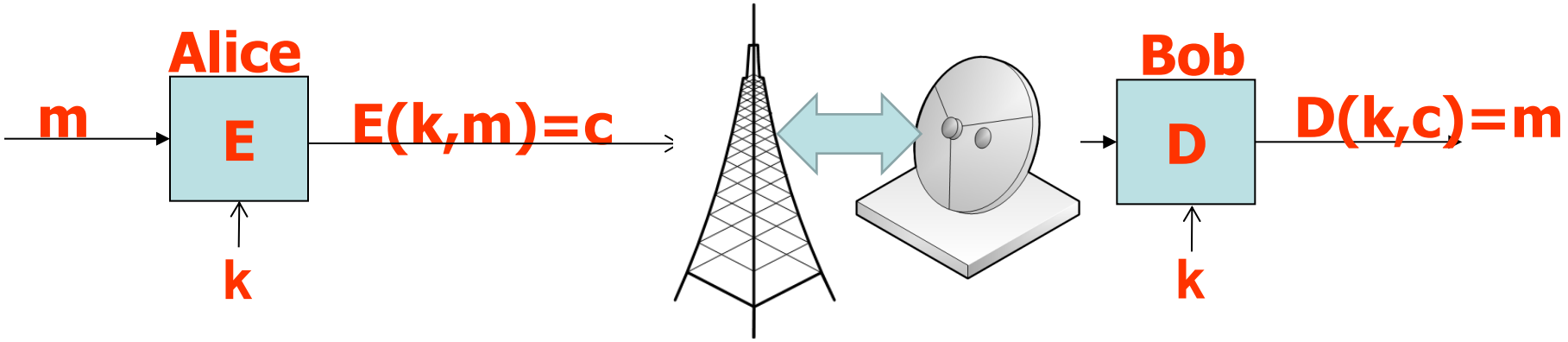


Wejście		WYJŚCIE
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Ważna właściwość funkcji XOR

- (Twierdzenie, nieformalnie)
  Jeśli klucz szyfrujący jest ciągiem losowym, to wykonanie operacji $\oplus$ (XOR) na dowolnym ciągu bitowym i tym kluczu daje dalej ciągiem losowym.

Schemat kryptografii symetrycznej



E, D: algorytm szyfrujący/deszyfrujący
k: tajny symetryczny klucz (np. 128 bitowy)
m, c: wiadomość, zaszyfrowana wiadomość

- Algorytm szyfrowania jest upubliczniony
- Nie używamy własnych szyfrów
- Jedynym tajnym elementem jest klucz

Szyfr z kluczem jednorazowym (One Time Pad) (Vernam 1917)

- Pierwszy przykład „bezpiecznego” szyfru.
- Przestrzeń wiadomości ($\mathcal{M}$) i szyfrów ($\mathcal{C}$) jest n-bitowym ciągiem.
- Klucz jest losowym cięgiem bitów o długości wiadomości.

Definicja szyfru z kluczem jednorazowym

- C = E(k, m) = k ⊕ m
- D(k, m) = k ⊕ c

msg: 0 1 1 0 1 1 1

key: 1 0 1 1 0 1 0 ⊕

CT:

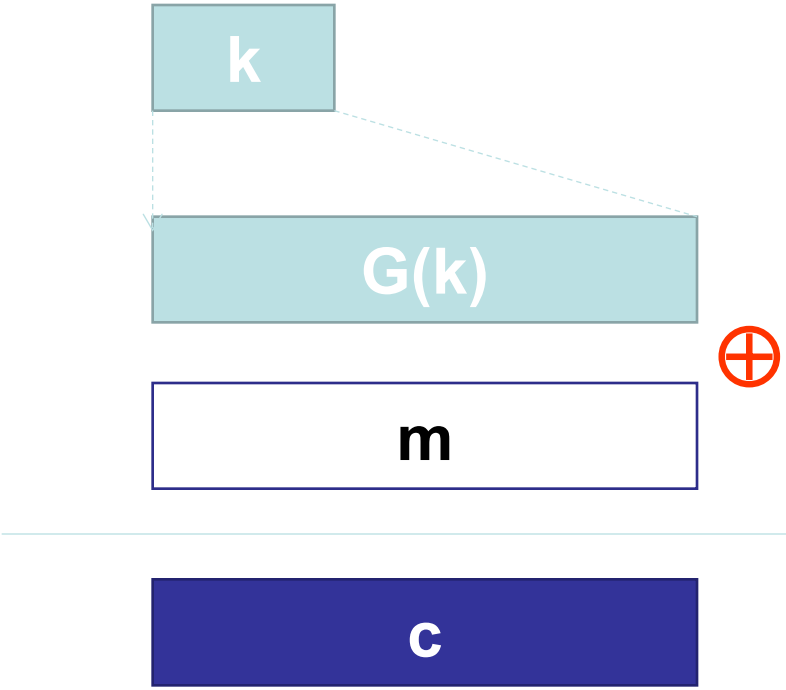
Szyfr strumieniowy: przekształcenie pomysłu na szyfr z kluczem jednorazowym na coś praktycznego

- Pomysł: należy zastąpić „losowy” klucz kluczem generowanym przez algorytm „pseudolosowy”.
- Do generowania kluczy służą generatory liczb pseudolosowych (**PRG – pseudo-random generator**): algorytm, który na podstawie pewnego ciągu zero-jedynkowego (**ziarna**) generuje znacznie dłuższy ciąg mający charakter pseudolosowy.
- Ciąg musi być efektywnie „wyliczalny” i generowany przez deterministyczny algorytm.
- Wygenerowany ciąg musi „wyglądać” jak losowy.

Szyfr strumieniowy zasada działania

$$C = E(k,m) = m \oplus G(k)$$

$$D(k,c) = c \oplus G(k)$$



Współczesne szyfry:

Salasa 20, Cha-Cha 20

Atak 1 na szyfr z kluczem jednorazowym

- Zastosowanie dwa razy tego samego klucza nie jest bezpieczne!!!

$$C_1 \leftarrow m_1 \oplus PRG(k)$$

$$C_2 \leftarrow m_2 \oplus PRG(k)$$

- Atakujący wykonuje:

$$\mathbf{C_1 \oplus C_2 \rightarrow}$$

- Jest wystarczająca liczba powtarzających się znaków, żeby:

$$\mathbf{m_1 \oplus m_2 \rightarrow m_1 , m_2}$$

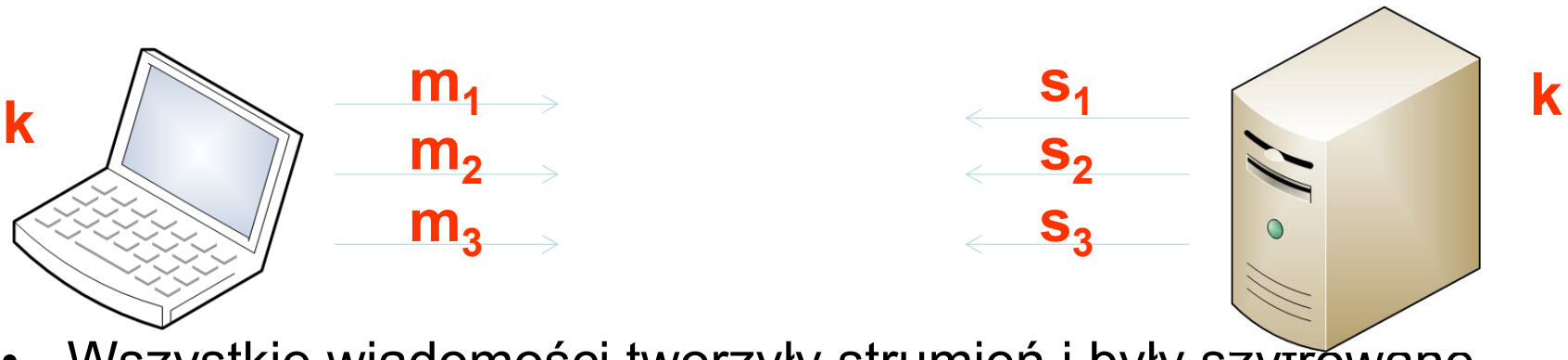
- **WNIOSEK: w szyfrze z kluczem jednorazowym i w szyfrach strumieniowych nie wolno dwa razy zastosować tego samego klucza!!!**

Rzeczywiste przykłady (1)

- Projekt Venona (Związek Radziecki, 1941-1946).
  - szyfrowanie z kluczem jednorazowym
  - klucz był uzyskiwany przez notowanie rzutem kostką
  - ponieważ było to pracochłonne, to używano tego samego klucza do kodowania wielu wiadomości
  - Rząd Stanów Zjednoczonych odczytał około 3000 zaszyfrowanych wiadomości

Rzeczywiste przykłady (2)

- MS-PPTP (Point to Point Tuneling Protocol Windows NT)



