Bezpieczeństwo bezprzewodowych sieci LAN 802.11

Maciej Smoleński smolen@students.mimuw.edu.pl

Wydział Matematyki Informatyki i Mechaniki Uniwersytetu Warszawskiego

16 stycznia 2007

Spis treści

Standardy

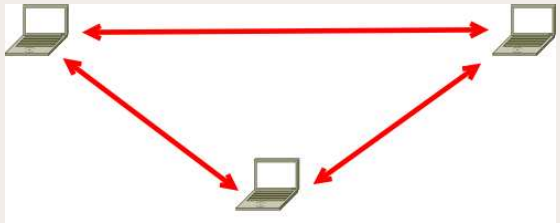
protokoły transmisji danych

Nazwa	802.11(802.11y)	802.11b	802.11a	802.11g	802.11n
Rok wydania	1997	1999	1999	2003	2008
Częstotliwość (GHz)	2.4	2.4	5.0	2.4	?
Max Transfer (Mb/s)	1,2	11	54	54 (superG 108)	540
Zasięg (m)	?	30	30	30	50

Topologie sieci WLAN

1. IBSS
2. BSS
3. ESS

IBSS (Independent Basic Service Set)

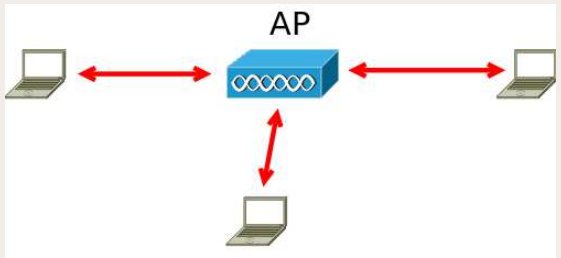


- nazywana siecią tymczasową (ad-hoc network)
- wszystkie urządzenia są równorzędne
- urządzenia komunikują się bezpośrednio

Topologie sieci WLAN

1. IBSS
2. BSS
3. ESS

BSS (Basic Service Set)

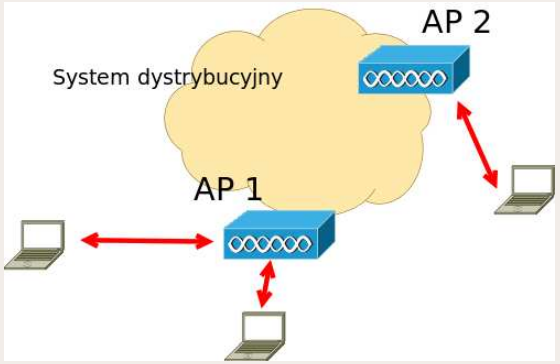


- specjalna stacja AP (access point)
- komunikacja nie jest bezpośrednia - wszystkie ramki przekazywane przez AP
- nazywana siecią strukturalną (infrastructual) - może być połączona z siecią kablową

Topologie sieci WLAN

1. IBSS
2. BSS
3. ESS

ESS (Extended Service Set)



- AP-y są połączone przez system dystrybucyjny (ethernet, ieee802.11 ...)
- fizycznie są to połączone sieci BSS
- nazywana siecią strukturalną (infrastructual)

Tryby działania kart sieciowych

- Master
- Managed
- Ad-hoc
- Monitor

Master

- karta udostępnia usługi AP-a
  - rozgłaszanie informacji o sieci
  - obsługa protokołów uwierzytelniających
  - przekazywanie ramek

Tryby działania kart sieciowych

- Master
- Managed
- Ad-hoc
- Monitor

Managed

- tryb bywa nazywany trybem **klient**a
- tryb używany w topologiach BSS i ESS
- możliwa komunikacja ze wszystkimi węzłami sieci

Tryby działania kart sieciowych

- Master
- Managed
- Ad-hoc
- Monitor

Ad-hoc

- używany w topologii IBSS
- możliwa komunikacja wyłącznie z węzłami sieci w zasięgu stacji

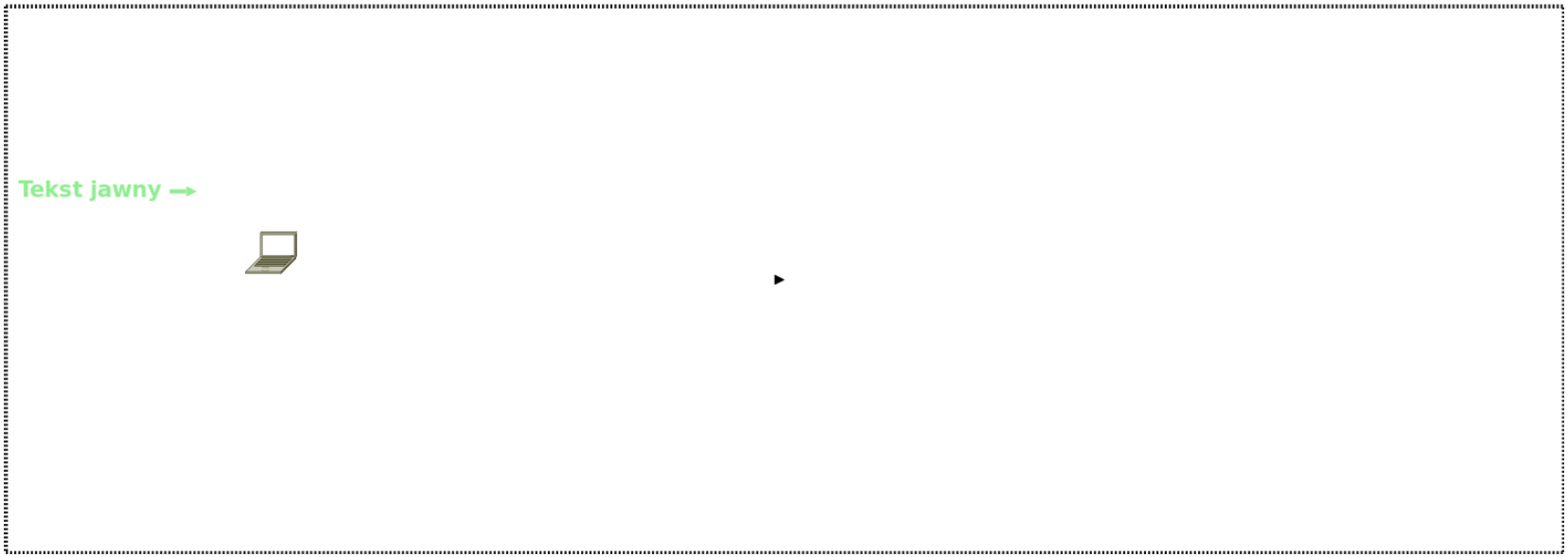
Tryby działania kart sieciowych

- Master
- Managed
- Ad-hoc
- Monitor

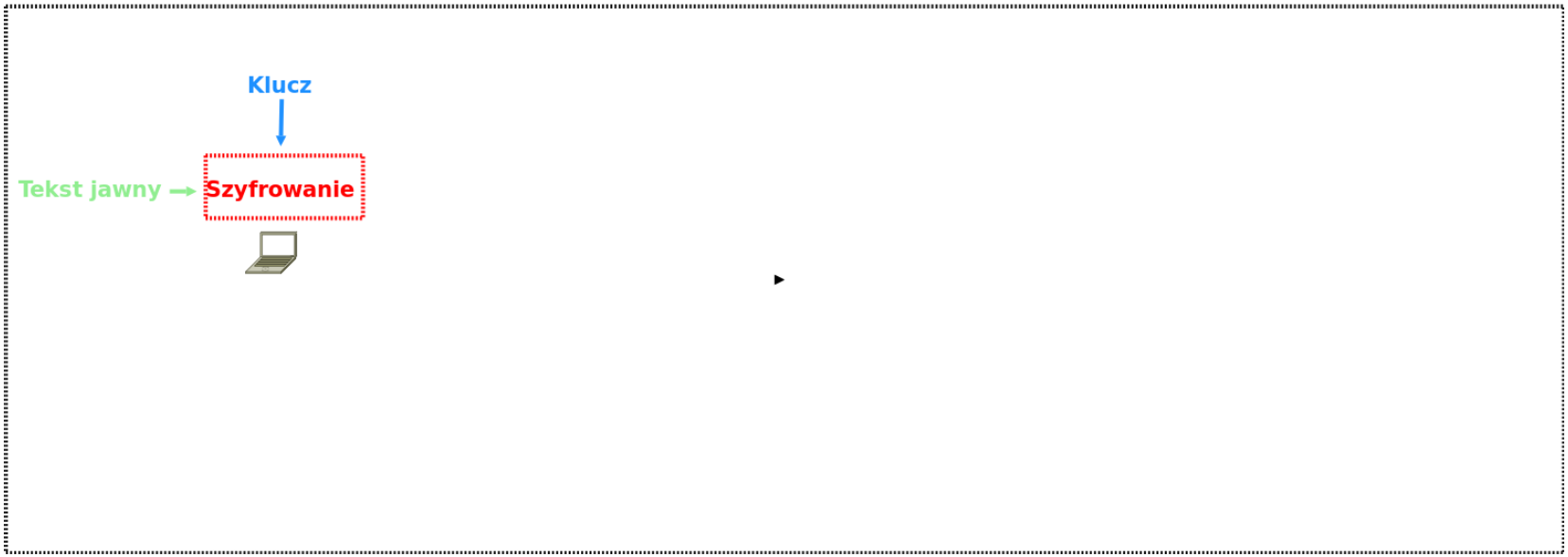
Monitor

- używany do nadzorowania sieci
- karta w tym trybie odbiera wszystkie dostępne ramki

