Rozdział 8

Schematy podpisów

W tym rozdziale analizujemy schematy podpisów, które nazywane są także podpisami cyfrowymi. Opisujemy różne schematy podpisów na podstawie problemów **rozkładu na czynniki** i **logarytmu dyskretnego**, w tym *Standard podpisu cyfrowego*.

8.1. Wprowadzenie

„Konwencjonalne" ręczne podpisy dołączane do dokumentów są używane, aby określić osobę odpowiedzialną za jego złożenie. Podpis jest używany w codziennych sytuacjach, takich jak pisanie listu, pobieranie pieniędzy z banku, podpisywanie kontraktu itp.

Schemat podpisu to metoda podpisywania komunikatu zapisanego w postaci elektronicznej. Jako taki podpisany komunikat może być transmitowany przez sieć komputerową, W tym rozdziale analizujemy kilka schematów podpisu, ale najpierw omawiamy niektóre podstawowe różnice między podpisem konwencjonalnym a cyfrowym.

Pierwsze pytanie dotyczy podpisywania dokumentu. Przy tradycyjnym podpisie podpis jest częścią podpisywanego dokumentu. Jednak podpis cyfrowy nie jest fizycznie dołączony do podpisywanego komunikatu, więc używany algorytm musi jakoś „powiązać" podpis z komunikatem.

Po drugie mamy kwestię weryfikacji podpisu. Tradycyjny podpis jest weryfikowany przez porównanie z innymi, autentycznymi podpisami. Na przykład, jeśli ktoś podpisuje zakup kartą kredytową (co dziś nie jest zbyt częste, biorąc pod uwagę uwiarygodnienie za pomocą technologii chipa i pinu), sprzedawca powinien porównać podpis na rachunku z podpisem na odwrocie karty kredytowej, aby zweryfikować podpis. Oczywiście nie jest to zbyt bezpieczna metoda, gdyż sfałszowanie czyjegoś podpisu jest dość łatwe. Z drugiej strony podpisy cyfrowe mogą zastać zweryfikowane za pomocą znanego publicznie algorytmu weryfikacji. Dlatego „każdy" może zweryfikować podpis cyfrowy. Wykorzystanie bezpiecznego podpisu zapobiega możliwości fałszerstw.

Inną podstawową różnicą między tradycyjnym i cyfrowym podpisem jest to, że „kopia" podpisanego komunikatu cyfrowego jest identyczna z oryginałem. Z drugiej strony kopię podpisanego dokumentu papierowego można zwykle odróżnić od oryginału. Ta cecha oznacza, że trzeba uważać, aby zapobiec ponownemu wykorzystaniu podpisanego dokumentu cyfrowego. Na przykład, jeśli Alicja podpisze komunikat cyfrowy autoryzujący Boba do pobrania 100 dolarów z jej konta bankowego (np. czek), chce, aby Bob mógł to zrobić tylko raz. Więc sam komunikat powinien zawierać informację, taką jak data, która zapobiega ponownemu jego użyciu.

Schemat podpisu składa się z dwóch składników: algorytmu podpisywania i algorytmu weryfikacji. Alicja może podpisać komunikat x za pomocą (prywatnego) algorytmu podpisywania $\mathbf{sig}_K$, który zależy od klucza prywatnego K. Wynikowy podpis $\mathbf{sig}_K(x)$ może być następnie zweryfikowany za pomocą publicznego algorytmu weryfikującego $\mathbf{ver}_K$. Mając parę (x, y), gdzie x jest komunikatem, a y podpisem dla x, algorytm weryfikacji zwraca odpowiedź *true* (prawda) lub *false* (fałsz) zależnie od tego, czy y jest poprawnym podpisem komunikatu x.

Oto formalna definicja tego schematu podpisywania.

DEFINICJA 8.1. *Schemat podpisu* jest pięcioelementową krotką $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, w której spełnione są następujące warunki:

1. $\mathcal{P}$ jest skończonym zbiorem możliwych *komunikatów*.
2. $\mathcal{A}$ jest skończonym zbiorem możliwych *podpisów*.
3. $\mathcal{K}$ jest *przestrzenią kluczy*, skończonym zbiorem możliwych *kluczy*.
4. Dla każdego $K \in \mathcal{K}$, istnieje *algorytm podpisujący* $\mathbf{sig}_K \in \mathcal{S}$ i odpowiadający mu *algorytm weryfikujący* $\mathbf{ver}_K \in \mathcal{V}$. Każdy $\mathbf{sig}_K : \mathcal{P} \to \mathcal{A}$ i $\mathbf{ver}_K : \mathcal{P} \times \mathcal{A} \to \{true, false\}$ to takie funkcje[1], że poniższe równanie jest spełnione dla każdego komunikatu $x \in \mathcal{P}$ i każdego podpisu $y \in \mathcal{A}$:

$$\mathbf{ver}_K(x, y) = \begin{cases} true, & \text{jeśli } y = \mathbf{sig}_K(x) \\ false, & \text{jeśli } y \neq \mathbf{sig}_K(x). \end{cases}$$

Para (x, y) przy $x \in \mathcal{P}$ i $y \in \mathcal{A}$ jest nazywania *podpisanym komunikatem*.

