Laboratorium nr 8

Kryptografia oparta o funkcje skrótu

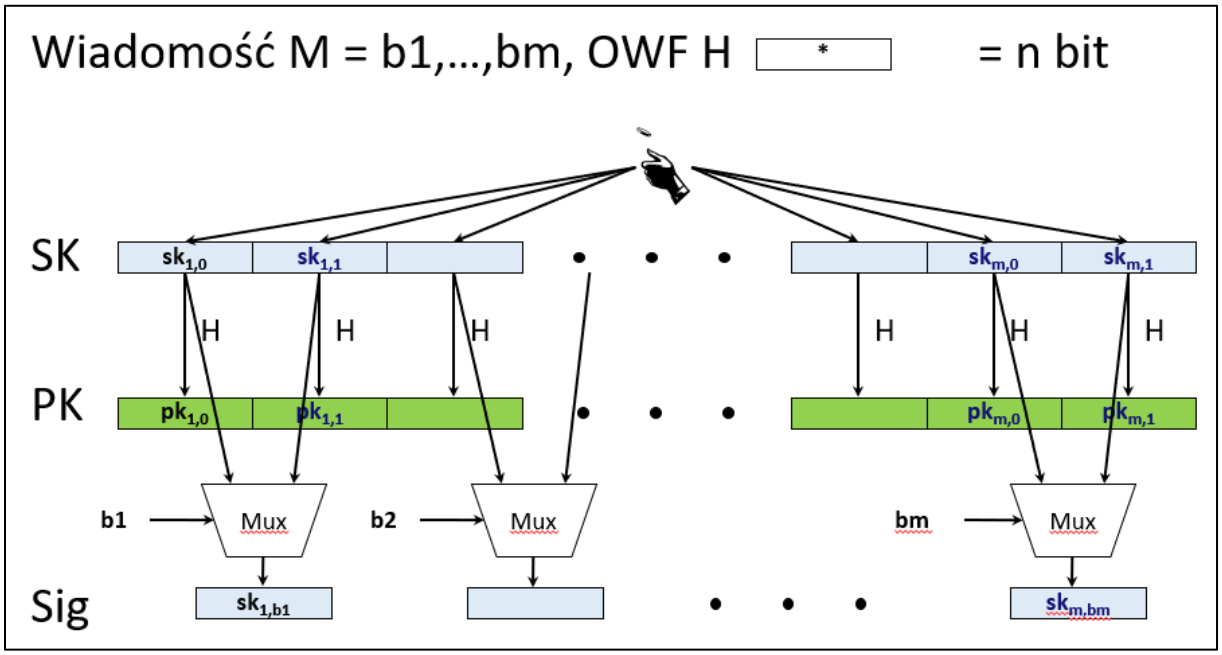
Termin wykonania: **do laboratorium nr 8**
Liczba punktów: **3 + 2 + 3**
PRK: T-L-7

1. Opis laboratorium

Celem laboratorium jest przebadanie podstawowych koncepcji wykorzystywanych w kryptografii opartej o skróty (*ang. hash-based cryptography*). Jest to jedna z dziedzin kryptografii należącą do tzw. kryptografii postkwantowej. Za jej pomocą można tworzyć schematy podpisu odporne na adwersarza posiadającego komputer kwantowy. Bezpieczeństwo tych schematów bazuje wyłącznie na bezpieczeństwie wybranej funkcji skrótu. Główną wadą jest albo duży rozmiar pary kluczy i podpisu albo ograniczenie podpisów możliwych do wykonania z użyciem jednej pary kluczy. W ramach laboratorium przebadamy pierwsze zaproponowane schematy, które obecnie nie są bezpośrednio używane, ale na ich koncepcjach są oparte aktualne schematy tj. XMSS, GMSS, czy SPHINCS.

Leslie Lamport zaproponował jednorazowy podpis oparty o funkcje skrótu w 1979 roku [1]. W tym schemacie Alicja posiada 256 bitową kryptograficzną funkcję skrótu oraz bezpieczny generator liczb losowych.