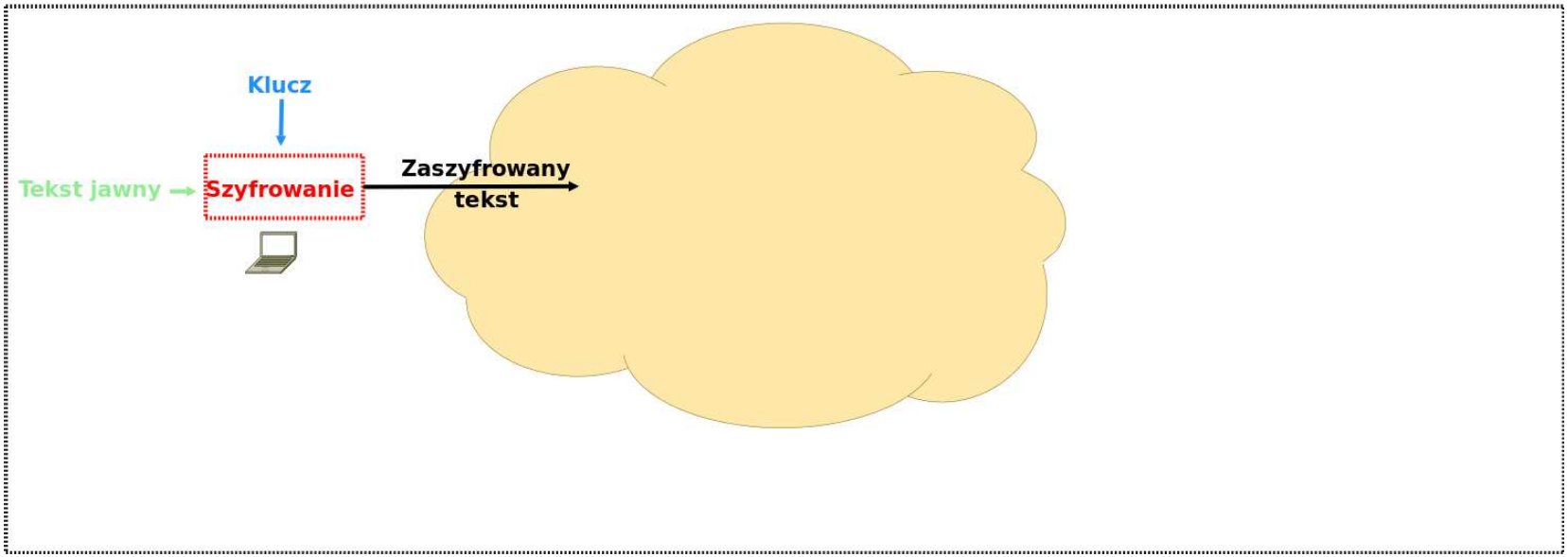
Szyfrowanie w sieciach WLAN



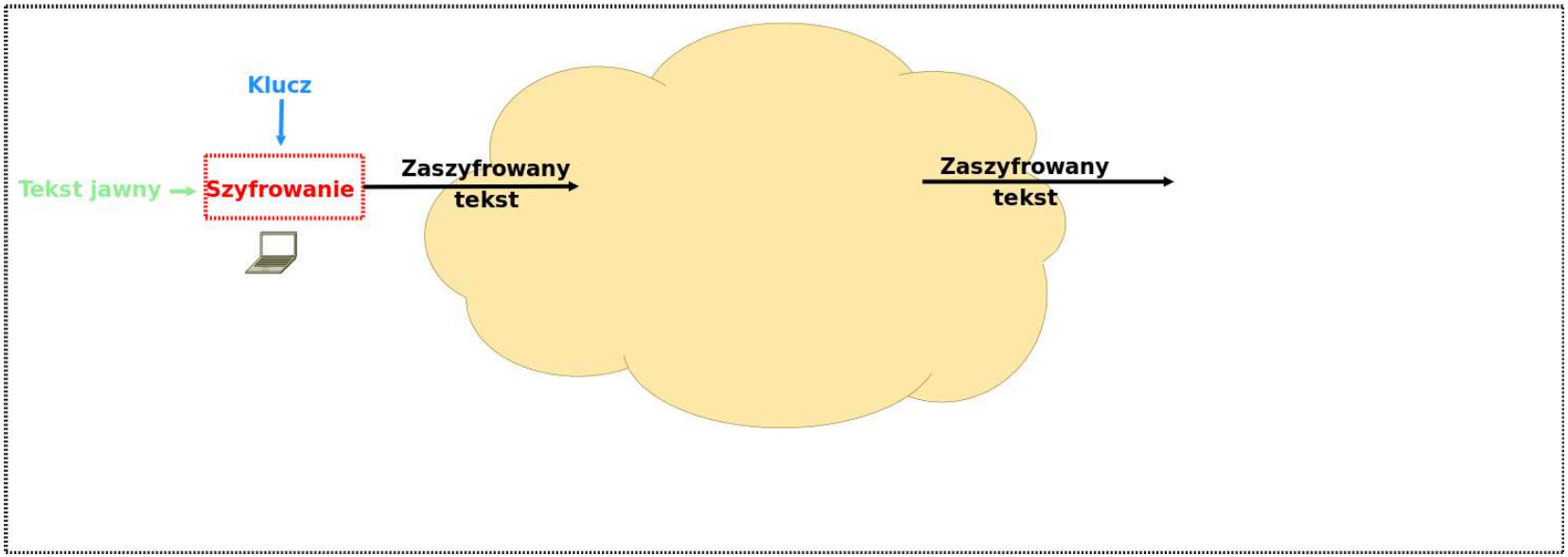
Szyfrowanie w sieciach WLAN



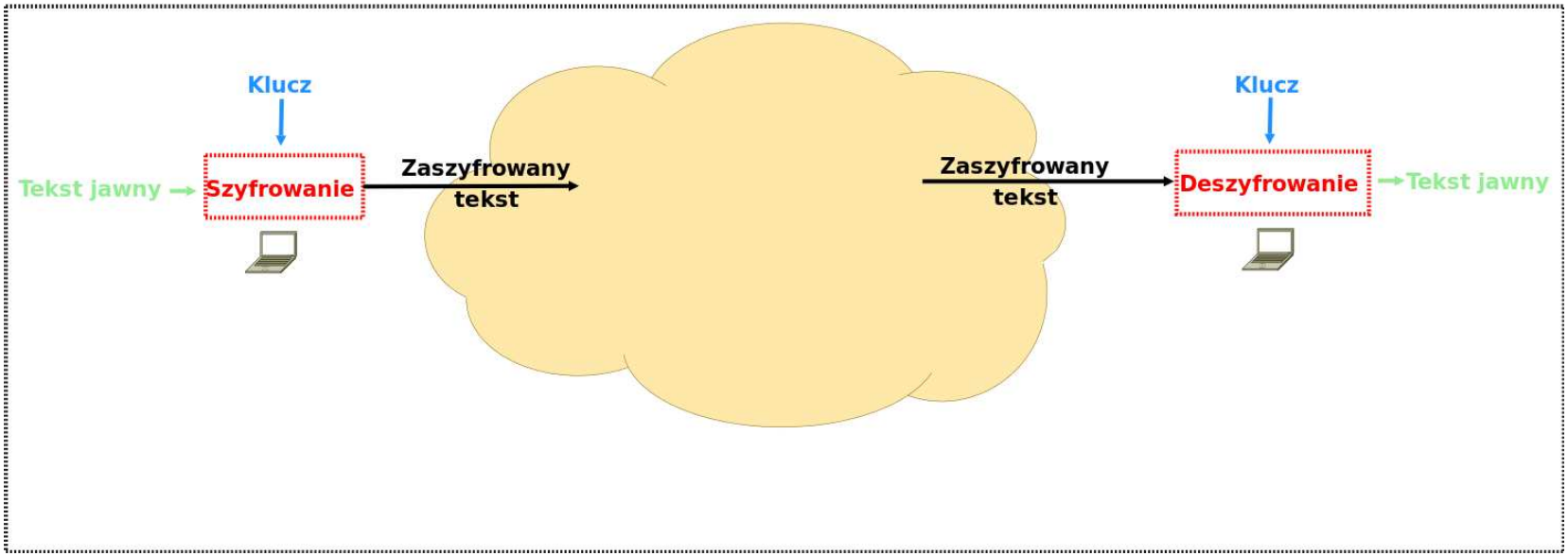
Szyfrowanie w sieciach WLAN



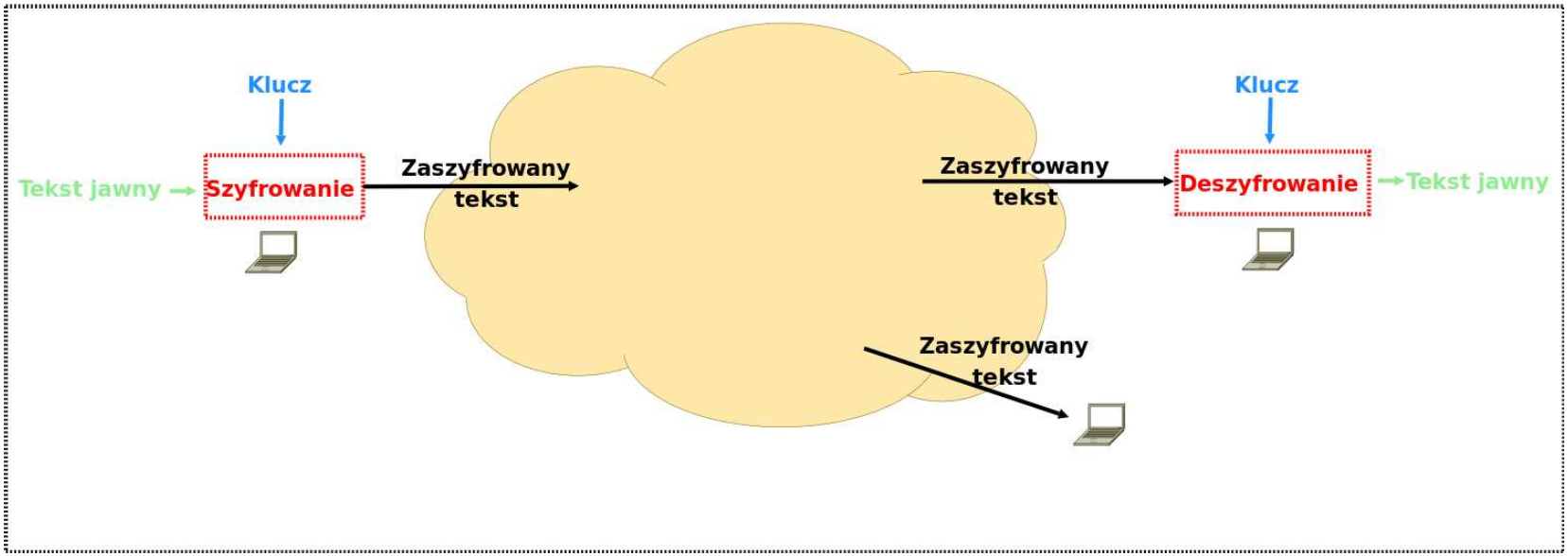
Szyfrowanie w sieciach WLAN



Szyfrowanie w sieciach WLAN



Szyfrowanie w sieciach WLAN



Zebezpieczenia sieci standardu 802.11 - WEP (Wired Equivalent Privacy)

