**Załącznik nr 7 do umowy**

**Zabezpieczenie komunikacji   
z Systemem e-Płatności**

# Wprowadzenie

Dokument zawiera opis technik stosowanych przy zabezpieczeniu komunikacji systemów zewnętrznych z Systemem e-Płatności.

Zabezpieczenia można wyróżnić w następujących warstwach:

* w warstwie sieciowej - na poziomie konfiguracji adresów IP systemów uprawnionych do komunikacji z Systemem e-Płatności
* w warstwie prezentacji - na poziomie zabezpieczenia sesji komunikacyjnej pomiędzy systemami
* w warstwie aplikacji - na poziomie zabezpieczenia komunikatów przesyłanych między systemami

## Terminologia

|  |  |
| --- | --- |
| Termin | Opis |
| e-Płatności | System elektronicznych płatności realizowanych dla Sądownictwa |
| System zewnętrzny | Dowolny system komunikujący się z Systemem e-Płatności w zakresie realizacji lub zlecania płatności (Operator Płatności lub System Merytoryczny) |
| Operator Płatności | Każda zewnętrzna instytucja obsługująca płatności elektroniczne na podstawie obowiązujących przepisów prawa |
| System Merytoryczny | Systemy informatyczne funkcjonujące w ramach Resortu i realizujące podstawowe cele sądownictwa |
| back-end | Metoda polegająca na wywoływaniu skryptów po stronie serwera, z pominięciem przeglądarki webowej (komunikacja wewnętrzna) |
| SHA-256 | Funkcja skrótu przedstawiona w FIPS 180-4 |
| HMAC SHA-256 | Kod uwierzytelniający wiadomość bazujący na funkcji skrótu SHA-256 (FIPS 198-1) |

# Zabezpieczenie w warstwie sieciowej

Zabezpieczenie w warstwie sieciowej wdrażane jest w przypadku komunikacji typu back-end.

W celu ochrony przed niechcianymi wywołaniami, po stronie systemu zewnętrznego zostaną dodane adresy IP serwerów (lub zakresy adresów IP), z których będą przychodziły zapytania pochodzące z Systemu e-Płatności.

Analogicznie, po stronie Systemu e-Płatności, wymagane jest podanie adresów IP serwerów systemu zewnętrznego, który uprawniony jest do przesyłania zapytań.

# Zabezpieczenie w warstwie prezentacji

Zabezpieczenie w warstwie prezentacji obejmuje zarówno komunikację typu back-end pomiędzy systemami zewnętrznymi a Systemem e-Płatności jak również komunikację polegającą na zlecaniu płatności poprzez portal Systemu e-Płatności. Zabezpieczenie to realizowane będzie za pomocą protokołu HTTPS.

Na konfigurację protokołu SSL/TLS składają się dwa elementy:

* konfiguracja serwera oraz klienta protokołu (wspierane zestawy szyfrów, zasady walidacji certyfikatu),
* wygenerowanie pary kluczy oraz certyfikatu serwera.

Zalecamy okresowy przegląd poniższych wymagań zgodnie z aktualnym stanem wiedzy dotyczącym bezpieczeństwa protokołu TLS oraz stosowanych zestawów szyfrów.

## Zlecenia konfiguracji serwera oraz klienta SSL/TLS poprzez back-end

Stosowany będzie protokół TLS z uwierzytelnieniem klienta. Systemy zewnętrzne przekażą certyfikaty swoich serwerów do Systemu e-Płatności, gdzie zostaną one zaimportowane przez Administratora do repozytorium zaufanych certyfikatów. Na tej podstawie System e-Płatności będzie uwierzytelniał systemy zewnętrzne.

Administrator Systemu e-Płatności przekaże certyfikat serwera Systemu e-Płatności do systemów zewnętrznych. Na tej podstawie systemy zewnętrzne będą uwierzytelniać przychodzące połączenia z Systemu e-Płatności.

