

## Wykład 12. Schematy Wykluczania Zdrajców

*Wykładowca: Stefan Dziembowski**Skryba: Leszek Mierzejewski, Artur Cichocki*

*Streszczenie.*

*Wprowadzamy pojęcie schematów rewokacji. Przedstawiamy dwie metody rewokacyjne: Complete Subtree i Subset difference.*

## 1 Wstęp

Wykład oparty na [1].

Na dzisiejszym wykładzie podamy przykład protokołu z działu kryptografii zwanego *zaszyfrowanym rozgłoszeniem* (ang.: *broadcast encryption*). Ogólnie chodzi o to, żeby umożliwić *Centrum wydajne* rozgłaszanie informacji do dynamicznie zmieniającego się grona *odbiorców*. Według autorów publikujących w tej dziedzinie ma ona zastawowanie m.in. w:

1. płatna telewizja i radio
2. dystrybucja obrazu i dźwięku objętego prawami autorskimi (CD,DVD).

W przypadku 2 praktyczność wydaje się wątpliwa Lepiej wygląda 1.

Konkretnie, na dzisiejszym wykładzie będziemy zajmować się *schematami rewokacji*, tzn będziemy zakładać, że pierwotnie wszyscy odbiorcy są uprawnieni do odbierania informacji, ale niektórych z nich możemy chcieć w pewnym momencie *wykluczyć* (jest tak np. w sytuacji w której dany odbiorca sklonował swoje urządzenie odbiorcze i sprzedaje je na stadionie X-lecia). Będziemy więc konstruować *schemat wykluczania*.

Pożytecznym uzupełnieniem jest *mechanizm śledzenia zdrajców* (na następnym wykładzie).

Schematy, które będziemy omawiać są *bezstanowe*. Oznacza to, że odbiorca nie przechowuje informacji o wcześniejszej transmisji, ani nie zmienia swojego stanu w oparciu o nią. Dlatego też, wykonywane operacje mogą opierać się tylko na bieżącym przekazie i konfiguracji początkowej.

## 2 Definicje

Całkowitą liczbę odbiorców oznaczamy przez  $N$ . Zbiór wykluczonych odbiorców oznaczamy przez  $\mathcal{R}$ , a ich liczbę przez  $t$ .

Schemat rewokacji polega na wysłaniu przez centrum wiadomości  $M$  do wszystkich użytkowników, tak aby każdy użytkownik  $u \in N \setminus \mathcal{R}$  mógł odczytać poprawnie otrzymaną wiadomość, a wykluczeni użytkownicy nie. Dla utrudnienia zakładamy, że wykluczeni użytkownicy mogą tworzyć koalicje.

Schemat składa się z trzech części: (1) Inicjalizacja - przekazujemy każdemu użytkownikowi prywatną informację pozwalającą mu odszyfrować otrzymany przekaz. (2) Rozgłoszenie - dla danej wiadomości  $M$  i zbioru wykluczonych użytkowników produkujemy zaszyfrowaną wiadomość  $M'$  i wysyłamy ją do wszystkich użytkowników. (3) Odszyfrowanie - nie wykluczony użytkownik może na podstawie otrzymanej wiadomości  $M'$  i prywatnej informacji wyprodukować wiadomość  $M$ .

### 2.1 Schemat algorytmu

Niech  $S_1, \dots, S_w$  będą podzbiórami wszystkich odbiorców.  $S_1, \dots, S_w \in N$

Każdy podzbiór  $S_i$  ma przyporządkowany klucz  $L_i$ .

Każdy użytkownik  $u \in S_i$  może wyznaczyć  $L_i$  na podstawie swojej prywatnej informacji.

