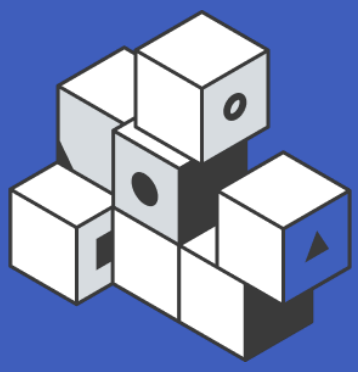
MC

Tłumaczenie standardów i rekomendacji
w zakresie cyberbezpieczeństwa

Specyfikacja kryptograficzna projektów realizowanych przez organizacje:

Metody komunikacji w aplikacjach

BSI TR-03116-4_wer. 1.0_PL



BSI TR-03116-4_wer. 1.0_PL

11 czerwca 2024

Specyfikacja kryptograficzna projektów realizowanych przez organizacje:

Metody komunikacji w aplikacjach

Publikacja dostępna pod adresem:

Rekomendacje cyberbezpieczeństwa

Bundesamt
für Sicherheit in der
Informationstechnik

Deutschland
Digital•Sicher•BSI•

Technische Richtlinie BSI TR-03116 Kryptographische Vorgaben für Projekte der Bundesregierung

Teil 4: Kommunikationsverfahren in Anwendungen

Stand 2022
Datum: 24. Januar 2022

O PUBLIKACJI

Niniejsze opracowanie **BSI TR-03116-4_wer. 1.0_PL, *Specyfikacja kryptograficzna projektów realizowanych przez organizacje: Metody komunikacji w aplikacjach***, stanowi tłumaczenie publikacji Technische Richtlinie BSI TR-03116-4, *Kryptographische Vorgaben für Projekte der Bundesregierung*, i zostało opracowane za zgodą Bundesamtes für Sicherheit in der Informationstechnik (BSI). Wykorzystane może być do celów zgodnych z prawem i niekomercyjnych, które służą interesom bezpieczeństwa technologii informacyjnych (Terms and conditions of use).

Przytaczane i cytowane w publikacji przepisy, okólniki, rozporządzenia wykonawcze, dyrektywy, normy, standardy, polityki, memoranda itp. odnoszą się, o ile nie zaznaczono inaczej, do prawodawstwa i rynku niemieckiego. Jeżeli cytowany fragment ma przełożenie lub odpowiednik w polskim porządku prawnym lub normalizacyjnym, wówczas informacje te wskazane są bezpośrednio w tekście lub w przypisach.

W publikacji posłużono się pojęciami zdefiniowanymi w oryginalnej (niemieckiej) wersji dokumentu, na podstawie którego powstały niniejsze zalecenia.

Tam, gdzie to było możliwe i nie budziło kontrowersji, nazwy ról i kluczowych uczestników procesu zarządzania ryzykiem zostały podane w języku polskim. Pozostałe role i funkcje zostały przedstawione w języku angielskim[1]. Do wszystkich tych ról / funkcji zastosowano akronimy terminologii angielskiej.

Terminologia angielska i akronimy występujące w publikacji zdefiniowane są w dokumencie ***Słownik kluczowych pojęć z zakresu cyberbezpieczeństwa*.**

Podmioty, urządzenia lub materiały o charakterze komercyjnym prezentowane są w niniejszym dokumencie w celu odpowiedniego opisania procedury lub koncepcji eksperymentalnej. Celem ich wskazania nie jest nakłanianie do korzystania z ww. podmiotów, urządzeń lub materiałów lub ich poparcie. Wskazanie ich nie ma również na celu sugerowania, że te podmioty, materiały lub sprzęt są najlepsze z dostępnych w danej dziedzinie. Taka identyfikacja nie stanowi rekomendacji, poparcia ani nie ma na celu sugerowania, że dane podmioty, materiały lub urządzenia są bezwzględnie najlepsze z dostępnych dla osiągnięcia danego celu.

[1] Kluczowi uczestnicy zarządzania ryzykiem – patrz: Narodowe Standardy Cyberbezpieczeństwa

Spis treści

Spis tabel

1. WPROWADZENIE

Wytyczna techniczna BSI TR-03116-4 opisuje specyfikację projektów realizowanych przez rząd federalny Republiki Federalnej Niemiec. Wytyczna techniczna dzieli się na sześć części[2]:

- Część 1 wytycznej technicznej opisuje wymogi bezpieczeństwa obowiązujące w zakresie stosowania metod kryptograficznych w sektorze opieki zdrowotnej w odniesieniu do użycia elektronicznej karty zdrowia (eGK), legitymacji pracownika służby zdrowia (HBA) oraz elementów technicznych infrastruktury telematycznej.
- Część 2 opisuje wymogi bezpieczeństwa dotyczące sposobu stosowania metod kryptograficznych w dokumentach nadrzędnych i dokumentach eID opartych na zasadzie rozszerzonej kontroli dostępu (*ang. extended access control*), czyli na chwilę obecną paszportu elektronicznego, elektronicznego dowodu osobistego, elektronicznego potwierdzenia prawa do pobytu, karty eID dla obywateli unijnych, smart-eID, naklejek kodyfikujących zmiany dokumentów nadrzędnych, naklejek wizowych i potwierdzenia przybycia.
- Część 3 niniejszej wytycznej opisuje wymagania bezpieczeństwa obowiązujące w zakresie stosowania metod kryptograficznych w infrastrukturze inteligentnych systemów pomiarowych w sektorze energetycznym.
- Poniższa część 4 wytycznej opisuje wymagania bezpieczeństwa w zakresie stosowania standardów komunikacyjnych SSL/TLS, S/MIME, SAML/XML Security i OpenPGP w aplikacjach organizacji sektora publicznego.
- Część 5 wytycznej opisuje wymagania bezpieczeństwa dotyczące sposobu stosowania metod kryptograficznych w aplikacjach Secure Element API (jak np. technicznej infrastrukturze bezpieczeństwa inteligentnych systemów rejestracyjnych).

[2] Części 1-3 oraz 5-6 nie mają zastosowania w polskich realiach i nie zostały opracowane w języku polskim.

- Część 6 wytycznej opisuje wymagania bezpieczeństwa dotyczące sposobu stosowania metod kryptograficznych w infrastrukturze współpracujących inteligentnych systemów zarządzania ruchem drogowym (*ang. cooperative intelligent transport systems – C-ITS*).

Specyfikacje określone w niniejszej wytycznej technicznej opierają się na prognozach (por.[3] [1]) odnośnie bezpieczeństwa stosowanych metod kryptograficznych i długości kluczy na okres 7 lat, do roku 2028 włącznie. Oznaczenie 2028+ oznacza, że zastosowanie określonej metody będzie najprawdopodobniej dopuszczalne również po upływie tego czasu.

Poniższa wytyczna określa wymagania wstępne w zakresie stosowania metod komunikacji. W tym kontekście rozróżnić należy proces właściwej komunikacji oraz proces przypisywania tożsamości uczestnikom komunikacji:

- Rozdział 2 zawiera specyfikacje dotyczące sposobów zabezpieczania komunikacji przy użyciu protokołu TLS.
- Rozdział 3 zawiera specyfikacje dotyczące sposobów zabezpieczania komunikacji przy użyciu protokołu SAML/ XML Security.
- Rozdział 4 zawiera specyfikacje dotyczące sposobów zabezpieczania komunikacji pocztą elektroniczną przy użyciu protokołu S/MIME.
- Rozdział 5 zawiera specyfikacje dotyczące sposobów zabezpieczania komunikacji pocztą elektroniczną przy użyciu protokołu OpenPGP.
- Rozdział 6 zawiera specyfikacje w zakresie identyfikacji partnerów komunikacyjnych. To kontekst aplikacji decyduje o konieczności identyfikacji/ uwierzytelnienia jednego lub obydwu partnerów w konkretnym przypadku aplikacji.
- Rozdział 7 opisuje wymagania obowiązujące w zakresie generowania, zapisywania, przetwarzania i usuwania kluczy kryptograficznych. Ponadto zawiera on zalecenia dotyczące generatorów liczb losowych.

[3] Por. – porównaj.

2. SPECYFIKACJA SSL/TLS

Protokół *transport layer security* (*ang transport layer security – TLS*), wcześniej znany jako *secure socket layer* (*ang. secure socket layer – SSL*), służy do zapewniania bezpieczeństwa komunikacji w Internecie, np. w połączeniu z HTTP (HTTPS) lub FTP (FTPS). Protokół ten umożliwia negocjowanie i nawiązywanie bezpiecznego połączenia pomiędzy dwoma komputerami, czyli pomiędzy *klientem* a *serwerem*.

Podczas nawiązywania połączenia *(tzw. ang. handshake*) obie strony samodzielnie negocjują pomiędzy sobą algorytmy szyfrowania i uwierzytelniania (czyli *pakiety szyfrów cipher suite*) wykorzystywane w inicjalizowanej sesji, a także wykorzystywane w tym celu klucze.

