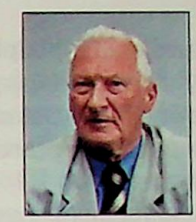
Brunon Hołyst, Jacek Pomykała



Bezpieczeństwo sygnatariuszy w komunikacji elektronicznej

Społeczeństwo informacyjne jest coraz bardziej uzależnione od współczesnych technologii telekomunikacyjnych i systemów informatycznych. Wśród dobrodziejstw, jakie niosą one dla ludzkości, trudno nie dostrzec, że są one także źródłem nowych zagrożeń oraz narzędziem wykorzystywanym przez środowiska przestępcze, szczególnie terrorystyczne. Przykładem jest cyberterroryzm, rozumiany jako atak na informację komputerową, skutkujący użyciem przemocy wobec celów niewalczących[1]. Powszechnie wiadomo, że współczesne grupy terrorystyczne wykorzystują do planowania i przeprowadzania ataków między innymi takie techniki informatyczne jak steganografie, kanały podprogowe czy systemy anonimizacji[2]. Te i wiele innych technik ochrony informacji stanowią przykłady bardzo dynamicznie rozwijającej się obecnie dziedziny nauki zwanej kryptologią. Jest ona oczkiem w głowie nie tylko krajowych i międzynarodowych agencji bezpieczeństwa, lecz także świata przestępczego i grup terrorystycznych.

Jednym z najbardziej zaawansowanych narzędzi kryptograficznych wszechobecnych w komunikacji elektronicznej jest podpis cyfrowy[3]. Jest on wykorzystywany między innymi jako metoda potwierdzenia integralności danych, uwierzytelnienia podmiotów komunikacji elektronicznej, autoryzacji dostępu do danych czy udowodnienie praw autorskich do wszelkiego rodzaju twórczości i wynalazczości na rynku usług elektronicznych[4]. Podstawowymi kryteriami przydatności podpisu cyfrowego są funkcjonalność i bezpieczeństwo, którym poświęcona jest praca B. Hołysta i J. Pomykały[5]. W odniesieniu do funkcjonowania podpisu na polskim rynku obowiązuje Ustawa o podpisie elektronicznym z dnia 18 września 2001 roku. Jeszcze w tym roku ma być gotowa jej nowelizacja, która stworzy warunki do upowszechnienia tego rozwiązania, a także umożliwi korzystanie z tzw. pieczątki elektronicznej[6]. Od pierwszego maja tego roku administracja państwowa ma już umożliwić przesyłanie podań na elektronicznych formularzach. W tym artykule akcentowane będzie głównie nie bezpieczeństwo samego schematu podpisu cyfrowego, lecz bezpieczeństwo jego sygnatariuszy. Taka perspektywa pokazuje nowe wyzwania (zazwyczaj niedostrzegalne w konwencjonalnych badaniach zagadnienia sygnatury cyfrowej) i nowe rozwiązania, którym w dużym stopniu poświęcona będzie dalsza część artykułu. Punkt wyjścia stanowią okoliczności składania podpisu. Z informatycznego punktu widzenia efekt złożenia podpisu dobrowolnego jest taki sam jak efekt złożenia podpisu w warunkach wymuszenia (groźba, szantaż).

W niniejszej pracy autorzy wykażą, że w istocie tak być nie musi i, w szczególności, omówiony zostanie nowy model podpisu cyfrowego, „wrażliwy" na okoliczności jego składania. Zastosowanie odpowiedniego protokołu może zwiększyć bezpieczeństwo sygnatariusza, a przede wszystkim ograniczyć jego odpowiedzialność za skutki prawne złożonego pod przymusem podpisu. Prezentowany tu pomysł przypomina trochę ideę zastosowania kanału podprogowego w podpisie cyfrowym[7]. Jednak kontekst i realizacja są zupełnie inne, gdyż informacja o wymuszeniu dociera do zaufanej strony trzeciej w trakcie deszyfrowania podpisu. W tradycyjnym ujęciu kanały podprogowe są natomiast wykorzystywane głównie jako źródło zagrożenia ze strony przeciwnika systemu (adwersarza)[8].

Kolejny wątek bezpieczeństwa sygnatariuszy podpisu cyfrowego dotyczy problemu ich anonimowości. Podpisy anonimowe stosowane są w przypadkach, gdy ujawnienie tożsamości podpisującego może być źródłem jego zagrożenia (np. ze strony przeciwników politycznych lub organizacji mafijnych), ale także dla zapewnienia ochrony informacji w związku z ustawą o ochronie danych osobowych. Inspirację do wielu przemyśleń w tym zakresie może stanowić praca Y. Desmedta i Y. Frankela[9] związana z ogólną koncepcją demokracji w procesach wymiany i uwiarygodnienia informacji, a także wzajemnego zaufania podmiotów komunikacji. Rozpraszanie uprawnień (i odpowiedzialności) indywidualnych podmiotów na grupy użytkowników pozwala tworzyć bezpieczne systemy podpisu, realizując przy tym wymóg wyższego stopnia zaufania stron (sygnatariuszy), a także wyższej tolerancji na ewentualne błędy przypadkowe lub wynikające z zamierzonych działań adwersarzy[10]. Szczególnie owocne są tu zastosowania w ramach tzw. podpisów progowych i grupowych[11] oraz sprawiedliwej wymiany informacji[12]. Zauważmy też, że bezpieczeństwo sygnatariuszy (w odróżnieniu od ochrony danych osobowych lub przeciwdziałania szantażom, wyłudzeniom czy oszustwom) jest we współczesnych transakcjach elektronicznych rozwiązywane często za pomocą tzw. ślepych

podpisów[13] oraz ślepych podpisów grupowych[14]. Stosowane metody uwiarygodniają klienta (sygnatariusza) jako członka pewnej grupy, chroniąc jednocześnie jego profile lub dane osobowe przed nieuczciwością innych użytkowników systemu.