Staramy się pokryć zbiór  $N \setminus \mathcal{R}$  rozłącznymi podzbiórami  $S_{i_1}, \dots, S_{i_m}$

$$N \setminus \mathcal{R} = \cup_{j=1}^m S_{i_j}$$

Będziemy używali dwóch enkrypcji:

1.  $F_K : \{0, 1\}^* \rightarrow \{0, 1\}^*$  - szyfr strumieniowy. Dla każdej wiadomości  $M$  jest wybierany nowy klucz  $K$ , będący losowym ciągiem bitów.  $F_K$  jest szybką metodą, która nie rozciąga wysyłanej wiadomości. Najprostszą implementacją jest wykonanie  $XOR$  na  $M$  oraz strumieniu wygenerowanym za pomocą  $K$ .
2.  $E_{L_i} : \{0, 1\}^l \rightarrow \{0, 1\}^l$  Przekazujemy klucz sesyjny  $K$  do wszystkich niewykluczonych użytkowników. Przekazany klucz jest wykorzystywany wielokrotnie. Klucz sesyjny  $K$  jest szyfrowany  $m$  razy za pomocą  $L_{i_1}, \dots, L_{i_m}$

- Inicjalizacja

Każdy odbiorca  $u \in N$  dostaje prywatną informację  $I_u$ . Dla każdego  $1 \leq i \leq w$  użytkownik  $u \in S_i$  może wyprowadzić  $L_i$  na podstawie  $I_u$ .

Klucz  $L_i$  może być wybrany: (i) niezależnie od pozostałych, (ii) jako funkcja innych informacji, a więc może zależeć od innych kluczy.

- Rozgłoszenie

Centrum:

1. Wybiera klucz sesyjny  $K$ .
2. Znajduje podział zbioru  $N \setminus \mathcal{R}$  na podzbiory  $S_{i_1}, \dots, S_{i_m}$
3. Wysyła  $\langle [i_1, i_2, \dots, i_m, E_{L_{i_1}}(K), E_{L_{i_2}}(K), \dots, E_{L_{i_m}}(K)], F_K(M) \rangle$

$i_1, \dots, i_m$  - wyznaczają podział

$L_i$  - klucz powiązany z  $S_i$

[...] - nagłówek

$F_K(M)$  - ciało

- Rozkodowanie

Użytkownik otrzymuje przekaz

$\langle [i_1, i_2, \dots, i_m, C_1, C_2, \dots, C_m], M' \rangle$

1. Znajduje  $i_j$  takie, że  $u \in S_{i_j}$  (jeśli  $u \in \mathcal{R}$  rezultatem jest **null**).
2. Wyznacza klucz  $L_{i_j}$  za pomocą  $I_u$ .
3. Wylicza  $D_{L_{i_j}}(C_j)$ , aby uzyskać  $K$ .
4. Wylicza  $D_K(M')$ , aby uzyskać wiadomość  $M$ .

Implementacja schematu rewokacji składa się z (1) określenia kolekcji podzbiorów  $S_1, \dots, S_w$ , (2) wyznaczenia klucza powiązanego z każdym podzbiorem, (3) metody pokrycia nie wykluczonych odbiorców  $N \setminus \mathcal{R}$ , wykorzystującej rozłączne podzbiory z powyższej kolekcji, (4) metody pozwalającej użytkownikowi  $u$  znaleźć zbiór  $S_j$  i wyliczyć jego klucz  $L_{S_j}$  za pomocą  $I_u$ .

### 3 Wstęp do algorytmów rewokacyjnych

W obydwu algorytmach użytkownicy reprezentowani są jako liście pełnego drzewa binarnego o  $N$  liściach (dla ułatwienia zakładamy, że  $N$  jest potęgą 2). Takie drzewo zawiera  $2N - 1$  wierzchołków (liście i wewnętrzne wierzchołki) i dla każdego  $1 \leq i \leq 2N - 1$ ,  $v_i$  jest wierzchołkiem w drzewie.

Oznaczmy przez  $ST(\mathcal{R})$  (Steiner Tree) drzewo użytkowników  $\mathcal{R}$ , t.j. minimalne poddrzewo pełnego drzewa binarnego, które łączy wszystkie liście reprezentujące użytkowników z  $\mathcal{R}$ .

