GESTIONE AUTENTICAZIONE APPLICATIVA CON CERTIFICATI DIGITALI

Manuale operativo per autenticazione tramite certificati X509 per le funzionalità Application To Application (A2A)

Gestione autenticazione applicativa (A2A)

Gestione autenticazione con certificati X509 su canale TLS

Gestione autenticazione applicativa (A2A)

Versione 0.2 del 03/01/2018

Storia del documento

VERSIONE	Data	Descrizione
0.1	29/12/2017	Prima bozza
0.2	03/01/2018	Documento completato, bozza pronta per la revisione del *system owner.*

Sommario

Contesto

Il presente manuale ha lo scopo di fornire all'utente una serie di procedure standard per verificare la corretta configurazione delle credenziali di autenticazione applicativa tramite certificati digitali X509 ai fini del colloquio con la piattaforma SIOPE+.

Verranno trattati i criteri relativi al solo certificato di autenticazione: sebbene l'interfaccia "Gestione delle credenziali applicative" preveda la possibilità di utilizzare anche il certificato di cifratura, quest'ultimo non è necessario ai fini del colloquio con SIOPE+ e risulta quindi facoltativo.

Tali credenziali sono necessarie per autenticarsi verso i sistemi informatici che erogano i servizi applicativi esposti sulla rete Internet.

Al termine del documento sono riportate le risposte alle domande poste più frequentemente al servizio di supporto (FAQ).

Si precisa che le procedure descritte si affiancano ma non sostituiscono gli standard internazionali ai quali si farà sempre riferimento (RFC5246, RFC5280, RFC6810).

Nella trattazione si farà riferimento al alcuni *software open source:*

- **openssl,** disponibile su tutti i sistemi operativi Linux/Unix. Per il sistema operativo MS Windows è possibile scaricare il software dal sito https://wiki.openssl.org/index.php/Binaries;
- **curl**, disponibile su tutti i sistemi operativi Linux/Unix. Per il sistema operativo MS Windows è possibile scaricare il software dal sito https://curl.haxx.se/download.html#Win32 .

Per il supporto sui software open source indicati, si rimanda alle fonti liberamente consultabili sulla rete Internet.

I certificati ritenuti validi devono presentare le caratteristiche previste nel documento "Manuale per la registrazione e autenticazione a SIOPE+". In particolare il certificato di autenticazione deve essere rilasciato da un'Autorità di Certificazione da individuarsi tra quelle disponibili nel bundle Mozilla (https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/). Il certificato (usato per autenticazione client SSL) deve avere l'attributo "X509v3 Extended Key Usage: TLS Web Client Authentication". Il certificato di firma digitale generalmente non rispetta questi requisiti.

Estrazione del certificato pubblico

I certificati acquistati sono, di norma, forniti dalle *Certification Authority* in formato pkcs12 (estensione .p12 o .pfx); in tale caso non sono direttamente utilizzabili nella sezione "Gestione delle credenziali applicative" del sito di Registrazione alla piattaforma SIOPE+.

Nel seguito è indicata una delle possibili procedure per estrarre il certificato pubblico dalla busta PKCS12. Il *file* ottenuto potrà essere successivamente associato alla credenziale applicativa. La procedura utilizzerà il *software open source* **openssl** che è disponibile su tutti i sistemi operativi Linux. Per il sistema operativo MS Windows è possibile scaricare il software dal sito https://wiki.openssl.org/index.php/Binaries .

Nell'esempio si genererà il certificato pubblico in un file denominato "CERTIFICATO_AUTENTICAZIONE_CLIENT_PEM.cer" a partire dal file pkcs12 denominato "CERTIFICATO_AUTENTICAZIONE.p12", presente nella cartella in cui viene lanciato il comando `openssl`.

```
[utente@sviluppo~]$ openssl pkcs12 -in CERTIFICATO_AUTENTICAZIONE.p12 -out
CERTIFICATO_AUTENTICAZIONE_CLIENT_PEM.cer -clcerts -nokeys

Enter Import Password:

MAC verified OK
```