Kolejną częścią składową procesu nawiązywania połączenia (*ang. handshake*) jest oparta na certyfikacie procedura wzajemnego uwierzytelniania jednego lub obydwu partnerów.

2.1. Specyfikacje ogólne

Podczas korzystania z protokołu TLS należy zawsze przestrzegać wymagań i zaleceń opisanych w rozdziale 3 wytycznych technicznych BSI TR-02102-2 [2]. Jakiekolwiek dopuszczalne różnice lub odstępstwa od zaleceń TR-02102-2 są wyraźnie opisane w niniejszym dokumencie.

W zakresie stosowanych kluczy kryptograficznych i liczb losowych zastosowanie mają zalecenia opisane w rozdziale 7 niniejszej wytycznej technicznej.

W uzasadnionych i wyjątkowych przypadkach oraz w odniesieniu do specjalnych scenariuszy aplikacji można odstąpić od niektórych wymagań opisanych w niniejszym rozdziale, pod warunkiem jednak, że jest to konieczne w celu zapewnienia interoperacyjności i nie dojdzie do ograniczenia docelowego poziomu bezpieczeństwa.

2.1.1 Wersje TLS i sesji

W celu zapewnienia zgodności z niniejszą wytyczną techniczną wymagana jest obsługa przynajmniej wersji 1.2 [3] protokołu TLS. Zaleca się obsługę wersji 1.3 [4] TLS. W momencie nawiązania połączenia (tzw. *handshake*) pomiędzy klientami a serwerami, spełniającymi wymagania niniejszej wytycznej technicznej, ustala się

zawsze jedną z wymienionych wersji protokołu TLS. Nie wolno używać starszych wersji SSL/TLS [5].

Jedna sesja TLS nie powinna trwać dłużej niż 48 godzin. Dotyczy to również przypadków korzystania z funkcji wznowienia sesji (*ang. session resumption).*

2.2. Specyfikacja TLS 1.2

2.2.1 Pakiety szyfrów

Pakiet szyfrów w przypadku TLS 1.2 [3] określa algorytmy stosowane w celu:

- uzgodnienia szyfru,
- szyfrowania pakietów danych (szyfrowanie strumieniowe/ blokowe, wraz z trybem operacyjnym) oraz
- funkcji haszującej stosowanej w algorytmie HMAC w celu zapewnienia integralności pakietów danych i w celu wykorzystania w roli generatora liczb pseudolosowych (powyżej wersji TLS 1.2).

Kompletna lista wszystkich określonych pakietów szyfrów z odnośnikiem do wybranych specyfikacji jest dostępna w punkcie [6].

2.2.1.1. Klienci TLS

Tabela 1 wymienia wiążące pakiety szyfrów, które obsługiwać powinny programy klientów. Ponadto klienci TLS obsługiwać powinni pozostałe pakiety szyfrów zalecane w punkcie [2].

Pakiety szyfrów	*Obsługiwane od*	*Obsługiwane do*
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	2013	2026
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	2013	2026
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	2015	2028+
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	2015	2028+

Tabela 1 Minimalny zakres pakietów szyfrów obsługiwany przez klientów TLS

Jeśli obsługiwane będą pakiety szyfrów, których obsługa wynika z regulacji przejściowych zgodnie z [2], rozdział 3.3.1.4, to pakiety takie oferowane muszą być przez klientów TLS w ramach powitania z klientem z niższym priorytetem niż pakiety szyfrów zalecane standardowo w [2].

2.2.1.2. Serwer TLS

Serwery TLS muszą posiadać przynajmniej jeden certyfikat zawierający klucz publiczny szyfrowania ECDSA lub RSA[4]. Jeśli serwer TLS nie posiada dwóch certyfikatów, tzn. posiada po jednym certyfikacie dla każdego typu klucza, to zaleca się stosowanie kluczy ECDSA.

Serwery TLS muszą wiążąco obsługiwać co najmniej jeden z pakietów szyfrów wymienionych w tabeli 2. Ponadto serwer obsługiwać powinien dodatkowe pakiety szyfrów zalecane w [2].

Pakiety szyfrów	*Obsługiwane od*	*Obsługiwane do*
Serwer TLS z certyfikatem ECDSA		
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	2013	2026
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	2015	2028+
Serwer TLS z certyfikatem RSA		
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	2013	2026
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	2015	2028+

Tabela 2 Minimalny zakres pakietów szyfrów obsługiwany przez serwery TLS

Jeśli obsługiwane są pakiety szyfrów, których obsługa wynika z regulacji przejściowych zgodnie z [2], rozdział 3.3.1.4, to pakiety takie oferowane powinny być przez serwery TLS z niższym priorytetem niż pakiety szyfrów zalecane standardowo w [2].

2.2.1.3. Przypadki szczególne

Jeśli używane są zestawy szyfrów związane z aplikacją, w których oprócz uwierzytelniania serwera TLS za pomocą certyfikatów, wstępnie wymieniane dane

[4] Patrz również rozdział 6.1.

(klucz wstępny; *ang. pre-shared-key – PSK*) są zawarte w uwierzytelnianiu i uzgadnianiu klucza, musi być obsługiwany co najmniej zestaw szyfrów zgodnie z tabelą 3.

Pakiety szyfrów	*Użycie do*
TLS_RSA_PSK_WITH_AES_128_CBC_SHA256	2028+

Tabela 3 Pakiety szyfrów z kluczem współdzielonym pre-shared-key

Jeśli w odniesieniu do konkretnej aplikacji wykorzystywane są pakiety szyfrów oparte wyłącznie na PSK, to zapewnić należy obsługę pakietów szyfrów z tabeli 4.

Pakiety szyfrów	*Użycie do*
TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256	2028+

Tabela 4 Pakiety szyfrów oparte na certyfikatach i kluczu współdzielonym pre-shared-key

Zaleca się zapewnienie obsługi pozostałych pakietów szyfrów, zalecanych dla danego przypadku aplikacji zgodnie z [2] przynajmniej po stronie serwera.

Jeśli obsługiwane są pakiety szyfrów, których obsługa wynika z regulacji przejściowych zgodnie z [2], rozdział 3.3.1.4, to pakiety takie muszą być wykorzystywane przez klientów i serwery z niższym priorytetem niż pakiety szyfrów zalecane standardowo w [2].

2.2.2 Parametry domeny

W przypadku krzywych eliptycznych można używać tylko krzywych nazwanych (*ang. named curve*) - (patrz [6]), aby zapobiec atakom za pomocą niezweryfikowanych wartości parametrów domeny.

Krzywe eliptyczne *brainpool* należy wykorzystywać z najwyższym priorytetem.

Tabela 5 wymienia minimalny zakres obsługiwanych krzywych eliptycznych w sposób wiążący.

Krzywe eliptyczne	*Obsługiwane od*	*Obsługiwane do*
secp256r1 (nr IANA 23)	2015	2028+
brainpoolP256r1 (patrz [7], nr IANA 26)	2016	2028+

Tabela 5 Minimalny zakres obsługiwanych krzywych eliptycznych

Ponadto zaleca się zapewnienie obsługi pozostałych krzywych eliptycznych zalecanych w [2] przynajmniej po stronie serwera. W odróżnieniu do TR-02102-2 do roku 2021 włącznie wykorzystywać można również krzywą eliptyczną secp224r1 (nr IANA 22) pod warunkiem spełnienia pozostałych zaleceń odnośnie TLS zawartych w TR-02102-2.

Klienci TLS muszą używać rozszerzenia Supported-Groups lub Supported-Elliptic-Curves2[5], aby przekazać serwerowi wspierane krzywe eliptyczne. W przypadku obsługi pakietów szyfrów opartych na DHE zaleca się również stosowanie rozszerzenia grup obsługiwanych zgodnie z [8].

Zarówno klienci, jak i serwery TLS muszą odmówić akceptacji parametrów domeny, jeśli nie spełniają one wymagań niniejszej wytycznej technicznej.

2.2.3 Pozostałe specyfikacje

2.2.3.1. Algorytmy podpisów

Klienci TLS muszą stosować rozszerzenia algorytmów podpisu, aby wyświetlać pary algorytmów podpisu i algorytmów haszujących obsługiwanych w celu weryfikacji podpisu. Jeśli w konkretnej aplikacji dochodzi również do uwierzytelniania klienta TLS, to serwer TLS podaje obsługiwane algorytmy w wiadomości z żądaniem certyfikatu.

Tabela 6 wymienia minimalny zakres obsługiwanych algorytmów w sposób wiążący.

Algorytm podpisu	*Funkcje haszujące*	*Obsługiwane od*	*Obsługiwane do*
ECDSA[6]	SHA-256	2015	2028+
RSA	SHA-256	2015	2028+

Tabela 6 Minimalny zakres obsługiwanych algorytmów podpisu

[5] Rozszerzenie obsługiwanych krzywych eliptycznych nazywane jest w [8] rozszerzeniem grup obsługiwanych.