### 4 Metoda pełnych poddrzew (Complete Subtree Method)

Podzbiory  $S_1, \dots, S_w$  przedstawiamy jako poddrzewa w pełnym drzewie binarnym. Liśćmi są wszyscy użytkownicy. Węzeł  $v_i$  odpowiada podzbiorowi  $S_i$  zawierającemu wszystkie liście, których jest przodkiem.

Przyporządkowanie kluczy jest proste: wystarczy z każdym wierzchołkiem  $v_i$  związać niezależny i losowy klucz  $L_i$ . Każdy użytkownik dostaje  $\log N + 1$  kluczy, które znajdują się na ścieżce od korzenia do liścia  $u$ .

Niech  $u_1, \dots, u_r$  będą liśćmi odpowiadającymi użytkownikom wykluczonym  $\mathcal{R}$ . Podzielmy  $N \setminus \mathcal{R}$  na rozłączne podzbiory.

Niech  $v_1, \dots, v_m$  będą wszystkimi dziećmi wierzchołków z  $ST(\mathcal{R})$  nie należącymi do  $ST(\mathcal{R})$ . Odpowiadające im zbiory  $S_{i_1}$  pokrywają cały zbiór  $N \setminus \mathcal{R}$  i nic poza tym.

Rozmiar pokrycia: Szukany rozmiar pokrycia  $m$  jest równy liczbie wierzchołków w  $ST(\mathcal{R})$  stopnia jeden.  $ST(\mathcal{R})$  ma  $r$  liści. Wierzchołek w  $ST(\mathcal{R})$  to taki wierzchołek naszego pełnego drzewa, dla którego istnieje ścieżka do liścia reprezentującego użytkownika  $\mathcal{R}$ . Jest ich co najwyżej  $t \log N$ . Rozpatrzmy teraz wierzchołki, które są wielokrotnie liczone oraz te, które mają stopień równy 2 (czyli te, które nie produkują podzbioru  $N \setminus \mathcal{R}$ ).

**Twierdzenie 1** *Liczba podzbiorów pokrywających zbiór użytkowników  $N$ , w którym jest  $t$  wykluczonych, jest nie większa niż  $t \log(N/t)$*

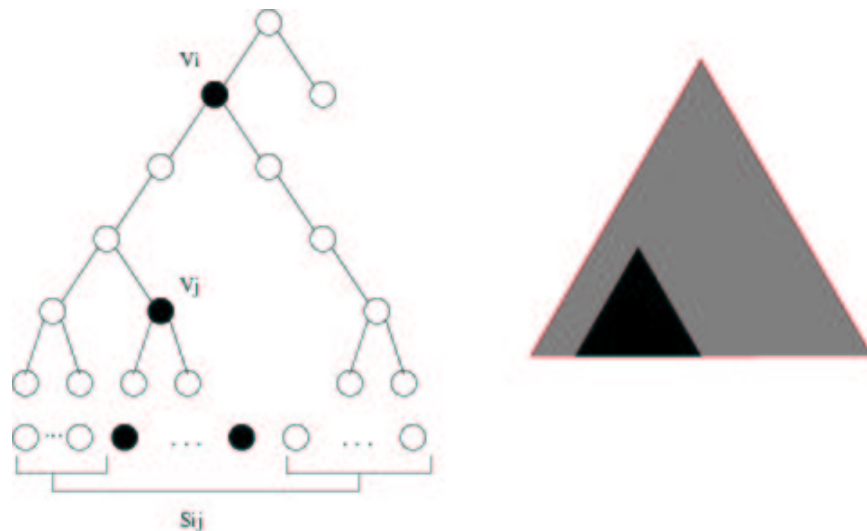
**Dowód:** Liczba podzbiorów jest równa liczbie wierzchołków w  $ST(\mathcal{R})$  stopnia 1. Udowodnimy przez indukcję, że twierdzenie jest prawdziwe dla drzewa o wysokości  $i$ ,

to znaczy, że dla poddrzewa  $N$ , które ma  $t$  'złych' liści, liczba wierzchołków stopnia 1 jest nie większa niż  $t(i - \log t)$ .