- Wszystkie wiadomości tworzyły strumień i były szyfrowane kluczem: $[m_1 \| m_2 \| m_3 \| \ldots] \oplus PRG(k)$
- Niestety odpowiedzi na wiadomości były szyfrowane tym samym kluczem: $[s_1 \| s_2 \| s_3 \| \ldots] \oplus PRG(k)$
- Taki system można zaatakować tak jak na slajdzie nr 15.
- **Wniosek: Przy szyfrowaniu komunikacji Klient-Serwer i Serwer-Klient potrzeba stosować różne klucze**
- **Klucz do komunikacji jest wtedy parą kluczy i obie strony muszą znać oba klucze.**

Rzeczywiste przykłady (3)

802.11b WEP:

- Ramka jest szyfrowana szyfrem strumieniowym z uzgodnionym kluczem k
- Klucz każdej ramki uzyskuje się łącząc go z 24 bitowym ciągiem IV
- Można sobie wyobrazić, że ten ciąg rozpoczyna się od 0 i co ramkę jest zwiększany o jeden
- Pomysł gwarantował zmianę klucza co ramkę. Odbiorca znał klucz i dostawał wraz z ramką wartość IV
- Problemem okazała się długość IV: jest tylko 2^{24} IV, w przybliżeniu 16 milionów i potem następuje reset IV.
- **Problem 1:** W dostatecznie długim strumieniu ramek są więc dwie zaszyfrowane tak samo
- **Problem 2:** Po resecie **niektórych** kart bezprzewodowej komunikacja startowała od nowa z IV == 0

Rzeczywiste przykłady (4) – należy unikać „powiązanych ze sobą kluczy”

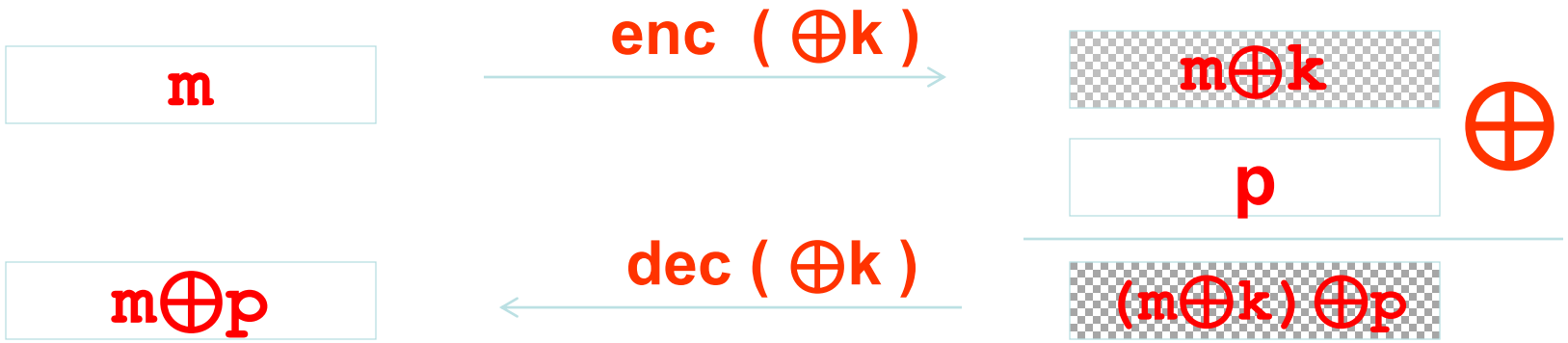
802.11b WEP:

- Problem 3: kolejne klucze uzyskuje się przez połączenie stałego klucza i wartości IV zwiększonej o 1.
  - Klucze są do siebie podobne
- Problem 4: zastosowany algorytm do generowania liczb pseudolosowych RC4 był źle stosowany (udany atak opublikowany 2001 roku)

Podsumowanie ataków na szyfry, w których wielokrotnie użyto tego samego klucza:

- **W szyfrowaniu strumieniowym klucz może być użyty tylko raz!!!**
  - Transmisja danych w sieci: należy negocjować klucze dla każdej sesji (np. TLS)
  - Szyfrowanie dysków: najczęściej nie stosuje się do tego szyfru strumieniowego

Atak 2: brak integralności (1)



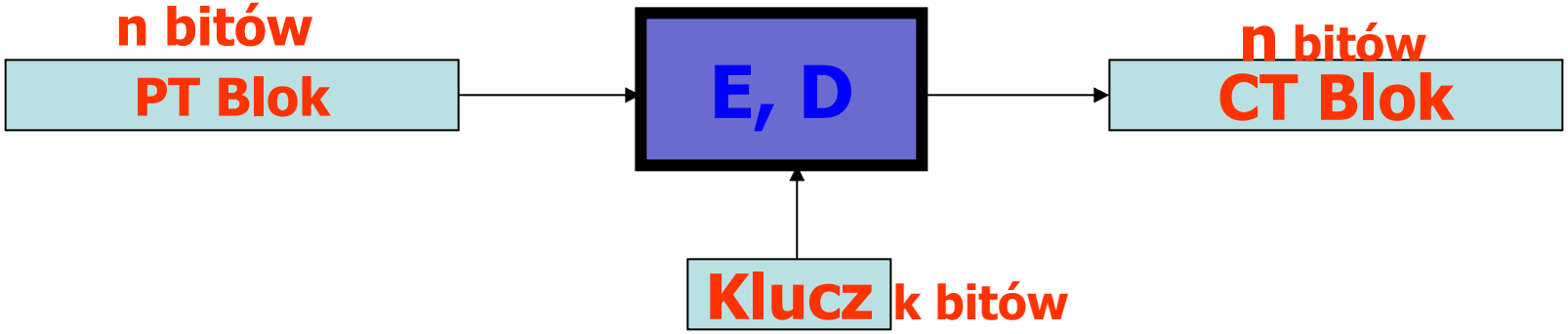
- Modyfikacja wiadomości zaszyfrowanej jest **niewykryta** i ma **przewidywalny wpływ na treść wiadomości**.

Wydajność (biblioteka Crypto++ 5.6.0)

AMD Opteron, 2.2 GHz (Linux)

	PRG	Speed (MB/sec)
	RC4	126
eStream	Salsa20/12	643
	Sosemanuk	727

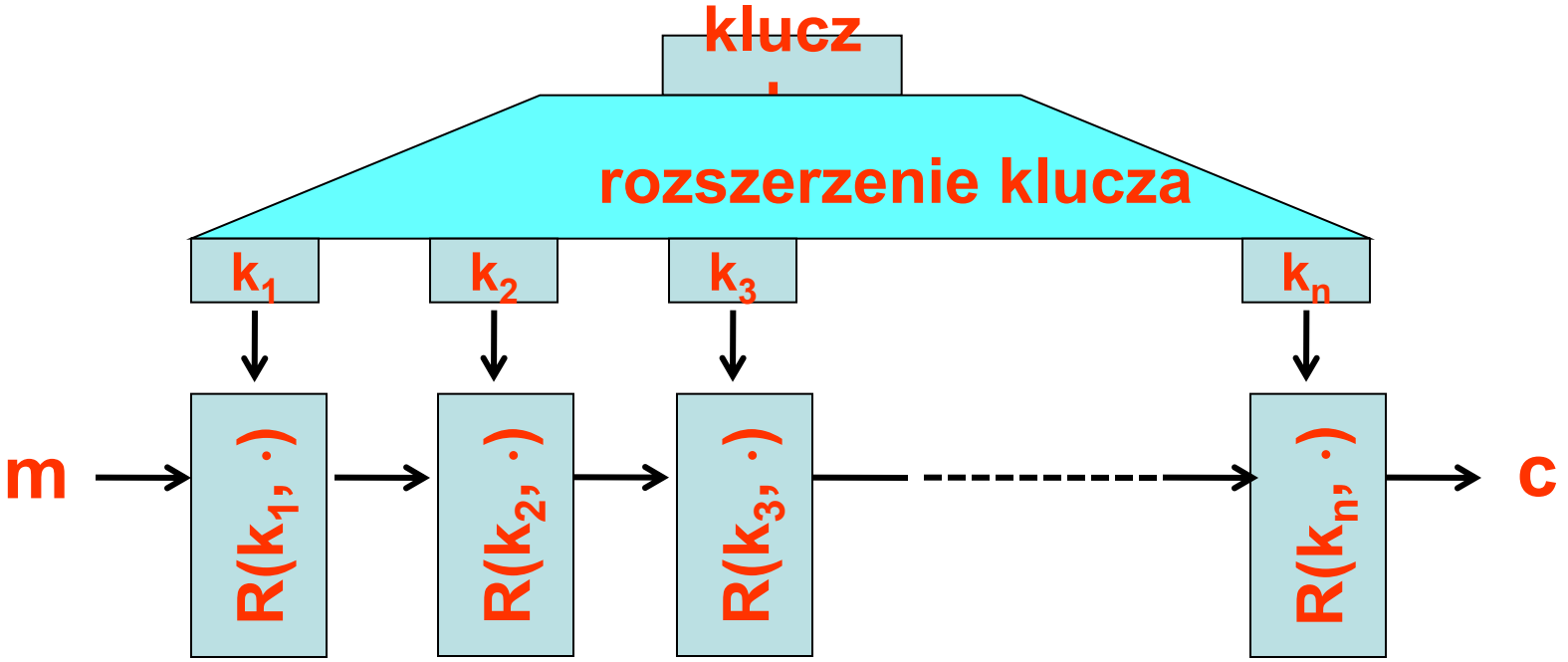
Szyfry blokowe – zasada działania



Popularne przykładowe algorytmy:

1. **3DES: n= 64 bitów, k = 168 bitów**
2. **AES: n=128 bitów, k = 128, 192, 256 bitów**

Szyfry blokowe są budowane z zastosowaniem iteracji



R(k,m) jest nazywana funkcją rundy

dla 3DES (n=48), dla AES-128 (n=10)

Uwagi o bezpieczeństwie - nieformalnie

- Wykonywanie poszczególnych faz szyfru blokowego (funkcje rund) można nazwać generowaniem kolejnych pseudolosowych permutacji (permutacja - ustawianie elementów zbioru w pewnej kolejności)
- Bezpieczeństwo szyfrów blokowych opiera się na tym, że wygenerowane permutacje muszą być nierozróżnialne od ciągów losowych.

