Wprowadzenie do zagadnień bezpieczeństwa i kryptografii

Patryk Czarnik

Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski

Bezpieczeństwo sieci komputerowych – MSUI 2010/11

Przedmiot

- 5×2 wykładów, poniedziałki 17:00, sala 4060
- Slajdy dostępne w sieci:
  `http://www.mimuw.edu.pl/~czarnik/bsk`
- Egzamin testowy (głównie pytania wielokrotnego wyboru)

Wykładowca

mgr Patryk Czarnik
asystent @ Instytut Informatyki UW
email: `czarnik` at `mimuw.edu.pl`
pokój: 4580

Materiały

Plan wykładu i wiele materiałów odziedziczone po poprzednich edycjach od Aleksego Schuberta i Sławomira Leszczyńskiego.

Przedmiot

- 5×2 wykładów, poniedziałki 17:00, sala 4060
- Slajdy dostępne w sieci:
  `http://www.mimuw.edu.pl/~czarnik/bsk`
- Egzamin testowy (głównie pytania wielokrotnego wyboru)

Wykładowca

mgr Patryk Czarnik
asystent @ Instytut Informatyki UW
email: `czarnik` at `mimuw.edu.pl`
pokój: 4580

Materiały

Plan wykładu i wiele materiałów odziedziczone po poprzednich edycjach od Aleksego Schuberta i Sławomira Leszczyńskiego.

Przedmiot

- 5×2 wykładów, poniedziałki 17:00, sala 4060
- Slajdy dostępne w sieci:
  `http://www.mimuw.edu.pl/~czarnik/bsk`
- Egzamin testowy (głównie pytania wielokrotnego wyboru)

Wykładowca

mgr Patryk Czarnik
asystent @ Instytut Informatyki UW
email: `czarnik` at `mimuw.edu.pl`
pokój: 4580

Materiały

Plan wykładu i wiele materiałów odziedziczone po poprzednich edycjach od Aleksego Schuberta i Sławomira Leszczyńskiego.

Prawdopodobny plan wykładu

- Wprowadzenie do zagadnień bezpieczeństwa i kryptografii.
- Protokół IPSec.
- Protokół SSL.
- Protokół SSH.
- System Kerberos.
- Ściany ogniowe, maskarada, proxy.
- Bezpieczna poczta.
- SNMP.
- NIS, DHCP.

Materiały dodatkowe

Literatura

- W. Stallings, „Ochrona danych w sieci i intersieci”, WNT 1997
- B. Schneier, „Bezpieczeństwo poczty elektronicznej”, WNT 1996
- J. Stoklosa, T. Bilski, T. Pankowski, „Bezpieczeństwo danych w systemach informatycznych”, PWN, Warszawa, 2001
- ...

WWW

- Wikipedia.org
- Specyfikacje standardów, głównie RFC: `http://www.rfc-editor.org/`
- Artykuły, tutoriale ...

Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepowołanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępnamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepowołanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępnamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepowołanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępnamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepowołanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępnamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepowołanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępnamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepowołanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępnamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

Właściwe podejście do ochrony

Kryptografia i bezpieczne protokoły

- Dają wysoce skuteczne zabezpieczenia:
  - poufność
  - uwierzytelnienie i integralność danych
- Nie dają 100% gwarancji:
  - trudność złamania algorytmów nie dowiedziona
  - zdarzają się odkrycia metod łamania algorytmów kryptograficznych (przykłady: algorytm SHA-1, protokół WEP)
  - ponadto zawsze możliwy (acz kosztowny) atak *brute force*
- Należy stosować aktualne rozwiązania (algorytm, rozmiar klucza)

Uwaga socjologiczna

Stosowanie bezpiecznych rozwiązań nie może być na tyle niewygodne, aby skłaniało użytkowników do ich obchodzenia.

Właściwe podejście do ochrony

Kryptografia i bezpieczne protokoły

- Dają wysoce skuteczne zabezpieczenia:
  - poufność
  - uwierzytelnienie i integralność danych
- Nie dają 100% gwarancji:
  - trudność złamania algorytmów nie dowiedziona
  - zdarzają się odkrycia metod łamania algorytmów kryptograficznych (przykłady: algorytm SHA-1, protokół WEP)
  - ponadto zawsze możliwy (acz kosztowny) atak *brute force*
- Należy stosować aktualne rozwiązania (algorytm, rozmiar klucza)

Uwaga socjologiczna

Stosowanie bezpiecznych rozwiązań nie może być na tyle niewygodne, aby skłaniało użytkowników do ich obchodzenia.

Właściwe podejście do ochrony

Kryptografia i bezpieczne protokoły

- Dają wysoce skuteczne zabezpieczenia:
  - poufność
  - uwierzytelnienie i integralność danych
- Nie dają 100% gwarancji:
  - trudność złamania algorytmów nie dowiedziona
  - zdarzają się odkrycia metod łamania algorytmów kryptograficznych (przykłady: algorytm SHA-1, protokół WEP)
  - ponadto zawsze możliwy (acz kosztowny) atak *brute force*
- Należy stosować aktualne rozwiązania (algorytm, rozmiar klucza)

Uwaga socjologiczna

Stosowanie bezpiecznych rozwiązań nie może być na tyle niewygodne, aby skłaniało użytkowników do ich obchodzenia.

Właściwe podejście do ochrony

Kryptografia i bezpieczne protokoły

- Dają wysoce skuteczne zabezpieczenia:
  - poufność
  - uwierzytelnienie i integralność danych
- Nie dają 100% gwarancji:
  - trudność złamania algorytmów nie dowiedziona
  - zdarzają się odkrycia metod łamania algorytmów kryptograficznych (przykłady: algorytm SHA-1, protokół WEP)
  - ponadto zawsze możliwy (acz kosztowny) atak *brute force*
- Należy stosować aktualne rozwiązania (algorytm, rozmiar klucza)

Uwaga socjologiczna

Stosowanie bezpiecznych rozwiązań nie może być na tyle niewygodne, aby skłaniało użytkowników do ich obchodzenia.

Kryptologia

Nauka o przekazywaniu informacji w sposób zabezpieczony przed niepowołanym dostępem

Podział

kryptografia tworzenie systemów zabezpieczeń

kryptoanaliza łamanie systemów zabezpieczeń

Niektóre założenia kryptografii

- **złożoność** – pewne problemy są trudno rozwiązywalne
- **prawdopodobieństwo** – losowość dostępna w komputerach
- **mechanika kwantowa** – nie możemy dokładnie określić stanu cząstek elementarnych

Kryptologia

Nauka o przekazywaniu informacji w sposób zabezpieczony przed niepowołanym dostępem

Podział

kryptografia	tworzenie systemów zabezpieczeń
kryptoanaliza	łamanie systemów zabezpieczeń

Niektóre założenia kryptografii

- **złożoność** – pewne problemy są trudno rozwiązywalne
- **prawdopodobieństwo** – losowość dostępna w komputerach
- **mechanika kwantowa** – nie możemy dokładnie określić stanu cząstek elementarnych

Kryptologia

Nauka o przekazywaniu informacji w sposób zabezpieczony przed niepowołanym dostępem

Podział

kryptografia tworzenie systemów zabezpieczeń

kryptoanaliza łamanie systemów zabezpieczeń

Niektóre założenia kryptografii

- **złożoność** – pewne problemy są trudno rozwiązywalne
- **prawdopodobieństwo** – losowość dostępna w komputerach
- **mechanika kwantowa** – nie możemy dokładnie określić stanu cząstek elementarnych

Poufność

Do zapewnienia poufności informacji korzystamy z szyfrów.

Podział szyfrów

- Szyfry symetryczne
  - blokowe
  - strumieniowe
- Szyfry asymetryczne (kryptografia klucza publicznego)

Poufność

Do zapewnienia poufności informacji korzystamy z szyfrów.

Podział szyfrów

- Szyfry symetryczne
  - blokowe
  - strumieniowe
- Szyfry asymetryczne (kryptografia klucza publicznego)

Szyfr

Szyfr symetryczny to para funkcji:

- szyfrującej

$$E : K \times M \to C$$

- deszyfrującej

$$D : K \times C \to M$$

taka, że dla każdego klucza k i każdej wiadomości m musi zachodzić

$$D(k, E(k, m)) = m$$

Oczekiwane własności szyfrów

Aby szyfr zapewniał bezpieczeństwo, musimy na funkcję szyfrującą nałożyć dodatkowe ograniczenia. Można je sformalizować na kilka sposobów.

Najmocniejsza własność to **bezpieczeństwo bezwarunkowe**.

Znajomość szyfrogramu nie ujawnia żadnej informacji o odpowiadającym mu tekście jawnym.

W języku bardziej formalnym:

Entropia tekstu jawnego jest taka sama przed jak i po poznaniu szyfrogramu.

Oczekiwane własności szyfrów

Aby szyfr zapewniał bezpieczeństwo, musimy na funkcję szyfrującą nałożyć dodatkowe ograniczenia. Można je sformalizować na kilka sposobów.

Najmocniejsza własność to **bezpieczeństwo bezwarunkowe**.

> Znajomość szyfrogramu nie ujawnia żadnej informacji o odpowiadającym mu tekście jawnym.

W języku bardziej formalnym:

> Entropia tekstu jawnego jest taka sama przed jak i po poznaniu szyfrogramu.

Oczekiwane własności szyfrów

Aby szyfr zapewniał bezpieczeństwo, musimy na funkcję szyfrującą nałożyć dodatkowe ograniczenia. Można je sformalizować na kilka sposobów.

Najmocniejsza własność to **bezpieczeństwo bezwarunkowe**.

> Znajomość szyfrogramu nie ujawnia żadnej informacji o odpowiadającym mu tekście jawnym.

W języku bardziej formalnym:

> Entropia tekstu jawnego jest taka sama przed jak i po poznaniu szyfrogramu.

Własności semanycznego bezpieczeństwa

Słabszymi własnościami są **własności semantycznego bezpieczeństwa**, zwykle definiowane jako **nierozróżnialność ze względu na** atak określonego rodzaju.