Szczególnie aktualnym zagadnieniem związanym z bezpieczeństwem sygnatariuszy jest ryzyko utraty (kradzieży lub zgubienia) ich prywatnego klucza. W takim przypadku łączy się to z zagrożeniem utraty „tożsamości" sygnatariusza. Częściowym rozwiązaniem tego problemu są tzw. *forward secure signatures* – schematy, które uniemożliwiają (w przypadku przejęcia klucza prywatnego sygnatariusza) adwersarzowi podpisanie wiadomości „wstecz". Realizacja takiego podpisu jest oparta na procesie (okresowej) aktualizacji klucza prywatnego sygnatariusza za pomocą funkcji jednokierunkowej. Ogólny schemat podpisu, spełniającego takie wymagania, przedstawiony będzie w następnym rozdziale.

Celem tej pracy jest omówienie kluczowych idei i zastosowań praktycznych związanych z bezpieczeństwem sygnatariuszy – uczestników komunikacji elektronicznej. Zaproponowane koncepcje wymagają pewnej formalizacji matematycznej, służącej sprecyzowaniu ważnych pojęć, bez których zrozumienie tematu mogłoby zostać bardzo utrudnione. Odpowiednie przypisy pozwolą jednak na dopełnienie i szersze rozwinięcie prezentowanych tu idei kryptograficznych.

Podpis aktualizowany

Ten typ podpisu zapewnia sygnatariuszowi (w odróżnieniu od standardowej sygnatury cyfrowej) częściowe bezpieczeństwo w przypadku, gdy jego prywatny klucz zostanie skompromitowany (zaginięcie, kradzież lub przypadkowe ujawnienie). Mianowicie posiadacz skompromitowanego klucza prywatnego nie będzie w stanie podpisać w imieniu właściciela tych wiadomości, które pochodzą z okresu wcześniejszego. Dokładniej klucze prywatne do podpisu są aktualizowane co pewien czas, w sposób jednokierunkowy, więc fałszerstwo podpisu pod wiadomościami pochodzącymi z okresu poprzedniego wymagałoby odwrócenia funkcji „jednokierunkowej", co jest praktycznie niewykonalne. Należy zauważyć, że taki podpis jest zupełnie czym innym niż sygnatura elektroniczna z systemem znakowania czasu. Ta ostatnia jest technologicznie skomplikowana i wymaga kosztownej aktualizacji czasu w trybie on-line. Tutaj odbywa się ona co pewien okres zależny od wymagań przyjętych dla podpisu. Szczególnie interesująca jest wersja wielopodmiotowa, w przypadku której grupa podejmuje decyzje o aktualizacji klucza. Poniżej autorzy omówią szkicowo schemat podstawowy pochodzący od M. Bellare'a i S. Miner[15].

Autorzy artykułu zakładają, że dany jest multiplikatywny homomorfizm jednokierunkowy f, którego j-tą iterację oznaczamy przez f_j, oraz funkcja haszująca H. Przyjmujemy, że przeciwdziedziny dla f i H to: grupa multyplikatywna G, w której problem logarytmu dyskretnego jest trudny, oraz pierścień Z_q odpowiednio. Schemat podstawowy składa się z czterech algorytmów:

Algorytm generowania kluczy, który mając na wejściu parametry publiczne daje na wyjściu parę kluczy: $(s, f_T(s))$; tutaj $sk = s$ jest kluczem prywatnym, natomiast $pk = f_T(s)$ kluczem publicznym, gdzie T oznacza liczbę okresów aktualizacyjnych.

Algorytm aktualizacji kluczy, który mając na wejściu numer okresu i j-ty klucz prywatny $f_j(s)$ daje na wyjściu numer kolejnego okresu i wartości kolejnej aktualizacji $f(f_j(s)) = f_{j+1}(s)$.

Algorytm podpisywania, który mając na wejściu wiadomość m i j-ty klucz prywatny daje na wyjściu wartość podpisu równą $\mathrm{sig}_j(m) = (m, f_j(s)^{H(m, j)})$.

Algorytm weryfikacji podpisu, który dla pary: (podpis, klucz publiczny) = $(\mathrm{sig}_j(m), f_T(s))$ daje na wyjściu odpowiedź: akceptacja lub odrzucenie w zależności od tego, czy prawdziwa jest równość:

$$f_{T-j}(\mathrm{sig}_j(m)) = f_T(s)^{H(m, j)} = (pk)^{H(m, j)}.$$

Z uwagi na homomorficzność f widać, że poprawny podpis zawsze przechodzi weryfikację.

Słabością tego schematu może być fakt, że te same wiadomości podpisane w tym samym okresie dają tę samą wartość podpisu. W niektórych okolicznościach takie własności osłabiają bezpieczeństwo sygnatariuszy. Rozwiązaniem jest tu randomizacja podpisu polegająca na losowym wyborze parametru r_j angażowanego jednorazowo do podpisywania każdej wiadomości. Odpowiednia modyfikacja dotyczy wtedy procesu aktualizacji:

$$[r_j f_j(s), f_{T-j}(r_j)] \rightarrow [r_{j+1} f_{j+1}(s), f_{T-j-1}(r_{j+1})]$$

oraz podpisywania:

$$\mathrm{sig}_j(m) = [m, r_j f_j(s)^{H(m, j)}, f_{T-j}(r_j), j] := [m, a, b, j].$$

Podpis $[m, a, b, j]$ jest akceptowany wtedy i tylko wtedy, gdy:

$$f_{T-j}(a) = b(pk)^{H(m, j)}.$$

W przypadku schematu grupowego członkowie grupy składają swoje podpisy częściowe, które są następnie łączone w jeden bezpieczny podpis aktualizowany. Aby dostatecznie duża podgrupa całej grupy (tzn. mocy t lub większej) mogła wygenerować podpis w imieniu całej grupy, sekret s powinien być rozdzielony bezpiecznie w systemie (t, n) progowym. Takie wymagania realizuje protokół Shamira dzielenia sekretu[16], wykorzystujący dobrze znany wzór interpolacyjny Lagrange'a dla wielomianów.