Porównanie wydajności Crypto++ 5.6.0 [Wei Dai]

AMD Opteron, 2.2 GHz (Linux)

	Szyfr	Blok/roz. **klucza**	Szybkość (MB/sec)
strumieniowy	RC4		126
	Salsa20/12		643
	Sosemanuk		727
blokowy	3DES	64/168	13
	AES-128	128/128	109

Historia szyfru AES

- 1997: NIST (National Institute of Standards and Technology) ogłasza konkurs na nowy standard szyfrowania blokowego

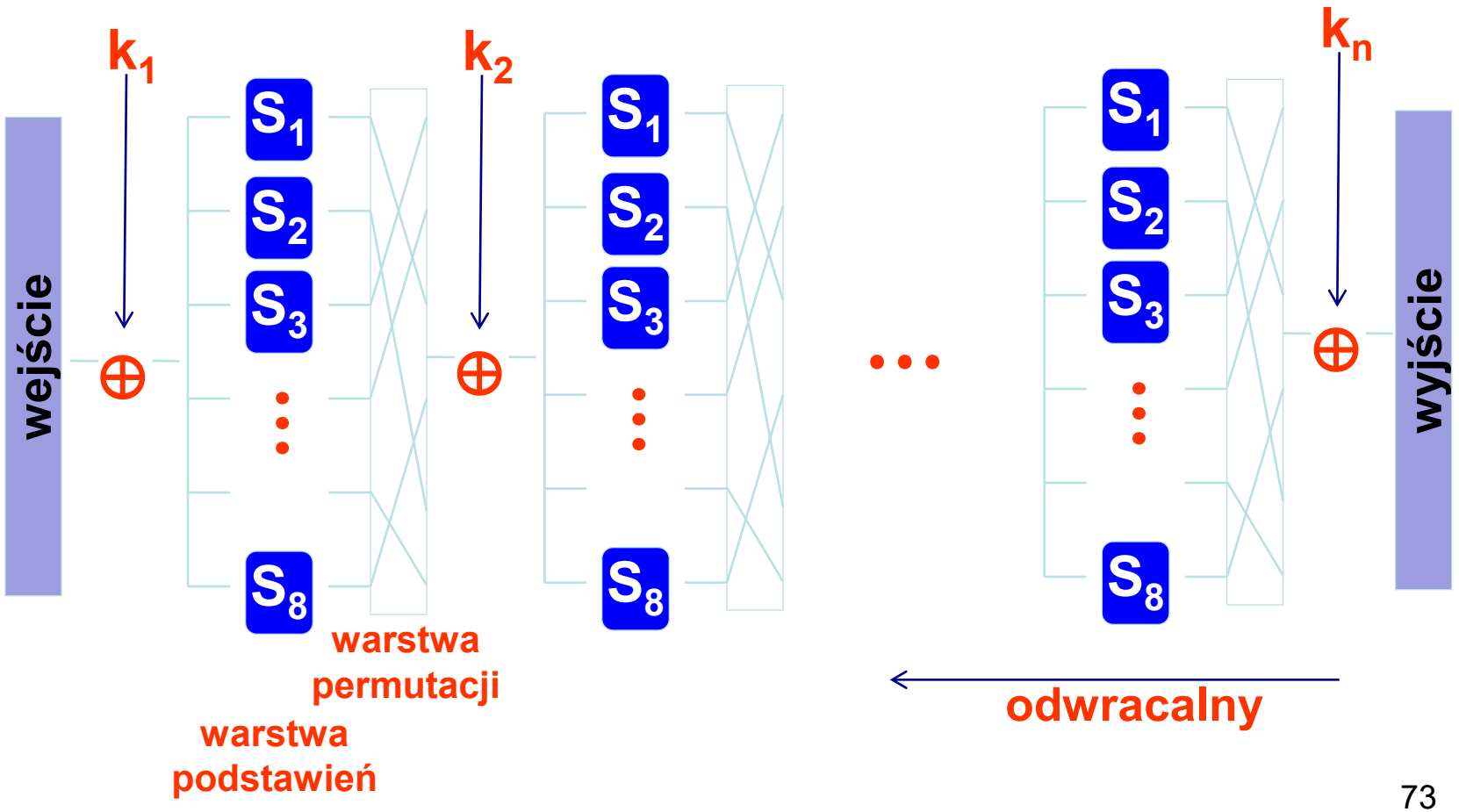
- 1998: 15 zgłoszeń.

- 1999: NIST wybiera 5 finalistów

- 2000: NIST wybiera szyfr Rijndael jako AES (Advanced Encryption Standard) (zaprojektowany w Belgii)

Rozmiary kluczy: 128, 192, 256 bitów.
Rozmiary bloków: 128 bitów

AES jest siecią podstawieniowo-permutacyjną



AES w implementacjach sprzętowych

Instrukcje AES w procesorach Intel Westmere (Core i3, Core i5, Core i7, Pentium, Celeron i Xeon):

- **aesenc, aesenclast**: wykonaj jedną rundę AES
  128-bit rejestry: xmm1=stan, xmm2=klucz rundy
  aesenc xmm1, xmm2; przekaż wynik do rej. xmm1
- **aeskeygenassist**: wykonuje rozszerzenie klucza AES
- Intel twierdzi, że daje to 14 – krotne przyspieszenie w działaniu biblioteki OpenSSL na tym samym sprzęcie

Podobne instrukcje zaimplementowano w architekturze AMD Bulldozer

Atak siłowy w celu uzyskania klucza szyfru blokowego

- Zostało wykazane matematycznie, że dysponując dwiema parami: wiadomość/szyfr można dokonać ataku siłowego.
- Przeprowadzenie ataku siłowego oznacza więc odkrycie klucza na podstawie co najmniej dwóch par wiadomość – zaszyfrowana wiadomość
- Dokonuje się tego przez podawanie na wejście wszystkich możliwych kluczy.
- Problem jest do rozwiązania. Stawianym pytaniem jest w **jakim czasie** i **przy pomocy jakich środków** można tego dokonać.

Pełne przeszukiwanie klucza dla szyfrów blokowych

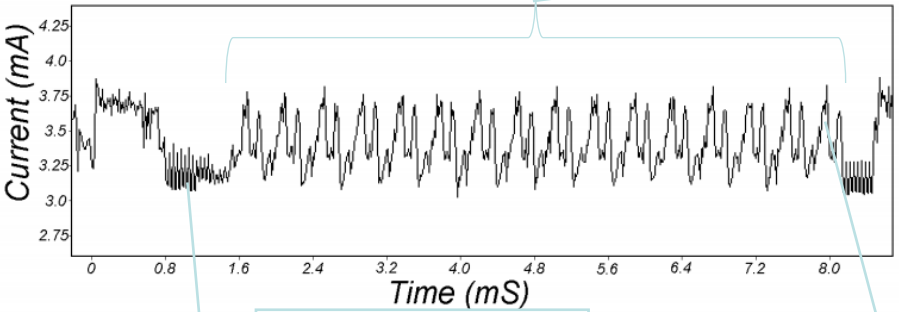
Dla AES-128: przy posiadaniu dwóch par klucz/szyfrogram, prawdopodobieństwo znalezienia klucza ≈ $1 - 1/2^{128}$

⇒ wystarczy dwie pary wiadomość - szyfrogram, żeby znaleźć klucz

Ataki na implementację

1. Ataki przez boczny kanał:
   - Pomiar **czasu** podczas pracy algorytmu,
   - Pomiar **mocy** podczas pracy algorytmu

smartcard



16 rund szyfrowania

[Kocher, Jaffe, Jun, 1998]