Rozpatrzmy drzewo o wysokości  $i + 1$ . Jeśli wszystkie liście zawarte są w jednym poddrzewie, to wykorzystując założenie indukcyjne mamy, że  $t(i - \log t) + 1 \leq t(i + 1 - \log t)$ . W przeciwnym przypadku liczba wierzchołków stopnia 1 jest sumą wierzchołków stopnia 1 prawego ( $t_1 \geq 1$ ) i lewego poddrzewa ( $t_2 \geq 1$ ) i  $t_1 + t_2 = t$ . Przez indukcję mamy, że jest ich nie więcej niż  $t_1(i - \log t_1) + t_2(i - \log t_2) = ti - (t_1 \log t_1 + t_2 \log t_2) \leq t(i + 1 - \log t)$ , ponieważ  $(t_1 \log t_1 + t_2 \log t_2) \geq t(\log t - 1)$ .  $\square$

## 5 Metoda Subset Difference

Przedstawiona w poprzednim rozdziale metoda pokrywając w najgorszym przypadku zbiór  $N \setminus R$  przez  $r \log N/r$  podzbiorów zwiększa dramatycznie długość wiadomości. Zaproponujemy nieco zmodyfikowany schemat, który ma tą przewagę nad poprzednim, że wystarczy mu  $2r - 1$  podzbiorów. W zamian zapłacimy większym obciążeniem pojedynczego odbiorcy, który będzie ewentualnie mógł należeć do  $O(N)$  podzbiorów, a nie jak poprzednio jedynie do  $\log N$ . Nową metodę nazwiemy *Subset Difference*.



Rysunek 1: Podzbiór  $S_{i,j}$  zawiera wszystkie zaznaczone liście

### 5.1 Opis podzbioru

Powtarzając pomysł z pierwszego schematu odbiorców traktujemy jako liście pełnego drzewa binarnego, jednak każdy poprawny podzbiór  $S$  jest definiowany przez dwa

węzły drzewa  $(v_i, v_j)$  takie, że  $v_i$  jest przodkiem  $v_j$ . Na oznaczenie takich podzbiorów będziemy stosować symbole  $S_{i,j}$ . Liść  $u$  należy do  $S_{i,j}$  wtedy i tylko wtedy, gdy jest potomkiem  $v_i$  i *nie* jest potomkiem  $v_j$ . Na rysunku 1 możemy zaobserwować przykładowy zbiór  $S_{i,j}$ . Na początek możemy zauważyć, że wszystkie podzbiory z poprzedniej metody są też podzbiorem tego schematu ( $v_i$  ojciec korzenia, a  $v_j$  jego brat). Wyjątkiem jest całe drzewo.

Na razie założymy, że każdy podzbiór  $S_{i,j}$  ma przypisany klucz  $L_{i,j}$ .

## 5.2 Pokrycie

$R$  jak poprzednio oznacza zbiór odbiorców wykluczonych, liście, które im odpowiadają nazwiemy  $u_1, u_2, \dots, u_r$ . Symbole  $S_{i_1, j_1}, S_{i_2, j_2}, \dots, S_{i_m, j_m}$  będą wskazywać rozłączne zbiory w sumie pokrywające  $N \setminus R$ .