Protokół TLS powinien mieć następujące parametry:

* wersja protokołu TLS 1.2
* aktywny mechanizm odtwarzania poprzedniej sesji (session reuse) lub połączenia trwałego (persistent connection)
* wsparcie dla zestawów algorytmów z zakresu:
  + TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384
  + TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256
  + TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256
  + TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384
  + TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
  + TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256

Podczas komunikacji w systemie serwer Systemu e-Płatności jak i serwery systemów zewnętrznych będą odgrywały rolę zarówno klienta jak i serwera w protokole TLS. Z tego powodu konieczne jest użycie pary certyfikatów z ograniczeniem użycia klucza jako klient albo serwer SSL lub wystawić certyfikat pozwalający odgrywać danemu systemowi rolę zarówno serwera jak i klienta w protokole SSL.

Powyższe zalecenia konfiguracyjne dotyczą środowiska testowego, developerskiego oraz produkcyjnego.

## Zlecenia certyfikacji serwera oraz klienta SSL/TLS poprzez back-end

Połączenia poprzez back-endrealizowane są pomiędzy określonymi punktami systemu. Ponadto między ich właścicielami (Właściciel Systemu i Operatorzy) istnieje możliwość bezpiecznej wymiany certyfikatów.

Dla serwerów i klientów back-end oraz każdej z końcówek musi być wygenerowany autocertyfikat oddzielnie w każdym ze środowisk. Certyfikaty te powinny być przekazane oraz zaimportowane do zaufanych repozytoriów drugiej strony. Systemy winny być skonfigurowane tak, by akceptować wyłącznie te certyfikaty. Niedopuszczalne jest akceptowanie certyfikatów poświadczonych przez wygenerowane autocertyfikaty. Certyfikaty powinny być wygenerowane odrębnie w każdym ze środowisk, aby nie mogły one omyłkowo (np. przez nieprawidłową konfigurację) skomunikować się ze sobą.

Jeśli nazwa domenowa na to nie wskazuje w CN certyfikatu należy użyć dodatkowych opisów TST/DEV. Podczas importu certyfikatów do zaufanego repozytorium obowiązkowo należy się upewnić , że zastosowano e prawidłowe certyfikaty weryfikując ich źródło oraz odcisk klucza z właścicielem drugiej końcówki systemu.

Autocertyfikaty można wygenerować za pomocą oprogramowania openssl stosując konfigurację przygotowaną na podstawie szablonu:

[req]  
prompt = no  
distinguished\_name = dn  
req\_extensions = ext  
x509\_extensions = ext  
default\_md = sha256  
  
[dn]  
ST = mazowieckie  
L = Warszawa  
OU = Departament Informatyzacji i Rejestrów Sądowych  
O = Ministerstwo Sprawiedliwości  
C = PL  
CN = example.com  
  
[ext]  
subjectKeyIdentifier=hash  
authorityKeyIdentifier=keyid:always,issuer:always  
basicConstraints = critical,CA:false  
# dla certyfikatu serwera TLS  
#keyUsage = keyEncipherment, keyAgreement, digitalSignature  
# dla certyfikatu klienta TLS  
#keyUsage = nonRepudiation, digitalSignature, keyEncipherment, keyAgreement  
subjectAltName = DNS:example.com,DNS:\*.example.com

oraz polecenie:

openssl req -new -x509 -utf8 -nameopt multiline,utf8 -config ms.cnf -newkey rsa:2048 -nodes -days 760 -keyout ms.prv -out ms.crt

W konfiguracji należy odkomentować odpowiednie pole keyUsage, wprowadzić adresy domenowe oraz oznaczenie środowiska w CN.

Na serwerach należy zadbać o bezpieczeństwo klucza prywatnego. W przypadku jego kompromitacji należy niezwłocznie:

* zweryfikować siłę kluczy i stosowane algorytmy zgodnie z aktualnymi zaleceniami bezpieczeństwa
* wygenerować nowe klucze i autocertyfikat
* przekazać drugiej stronie nowy autocertyfikat z żądaniem usunięcia poprzedniego autocertyfikatu
* wymienić certyfikat na serwerze/kliencie

Muszą być zastosowane klucze RSA o długości 2048 bit oraz dwuletniy okresu ważności certyfikatów. Na miesiąc przed upłynięciem ważności certyfikatów należy:

* zweryfikować siłę kluczy i stosowane algorytmy zgodnie z aktualnymi zaleceniami bezpieczeństwa
* wygenerować nowe klucze i autocertyfikat
* przekazać drugiej stronie nowy autocertyfikat z żądaniem wprowadzenia go do repozytorium akceptowanych certyfikatów
* wymienić certyfikat na serwerze/kliencie