Z uwagi na fakt, że podział Shamira jest addytywny, a w podpisie wykorzystujemy homomorfizm multiplikatywny f, należy dokonać transformacji sekretu s do $S = g^s$, gdzie g jest odpowiednio wybranym elementem z dziedziny homomorfizmu f (mającej strukturę grupy abstrakcyjnej). Wtedy w dowolnym podzbiorze B mocy $\geq t$ interpolacja nowego sekretu S jest multiplikatywna, tzn. S jest iloczynem $U(i)$ dla $i \in B$, $U(i) = g^{u(i)}$, gdzie $u(i) = u_B(i)$ to odpowiedni udział i-tego członka grupy (zależny od wyboru podgrupy B). Dalsze postępowanie jest analogiczne jak w przypadku podpisu indywidualnego. Podpis zrandomizowany częściowy dla i-tego członka w okresie j-tym ma postać:

$$\mathrm{sig}(i, j) = [m, r_j f_j(S_i), f_{T-j}(r_j), j].$$

Podpis złączony (w podgrupie B) w okresie j ma więc postać:

$$\mathrm{sig}(j) = [m, r_j f_j(S), f_{T-j}(r_j), j],$$

ponieważ:

$$\prod_{i \in B} f_j(U_i) = f_j\left(\prod_{i \in B} U(i)\right) = f_j(S).$$

Szczegółowo realizacja podpisu progowego aktualizowanego z homomorfizmem $f(x) = x^2$ jest przedstawiona w pracy M. Abdalla, S. Miner, C. Namprempre'a[17].

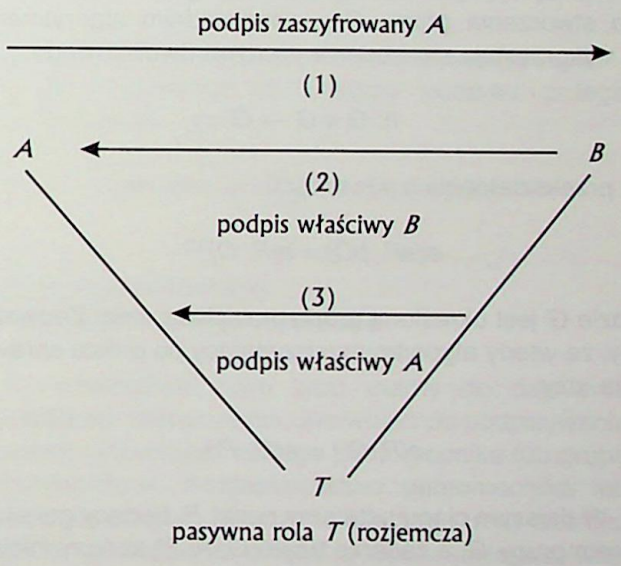
Podpis szyfrowany weryfikowalny

Ten typ podpisu jest z powodzeniem wykorzystywany w kontraktach cyfrowych. Występuje w nim dwóch sygnatariuszy A, B oraz zaufana trzecia strona T (która może odgrywać rolę zarówno pasywną, jak i aktywną). Przykładowo strona A przekazuje swoje prawa autorskie stronie B w zamian za określoną kwotę pieniężną.

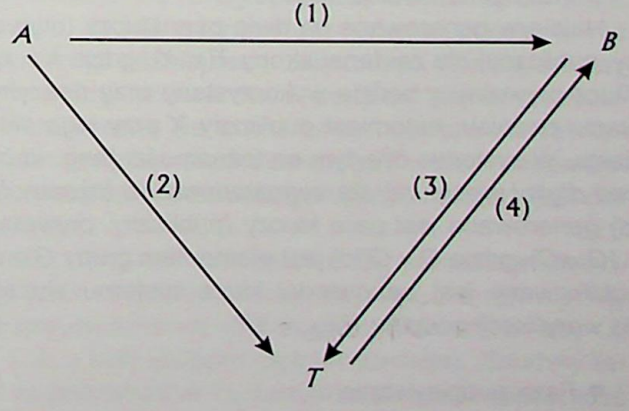
W przypadku aktywnej roli T, np. w transakcjach płatniczych, podpis zaszyfrowany może być podstawą do zablokowania określonych środków pieniężnych na koncie dłużnika lub wykonania przelewu z ich blokadą na koncie wierzyciela. Właściwy podpis jest wtedy podstawą albo do wykonania przelewu, albo do odblokowania odpowiednich środków na koncie wierzyciela.

Częściej wykorzystywanym typem kontraktu jest kontrakt z rolą T działającą w trybie off-line (tylko w przypadku konieczności rozstrzygnięcia sporów pomiędzy A i B). Możliwe są następujące podstawowe scenariusze:

1. Sygnatariusze A i B postępują uczciwie, tj. A wysyła podpis zaszyfrowany pod zawieranym kontraktem stronie B, następnie B wysyła właściwy podpis do A i ostatecznie A właściwy podpis do B.
2. Jedna ze stron wysłała podpis zaszyfrowany, nie otrzymując podpisu właściwego strony drugiej. W takim przypadku przekazuje go do strony T wraz z oskarżeniem przeciwnej strony.
3. Jeśli obie strony wysunęły podobne oskarżenia, strona T odszyfrowuje podpisy i przekazuje je odpowiednio stronom A i B.



Ryc. 1. Schemat graficzny pasywnej roli T
Fig. 1. *Model of T passive role*



Ryc. 2. Schemat graficzny dla aktywnej roli T
Fig. 2. *Model of T active role*

4. Jeśli jedna ze stron „milczy", T uznaje oskarżenie drugiej za uzasadnione i anuluje kontrakt powiadamiając o tym druga stronę.