[6] Nie dotyczy stosowania pakietów szyfrów RSA_PSK_*

W odróżnieniu do TR-02102-2 w odniesieniu do algorytmów podpisu stosować można funkcję haszującą SHA-224 do roku 2021 włącznie.

2.2.3.2. Schemat szyfrowania encrypt-then-MAC

Zgodnie z [3] w TLS dane tekstowe są najpierw zabezpieczane pod względem integralności (MAC), a następnie szyfrowane w formie tekstowej i MAC (*ang. MAC-then-Encrypt*). W połączeniu z niezabezpieczonym paddingiem prowadzi to do ataków Oracle [9].

Zasadniczo preferuje się stosowanie schematu encrypt-then-MAC lub szyfrowania uwierzytelnionego [10]. W przypadku schematu encrypt-then-MAC przesyłane dane są najpierw szyfrowane, a następnie zabezpieczane MAC. Dlatego zaleca się korzystanie z rozszerzenia encrypt-then-MAC zgodnie z opisem w [11], co znaczy, że klienci powinni zapewniać rozszerzenie encrypt-then-MAC w powitaniu klienta (*ang. client-hello*), a serwery powinny dobierać albo pakiet szyfrów GCM albo CCM, albo skorzystać z rozszerzenia encrypt-then-MAC w powitaniu serwera (*ang. server-hello*). Jeśli stosowana w danym przypadku implementacja TLS obsługuje rozszerzenie encrypt-then-MAC, to należy je zapewnić i wykorzystać zgodnie z [11].

2.2.3.3. Schemat OCSP-stapling

W przypadku TLS wymagana jest w szczególności weryfikacja certyfikatu serwera (por. rozdz. 6.1.3). Zasadniczo istnieją różne sposoby weryfikowania informacji blokujących z certyfikatów. Weryfikacja informacji zwrotnych za pośrednictwem OCSP może spowodować zwiększenie ilości połączeń na przynależnym CA oraz problemy z ochroną danych klienta.

Schemat OCSP-stapling to metoda, w której serwer udostępnia bezpośrednio klientowi informacje zwrotne w postaci podpisanych odpowiedzi OCSP ze znacznikiem czasu podczas nawiązania połączenia (*ang. handshake*).

Zaleca się stosowanie schematu OCSP-stapling zgodnie z [12] (lub [13]).

2.2.3.4. Haszowanie sesji i rozszerzenie extended master secret

Zasadniczo podczas nawiązania połączenia TLS zgodnie z [3] obliczenia *master secrets* wykonywane są w sposób, który nie wymusza uwzględnienia wszystkich parametrów

kryptograficznych TLS w obliczeniach. W zależności od stosowanych parametrów kryptograficznych, brak uwzględnienia danych może spowodować atak na sesję TLS (por. np. atak *triple handshake* [14]).

Zasadniczo zaleca się również uwzględnienie specyficznych danych kontekstowych podczas obliczania kluczy sesji. Dlatego zaleca się stosowanie rozszerzenia extended master secret zgodnie z [15]. W tym przypadku parametry kryptograficzne w postaci *haszowania sesji* (wartość haszowania dla wszystkich wiadomości TLS-handshake) należy uwzględnić podczas obliczania rozszerzenia master secrets.

2.3. Specyfikacja TLS 1.3

2.3.1 Tryby nawiązywania połączenia (*handshake*)

W przypadku korzystania z TLS 1.3 [4] zapewnić należy obsługę następującego trybu nawiązywania połączenia (*ang. handshake*):

- (EC)DHE

W celu obsługi funkcji wznowienia sesji, obsługiwany może być również następujący tryb nawiązywania połączenia (*ang. handshake*):

- PSK z (EC)DHE

Nie wolno ani wysyłać, ani odbierać danych 0-RTT.

2.3.2 Pakiety szyfrów

Pakiet szyfrów w przypadku TLS 1.3 określają algorytmy stosowane w celu

- uwierzytelnionego szyfrowania pakietów danych (szyfrowanie strumieniowe/blokowe, wraz z trybem operacyjnym) oraz
- funkcji haszowania wyprowadzania klucza.

Kompletna lista wszystkich określonych pakietów szyfrów z odnośnikiem do wybranych specyfikacji jest dostępna w punkcie [6].

W przypadku korzystania z TLS 1.3, klienci i serwery TLS powinny obsługiwać co najmniej pakiety szyfrów wymienione w tabeli 7.

Pakiety szyfrów	*Użycie do*
TLS_AES_128_GCM_SHA256	2028+

Tabela 7 Minimalny zakres obsługiwanych pakietów szyfrów

Zaleca się zapewnienie obsługi pozostałych pakietów szyfrów, zalecanych dla danego przypadku aplikacji zgodnie z [2] przynajmniej po stronie serwera.

2.3.3 Parametry domeny

W przypadku TLS 1.3 i stosowania krzywych eliptycznych użyć należy krzywych nazwanych (patrz [6]), aby zapobiec atakom przez niezweryfikowane słabe parametry domeny. Klienci i serwery TLS powinno obsługiwać co najmniej krzywe eliptyczne wymienione w tabeli 8.

Krzywe eliptyczne	*Użycie do*
secp256r1 (nr IANA 23)	2028+
brainpoolP256r1tls13 (patrz [16], nr IANA 31)	2028+

Tabela 8 Minimalny zakres obsługiwanych krzywych eliptycznych

Ponadto zaleca się zapewnienie obsługi pozostałych krzywych eliptycznych zalecanych w [2] przynajmniej po stronie serwera.

2.3.4 Algorytmy podpisów

2.3.4.1. Algorytmy podpisów na potrzeby handshake

W przypadku korzystania z protokołu TLS 1.3, klienci TLS muszą korzystać z rozszerzenia algorytmów podpisu w celu wyświetlenia obsługiwanych algorytmów podpisów do weryfikacji podpisów serwera w ramach procesu nawiązywania połączenia handshake.

W tym celu klienci i serwery TLS powinni obsługiwać co najmniej algorytmy podpisu wymienione w tabeli 9.

Algorytm	*Użycie do*
ecdsa secp256r1 sha256	2028+
ecdsa brainpoolP256r1tls13 sha256	2028+
rsa_pss_rsae_sha256	2028+
rsa_pss_pss_sha256	2028+

Tabela 9 Minimalny zakres obsługiwanych algorytmów podpisu

2.3.4.2. Algorytmy podpisu używane do weryfikacji certyfikatów

W przypadku korzystania z TLS 1.3, klienci TLS muszą korzystać z rozszerzenia certyfikatów algorytmów podpisu w celu wyświetlenia obsługiwanych algorytmów podpisów służących do weryfikacji podpisów.

W celu weryfikacji certyfikatu klienci i serwery TLS powinny obsługiwać co najmniej algorytmy podpisu wymienione w tabeli 10.

Algorytm	*Użycie do*
rsa pkcsl sha256	2025
rsa pkcsl sha384	2025
rsa pss rsae sha256	2028+
rsa pss rase sha256	2028+
rsa pss rsae sha384	2028+
ecdsa secp256r1 sha256	2028+
ecdsa brainpoolP256r1tls13 sha256	2028+
ecdsa secp384r1 sha384	2028+
ecdsa brainpoolP384r1tls13 sha384	2028+

Tabela 10 Minimalny zakres obsługiwanych algorytmów podpisu do weryfikacji certyfikatu

3. SPECYFIKACJA SAML/XML SECURITY

Extensible markup language (XML) to standard W3C służący do przedstawiania danych ustrukturyzowanych hierarchicznie, który jest używany w aplikacjach takich jak SOAP (*ang. simple object access protocol*) lub SAML (*ang. security assertion markup language*).

Zabezpieczenie kryptograficzne wiadomości XML lub ich części opiera się na standardzie XML Security (podpisie XML i szyfrowaniu XML). Protokół XML Security umożliwia bezpieczną identyfikację partnerów komunikacyjnych i pozwala na:

- zapewnienie poufności oraz
- autentyczności i integralności wiadomości.

XML Security umożliwia zabezpieczanie zarówno wybranych treści wiadomości, jak i całych wiadomości. To aplikacja decyduje o tym, które treści będą szyfrowane lub podlegać będą uwierzytelnieniu.

Identyfikacja partnerów komunikacyjnych odbywa się zwykle za pomocą certyfikatów X.509. Zaufana wymiana może opierać się na PKI i być realizowana przez łańcuch certyfikatów lub podpisane metadane albo przez dwustronną wymianę kluczy. Należy przestrzegać specyfikacji z rozdziału 6.

3.1. Wersje

Do podpisywania stosować należy podpis XML zgodnie z [17], a do szyfrowania szyfr XML zgodnie z [18]. Metody obsługiwane lub stosowane w poszczególnych przypadkach są opisane poniżej.

3.2. Funkcje haszujące

W schemacie XML security stosuje się funkcje haszujące do różnych celów, takich jak podpisy cyfrowe lub wyprowadzanie kluczy. W tym celu należy zastosować funkcję haszowania z tabeli 11.