## Zlecenia konfiguracji serwera oraz klienta SSL/TLS poprzez przeglądarkę klienta

Przy obsłudze zleceń dla Systemu e-Płatności poprzez przesłanie danych za pomocą przeglądarki klienta stosowany będzie protokół TLS bez uwierzytelnienia klienta (przeglądarki użytkownika). System e-Płatności winien posługiwać się certyfikatem wystawionym przez jedno z zaufanych centrów certyfikacji, którego certyfikat wbudowany jest w repozytoria przeglądarek oraz systemy operacyjne. Ze względów bezpieczeństwa nie dotyczy to systemu testowego oraz developerskiego.

Parametry ustawień protokołu SSL serwera Systemu e-Płatności powinny być tak dobrane, aby powszechne przeglądarki były w stanie go obsłużyć. Po stronie serwera zalecana jest następująca konfiguracja:

* wsparcie dla protokołów TLS 1.2, 1.1, 1.0, brak wsparcia dla SSL 2.0, 3.0, preferowany protokół TLS 1.2
* wyłączona kompresja dla TLS 1.0
* aktywny mechanizm odtwarzania poprzedniej sesji (session reuse)
* wyłączone słabe szyfry (RC4, DES), preferowane silniejsze algorytmy (AES, 3DES) oraz tryby (GCM); wsparcie dla algorytmów z zakresu:
  + TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384
  + TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384
  + TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
  + TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
  + TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256
  + TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
  + TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA
  + TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA
  + TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA
  + TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA
  + TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA
  + TLS\_DHE\_DSS\_WITH\_AES\_128\_CBC\_SHA
  + TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA
  + TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA
  + TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA
  + TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

## Zlecenia certyfikacji serwera oraz klienta SSL/TLS poprzez przeglądarkę klienta

Poprzez przeglądarkę klienta klienci protokołu SSL/TLS nie używają certyfikatów do uwierzytelnienia w żadnym ze środowisk.

Dla środowisk testowego i developerskiego zalecamy wygenerowanie autocertyfikatów dla wszystkich portali. Przeglądarki klienta będą reagowały ostrzeżeniem przed nieważnym certyfikatem, ale takie zachowanie jest wskazane ponieważ użytkownik będzie świadomy, że łączy się z testową lub developerską wersją systemu i nie wprowadzi do niego omyłkowo danych produkcyjnych.

Dla środowiska produkcyjnego zalecamy:

* dla portalu zewnętrznego <https://oplaty.ms.gov.pl> wygenerowanie zgłoszenia certyfikacyjnego i złożenie wniosku o certyfikat z rozszerzoną weryfikacją właściciela certyfikatu (Extended Validation, EV). Taka zasada stosowana jest obecnie dla innych portali w domenie [ms.gov.pl](http://ms.gov.pl) (np. <https://ekw.ms.gov.pl/>); należy zwrócić uwagę czy CA wystawiające certyfikat EV posiada niezbędne zaświadczenia do prowadzenia takiej działalności oraz stosuje funkcję skrótu SHA-2 w podpisach wystawianych certyfikatów,
* dla portalu wewnętrznego https://administracja.opl[aty.ms.gov.pl](http://aty.ms.gov.pl) wygenerowanie zgłoszenia certyfikacyjnego i złożenie wniosku o certyfikat z weryfikacją domeny (Domain Validation, DV). Z uwagi na dużą ilość użytkowników utrudnione byłoby stosowanie certyfikatu wygenerowanego przez CA Właściciela Systemu (konieczność importu certyfikatu do przeglądarki); należy zwrócić uwagę czy CA wystawiające certyfikat DV posiada niezbędne zaświadczenia do prowadzenia takiej działalności oraz stosuje funkcję skrótu SHA-2 w podpisach wystawianych certyfikatów.

# Zabezpieczenie w warstwie aplikacji

Zabezpieczenie w warstwie aplikacji obejmuje zarówno komunikację typu back-end pomiędzy systemami zewnętrznymi a Systemem e-Płatności jak również komunikację polegającą na zlecaniu płatności poprzez portal Systemu e-Płatności. Zabezpieczenie to realizowane będzie za pomocą stosowania kodu uwierzytelniającego wiadomość, który będzie chronił integralność przesyłanych danych oraz uwierzytelni ich źródło.