Atak kryptologiczny

- Działanie mające na celu złamanie zabezpieczeń kryptograficznych.
- W skrajnych przypadkach umożliwia poznanie tajnego klucza, odczytanie zaszyfrowanej informacji lub podszycie się.
- Niepożądane jest jakiekolwiek osłabienie bezpieczeństwa bezwarunkowego.

Własności semanycznego bezpieczeństwa

Słabszymi własnościami są **własności semantycznego bezpieczeństwa**, zwykle definiowane jako **nierozróżnialność ze względu na** atak określonego rodzaju.

Atak kryptologiczny

- Działanie mające na celu złamanie zabezpieczeń kryptograficznych.
- W skrajnych przypadkach umożliwia poznanie tajnego klucza, odczytanie zaszyfrowanej informacji lub podszycie się.
- Niepożądane jest jakiekolwiek osłabienie bezpieczeństwa bezwarunkowego.

Często rozważane rodzaje ataków

- atak ze znanym szyfrogramem (oczywisty)
  - atak siłowy (*brute force*),
  - analiza statystyczna,
- atak ze znanym tekstem jawnym
  (na ogół łatwy do przeprowadzenia),
- atak z wybranym tekstem jawnym (łatwy w przypadku klucza publicznego)
- atak z wybranym szyfrogramem (wymaga dostępu do maszyny deszyfrującej)

- atak na hasło (słownikowy, „socjalny”),
- atak „człowiek w środku”
  (dotyczy protokołów komunikacyjnych,
  a nie samych algorytmów kryptograficznych)

Atak z wybranym tekstem jawnym

- Atakujący (Bartek) ma dostęp do maszyny szyfrującej, tj. może wielokrotnie wybrać tekst jawny i poznać jego szyfrogram.
- Korzystając z tego stara się uzyskać informacje, które pozwolą mu obniżyć bezpieczeństwo mechanizmu (np. tajny klucz).
- Własność krytyczna dla mechanizmów opartych o klucz publiczny.

Nierozróżnialność zwn. atak z wyb. tekstem jawnym

Po wykonaniu powyższego ataku tak można sprawdzić własność nierozróżnialności:

- Alicja posiada maszynę szyfrującą, Bartek nie posiada (już) do niej dostępu.
- Bartek wysyła do Alicji dwie różne wiadomości m_1 i m_2.
- Alicja rzuca monetą. W zależności od wyniku szyfruje wiadomość m_1 lub m_2 i wysyła szyfrogram do Bartka.
- Bartek próbuje odgadnąć szyfrogram której wiadomości otrzymał.

Jeśli Bartek nie jest w stanie odpowiedzieć na to pytanie z prawdopodobieństwem znacząco większym od $\frac{1}{2}$, własność **nierozróżnialności** zachodzi.

Nierozróżnialność zwn. atak z wyb. tekstem jawnym

Po wykonaniu powyższego ataku tak można sprawdzić własność nierozróżnialności:

- Alicja posiada maszynę szyfrującą, Bartek nie posiada (już) do niej dostępu.
- Bartek wysyła do Alicji dwie różne wiadomości m_1 i m_2.
- Alicja rzuca monetą. W zależności od wyniku szyfruje wiadomość m_1 lub m_2 i wysyła szyfrogram do Bartka.
- Bartek próbuje odgadnąć szyfrogram której wiadomości otrzymał.

Jeśli Bartek nie jest w stanie odpowiedzieć na to pytanie z prawdopodobieństwem znacząco większym od $\frac{1}{2}$, własność **nierozróżnialności** zachodzi.

Atak z wybranym szyfrogramem

- Atakujący (Bartek) ma dostęp do maszyny deszyfrującej, tj. może wielokrotnie wybrać szyfrogram i poznać odpowiadający mu tekst jawny.
- Korzystając z tego stara się uzyskać informacje, które pozwolą mu obniżyć bezpieczeństwo mechanizmu (np. tajny klucz).
- Atakiem tego rodzaju jest także taki, gdy atakujący może odszyfrować tylko niektóre szyfrogramy.

Nierozróżnialność zwn. atak z wyb. szyfrogramem

Po wykonaniu ataku tak można sprawdzić własność nierozróżnialności:

- Alicja posiada maszynę deszyfrującą, Bartek nie posiada (już) do niej dostępu.
- Bartek wysyła do Alicji dwa różne (poprawne) szyfrogramy s_1 i s_2.
- Alicja rzuca monetą. W zależności od wyniku deszyfruje szyfrogram s_1 lub s_2 i wysyła odszyfrowaną wiadomość do Bartka.
- Bartek próbuje odgadnąć wiadomość odpowiadającą któremu szyfrogramowi otrzymał.

Jeśli Bartek nie potrafi odpowiedzieć na to pytanie z prawdopodobieństwem znacząco większym od $\frac{1}{2}$, własność **nierozróżnialności** zachodzi.

Nierozróżnialność zwn. atak z wyb. szyfrogramem

Po wykonaniu ataku tak można sprawdzić własność nierozróżnialności:

- Alicja posiada maszynę deszyfrującą, Bartek nie posiada (już) do niej dostępu.
- Bartek wysyła do Alicji dwa różne (poprawne) szyfrogramy s_1 i s_2.
- Alicja rzuca monetą. W zależności od wyniku deszyfruje szyfrogram s_1 lub s_2 i wysyła odszyfrowaną wiadomość do Bartka.
- Bartek próbuje odgadnąć wiadomość odpowiadającą któremu szyfrogramowi otrzymał.

Jeśli Bartek nie potrafi odpowiedzieć na to pytanie z prawdopodobieństwem znacząco większym od $\frac{1}{2}$, własność **nierozróżnialności** zachodzi.

Klasyfikacja ataków ze względu na adaptywność

Ataki powyższych rodzajów dodatkowo dzieli się ze względu na adaptywność (wynikającą często z czasu dostępu do maszyny).

- **Atak nieadaptywny (*lunchtime attack*)** – atakujący ma uprzednio przygotowany zestaw wiadomości / szyfrogramów.
- **Atak adaptywny (*midnight attack*)** – atakujący może na bieżąco preparować nowe wiadomości / szyfrogramy.

Szyfry symetryczne

- Ten sam klucz używany do szyfrowania jak i deszyfrowania (ewentualnie istnieje łatwy sposób na otrzymanie klucza deszyfrującego z klucza szyfrującego).
- Stosowane od bardzo dawna. Jako przykład jednego z pierwszych szyfrów podaje się szyfr Cezara.
- Działają na zasadzie jak największego zagmatwania.
- Są efektywne:
  - szyfrowanie i deszyfrowanie przebiega bardzo szybko,
  - używane klucze są małe – aktualnie 128–256 bitów.

Szyfry blokowe i strumieniowe

Szyfry symetryczne można podzielić na dwie kategorie:

- **szyfry blokowe** – szyfrowane są bloki danych,
- **szyfry strumieniowe** – szyfrowanie następuje bit po bicie.

Zasady konstrukcji szyfrów blokowych

Schemat Feistel

Najbardziej powszechny schemat budowy szyfru w oparciu o funkcję jednej „rundy”. Idea:

- Na bloku do zaszyfrowania wykonywanych jest sukcesywnie wiele rund tego samego przekształcenia, ale z różnymi kluczami.
- Runda składa się ze złożenia prostych operacji, które same w sobie nie są wystarczająco bezpieczne (np. przekształcenia liniowe, translacje, podstawienia).
- Klucze dla poszczególnych rund są generowane przez ekspansję klucza głównego.

Tryby szyfrowania

- Dodatkowo szyfrowania kolejnych bloków dokonuje się w **trybach** ECB, CBC, CFB, OFB itp.

Zasady konstrukcji szyfrów blokowych

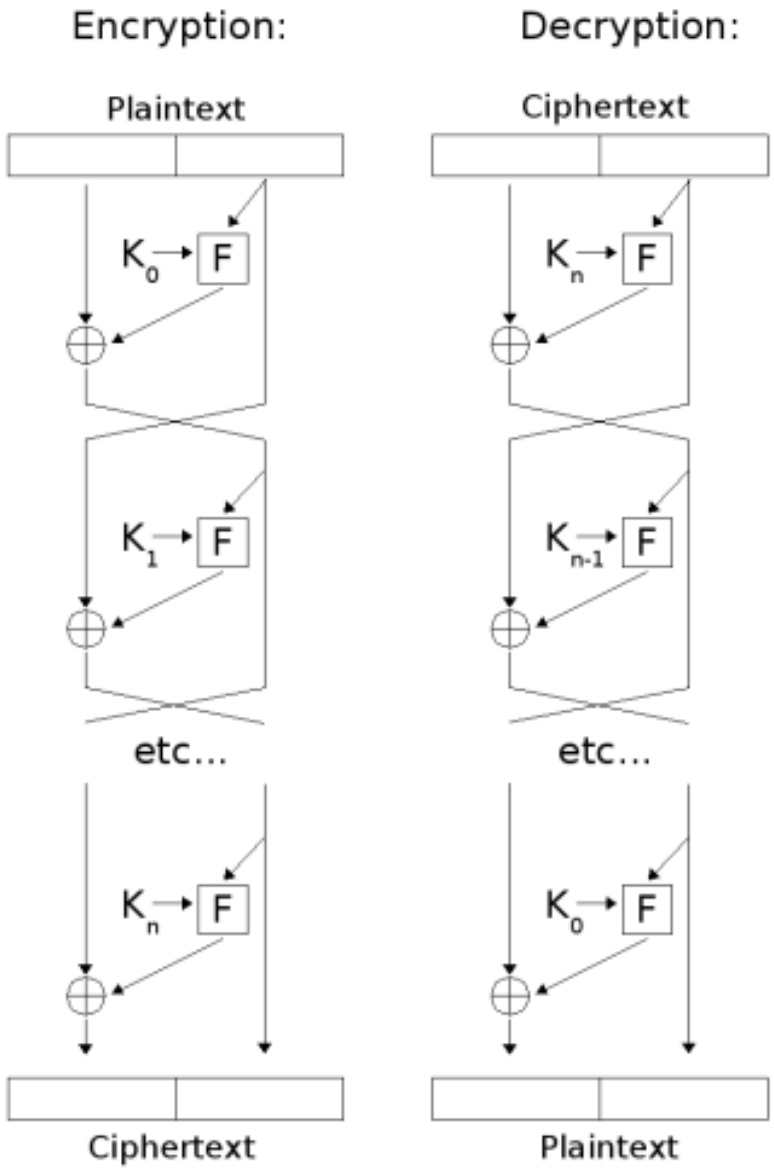
Schemat Feistel

Najbardziej powszechny schemat budowy szyfru w oparciu o funkcję jednej „rundy”. Idea:

- Na bloku do zaszyfrowania wykonywanych jest sukcesywnie wiele rund tego samego przekształcenia, ale z różnymi kluczami.
- Runda składa się ze złożenia prostych operacji, które same w sobie nie są wystarczająco bezpieczne (np. przekształcenia liniowe, translacje, podstawienia).
- Klucze dla poszczególnych rund są generowane przez ekspansję klucza głównego.

