# Kryptografia - projekt

Oskar Makowski 236554

21 czerwca 2020

## 1 Wstęp

Wybrany został projekt o numerze 9, dla wygody zostaje przytoczona jego treść: *Choose an NP-Hard problem and use it as signature scheme (start with finding a zero-knowledge proof and apply Fiat-Shamir heuristic)*. Jako problem NP-trudny, o który oparty zostanie protokół, przyjęty został problem 3-kolorowania grafu prostego. Zostanie przedstawiony dowód o wiedzy zerowej dla tego zagadnienia. W ostatniej części zaprezentowana zostanie transformacja Fiata-Shamira.

## 2 Problem 3-kolorowania

**Definicja 1** $NTIME(f)$ - zbiór maszyn Turinga pracujących niedeterministycznie na danych długości $n$ w czasie co najwyżej $f(n)$.

**Definicja 2** Klasa $NP = \bigcup_{k \in \mathbb{N}} NTIME(n^k)$

Innymi słowy, klasa $NP$ składa się z tych problemów, dla których istnieje algorytm działający w czasie wielomianowym na niedeterministycznej maszynie Turinga.

**Definicja 3** Problem $A$ należy do klasy $C$, jeśli istnieje maszyna Turinga $M$ rozwiązująca $A$ i $M \in C$.

**Definicja 4** Problem $A$ jest $C$-trudny, jeżeli każdy problem $B \in C$ redukuje się do $A$.

**Definicja 5** Problem $A$ jest $C$-zupełny, jeśli jest $C$-trudny i $A \in C$.

**Twierdzenie 1** Problem spełnialności formuł boolowskich $3-SAT$ jest $NP$-zupełny.

Twierdzenie pozostaje bez dowodu, jednak w ramach szkicu należy zaznaczyć, że $3-SAT$ można zredukować do problemu znalezienia ścieżki Hamiltona w grafie, który jest $NP$-zupełny.

**Definicja 6** Graf $G$ jest 3-kolorowalny, jeśli wszystkie wierzchołki mogą zostać pokolorowane na jeden z trzech kolorów w taki sposób, że żadne dwa wierzchołki o tym samym kolorze nie są do siebie incydentne (nie są połączone krawędzią).

**Twierdzenie 2** Dla danego poprawnego 3-kolorowania, permutacja kolorów daje poprawne kolorowanie.

Przykład: dla danych kolorów $\{czerwony, zielony, niebieski\}$, zamiana w każdym wierzchołku o kolorze czerwonym koloru na niebieski, zamiana pokrycia kolorem niebieskim na zielony, i finalnie przemalowanie koloru zielonego na czerwony, daje poprawne 3-kolorowanie.

**Dowód 1** Dla poprawnego 3-kolorowania, permutacja $\pi$ zamienia pary sąsiadujących kolorów z $(c_i, c_j)$ na $(\pi(c_i), \pi(c_j))$. Ponieważ w permutacji żadne dwa elementy $a, b$ nie przechodzą na ten sam element, tzn. $(\forall_{a,b} a \neq b)(\pi(a) \neq \pi(b))$, stąd $\pi(c_i) \neq \pi(c_j)$.

**Twierdzenie 3** Problem 3-kolorowania grafu jest $NP$-zupełny. Istnieje redukcja do problemu $3 - SAT$.

Szkic dowodu można znaleźć pod linkiem [1]. Wiadomo już, że zdefiniowany problem 3-kolorowaniu grafu prostego jest $NP$-zupełny. Teraz przedstawiony zostanie dla niego dowód o wiedzy zerowej.

**Definicja 7** Dowód z wiedzą zerową musi spełniać 3 warunki:

- Kompletność (*completness*) - jeżeli stwierdzenie jest prawdziwe, uczciwy weryfikujący zostanie przekonany przez uczciwego dowodzącego.
- Solidność (*soundness*) - jeżeli stwierdzenie jest fałszywe, oszukujący dowodzący przekona uczciwego weryfikatora z pomijalnie małym prawdopodobieństwem.
- Wiedza zerowa (*zero knowledge*) - jeżeli stwierdzenie jest prawdziwe, żaden z weryfikatorów nie pozyska żadnej wiedzy o stwierdzeniu innej niż ta, że jest prawdziwe.

Przedstawiony zostanie teraz protokół komunikacji między weryfikującym a dowodzącym. Niech dany będzie graf $G$, który na wejściu znany jest każdej ze stron. Dowodzący na wejściu posiada także świadka (ang. *witness*), tzn. poprawne 3-kolorowanie grafu $G$. Kolejne akcje wyglądają następująco:

- Dowodzący - niech $w$ będzie świadkiem rozwiązania danego problemu $NP$-trudnego. Niech $\pi$ będzie permutacją kolorów, stanowiącą nowe kolorowanie. Dowodzący zgłasza zobowiązanie (ang. *commitment*), zawierające kolorowanie wierzchołków: $(\forall v \in V)(c_i = COM(v_i, \pi(v_i)))$, które przesyła weryfikującemu.
- Weryfikator - wybiera krawędź $e_{ij}$, którą przesyła dowodzącemu.

- Dowodzący - otwiera zobowiązanie $c_i, c_j$ dla wierzchołków $i, j$ stanowiących koniec krawędzi $e_{ij}$.
- Weryfikator - zaakceptuj odpowiedź jeśli $c_i \neq c_j$. W przeciwnym razie odrzuć odpowiedź.