Odnieśmy teraz przypadek aktywnej roli T do strony podpisującej A i strony weryfikującej B (ryc. 2). W kroku (3) weryfikujący zwraca się do T z żądaniem odszyfrowania podpisu, a w kroku (4) T zatwierdza podpis (odszyfrowując odpowiedni kryptogram).

Fazy podpisu szyfrowanego weryfikowalnego

- Faza inicjalizacji

W tej fazie tworzona jest grupa addytywna, cykliczna $G = (G,+,0)$ o elementach P, Q, R, w której problem obliczeniowy Diffie-Hellmana jest trudny[18]. Zakładamy, że istnieje wydajny algorytm A, który rozstrzyga, czy dwie pary (P, Q) i (P', Q') są kodowane tym samym kluczem s, tj. czy $Q = sP$ wtedy i tylko wtedy, gdy $Q' = sP'$ (nie znajdując jednak wartości samego s). W praktyce do stworzenia grupy G z odpowiednim algorytmem A wykorzystuje się istnienie iloczynu dwuliniowego:

$$e: G \times G \rightarrow G'$$

tj. przekształcenia o własności:

$$e(aP, bQ) = e(P, Q)^{ab}$$

gdzie G' jest określoną grupą multiplikatywną. Zauważmy, że wtedy algorytm rozstrzygający po prostu sprawdza, czy:

$$e(P, Q') = e(Q, P').$$

W dalszym ciągu ustalamy punkt P, będący generatorem grupy G, a zadanie trójki (G, A, P) kończy inicjalizację systemu.

- Faza generowania kluczy

Najpierw generowane są dwie pary kluczy (prywatny, publiczny) dla zaufanej strony $T(x, X)$, gdzie $X = xP$. Klucz prywatny x będzie wykorzystany przy deszyfrowaniu podpisu, natomiast publiczny X przy jego składaniu. W systemie opartym na tożsamości (ang. *Id-based digital signature*) dla sygnatariusza (o tożsamości Id) generowana jest para kluczy (publiczny, prywatny) $= (Q, sQ)$, gdzie $Q = Q(\mathrm{Id})$ jest elementem grupy G oraz publikowany jest odpowiedni klucz systemu służący do weryfikacji podpisu $P_{pub} = sP$.

- Faza podpisywania

Sygnatariusz generuje dwa losowe parametry r, v ze zbioru $q = |G|$-elementowego $\{0,1,..., q-1\}$ i oblicza podpis pod wiadomością m równy $[m, R, V, W]$, gdzie $R = rP$, $V = vP$, $W = rP_{pub} + h(m, R)sQ + vX := S + vX$, gdzie X jest kluczem publicznym strony T, S jest składnikiem odpowiedzialnym za właściwy podpis sygnatariusza, natomiast vX powoduje, że podpis staje się nieczytelny, ale umożliwiający sprawdzenie jego poprawności.

- Sprawdzenie poprawności podpisu

W tej fazie dowolny podmiot może na podstawie podpisu $[m, R, V, W]$ i kluczy publicznych P_{pub} i Q stwierdzić, że podpis zaszyfrowany jest poprawny. W tym celu sprawdza, że para $(R + h(m, R)Q, W)$ jest kodowana tym samym kluczem co (P, sP) oraz para (P, vX) tym samym kluczem co (P, V). Ponieważ wartość vX jest ukryta w wartości W, sprawdzenie odbywa się w oparciu o własność dwuliniowości iloczynu e, tj. że:

$$e(P, W) = e(P, S)e(P, vX) = e(P_{pub}, R + h(m, R)Q) \cdot e(V, X).$$

Pierwszy czynnik jest odpowiedzialny za weryfikację podpisu właściwego, a drugi za szyfrowanie podpisu kluczem publicznym strony T.

- Odszyfrowanie podpisu

W tej fazie istotną rolę odgrywa strona T, która na podstawie podpisu $[m, R, V, W]$ (stosując odpowiedni klucz prywatny x) oblicza wartość xV i udostępnia właściwy podpis $[m, R, S]$, gdzie $S = W - xV$ odpowiedniemu weryfikującemu.

- Weryfikacja właściwego podpisu

Dowolny podmiot przeprowadza weryfikację podpisu $[m, R, S]$ za pomocą kluczy publicznych P_{pub} i Q, sprawdzając, czy para $(R + h(m, R)Q, S)$ jest kodowana tym samym kluczem co (P, P_{pub}).

Podpis odporny na wymuszenia

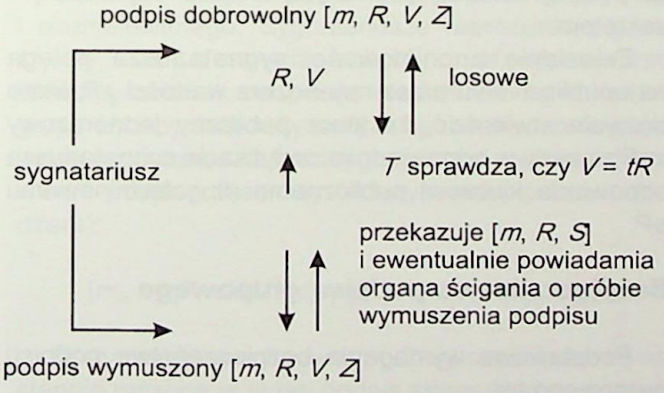
Bezpieczeństwo sygnatariusza wiąże się nie tylko z bezpieczeństwem samego podpisu, lecz także z okolicznościami, w jakich wykonywany jest podpis. Tradycyjny schemat podpisu nie jest zależny od okoliczności, w jakich jest składany: jego weryfikacja przebiega identycznie.