## Konfiguracja

Wykorzystanie kodu uwierzytelniającego wiadomość wiąże się z wymianą pomiędzy nadawcą a odbiorcą wiadomości tajnego klucza. Klucz taki powinien być ustanowiony odrębnie dla każdego z systemów, które mogą komunikować się z Systemem e-Płatności. Każdy z systemów zewnętrznych powinien otrzymać inny klucz. Administrator Systemu e-Płatności wygeneruje i w bezpieczny sposób przekaże systemom zewnętrznym klucz (klucze) służące do wygenerowania kodu uwierzytelniającego wiadomość dla przesyłanych komunikatów. Klucz ma długość równą wielokrotności 8 bitów i wynoszącą co najmniej 256 bitów. Przekazywany jest w formie heksadecymalnej (znaki z zakresu 0-9, a-f), przy wejściu do algorytmu używany jest w reprezentacji binarnej. Przekazany klucz ma identyfikator alfanumeryczny.

Przykład klucza:

KLUCZ1=51546eb53e8439f156acd2a7b7301cadec13d0ff85f46ff0cc97005ae16776b7

Ustanowione klucze powinny być zmieniane w przypadku ich kompromitacji oraz okresowo ze względów bezpieczeństwa. Zaleca się aby klucz był wymieniany co 12 miesięcy. Procedura wymiany klucza polega na wygenerowaniu nowego klucza przez Administratora Systemu e-Płatności i przekazaniu go do systemu zewnętrznego. W okresie przejściowym akceptowane będą żądania z kodem wygenerowanym za pomocą starego i nowego klucza. Po okresie przejściowym Administrator usuwa z konfiguracji Systemu e-Płatności stary klucz wraz z identyfikatorem.

## Zlecenia poprzez back-end

Zlecenia poprzez back-end przekazywane są za pomocą żądań HTTP. W zależności od typu żądania (GET, POST) może on zawierać argumenty przetwarzania oraz zawartość, najczęściej w formacie JSON.

### Zabezpieczenie żądań

Algorytm zabezpieczenia żądań HTTP jest następujący:

|  |  |  |
| --- | --- | --- |
| Krok | Opis kroku | Przykłady zapytań |
| 0 | Bazowy komunikat | |  | | --- | | POST https://www.system-zewnetrzny.pl/payment HTTP/1.1 Host: www.system-zewnetrzny.pl Content-type: application/json; charset=utf-8 Content-Length: 708 Date: Mon, 20 Oct 2014 12:00:00 GMT  {  "partnerId": "EPLATNOSCIID",  "orderId": "EP56958546",  "paymentMethod": "VISA",  "totalAmount": "350",  "commission": "1,50",  "currencyCode": "PLN",  "languageCode": "pl",  "paymentDetails": [   {  "id": "48435456"  "merchantPosId": "EPL84656",  "amount": "350",  "transferLabel": "Opłata za sprawę PO.VII\_CPOI-9302-2938-9393-0",  "description": "Dotyczy: Jan Kowalski"  }  ],  "confirmationUrl": "https://www.system-zlecający-sprzedaż.pl/confirmation",  "cancellationUrl": "https://www.system-zlecający-sprzedaż.pl/cancellation" } |  |  | | --- | | GET https://www.system-zewnetrzny.pl/payment/types HTTP/1.1 Host: www.system-zewnetrzny.pl Date: Mon, 20 Oct 2014 12:00:00 GMT | |
| 1 | Komunikat POST uzupełniany jest o nagłówek ep-content-sha256, którego wartość wyliczana jest analogicznie jak Content-MD5 (RFC 1864), z tą różnicą, że zamiast funkcji skrótu MD5 stosowana jest funkcja SHA-256 oraz zamiast kodowania Base64 wynik jest zapisywany w formie heksadecymalnej z użyciem znaków 0-9, a-f.  W szczególności jako wejście do funkcji skrótu powinna być przekazana zawartość żądania w formie kanonicznej opisanej w Content-Type wiadomości (a przed modyfikacją wg. Content-Transfer-Encoding).  Dla komunikatu GET ten krok jest pominięty. | |  | | --- | | POST https://www.system-zewnetrzny.pl/payment HTTP/1.1 Host: www.system-zewnetrzny.pl Content-type: application/json; charset=utf-8 Content-Length: 708 Date: Mon, 20 Oct 2014 12:00:00 GMT ep-content-sha256: 5f5c989f71a5b68d1b99662089a05221ddc50c01182f3c6d53d2715a096436cf  {  "partnerId": "EPLATNOSCIID",  "orderId": "EP56958546",  "paymentMethod": "VISA",  "totalAmount": "350",  "commission": "1,50",  "currencyCode": "PLN",  "languageCode": "pl",  "paymentDetails": [   {  "id": "48435456"  "merchantPosId": "EPL84656",  "amount": "350",  "transferLabel": "Opłata za sprawę PO.VII\_CPOI-9302-2938-9393-0",  "description": "Dotyczy: Jan Kowalski"  }  ],  "confirmationUrl": "https://www.system-zlecający-sprzedaż.pl/confirmation",  "cancellationUrl": "https://www.system-zlecający-sprzedaż.pl/cancellation" } |  |  | | --- | | GET https://www.system-zewnetrzny.pl/payment/types HTTP/1.1 Host: www.system-zewnetrzny.pl Date: Mon, 20 Oct 2014 12:00:00 GMT | |
| 2 | Dokonywana jest konkatenacja wybranych wartości z nagłówka żądania, które utworzą ciąg będący wejściem dla funkcji HMAC SHA-256. Wszystkie wartości kodowane są wg. formy kanonicznej stosowanej dla URI. Kolejne wartości to:   * identyfikator żądania (POST, GET, ...) * identyfikator zasobu * argumenty przetwarzania * wykaz podpisywanych nagłówków żądania (identyfikatory skonwertowane do małych liter, posortowane wg. nazwy identyfikatora, w formacie *identyfikator:wartość*, wartość bez wiodących i kończących białych znaków, oddzielone znakiem "\n") + "\n" * wykaz identyfikatorów podpisywanych nagłówków żądania (posortowane, skonwertowane do małych liter, oddzielone ";") + "\n"   Wymagane nagłówki wchodzące w skład danych do podpisu dla żądań pomiędzy Systemem e-Płatności i systemem zewnętrznym to:   * dla komunikatu POST: Host, Date, Content-Type oraz ep-content-sha256, * dla komunikatu GET: Host, Date. | W przykładzie oznaczono niewidoczny znak "\n".   |  | | --- | | POST\n /payment\n \n content-type:application/json; charset=utf-8\n date:mon, 20 oct 2014 12:00:00 gmt\n ep-content-sha256:5f5c989f71a5b68d1b99662089a05221ddc50c01182f3c6d53d2715a096436cf\n host:www.system-zewnetrzny.pl\n content-type;date;ep-content-sha256;host\n |  |  | | --- | | GET\n /payment/types\n \n date:mon, 20 oct 2014 12:00:00 gmt\n host:www.system-zewnetrzny.pl\n date;host\n | |
| 3 | Dla otrzymanego ciągu wyliczana jest wartość HMAC SHA-256 oraz przygotowywany nagłówek Authorization w formie:   * identyfikator algorytmu (EP-HMAC-SHA256) * identyfikator użytego klucza (Credential=id-klucza) * lista podpisanych nagłówków oddzielonych znakiem ";" (SignedHeaders=headers) * podpis w formie heksadecymalnej, małe litery (Signature=signature) | |  | | --- | | Authorization: EP-HMAC-SHA256 Credential=KLUCZ1,SignedHeaders=content-type;date;ep-content-sha256;host,Signature=97cac92d8edb5f270f2571266d01f883d33f96f07f86e123a55f31e7a444bc20 |  |  | | --- | | Authorization: EP-HMAC-SHA256 Credential=KLUCZ1,SignedHeaders=date;host,Signature=e90c10ea7ec4fb77d38ce5490aac2e35c12525d5d7524238310aa7dfc3d5030e | |
| 4 | Żądanie przesyłane jest do serwera | |  | | --- | | POST https://www.system-zewnetrzny.pl/refundStatus HTTP/1.1 Host: www.system-zewnetrzny.pl Content-type: application/json; charset=utf-8 Content-Length: 708 Date: Mon, 20 Oct 2014 12:00:00 GMT  Authorization: EP-HMAC-SHA256 Credential=KLUCZ-A,SignedHeaders=content-type;date;ep-content-sha256;host,Signature=97cac92d8edb5f270f2571266d01f883d33f96f07f86e123a55f31e7a444bc20 ep-content-sha256: 5f5c989f71a5b68d1b99662089a05221ddc50c01182f3c6d53d2715a096436cf  {  "partnerId": "EPLATNOSCIID",  "orderId": "EP56958546",  "paymentMethod": "VISA",  "totalAmount": "350",  "commission": "1,50",  "currencyCode": "PLN",  "languageCode": "pl",  "paymentDetails": [   {  "id": "48435456"  "merchantPosId": "EPL84656",  "amount": "350",  "transferLabel": "Opłata za sprawę PO.VII\_CPOI-9302-2938-9393-0",  "description": "Dotyczy: Jan Kowalski"  }  ],  "confirmationUrl": "https://www.system-zlecający-sprzedaż.pl/confirmation",  "cancellationUrl": "https://www.system-zlecający-sprzedaż.pl/cancellation" } |  |  | | --- | | GET https://www.system-zewnetrzny.pl/payment/types HTTP/1.1 Host: www.system-zewnetrzny.pl Date: Mon, 20 Oct 2014 12:00:00 GMT  Authorization: EP-HMAC-SHA256 Credential=KLUCZ1,SignedHeaders=date;host,Signature=e90c10ea7ec4fb77d38ce5490aac2e35c12525d5d7524238310aa7dfc3d5030e | |