**Twierdzenie 4** Powyższy protokół spełnia warunki dowodu z wiedzą zerową.

**Dowód 2**

- Kompletność - jeżeli dowodzący posiada prawidłowe 3-kolorowanie grafu, wtedy dla dowolnej krawędzi poddanej weryfikacji $e_{ij}$, może zwrócić odpowiednie kolorowanie dla wierzchołków $v_i, v_j$ takie, że odpowiadające im kolory $c_i, c_j$ są różne, co weryfikujący zatwierdzi.
- Solidność - niech dowodzący posiada złe 3-kolorowanie. Oznacza to, że istnieje przynajmniej jedna krawędź $e_{ij}$ taka, że kolory $c_i, c_j$ na krańcach jej wierzchołków są takie same. Niech $A$ oznacza zdarzenie, że weryfikujący nie zaakceptuje odpowiedzi, $B$ niech będzie zdarzeniem, w którym weryfikujący wskazał krawędź $e_{ij}$. Wtedy

$$P(A) \geqslant P(B) = \frac{1}{|E|}$$

  gdzie $E$ jest zbiorem krawędzi grafu. Oznacza to, że po jednorazowym wykonaniu protokołu prawdopodobieństwo wykrycia oszustwa wynosi $\frac{1}{|E|}$. Jeżeli protokół zostanie przeprowadzony kilka razy, szansa błędu weryfikatora spadnie. Niech $C$ będzie zdarzeniem, w którym weryfikujący zaakceptował odpowiedz. Wtedy

$$P(C) \leqslant (1 - \frac{1}{|E|})^k$$

  co jest pomijalnie małe, gdzie w praktyce $k$ powinno być większe od $|E|$.
- Wiedza zerowa - (*szkic*) za każdym razem weryfikator poznaje jedynie permutację kolorowania dla losowo wybranej krawędzi i na tej podstawie nie pozyskuje wiedzy jak pokolorować cały graf. Po więcej szczegółów, np. jak zbudować symulator protokołu, można spojrzeć na [2], [3], [4].

Interaktywny protokół można zobaczyć pod linkiem [5].

# 3 Transformacja Fiata-Shamira

Omawiana transformacja (czy też heurystyka), służy do przekształcenia interaktywnych dowodów z wiedzą zerową składających z kilku kroków, w wersję probabilistyczną, gdzie *challenge* pochodzący od weryfikującego, który ma potwierdzić autentyczność dowodzącego, został zastąpiony przez pseudolosowe źródło będące publicznie dostępną funkcją haszującą $H$ (na stronie [6] pokazany jest przykład interpretacji pseudolosowej wartości hasza jako realizacji zmiennej losowej o rozkładzie jednostajnym na odcinku $(0, 1)$). Przykład dla problemu logarytmu dyskretnego można zobaczyć na stronie [7]. Dla wybranego zagadnienia 3-kolorowania grafu transformacja protokołu pokazanego w poprzednim rozdziale prezentuje się tak:

- Dowodzący składa analogiczne zobowiązanie, niech zostanie ono oznaczone przez $\Sigma_1$.
- Wybór krawędzi do weryfikacji, oznaczony przez $\Sigma_2$, standardowo realizowany przez weryfikującego, przeprowadzany jest w następujący sposób $\Sigma_2 = H(\Sigma_1)$, gdzie wartość hasza jest następnie interpretowana jako liczba całkowita modulo $|E|$, która mapowana na zbiór krawędzi wyznacza dokładnie jedną krawędź (wcześniej można ponumerować krawędzie grafu).
- Ponieważ funkcja $H$ jest publiczna, całość obliczenia może zostać przeprowadzona przez dowodzącego, który zwraca $\Sigma_1$ - zobowiązanie, $\Sigma_2$ - wybór krawędzi, $\Sigma_3$ - otwarcie odpowiedniego zobowiązania z kolorami wierzchołków.

# 4 Wersja alpha

Zaimplementowany został przedstawiony protokół w wersji interaktywnej. Dla uproszczenia zafiksowany został graf Petersena, dla którego znane jest 3-kolorowanie. Wersja alpha pozbawiona jest komunikacji, wszystko dzieje się w obrębie jednej klasy, gdzie kolejne rundy działania protokołu przebiegają w pętli, aż do uzyskania satysfakcjonującego poziomu pewności dotyczącego uczciwości dowodzącego. Dowodzący dokonuje permutacji kolorowania, które funkcjonuje jako zobowiązanie i na przestrzeni przebiegu jednej iteracji nie może zostać zmienione. Następuje wybranie krawędzi i porównanie kolorów wierzchołków znajdujących się na jej krańcach. Jeżeli są różne wzrasta poziom ufności wobec realnej możliwości posiadania 3-kolorowania, w przeciwnym przypadku wiadomo, że doszło do kontaktu z oszukującym dowodzącym.

Dla grafu Petersena po 45 interaktywnych wykonaniach protokołu, poziom pewności przekroczył wymagane 95%.

## Literatura


[1] http://cs.bme.hu/thalg/3sat-to-3col.pdf.

[2] https://www.cs.cmu.edu/~goyal/s18/15503/scribe_notes/lecture23.pdf.

[3] http://www.cs.cornell.edu/courses/cs6830/2011fa/scribes/lecture18.pdf.

[4] https://www.cs.princeton.edu/courses/archive/fall07/cos433/lec16.pdf.

[5] http://web.mit.edu/~ezyang/Public/graph/svg.html.

[6] https://ki.pwr.edu.pl/lemiesz/AA/counting.pdf.

[7] https://en.wikipedia.org/wiki/Fiat%E2%80%93Shamir_heuristic.