W wyjątkowych okolicznościach (groźba, wymuszenie, szantaż) ideałem jest stworzenie podpisu, który zawiera implicite dowód, że sygnatariusz działał pod presją. Taką informację można by uznać za łagodząca ewentualne skutki prawne takiego podpisu. Naturalny pomysł, który narzuca się w tej sytuacji, polega na zastosowaniu dwóch różnych par kluczy (prywatny,

publiczny) stosowanych zależnie od okoliczności składania podpisu (wymuszenie/dobrowolność).

Aby odpowiednie klucze publiczne nie zdradzały tych okoliczności, sygnatariusz i weryfikujący powinni posiadać im tylko wiadomy sposób ustalenia, który z kluczy jest w danym momencie uznany za dobrowolny, a który za wymuszony. To może powodować jednak pewne komplikacje związane z kosztami obliczeniowymi stosowanego protokołu. Rozwiązanie przedstawione w dalszej części pracy opiera się na zakodowaniu informacji o okolicznościach jego wykonania (wymuszenie/dobrowolność) raczej w „nadmiarowości" składanego podpisu, niż wyborze odpowiedniego klucza prywatnego. Podpis zaszyfrowany nie zdradza jednak charakteru tej „nadmiarowości" dla nikogo poza zaufaną stroną *T*. Gdy weryfikujący podpis zwraca się do *T* z żądaniem odszyfrowania podpisu, informacja o ewentualnym wymuszeniu podpisu zostaje przekazana przez nią odpowiednim organom ścigania. Formalny opis odpowiedniego protokołu wraz z dowodem jego bezpieczeństwa został przedstawiony w pracy J. Pomykały i T. Trabszysa[19]. W dalszej części autorzy przedstawią schemat podpisu odpornego na wymuszenia zrealizowany w systemie opartym na tożsamości (ang. *Identity-based cryptosystems*).

Fazy podpisu odpornego na wymuszenia (ryc. 3)



Ryc. 3. Graficzny schemat omówionych faz podpisu
Fig. 3. *Scheme of discussed signature phases*

W fazie inicjalizacji tworzony jest system (G, A, P). W fazie generacji kluczy zostaje obliczony i opublikowany klucz $sP = P_{pub}$ służący do weryfikacji podpisu, a sygnatariuszowi przydzielane są dwa klucze: publiczny Q_{ID} oraz prywatny sQ_{ID}. Jednocześnie zostaje wygenerowany sekret t współdzielony przez sygnatariusza podpisu i zaufaną stronę *T*.

Faza podpisywania:

W odróżnieniu od algorytmu podpisu szyfrowego sygnatariusz wybiera r, v i oblicza podpis $[m, R, V, Z]$, gdzie $R = rP$, $V = vP$, $Z = rP_{pub} + h(m, R)sQ + vX$

$$S = rP_{pub} + h(m, R, V)D + vX.$$

Faza weryfikacji podpisu zaszyfrowanego Z przebiega podobnie jak w schemacie podstawowym.

Faza deszyfrowania podpisu przez stronę *T* składa się z dwu części: pierwszej, która polega na sprawdzeniu, czy w podpisie:

$$[m, R, V, Z]$$

zachodzi związek $V = tR$ (wtedy następuje powiadomienie o próbie wymuszenia) i drugiej polegającej na obliczaniu właściwego podpisu:

$$S = Z - xV$$

i przekazaniu żądającej stronie weryfikującej podpis.

Faza weryfikacji właściwego podpisu polega na sprawdzeniu, czy:

$$e(P, S) = e(P_{pub}, R + h(m, r)Q).$$

Podpis skumulowany z łańcuchem pełnomocnictw

Ze schematem tego typu mamy do czynienia w przypadku delegowania uprawnień do podpisywania w imieniu właściciela na jego pełnomocnika lub grupę pełnomocników. Bezpieczeństwo pełnomocnika lub właściciela podpisu dotyczy tu z jednej strony szeroko rozumianej niepodrabialności podpisu jakiegokolwiek sygnatariusza przez kogokolwiek innego (w tym właściciela delegowanego pełnomocnictwa lub innych pełnomocników), z drugiej niezaprzeczalności, która oznacza, że jeśli sygnatariusz wykonał podpis, to nie będzie mógł później temu zaprzeczyć. Poniżej autorzy zaprezentują rozszerzenie podpisu pełnomocniczego[20] na iterację (łańcuch) pełnomocnictw z wykorzystaniem podpisu skumulowanego[21]. Idea odpowiedniego podpisu skumulowanych pełnomocnictw jest następująca:

Delegujący pełnomocnictwo U_0 przekazuje pełnomocnictwo sygnatariuszowi U_1, ten z kolei U_2 itd. (w praktyce schemat można zmodyfikować tak, aby U_i, $i = 1, 2,...$ były grupami pełnomocników). Załóżmy teraz, że pełnomocnik U_n zamierza podpisać wiadomość m. Aby podpis był wiarygodny, U_n musi udowodnić, że istnieje łańcuch pełnomocnictw $L(0, 1)$, $L(1, 2)$,...,

$L(n-1, n)$ autoryzowany przez kolejnych pełnomocników aż do $n-1$. Odpowiednie pełnomocnictwa to będą wiadomości $m(i, i+1)$, $i = 0,1,..., n-1$ podpisywane przez odpowiednich sygnatariuszy, a złożenie tych podpisów w jeden (akumulacja) będzie wartością podpisu skumulowanego.