Metoda	*Minimalna długość wynikowa*	*Użycie do*
SHA-2 [19]	224	2022
	256	2028+

Tabela 11 Funkcje haszujące dla XML security

3.3. Podpis XML

3.3.1 Podpisy

Do podpisywania danych w ramach XML security należy użyć jednej z metod podpisywania wymienionej w tabeli 12.

Metoda	*Minimalna długość klucza*	*Użycie do*
RSASSA-PSS [20]	2048	2022
	3072	2028+
ECDSA [21], [19]	224	2022
	256	2028+

Tabela 12 Metody podpisywania w XML security

3.4. Szyfrowanie XML

Jeśli dane podczas transmisji są szyfrowane za pomocą XML security, to użyć należy w tym celu hybrydowego systemu kryptograficznego. W przypadku tym analogicznie do rozdziału 4.4 stosuje się klucz publiczny odbiorcy do szyfrowania kluczy sesji *(szyfrowanie klucza*), a właściwe pakiety danych są szyfrowane *(szyfrowanie treści*) przy użyciu symetrycznych metod szyfrowania. Przynależny klucz szyfrowania treści należy wygenerować losowo dla każdej transmisji.

3.4.1 Szyfrowanie treści

Do szyfrowania treści użyć należy metody wymienionej w tabeli 13.

Metoda	*Minimalna długość klucza*	*Użycie do*
Tryb AES GCM [18]	128	2028+

Tabela 13 Szyfrowanie treści w XML security

3.4.2 Szyfrowanie klucza

Szyfrowanie klucza realizowane jest albo za pomocą transportu klucza (*ang. key transport*) albo uzgadniania klucza (*ang. key agreement*).

3.4.3 Transport klucza

Do transportu klucza należy użyć metody z tabeli 14.

Metoda	*Minimalna długość klucza*	*Użycie do*
RSAES-OAEP [18]	2048	2022
	3072	2028+

Tabela 14 Transport klucza w XML Security

3.4.3.1. Uzgadnianie klucza

Do uzgadniania klucza należy użyć metody z tabeli 15. Do uzgadniania klucza należy użyć funkcji haszującej zgodnie z rozdziałem 3.2.

Metoda	*Minimalna długość klucza*	*Użycie do*
Uzgadnianie klucza		
ECDH [18]	224	2022
	256	2028+
Algorytm key-wrap		
AES-Wrap [18]	128	2028+

Tabela 15 Uzgadnianie klucza w XML security

3.5. Krzywe eliptyczne

W przypadku korzystania z krzywych eliptycznych można używać wyłącznie krzywych nazywanych (*ang. named curves*), w celu ochrony przed atakami przez nieweryfikowalne słabe parametry domeny. W. Należy używać krzywych nazwanych (patrz [6]) zgodnie z tabelą 16.

Krzywe eliptyczne	*Użycie do*
BrainpoolP224r1 [22]	2022
BrainpoolP256r1 [22]	2028+

Krzywe eliptyczne	*Użycie do*
BrainpoolP384r1 [22]	2028+
BrainpoolP512r1 [22]	2028+
NIST Curve P-224	2022
NIST Curve P-256	2028+
NIST Curve P-384	2028+
NIST Curve P-521	2028+

Tabela 16 Krzywe eliptyczne w XML security

Zaleca się stosowanie krzywych brainpool.

3.6. Minimalne wymagania dotyczące interoperacyjności

W celu zapewnienia zgodności z niniejszą wytyczną techniczną wymagana jest obsługa przynajmniej następujących metod:

- Funkcja haszująca:
  - ✓ SHA-256 z URI http://www.w3.org/2001/04/xmlenc#sha256 [23]
- Metoda podpisu:
  - ✓ ECDSA z URI http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256 [24]
- Szyfrowanie treści:
  - ✓ AES-128 (tryb GCM) z URI http://www.w3.org/2009/xmlenc11#aes128-gcm [23]
- Szyfrowanie klucza:
  - ✓ ECDH z URI http://www.w3.Org/2009/xmlenc11#ECDH-ES [23]
  - ✓ Key Wrap AES-128 z URI http://www.w3.Org/2001/04/xmlenc#kw-aes128 [23]
  - ✓ Key Derivation ConcatKDF z URI http://www.w3.org/2009/xmlenc11#ConcatKDF [23]

Ponadto wymagana jest obsługa krzywych eliptycznych BrainpoolP256r1 i NIST Curve P-256 dla metody ECC.

3.7. Wymagania przejściowe

W odróżnieniu od wymienionych powyżej wymagań w aplikacjach istniejących w uzasadnionych przypadkach wyjątkowych dopuszczalne są regulacje przejściowe zgodnie z tabelą 17.

Odstępstwo	*Zastosowanie maksymalnie do*	*Zalecenie*
Podpis		
RSASSA-PKCS1-v1_5	2022	Migracja do RSASSA-PSS lub ECDSA
Szyfrowanie treści		
Tryb AES CBC	2020	Migracja do trybu AES GCM

Tabela 17 Wymagania przejściowe dla XML security

W przypadku ich stosowania należy zapewnić odpowiednie środki zaradcze przeciwko atakom z wybranym szyfrogramem (por. [25], [26], [20]).

Niezależnie od podanego maksymalnego terminu użycia, zaleca się jak najszybszą migrację.

4. SPECYFIKACJA S/MIME

Rozszerzenia secure/multipurpose internet mail extension (S/MIME) to standard IETF służący do kryptograficznego zabezpieczania wiadomości typu MIME, takich jak e-maile. Umożliwiają one cyfrowe podpisywanie, szyfrowanie i kompresowanie wiadomości MIME.

W tym celu stosowane są klucze publiczne do szyfrowania danych lub weryfikacji podpisów. Przypisany klucz prywatny stosuje się do ponownego odszyfrowania zaszyfrowanych danych lub do składania podpisów. Klucz prywatny jest poufny i należy go chronić przed dostępem osób nieupoważnionych za pomocą odpowiednich środków, takich jak hasło, które powinno być znane tylko właścicielowi klucza.

S/MIME bazuje na standardzie CMS [27] i korzysta ze struktury danych CMS SignedData, EnvelopedData i CompressedData w celu zapisywania danych, które mają być zabezpieczone w kontenerze.

W przypadku S/MIME uwierzytelnianie partnerów komunikacyjnych opiera się na PKI za pośrednictwem certyfikatów X.509. Muszą być one wystawione przez jednostkę certyfikacyjną, aby zapewnić pełne zaufanie partnerów komunikacyjnych odnośnie certyfikatów. Należy przestrzegać specyfikacji z rozdziału 6.

4.1. Wersje

S/MIME jest dostępne w kilku częściach i wersjach:

- zaleca się korzystanie z wersji S/MIME 4.0. RFC 8551 [28] określa format wiadomości, a RFC 8550 [29] obsługę certyfikatu S/MIME 4.0.
- Wersję S/MIME 3.2 ([30], [31]) można dalej stosować w istniejących aplikacjach, pod warunkiem spełnienia wymagań niniejszych wytycznych technicznych dotyczących stosowanej kryptografii. Wersję S/MIME 3.1 ([32], [33]) można stosować w istniejących aplikacjach do końca roku 2022, pod warunkiem spełnienia wymagań niniejszych wytycznych technicznych dotyczących stosowanej kryptografii.
- Innych niż wymienione powyżej wersji S/MIME nie wolno stosować.

4.2. Funkcje haszujące

S/MIME korzysta z funkcji haszujących w szczególności do podpisywania wiadomości, a także do uzgadniania kluczy.

W tym celu należy zastosować funkcję haszującą z tabeli 18.

Metoda	*Minimalna długość wynikowa*	*Użycie do*
SHA-2 [19]	224	2022
	256	2028+
	512	2028+

Tabela 18 Funkcje haszujące dla S/MIME

4.3. Podpisy

W celu generowania podpisów przy pomocy S/MIME należy użyć jednej z metod podpisywania zgodnie z tabelą 19.

Metoda	*Minimalna długość klucza*	*Użycie do*
RSASSA-PSS [34]	2048	2022
	3072	2028+
ECDSA [21], [19]	224	2022
	256	2028+

Tabela 19 Metody podpisywania dla S/MIME

4.4. Szyfrowanie

S/MIME korzysta z hybrydowego systemu kryptograficznego w celu szyfrowania wiadomości. Szyfrowanie właściwych pakietów danych (*szyfrowania treści*) odbywa się w ramach symetrycznej metody szyfrowania. Przynależny klucz sesji *(ang. session key)* jest generowany losowo, a klucz publiczny odbiorcy jest wykorzystywany do szyfrowania kluczy sesji *(ang. key encryption*).

4.4.1 Szyfrowanie treści

Do szyfrowania treści użyć należy metody wymienionej w tabeli 20.