Tryby szyfrowania

- Dodatkowo szyfrowania kolejnych bloków dokonuje się w **trybach** ECB, CBC, CFB, OFB itp.

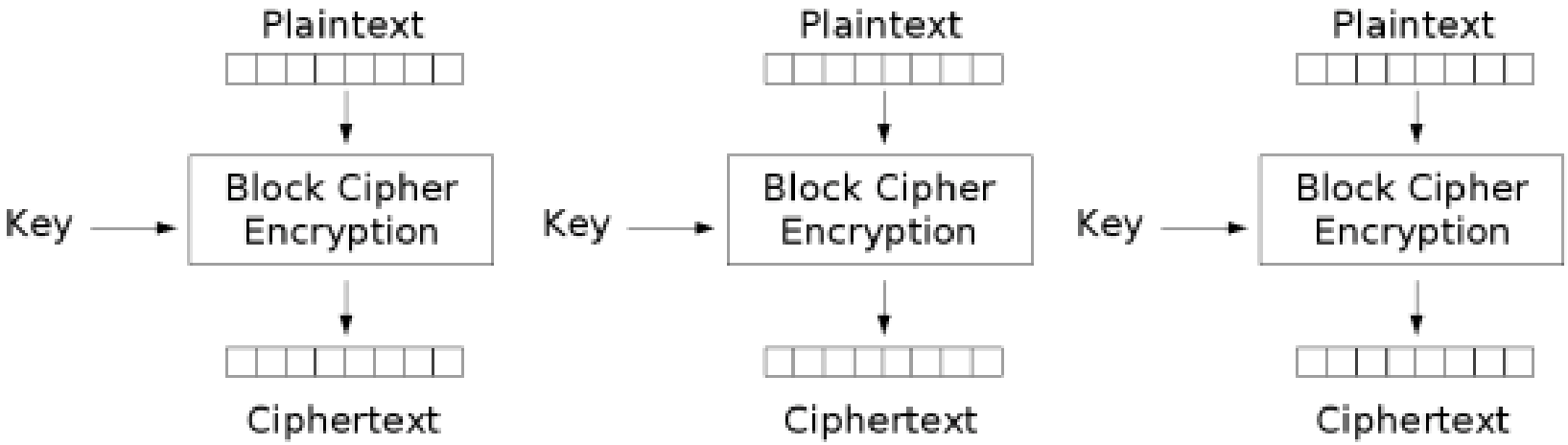


Feistel Cipher

Źródło: Wikipedia

Tryb ECB (*Electronic Code Book*)

Jeden blok tekstu jawnego jest przekształcany za pomocą przekształcenia szyfrującego na jeden blok tekstu zaszyfrowanego.

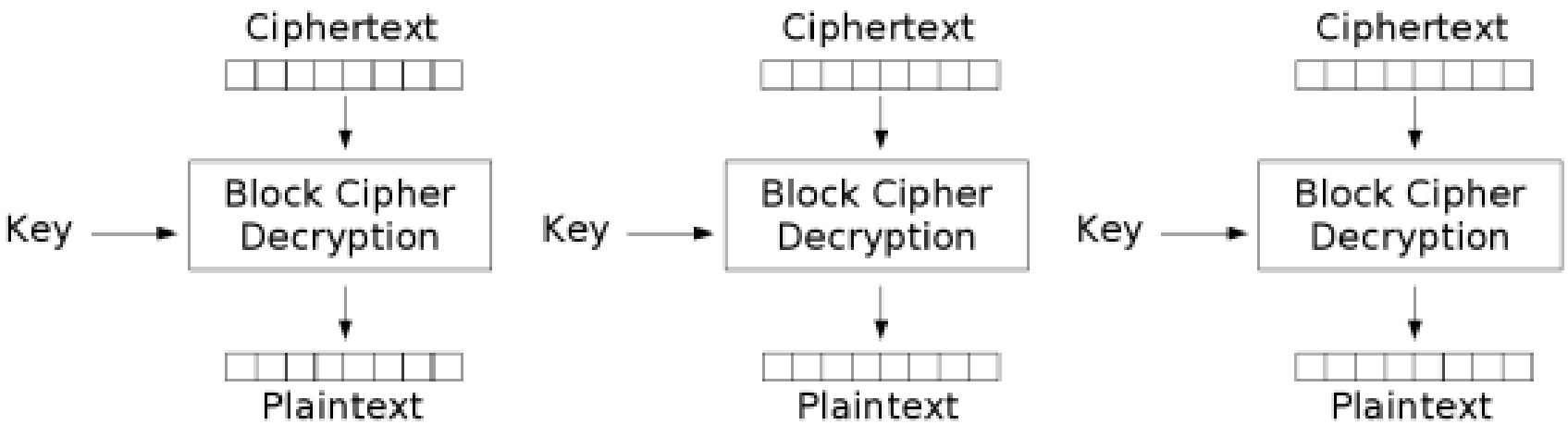


Electronic Codebook (ECB) mode encryption

Źródło obrazków: Wikipedia

Tryb ECB (*Electronic Code Book*)

Jeden blok tekstu jawnego jest przekształcany za pomocą przekształcenia szyfrującego na jeden blok tekstu zaszyfrowanego.



Electronic Codebook (ECB) mode decryption

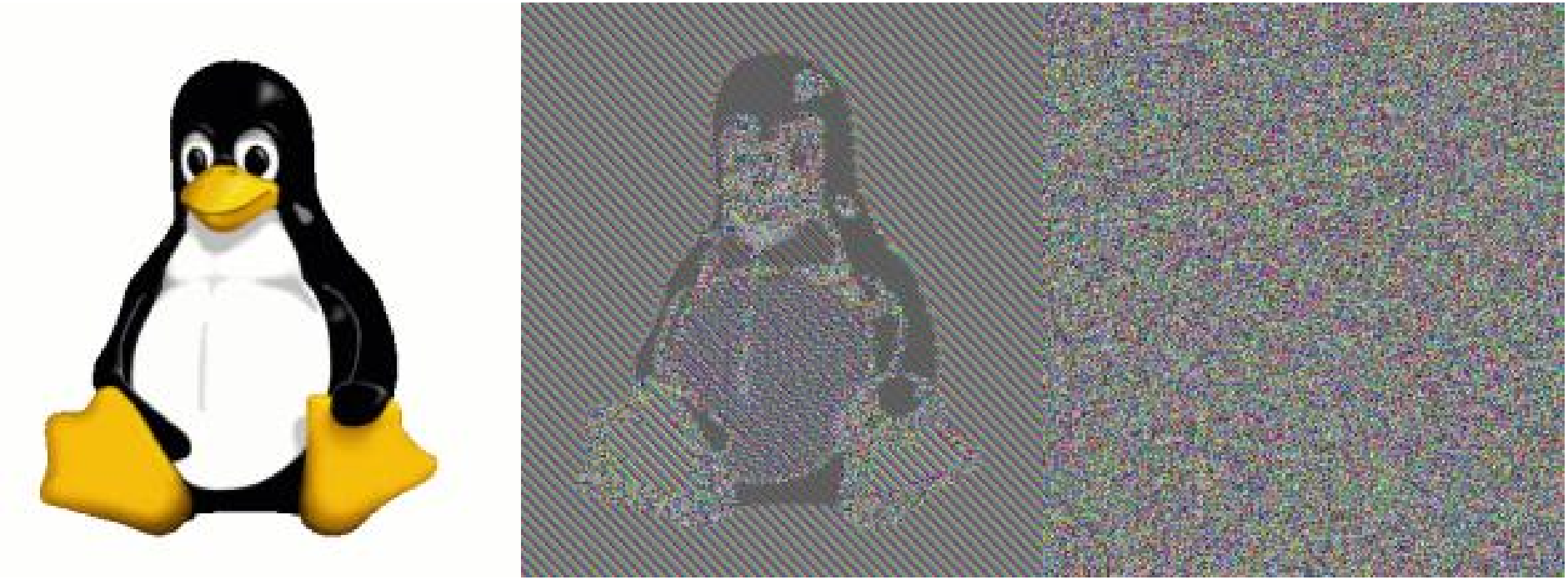
Źródło obrazków: Wikipedia

Tryb ECB – wady

- jednakowe bloki tekstu jawnego dają jednakowy szyfrogram,
- możliwa analiza statystyczna szyfrogramu.