Opis schematu

Dany jest właściciel podpisu U_0 i grupa pełnomocników $\{U_1, U_2,..., U_n\}$. W systemie (G, A, P) każdy sygnatariusz U posiada parę kluczy (certyfikowanych):

$$sk(U) = sk_i \text{ i } pk(U) = pk_i.$$

Podpis skumulowany pod wiadomością m ma postać:

$$[m, pk_0, m(0,1), m(1,2),..., m(n-1, n), sig_0 +...+ sig_n]$$

gdzie:

$sig_i = sk_i H(m(i, i+1))$ $\quad i = 0, 1, ..., n-1$

$sig_n = sk_n H(m)$

$m(i, i+1) = (pk_i, pk_{i+1}, L(i, i+1))$.

Weryfikacja podpisu wykorzystuje własność dwuliniowości iloczynu e i polega na sprawdzeniu, czy:

$$e(P, sig_0 + ... + sig_n) = e(pk_0, H(m(0,1))e(pk_1, H(m(1,2)) ... e(pk_n, H(m)).$$

Podpis anonimowy grupowy

Podpis tego rodzaju dotyczy relacji typu sygnatariusz – grupa. Bezpieczeństwo sygnatariusza zależy od zaufania do menadżera grupy, który ma możliwość zniesienia anonimowości sygnatariusza w szczególnych okolicznościach (ang. *revoking the anonimity*). W dalszej części omówiony zostanie dokładniej taki schemat.

Opis systemu

Dana jest grupa $G = \{U_1, ..., Un\}$ oraz jej menadżer M. Celem jest stworzenie systemu podpisu anonimowego w imieniu grupy G, składanego przez dowolnego członka grupy G (sygnatariusza). Anonimowość sygnatariusza powinna być niepodważalna i jedynie w wyjątkowych okolicznościach (podejrzenie o przestępstwo) może zostać ujawniona w oparciu o informacje posiadane przez menadżera M. W fazie inicjalizacji systemu tworzona jest trójka (G, A, P) (patrz rozdział Podpis szyfrowany weryfikowalny). Faza generacji kluczy jest dwuetapowa. Najpierw zaufana trzecia strona generuje (certyfikowane) pary kluczy długoterminowych dla menadżera $(sk(M), pk(M))$ oraz sygnatariuszy U ze zbioru G postaci:

$$(sk(U), pk(U)) = (s, sP).$$

Następnie sygnatariusz tworzy określoną liczbę par kluczy jednorazowych dla losowych wartości x postaci:

$$(xs, xsP)$$

wykorzystywanych do jednorazowych podpisów. Jako dowód powiązania klucza publicznego jednorazowego xsP z kluczem publicznym długoterminowym sP sygnatariusz przedstawia menadżerowi wartość xP. Menadżer sprawdza, czy pary (sP, xsP) oraz (P, xP) są kodowane tym samym kluczem. Jeśli tak jest, to certyfikuje jednorazowy klucz publiczny xsP swoim podpisem postaci:

$$sk(M)H(xsP),$$

gdzie H jest odpowiednią bezpieczną funkcją haszującą. Podpisany certyfikat jest dalej używany przez sygnatariusza do wygenerowania podpisu grupowego pod wiadomością m postaci:

$$sig(m) = [m, xsH(m), xsP, sk(M)H(xsP)].$$

Weryfikacja podpisu polega na sprawdzeniu, czy pary $[H(m), xsH(m)]$ i $[P, xsP]$ są kodowane tym samym sekretem.

Zniesienie anonimowości sygnatariusza polega na upublicznieniu przez menadżera wartości xP, która pozwala stwierdzić, że klucz publiczny jednorazowy xsP zawarty w odpowiednim certyfikacie sygnatariusza odpowiada kluczowi publicznemu długoterminowemu sP.

Bezpieczeństwo podpisu grupowego

Podstawowe wymagania bezpieczeństwa podpisu grupowego to:

- niepodrabialność,
- anonimowość,
- możliwość zniesienia anonimowości przez menadżera.

Ostatni wymóg jest konsekwencją tego, że klucz długoterminowy sP odpowiada kluczowi jednorazowemu xsP (dzięki kluczowi weryfikacyjnemu xP).

Niepodrabialność podpisu $xsH(m)$ przez nikogo poza posiadaczem klucza s lub klucza x wynika z następującej uwagi. Niech $H(m) = aP$, gdzie losowe a jest symulowane przez wyrocznię obliczającą wartość funkcji haszującej H. Mając dostępne dane publiczne: P, sP,

xP, xsP oraz $H(m) = aP$, próbujemy podrobić podpis $xsaP$. Możliwe drogi są następujące:

$$P, aP, sP, xP \rightarrow xsaP$$

lub

$$P, sP, xsP \rightarrow xsaP.$$

Każda z nich napotyka jednak trudny problem obliczeniowy Diffie-Hellmana, na którym bazuje bezpieczeństwo całego systemu.

Anonimowość z kolei oznacza, że nikt poza menadżerem grupy nie jest w stanie obliczyć xP, mając na wejściu dane: P, aP, sP, xsP, $xsaP$. Warto zauważyć, że nie zmniejszając ogólności mamy dwie drogi obliczenia xP:

$$P, sP, xsP, \rightarrow xP,$$
$$P, aP, sP, xsaP, \rightarrow xP.$$

Każda z nich napotyka problem następujący: mając dane P, aP, bP, gdzie b jest wielokrotnością a, obliczyć wartość b/a P. Podstawiajc $Q = a^{-2}P$, widać, że problem (P, aP, bP, abP) redukuje się do problemu $(Q, a^{\{-1\}}Q, ba^{\{-2\}}Q, ba^{\{-1\}}Q = baP)$. Powyższa redukcja pokazuje, że obliczenie klucza weryfikacyjnego xP w oparciu o dane publiczne jest praktycznie niemożliwe.