Cele zabezpieczenia WEP

- szyfrowanie danych
- uwierzytelnianie otwarte
- uwierzytelnianie ze współdzielonym kluczem

Zebezpieczenia sieci standardu 802.11 - WEP (Wired Equivalent Privacy)

Cele zabezpieczenia WEP

- szyfrowanie danych
- uwierzytelnianie otwarte
- uwierzytelnianie ze współdzielonym kluczem

Zebezpieczenia sieci standardu 802.11 - WEP (Wired Equivalent Privacy)

Cele zabezpieczenia WEP

- szyfrowanie danych
- uwierzytelnianie otwarte
- uwierzytelnianie ze współdzielonym kluczem

Zebezpieczenia sieci standardu 802.11 - WEP (Wired Equivalent Privacy)

Cele zabezpieczenia WEP

- szyfrowanie danych
- uwierzytelnianie otwarte
- uwierzytelnianie ze współdzielonym kluczem

Specyfikacja i własności WEP

- statyczne klucze (40 lub 104-bitowe)
- użytkownicy współdzielą klucze
- wprowadzono 24-bitowe wektory inicjujące (IV)

Zebezpieczenia sieci standardu 802.11 - WEP (Wired Equivalent Privacy)

Cele zabezpieczenia WEP

- szyfrowanie danych
- uwierzytelnianie otwarte
- uwierzytelnianie ze współdzielonym kluczem

Specyfikacja i własności WEP

- statyczne klucze (40 lub 104-bitowe)
- użytkownicy współdzielą klucze
- wprowadzono 24-bitowe wektory inicjujące (IV)

Zebezpieczenia sieci standardu 802.11 - WEP (Wired Equivalent Privacy)

Cele zabezpieczenia WEP

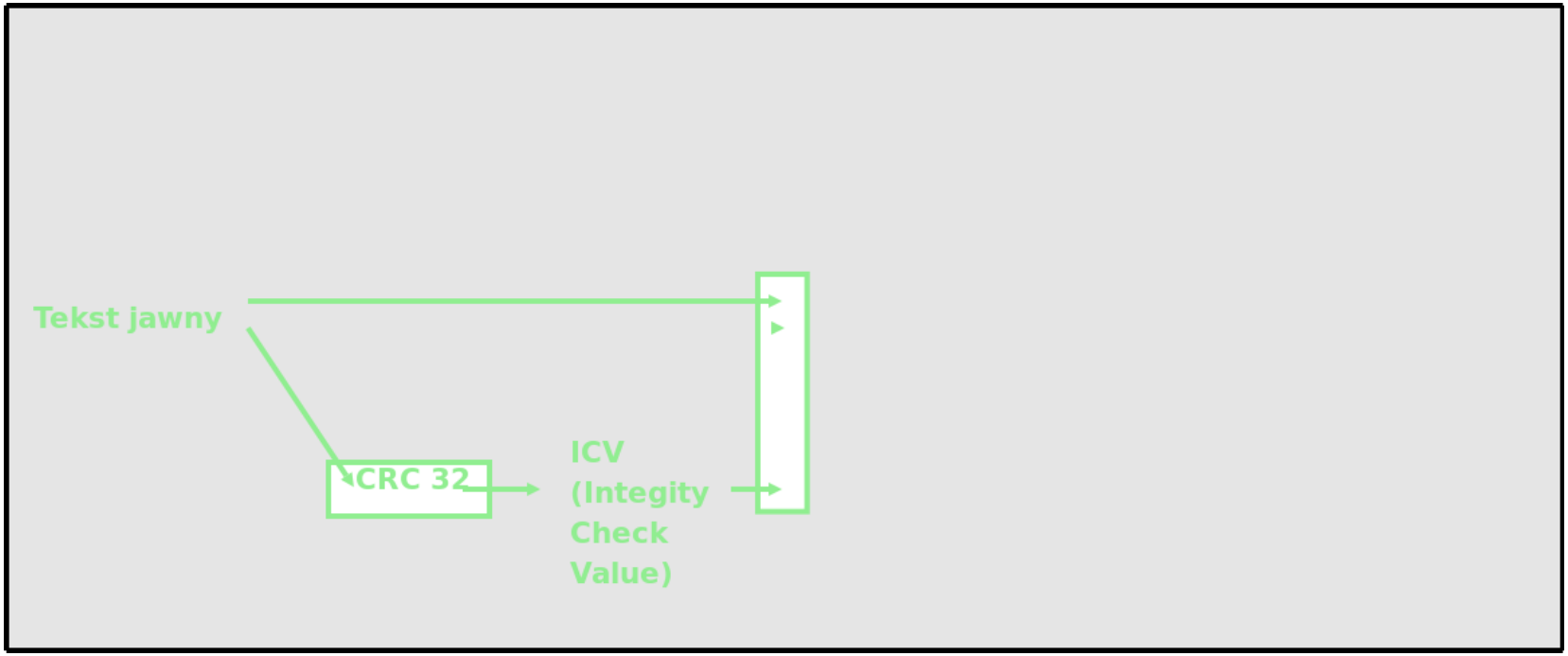
- szyfrowanie danych
- uwierzytelnianie otwarte
- uwierzytelnianie ze współdzielonym kluczem

Specyfikacja i własności WEP

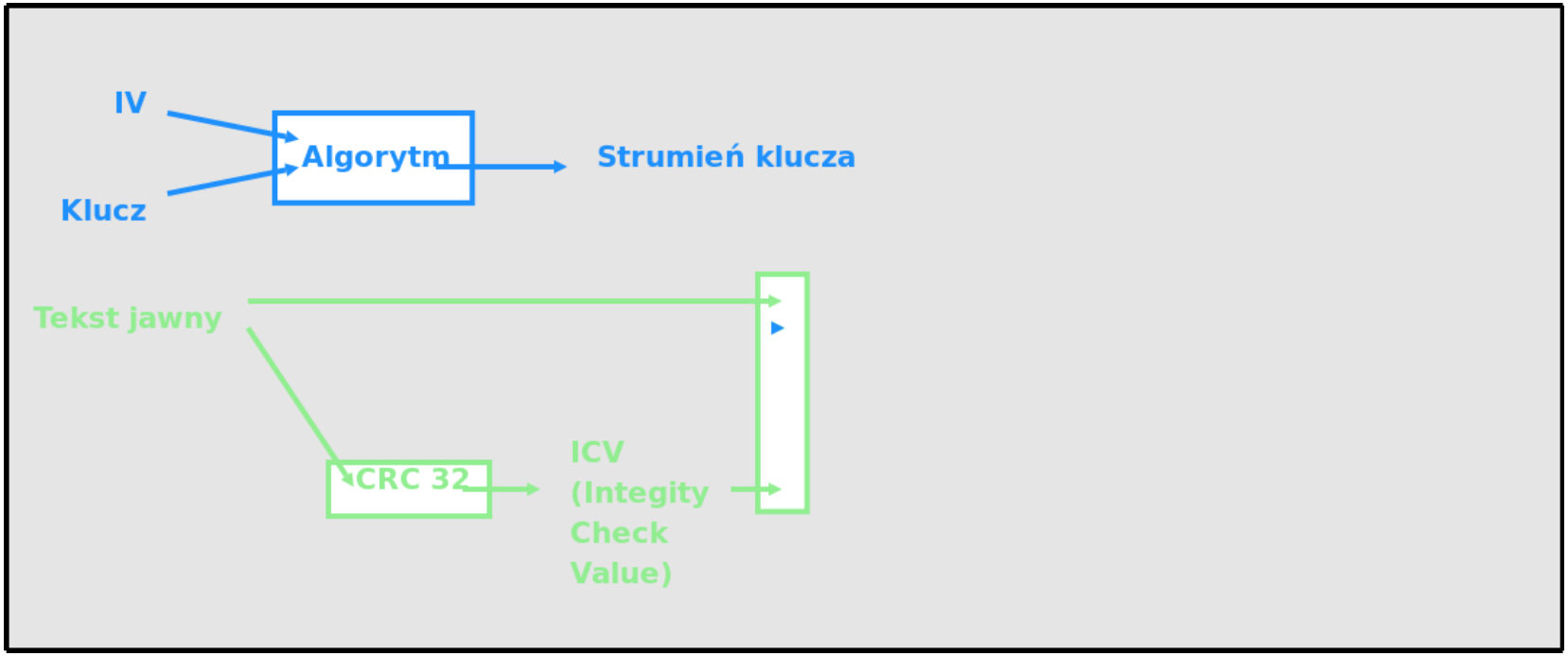
- statyczne klucze (40 lub 104-bitowe)
- użytkownicy współdzielą klucze
- wprowadzono 24-bitowe wektory inicjujące (IV)

