WEP: przykład statystycznego ataku na źle zaprojektowany algorytm szyfrowania

Mateusz Kwaśnicki
Politechnika Wrocławska

Wykład habilitacyjny
Warszawa, 25 października 2012

Plan wykładu:

- Słabości standardu WEP
- Opis WEP i algorytmu RC4
- „Statystyczne” łamanie algorytmu RC4

Czym jest WEP

- W komputerowej lokalnej sieci kablowej wiadomości słyszą wszyscy użytkownicy podłączeni do sieci.
- W nowszych instalacjach: (teoretycznie) wyłącznie nadawca, odbiorca i urządzenie pośredniczące.
- W sieciach bezprzewodowych wiadomości podsłuchać może każdy.
- W 1999 roku ustalono standard WEP (ang. *Wired Equivalent Privacy*), który miał zapewnić poufność na poziomie sieci kablowych.
- Przy projektowaniu tego standardu popełniono wiele błędów.
- Obecnie standardem jest WPA2 (ang. *Wi-Fi Protected Access 2*), uznawany za bezpieczny.

Idea WEP (1)

- Wiadomości są szyfrowane za pomocą sekretnego klucza, znanego tylko osobom uprawnionym.
- Ten sam klucz jest wykorzystywany do uwierzytelnienia oraz szyfrowania.
- Szyfrowanie (algorytm RC4) jest symetryczne.
- Powód: względy wydajnościowe.

Pierwsza słabość

Problem 1

Wszystkie osoby znające sekretny klucz (wszyscy użytkownicy sieci) są w stanie odszyfrować wiadomości.

- Ten sam problem występuje w sieciach kablowych.
- Rozwiązanie: wykorzystać szyfrowanie asymetryczne lub tymczasowym kluczem.

Idea WEP (2)

- Na podstawie klucza generowany jest ciąg bitów X.
- Szyfrogramem wiadomości M jest $M \oplus X$ ($\oplus$ oznacza sumę dwójkową).
- Suma dwójkowa ma dużo własności algebraicznych.
- Do wiadomości zawsze jest dołączana suma kontrolna CRC32.
- Urządzenia sieciowe zwykle informują o wiadomościach ze złą sumą kontrolną.

Druga słabość

Problem 2

Można bajt po bajcie interakcyjnie odszyfrować wiadomość (atak „chop-chop”).

- Ten problem jest niezależny od sposobu generowania X (tj. algorytmu RC4).
- Rozwiązanie: stosować inny algorytm wyznaczania sumy kontrolnej.

Trzecia słabość

Problem 3

Algorytm RC4 nie jest bezpieczny!

- Rozwiązanie: stosować inny algorytm szyfrowania.
- Wszystkie trzy kwestie są rozwiązane w WPA2 (częściowo w WPA).

Historia ataków na WEP

- 2001: „**FMS**”
  Scott R. Fluhrer, Itsik Mantin oraz Adi Shamir wskazują błąd w RC4, który pozwala odtworzyć klucz WEP.
- 2004: „**KoreK**”
  Anonimowy autor *KoreK* istotnie usprawnia powyższą technikę.
- 2004: „**chop-chop**”
  KoreK ujawnia metodę „chop-chop”.
- 2007: „**PTW**”
  Erik Tews, Ralf-Philipp Weinmann oraz Andrei Pyshkin jeszcze usprawniają atak FMS.

Standard WEP

Oznaczenia:

- **Wiadomość** to ciąg bitów M.
- **Szyfrogram** to $M \oplus X$. („$\oplus$” — suma dwójkowa)
- X to **klucz**.
- $X = RC4(IV \cdot RK)$. („$\cdot$” — połączenie ciągów)
- IV to **wektor inicjalizujący** (24 bity), generowany dowolnie dla każdej wiadomości.
- RK to sekretny **klucz główny** (104 bity)

Algorytm:

- Nadawca wybiera IV, oblicza $X = RC4(IV \cdot RK)$ i przesyła $(IV, M \oplus X)$.
- Odbiorca wyznacza $X = RC4(IV \cdot RK)$ i oblicza $M = (M \oplus X) \oplus X$.

Obserwacja

- Jeśli znamy część wiadomości M i podsłuchamy $M \oplus X$, to możemy odtworzyć część X.
- Większość wiadomości ma ustaloną strukturę i możemy odgadnąć pierwsze bity M.
- Wszystkie wiadomości protokołów IP oraz ARP zaczynają się od 10101010!
- Możemy skłonić punkt dostępowy do generowania takich wiadomości!

Wniosek

Atakujący zna IV oraz pierwszych 8 bitów X dla każdej przesłanej wiadomości.