Tryb ECB – wady

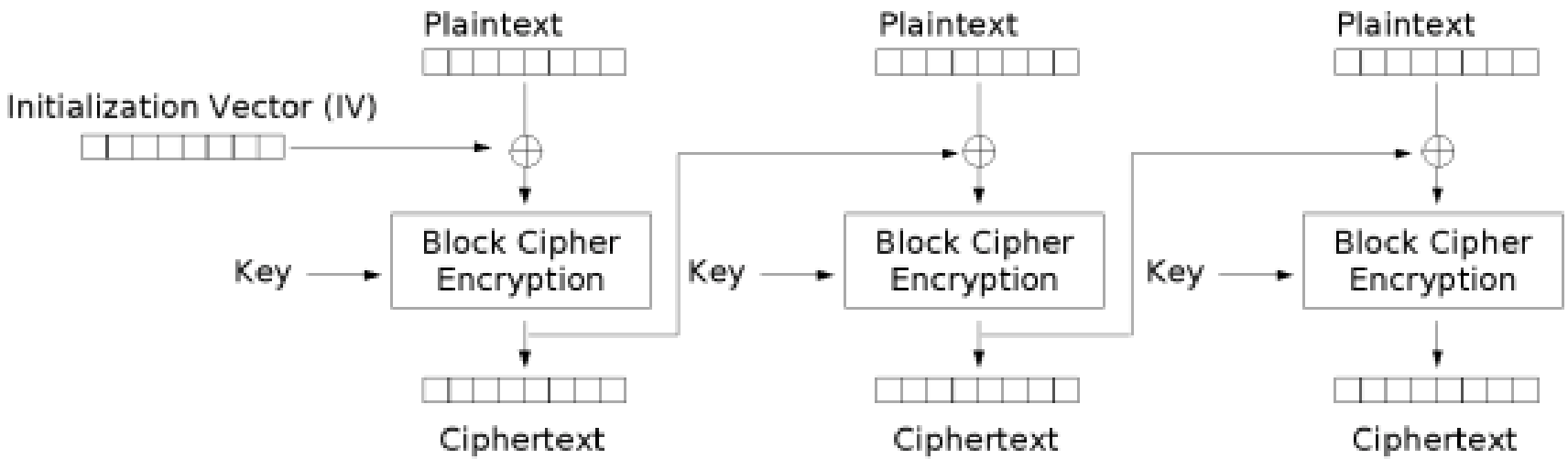
- jednakowe bloki tekstu jawnego dają jednakowy szyfrogram,
- możliwa analiza statystyczna szyfrogramu.



Źródło: Wikipedia

Tryb CBC (*Cipher Block Chaining*)

Na każdym kolejnym bloku tekstu jawnego jest wykonywana operacja XOR z poprzednio uzyskanym blokiem zaszyfrowanym i tak uzyskany wynik jest poddawany szyfrowaniu. Wymaga wektora inicjującego stan.

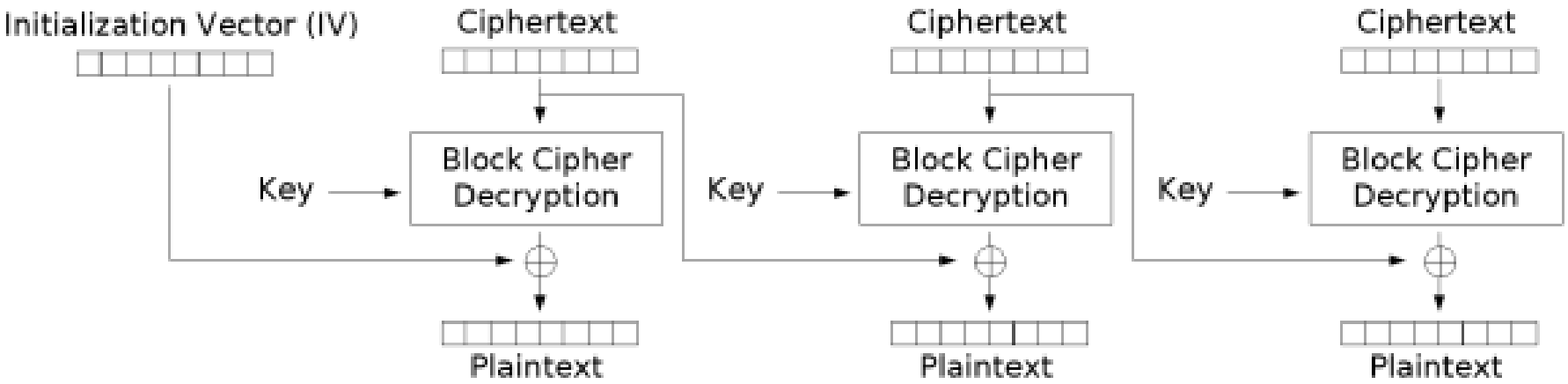


Cipher Block Chaining (CBC) mode encryption

Źródło obrazków: Wikipedia

Tryb CBC (*Cipher Block Chaining*)

Na każdym kolejnym bloku tekstu jawnego jest wykonywana operacja XOR z poprzednio uzyskanym blokiem zaszyfrowanym i tak uzyskany wynik jest poddawany szyfrowaniu. Wymaga wektora inicjującego stan.

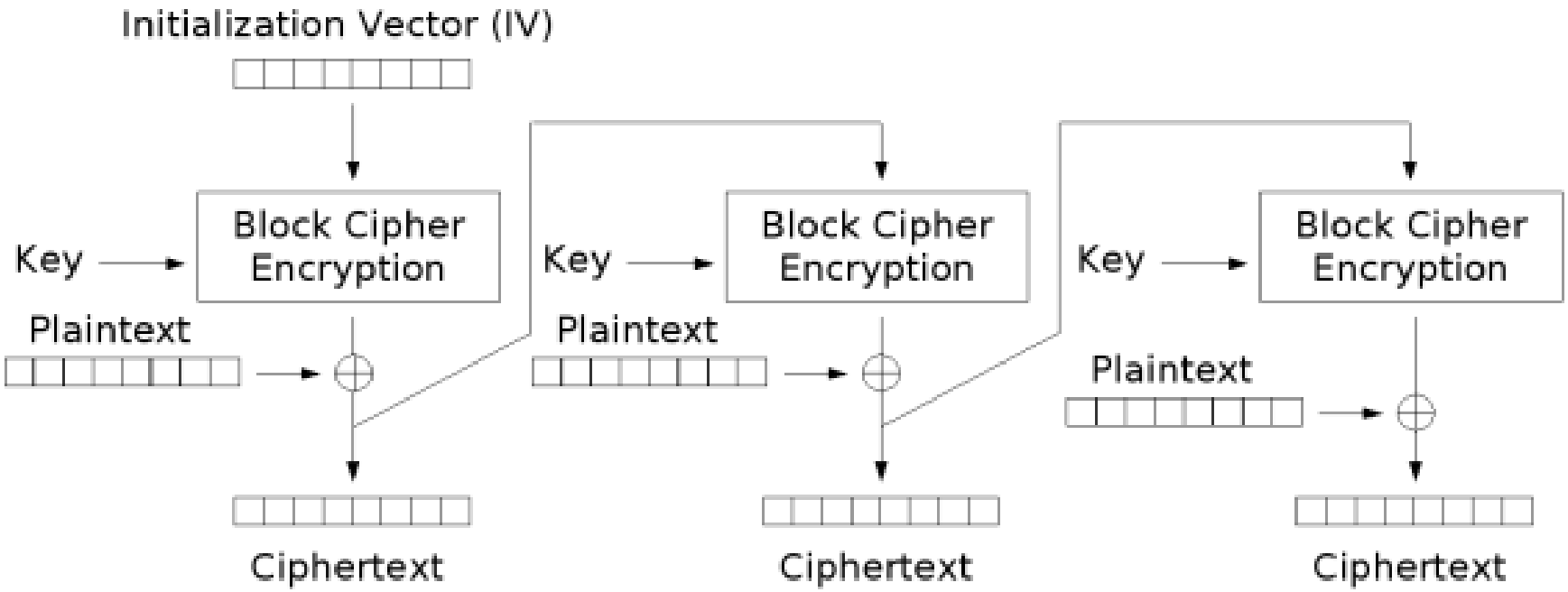


Cipher Block Chaining (CBC) mode decryption

Źródło obrazków: Wikipedia

Tryb CFB (*Cipher Feedback*)

Szyfrowany jest szyfrogram z poprzedniego bloku, jawny tekst jest XOR-owany z wynikiem tego szyfrowania.

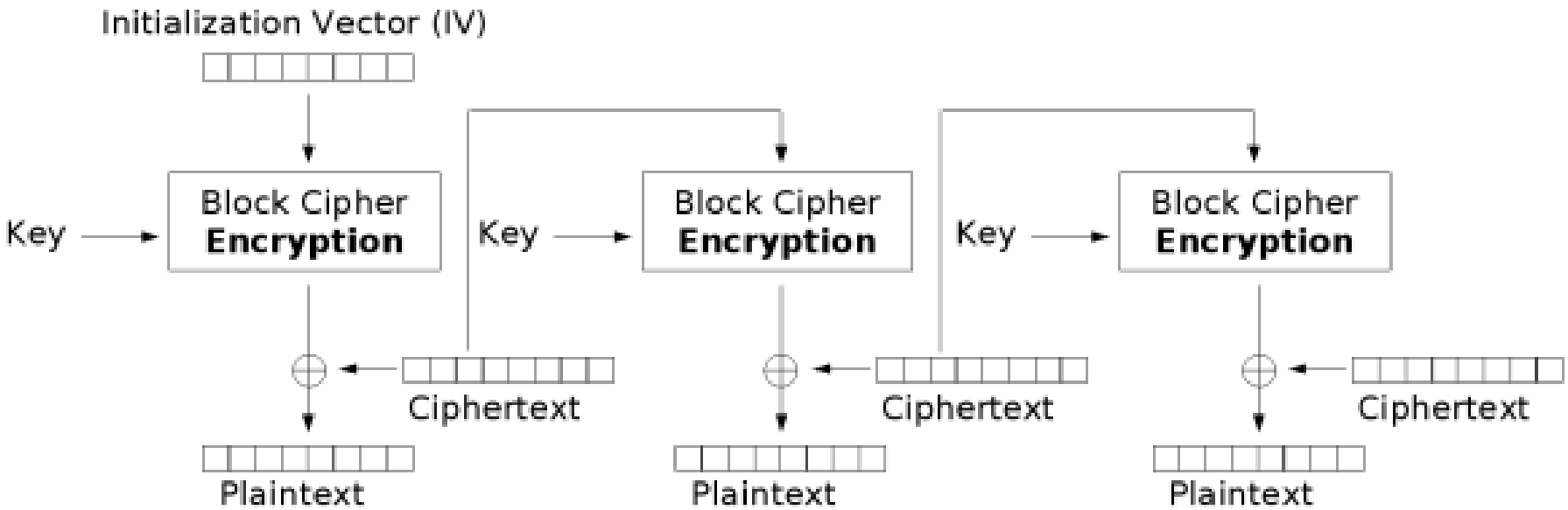


Cipher Feedback (CFB) mode encryption

Źródło obrazków: Wikipedia

Tryb CFB (*Cipher Feedback*)

Szyfrowany jest szyfrogram z poprzedniego bloku, jawny tekst jest XOR-owany z wynikiem tego szyfrowania.