Metoda	*Minimalna długość klucza*	*Użycie do*
AES CBC-Mode [35]	128	2025
Tryb AES GCM [28]	128	2028+

Tabela 20 Szyfrowanie treści w S/MIME

4.4.2 Szyfrowanie klucza

W zależności od zastosowanej kryptografii, klucz szyfrowania treści jest szyfrowany asymetrycznie bezpośrednio za pomocą klucza publicznego odbiorcy (transport klucza) lub nadawca generuje efemeryczną parę kluczy i generuje klucz symetryczny (uzgadnianie klucza) na jego podstawie oraz na podstawie klucza publicznego odbiorcy, który następnie wykorzystuje się do symetrycznego szyfrowania klucza szyfrowania treści.

Do asymetrycznego szyfrowania klucza za pomocą transportu klucza używać należy metody wymienionej w tabeli 21.

Metoda	*Minimalna długość klucza*	*Użycie do*
RSAES-OAEP [20]	2048	2022
	3072	2028+

Tabela 21 Asymetryczne szyfrowanie klucza przy pomocy transportu klucza w S/MIME

Do symetrycznego szyfrowania klucza metodą uzgadniania klucza użyć należy metody wymienionej w tabeli 22.

Metoda	*Minimalna długość klucza*	*Użycie do*
Uzgadnianie klucza		
ECDH [21]	224	2022
	256	2028+
Algorytm key-wrap		
AES-Wrap [35]	128	2028+

Tabela 22 Symetryczne szyfrowanie klucza przy pomocy uzgadniania klucza w S/MIME

Do uzgadniania klucza symetrycznego w celu szyfrowania klucza za pomocą DH lub ECDH należy zastosować funkcję haszującą wymienioną w tabeli 18. Dodatkowo do uzgadniania klucza wykorzystać należy dodatkowe dane efemeryczne.

4.5. Krzywe eliptyczne

W przypadku korzystania z krzywych eliptycznych można używać wyłącznie krzywych nazywanych (*ang. named curves*), w celu ochrony przed atakami przez nieweryfikowalne słabe parametry domeny. Należy używać krzywych nazwanych (patrz [6]) zgodnie z tabelą 23.

Krzywe eliptyczne	*Użycie do*
BrainpoolP224r1 [22]	2022
BrainpoolP256r1 [22]	2028+
BrainpoolP384r1 [22]	2028+
BrainpoolP512r1 [22]	2028+
NIST Curve P-224	2022
NIST Curve P-256	2028+
NIST Curve P-384	2028+
NIST Curve P-521	2028+

Tabela 23 Krzywe eliptyczne w S/MIME

Zaleca się stosowanie krzywych brainpool.

4.6. Pozostałe specyfikacje

- Implementacje S/MIME mogą być podatne na ataki z wybranym szyfrogramem lub inne ataki pokrewne (por. na przykład [26], [36]): Implementacje szyfrowania RSA mogą stosunkowo często wykazywać luki bezpieczeństwa. Do szyfrowania treści w S/MIME dotychczas wykorzystywano głównie symetryczne algorytmy szyfrowania bez zabezpieczania integralności. Ponadto implementacje S/MIME powinny obsługiwać dowolne rozgałęzienia kontenerów podpisu i szyfrowania.

  Dla implementacji S/MIME należy w związku z tym przewidzieć odpowiednie środki zaradcze, tak aby ataki takie nie były możliwe w praktyce (por. [20]):

Zasadniczo zaleca się szyfrowanie uwierzytelnione. Poza tym nie należy stosować lub wykonywać żadnych aktywnych treści. Ponadto zaleca się zastosowanie dodatkowych środków bezpieczeństwa na poziomie komunikacji w celu zapobiegania lub wykrywania ataków z wybranym szyfrogramem, patrz na przykład [37] (TLS i DNSSEC/DANE)[7].

- Implementacja S/MIME powinna w momencie odebrania podpisanej lub wysłania zaszyfrowanej wiadomości S/MIME uruchomić walidację użytych certyfikatów, aby zapobiec atakom z odmową świadczenia usługi (*ang denial-of-service*) w związku z nieprawidłowymi kluczami o ekstremalnych długościach.

4.7. Minimalne wymagania dotyczące interoperacyjności

W celu zapewnienia zgodności z niniejszą wytyczną techniczną wymagana jest obsługa przynajmniej następujących metod:

- Funkcja haszująca: SHA-256.
- Metoda podpisu: RSASSA-PSS, ECDSA.
- Szyfrowanie asymetryczne: RSAES-OAEP, ECDH.
- Szyfrowanie symetryczne: AES-128 (CBC-Mode, od 2025: GCM-Mode).

Ponadto wymagana jest obsługa krzywych eliptycznych BrainpoolP256r1 i NIST Curve P-256 dla metody ECC.

4.8. Wymagania przejściowe

W odróżnieniu od wymienionych powyżej wymagań w aplikacjach istniejących w uzasadnionych przypadkach wyjątkowych dopuszczalne są regulacje przejściowe zgodnie z tabelą 24.

[7] Środki bezpieczeństwa na poziomie transportu sprawdzają się również do ochrony danych nagłówka.

Odstępstwo	*Zastosowanie maksymalnie do*	*Zalecenie*
Podpis		
RSASSA-PKCS1-v1_5 [38], [19]	2022	Migracja do RSASSA-PSS lub ECDSA

Tabela 24 Wymagania przejściowe dla S/MIME

W przypadku zastosowania zasad przejściowych należy zapewnić odpowiednie środki zaradcze przeciwko atakom z wybranym szyfrogramem (por. [25], [26], [20]).

Metody te mogą być obsługiwane w celu szyfrowania wiadomości i weryfikacji podpisów w istniejących aplikacjach, o ile jest to konieczne w celu zapewnienia interoperacyjności z istniejącymi partnerami komunikacyjnymi, którzy nie spełniają warunków zgodności.

Niezależnie od podanego *maksymalnego* okresu stosowania, zaleca się jak najszybszą migrację.

5. SPECYFIKACJA OPENPGP

Pretty good privacy (PGP) to system kryptograficznego zabezpieczania danych, zwłaszcza wiadomości e-mail i plików. PGP umożliwia cyfrowe podpisywanie i szyfrowanie danych.

Standard ten polega zasadniczo na używaniu przez użytkowników kluczy publicznych partnera komunikacyjnego do przesyłania mu zaszyfrowanych danych lub do weryfikacji podpisu partnera na danych otrzymywanych od niego. Wyłącznie odpowiedni właściciel klucza posiada przynależne klucze prywatne. Służą one do odszyfrowywania zaszyfrowanych danych lub generowania podpisów przez właściciela klucza i są najczęściej chronione hasłem.

Uwierzytelnianie partnerów komunikacyjnych bazuje na sieci zaufania (*ang. web of trust*). Należy przestrzegać specyfikacji z rozdziału 6. W przypadku standardu OpenPGP to sami użytkownicy końcowi ponoszą odpowiedzialność za zapewnienie zaufania do certyfikatów, podobnie jak ich zwrot. W tym celu wymagana jest znaczna wiedza specjalistyczna odnośnie funkcjonowania sieci zaufania, a kontrola zgodności z niezbędnymi wymogami w zakresie identyfikacji jest możliwa tylko w bardzo ograniczonym zakresie. OpenPGP należy używać więc wyłącznie pod warunkiem zapewnienia, że wszyscy uczestnicy posiadają odpowiednią wiedzę i spełnione są wszystkie wymagania odnośnie identyfikacji. W takim przypadku spełnić należy wymagania niniejszego załącznika.

Istnieją liczne, częściowo niekompatybilne wersje standardu PGP.
Standard OpenPGP (opierający się na PGP 5.x) to aktualny międzynarodowy standard internetowy [39].

5.1. Wersje

Stosować należy PGP w wersji standaryzowanej OpenPGP zgodnie z [39]. Zaleca się zapewnienie obsługi kryptografii krzywych eliptycznych (ECC) zgodnie z [40].

Dla e-maili podpisywanych i zaszyfrowanych stosować należy format wiadomości PGP/MIME zgodnie z [41]. Nie zaleca się używania formatu PGP/INLINE.

5.2. Funkcje haszujące

Należy zastosować funkcję haszującą z tabeli 25.

Metoda	*Minimalna długość klucza/wyjściowa*	*Użycie do*
SHA-2 [39]	224	2022
	256	2028+

Tabela 25 Funkcje haszujące w OpenPGP

5.3. Podpisy

W przypadku korzystania ze standardu OpenPGP do generowania podpisów należy użyć metody z tabeli 26.

Metoda	*Minimalna długość klucza/wyjściowa*	*Użycie do*
RSASSA-PKCS1-v1_5 [39]	2048	2019
DSA [39]	2048	2022
	3072	2028+
ECDSA [40]	256	2028+

Tabela 26 Metody podpisywania w OpenPGP

5.4. Szyfrowanie

W standardzie OpenPGP właściwe pakiety danych (*szyfrowanie treści)* są szyfrowane symetrycznie, przy czym przynależny klucz sesji (*ang. session key*) jest generowany losowo i zasadniczo szyfrowany asymetrycznie przy pomocy publicznego klucza odbiorcy (*szyfrowanie klucza sesji*). Alternatywnie OpenPGP umożliwia również symetryczne szyfrowanie kluczy sesji za pomocą wynegocjowanych odgórnie danych poufnych (hasła).