### 5.2.1 Rozmiar pokrycia

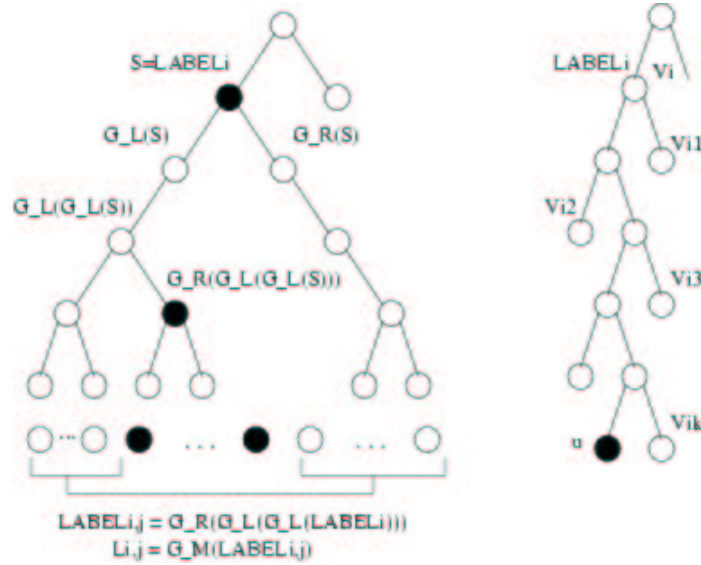
Lemat 2 przekonuje nas, że osiągnęliśmy swój cel.

**Lemat 2** *Dla dowolnego zbioru wykluczonych odbiorców  $R$ , schemat Subset Difference pokrywa  $N \setminus R$  co najwyżej  $2r - 1$  rozłącznymi podzbiorem.*

### 5.2.2 Przypisywanie kluczy do podzbiorów

Zastanowimy się teraz ile i jakie informacje musi przechowywać pojedynczy odbiorca, aby mógł uczestniczyć w wymianie wiadomości. Jeżeli przeniesiemy pomysł z poprzedniego schematu i odbiorca będzie przechowywał *explicite* klucze wszystkich podzbiorów, do których należy, to wymagania pamięciowe wzrosną o rząd wielkości. Dowolny odbiorca  $u$  dla każdego pełnego poddrzewa  $T_k$ , do którego należy, musiałby przechowywać liczbę kluczy proporcjonalną do ilości tych węzłów w poddrzewie  $T_k$ , które nie leżą na ścieżce z korzenia  $T_k$  do  $u$ . Takich poddrzew dla dowolnego  $u$  jest  $\log N$ , więc wszystkich wymaganych przez  $u$  kluczy jest  $\sum_{k=1}^{\log N} (2^k - k) = O(N)$ . Pójdziemy dalej i rozwiniemy inną metodę "przechowywania *implicite*" kluczy, która nakłada na odbiorcę obowiązek przechowywania dla każdego poddrzewa  $T_k$  zaledwie  $O(\log N)$  kluczy, co w sumie daje  $O(\log^2 N)$ .

Liczba wszystkich podzbiorów  $S$ , do których należy określony odbiorca  $u$ , jest ograniczona przez  $O(N)$ . Grupujemy te podzbiory  $S_{i,j}$  po pierwszej współrzędnej  $i$  i uzyskujemy  $O(\log N)$  klastrów. Każdemu wewnętrznemu wierzchołkowi  $1 \leq i \leq N - 1$  przypisujemy losową i niezależną etykietę  $LABEL_i$ . Sprawimy, aby ta wartość determinowała klucze dla wszystkich poprawnych zbiorów postaci  $S_{i,j}$ .



Rysunek 2: Przypisywanie kluczy w metodzie Subset Difference. Lewy: generowanie  $LABEL_{i,j}$  i  $L_{i,j}$ . Prawy: liść  $u$  otrzymuje etykiety  $v_{i_1}, v_{i_2}, \dots, v_{i_k}$ , które są określone przez etykietę  $LABEL_i$  wężła  $v_i$ .

Niech  $G$  będzie (*kryptograficznym*) *pseudolosowym generatorem*, który potraja wyjście. Powiemy, że  $G : \{0, 1\}^n \mapsto \{0, 1\}^{3n}$  jest pseudolosowym generatorem, jeżeli żaden wielomianowo ograniczony przeciwnik nie może odróżnić wyjścia  $G$  na losowym wejściu od zupełnie losowego ciągu takiej samej długości. Przez  $G_L(S)$  oznaczmy lewą trzecią część  $G(S)$ ,  $G_R(S)$  prawą, a  $G_M(S)$  środkową ( $G(S) = G_L(S)G_M(S)G_R(S)$ ).