Algorytm RC4

- Przypomnijmy: $\mathtt{X} = \mathtt{RC4}(\mathtt{IV} \cdot \mathtt{RK})$.
- Oznaczmy $\mathtt{K} = \mathtt{IV} \cdot \mathtt{RK}$.
- Dzielimy K na bajty (bloki 8 bitów) i powtarzamy cyklicznie, aby uzyskać $\mathtt{K}_0, \ldots, \mathtt{K}_{255}$.
- Przez $\langle x \rangle$ oznaczamy resztę z dzielenia x przez 256.
- Algorytm RC4 składa się z dwóch części.

RC4 — część I

(1) Niech $k_0 = 0$ oraz $\sigma_0(i) = i$ dla $i = 0, \ldots, 255$.

(2) Dla $n = 0, \ldots, 255$ określamy

$$k_{n+1} = \left\langle k_n + \sigma_n(n) + \mathtt{K}_n \right\rangle,$$

$$\begin{cases} \sigma_{n+1}(k_{n+1}) = \sigma_n(n), \\ \sigma_{n+1}(n) = \sigma_n(k_{n+1}), \\ \sigma_{n+1}(i) = \sigma_n(i) & \text{dla pozostałych } i. \end{cases}$$

Algorytm RC4

- Po części I: σ_{256} jest permutacją $\{0, \ldots, 255\}$.

RC4 — część II

(1) Niech $j_{256} = 0$.

(2) Dla $n = 256, 257, \ldots$ określamy

$$j_{n+1} = j_n + \sigma_n\langle n+1\rangle,$$

$$\begin{cases} \sigma_{n+1}\langle j_{n+1}\rangle = \sigma_n\langle n+1\rangle, \\ \sigma_{n+1}\langle n+1\rangle = \sigma_n\langle j_{n+1}\rangle, \\ \sigma_{n+1}(i) = \sigma_n(i) & \text{dla pozostałych } i, \end{cases}$$

$$X_{n-256} = \sigma_{n+1}\Big\langle \sigma_{n+1}\langle n+1\rangle + \sigma_{n+1}\langle j_{n+1}\rangle \Big\rangle.$$

Przypomnienie

Cel

Poznanie klucza głównego RK. (Umożliwi to np. rozszyfrowanie wszystkich podsłuchanych wiadomości.)

Dla każdej podsłuchanej wiadomości znamy:

- pierwszy bajt klucza X, tj. X_0,
- ciąg inicjalizacyjny IV, tj. K_0, K_1, K_2 (bo $K = IV \cdot RK$).

Pokażemy, jak odgadnąć pierwszy bajt RK.
Wtedy dla każdej wiadomości znać będziemy:

- pierwszy bajt klucza X, tj. X_0,
- ciąg inicjalizacyjny IV, tj. K_0, K_1, K_2 (bo $K = IV \cdot RK$),
- pierwszy bajt RK, tj. $K_3 = RK_0$

i w analogiczny sposób odgadniemy RK_1, potem RK_2 itd.

Analiza RC4

RC4 — część I

$$k_0 = 0, \qquad \sigma_0(i) = i, \qquad k_{n+1} = \left\langle k_n + \sigma_n(n) + \mathsf{K}_n \right\rangle,$$

$$\begin{cases} \sigma_{n+1}(k_{n+1}) = \sigma_n(n), \\ \sigma_{n+1}(n) = \sigma_n(k_{n+1}), \\ \sigma_{n+1}(i) = \sigma_n(i) & \text{dla pozostałych } i. \end{cases}$$

- Umiemy odtworzyć pierwsze trzy kroki, tj. k_3 i σ_3.
- Pokażemy, że (wśród wybranych wiadomości) ponadprzeciętnie często $\mathsf{X}_0 = \sigma_4(3)$.
- Wtedy $\mathsf{X}_0 = \sigma_4(3) = \sigma_3(k_4)$.
- Stąd $k_4 = \sigma_3^{-1}(\mathsf{X}_0)$.
- Zatem poznamy $\mathsf{RK}_0 = \mathsf{K}_3 = \langle k_4 - k_3 - \sigma_3(3) \rangle$.
- Jeśli $\mathsf{X}_0 = \sigma_4(3)$, to wiadomość nazwiemy **istotną**.

Analiza RC4

RC4 — część I

$$k_0 = 0, \qquad \sigma_0(i) = i, \qquad k_{n+1} = \left\langle k_n + \sigma_n(n) + \mathsf{K}_n \right\rangle,$$

$$\begin{cases} \sigma_{n+1}(k_{n+1}) = \sigma_n(n), \\ \sigma_{n+1}(n) = \sigma_n(k_{n+1}), \\ \sigma_{n+1}(i) = \sigma_n(i) & \text{dla pozostałych } i. \end{cases}$$

- W n-tym kroku σ zmienia się na pozycji n i k_{n+1}.
- $k_4, \ldots, k_{256}$ symulują ciąg losowy.
- **Załóżmy**, że są losowe.
- Dla $i_0 \in \{0, 1, 2, 3\}$:

$$\mathbb{P}(\sigma_{256}(i_0) = \sigma_4(i_0)) = (1 - \tfrac{1}{256})^{252} \approx \tfrac{1}{e}.$$