Jeśli jest to możliwe w danej aplikacji, to użyć należy asymetrycznego szyfrowania klucza.

5.4.1 Szyfrowanie pakietów danych (szyfrowanie treści)

Do szyfrowania pakietów danych użyć należy metody wymienionej w tabeli 27.

Metoda	*Minimalna długość klucza*	*Użycie do*
AES CFB-Mode[8] [39]	128	2028+

Tabela 27 Symetryczne szyfrowanie pakietów danych (szyfrowanie treści) za pomocą OpenPGP

Pakiety danych należy chronić przed fałszowaniem dzięki użyciu kodu wykrywania modyfikacji (*ang. modification detection code*), (*ang. symmetrically encrypted integrity protected data packets*).

5.4.2 Asymetryczne szyfrowanie kluczy sesji

Do asymetrycznego szyfrowania kluczy sesji używać należy metody wymienionej w tabeli 28.

Metoda	*Minimalna długość klucza*	*Użycie do*
ElGamal [39]	2048	2022
	3072	2028+

Tabela 28 Asymetryczne szyfrowanie kluczy sesji za pomocą OpenPGP

Szyfrowanie możliwe jest również przy pomocy uzgadniania kluczy. W tym celu spełnić należy wymagania z tabeli 29. Zalecenia dotyczące łączenia odpowiednich długości kluczy opisano w [40].

Metoda	*Minimalna długość klucza*	*Użycie do*
Uzgadnianie klucza		
ECDH [40]	256	2028+
Algorytm szyfrowania KEK		
AES-Wrap [40]	128	2028+

Tabela 29 Szyfrowanie kluczy sesji za pomocą OpenPGP przez uzgadnianie klucza

[8] Trybem pracy używanym przez OpenPGP jest wariant trybu CFB (por. [39]).

Do uzgadniania klucza symetrycznego za pomocą ECDH z shared secret należy zastosować dopuszczalną funkcję haszującą wymienioną w tabeli 25.

5.4.3 Symetryczne szyfrowanie kluczy sesji i ochrona klucza prywatnego

Jeśli wymaga tego dana aplikacja, to klucze sesji szyfrować można również za pomocą danych tajnych (frazy hasła) uzgodnionych z góry. Ponadto, frazy hasła są używane w OpenPGP w celu zapewnienia ochrony klucza prywatnego. W tym celu spełnić należy wymagania z tabeli 30.

Metoda	*Minimalna długość klucza*	*Użycie do*
AES CFB-Mode [39]	128	2028+

Tabela 30 Symetryczne szyfrowanie kluczy sesji w OpenPGP

W procedurze uzgadniania klucza do szyfrowania klucza metodą szyfrowania symetrycznego uwzględnić należy dodatkowe dane efemeryczne (*ang. salt value*).

Do uzgadniania klucza symetrycznego frazy hasła należy zastosować dopuszczalną funkcję haszującą wymienioną w tabeli 25.

5.5. Krzywe eliptyczne

W przypadku korzystania z krzywych eliptycznych można używać wyłącznie krzywych nazywanych (*ang. named curves*), w celu ochrony przed atakami przez nieweryfikowalne słabe parametry domeny. Należy używać krzywych nazwanych (patrz [6]) zgodnie z tabelą 31.

Krzywe eliptyczne	*Użycie do*
NIST Curve P-256	2028+
NIST Curve P-384	2028+
NIST Curve P-521	2028+

Tabela 31 Krzywe eliptyczne w OpenPGP

5.6. Pozostałe specyfikacje

- Szyfrowanie RSA jest zasadniczo podatne na ataki z wybranym szyfrogramem (por. [25], [26]). Dla implementacji należy w związku z tym przewidzieć odpowiednie środki zaradcze, tak aby ataki takie nie były możliwe w praktyce (por. [42]).
- Do szyfrowania treści wykorzystywane są symetryczne algorytmy szyfrowania OpenPGP bez zabezpieczania integralności. Są one zasadniczo podatne na ataki z wybranym szyfrogramem lub inne ataki pokrewne (por. na przykład [36], [43]). W związku z tym należy podjąć odpowiednie środki zabezpieczające przed atakami z wybranym szyfrogramem na symetryczne szyfrowanie implementacji OpenPGP. W szczególności nie należy stosować lub wykonywać żadnych aktywnych treści.
- Implementacja OpenPGP powinna w momencie odebrania podpisanej lub wysłania zaszyfrowanej wiadomości PGP uruchomić walidację zastosowanych certyfikatów, tak aby zapobiec atakom z odmową świadczenia usługi (*ang. denial-of-service*) w związku z nieprawidłowymi kluczami o ekstremalnych długościach.

5.7. Minimalne wymagania dotyczące interoperacyjności

W celu zapewnienia zgodności z niniejszą wytyczną techniczną wymagana jest obsługa przynajmniej następujących metod:

- Funkcja haszująca: SHA-256,
- Metoda podpisu: DSA, ECDSA,
- Szyfrowanie asymetryczne: ElGamal, ECDH,
- Szyfrowanie symetryczne: AES-128.

Ponadto zapewnić należy obsługę krzywej eliptycznej NIST Curve P-256.

6. IDENTYFIKACJA PARTNERÓW KOMUNIKACYJNYCH

W celu zapewnienia bezpiecznej komunikacji zachodzi konieczność identyfikacji jednego lub kilku uczestników. Procedury opisane w niniejszej wytycznej wykorzystują przypisywanie asymetrycznej pary kluczy do podmiotu w celu jego identyfikacji. Przypisanie pary kluczy realizuje się przy wykorzystaniu infrastruktury klucza publicznego (patrz 6.1) lub bezpośredniej dwustronnej wymiany kluczy publicznych lub certyfikatów za pośrednictwem zaufanego kanału (patrz 6.2).

6.1. Identyfikacja oparta na PKI

SSL/TLS, S/MIME i XML Security (takie jak np. SAML) umożliwiają obsługę opartą na PKI w celu identyfikacji jednego lub obydwu partnerów komunikacyjnych. Używana w tym celu struktura PKI, *Internet PKI*, jest opisana w [44].

6.1.1 Jednostki certyfikacyjne/ kotwice zaufania

W przypadku korzystania z identyfikacji opartej na PKI, certyfikaty partnerów komunikacyjnych wydaje jedna lub kilka jednostek certyfikacyjnych (*ang. Certification authority - CA*). Aplikacja musi posiadać jedną lub kilka kotwic zaufania w celu weryfikacji certyfikatów, tj. certyfikatów głównych (*ang. root*) uznawanych jednostek certyfikacyjnych.

Dobór jednostek certyfikacyjnych wystawiających certyfikaty oraz dobór kotwic zaufania należy wykonać szczególnie starannie. W procedurze wyboru należy wziąć pod uwagę w szczególności następujące kryteria:

- transparentne zasady wydawania certyfikatów i funkcji CA, publikowane w wytycznej certyfikatu (*ang. certificate policy*);
- bezpieczeństwo IT funkcji CA, potwierdzone w ramach audytu/ certyfikacji zgodnie z obowiązującym standardem audytu/certyfikacji;
  - ✓ zaleca się certyfikację zgodnie z BSI TR-03145 [45];
- wysoki poziom bezpieczeństwa usług rejestracyjnych, w tym usług podzlecanych zewnętrznym usługodawcom (rejestratorom);
- wiarygodność operatora i działania, również po uwzględnieniu praw dostępowych osób trzecich;

- dostępność opcji zwrotu (wycofania);
- status prawny, w szczególności w odniesieniu do obowiązujących przepisów prawa odpowiedzialności i ochrony danych.

W przypadku aplikacji wymagających osiągnięcia wysokiego poziomu zaufania, jednostki certyfikacyjne muszą dysponować certyfikatem TR zgodnie z [45].

Liczba kotwic zaufania powinna być możliwie najmniejsza.

6.1.2 Certyfikaty

Struktura certyfikatu jest opisana w [44], ale może być ograniczona lub rozbudowana o dodatkowe rozszerzenia w odniesieniu do konkretnego przypadku zastosowania.

W celu zapewnienia zgodności z niniejszą wytyczną certyfikaty użytkowników końcowych oraz certyfikaty CA dla aplikacji muszą spełniać następujące wymagania:

- wszystkie certyfikaty muszą zawierać informacje niezbędne do wykonania kontroli zwrotu, tzn.
  - ✓ punkt cRLDistributionPoint, który udostępnia obowiązujące aktualnie CRL, lub
  - ✓ rozszerzenie AuthorityInfoAccess, które zawiera informacje niezbędne do kontroli serwera OCSP.
- Wszystkie certyfikaty muszą posiadać odpowiedni okres ważności.
  - ✓ Okres ważności certyfikatów użytkowników końcowych nie może przekraczać trzech lat.
  - ✓ Okres ważności certyfikatów Sub-CA nie może przekraczać maksymalnie pięciu lat.
  - ✓ Kotwice zaufania (certyfikaty główne) nie powinny przekraczać maksymalnego okresu ważności wynoszącego 6 lat. Jeśli zaufana jednostka w przypadku kompromitacji/ wygaśnięcia przydatności algorytmów kryptograficznych nie zapewnia aktualizacji kotwic zaufania, to wybrać należy odpowiednio krótszy okres ważności.