Podpis anonimowy skumulowany

Schemat łączy w sobie idee podpisu grupowego i skumulowanego. Sygnatariusze reprezentują grupy $U_1, U_2, ..., U_n$ o menadżerach $M_1, ..., M_n$ odpowiednio. Dowolny członek podpisuje wiadomość m_i w imieniu grupy U_i. Aby całkowity podpis był ważny, co najmniej jeden sygnatariusz z każdej grupy musi wykonać odpowiedni podpis częściowy (autoryzowany przez menadżera):

$$[m_i, s_i x_i P, sk(M_i)H(s_i x_i P)] = [m_i, pk(x_i), \text{sig}_i].$$

Otrzymane (anonimowe) podpisy częściowe są następnie łączone w jeden podpis skumulowany postaci:

$$[m, pk(x), \text{sig}],$$

gdzie:

$$pk(x) = (pk(x_1), ..., pk(x_n))$$
$$m = (m_1, ..., m_n)$$
$$\text{sig} = \text{sig}_1 + \text{sig}_2 + ... + \text{sig}_n.$$

Weryfikacja podpisu polega na sprawdzeniu równości:

$$e(P, \text{sig}) = e(pk(x_1), H(m_1))e(pk(M_1), H(x_1 s_1 P))$$
$$e(pk(x_2), H(m_2))e(pk(M_2), H(x_2 s_2 P))...$$
$$...e(pk(x_n), H(m_n))e(pk(M_n), H(x_n s_n P)).$$

Bezpieczeństwo schematu wynika z bezpieczeństwa podpisu grupowego oraz bezpieczeństwa podpisu skumulowanego. Przykładem demonstrującym ważne zastosowanie anonimowego podpisu skumulowanego jest system kontroli uprawnień do korzystania z zasobów baz danych. Użytkownicy mają prawo do zachowania swojej anonimowości wobec zarządcy grupy, zachowując swoje przywileje wydane przez niego dla wybranych grup (roli). Dokładniej administrator udziela pozwolenia na dostęp do danych (w ramach określonego profilu), jeśli na losowej wiadomości (wezwaniu generowanym przez system) sygnatariusz wykona poprawny podpis częściowy oraz dodatkowo upoważnienie m_i członka innej grupy U_i do podpisu w imieniu grupy U_i. Upoważniający załącza z kolei podpis pod wiadomością m_j potwierdzającą jego uprawnienie od członka innej grupy U_j itd. Jeśli tak skonstruowany łańcuch upoważnień prowadzi do upoważnienia wydanego przez zarządcę systemu, to podpis skumulowany zostaje zweryfikowany pozytywnie.

Szyfrowany weryfikowalny podpis anonimowy

Zaprezentowane zostaną teraz nowe propozycje podpisu cyfrowego, który może znaleźć zastosowanie w kontraktach elektronicznych zawieranych przez anonimowe podmioty z grup zarządzanych przez odpowiednich menadżerów. Menadżer odgrywa w tym wypadku przede wszystkim rolę zaufanej trzeciej strony, a dodatkowo może znieść anonimowość odpowiedniego podmiotu podpisującego kontrakt. Zatem sygnatariusze pozostający między sobą w relacji anonimowości składają podpisy zaszyfrowane (kluczami publicznymi swoich menadżerów), których poprawność każdy z nich może zweryfikować. Ostatecznie kontrakt zostaje (opcjonalnie) uprawomocniony przez menadżerów odpowiednich grup, którzy, jeśli zachodzi taka potrzeba, odkrywają również tożsamość odpowiednich podmiotów.

Opis schematu

- Inicjalizacja systemu

Tworzona jest trójka: (G, A, P) (patrz rozdział Podpis szyfrowany weryfikowalny).

● Generowanie kluczy

Generowane są pary kluczy (certyfikowanych) dla menadżerów ($pk(M_i)$, $sk(M_i)$), $i = 1, 2$ oraz dla sygnatariuszy (s_i, s_iP). Następnie każdy członek grupy generuje parę jednorazowych kluczy:

$$(x_is_i, x_is_iP)$$

a odpowiedni klucz weryfikacyjny x_iP przekazuje swojemu menadżerowi.

● Faza podpisywania

Dla danej wiadomości sygnatariusz każdej grupy oblicza podpis zaszyfrowany:

$$[m, pk(x_i), v_iP, \text{sig}_i],$$

gdzie:

$$pk(x_i) = x_is_iP, \text{sig}z_i = x_is_iH(m) + sk(M)H(pk(x_i)) + v_isk(M_i)P, \quad i = 1,2$$

który przekazuje swojemu kontrahentowi.

● Faza weryfikacji

Kontrahenci weryfikują podpisy zaszyfrowane sprawdzające, czy:

$$e(P, \text{sig}z_i) = e(pk(x_i), H(m))e(pk(x_i), pk(M_i))e(v_iP, pk(M_i)), \quad i = 1, 2$$

Jeśli weryfikacja zaakceptuje podpisy, są one przekazywane menadżerom odpowiednich grup, którzy obliczają właściwe podpisy postaci:

$$\text{sig}_i = \text{sig}z_isk(M_i)v_iP, \quad i = 1, 2$$

przekazując je odpowiednim kontrahentom, którzy przeprowadzają ich ostateczną weryfikację sprawdzając, czy:

$$e(P, \text{sig}_i) = e(pk(x_i), H(m))e(pk(x_i), pk(M_i)), \quad i = 1, 2.$$

Odpowiedni protokół kryptograficzny z formalną analizą bezpieczeństwa jest opisany w pracy J. Pomykały i T. Trabszysa[22].

Podpis pierścieniowy

Podpis pierścieniowy jest rodzajem podpisu gwarantującego pełną anonimowość sygnatariuszy[23]. W tym przypadku dowolny członek grupy (np. stworzonej ad hoc) może, bez ryzyka identyfikacji, podpisać dowolną wiadomość w imieniu grupy. Co więcej pozostali członkowie grupy mogą nawet nie wiedzieć o samym fakcie wygenerowania podpisu przez jednego z jej członków. W takiej sytuacji mają jedynie takie samo prawo do dementowania wiadomości fałszywych (pomówień etc.) w imieniu całej grupy.