Inserire la password del certificato pkcs12

E’ possibile verificare la corretta estrazione del certificato, visualizzando il contenuto del file “CERTIFICATO_AUTENTICAZIONE_CLIENT_PEM.cer”, ad esempio utilizzando il comando `cat`:

```
[utente@sviluppo ~]$ cat CERTIFICATO_AUTENTICAZIONE_CLIENT_PEM.cer
-----BEGIN CERTIFICATE-----
MIIEJzCCAw+gAwIBAgIBSDANBgkqhkiG9w0BAQsFADAtMSswKQYDVQQDEyJPcmFj
bGUgQWNjZXNzIE1hbmFnZXIgVEVTVCBPTkxZIENBMB4XDTE3MDUwMzE0NDMwMFoX
DTIwMDEwNzEwMjAwMFowggE3MQswCQYDVQQGEwJJVDEPMA0GA1UECBMGSXRhbGlh
MSIwIAYDVQQKExlCYW5jYSBkSXRhbGlhLzAwOTUwNTAxMDA3MTIwMAYDVQQLEylT
ZXJ2aXppIGRpIGNlcnRpZmljYXppb25lIC0gd2ViZmFybSBwcm94eTFHMEUGA1UE
AxM+VkxQTVJDNzlDMTlINTAxUC8xMTExMTExMTExMTExMTEyLmJiYmJiYmJiYmJi
.......................... <omissis> ..........................
RjBEMB0GA1UdDgQWBBSJ55X18FaV/J8uE8ZOXM9e2ZteCTAOBgNVHQ8BAf8EBAMC
ArQwEwYDVR0lBAwwCgYIKwYBBQUHAwIwDQYJKoZIhvcNAQELBQADggEBAJFUltCs
ClYzr+dsFg37s9ezHOCOsDJ663Yb2V+1opc6trnRiso6g22WHwKyBnP+3asXrORJ
3iIt9MqcZYf+vgnJXV8IZS+LhJSG6VNCliRqyi000ifi3A+It2omlgDAKzyUwIji
cZP/WEAkqivL82mDYKQNNtvLkJCRnE0gFWzP6WDdAWGcWjVLlN1kDBRA9VK8XTe0
HzMaN4ITAfH0YhPniqPdgdtmSZ+0s8/xqH/xCo9GXHEV3Plhn0hg/xYGo9SLrkgP
49O9qVL5bJLFeoLPLNwOoEVHZpsxtj+74K/xpBn3uS2mB7kauZ3PRlfh3OSS35Qq
1tyeMZNugdunXjs=
-----END CERTIFICATE-----
```

Certificato di autenticazione

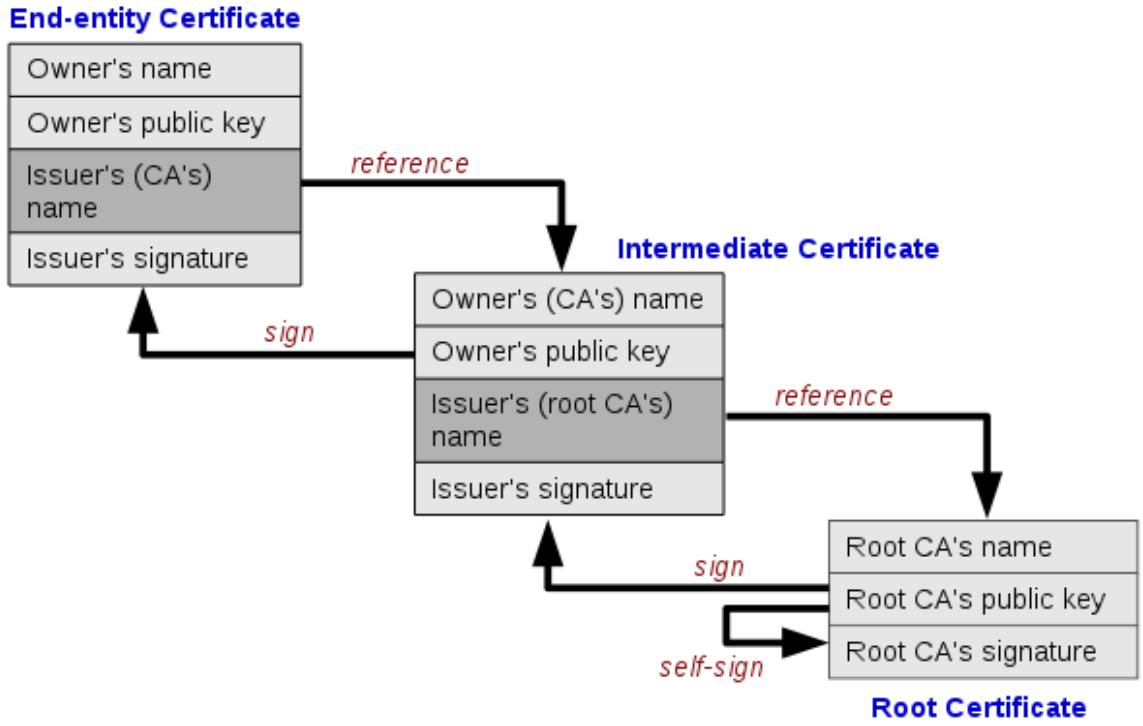
Il certificato client **CERTIFICATO_AUTENTICAZIONE_CLIENT_PEM.cer**, **appena estratto DEVE quindi essere caricato come CERTIFICATO di AUTENTICAZIONE** nella sezione “Gestione delle credenziali applicative” del sito di Registrazione alla piattaforma SIOPE+.