- Certyfikaty CA muszą zawierać rozszerzenie BasicConstraints oznaczone jako krytyczne. Certyfikaty CA muszą w rozszerzeniu zawierać pole pathLenConstraint ustawione na jak najmniejszą możliwą wartość.
- Wszystkie certyfikaty muszą zawierać rozszerzenie KeyUsage, które w możliwie jak największym stopniu ograniczać powinno prawa związane z certyfikatem i być oznaczone jako krytyczne. Certyfikaty użytkowników końcowych powinny ponadto zawierać rozszerzenie ExtendedKeyUsage, które w możliwie jak największym stopniu ogranicza prawa związane z certyfikatem.
- Dla różnych celów zastosowania (podpis, szyfrowanie, uwierzytelnianie itp.) należy w miarę możliwości wygenerować różne pary kluczy oraz wystawiać i wykorzystywać różne certyfikaty.
- Certyfikaty nie mogą zawierać symboli wieloznacznych (*ang. wildcards*) w CommonName podmiotu lub SubjectAltName.

Dla aplikacji obsługiwanych przez przeglądarki (jak strony internetowe) zaleca się stosowanie kwalifikowanych certyfikatów witryn internetowych zgodnie z [46] lub certyfikatów o rozszerzonej walidacji, zwłaszcza jeśli aplikacja przetwarza dane osobowe.

6.1.3 Weryfikacja certyfikatu

W ramach procedury walidacji certyfikatu należy stosować się do zasad opisanych w [44], rozdział 6 „Walidacja ścieżki certyfikacji”. Procedura ta obejmuje w szczególności:

- pełną weryfikację łańcucha certyfikatów aż do kotwicy zaufania o potwierdzonym poziomie zaufana i autentyczności dla danej aplikacji (por. 6.1.1);
- Weryfikacja ważności (daty wystawienia i wygaśnięcia);
- Weryfikacja zwrotu *wszystkich* certyfikatów w łańcuchu;
  - ✓ W przypadku TLS zaleca się stosowanie OCSP stapling ze względu na wydajność i ochronę danych, por. rozdział 2.2.3.3.
- Analiza rozszerzeń zawartych w certyfikatach na podstawie zasad opisanych w [44], w szczególności wszystkich rozszerzeń opisanych w rozdziale 6.1.2.

W niektórych zastosowaniach można odstąpić od wymagań z niniejszego rozdziału, ale tylko w uzasadnionych przypadkach i w porozumieniu z BSI.

6.1.4 Parametry domeny i długości kluczy

Klucze do podpisywania certyfikatów muszą spełniać przynajmniej wymagania określone w tabeli 32.

Algorytm	*Minimalna długość klucza*	*Minimalna długość końcowa funkcji haszującej*	*Użycie do*
ECDSA [47]	224 bitów	SHA-224	2021
	256 bitów	SHA-256	2028+
DSA [47]	2048 Bit	SHA-224	2021
	3072 bitów	SHA-256	2028+
RSASSA-PSS [48]	2048 bitów	SHA-224	2021
	3072 bitów	SHA-256	2028+

Tabela 32 Algorytmy podpisu i minimalne długości kluczy dla certyfikatów X.509

Do podpisywania certyfikatów zaleca się używanie kluczy o długości bitowej przynajmniej identycznej jak długość klucza zawartego w certyfikacie. Ponadto dla certyfikatów głównych (*ang. root*) zaleca się stosowanie w razie możliwości kluczy dłuższych niż dla certyfikatów podrzędnych lub kluczy użytkowników końcowych.

W przypadku korzystania z krzywych eliptycznych (ECC) można używać wyłącznie krzywych nazywanych (*ang. named curves*), w celu ochrony przed atakami przez nieweryfikowalne słabe parametry domeny. Należy używać krzywych nazwanych (patrz [6]) zgodnie z tabelą 33.

Krzywe eliptyczne	*Użycie do*
BrainpoolP224r1[9] [22]	2021
BrainpoolP256r1 [22]	2028+

[9] Protokół TLS nie posiada ID dla tej krzywej.

Krzywe eliptyczne	*Użycie do*
BrainpoolP384r1 [22]	2028+
BrainpoolP512r1 [22]	2028+
NIST Curve P-224	2021
NIST Curve P-256	2028+
NIST Curve P-384	2028+
NIST Curve P-521	2028+

Tabela 33 Krzywe eliptyczne dla certyfikatów X.509

Zaleca się stosowanie ECC z krzywymi typu Brainpool.

6.1.4.1. Wymagania przejściowe

W odróżnieniu do powyższej specyfikacji w aplikacjach wykorzystywanych do generowania certyfikatów w przypadkach uzasadnionych i wyjątkowych stosować można następujące reguły przejściowe opisane w tabeli 34 wraz z odpowiadającymi im długościami kluczy z tabeli 32.

Odstępstwo	*Zastosowanie maksymalnie do*	*Zalecenie*
RSASSA-PKCS1-v1_5	2022	Migracja do RSASSA-PSS lub ECDSA

Tabela 34 Reguły przejściowe dotyczące podpisywania certyfikatów

Niezależnie od podanego *maksymalnego* okresu stosowania, zaleca się jak najszybszą migrację.

6.2. Identyfikacja przez dwustronną wymianę kluczy lub sieć zaufania

Identyfikacja podmiotu w ramach wymiany kotwic zaufania PKI i w sieci zaufania (np. OpenPGP) nie bazuje na PKI.

Kotwice zaufania PKI korzystają z certyfikatów samopodpisanych, których autentyczność zapewnia zaufana dwustronna wymiana kluczy.

W sieci zaufania uczestnicy stosują certyfikaty samopodpisane, których autentyczność potwierdzają decentralnie podpisy innych podmiotów sieci zaufania. Certyfikaty samopodpisane można stosować również w SAML, w zależności od scenariusza aplikacji.

6.2.1 Identyfikacja właścicieli certyfikatów

Tożsamość podmiotu w takich systemach nie jest weryfikowana centralnie przez jednostkę rejestracyjną instancji certyfikacyjnej, ale na drodze przesłania certyfikatu przez zaufany kanał komunikacyjny.

Zaufanie do certyfikatu podmiotu musi należy zapewnić za pomocą jednej z poniższych procedur.

1. Bezpośrednia dwustronna wymiana certyfikatów samopodpisanych za pośrednictwem zaufanego kanału. Partnerzy komunikacyjni ustalają i weryfikują dwustronnie wymagania bezpieczeństwa, które należy spełnić. Wymagania bezpieczeństwa muszą być zdefiniowane w taki sposób, aby partnerzy komunikacyjni otrzymali poziom zaufania względem autentyczności certyfikatów odpowiedni do potrzeb ochrony aplikacji. Przykładami takich wymagań mogą być: osobista wymiana certyfikatów z uprzednią identyfikacją za pomocą identyfikatora lub porównanie odcisku palca certyfikatu w niezależnym i uwierzytelnionym kanale.
2. (Sieć zaufana) Podpisywanie certyfikatów przez zaufaną stronę trzecią. W tym przypadku należy zapewnić zarówno uwierzytelnienie zaufanej strony trzeciej, jak i podpisanego klucza. Odpowiednie wymogi bezpieczeństwa spełnić muszą wszystkie uczestniczące podmioty, na których opiera się uwierzytelnianie.

Szczególny nacisk należy położyć na wysoki poziom bezpieczeństwa procesu identyfikacji.

6.2.2 Przekazywanie certyfikatów

Jeśli podmiot w sieci zaufania udostępnia klucze osób trzecich partnerowi komunikacyjnemu, to podmiot taki musi zapewnić na mocy umowy spełnienie wymaganego poziomu bezpieczeństwa identyfikacji przez wszystkie uczestniczące jednostki.

Serwery kluczy lub listy master umożliwiają publikowanie certyfikatów.

6.2.3 Zwrot

Zwrot (cofnięcie) certyfikatów stanowi szczególny problem w systemach nieopartych na PKI, ponieważ nie ma pewności, że procedura zwrotu (na przykład z powodu kompromitacji klucza) zostanie udostępniona do ogólnej wiadomości.

Jeśli certyfikaty wymienia się dwustronnie, to właściciel musi niezwłocznie poinformować wszystkich bezpośrednich partnerów komunikacyjnych, z którymi dwustronnie wymieniał certyfikaty o zwrocie/wycofaniu klucza.

W sieci zaufania należy dodatkowo opublikować na serwerach kluczy certyfikat zwrotu/wycofania klucza, którego właściciel będzie wiedział, że dany klucz został opublikowany.