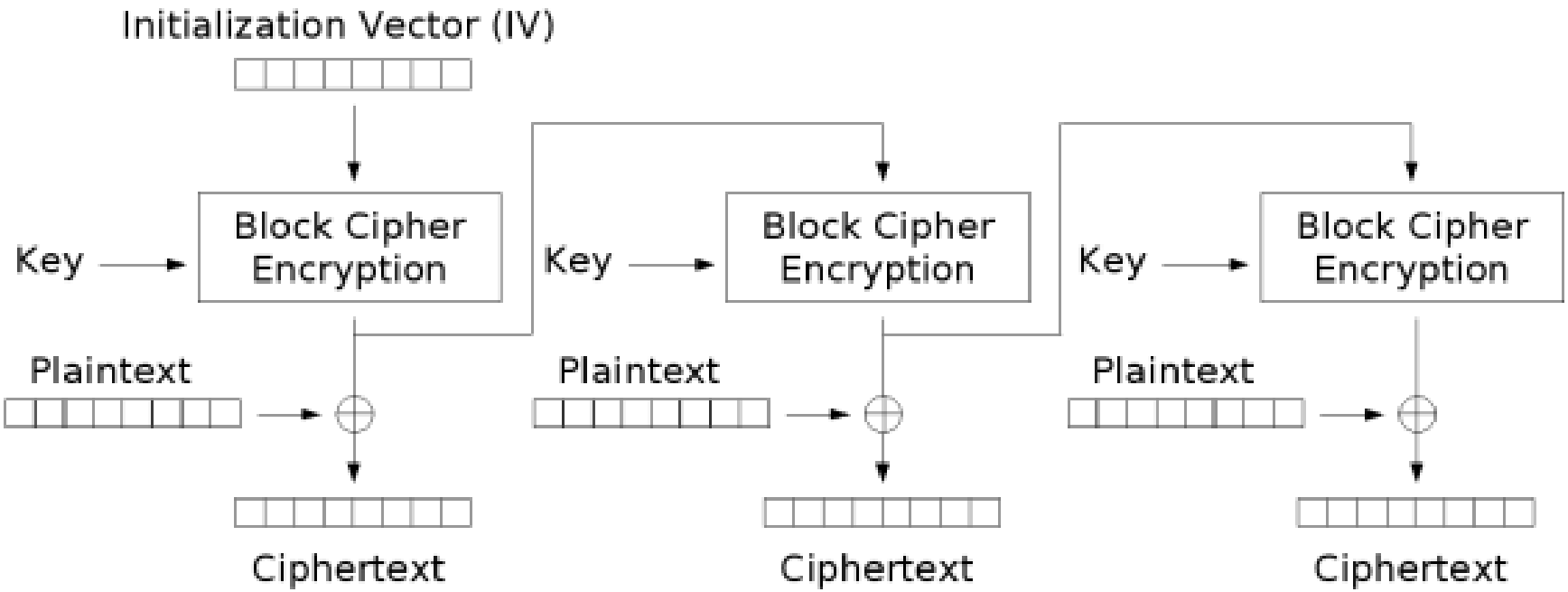


Cipher Feedback (CFB) mode decryption

Źródło obrazków: Wikipedia

OFB (*Output Feedback*)

Podobne do CFB, ale dane wejściowe nie są brane pod uwagę przy generowaniu następnego bloku do szyfrowania.

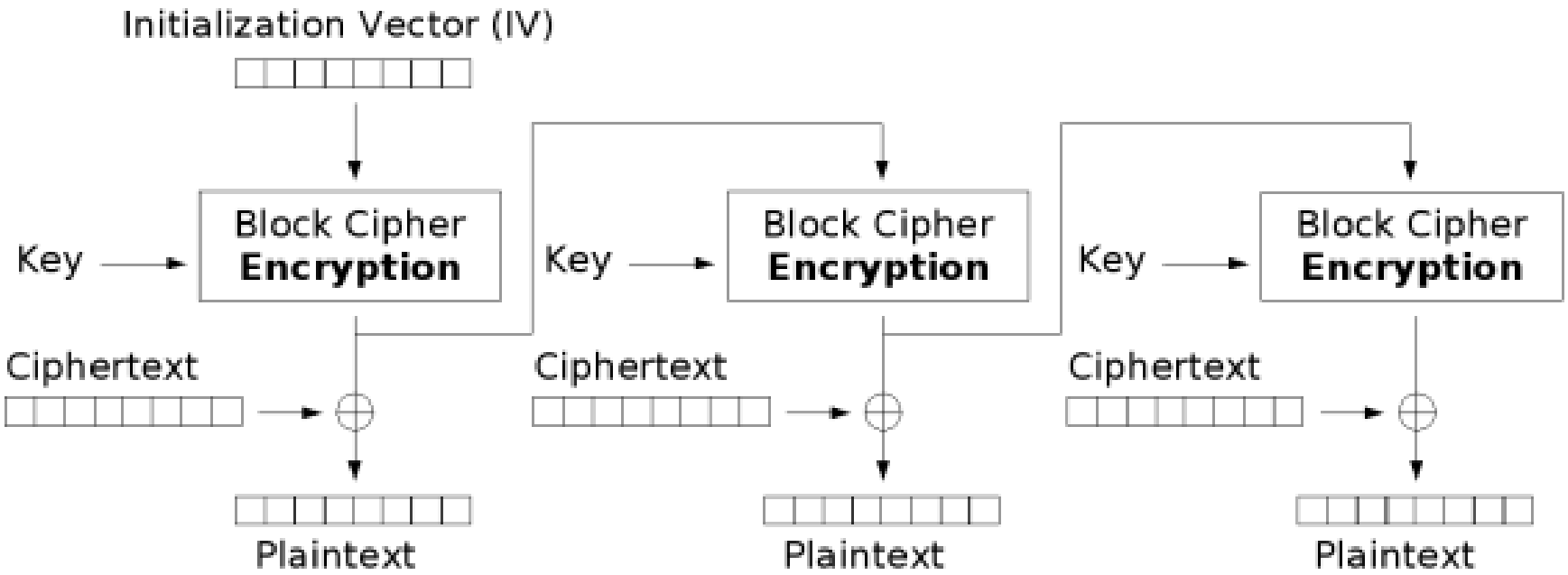


Output Feedback (OFB) mode encryption

Źródło obrazków: Wikipedia

OFB (*Output Feedback*)

Podobne do CFB, ale dane wejściowe nie są brane pod uwagę przy generowaniu następnego bloku do szyfrowania.



Output Feedback (OFB) mode decryption

Źródło obrazków: Wikipedia

Klasyfikacja szyfrów strumieniowych

- **Z jednorazowym strumieniem kluczy** – *one time pad*,
- **Synchroniczne** – strumień kluczy jest generowany niezależnie od tekstu jawnego i szyfrogramu.
- **Samosynchronizujące** – strumień kluczy jest generowany jako funkcja klucza głównego oraz pewnego fragmentu ostatnio zaszyfrowanych/odszyfrowanych danych.

Własności strumieniowych szyfrów synchronicznych

- Nadawca i odbiorca muszą być zsynchronizowani.
- Brak propagacji błędów – zmodyfikowanie fragmentu szyfrogramu nie ma wpływu na wynik deszyfrowania pozostałych danych.
- Możliwe są ataki aktywne (wstawienie, usuwanie lub powtarzanie danych) korzystające z powyższych własności.

Własności strumieniowych szyfrów samosynchronizujących

- Samosynchronizacja.
- Ograniczona propagacja błędów.
- Trudniejsze ataki aktywne – modyfikacja jednego bitu szyfrogramu ma wpływ na pewną liczbę bitów tekstu po odszyfrowaniu.
- Rozpraszanie tekstu jawnego – każdy bit tekstu jawnego ma wpływ na to, jak zostaną zaszyfrowane bity następujące po nim.

Najbardziej znane szyfry symetryczne blokowe

- DES (*Data Encryption Standard*) – Feistell
  (M=64, K=56, C=64)
- FEAL (*Fast Data Encryption Algorithm*) – Feistell
  (M=64, K=64, C=64)
- IDEA (*International Data Encryption Standard*) – Feistell
  (M=64, K=128, C=64)
- SAFER (*Secure And Fast Encryption Routine*)
  (M=64, K=64, C=64)
- RC5
  (zmienne)
- AES/Rijndael (*Advanced Encryption Standard*)
  (M=128, K=128–192–256, C=128)

Najbardziej znane szyfry symetryczne strumieniowe

- Szyfry bazujące na LFSR (*Linear Feedback Shift Registers*).
- SEAL (1993) – addytywny szyfr binarny bazujący na funkcji pseudolosowej zaprojektowany do efektywnej implementacji softwarowej.