Początkowa permutacja

Końcowa permutacja

2. Ataki na błędy:
   - Spowodowanie błędu w ostatniej rundzie ujawnia ukryty klucz

⇒ Proszę nie **implementować** struktur kryptograficznych samodzielnie …
Należy posługiwać się gotowymi bibliotekami/urządzeniami, których kod i implementacja były opublikowane i poddane atakom i te ataki były opublikowane

Siłowy kwantowy atak na szyfry blokowe

Mając m, c=E(k,m) definiujemy

$$f(k) = \begin{cases} 1 & \text{jeśli } E(k,m) = c \\ 0 & \text{w przeciwnym wypadku} \end{cases}$$

Funkja z dziedziną K → wszystkie klucze

Algorytm Grover'a

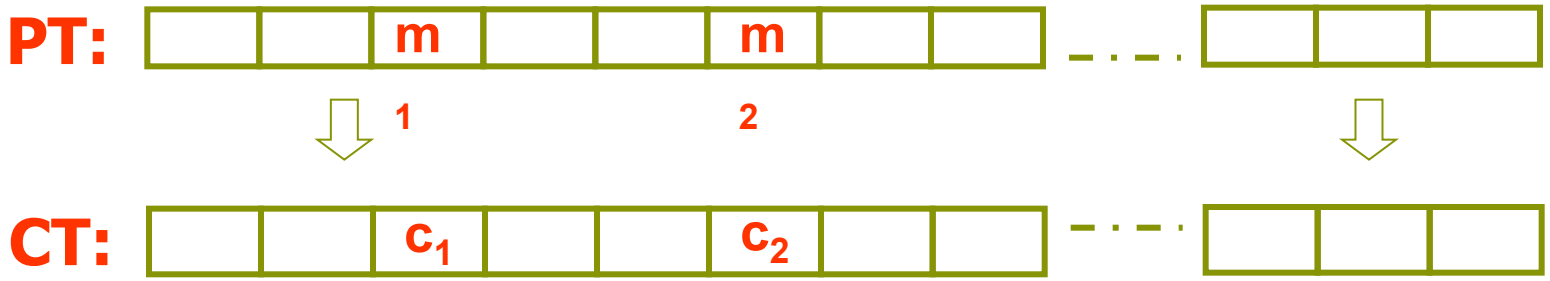
⇒ komputer kwantowy może znaleźć k w czasie $O(|K|^{1/2})$

dla DES: czas $\approx 2^{28}$, dla AES-128: czas $\approx 2^{64}$

jeśli do użytku wejdą komputery kwantowe będziemy musieli stosować klucze 256-bitowe (np. AES-256)

Niewłaściwe użycie PRP

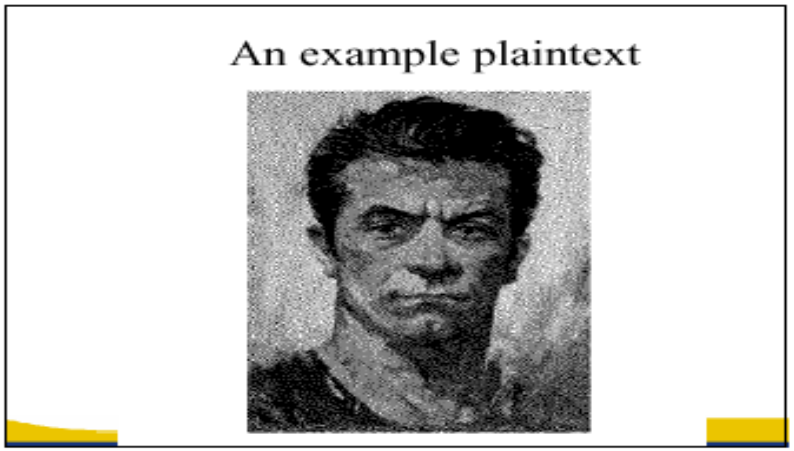
Elektroniczna **Książka Kodowa** (ECB):
(ang. Electronic Code Book)

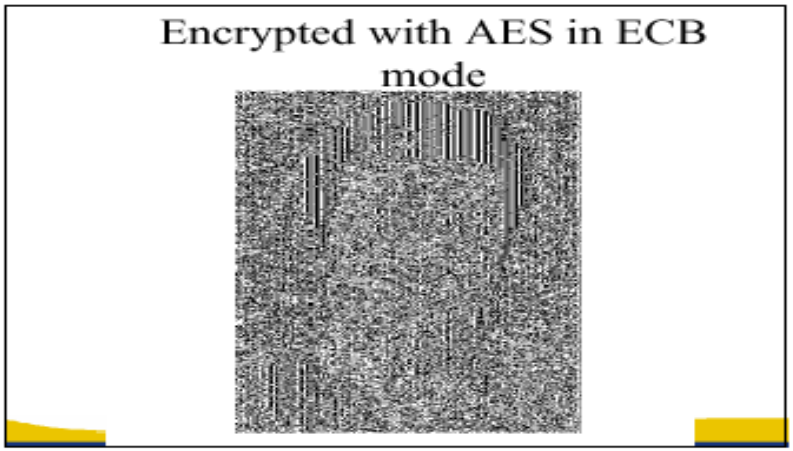


Problem:

– **jeśli** $m_1=m_2$ to $c_1=c_2$

Ilustracja na obrazie

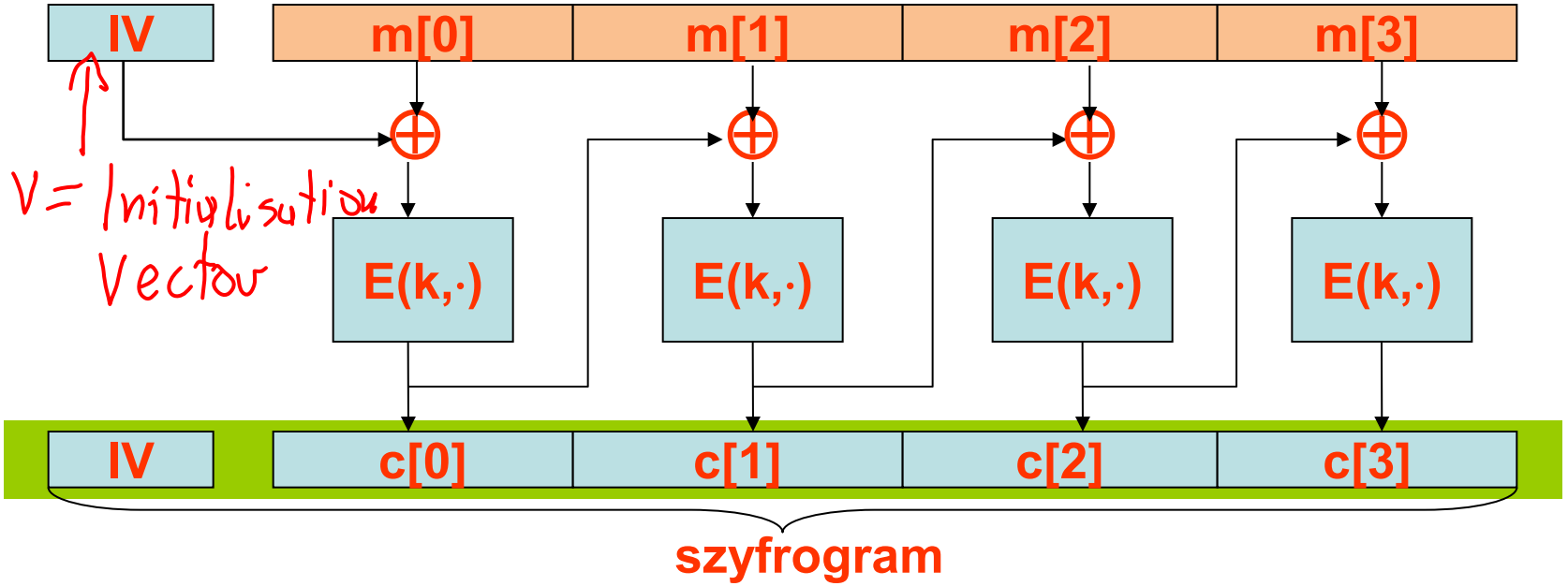




(Udostępnione przez B. Preneel)