### Zabezpieczenie odpowiedzi

Algorytm zabezpieczania odpowiedzi HTTP jest następujący:

|  |  |  |
| --- | --- | --- |
| Krok | Opis kroku | Przykład odpowiedzi |
| 0 | Bazowy komunikat.  Jeżeli żądanie HTTP miało nieprawidłowy podpis zwracamy odpowiedź 401 (Unauthorized) bez podpisu. | |  | | --- | | HTTP/1.1 200 OK Content-type: application/json; charset=utf-8 Content-Length: 67 Date: Mon, 20 Oct 2014 12:00:00 GMT   {  "response": "ACCEPT" } |  |  | | --- | | HTTP/1.1 501 Method Not Implemented Date: Mon, 20 Oct 2014 12:00:00 GMT | |
| 1 | Jeżeli komunikat zawiera dane to uzupełniany jest o nagłówek ep-content-sha256, którego wartość wyliczana jest analogicznie jak Content-MD5 (RFC 1864), z tą różnicą, że zamiast funkcji skrótu MD5 stosowana jest funkcja SHA-256 oraz zamiast kodowania Base64 wynik jest zapisywany w formie heksadecymalnej z użyciem znaków 0-9, a-f.  W szczególności jako wejście do funkcji skrótu powinna być przekazana zawartość żądania w formie kanonicznej opisanej w Content-Type wiadomości (a przed modyfikacją wg. Content-Transfer-Encoding).  Jeżeli komunikat nie zawiera danych krok ten jest pomijany. | |  | | --- | | HTTP/1.1 200 OK Content-type: application/json; charset=utf-8 Content-Length: 67 Date: Mon, 20 Oct 2014 12:00:00 GMT  ep-content-sha256: 8e08234df9f1f3aab71b2c898d92236f7bab51319a20f03d87569a2a1f7c3c20  {  "response": "ACCEPT" } |  |  | | --- | | HTTP/1.1 501 Method Not Implemented Date: Mon, 20 Oct 2014 12:00:00 GMT | |
| 2 | Dokonywana jest konkatenacja wybranych wartości z nagłówka odpowiedzi, które utworzą ciąg będący wejściem dla funkcji HMAC SHA-256. Wszystkie wartości kodowane są wg. formy kanonicznej stosowanej dla URI. Kolejne wartości to:   * kod odpowiedzi (200, 403, ...) * wykaz podpisywanych nagłówków żądania (identyfikatory skonwertowane do małych liter, posortowane wg. nazwy identyfikatora, w formacie [*identyfikator:wartość*](http://identyfikatorwartość), wartość bez wiodących i kończących białych znaków, oddzielone znakiem "\n") + "\n" * wykaz identyfikatorów podpisywanych nagłówków żądania (posortowane, skonwertowane do małych liter, oddzielone ";") + "\n"   Wymagane nagłówki wchodzące w skład danych do podpisu dla odpowiedzi pomiędzy Systemem e-Płatności i systemem zewnętrznym to:   * jeżeli komunikat posiada dane: Content-Type, Date oraz ep-content-sha256, * jeżeli komunikat nie posiada danych: Date. | W przykładzie oznaczono niewidoczny znak "\n".   |  | | --- | | 200\n content-type:application/json; charset=utf-8\n date:mon, 20 oct 2014 12:00:00 gmt\n ep-content-sha256:8e08234df9f1f3aab71b2c898d92236f7bab51319a20f03d87569a2a1f7c3c20\n content-type;date;ep-content-sha256\n |  |  | | --- | | 501\n date:mon, 20 oct 2014 12:00:00 gmt\n date\n | |
| 3 | Dla otrzymanego ciągu wyliczana jest wartość HMAC SHA-256 oraz przygotowywany nagłówek Authorization w formie:   * identyfikator algorytmu (EP-HMAC-SHA256) * identyfikator użytego klucza (Credential=id-klucza) * lista podpisanych nagłówków oddzielonych znakiem ";" (SignedHeaders=headers) * podpis w formie heksadecymalnej, małe litery (Signature=signature) | |  | | --- | | Authorization: EP-HMAC-SHA256 Credential=KLUCZ1,SignedHeaders=content-type;date;ep-content-sha256;Signature=e98f19433a76af57bd53bf14e46713c6d64276c92b7554c52176be22af3678cd |  |  | | --- | | Authorization: EP-HMAC-SHA256 Credential=KLUCZ1,SignedHeaders=date;Signature=c08532dec07ae882225160f8636f5eea8663e9784b59eddb680a7ddd572c6c7b | |
| 4 | Odpowiedź przesyłana jest do klienta | |  | | --- | | HTTP/1.1 200 OK Content-type: application/json; charset=utf-8 Content-Length: 67 Date: Mon, 20 Oct 2014 12:00:00 GMT ep-content-sha256: 8e08234df9f1f3aab71b2c898d92236f7bab51319a20f03d87569a2a1f7c3c20 Authorization: EP-HMAC-SHA256 Credential=KLUCZ1,SignedHeaders=content-type;date;ep-content-sha256;Signature=e98f19433a76af57bd53bf14e46713c6d64276c92b7554c52176be22af3678cd  {  "response": "ACCEPT" } |  |  | | --- | | HTTP/1.1 501 Method Not Implemented Date: Mon, 20 Oct 2014 12:00:00 GMT  Authorization: EP-HMAC-SHA256 Credential=KLUCZ1,SignedHeaders=date;Signature=c08532dec07ae882225160f8636f5eea8663e9784b59eddb680a7ddd572c6c7b | |

## Zlecenia poprzez przeglądarkę klienta

Podczas przesyłania wywołań do Portalu Systemu e-Płatności nie ma możliwości edycji nagłówków wysyłanych żądań HTTP. W takim przypadku zabezpieczane są jedynie pola formularza. Operacja wykonywana jest przez system, który przygotuje zlecenie. Zabronione jest wyliczanie kodu uwierzytelniającego wiadomość przez przeglądarkę klienta ponieważ wiązałoby się z ujawnieniem klucza.

Do przesłania wartości kodu HMAC SHA-256 wykorzystywane jest pole formularza Authorization. W parametrze tym przesyłany jest identyfikator klucza oraz wartość funkcji HMAC SHA-256 z parametrów komunikatu.