Verifica della connessione verso l'ambiente di collaudo SIOPE+

Il certificato pubblico di autenticazione caricato permette di riconoscere il sistema esterno (di ente, tesoriere o tramite) e autenticarlo al colloquio con la piattaforma SIOPE+. Ogni certificato deve essere associato a una credenziale A2A (id=A2A-<nnnnnnnn>) tramite l'interfaccia WEB di gestione delle credenziali applicative.

Per verificare che il caricamento del certificato pubblico nella sezione "Gestione delle credenziali applicative" del sito di Registrazione alla piattaforma SIOPE+ sia andato a buon fine, è possibile invocare le API REST della piattaforma SIOPE+ attraverso l'uso del *software* **curl**.

Nell'esempio, **come azione propedeutica**, si procederà alla generazione del **file necessario per l'autenticazione** denominato "CERTIFICATO+CHAIN+KEY_PEM.cer" a partire dal file pkcs12 denominato "CERTIFICATO_AUTENTICAZIONE.p12", presente nella cartella in cui viene lanciato il comando `openssl`. Il motivo di tale operazione deriva dalla necessità di esportare in un unico file nel formato PEM la chiave privata, il certificato e tutti i certificati delle CA intermedie (meccanismo noto come catena di fiducia **chain of trust**):



Si procede quindi con il seguente comando:

```
[utente@sviluppo ~]$ openssl pkcs12 -in CERTIFICATO_AUTENTICAZIONE.p12 -out
CERTIFICATO+CHAIN+KEY_PEM.cer

Enter Import Password:

MAC verified OK

Enter PEM pass phrase:

Verifying - Enter PEM pass phrase:
```

Inserire la password del certificato pkcs12

Inserire e ripetere la password che sarà necessaria per l'utilizzo del nuovo certificato privato che stiamo estraendo

E’ possibile verificare la corretta estrazione del certificato, visualizzando il contenuto del file “CERTIFICATO+CHAIN+KEY_PEM.cer”, ad esempio utilizzando il comando `cat`:

Possiamo ora lanciare il comando `curl` e verificare il corretto funzionamento:

```
curl --cert ./CERTIFICATO+CHAIN+KEY_PEM.cer:<password> -k -H
"Accept:application/json;charset=UTF-8" -v https://certa2a.siopeplus.it/v1/<codice
A2A>/PA/<codice UNI UO>/flusso/ack/?download=false

*   Trying xxx.xxx.xxx.xxx...

* TCP_NODELAY set

* Connected to certa2a.siopeplus.it (xxx.xxx.xxx.xxx) port 443 (#0)

* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH

* error setting certificate verify locations, continuing anyway:

*   CAfile: /etc/pki/tls/certs/ca-bundle.crt

  CApath: none

* TLSv1.2 (OUT), TLS handshake, Client hello (1):

* TLSv1.2 (IN), TLS handshake, Server hello (2):

* TLSv1.2 (IN), TLS handshake, Certificate (11):

* TLSv1.2 (IN), TLS handshake, Server key exchange (12):

* TLSv1.2 (IN), TLS handshake, Request CERT (13):

* TLSv1.2 (IN), TLS handshake, Server finished (14):

* TLSv1.2 (OUT), TLS handshake, Certificate (11):

* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):

* TLSv1.2 (OUT), TLS handshake, CERT verify (15):

* TLSv1.2 (OUT), TLS change cipher, Client hello (1):

* TLSv1.2 (OUT), TLS handshake, Finished (20):

* TLSv1.2 (IN), TLS change cipher, Client hello (1):

* TLSv1.2 (IN), TLS handshake, Finished (20):

* SSL connection using TLSv1.2 / DHE-RSA-AES256-GCM-SHA384

* Server certificate:

*  subject: C=IT; ST=Roma; L=Frascati; O=Banca d'Italia; OU=Servizi di certificazione
dei sistemi informatici; CN=certa2a.siopeplus.it

*  start date: Jun 22 08:25:02 2017 GMT

*  expire date: Jun 22 08:25:02 2020 GMT

*  issuer: C=IT; ST=Milano; L=Milano; O=Actalis S.p.A./03358520967; CN=Actalis
Authentication CA G3

*  SSL certificate verify result: unable to get local issuer certificate (20),
continuing anyway.

> GET v1/<codice A2A>/PA/<codice UNI UO>/flusso/ack/?download=false HTTP/1.1

> Host: certa2a.siopeplus.it

> User-Agent: curl/7.50.3

> Accept:application/json;charset=UTF-8
```

Scambio dei certificati di autenticazione avvenuto correttamente

```
>
< HTTP/1.1 200 OK
< Date: Fri, 29 Dec 2017 10:31:06 GMT
< Server: Apache
< Set-Cookie: siopeplus-a2a=bfec27e6.56178214b33cc; path=/; domain=.siopeplus.it
< X-Frame-Options: SAMEORIGIN
< Access-Control-Allow-Origin: *
< Content-Type: application/json;charset=UTF-8
< Transfer-Encoding: chunked
<
* Curl_http_done: called premature == 0
* Connection #0 to host certa2a.siopeplus.it left intact
{"numRisultati":0,"numPagine":1,"risultatiPerPagina":20,"pagina":1,"risultati":[]}
```

Chiamata effettuata correttamente

Certificati di autenticazione con più di una CA intermedia

Nell'esempio seguente ipotizziamo che il certificato di autenticazione identificato come "Cert+Chain+Key.pem", sia firmato da una SUB_CA (#1) a sua volta firmata da una seconda SUB_CA (#2) a sua volta firmata da una Certification Authority segnalata nel CA_Bundle della Mozilla Foundation (https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/).

Per la generazione del file, come visto prima, ci si può affidare al software openssl

```
openssl pkcs12 –in certificato_autenticazione.p12 –nodes  -out Cert+Chain+Key.pem
```