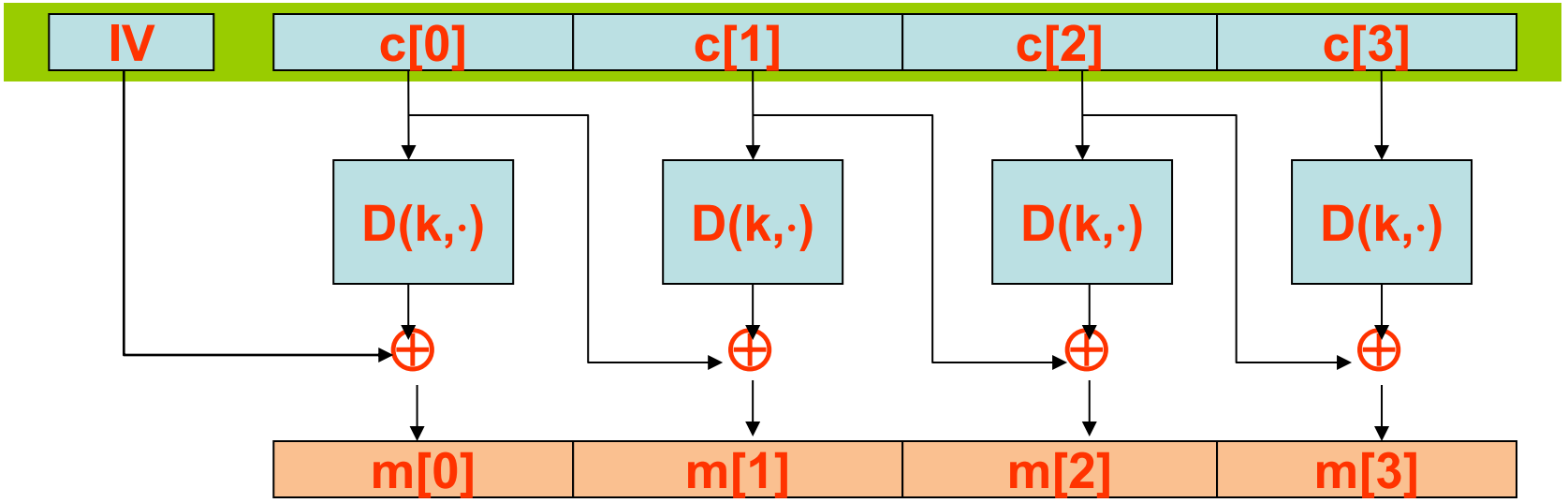
Konstrukcja I: CBC z losowym IV

Niech (E,D) **będzie** PRP. $E_{CBC}(k,m)$: wybierz **losowe** IV∈X i wykonuj:



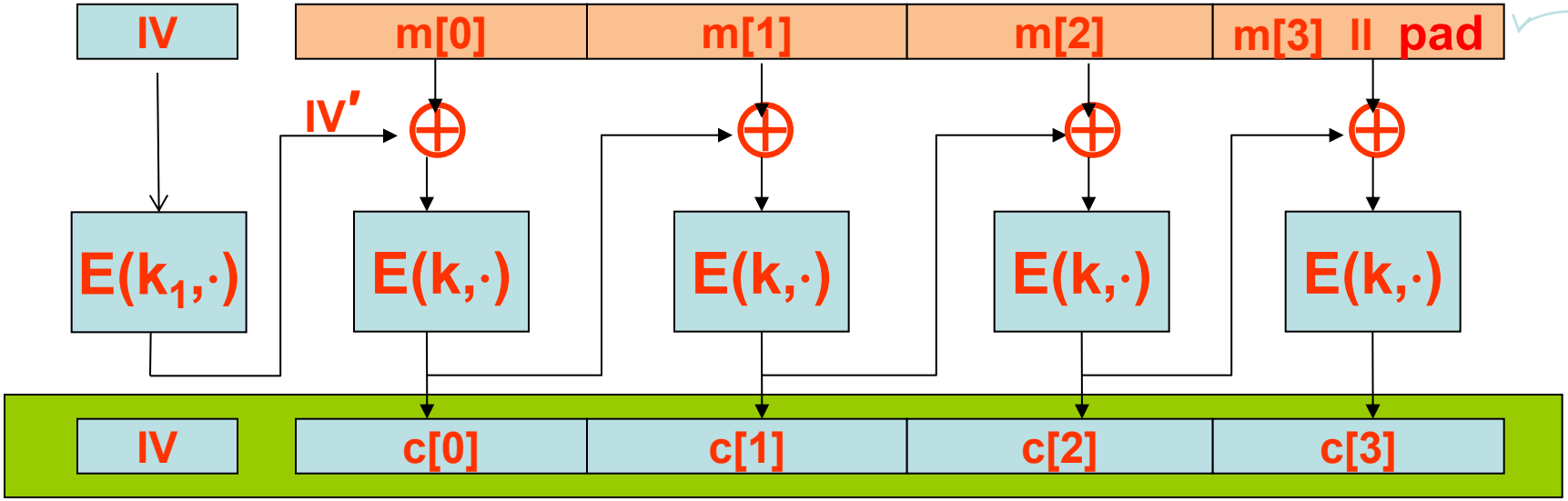
„Obwód” deszyfrujący

Symbolicznie: $c[0] = E(k, IV \oplus m[0])$ $\Rightarrow$ $m[0] = D(k, c[0]) \oplus IV$



- Można udowodnić, że metoda CBC jest semantycznie bezpieczna i odporna na atak z wybranym tekstem jawnym (CPA), jeśli nie przekroczymy pewnej ilości bloków w łańcuchu
- Dla AES: po 2^{48} blokach należy zmienić klucz
- Dla 3DES po 2^{16} blokach należy zmienić klucz

Techniczna uwaga do CBC - padding



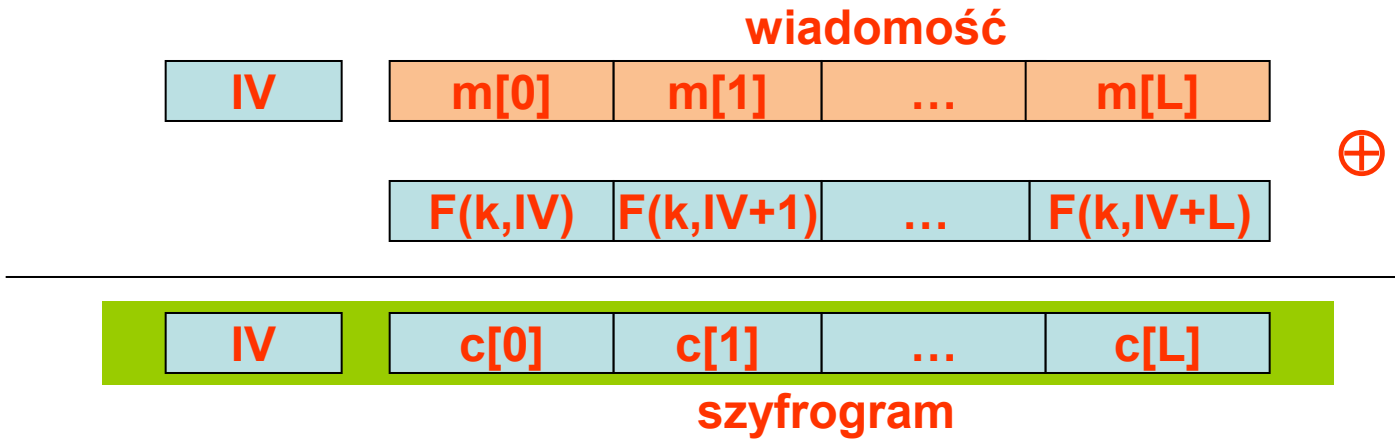
TLS: dla n>0, n bajtowy pad
jeśli nie potrzeba paddingu,
dodaj sztuczny blok.

Usuwane podczas deszyfrowania

Konstrukcja II: losowy tryb licznikowy

Niech $F: K \times \{0,1\}^n \longrightarrow \{0,1\}^n$ będzie bezpieczną PRF.

E(k,m): wybierz losowy $IV \in \{0,1\}^n$ i wykonuj:



uwaga: możliwe do zrównoleglenia (w przeciwieństwie do CBC)

Integralność wiadomości: MAC (ang. Message Authentication Code)

Generuj tag:
tag ← S(k, m)