Szyfrowanie WEP

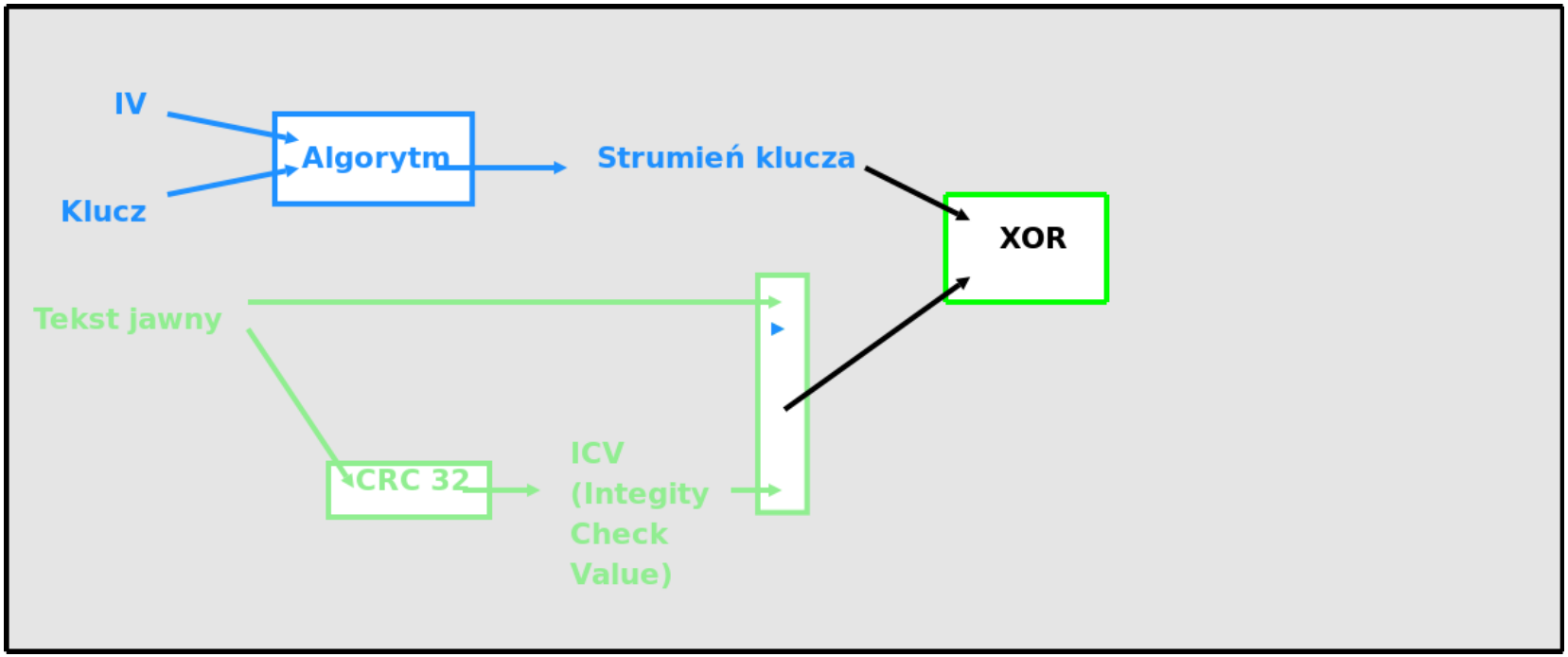
Szyfrowanie WEP



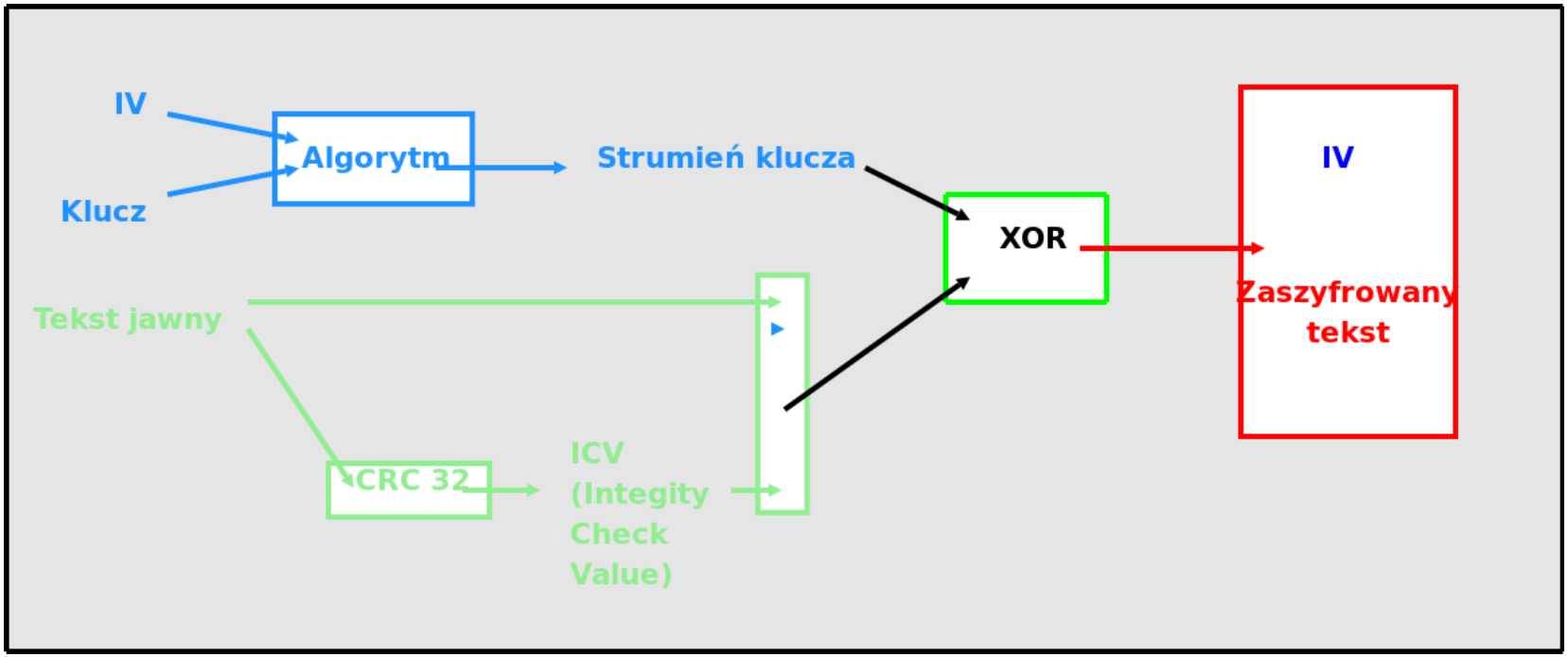
Szyfrowanie WEP



Szyfrowanie WEP



Szyfrowanie WEP



Łamanie klucza WEP

krok 1: KSA (Key Scheduling Algorithm)

```
N=256
K[] = SecretKey
for(i=0;i<N,i++)
    S[i] = i
j=0
for(i=0;i<N;i++)
    j = j + S[i] + K[i]
    swap(S[i],S[j])
```

krok 2: PRGA (Psuedo Random Generation Algorithm)

```
i=0
j=0
while( b = nextByte(input) )
    i = i+1
    j = j+S[i]
    swap(S[i], S[j])
    z = S[S[i]+S[j]]
    output(b XOR z)
```

słabości WEP

1. prawdopodobieństwo 5% że wartości S[0] ... S[3] po wykonaniu KSA będą równe wartościom po czwartej iteracji algorytmu KSA
2. szyfrowana część ramki **IEEE 802.11** zawiera na początku nagłówek SNAP: 0xAA 0xAA 0x00 0x00 0x00 0x00 0x80 0x00
3. wśród słabych wektorów wyróżniamy grupę postaci (B+3, 255, X)

Łamanie klucza WEP

krok 1: KSA (Key Scheduling Algorithm)

```
N=256
K[] = SecretKey
for(i=0;i<N,i++)
    S[i] = i
j=0
for(i=0;i<N;i++)
    j = j + S[i] + K[i]
    swap(S[i],S[j])
```

krok 2: PRGA (Psuedo Random Generation Algorithm)

```
i=0
j=0
while( b = nextByte(input) )
    i = i+1
    j = j+S[i]
    swap(S[i], S[j])
    z = S[S[i]+S[j]]
    output(b XOR z)
```

słabości WEP - analiza algorytmów

1. niech IV podsłuchanego pakietu będzie równe (3,255,7)

Zmienna	i	j	S[0]	S[1]	S[2]	S[3]
przed iteracją	0	0	0	1	2	3
po 1 iteracji	1	3	3	1	2	0
po 2 iteracji	2	3	3	0	2	1
po 3 iteracji	3	12	3	0	12	1
po 4 iteracji	4	13+K[3]	?	?	?	?

1. pierwszy bajt z wejścia jest xorowany z S[3]
2. wartość S[3] po zakończeniu KSA musiała wynosić (0xAA xor Data) załóżmy że wynosiła 0x0E
3. dlatego w trzeciej iteracji nastąpiło swap(S[3],X) gdzie X=S[0x0E], a więc j było równe w ostatniej iteracji równe 0x0E dlatego 13+K[3]=15
4. K[3]=2 czyli znamy pierwszy bajt klucza

Łamanie klucza WEP

krok 1: KSA (Key Scheduling Algorithm)

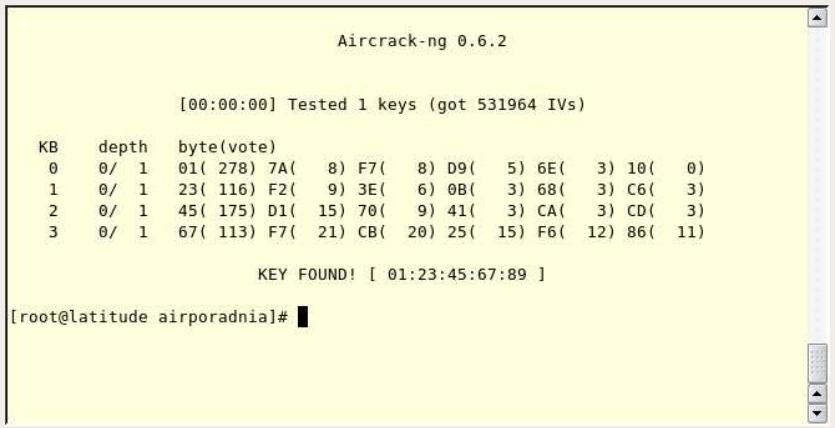
```
N=256
K[] = SecretKey
for(i=0;i<N,i++)
    S[i] = i
j=0
for(i=0;i<N;i++)
    j = j + S[i] + K[i]
    swap(S[i],S[j])
```

krok 2: PRGA (Psuedo Random Generation Algorithm)

```
i=0
j=0
while( b = nextByte(input) )
    i = i+1
    j = j+S[i]
    swap(S[i], S[j])
    z = S[S[i]+S[j]]
    output(b XOR z)
```

słabości WEP - narzędzia

- pakiet aircrack
- wystarczy:
  - przełączyć kartę w tryb **Monitor**
  - przechwycić (airodump):
    - 5-bajtowy klucz - 250000 pakietów
    - 13-bajtowy klucz - 800000 pakietów
  - obliczyć klucz WEP (aircrack).