Con il file ottenuto è possibile procedere con i test di autenticazione:

```
$ view Cert+Chain+Key.pem
#chiave privata non cifrata
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEApaD2k7vcETcwiKyXGcQCy82huspsigZfFt26WYwiX0c
        /omissis
EamAl1XtIzgVwoCStwD/kRb8DFyPov1/aySKP3bRNkd77JtX/gwF
-----END RSA PRIVATE KEY-----
# certificato X509 associato alla credenziale A2A-<nnnnnnnn> , ISSUER=SUB_CA#1 (Ca intermedia)
# Solo questo deve essere caricato sull'interfaccia "Gestione delle credenziali applicative"
-----BEGIN CERTIFICATE-----
MIIFPDCCBCSgAwIBAgIBEzANBgkqhkiG9w0BAQUFADAtMSswKQYDVQQDEyJP/omissis
        /omissis
NAc/UCAZ9KG4vkg05Ot4VVCh8ihPl04RrgcXtmXv/m1UMnOu8VW9KKVL1rLd
-----END CERTIFICATE-----
#Primo componente della CHAIN: certificato x509 della CA intermedia SUB_CA#1, ISSUER= SUB_CA#2
-----BEGIN CERTIFICATE-----
MIIFPDCCBCSgAwIBAgIBEzANBgkqhkiG9w0BAQUFADAtMSswKQYDVQQ/
        /omissis
NAc/UCAZ9KG4vkg05Ot4VVCh8ihPl04RrgcXtmXv/m1UMnOu8VW9KKV
-----END CERTIFICATE-----
#Secondo componente della CHAIN: certificato x509 della CA intemedia SUB_CA#2, ISSUER= CA_ROOT presente su CA_bundle Mozilla
-----BEGIN CERTIFICATE-----
MIIIPjCCBiagAwIBAgIIRIQedGGfUsswDQYJKoZIhvcNAQELBQAwgawxCzAJBgNV
        /Omissis
x/Hce43AdsgI6UTcF4p0B+7Ma2OxsOJPuPh0HOK1N/+ygBGASW+sNhyI/LpueznmonY=
-----END CERTIFICATE-----
#chiave privata non cifrata
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEApaD2k7vcETcwiKyXGcQCy82huspsigZfFt26WYwiX0c
        /omissis
EamAl1XtIzgVwoCStwD/kRb8DFyPov1/aySKP3bRNkd77JtX/gwF
-----END RSA PRIVATE KEY-----
```

Esempio di autenticazione avvenuta con successo:

```
$ curl -kI -H "Accept:application/json;charset=UTF-8" "https://certa2a.siopeplus.it/v1/" --cert
Key+Cert+Chain.pem
HTTP/1.1 404 Not Found
Date: Fri, 29 Dec 2017 15:05:04 GMT
Server: Apache
Set-Cookie: siopeplus-a2a=5725dd36.5617bf514abbd; path=/; domain=.siopeplus.it
X-Frame-Options: SAMEORIGIN
Access-Control-Allow-Origin: *
Content-Type: text/html;charset=UTF-8
Content-Length: 68
Set-Cookie: FDX=1gkv7315a4vlfy7e0e1mn5yil;Path=/;HttpOnly;Secure
```

Esempio di autenticazione con CERTIFICATO emesso da Certification Autority non di fiducia sulla lista di Mozilla Foundation :

```
$ curl -Ik “https://certa2a.siopeplus.it/v1/” --cert WRONG_CERT.pem
curl: (56) SSL read: error:14094418:SSL routines:ssl3_read_bytes:tlsv1 alert unknown ca, errno 0
```

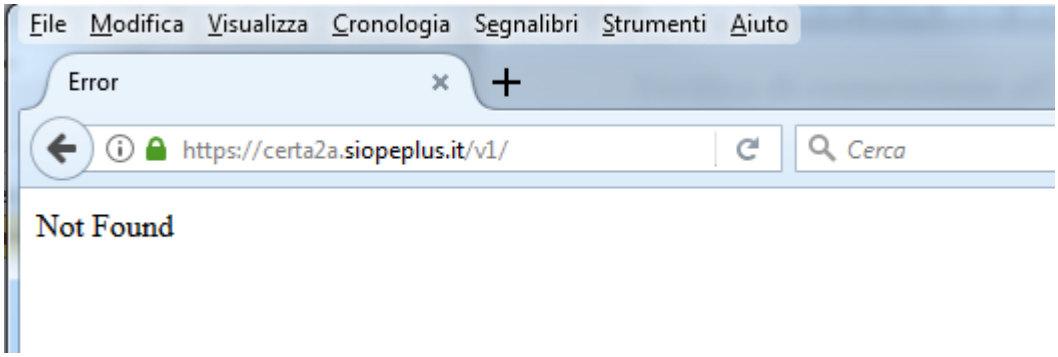
Utilizzo del Browser

Nell'impossibilità tecnica dell'utilizzo degli strumenti software CURL e OPENSSL è possibile importare il certificato di autenticazione nel formato P12/PFX sul browser (preferibilmente Firefox): se l'importazione è priva di errori allora il certificato può, di norma, essere utilizzato per l'autenticazione; al termine dell'importazione è possibile tentare l'autenticazione con il browser componendo la URL del servizio applicativo.