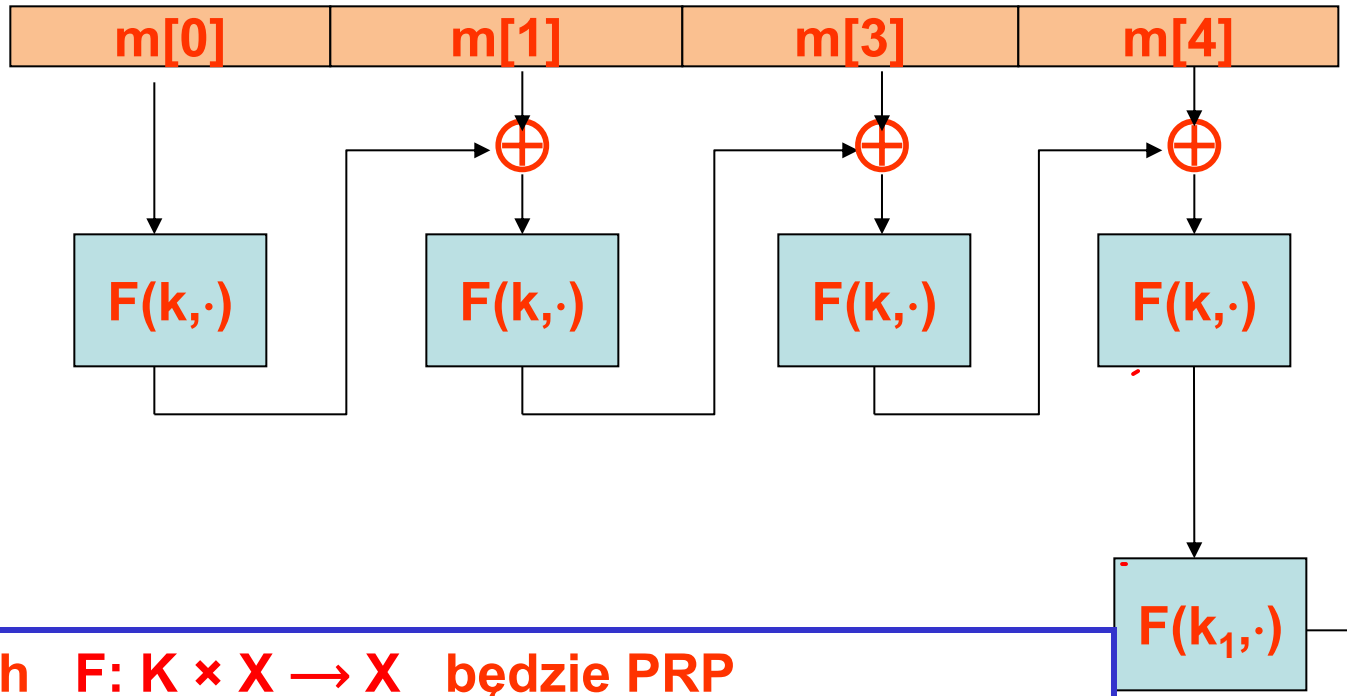
Weryfikuj tag:
V(k, m, tag) ? = 'tak'
lub
'Nie'

Definicja: **MAC** I = (S,V) zdefiniowany na trójce zbiorów (K,M,T) jest parą algorytmów:

- S(k,m) zwraca t ze zbioru T
- V(k,m,t) zwraca `**T**ak' lub `**N**ie'

Szyfrowany CBC-MAC

(ECBC)



$X^{\leq L} = \bigcup_{i=1}^{L} X^i$

⇓

system akceptuje różne dł. wiad.

Niech $F: K \times X \longrightarrow X$ **będzie PRP**

Definiujemy nowy PRF $F_{ECBC} : K^2 \times X^{\leq L} \longrightarrow X$

Wyniki analiz matematycznych

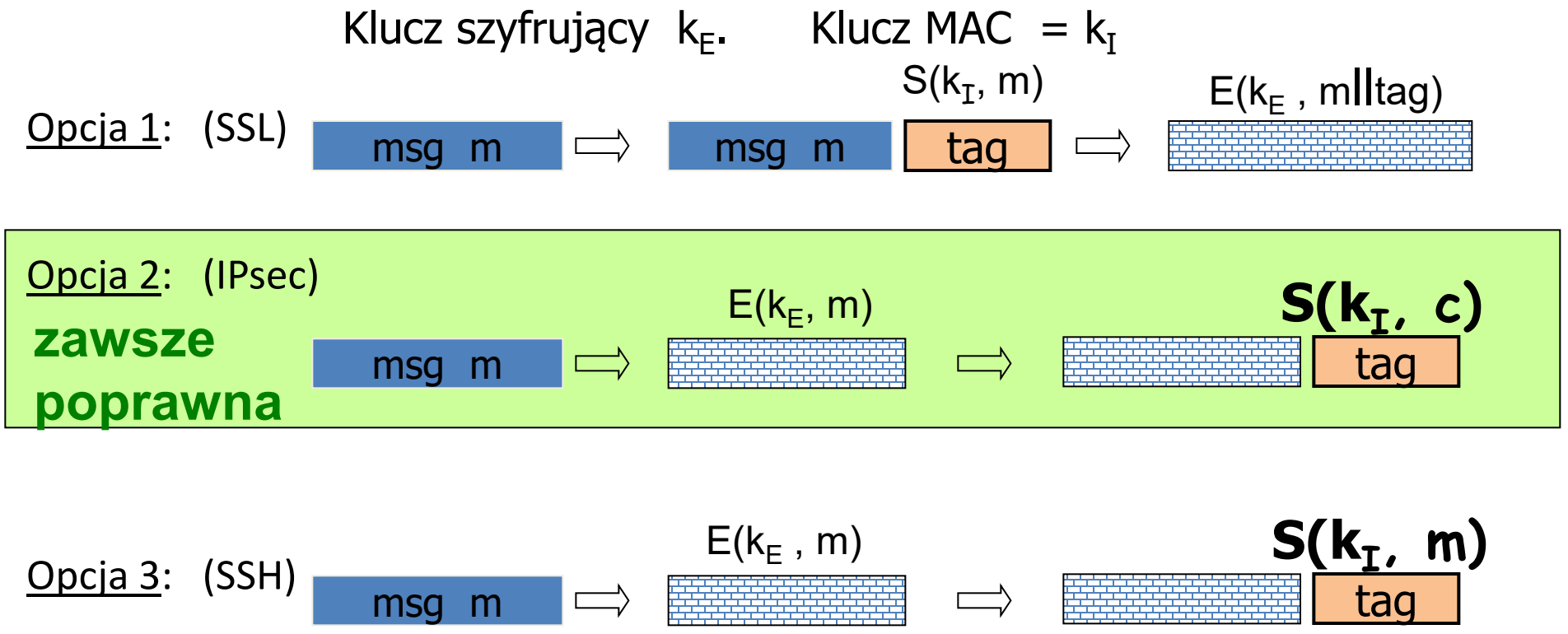
- Po podpisaniu
  $|X|^{1/2}$ wiadomości z zastosowaniem ECBC-MAC lub
  $|K|^{1/2}$ wiadomości z zastosowaniem NMAC

tak skonstruowane MAC przestają być bezpieczne

(taka zależność wynika z paradoksu dnia urodzin, por. wyk. nr 1)

- W konsekwencji:
  - Jeśli stosujemy AES-128, to $|X| = 2^{128}$,
    wtedy po 2^{48} wiadomości trzeba zmienić klucz
  - Jeśli stosujemy 3DES, to $|X| = 2^{64}$,
    wtedy po 2^{16} wiadomości trzeba zmienić klucz

Łączenie MAC i szyfrowania (CCA)

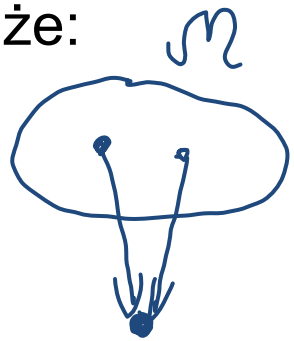


Odporność na kolizje

Niech H: M $\rightarrow$T będzie funkcją skrótu (hash) (|M| >> |T|)

Kolizją dla H jest para wiadomości m_0 , $m_1 \in M$ takich, że:

$$H(m_0) = H(m_1) \quad i \quad m_0 \neq m_1$$

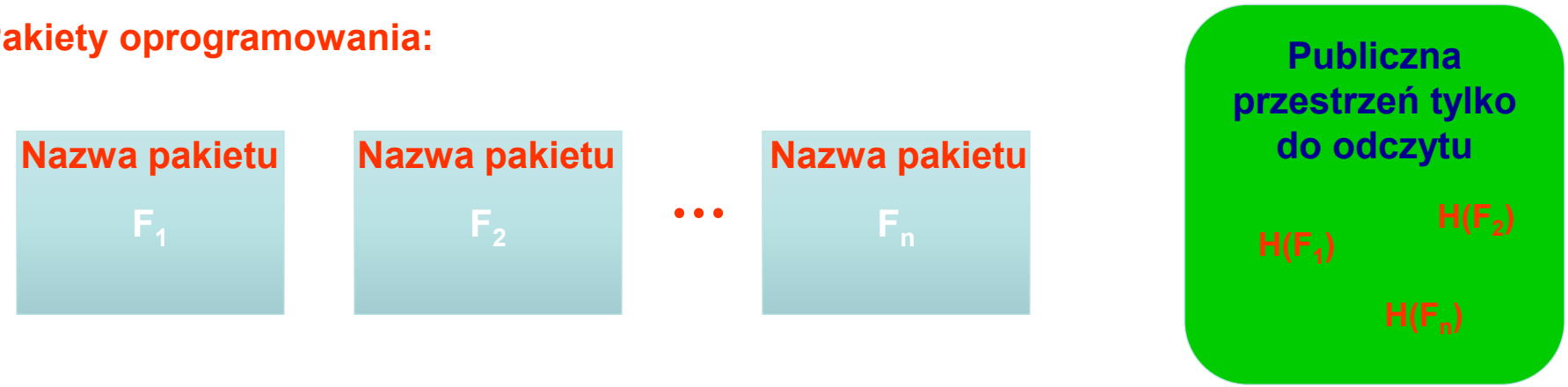


Funkcja H jest **odporna na kolizje** jeśli prawdopodobieństwo wygenerowania identycznych funkcji skrótu dla różnych wiadomości jest pomijalne.
Uwaga: do wygenerowania kolizji mają zostać stosowane konkretne istniejące algorytmy. Nie interesuje nas fakt istnienia kolizji, tylko pokazania algorytmu, który taką kolizję wyliczy.