Łamanie klucza WEP

krok 1: KSA (Key Scheduling Algorithm)

```
N=256
K[] = SecretKey
for(i=0;i<N,i++)
    S[i] = i
j=0
for(i=0;i<N;i++)
    j = j + S[i] + K[i]
    swap(S[i],S[j])
```

krok 2: PRGA (Psuedo Random Generation Algorithm)

```
i=0
j=0
while( b = nextByte(input) )
    i = i+1
    j = j+S[i]
    swap(S[i], S[j])
    z = S[S[i]+S[j]]
    output(b XOR z)
```

słabości WEP - narzędzia

W celu szybszego zgromadzenia pakietów, można generować sztuczny ruch (aireplay):

- przechwycić ramkę rządania ARP (adres docelowy to adres rozgłoszeniowy, długość 68 bajtów)
- wysłać ją wiele razy
- gromadzić odpowiedzi ARP

Inne metody zabezpieczenia sieci 802.11

- filtrowanie MAC adresów
- ukrywanie identyfikatora sieci (SSID)

Inne metody zabezpieczenia sieci 802.11

- filtrowanie MAC adresów
- ukrywanie identyfikatora sieci (SSID)

filtrowanie MAC adresów

- w konfiguracji AP ustalamy domyślną politykę oraz listy dozwolonych i zabronionych MAC adresów

Inne metody zabezpieczenia sieci 802.11

- filtrowanie MAC adresów
- ukrywanie identyfikatora sieci (SSID)

filtrowanie MAC adresów

- w konfiguracji AP ustalamy domyślną politykę oraz listy dozwolonych i zabronionych MAC adresów

omijanie zabezpieczenia

- zmiana MAC adresu

Inne metody zabezpieczenia sieci 802.11

- filtrowanie MAC adresów
- ukrywanie identyfikatora sieci (SSID)

ukrywanie identyfikatora SSID

- AP wysyła ramki **beacon** aby powiadomić sieć BSS o swojej obecności i parametrach, można ustawić na AP aby SSID był ukrywany (zamiast prawdziwej nazwy jest wysyłany <no-ssid>)

Inne metody zabezpieczenia sieci 802.11

- filtrowanie MAC adresów
- ukrywanie identyfikatora sieci (SSID)

omijanie zabezpieczenia

- klient w trakcie przyłączania się do sieci, wysyła SSID
- wystarczy:
  - uruchomić program **kismet** i poczekać aż jakiś klient będzie się przyłączał do sieci
  - przyspieszyć ten proces, znaleźć klienta (**kismet**), wysłać ramkę deautentykacji do AP z jego MAC adresem - można użyć **aireply**

standard 802.11i

specyfikacja

- schemat uwierzytelniania oparty na:
  - PSK
  - 802.1X
- dynamicznie generowane klucze, dla każdego użytkownika
- wzajemne uwierzytelnianie
- nowy algorytm szyfrowania - AES
- nowy algorytm sprawdzania integralności danych (zapobiega podrabianiu)
- sekwencje (zapobiega powielaniu)

standard 802.11i

specyfikacja

- schemat uwierzytelniania oparty na:
  - PSK
  - 802.1X
- dynamicznie generowane klucze, dla każdego użytkownika
- wzajemne uwierzytelnianie
- nowy algorytm szyfrowania - AES
- nowy algorytm sprawdzania integralności danych (zapobiega podrabianiu)
- sekwencje (zapobiega powielaniu)

standard 802.11i

specyfikacja

- schemat uwierzytelniania oparty na:
  - PSK
  - 802.1X
- dynamicznie generowane klucze, dla każdego użytkownika
- wzajemne uwierzytelnianie
- nowy algorytm szyfrowania - AES
- nowy algorytm sprawdzania integralności danych (zapobiega podrabianiu)
- sekwencje (zapobiega powielaniu)

standard 802.11i

specyfikacja

- schemat uwierzytelniania oparty na:
  - PSK
  - 802.1X
- dynamicznie generowane klucze, dla każdego użytkownika
- wzajemne uwierzytelnianie
- nowy algorytm szyfrowania - AES
- nowy algorytm sprawdzania integralności danych (zapobiega podrabianiu)
- sekwencje (zapobiega powielaniu)

standard 802.11i

specyfikacja

- schemat uwierzytelniania oparty na:
  - PSK
  - 802.1X
- dynamicznie generowane klucze, dla każdego użytkownika
- wzajemne uwierzytelnianie
- nowy algorytm szyfrowania - AES
- nowy algorytm sprawdzania integralności danych (zapobiega podrabianiu)
- sekwencje (zapobiega powielaniu)

standard 802.11i

specyfikacja

- schemat uwierzytelniania oparty na:
  - PSK
  - 802.1X
- dynamicznie generowane klucze, dla każdego użytkownika
- wzajemne uwierzytelnianie
- nowy algorytm szyfrowania - AES
- nowy algorytm sprawdzania integralności danych (zapobiega podrabianiu)
- sekwencje (zapobiega powielaniu)

standard 802.11i

specyfikacja

- schemat uwierzytelniania oparty na:
  - PSK
  - 802.1X
- dynamicznie generowane klucze, dla każdego użytkownika
- wzajemne uwierzytelnianie
- nowy algorytm szyfrowania - AES
- nowy algorytm sprawdzania integralności danych (zapobiega podrabianiu)
- sekwencje (zapobiega powielaniu)

standard 802.11i

specyfikacja

- schemat uwierzytelniania oparty na:
  - PSK
  - 802.1X
- dynamicznie generowane klucze, dla każdego użytkownika
- wzajemne uwierzytelnianie
- nowy algorytm szyfrowania - AES
- nowy algorytm sprawdzania integralności danych (zapobiega podrabianiu)
- sekwencje (zapobiega powielaniu)

standard 802.11i

Historia

- w 2001 zaczęto pracę nad standardem 802.11i
- 2003 - wprowadzenie standardu WPA
- 2004 - zakończonie prac nad 802.11i, wprowadzenie go pod nazwą WPA2

standard 802.11i

Historia

- w 2001 zaczęto pracę nad standardem 802.11i
- 2003 - wprowadzenie standardu WPA
- 2004 - zakończonie prac nad 802.11i, wprowadzenie go pod nazwą WPA2

standard 802.11i

Historia

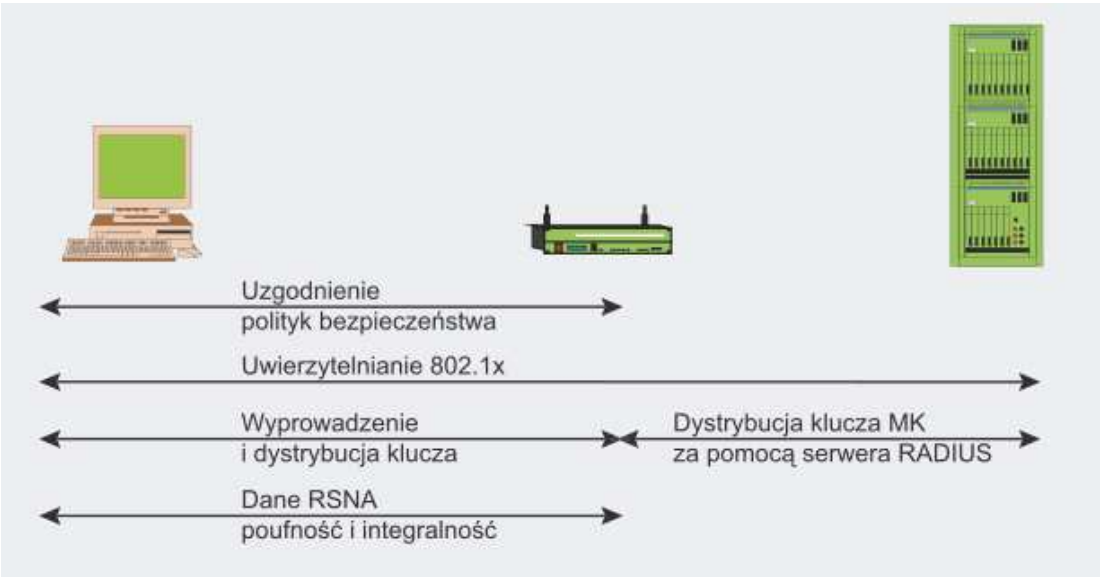
- w 2001 zaczęto pracę nad standardem 802.11i
- 2003 - wprowadzenie standardu WPA
- 2004 - zakończonie prac nad 802.11i, wprowadzenie go pod nazwą WPA2

standard 802.11i

Bezpieczna komunikacja w 802.11i

Nawiązanie bezpiecznego kontekstu komunikacji jest realizowane w czterech krokach:

- uzgodnienie polityki bezpieczeństwa
- uwierzytelnianie 802.1X lub PSK
- generowanie i dystrybucja klucza
- zapewnienie integralności i poufności przesyłanych danych

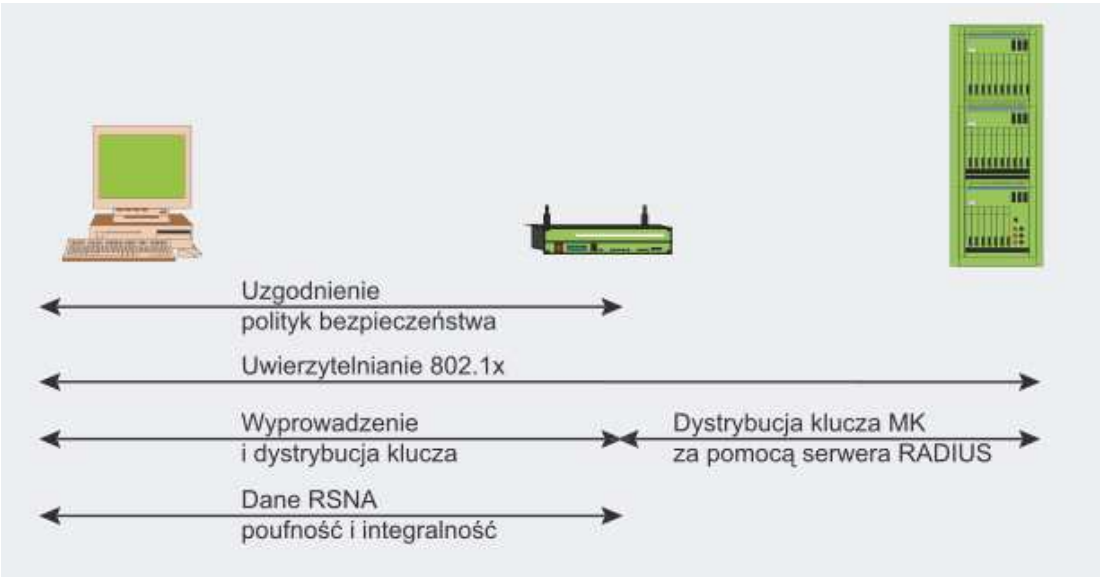


standard 802.11i

Bezpieczna komunikacja w 802.11i

Nawiązanie bezpiecznego kontekstu komunikacji jest realizowane w czterech krokach:

- uzgodnienie polityki bezpieczeństwa
- uwierzytelnianie 802.1X lub PSK
- generowanie i dystrybucja klucza
- zapewnienie integralności i poufności przesyłanych danych