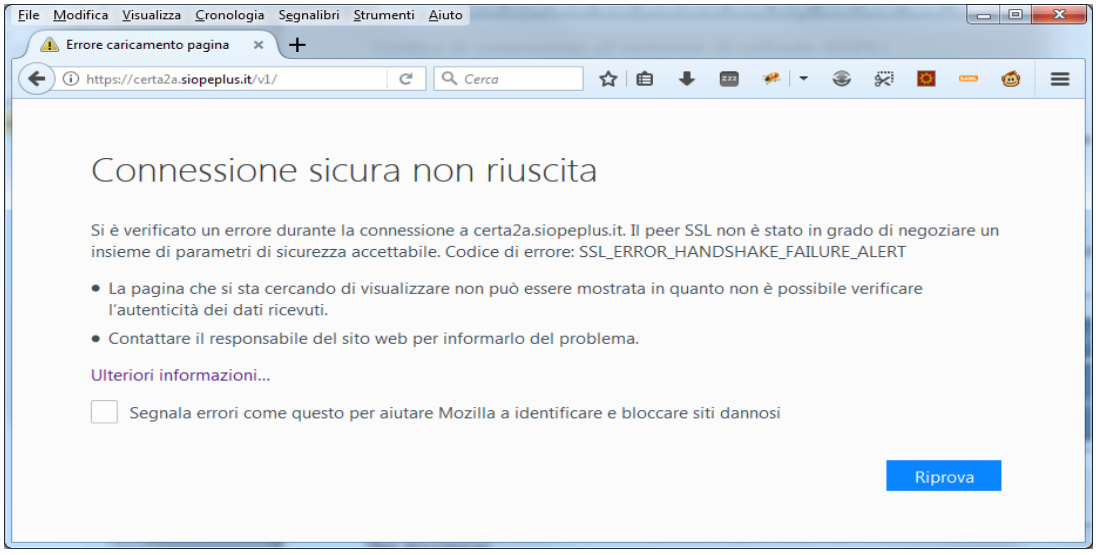
NOTA: è necessario rimuovere dal browser il certificato al termine dei test perché potrebbe interferire con altri processi di autenticazione.

Esempio di autenticazione

In questo esempio il **certificato è stato verificato correttamente** ma la credenziale A2A associata non è stata ancora generata o non è stata associata all'applicazione.



Esempio di autenticazione **FALLITA (HANDSHAKE FAILURE)** a seguito dell'utilizzo di un **CERTIFICATO emesso da Certification Autority non di fiducia**, cioè non appartenente alla lista di Mozilla Foundation:



In questo CASO il certificato rilasciato non potrà essere utilizzato per l'autenticazione applicativa.

Nel caso di certificato scaduto, la sua sostituzione con un certificato valido potrà essere effettuata tramite le modalità previste nel documento "Manuale per la registrazione e autenticazione a SIOPE+" .

FAQ

1. Gestione dei certificati digitali - È ammessa la gestione via software dei certificati per la protezione del canale, oppure risulta obbligatorio l'utilizzo di apparati HW (i cosiddetti HSM)?

Solo i certificati di firma devono essere conservati su dispositivi sicuri per l'apposizione della firma del tipo SmartCard (CNS). I certificati utilizzati per l'autenticazione del canale sono di norma oggetti su file protetti da opportuni SW (Keystore). La responsabilità della gestione della sicurezza ricade interamente sul possessore del certificato associato alla credenziale.

2. Formato dei certificati digitali - Che tipo di certificati sono i file con estensione ".**pem**"? Si fa sempre riferimento al certificato di cifratura e di autenticazione?

Il formato PEM è il formato più comunemente utilizzato dalle Certification Authorities per emettere i certificati. Altre estensioni convenzionali possono essere .crt e .cer.