Szyfry asymetryczne

- Wymyślone w latach 70-tych
- Używają dwóch różnych (ale „pasujących do siebie”) kluczy do szyfrowania i deszyfrowania
- Klucza deszyfrującego nie można otrzymać w efektywny sposób z klucza szyfrującego
- Opierają się na solidnych podstawach matematycznych (teoria liczb, teoria złożoności obliczeniowej)
- Są nieefektywne:
  - szyfrowanie i deszyfrowanie przebiega wolno
  - klucze muszą być duże – co najmniej 1000 bitów

Szyfry asymetryczne

- Wymyślone w latach 70-tych
- Używają dwóch różnych (ale „pasujących do siebie”) kluczy do szyfrowania i deszyfrowania
- Klucza deszyfrującego nie można otrzymać w efektywny sposób z klucza szyfrującego
- Opierają się na solidnych podstawach matematycznych (teoria liczb, teoria złożoności obliczeniowej)
- Są nieefektywne:
  - szyfrowanie i deszyfrowanie przebiega wolno
  - klucze muszą być duże – co najmniej 1000 bitów

Szyfry asymetryczne

- Wymyślone w latach 70-tych
- Używają dwóch różnych (ale „pasujących do siebie”) kluczy do szyfrowania i deszyfrowania
- Klucza deszyfrującego nie można otrzymać w efektywny sposób z klucza szyfrującego
- Opierają się na solidnych podstawach matematycznych (teoria liczb, teoria złożoności obliczeniowej)
- Są nieefektywne:
  - szyfrowanie i deszyfrowanie przebiega wolno
  - klucze muszą być duże – co najmniej 1000 bitów

Szyfry asymetryczne

- Wymyślone w latach 70-tych
- Używają dwóch różnych (ale „pasujących do siebie”) kluczy do szyfrowania i deszyfrowania
- Klucza deszyfrującego nie można otrzymać w efektywny sposób z klucza szyfrującego
- Opierają się na solidnych podstawach matematycznych (teoria liczb, teoria złożoności obliczeniowej)
- Są nieefektywne:
  - szyfrowanie i deszyfrowanie przebiega wolno
  - klucze muszą być duże – co najmniej 1000 bitów

Szyfry asymetryczne

- Wymyślone w latach 70-tych
- Używają dwóch różnych (ale „pasujących do siebie”) kluczy do szyfrowania i deszyfrowania
- Klucza deszyfrującego nie można otrzymać w efektywny sposób z klucza szyfrującego
- Opierają się na solidnych podstawach matematycznych (teoria liczb, teoria złożoności obliczeniowej)
- Są nieefektywne:
  - szyfrowanie i deszyfrowanie przebiega wolno
  - klucze muszą być duże – co najmniej 1000 bitów

Matematyczne podstawy szyfrów asymetrycznych

Faktoryzacja liczb naturalnych

- Mnożenie liczb szybkie
- Faktoryzacja (dla dużych czynników pierwszych) trudna
- Algorytm RSA, szyfr Rabina

Problem logarytmu dyskretnego

- Potęgowanie modulo szybkie
- Logarytmowanie dyskretne („modulo”) trudne
- Algorytmy ElGamal, DSA

Dyskretny problem plecakowy

- Problem NP-trudny
- Przy odpowiednim doborze wag (znowu teoria liczb) daje szyfr asymetryczny
- Szyfr plecakowy Chora-Rivesta

Log. d. na krzywych eliptycznych

- Potęgowanie i logarytmowanie w grupach dyskretnych opartych o krzywe eliptyczne
- Umożliwia stosowanie krótszych kluczy
- Algorytm ECC

Matematyczne podstawy szyfrów asymetrycznych

Faktoryzacja liczb naturalnych

- Mnożenie liczb szybkie
- Faktoryzacja (dla dużych czynników pierwszych) trudna
- Algorytm RSA, szyfr Rabina

Problem logarytmu dyskretnego

- Potęgowanie modulo szybkie
- Logarytmowanie dyskretne („modulo”) trudne
- Algorytmy ElGamal, DSA

Dyskretny problem plecakowy

- Problem NP-trudny
- Przy odpowiednim doborze wag (znowu teoria liczb) daje szyfr asymetryczny
- Szyfr plecakowy Chora-Rivesta

Log. d. na krzywych eliptycznych

- Potęgowanie i logarytmowanie w grupach dyskretnych opartych o krzywe eliptyczne
- Umożliwia stosowanie krótszych kluczy
- Algorytm ECC

Matematyczne podstawy szyfrów asymetrycznych

Faktoryzacja liczb naturalnych

- Mnożenie liczb szybkie
- Faktoryzacja (dla dużych czynników pierwszych) trudna
- Algorytm RSA, szyfr Rabina

Problem logarytmu dyskretnego

- Potęgowanie modulo szybkie
- Logarytmowanie dyskretne („modulo”) trudne
- Algorytmy ElGamal, DSA

Dyskretny problem plecakowy

- Problem NP-trudny
- Przy odpowiednim doborze wag (znowu teoria liczb) daje szyfr asymetryczny
- Szyfr plecakowy Chora-Rivesta

Log. d. na krzywych eliptycznych

- Potęgowanie i logarytmowanie w grupach dyskretnych opartych o krzywe eliptyczne
- Umożliwia stosowanie krótszych kluczy
- Algorytm ECC

Matematyczne podstawy szyfrów asymetrycznych

Faktoryzacja liczb naturalnych

- Mnożenie liczb szybkie
- Faktoryzacja (dla dużych czynników pierwszych) trudna
- Algorytm RSA, szyfr Rabina

Problem logarytmu dyskretnego

- Potęgowanie modulo szybkie
- Logarytmowanie dyskretne („modulo”) trudne
- Algorytmy ElGamal, DSA

Dyskretny problem plecakowy

- Problem NP-trudny
- Przy odpowiednim doborze wag (znowu teoria liczb) daje szyfr asymetryczny
- Szyfr plecakowy Chora-Rivesta

Log. d. na krzywych eliptycznych

- Potęgowanie i logarytmowanie w grupach dyskretnych opartych o krzywe eliptyczne
- Umożliwia stosowanie krótszych kluczy
- Algorytm ECC

Kryptografia klucza publicznego

Założenie o wykorzystywaniu pary kluczy (prywatnego i publicznego) nie jest specyficzne tylko dla systemów szyfrujących. Także inne podstawowe mechanizmy kryptograficzne mogą zyskać opierając swoje działanie na takim założeniu.
Kryptografię wykorzystującą dualność kluczy nazywamy **kryptografią klucza publicznego** w odróżnieniu od **kryptografii klucza sekretnego**.

Uwierzytelnianie

Uwierzytelnianie to tak naprawdę dwa zagadnienia:

- **protokół identyfikacji** – sprawdzenie, że uczestnicy protokołu są tymi, za których się podają
- **weryfikacja integralności danych** – zapewnienie, że dane pochodzą od właściwego uczestnika protokołu i nie zostały przez nikogo zmodyfikowane „po drodze”

W pierwszym przypadku wymagamy, aby uczestnik protokołu był dostępny w momencie uwierzytelniania, natomiast w drugim przypadku to wymaganie jest zbędne.

Uwierzytelnianie

Uwierzytelnianie to tak naprawdę dwa zagadnienia:

- **protokół identyfikacji** – sprawdzenie, że uczestnicy protokołu są tymi, za których się podają
- **weryfikacja integralności danych** – zapewnienie, że dane pochodzą od właściwego uczestnika protokołu i nie zostały przez nikogo zmodyfikowane „po drodze”

W pierwszym przypadku wymagamy, aby uczestnik protokołu był dostępny w momencie uwierzytelniania, natomiast w drugim przypadku to wymaganie jest zbędne.

Protokoły identyfikacji

Protokół identyfikacji to protokół między dwoma uczestnikami: **dowodzącym** P i **weryfikującym** V. Celem protokołu jest sprawdzenie przez V czy P jest rzeczywiście tym, za kogo się podaje.

Oczekiwane własności

- **poprawność** – jeśli P jest rzeczywiście tym, za kogo się podaje, to protokół zakończy działanie z wynikiem *prawda*
- **transferowalność** – V nie może wykorzystać informacji zdobytej podczas wykonania protokołu do podszycia się pod P
- **nie podszywanie się** – prawdopodobieństwo tego, że przy wykonaniu protokołu P uda się podszyć za kogoś innego jest zaniedbywalne

Protokoły identyfikacji

Protokół identyfikacji to protokół między dwoma uczestnikami: **dowodzącym** P i **weryfikującym** V. Celem protokołu jest sprawdzenie przez V czy P jest rzeczywiście tym, za kogo się podaje.

Oczekiwane własności

- **poprawność** – jeśli P jest rzeczywiście tym, za kogo się podaje, to protokół zakończy działanie z wynikiem *prawda*
- **transferowalność** – V nie może wykorzystać informacji zdobytej podczas wykonania protokołu do podszycia się pod P
- **nie podszywanie się** – prawdopodobieństwo tego, że przy wykonaniu protokołu P uda się podszyć za kogoś innego jest zaniedbywalne

Warunek nie podszywania się

Warunek nie podszywania się powinien być spełniony nawet jeśli podszywający się uczestniczył w wielu protokołach identyfikacji zarówno z P jak i V oraz wiele sesji protokołu działa w tym samym czasie.

Kategorie protokołów identyfikacji

Obecnie istniejące protokoły identyfikacji można podzielić na następujące kategorie:

- słabe uwierzytelnianie (hasła, PIN itp.),
  - raz podsłuchane hasło może być później użyte,
  - nawet jeśli wysyłane jest w postaci zaszyfrowanej!
- protokoły typu wyzwanie – odpowiedź (*challenge—response*):
  - pytanie o losową informację z dużego zbioru (małe prawdopodobieństwo powtórzeń),
  - hasła jednorazowe,
- protokoły z wiedzą zerową.

Kategorie protokołów identyfikacji

Obecnie istniejące protokoły identyfikacji można podzielić na następujące kategorie:

- słabe uwierzytelnianie (hasła, PIN itp.),
  - raz podsłuchane hasło może być później użyte,
  - nawet jeśli wysyłane jest w postaci zaszyfrowanej!
- protokoły typu wyzwanie – odpowiedź (*challenge—response*):
  - pytanie o losową informację z dużego zbioru (małe prawdopodobieństwo powtórzeń),
  - hasła jednorazowe,
- protokoły z wiedzą zerową.