Analiza RC4

RC4 — część I

$$k_0 = 0, \qquad \sigma_0(i) = i, \qquad k_{n+1} = \left\langle k_n + \sigma_n(n) + \mathsf{K}_n \right\rangle,$$

$$\begin{cases} \sigma_{n+1}(k_{n+1}) = \sigma_n(n), \\ \sigma_{n+1}(n) = \sigma_n(k_{n+1}), \\ \sigma_{n+1}(i) = \sigma_n(i) & \text{dla pozostałych } i. \end{cases}$$

- **Odrzucamy** wiadomość, jeśli $j := \sigma_3(1) \notin \{0, 1, 2\}$.
- Ponadto **odrzucamy** wiadomość, jeśli $\sigma_3(j) + j \neq 3$.
- Zostaje przeciętnie $\frac{3}{256} \cdot \frac{1}{256} \approx \frac{45}{1\,000\,000}$ wiadomości.
- $\mathbb{P}\left(\sigma_{256}(i) = \sigma_4(i) \text{ dla } i = 1, 3, j\right) \approx (\frac{1}{e})^3 \approx 5\%.$
- $\mathbb{P}(\sigma_3(1) = \sigma_4(1) \text{ oraz } \sigma_3(j) = \sigma_4(j)) = \frac{254}{256}.$

Analiza RC4

Podsumujmy:

- **nie odrzuciliśmy** ok. $\frac{45}{1\,000\,000}$ wiadomości;
- $j := \sigma_3(1) \in \{0, 1, 2\}$;
- $\sigma_3(j) + j = 3$;
- z prawdopodobieństwem ok. 4,9%:

$$\sigma_{256}(1) = \sigma_4(1) = \sigma_3(1) = j,$$
$$\sigma_{256}(j) = \sigma_4(j) = \sigma_3(j),$$
$$\sigma_{256}(3) = \sigma_4(3).$$

Analiza RC4

- Dla ok. 4,9% spośród **nieodrzuconych** wiadomości:

$$\sigma_{256}(1) = j, \quad \sigma_{256}(j) = \sigma_3(j), \quad \sigma_{256}(3) = \sigma_4(3)$$

oraz $j \in \{0, 1, 2\}$ i $\sigma_3(j) + j = 3$.

RC4 — część II

$$\begin{cases} \sigma_{257}(\sigma_{256}(1)) = \sigma_{256}(1), \\ \sigma_{257}(1) = \sigma_{256}(\sigma_{256}(1)), \\ \sigma_{257}(i) = \sigma_{256}(i) \qquad \text{dla pozostałych } i. \end{cases}$$

$$X_0 = \sigma_{257}\Big\langle \sigma_{256}(\sigma_{256}(1)) + \sigma_{256}(1) \Big\rangle.$$

- Zatem $X_0 = \sigma_{257}\langle \sigma_3(j) + j \rangle = \sigma_{257}(3)$.
- Skoro $\sigma_{256}(1) = j \neq 3$, $\sigma_{257}(3) = \sigma_{256}(3) = \sigma_4(3)$.
- Wobec tego wiadomość jest **istotna**.

Analiza RC4

Wniosek

Wśród **nieodrzuconych** wiadomości **istotnych** jest ok. 4,9%. Innymi słowy wzór:

$$\mathrm{RK}_0 = \left\langle \sigma_3^{-1}(X_0) - k_3 - \sigma_3(3) \right\rangle \qquad (\star)$$

zachodzi w ok. 4,9% przypadków.

(**Nieodrzuconych** wiadomości jest ok. $\frac{45}{1\,000\,000}$.)

Podobnie uzasadnia się, że dla pozostałych ok. 95,1% **nieodrzuconych** wiadomości prawa strona $(\star)$ ma mniej więcej równomierny rozkład na $\{0, \ldots, 255\}$.

Obserwując dostatecznie wiele wiadomości i badając empiryczny rozkład prawej strony $(\star)$, odgadujemy RK_0.

Analiza RC4

Uwagi:

- Wiadomości wykorzystane do odgadnięcia RK_0 można wykorzystać ponownie do odgadywania RK_1, RK_2 itd.
- Przy 5 000 000 wiadomości mamy ok. 50% szans powodzenia (wynik eksperymentalny).
- Ponieważ $k_4, \ldots, k_{256}$ nie są losowe, czasem atak zawodzi nawet przy dużej liczbie wiadomości.
- Usprawnione wersje powyższej techniki wymagają zaledwie ok. 50 000 wiadomości dla 95% szans powodzenia (wynik eksperymentalny).
- Do przeprowadzenia ataku wystarczy np. telefon z systemem *Android* i kilkanaście sekund.

Korzystałem z:


M. Beck, E. Tews
Practical attacks against WEP and WPA
Proc. WiSec ’09 (2009): 79–86


E. Tews
Attacks on the WEP protocol
Cryptology ePrint Archive, Report 2007/471 (2007)