Przykładowo: SHA-256 (zwraca 256 bitów)

Ochrona integralności plików z zastosowaniem C.R. (ang. Collision Resistance) hash

Pakiety oprogramowania:



Kiedy użytkownik pobiera plik, może sprawdzić, czy plik zachował integralność

H jest odporna na kolizje ⇒
atakujący nie może zmodyfikować pakietu danych bez wykrycia tego faktu

Nie potrzeba wtedy kluczy (tzw. mechanizm publicznej weryfikacji), ale przestrzeń, z której następuje pobieranie pakietów musi mieć atrybut „tylko do odczytu”

Przykładowe odporne na kolizje funkcje hash: Crypto++ 5.6.0 [Wei Dai]

AMD Opteron, 2.2 GHz (Linux)

	funkcja	długość skrótu(bity)	szybkość (MB/sec)	uogólniony czas ataku
Standardy NIST	SHA-1	160	153	2^{80}
	SHA-256	256	111	2^{128}
	SHA-512	512	99	2^{256}
	Whirlpool	512	57	2^{256}

***najlepszy znany algorytm znalezienia kolizji dla SHA-1 wymaga analizy 2^{51} hashów**

Protokół: TSL Record Protocol(TSL 1.2)

$k_{b \to s}$, $k_{s \to b}$

$k_{b \to s}$, $k_{s \to b}$

Jednokierunkowe klucze: $k_{b \to s}$ i $k_{s \to b}$

Szyfrowanie utrzymujące stan:

- Każda strona utrzymuje dwa 64-bitowe liczniki: $ctr_{b \to s}$, $ctr_{s \to b}$
- Liczniki są zerowane na początek sesji. Zwiększane o 1 dla każdego rekordu.
- Cel: zapobieganie atakom powtórzeniowym

Rekord TLS: szyfrowanie (CBC AES-128, HMAC-SHA1)

$k_{b \to s} = (k_{mac}, k_{enc})$

Strona przeglądarki **$enc(k_{b \to s}, data, ctr_{b \to s})$:**

krok 1: $tag \leftarrow S(k_{mac}, [++ctr_{b \to s} \,||\, header \,||\, data])$

krok 2: pad [header || data || tag] do rozmiaru bloku **AES**

krok 3: szyfrowanie CBC z k_{enc} **i** nowym losowym IV

krok 4: **dołączenie nagłówka**

Rekord TLS: rozszyfrowywanie

(CBC AES-128, HMAC-SHA1)

Strona Serwera: **dec($k_{b\to s}$, record, $ctr_{b\to s}$)** :

krok 1: rozszyfrowanie schematu CBC z zastoswaniem k_{enc}

krok 2: sprawdzenie formatu padu:
wysłanie bad_record_mac jeśli się nie zgadza

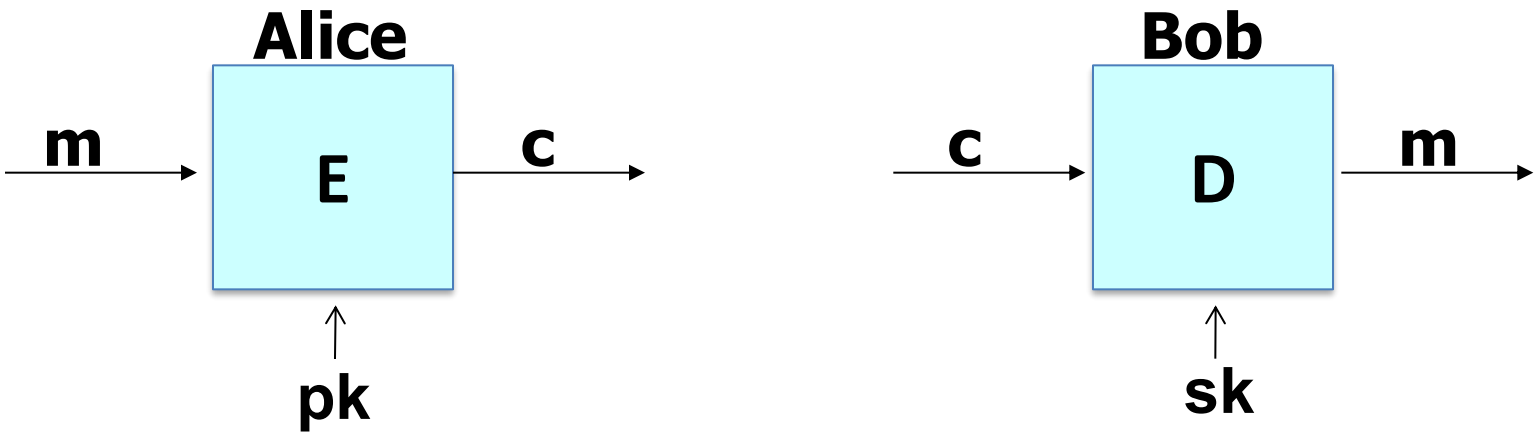
krok 3: sprawdzenie tagu bloku [++$ctr_{b\to s}$ ll header ll data]
wysłanie bad_record_mac jeśli się nie zgadza

Zapewnia szyfrowanie z uwierzytelnieniem

(pod warunkiem, że nie dostarcza żadnych dodatkowych informacji podczas odszyfrowywania)

Kryptografia klucza publicznego

Bob: generuje (PK, SK) i przekazuje PK do Alice



Zapadkowe permutacje

Trzy algorytmy: (G, F, F^{-1})

- G: zwraca pk, sk. pk definiuje funkcję $F(pk, \cdot): X \rightarrow X$
- F(pk, x): oblicza funkcję dla argumentu x
- $F^{-1}(sk, y)$: odwraca funkcję z argumentem y używając sk

Bezpieczna zapadkowa permutacja:

Funkcja $F(pk, \cdot)$ jest jednokierunkowa jeśli nie znamy sk

Wyprowadzenie kryptografii klucza publicznego z funkcji zapadkowych

- (G, F, F^{-1}): bezpieczna funkcja **zapadkowa** $X \longrightarrow Y$
- (E_s, D_s) : **system szyfrowania symetrycznego z uwierzytelnieniem** (K,M,C)
- $H: X \longrightarrow K$ funkcja hash

E(pk, m) : $\overset{R}{\leftarrow}$

$x \overset{R}{\leftarrow} X$, $y \leftarrow F(pk, x)$

$k \leftarrow H(x)$, $c \leftarrow E_s(k, m)$

wyjście (y, c)

D(sk, (y,c)) :

$x \leftarrow F^{-1}(sk, y)$,

$k \leftarrow H(x)$, $m \leftarrow D_s(k, c)$

wyjście m

Pułapkowa permutacja RSA (1)

G(): wybierz losowe l. pierwsze p, q ≈1024 bits.
Ustal **N=p*q**.
wybierz l. całkowite **e , d** takie, że **e·d = 1 (mod φ(N))**
zwróć pk = (N, e) , sk = (N, d)

F(pk, x): $\mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$; **RSA(x) = x^e** (w **Z_N**)

F^{-1}(sk, y) = y^d ; $y^d = RSA(x)^d = x^{ed} = x^{k\varphi(N)+1} = \left(x^{\varphi(N)}\right)^k \cdot x = x$

Przypomnienie: system klucza publicznego RSA (standard ISO)

(E_s, D_s): system szyfrowania symetrycznego z uwierzytelnieniem.

H: $Z_N \rightarrow K$ gdzie K jest przestrzenią kluczy (E_s, D_s)

- **G**(): Generuj parametry RSA: pk = (N,e), sk = (N,d)
- **E**(pk, m): (1) wybierz losowy x w Z_N

  (2) $y \leftarrow RSA(x) = x^e$, $k \leftarrow H(x)$

  (3) zwróć $(y , E_s(k,m))$
- **D**(sk, (y, c)): zwróć $D_s(H(RSA^{-1}(y)) , c)$

Długości kluczy

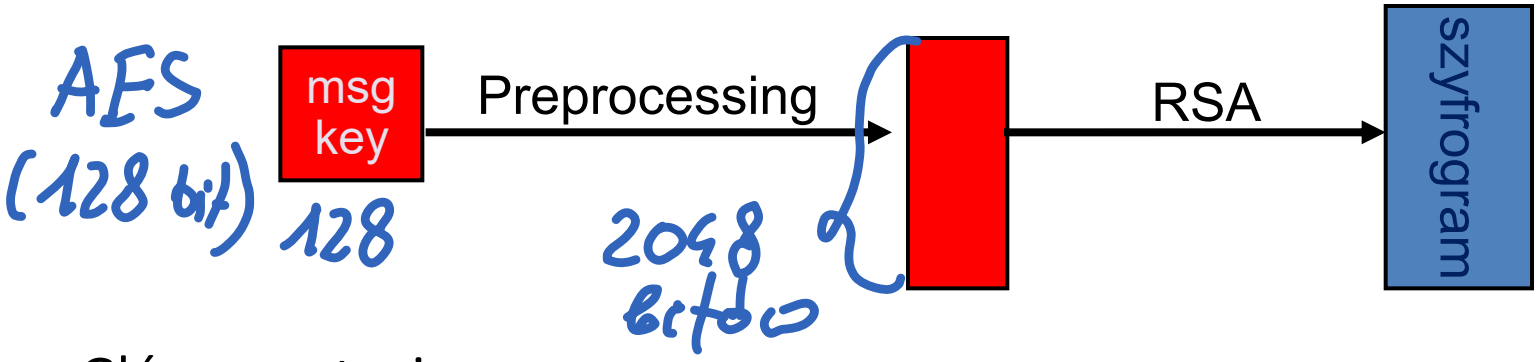
Bezpieczeństwo systemów z kluczem publicznym powinno być porównywalne do bezpieczeństwa szyfrowania zastosowaniem klucza symetrycznego::

Rozmiar klucza	RSA Rozmiar modułu
80 bity	1024 bity
128 bity	3072 bity
256 bity (AES)	**15360** bitów

Kryptografia klucza publicznego w praktyce

Nie stosujemy RSA do bezpośredniego szyfrowania danych czy kluczy!