Wartość parametru Authorization wylicza się według następującego algorytmu:

|  |  |  |
| --- | --- | --- |
| Krok | Opis kroku | Przykład dla zlecenia przy pomocy formularza wykonując operację http POST |
| 0 | Bazowy formularz | |  | | --- | | <form action= "https://secure.eplatnosci.ms.gov.pl/payment" method= "post" class= "form"><input type="hidden" name= "systemName" value= "S24-485432"/><input type="hidden" name= "serviceName" value= "SPOLKA-435268"/><input type="hidden" name= "paymentReference" value= "84354132468"/><input type="hidden" name= "paymentDescription" value= "JAN KOWALSKI"/><input type="hidden" name= "paymentTransferLabel" value= "OPŁATA ZA 84354132468"/><input type="hidden" name= "amount" value= "600"/><input type="hidden" name= "currencyCode" value= "PLN"/><input type="hidden" name= "languageCode" value= "pl"/><input type="hidden" name= "confirmationUrl" value= "http://system-merytoryczny.pl/confirmation"/><input type="hidden" name= "cancellationUrl" value= "http://system-merytoryczny.pl/cancellation"/> </form> | |
| 1 | Wszystkie parametry formularza sortowane są alfabetycznie według nazwy parametru (wartości atrybutu name), w porządku rosnącym. Używane jest kodowanie kanoniczne parametrów (jak przy nagłówku żądania *application/x-www-form-urlencoded*) | |  | | --- | | amount=600  cancellationUrl=http%3A%2F%2Fsystem-merytoryczny.pl%2Fcancellation  confirmationUrl=http%3A%2F%2Fsystem-merytoryczny.pl%2Fconfirmation  currencyCode=PLN  languageCode=pl  serviceName=SPOLKA-435268  systemName=S24-485432  paymentReference=84354132468  paymentDescription=JAN+KOWALSKI  paymentTransferLabel=OP%C5%81ATA+ZA+84354132468 | |
| 2 | Dokonywana jest konkatenacja wartości wszystkich pól formularza według wzoru: *nazwa-parametru=wartość-parametru*. Znacznikiem oddzielającym pola jest znak &.  Sposób postępowania jest analogiczny jak dla tworzenia zawartości żądania HTTP przesyłanego do serwera na podstawie formularza. | Aby zapewnić czytelność przykładu wstawiono dodatkowe znaki nowej linii.   |  | | --- | | amount=600&  cancellationUrl=http%3A%2F%2Fsystem-merytoryczny.pl%2Fcancellation&  confirmationUrl=http%3A%2F%2Fsystem-merytoryczny.pl%2Fconfirmation&  currencyCode=PLN&  languageCode=pl&  serviceName=SPOLKA-435268&  systemName=S24-485432&  paymentReference=84354132468&  paymentDescription=JAN+KOWALSKI&  paymentTransferLabel=OP%C5%81ATA+ZA+84354132468 | |
| 3 | Dla otrzymanego ciągu wyliczana jest wartość HMAC SHA-256 oraz przygotowywane jest pole Authorization w formie:   * identyfikator użytego klucza * znak spacji * podpis w formie heksadecymalnej, małe litery | |  | | --- | | <input type="hidden" name= "Authorization" value= "KLUCZ\_A 51546eb53e8439f156acd2a7b7301cadec13d0ff85f46ff0cc97005ae16776b7"/> | |
| 4 | Pole dołączane jest do formularza | |  | | --- | | <form action= "https://secure.eplatnosci.ms.gov.pl/payment" method= "post" class= "form"><input type="hidden" name= "systemName" value= "S24-485432"/><input type="hidden" name= "serviceName" value= "SPOLKA-435268"/><input type="hidden" name= "paymentReference" value= "84354132468"/><input type="hidden" name= "paymentDescription" value= "JAN KOWALSKI"/><input type="hidden" name= "paymentTransferLabel" value= "OPŁATA ZA 84354132468"/><input type="hidden" name= "amount" value= "600"/><input type="hidden" name= "currencyCode" value= "PLN"/><input type="hidden" name= "languageCode" value= "pl"/><input type="hidden" name= "confirmationUrl" value= "http://system-merytoryczny.pl/confirmation"/><input type="hidden" name= "cancellationUrl" value= "http://system-merytoryczny.pl/cancellation"/><input type="hidden" name= "Authorization" value= "KLUCZ\_A 51546eb53e8439f156acd2a7b7301cadec13d0ff85f46ff0cc97005ae16776b7"/></form> | |