Poetykietujemy pełne poddrzewo  $T_i$  rozpoczynając w jego korzeniu  $v_i$ , który ma przypisaną etykietę  $LABEL_i$ . Korzeń pozostawiamy przy swojej etykiecie, a następnie zstępująco etykietujemy każdy poziom po kolei. Jeżeli rodzic dostał etykietę  $S$ , to jego prawe i lewe dziecko etykietujemy odpowiednio  $G_L(S)$  i  $G_R(S)$ . Niech  $LABEL_{i,j}$  będzie etykietą wężła  $v_j$  uzyskaną w procesie etykietowania poddrzewa  $T_i$ . Od tej chwili kluczem  $L_{i,j}$  zbioru  $S_{i,j}$  będzie  $G_M(LABEL_{i,j})$ .

Przedstawiona procedura etykietowania ma tą własność, że jeżeli jest dana etykieta określonego wężła, to możemy odtworzyć etykiety wszystkich jego potomków. Z drugiej jednak strony nie mając etykiety któregoś z przodków danego wężła jego etykieta z naszego punktu widzenia jest pseudolosowa. Bardziej precyzyjnie, dla wężła  $v_j$  w drzewie  $T_i$ :

- Jeżeli dane są etykiety wszystkich wężłów za wyjątkiem przodków i potomków  $v_j$ , to  $LABEL_{i,j}$  jest nierozróżnialny od losowego.
- Jeżeli dane są etykiety wszystkich wężłów za wyjątkiem  $v_j$ , to klucz  $L_{i,j}$  jest pseudolosowy, choć etykieta  $LABEL_{i,j}$  nie jest pseudolosowa, ponieważ można

sprawdzić jej zgodność z etykietami potomków  $v_j$ .

Odnotujmy, że mając  $LABEL_i$  obliczenie  $L_{i,j}$  wymaga od nas wywołania  $G$  co najwyżej  $\log N$  razy.

Niezbędną dla odbiorcy  $u$  informacje do przeprowadzenia wyżej opisanej procedury znajdowania kluczy nazwiemy  $I_u$ . Oceniając, co na nią musi się składać, oszacujemy jej rozmiar. Dla każdego pełnego drzewa  $T_i$  takiego, że  $u$  jest liściem  $T_i$ , odbiorca  $u$  powinien móc obliczyć  $L_{i,j}$  wtedy i tylko wtedy, gdy  $j$  nie jest przodkiem  $u$ . Rozważmy ścieżkę z  $v_i$  do  $u$  i niech  $u_{i_1}, u_{i_2}, \dots, u_{i_k}$  będą kolejnymi "wiszącymi" węzłami na tej ścieżce, czyli incydentnymi ze ścieżką, ale nie będącymi przodkami  $u$  (prawy rysunek 2). Każdy  $j$  w  $T_i$ , który nie jest przodkiem  $u$ , jest potomkiem któregoś z tych węzłów. Dlatego jeżeli  $u$  otrzyma etykiety  $u_{i_1}, u_{i_2}, \dots, u_{i_k}$  jako część  $I_u$ , to wywołując  $G$  co najwyżej  $\log N$  razy obliczy  $L_{i,j}$  dla każdego  $j$ , który nie jest przodkiem  $u$ .

Każde pełne poddrzewo  $T_i$  głębokości  $k$ , które zawiera liść  $u$ , dodaje odbiorcy  $u$   $k - 1$  kluczy (plus jedno na wypadek braku wykluczonych), więc  $I_u$  jest nie większe niż:

$$1 + \sum_{k=1}^{\log N + 1} k - 1 = 1 + \frac{(\log N + 1) \log N}{2} = \frac{1}{2} \log^2 N + \frac{1}{2} \log N + 1$$

## Literatura

- [1] D Naor, M Naor, J Lotspiech *Revocation and Tracing Schemes for Stateless Receivers*, Crypto 2001.