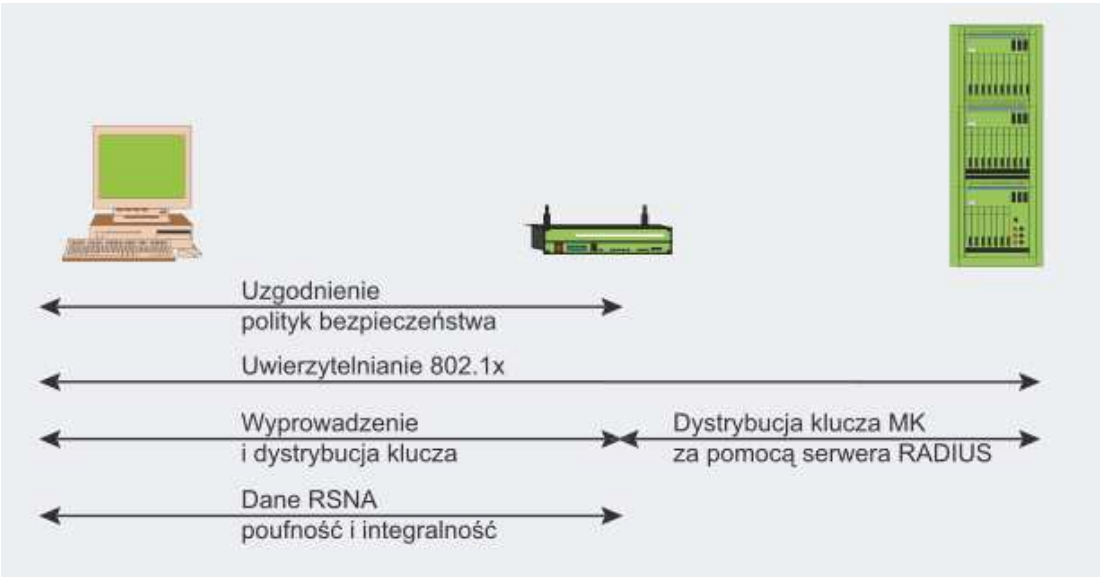


standard 802.11i

Bezpieczna komunikacja w 802.11i

Nawiązanie bezpiecznego kontekstu komunikacji jest realizowane w czterech krokach:

- uzgodnienie polityki bezpieczeństwa
- uwierzytelnianie 802.1X lub PSK
- generowanie i dystrybucja klucza
- zapewnienie integralności i poufności przesyłanych danych

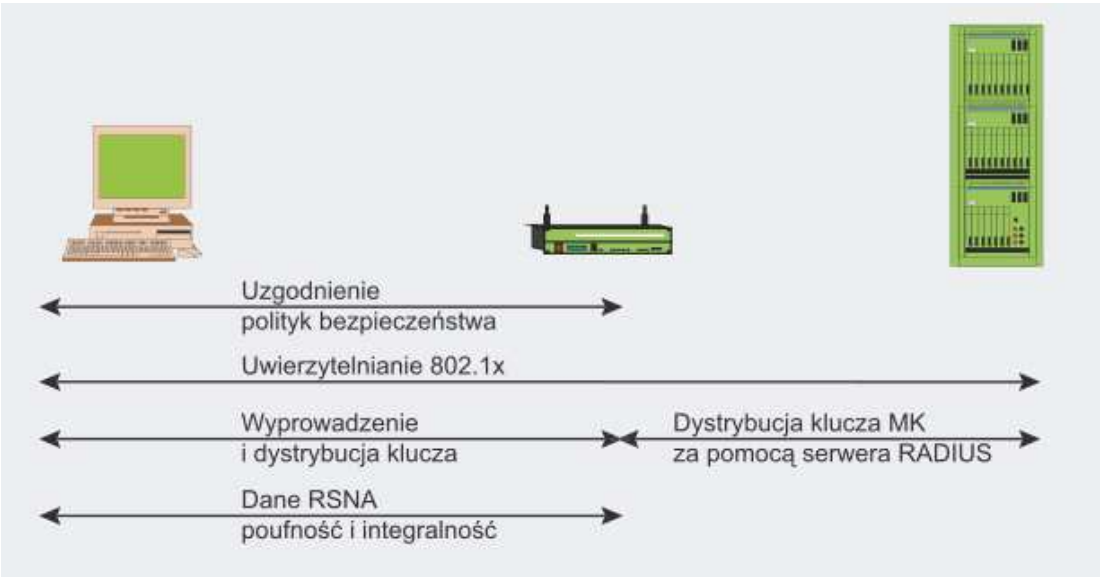


standard 802.11i

Bezpieczna komunikacja w 802.11i

Nawiązanie bezpiecznego kontekstu komunikacji jest realizowane w czterech krokach:

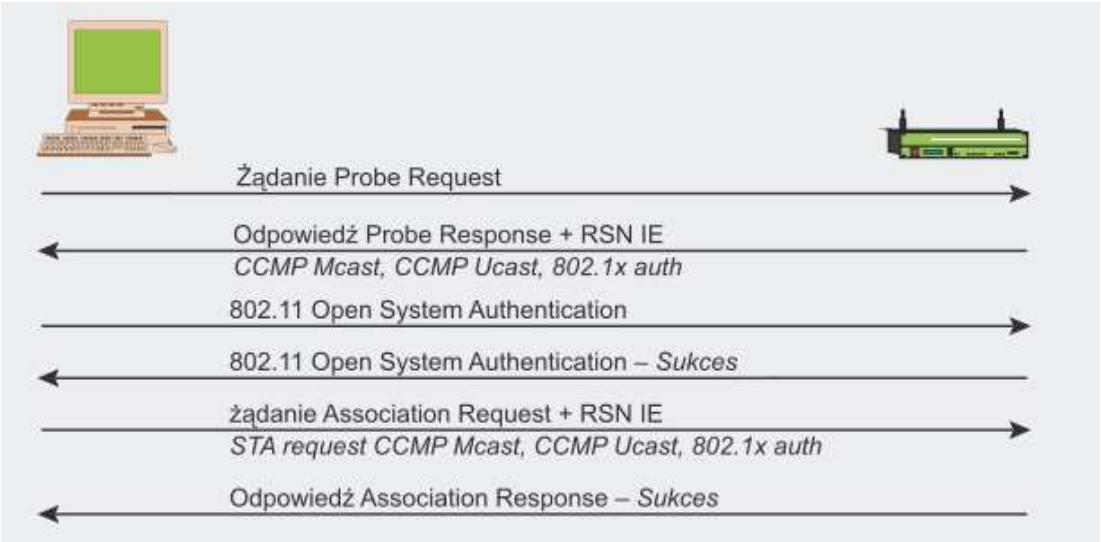
- uzgodnienie polityki bezpieczeństwa
- uwierzytelnianie 802.1X lub PSK
- generowanie i dystrybucja klucza
- zapewnienie integralności i poufności przesyłanych danych



FAZA 1: uzgodnienie polityki bezpieczeństwa

W ramach tego procesu ustalane są:

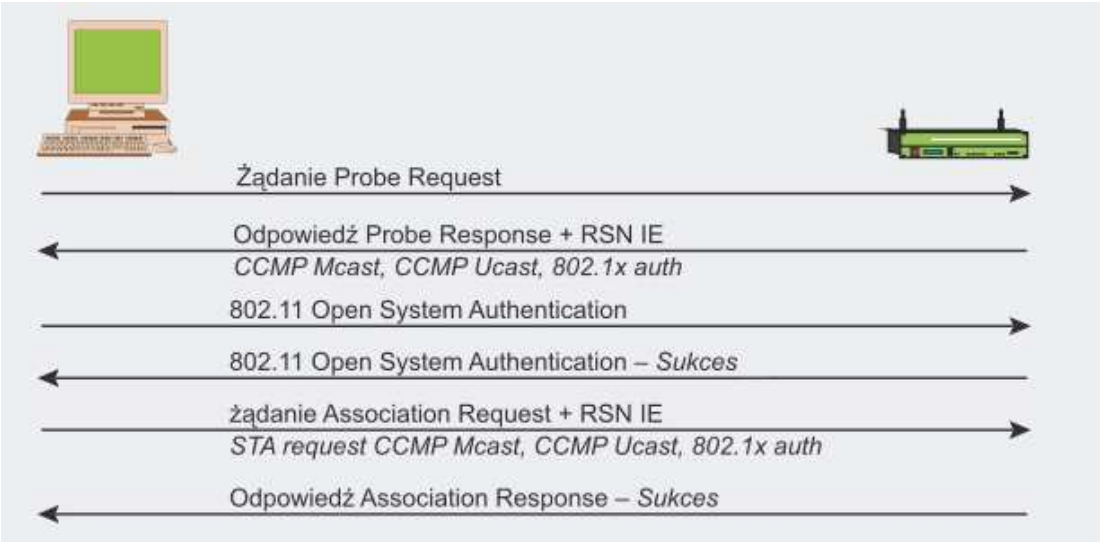
- metoda uwierzytelniania
- algorytm szyfrowania dla transmisji pojedynczej
- algorytm szyfrowania dla transmisji grupowej



FAZA 1: uzgodnienie polityki bezpieczeństwa

W ramach tego procesu ustalane są:

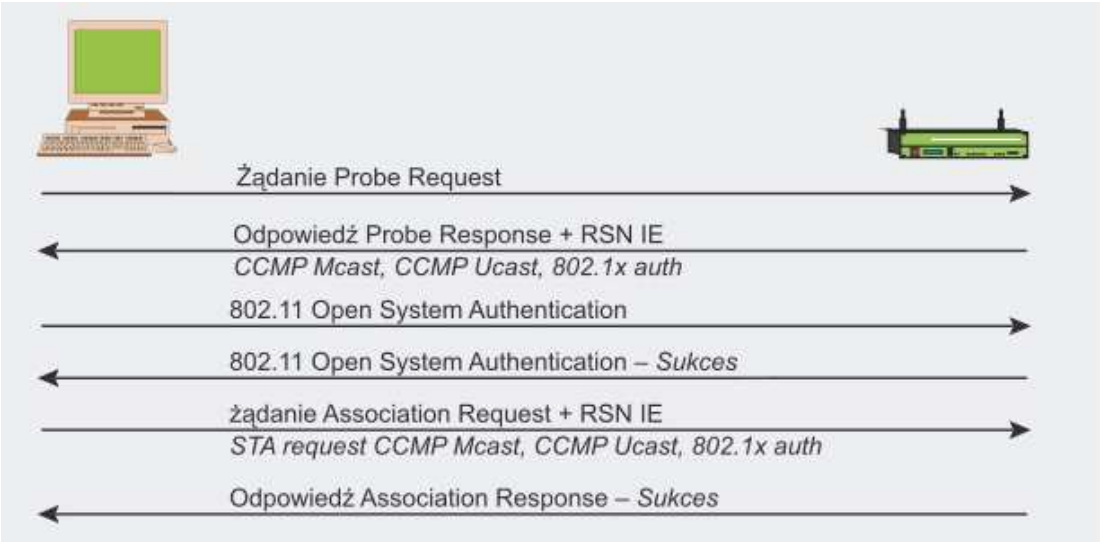
- metoda uwierzytelniania
- algorytm szyfrowania dla transmisji pojedynczej
- algorytm szyfrowania dla transmisji grupowej