Kategorie protokołów identyfikacji

Obecnie istniejące protokoły identyfikacji można podzielić na następujące kategorie:

- słabe uwierzytelnianie (hasła, PIN itp.),
  - raz podsłuchane hasło może być później użyte,
  - nawet jeśli wysyłane jest w postaci zaszyfrowanej!
- protokoły typu wyzwanie – odpowiedź (*challenge—response*):
  - pytanie o losową informację z dużego zbioru (małe prawdopodobieństwo powtórzeń),
  - hasła jednorazowe,
- protokoły z wiedzą zerową.

Protokoły z wiedzą zerową

Protokół z wiedzą zerową to protokół pomiędzy parą graczy P i V.
P próbuje przekonać V, że posiada pewien sekret *s*, ale w taki sposób, aby nie ujawnić żadnej informacji o *s*.

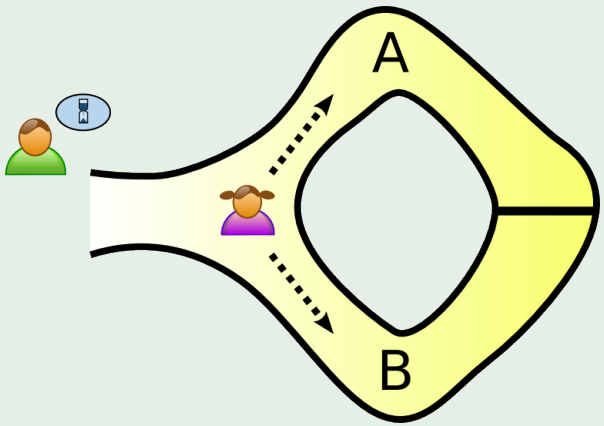
Przykład przedszkolny

Źródło: Wikipedia

Protokoły z wiedzą zerową

Protokół z wiedzą zerową to protokół pomiędzy parą graczy P i V.
P próbuje przekonać V, że posiada pewien sekret *s*, ale w taki sposób, aby nie ujawnić żadnej informacji o *s*.

Przykład przedszkolny

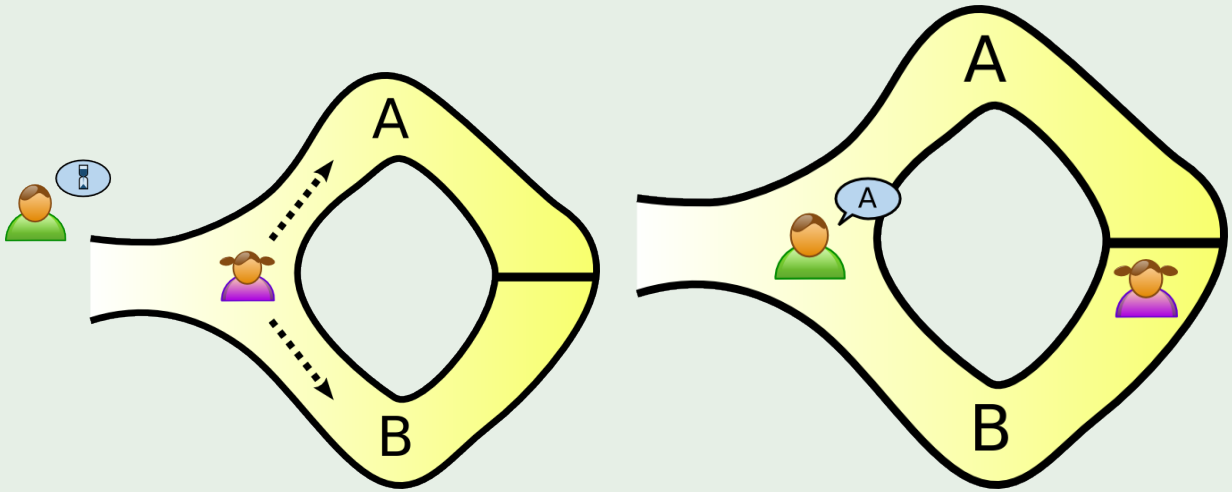


Źródło: Wikipedia

Protokoły z wiedzą zerową

Protokół z wiedzą zerową to protokół pomiędzy parą graczy P i V.
P próbuje przekonać V, że posiada pewien sekret *s*, ale w taki sposób, aby nie ujawnić żadnej informacji o *s*.

Przykład przedszkolny

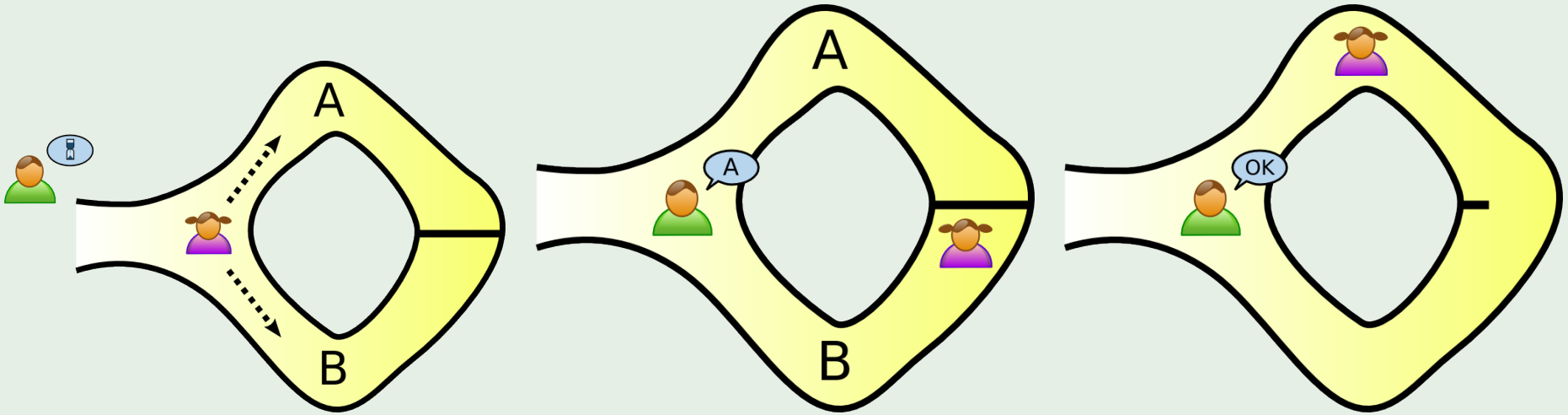


Źródło: Wikipedia

Protokoły z wiedzą zerową

Protokół z wiedzą zerową to protokół pomiędzy parą graczy P i V.
P próbuje przekonać V, że posiada pewien sekret *s*, ale w taki sposób, aby nie ujawnić żadnej informacji o *s*.

Przykład przedszkolny



Źródło: Wikipedia

Protokoły z wiedzą zerową

- Podział teoretyczny:
  - idealny protokół z wiedzą zerową,
  - statystyczny protokół z wiedzą zerową,
  - obliczeniowy protokół z wiedzą zerową.
- Najbardziej znane protokoły z wiedzą zerową:
  - Fiata-Shamira i Feige-Fiata-Shamira (oparty o faktoryzację liczb złożonych i dyskretny pierwiastek kwadratowy),
  - Schnorra (oparty o logarytm dyskretny),
  - GQ (Guillou-Quisquater).
- W (prawie?) wszystkich protokołach tego typu wykorzystywane są losowe liczby, generowane na potrzeby jednej sesji.
- W omawianych na przyszłych wykładach protokołach bezpiecznej komunikacji zwykle występuje unikalna/losowa wartość wysyłana na początku komunikacji.

Protokoły z wiedzą zerową

- Podział teoretyczny:
  - idealny protokół z wiedzą zerową,
  - statystyczny protokół z wiedzą zerową,
  - obliczeniowy protokół z wiedzą zerową.
- Najbardziej znane protokoły z wiedzą zerową:
  - Fiata-Shamira i Feige-Fiata-Shamira (oparty o faktoryzację liczb złożonych i dyskretny pierwiastek kwadratowy),
  - Schnorra (oparty o logarytm dyskretny),
  - GQ (Guillou-Quisquater).
- W (prawie?) wszystkich protokołach tego typu wykorzystywane są losowe liczby, generowane na potrzeby jednej sesji.
- W omawianych na przyszłych wykładach protokołach bezpiecznej komunikacji zwykle występuje unikalna/losowa wartość wysyłana na początku komunikacji.

Protokoły z wiedzą zerową

- Podział teoretyczny:
  - idealny protokół z wiedzą zerową,
  - statystyczny protokół z wiedzą zerową,
  - obliczeniowy protokół z wiedzą zerową.
- Najbardziej znane protokoły z wiedzą zerową:
  - Fiata-Shamira i Feige-Fiata-Shamira (oparty o faktoryzację liczb złożonych i dyskretny pierwiastek kwadratowy),
  - Schnorra (oparty o logarytm dyskretny),
  - GQ (Guillou-Quisquater).
- W (prawie?) wszystkich protokołach tego typu wykorzystywane są losowe liczby, generowane na potrzeby jednej sesji.
- W omawianych na przyszłych wykładach protokołach bezpiecznej komunikacji zwykle występuje unikalna/losowa wartość wysyłana na początku komunikacji.

Protokoły z wiedzą zerową

- Podział teoretyczny:
  - idealny protokół z wiedzą zerową,
  - statystyczny protokół z wiedzą zerową,
  - obliczeniowy protokół z wiedzą zerową.
- Najbardziej znane protokoły z wiedzą zerową:
  - Fiata-Shamira i Feige-Fiata-Shamira (oparty o faktoryzację liczb złożonych i dyskretny pierwiastek kwadratowy),
  - Schnorra (oparty o logarytm dyskretny),
  - GQ (Guillou-Quisquater).
- W (prawie?) wszystkich protokołach tego typu wykorzystywane są losowe liczby, generowane na potrzeby jednej sesji.
- W omawianych na przyszłych wykładach protokołach bezpiecznej komunikacji zwykle występuje unikalna/losowa wartość wysyłana na początku komunikacji.