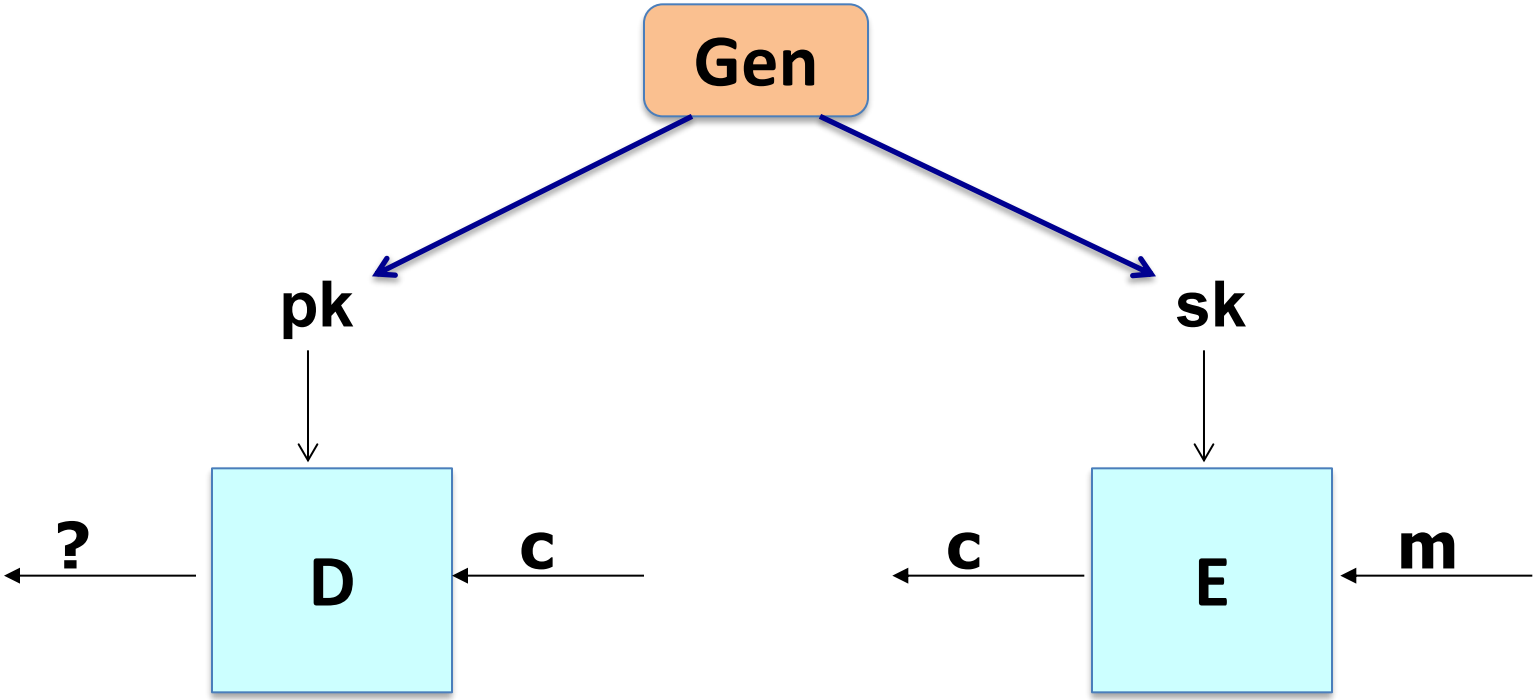
RSA w praktyce (ponieważ standard ISO nie jest często stosowany) :



Główne pytania:

- Jak powinien być przeprowadzony „preprocessing”?
- Czy możemy udowodnić bezpieczeństwo takiego systemu?

Co by się stało, gdybyśmy odwrócili przebieg szyfrowania z kluczem publicznym?



Znaczenie podpisu elektronicznego

- Weryfikowalność podpisu zależy do możliwości niezawodnego i możliwego do zweryfikowania udostepnienia klucza publicznego.
- Taka procedura jest kosztowna, także ze względów organizacyjnych, ale zwykle wykonywana jest tylko raz, a potem każdy łatwo może uzyskać klucz i zweryfikować podpis.
- Okazuje się, że właśnie schemat podpisu elektronicznego sam w sobie jest doskonałym mechanizmem dystrybucji kluczy publicznych.
- Tak więc publikowanie kluczy publicznych odbywa się z zastosowaniem podpisów elektronicznych i jest podstawą do tzw. Infrastruktury Klucza Publicznego (PKI – Public Key Infrastructure)

Definicja podpisu elektronicznego

Podpis elektroniczny: (Gen, Sign, Vrfy)

- Gen: algorytm generacji pary kluczy (pk i sk)
- Sign: algorytm podpisujący

  $(m, ds) \leftarrow Sign(sk,m)$
- Vrfy: algorytm weryfikujący

  Vrfy(pk, m, ds)

  Zwraca 1, gdy podpis jest prawidłowo zweryfikowany, 0 gdy weryfikacja się nie udała.

Dla wszystkich legalnych wiadomości zachodzi:

Vrfy(pk,m,Sign(sk,m)) = 1.

Paradygmat Hash-and-Sign (oblicz hash, potem podpisuj)

- Zaszyfrowanie całych, zwłaszcza długich wiadomości jest kosztowne obliczeniowo
- Popularne jest rozwiązanie „hybrydowe”:
  - Najpierw stosuje się funkcję hash do „streszczenia” wiadomości
  - Potem szyfrowanie przeprowadza się tylko na skrócie/streszczeniu wiadomości (ang. digest)
  - Rozwiązanie jest odpowiednikiem schematu hash-and-MAC

Certyfikat

- Załóżmy, że Charlie wygenerował parę kluczy (pk_C, sk_C)
- Załóżmy też, że Bob wygenerował parę kluczy (pk_B, sk_B)
- Załóżmy, że Charlie wie, że pk_B jest kluczem publicznym Bob'a
- Wtedy Charlie może obliczyć podpis:
  $cert_{C \to B}$ = Sign[sk_C,(„To jest klucz Boba", pk_B)]
  i odesłać ten podpis Bob'owi.
- Nazywamy $cert_{C \to B}$ certyfikatem dla klucza publicznego Bob'a wystawionym przez Charliego.
- W praktyce certyfikat powinien jednoznacznie identyfikować Bob'a, czyli zawierać jego imię i nazwisko, email, stronę WWW i inne atrybuty

Weryfikacja klucza publicznego na podstawie certyfikatu

- Teraz Bob chce skomunikować się z inną osobą/instytucją, np. Alice, która już zna pk_C.
- Może on wysłać do Alice wiadomość (pk_B,$cert_{C\rightarrow B}$), która może zweryfikować, że pk_B rzeczywiście należy do Bob'a, o czym poświadcza pk_C.
- Jeśli sprawdzenie podpisu się udało, to Alice wie, że Charlie podpisał klucz Bob'a
- Jeśli Alice ufa Charlie'mu, to może zaakceptować pk_B jako legalny klucz
- Cała wymiana potwierdzająca pk_B może się odbyć w publicznym i niechronionym kanale komunikacyjnym
- Dopóki Charlie (klucz prywatny Charlie'go) nie zostanie „skompromitowany" tak zweryfikowane klucze są uznawane za legalne i powiązane z daną osobą/instytucją.

Kilka pominiętych detali

- Skąd Alice zna pk_C i mu ufa?
- Skąd Charlie wie, że pk_B należy do Boba?
- Jak Alice decyduje, czy ufać Charlie'mu?

- Pełna specyfikacja, jak rozwiązać pokazane problemy opisana jest w dokumentach dotyczących Infrastruktury Klucza Publiczengo (PKI – Public Key Infrastructure)