FAZA 1: uzgodnienie polityki bezpieczeństwa

W ramach tego procesu ustalane są:

- metoda uwierzytelniania
- algorytm szyfrowania dla transmisji pojedynczej
- algorytm szyfrowania dla transmisji grupowej



FAZA 2: uwierzytelnianie

w standardzie 802.11i są dwie metody uwierzytelniania

- PSK - dla SOHO (small office/home office)
- 802.1X

FAZA 2: PSK (Pre-Shared Key)

- PSK to 8-63 znaki, lub 256 bitów
- klucz jest współdzielony ale klucze używane dalej są różne dla różnych użytkowników
- uwierzytelnianiem zajmuje się AP (sprawdza czy klient sieci zna PSK)

FAZA 2: standard 802.1X

cele 802.1X

- dostarczenie schematu do:
  - uwierzytelniania
  - autoryzacji
  - kontroli dostępu
- dystrybucja klucza

metody w schemacie uwierzytelniania 802.1X

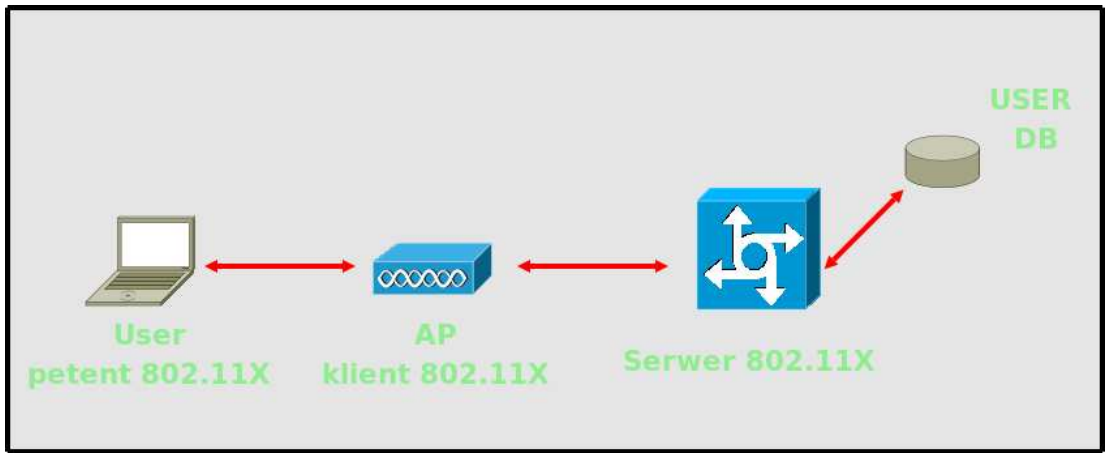
- EAP-TLS
- EAP-TTLS
- PEAP
- Kerberos V5
- EAP-SIM

FAZA 2: standard 802.1X

architektura 802.1X

architektura 802.1X obejmuje trzy podmioty:

- petent 802.1X (supplicant)
- klient 802.1X
- serwer 802.1X

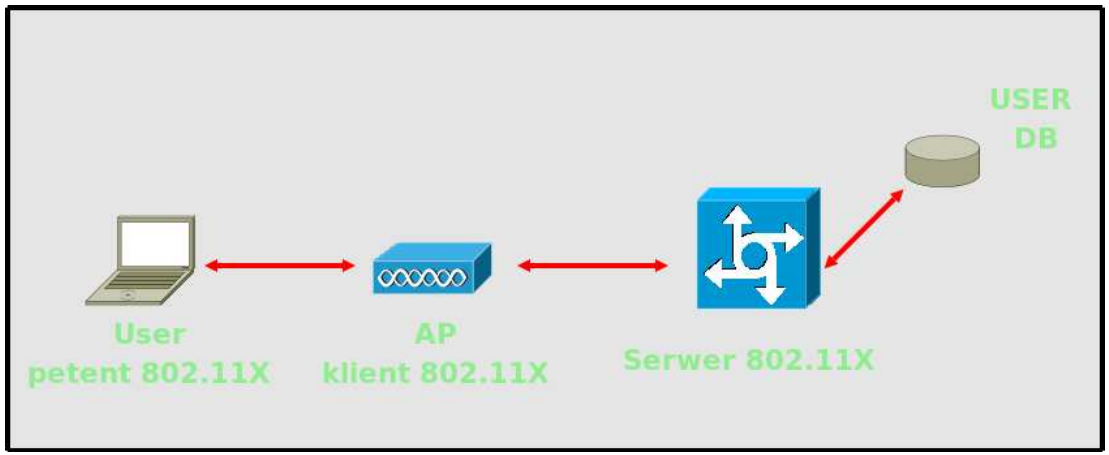


FAZA 2: standard 802.1X

komunikacja 802.1X

komunikacja 802.1X:

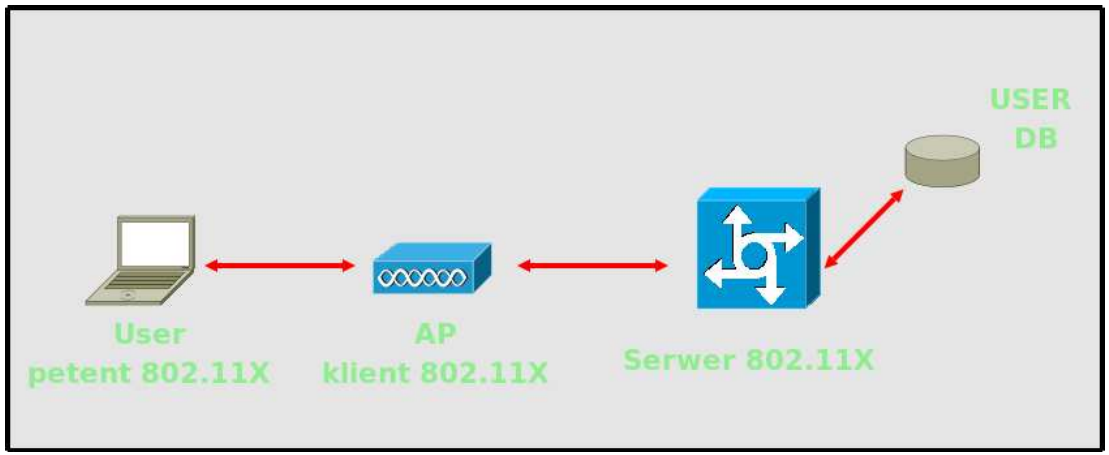
- komunikacja petent - klient - protokół EAP
- komunikacja klient - serwer - protokoł EAPoL



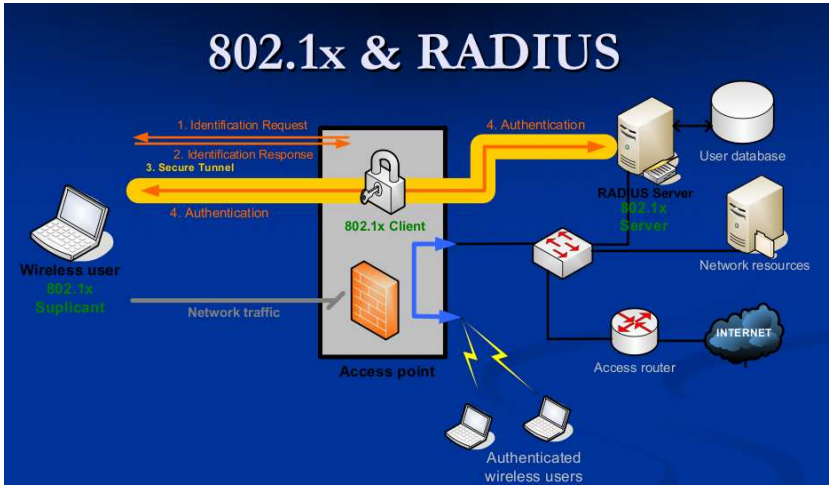
FAZA 2: standard 802.1X

rezultat 802.1X

- każda metoda uwierzytelniania kończy się tym że petent i klient mają ustalony tajny klucz nadrzedny (PMK)

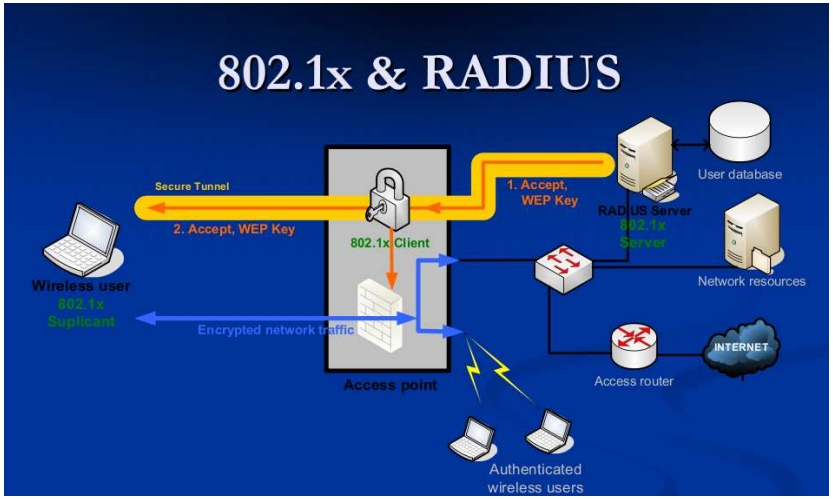


FAZA 2: standard 802.1X



- klient przekazuje tylko ruch klienta związany z procesem uwierzytelniania
- klient wysyła rządanie identyfikacji
- petent odpowiada
- ustanawiany jest bezpieczny tunel między petentem a serwerem
- serwer uwierzytelnia się jeśli tego wymaga petent
- petenet wysyła dane uwierzytelniające
- serwer sprawdza tożsamość użytkownika