Bezpieczne funkcje haszujące – definicja

Bezpieczna funkcja haszująca to funkcja

$$h : \{0,1\}^* \to \{0,1\}^n$$

taka, że trudne („obliczeniowo niewykonalne”) jest:

- dla ustalonego $c \in \{0,1\}^n$ znalezienie takiego m, że $h(m) = c$ („funkcja jednokierunkowa”),
- znalezienie dwóch ciągów m_1 i m_2 takich, że $h(m_1) = h(m_2)$.

Bezpieczne funkcje haszujące – zastosowania

Pozwalają one dokonać „kompresji” danych w taki sposób, aby nie popsuć bezpieczeństwa algorytmów, w których uczestniczą.

Zastosowania związane z bezpieczeństwem

- Podpisy cyfrowe
- Zapewnienie integralności danych
- Złożone protokoły kryptograficzne

Bezpieczne funkcje haszujące – zastosowania

Pozwalają one dokonać „kompresji” danych w taki sposób, aby nie popsuć bezpieczeństwa algorytmów, w których uczestniczą.

Zastosowania związane z bezpieczeństwem

- Podpisy cyfrowe
- Zapewnienie integralności danych
- Złożone protokoły kryptograficzne

MAC i MDC

- **MAC –** *Message Authentication Code*
  - Do generowania oraz weryfikacji kodu potrzebny jest (ten sam) tajny klucz.
- **MDC –** *Modification Detection Code*,
  MIC – *Message Integrity Code*
  - Nie wymaga tajnego klucza.
  - Stosowane do kontroli integralności danych oraz w schematach podpisów cyfrowych
    (wiadomość $\rightarrow$ MDC $\rightarrow$ podpis).

MAC i MDC

- **MAC –** *Message Authentication Code*
  - Do generowania oraz weryfikacji kodu potrzebny jest (ten sam) tajny klucz.
- **MDC –** *Modification Detection Code*, **MIC –** *Message Integrity Code*
  - Nie wymaga tajnego klucza.
  - Stosowane do kontroli integralności danych oraz w schematach podpisów cyfrowych (wiadomość $\rightarrow$ MDC $\rightarrow$ podpis).

Pokrewne algorytmy

Oba rodzaje bezpiecznych funkcji haszujących należy odróżnić od:

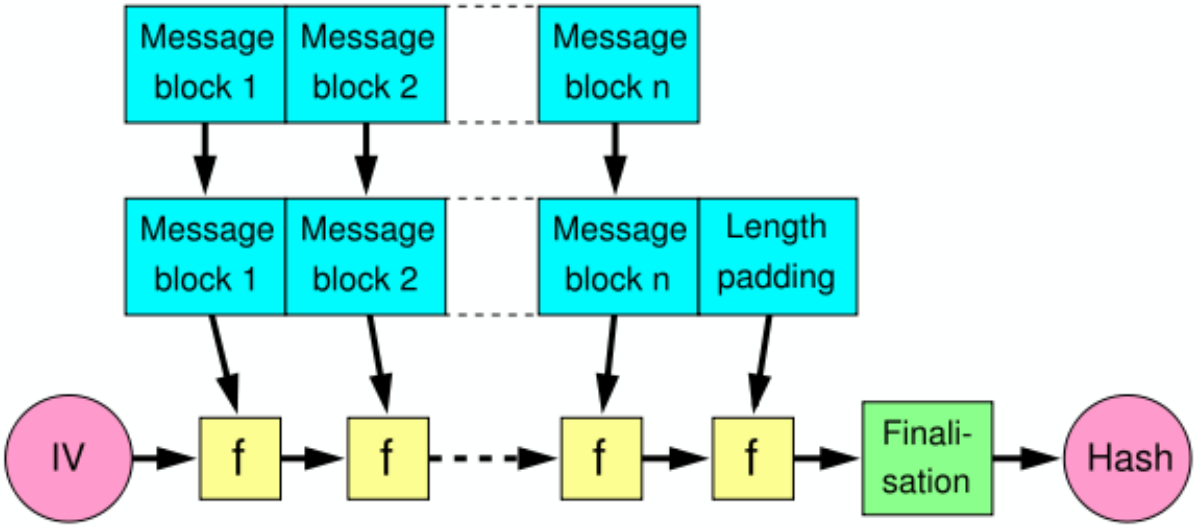
- sum kontrolnych – które nie muszą spełniać warunków bezpieczeństwa (MDC jest sumą kontrolną, ale nie każda suma kontrolna jest MDC),
- podpisów cyfrowych opartych o klucz publiczny – w MAC jest jeden tajny klucz.

Budowa funkcji haszującej z szyfru blokowego

- Szyfr blokowy może zostać wykorzystany do stworzenia bezpiecznej funkcji haszującej.
- Warunek: każdy ciąg bitów długości K jest dobrym kluczem.
- Popularny schemat na poziomie ciągu bloków:
  - Merkle-Damgård
- Popularne schematy na poziomie pojedynczych bloków:
  - Davies-Meyer
  - Matyas-Meyer-Oseas
  - Miyaguchi-Preneel

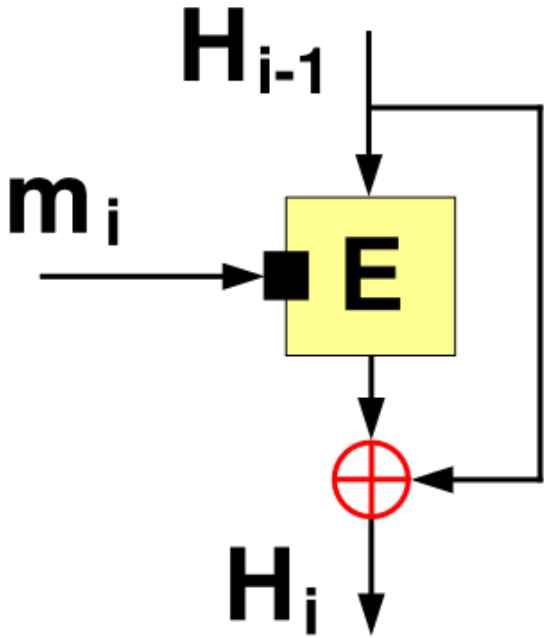
Schemat Merkle-Damgård (cała wiadomość)

- Dla bezpieczeństwa w ostatnim bloku uzupełnienie zawierające długość oryginalnej wiadomości.



Źródło obrazka: Wikipedia

Schemat Davies-Meyer (pojedyncze bloki)



Źródło obrazka: Wikipedia

Najpopularniejsze bezpieczne funkcje haszujące

- Algorytmy MDC:
  - MDC-2 (hasz 128 bitów, IBM 1987),
  - MDC-4 (hasz 128 bitów),
  - MD4 (hasz 128 bitów, MIT 1990, obecnie bardzo słaba),
  - MD5 (hasz 128 bitów, 1991, słaba),
  - RIPEMD-160 (hasz 160 bitów, europejski),
  - SHA-1 (hasz 160 bitów, pokazano schemat ataku o czasochłonności 2^{63}),
  - SHA-2 (hasz 224/256/384/512 bitów, na razie uznawana za bezpieczną).
- Schematy budowy MAC:
  - HMAC – z funkcji haszującej, np. HMAC-MD5, HMAC-SHA1
  - CMAC – z szyfru blokowego,
  - CBC-MAC – z szyfru blokowego,
    np. Data Authentication Algorithm (oparty o DES, standard ANSI),
  - UMAC – z wielu szyfrów blokowych.

Inne zagadnienia bezpieczeństwa w sieci

W sieciach komputerowych pojawiają się zagadnienia, dla których trudno jest znaleźć satysfakcjonujące z praktycznego punktu widzenia rozwiązanie kryptograficzne. Należą do nich:

- ataki typu DOS (uniemożliwienie świadczenia usług),
- niechciana poczta (spam),
- wirusy i robaki komputerowe.

Inne zagadnienia bezpieczeństwa w sieci

Zaproponowane w ich przypadku rozwiązania wymagają zbyt dużych zmian w istniejącej architekturze sieciowej.
Często w takim przypadku bardziej efektywne okazuje się zdroworozsądkowe rozwiązanie praktyczne. Jako przykład takiego podejścia można podać kwestię odróżniania człowieka od symulującego jego zachowanie programu komputerowego.

Złożone protokoły kryptograficzne

Protokoły tego typu rozwiązują problemy, które muszą spełniać wiele własności bezpieczeństwa jednocześnie. Przykłady:

- protokoły głosowania elektronicznego,
- protokoły dzielenia sekretu i bezpiecznych obliczeń rozproszonych,
- protokoły bezpiecznego rozsyłania grupowego,
- protokoły wspomagające przestrzeganie praw autorskich (cyfrowe znaki wodne),
- protokoły elektronicznej gotówki.

Przykład: Protokół elektronicznej gotówki

Podstawowe pożądane własności protokołu elektronicznej gotówki:

- pieniądz może być emitowany tylko przez upoważnioną instytucję,
- pieniędzy nie można powielać,
- pieniądz może być przekazywany pomiędzy stronami protokołu,
- pieniądz można rozmieniać,
- płatność pieniądzem elektronicznym zapewnia anonimowość stronom transakcji.

Inne zagadnienia dotyczące bezpieczeństwa w sieci

Przykładowe dodatkowe własności protokołu elektronicznej gotówki. Niektóre z nich mogą być ze sobą, lub z podstawowymi wymaganiami, sprzeczne.

- Utrudnianie „prania brudnych pieniędzy”.
- Możliwość opodatkowania transakcji dokonywanych elektroniczną gotówką.
- Możliwość dokonywania płatności nawet w przypadku, gdy strony nie mają dostępu do sieci.
- Możliwość identyfikacji i ewentualnie unieważnienia pieniędzy pochodzących z przestępstwa.