Dla każdego $K \in \mathcal{K}$ funkcje $\mathbf{sig}_K$ i $\mathbf{ver}_K$ powinny być wielomianowymi funkcjami czasu. Algorytm weryfikujący $\mathbf{ver}_K$ będzie publiczny, a algorytm podpisujący $\mathbf{sig}_K$ będzie prywatny. Dany komunikat x powinien być nierozwiązywalny obliczeniowo dla każdego poza Alicją, aby obliczyć podpis y taki, że $\mathbf{ver}_K(x, y) = true$ (i zauważmy, że może być więcej niż jedno takie y dla danego x, zależnie od tego, jak definiowana jest funkcja **ver**). Jeśli Oskar potrafi obliczyć parę (x, y) taką, że $\mathbf{ver}_K(x, y) = true$, a x nie był wcześniej podpisany przez Alicję, to podpis y jest nazywany *fałszerstwem*. Nieformalnie sfałszowany podpis jest poprawnym podpisem utworzonym przez kogoś innego niż Alicja.

8.1.1. Schemat podpisu RSA

Pierwszym przykładem schematu podpisu jest kryptosystem RSA, którego można używać do podpisów cyfrowych. W tym kontekście jest on znany jako schemat podpisu RSA. Kryptosystem 8.1 pokazuje „podstawową" wersję schematu, którą nieco dalej rozszerzamy.

[1] W niektórych schematach podpisu algorytm podpisujący jest wybierany losowo.

KRYPTOSYSTEM 8.1. Schemat podpisu RSA

Niech $n = pq$, gdzie p i q są liczbami pierwszymi. Niech $\mathcal{P} = \mathcal{A} = \mathbb{Z}_n$ i zdefiniujmy $\mathcal{K} = \{(n, p, q, a, b) : n = pq$, gdzie p i q są liczbami pierwszymi, $ab \equiv 1 \pmod{\varphi(n)}\}$. Wartości n i b są kluczem publicznym, wartości zaś p, q i a są kluczem prywatnym. Dla $\mathcal{K} = (n, p, q, a, b)$ definiujemy

$$\mathbf{sig}_K(x) = x^a \bmod n$$

i

$$\mathbf{ver}_K(x, y) = true \Leftrightarrow x \equiv y^b \pmod{n},$$

dla $x, y \in \mathbb{Z}_n$.

Zauważmy, że Alicja podpisuje komunikat x, korzystając z reguły odszyfrowywania RSA d_K. Alicja jest jedyną osobą, która może utworzyć podpis, gdyż $d_K = \mathbf{sig}_K$ jest prywatne. Algorytm weryfikacyjny używa reguły szyfrowania RSA e_K. Każdy może zweryfikować podpis, ponieważ e_K jest publiczne.

Zauważmy, że każdy może sfałszować podpis RSA Alicji, wybierając losowe y i obliczając $x = e_K(y)$. Wtedy $y = \mathbf{sig}_K(x)$ jest poprawnym podpisem dla komunikatu x. (Zauważmy jednak, że nie ma oczywistego sposobu wybrania najpierw x, a potem obliczenia odpowiadającego mu podpisu y. Jeśli można by to było zrobić, kryptosystem RSA nie byłby bezpieczny). Jednym ze sposobów zapobieżenia temu atakowi jest wymaganie, aby komunikat zawierał wystarczającą nadmiarowość, w której sfałszowany komunikat tego typu nie będzie odpowiadał „sensownemu" komunikatowi x, chyba że z bardzo małym prawdopodobieństwem. Alternatywnie wykorzystanie dwóch funkcji skrótu w połączeniu ze schematami podpisu wyeliminuje tę metodę fałszowania (kryptograficzne funkcje skrótu omawiane są w rozdziale 5). Omawiamy szerzej to podejście w kolejnym podrozdziale.

Pozostała część tego rozdziału jest zorganizowana w następujący sposób. Podrozdział 8.2 wprowadza pojęcia bezpieczeństwa schematów podpisu oraz sposoby, w jaki funkcje skrótu są używane w powiązaniu z tym schematami. Podrozdział 8.3 prezentuje schemat podpisu ElGamala i zawiera omówienie jego bezpieczeństwa. Podrozdział 8.4 dotyczy trzech ważnych schematów, które wyewoluowały ze schematu podpisu ElGamala, a konkretnie schemat podpisu Schnorra, algorytm podpisu cyfrowego oraz algorytm podpisu cyfrowego krzywej eliptycznej. Schemat podpisu, dla którego można udowodnić, że jest bezpieczny, zwany funkcją skrótu o pełnej dziedzinie, stanowi temat podrozdziału 8.5, w podrozdziale 8.6 zaś omawiane są certyfikaty. Wreszcie w podrozdziale 8.7 rozpatrywane są niektóre metody łączenia ze sobą schematów podpisu i schematów szyfrowania.