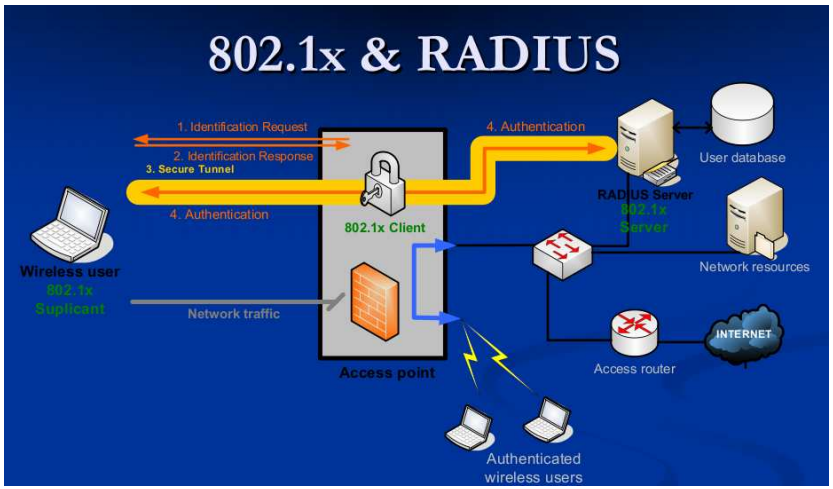
FAZA 2: standard 802.1X



jeśli uwieżytelnianie powiodło się

- serwer wysyła informacje o sukcesie procedury do petenta i klienta
- serwer wysyła klientowi klucz MK do komunikacji z petentem (petent także ma ten klucz MK)

FAZA 2: standard 802.1X



jeśli uwieżytelnianie nie powiodło się

- serwer wysyła informacje o niepowodzeniu procedury do petenta i klienta

FAZA 3: generowanie i dystrybucja klucza

klucze

- ograniczony czas ważności tajnych kluczy
- hierarchia kluczy

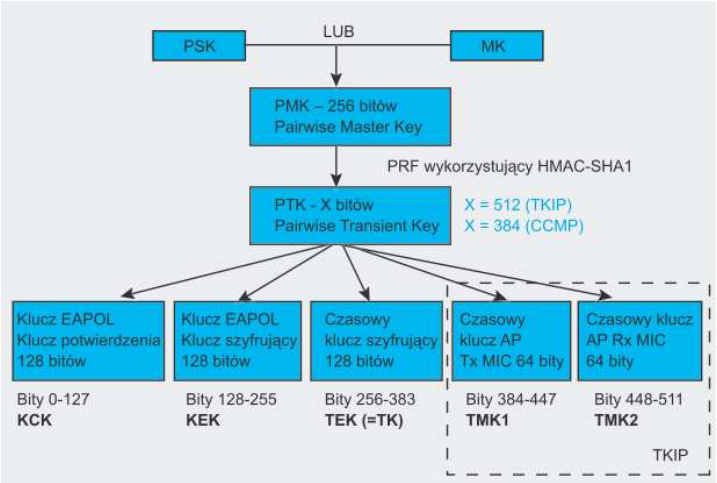
generowanie PMK (Pairwise Master Key)

- jeśli uwierzytelnianie przez PSK to PMK = PSK
- jeśli uwierzytelnianie przez 802.1X to PMK = JakaśFunkcja(MK)

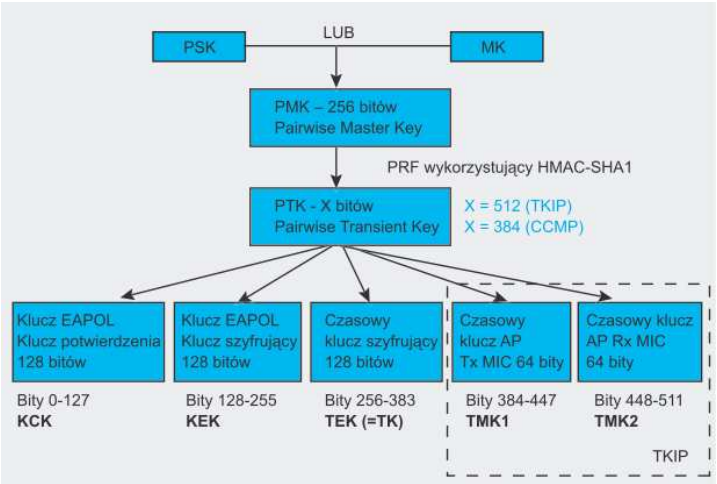
FAZA 3: generowanie i dystrybucja klucza

klucz PTK

- generowany na podstawie PMK, PTK = JakaśFunkcja2(PMK, InneArgumenty) - długość zależna od metody szyfrowania
- InneArgumenty są poznawane przez strony w ramach negocjacji czteroetapowej
  - 512 bitów - TKIP
  - 384 bity - CCMP



FAZA 3: generowanie i dystrybucja klucza



podział PTK

używane w ramach negocjacji czteroetapowej i w ramach negocjacji klucza grupowego

- KCK (Key Confirmation Key) - do generowania kodu uwierzytelniającego
- KEC - do szyfrowania danych

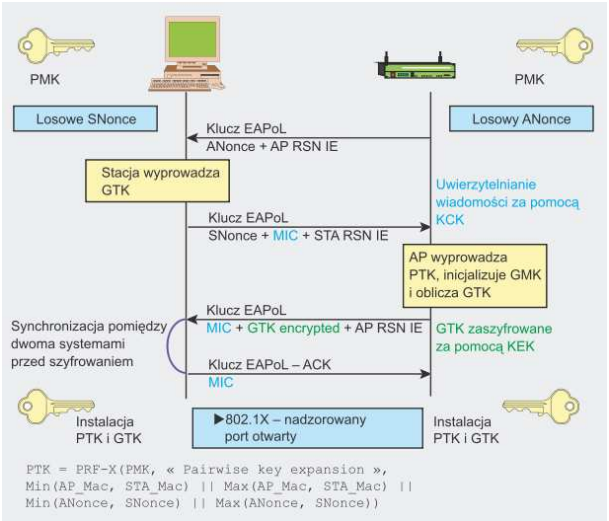
używane w ramach przesyłania normalnych danych

- TK - do szyfrowania danych
- TMK - używany tylko w TKIP - do generowania kodu uwierzytelniającego - każda strona ma swój klucz

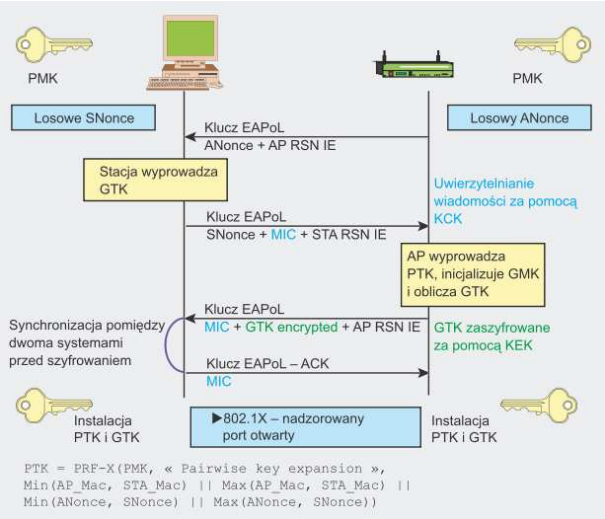
FAZA 3: generowanie i dystrybucja klucza - negocjacja czteroetapowa

Negocjacja czteroetapowa ma na celu

- potwierdzenie że klient faktycznie zna klucz PMK
- wygenerowanie nowego klucza PTK
- szyfrowanie transportu klucza GTK
- potwierdzenie zestawu wyboru szyfrów



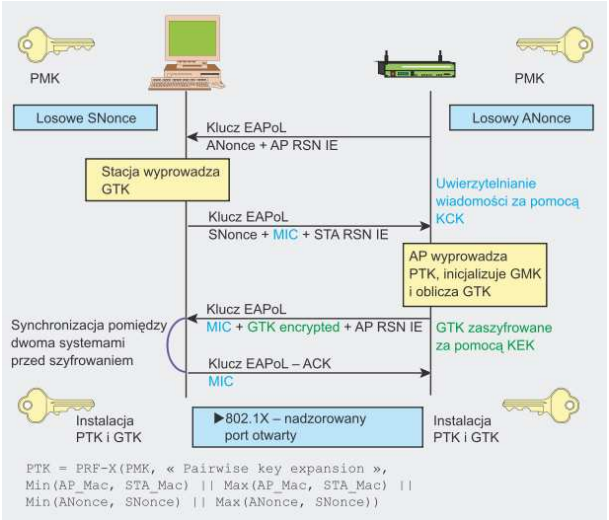
FAZA 3: generowanie i dystrybucja klucza - negocjacja czteroetapowa



klucz PTK jest wyliczany na podstawie

- klucza PMK
- stałego ciągu znaków
- MAC klienta
- MAC petenta
- wartości ANonce - wyznaczanej przez klienta
- wartości SNonce - wyznaczanej przez petenta

FAZA 3: generowanie i dystrybucja klucza - negocjacja czteroetapowa



cztery fazy negocjacji

- klient 802.1X losuja ANonce i wysyła(bez zabezpieczeń) ją petentowi
- petent losuje SNonce, oblicza PTK, liczy MIC, nie zaszyfrowane wysyła klientowi
- klient wylicza PTK, i sprawdza MIC, wylicza GTK, oblicza MIC i wysyła(zaszyfrowane) do petenta
- petent sprawdza MIC i wysyła ACK do klienta
- obydwie strony instalują klucze

FAZA 4: zapewnienie integralności i poufności przesyłanych danych

dostępne są dwa algorytmu, korzystające z kluczy wygenerowanych w fazie 3

- TKIP
- CCMP

TKIP

- stworzony aby zapewnić lepsze szyfrowanie starym urządzeniom (korzysta z algorytmów WEP)
- szyfr strumieniowy
- oparty na IV (48 bitowe)
- unika słabych IV
- do sprawdzania integralności danych stworzono nowy algorytm MIC
- wykrycie drugiego błędu MIC w przeciągu minuty, zawiesza komunikację na minutę (DoS) - generowane są nowe klucze

FAZA 4: zapewnienie integralności i poufności przesyłanych danych

dostępne są dwa algorytmu, korzystające z kluczy wygenerowanych w fazie 3

- TKIP
- CCMP

CCMP

- szyfr blokowy
- wykorzystuje szyfr AES (Rijndaeel)