Kluczy należących do wycofanych certyfikatów nie wolno używać.

6.2.4 Parametry domeny i długości kluczy

Podczas identyfikacji metodą dwustronnej wymiany certyfikatów, parametry domeny i długości kluczy, które należy zastosować, wynikają ze specyfikacji odpowiednich kluczy podpisów właścicieli certyfikatów lub autorów podpisów. Okres ważności certyfikatów może wynosić maksymalnie 5 lat.

7. KLUCZE KRYPTOGRAFICZNE

7.1. Generowanie

Klucze kryptograficzne należy generalnie generować pod kontrolą właściciela klucza. Generowanie klucza, np. przez CA wystawiającą certyfikat, jest dopuszczalne tylko w uzasadnionych, wyjątkowych przypadkach. W takim przypadku należy zapewnić, że po przekazaniu klucza właścicielowi, generująca jednostka nie zatrzyma sobie kopii, a sama procedura przekazania będzie poufna.

7.2. Liczby losowe

Do generowania liczb losowych, np. w celu generowania kluczy kryptograficznych lub generowania podpisów, należy użyć odpowiednich generatorów liczb losowych.

Zaleca się użycie generatora liczb losowych jednej z klas DRG.3, DRG.4, PTG.3 lub NTG.1 zgodnie z [49]. Dodatkowe informacje na temat generowania kluczy asymetrycznych można również znaleźć w [1], załącznik B.

7.3. Zapisywanie i przetwarzanie

Prywatne klucze kryptograficzne, w szczególności klucze statyczne i klucze podpisów należy zapisywać i przetwarzać w sposób bezpieczny. Oznacza to między innymi ochronę przed kopiowaniem, użyciem przez niepowołane osoby i manipulacją kluczami. Bezpieczne zapisywanie kluczy gwarantuje na przykład odpowiednio certyfikowany sprzęt (karta chipowa, HSM).

Klucze publiczne podmiotów uznanych za zaufane (kotwice zaufania) oraz klucze wymieniane obustronnie należy zapisywać również w sposób chroniący je przed manipulacjami.

7.4. Niszczenie

Prywatne klucze kryptograficzne, dane poufne itp. należy usuwać natychmiast, gdy tylko przestaną być potrzebne. Usuwanie musi odbywać się w sposób zabezpieczony. Zwykła dezaktywacja klucza nie jest wystarczająca.

8. REFERENCJE

NARODOWE STANDARDY CYBERBEZPIECZEŃSTWA[10]	
NSC 800-52	Wskazówki dotyczące wyboru, konfigurowania i wykorzystania implementacji TLS (*ang. Transport Layer Security*)

[10] Narodowe Standardy Cyberbezpieczeństwa

PUBLIKACJE OBCOJĘZYCZNE[11]	
[1]	BSI TR-02102-1, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Teil 1, 2021
[2]	BSI TR-02102-2, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Teil 2 - Verwendung von Transport Layer Security (TLS), 2021
[3]	IETF RFC 5246, T. Dierks, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, 2008
[4]	IETF RFC 8446, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.3, 2018
[5]	IETF RFC 8996, S. Farrell, K. Moriarty, Deprecating TLS 1.0 and TLS 1.1, 2021
[6]	IANA, http://www.iana.org/assignments/tls-parameters/tls-parameters.xml
[7]	IETF RFC 7027, J. Merkle, M. Lochter, Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS), 2013
[8]	IETF RFC 7919, D. Gillmor, Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS), 2016
[9]	N. AlFardan, K. Paterson, Lucky Thirteen: Breaking the TLS and DTLS Record Protocols, http://www.isg.rhul.ac.uk/tls/
[10]	M. Bellare, C. Namprempre, Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm; in Advances in Cryptology - Asiacrypt 2000 Proceedings, Lecture Notes in Computer Science Vol. 1976, T. Okamoto ed, Springer-Verlag, 2000
[11]	IETF RFC 7366, P. Gutman, Encrypt-then-MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), 2014
[12]	IETF RFC 6961, Y. Pettersen, The Transport Layer Security (TLS) Multiple Certificate Status Request Extension, 2013

[11] Publikacje zostały podane w celach uzupełniających dla osób zainteresowanych.

[13]	IETF RFC 6066, D. Eastlake, Transport Layer Security (TLS) Extensions: Extension Definitions, 2011
[14]	K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Pironti, P.-Y. Strub, Triple Handshake and Cookie Cutters: Breaking and Fixing Authentication over TLS, IEEE Symposium on Security and Privacy, 2014
[15]	IETF RFC 7627, K. Bhargavan, Ed., A. Delignat-Lavaud, A. Pironti, A. Langley, M. Ray, Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension, 2015
[16]	IETF RFC 8734, L. Bruckert, J. Merkle, M. Lochter, Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS) Version 1.3, 2020
[17]	W3C, XML Signature Syntax and Processing Version 1.1
[18]	W3C, XML Encryption Syntax and Processing Version 1.1
[19]	IETF RFC 5754, S. Turner, Using SHA2 Algorithms with Cryptographic Message Syntax, 2010
[20]	IETF RFC 8017, K. Moriarty, J. Jonsson, B. Kaliski, A. Rusch, PKCS #1: RSA Cryptography Specifications Version 2.2, 2016
[21]	IETF RFC 5753, S. Turner, D. Brown, Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS), 2010
[22]	IETF RFC 5639, M. Lochter, J. Merkle, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, 2010
[23]	W3C, https://www.w3.org/TR/xmlenc-core1
[24]	IETF RFC 6931, D. Eastlake, Additional XML Security Uniform Resource Identifiers (URIs), 2013
[25]	D. Bleichenbacher, Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1, Advances in Cryptology - Crypto '98, Lecture Notes in Computer Science, vol. 1462, pp. 1-12, Springer Verlag, 1998
[26]	J. Manger, A Chosen Ciphertext Attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as Standardized in PKCS #1 v2.0, Advances in Cryptology - Crypto 2001, Lecture Notes in Computer Science, vol. 2139, pp. 260-274, Springer Verlag, 2001

[27]	IETF RFC 5652, R. Housley, Cryptographic Message Syntax (CMS), 2009
[28]	IETF RFC 8551, J. Schaad, B. Ramsdell, S. Turner, Secure/Multipurpose Internat Mail Extension (S/MIME) Version 4.0 Message Specification, 2019
[29]	IETF RFC 8550, J. Schaad, B. Ramsdell, S. Turner, Secure/Multipurpose Internat Mail Extension (S/MIME) Version 4.0 Certificate Handling, 2019
[30]	IETF RFC 5751, B. Ramsdell, S. Turner, Secure/Multipurpose Interenet Mail Extensions (S/MIME) Version 3.2 Message Specification, 2010
[31]	IETF RFC 5750, B. Ramsdell, S. Turner, Secure/Multipurpose Internet Mail Extensions (S/ MIME) Version 3.2 Certificate Handling, 2010
[32]	IETF RFC 3851, B. Ramsdell, Secure/Multipurpose Mail Extensions (S/MIME) Version 3.1 Message Specification, 2004
[33]	IETF RFC 3850, B. Ramsdell, Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 - Certificate Handling, 2004
[34]	IETF RFC 4056, J. Schaad, Use of the RSASSA-PSS Signature Algorithm in Cryptographic Message Syntax (CMS), 2005
[35]	IETF RFC 3565, J. Schaad, Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS), 2003
[36]	J. Katz, B. Schneier, A Chosen Ciphertext Attack Against Several E-Mail Encryption Protocols, Usenix Security Symposium 2000
[37]	BSI TR-03108, Secure E-Mail Transport
[38]	IETF RFC 3370, R. Housley, Cryptographic Message Syntax (CMS) Algorithms, 2003
[39]	IETF RFC 4880, J. Callas, L. Donnerhacke, H. Finney, D. Shaw, R. Thayer, OpenPGP Message Format, 2007
[40]	IETF RFC 6637, A. Jivsov, Elliptic Curve Cryptography (ECC) in OpenPGP, 2012
[41]	IETF RFC 3156, M. Elkins, D. Del Torto, R. Levien, T. Rossler, MIME Security with OpenPGP, 2001
[42]	IETF RFC 3447, J. Jonsson, B. Kaliski, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, 2003

[43]	K. Jallal, J. Katz, J. J. Lee, B. Schneier, Implementation of Chosen Ciphertext Attacks against PGP and GnuPGP, Information Security 5th International Conference, Lecture Notes in Computer Science, vol. 2433, pp. 90-101, Springer Verlag, 2002
[44]	IETF RFC 5280, D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, 2008
[45]	BSI TR-03145, Secure Certification Authority operation
[46]	Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC
[47]	IETF RFC 5758, Q. Dang, S. Santesson, K. Moriarty, D. Brown, T. Polk, Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA, 2010
[48]	IETF RFC 4055, J. Schaad, B. Kaliski, R. Housley, Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, 2005
[49]	BSI AIS 20/31, A proposal for: Functionality classes for random number generators, 2011