I PEM sono file ASCII con codifica Base64 e contengono "-----BEGIN CERTIFICATE-----" all'inizio e "-----END CERTIFICATE-----" alla fine. Possono essere in formato PEM sia certificati server, che certificati intermedi e chiavi private. (cfr. https://it.wikipedia.org/wiki/X.509 , https://www.ietf.org/rfc/rfc5280.txt)

3. Tipologia dei certificati

*-- **Certificato X509 di autenticazione**: necessario per mutua autenticazione SSL tra gli applicativi delle controparti e i sistemi applicativi.*

***COMMON NAME**= <campo libero, si consiglia di utilizzare un nome descrittivo dell'ente/controparte/intermediario/tramite>*

X509v3 Key Usage critical:
Digital Signature, Key Encipherment

X509v3 Extended Key Usage:
TLS Web Server Authentication, TLS Web Client Authentication

*- **Certificato di cifratura**: utilizzato dall'sistema per cifrare i flussi di risposta verso le controparti. Può essere riutilizzato lo stesso certificato utilizzato per l'autenticazione. L'utilizzo di questo certificato è facoltativo ai fini del colloquio con SIOPE+*

4. Acquisto dei certificati digitali - L'acquisizione dei certificati per l'autenticazione e cifratura dei dati vanno richiesti presso un'azienda accreditata dall'Agenzia per l'Italia Digitale AGID sia per le informazioni da segnalante al sistema che viceversa? L'AGID è l'ente certificatore sia per i segnalanti che per il sistema?

*No, la normativa vigente (EIDAS-AGID) impone vincoli solo sui certificati digitali utilizzabili per Firma Digitale Qualificata, Marca Temporale e **CNS** (identificazione persona fisica).*

***I certificati di autenticazione** possono essere rilasciati da una qualunque CA il cui certificato ROOT sia presente nel CA_BUNDLE della fondazione Mozilla e consultabile al link:*

https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/

5. Se il sistema deve utilizzare la chiave AES per cifrare i dati che inoltra al segnalante, sarebbe possibile aprire quella cifratura solo con la chiave privata del sistema. Non è chiaro se il processo preveda la cifratura della chiave AES con la chiave pubblica del sistema o no.

Lo standard di riferimento per la cifratura è la RFC3852. Si segnala che la chiave simmetrica di cifratura AES viene cifrata con la chiave pubblica del destinatario in modo che il solo il destinatario la possa aprire usando la sua chiave privata.

6. Come si usa il certificato X509 di autenticazione

La mutua autenticazione tra applicazione secondo avviene con lo scambio di certificati digitali di tipo X509. Lo standard di riferimento per l'autenticazione è la RFC5246

7. La connessione SSL dell'applicazione fallisce. Il certificato del SERVER non sembra valido.

L'applicazione CLIENT deve avere nel suo archivio delle autorità attendibili (TRUSTSTORE) il certificato della CA_root che ha firmato il certificato della CA intermedia che. A sua volta , ha firmato il certificato SSL del SERVER che eroga le funzionalità applicative.

8. Abbiamo incluso la *CA_root* nell'archivio delle autorità attendibili ma la connessione fallisce ancora, il certificato del CLIENT non sembra accettato.

*Durante l'handshake TLS il client deve fornire anche il certificato della CA intermedia (CHAIN). Tale modalità è espressamente prevista dallo standard RFC5246 reperibile al link del Transport Layer Security (TLS) Protocol Versione 1.2 (https://tools.ietf.org/html/rfc5246#section-7.4.2): "certificate_list: this is a sequence (**chain**) of certificates. The sender's certificate MUST come first in the list. Each following certificate MUST directly certify the one preceding it. Because certificate validation requires that root keys be distributed independently, the self-signed certificate that specifies the root certificate authority MAY be omitted from the chain, under the assumption that the remote end must already possess it in order to validate it in any case. **The same message type and structure will be used for the client's response to a certificate request** message. Note that a client MAY send no certificates if it does not have an appropriate certificate to send in response to the server's authentication request."*

Glossario

PEM - certificato codificato con Base64, racchiuso tra "-----BEGIN CERTIFICATE-----" e "-----END CERTIFICATE-----";

DER - certificato codificato con DER, (codificato in forma binaria).

X509 – Standard per le infrastrutture a chiave pubblica (v. LINK)

RFC - documento pubblicato dalla Internet Engineering Task Force, che riporta informazioni o specifiche riguardanti nuove ricerche, innovazioni e metodologie dell'ambito informatico o, più nello specifico, di Internet.

RFC5246 : Standard che descrive il TLS 1.2.

_____