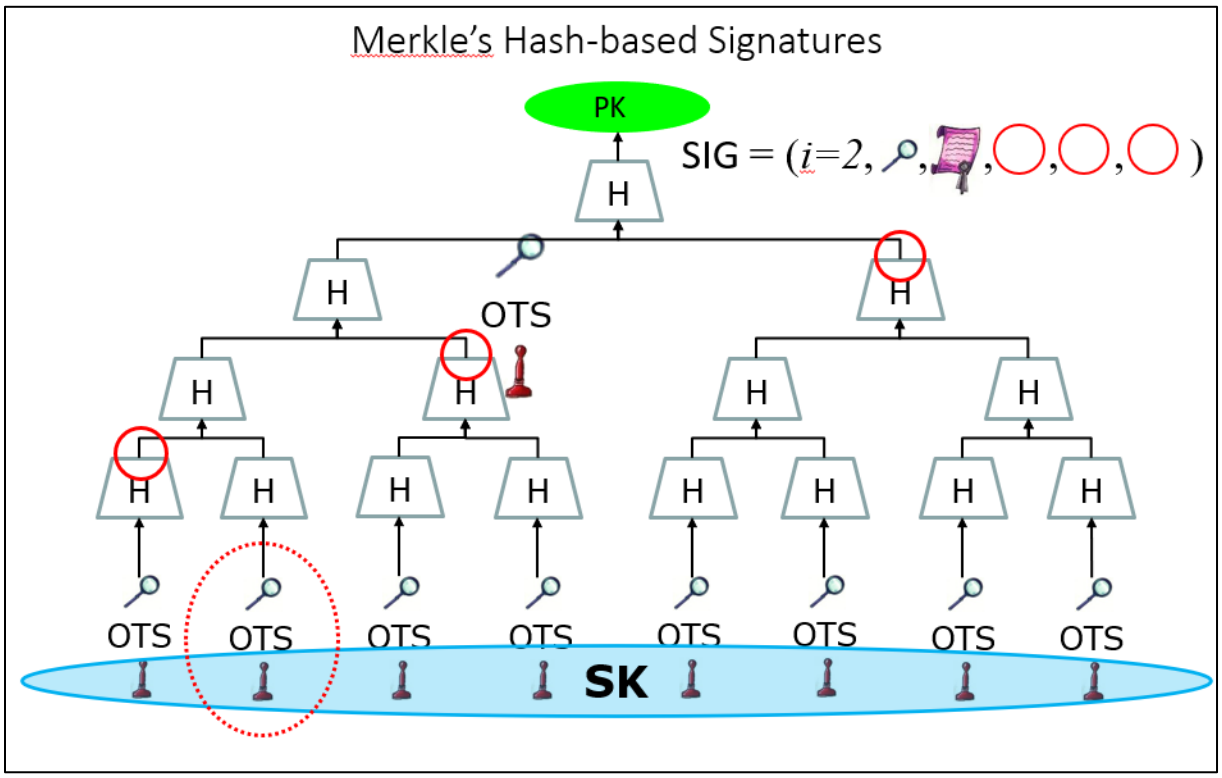
Tworzenie pary kluczy. Aby utworzyć klucz prywatny, Alice używa generatora liczb losowych do wytworzenia 256 par liczb losowych (w sumie 2×256 liczb), z których każda ma wielkość 256 bitów, czyli w sumie 2×256×256 bitów = 16 KB. To jest jej prywatny klucz i będzie go przechowywać w bezpiecznym miejscu do późniejszego użytku. W celu utworzenia klucza publicznego oblicza skrót z każdej z 512 losowych liczb w kluczu prywatnym, tworząc w ten sposób 512 skrótów, każdy o rozmiarze 256 bitów. Te 512 liczb tworzy jej klucz publiczny.