Poniżej przedstawiony zostanie opis takiego schematu wykorzystującego strukturę (G, A, P). Dla uproszczenia autorzy przedstawią schemat dla grupy złożonej z trzech podmiotów.

● Faza inicjalizacji systemu

W tej fazie tworzona jest trójka (G, A, P) (patrz rozdział Podpis szyfrowany weryfikowalny).

● Faza generowania kluczy

W tej fazie każdemu sygnatariuszowi przydzielana jest para kluczy – prywatny/publiczny:

$$(sk_i, pk_i) = (s_i, s_iP) \quad i = 1, 2, 3.$$

● Faza podpisywania i weryfikacji

Przypuśćmy, że sygnatariusz, posiadający parę kluczy (s_1, s_1P), zamierza wygenerować podpis pod wiadomością m. Wtedy wybiera losowe wartości r_2 i r_3, publikuje wskazówki r_2P, r_3P oraz oblicza podpis:

$$\text{sig}_1 = \text{sig}_1(s_1, r_2, r_3, m)$$

w ten sposób, by spełniony był następujący warunek (weryfikujący podpis):

$$e[P, H(m)] = e(s_1P, \text{sig}_1)e(s_2P, r_2P)e(s_3P, r_3P).$$

Powyższa równość jest podstawą do właściwego określenia podpisu sig_1. Mianowicie korzystając z dwuliniowości e wiemy, że jej prawa strona wynosi:

$$e(P, s_1 \text{sig}_1 + r_2s_2P + r_3s_3P)$$

skąd (wykorzystując fakt, że P jest generatorem grupy G) otrzymujemy, że wartość podpisu, który przejdzie weryfikację, wynosi:

$$\text{sig}_1 = 1/s_1[H(m) - r_2(s_2P) - r_3(s_3P)].$$

Tym samym poprawność podpisu została wykazana.

PRZYPISY

[1] **B. Hołyst:** Zwalczanie cyberterroryzmu, [w:] Terroryzm, rozdział II, w druku;

[2] **B. Hołyst, J. Pomykała:** Wykorzystywanie kryptografii przez środowiska terrorystyczne, „Prokuratura i Prawo" 2008, nr 2;

[3] **B. Hołyst, J. Pomykała:** Podpis elektroniczny – aspekty kryminalistyczne, prawne i informatyczne, „Problemy Kryminalistyki" 2008, nr 256, s. 5–21;
[4] **R. Anderson:** Inżynieria zabezpieczeń, WNT 2005, s. 491–539;
[5] **B. Hołyst, J. Pomykała:** Bezpieczeństwo podpisu progowego w grupach dynamicznych, „Problemy Kryminalistyki" 2008, nr 259, s. 16–21;
[6] **M. Musiał:** Rząd nadal szuka pomysłu na łagodne wdrożenie e-podpisu, „The Wall Street Journal. Polska", 25.04.2008, s. 12;
[7] **G.J. Simmons:** Subliminal channels: past and present, IEEE European Trans. on Telecom. vol. 5, no 4, 1994, s. 459–473;
[8] **M. Kutyłowski, W.B. Strothmann:** Kryptografia, teoria i praktyka zabezpieczania systemów komputerowych, Wyd. RM, Warszawa 1999, s. 121–123;
[9] **Y. Desmedt, Y. Frankel:** Threshold cryptosystems, Adv. in Cryptology, – Crypto, '89, LNCS 435;
[10] **B. Nakielski, J. Pomykała, J.A. Pomykała:** A multi-threshold signature scheme, „Journal of Telecomunications and Information Technology" (1) 2008, s. 51–55; **R. Anderson:** Two remarks on public-key cryptography, Manuscript, Sep. 2008;
[11] **J. Pomykała, T. Warchoł:** Threshold signatures in dynamic groups, Proceedings of Future Communication and Networking 2007, IEEE Computer Science, s. 32–37;
[12] **N. Asokan, V. Shoup, M. Vaidner:** Optimistic fair exchange of digital signatures, NCS 1403 (1998), s. 591–606;
[13] **D. Chaum:** Blind signatures for untraceable payments, Adv. In Cryptology, Proceed. Od Crypto'82 (1983), s. 199–203;
[14] **A. Lysyanskaya, Z. Ramzan:** Group blind digital signatures: a scalable solution to electronic cash, LNCS 1465 (1998), s. 197–238;
[15] **M. Bellare, S. Miner:** A forward – secure digital signature scheme, LNCS 1666, s. 431–448;
[16] **A. Shamir:** How to share a secret, Communications of the ACM, 22: s. 612–613, 1979;
[17] **M. Abdalla, S. Miner, Ch. Namprempre:** Forward secure threshold signature schemes, LNCS 2020, (2001) p. 441–456;
[18] **D. Boneh, M. Franklin:** Identity-based encryption from the Weil pairing, SIAM Journal on Computing, vol. 32, No. 3, 2003, s. 586–615;
[19] **J. Pomykała, T. Trabszys:** Blackmail protected verifiably encrypted signature form bilinear pairing (praca wysłana do publikacji);
[20] **J. Pomykała, B. Źrałek:** A model of Id-based proxy signature scheme, Procedings of 6th Collecter Iberoamerica: Collaborative Electronic Communications & eCommerce Tech and Research Conference, Madrid 2008;
[21] **D. Boneh, C. Gentry, B. Lynn, H. Shacham:** Short signatures from the Weil pairing, Journal of Cryptology, vol. 17, No. 4, 2004, s. 297–319;
[22] **J. Pomykała, T. Trabszys:** Anonymous signer verifiable encrypted signature from bilinear paring, praca wysłana do publikacji;
[23] **R.L. Rivest, A. Shamir, Y. Tauman:** How to leak a secret, LNCS 2248, 2001, s. 552–565.