Źródło rysunku: https://pqcrypto2016.jp/data/Huelsing-20160223_pq_winter_school.pptx

Podpisywanie wiadomości. Alicja aby podpisać wiadomość oblicza 256 bitowy skrót z wiadomości. Następnie dla każdego bitu w skrócie, na podstawie jego wartości, wybiera jedną liczbę z odpowiednich par liczb tworzących jej klucz prywatny (tzn., jeśli bitem jest 0, wybieramy pierwszą liczbę, a jeśli bitem jest 1, wybieramy drugą). W ten sposób powstaje ciąg 256 liczb. Ponieważ każda liczba sama w sobie ma 256 bitów, całkowita wielkość podpisu będzie wynosiła 256×256 bitów = 8 KB. Te (pierwotnie wybrane losowo) liczby są jej podpisem i publikuje je wraz z wiadomością. Klucz prywatny Alice po użyciu nigdy nie może być użyty ponownie. Alicja musi zniszczyć pozostałe 256 skrótów, których nie użyła do podpisu. W przeciwnym razie, każdy dodatkowy podpis ponownie wykorzystujący klucz prywatny zmniejsza poziom bezpieczeństwa przeciwko przeciwnikom, którzy mogą później tworzyć z nich fałszywe podpisy.

Weryfikacja podpisu. Bob chce zweryfikować podpis Alicji. W tym celu oblicza również skrót z wiadomości. Następnie używa bitów ze skrótu, aby wybrać 256 skrótów w kluczu publicznym Alicji. Wybiera skróty w ten sam sposób, w jaki Alicja wybrała losowe liczby dla podpisu. Oznacza to, że jeśli pierwszym bitem skrótu wiadomości jest 0, wybiera on pierwszy skrót z pierwszej pary, i tak dalej. Następnie Bob oblicza skrót każdej z 256 losowych liczb w podpisie Alicji. To daje mu 256 skrótów. Jeśli te 256 skrótów dokładnie odpowiada 256 skrótom, które właśnie wybrał z klucza publicznego Alice, to podpis poprawny. Jeśli nie, to podpis jest błędny.



Źródło rysunku: https://pqcrypto2016.jp/data/Huelsing-20160223_pq_winter_school.pptx

Ralph Merkle zaproponował schemat umożliwiający rozszerzenie schematu podpisu jednorazowego Lamporta do schematu umożliwiającego wykonanie *k* podpisów. Schemat polega na wygenerowaniu *k* par kluczy (np. wg schematu Lamoporta), a następnie połączenie ich z użyciem drzewa skrótów. Dokładny opis znajduje się w [3].

Materiały pomocnicze:

1. https://en.wikipedia.org/wiki/Lamport_signature
2. https://pqcrypto2016.jp/winter/
3. https://en.wikipedia.org/wiki/Merkle_signature_scheme
4. https://www.ralphmerkle.com/papers/Thesis1979.pdf

2. Zadania do wykonania

Zadanie 1 (3 pkt) – Zaimplementuj podpis jednorazowy Lamporta (*Lamport One-Time Signature*):

a) (2 pkt do terminu lab 8) W aplikacji konsolowej zademonstruj działanie, pokaż, że podpis weryfikuje się poprawnie.
b) Jaki jest czas generowania pary kluczy, tworzenia podpisu i jego weryfikacji na twoim komputerze? Sprawdź dla funkcji skrótu SHA-2 o rozmiarach bloku 256, 384 i 512 (SHA256, SHA384, SHA512).
c) Jaki jest rozmiar klucza prywatnego, publicznego, a jaki podpisu?
d) Zaimplementuj rozwiązanie umożliwiające zmniejszenie rozmiaru przechowywanego klucza prywatnego. Klucz prywatny generowany jest z użyciem generatora pseudolosowego z ziarnem na wejściu. Zamiast przechowywać klucz, przechowujemy ziarno i odtwarzamy go z użyciem generatora pseudolosowego, gdy jest potrzebny.

2) **Zadanie 2** – (opcjonalne, 3 pkt) – Zaimplementuj schemat podpisu Merkla:

a) W aplikacji konsolowej zademonstruj działanie, pokaż, że podpis weryfikuje się poprawnie.
b) Wykonaj 100 operacji podpisywania różnych wiadomości z użyciem tej samej pary kluczy? Jaki musi być rozmiar klucza prywatnego/publicznego, aby możliwe było wykonanie takiej liczby podpisów?
c) Jak różni się czas tworzenia podpisu i jego weryfikacji w porównaniu do samego schematu Lamporta?