SPID – SISTEMA PUBBLICO PER L'IDENTITÀ DIGITALE
Avviso nr. 41 – Versione 1.0
15/04/2022

INTEGRAZIONE LL.GG. "OPENID CONNECT IN SPID"

METADATA OPENID PROVIDER

Gli OpenID Provider (OP) devono usare "jwks_uri".
Per il "jwks_uri" è necessario che la URL afferisca allo stesso dominio, di titolarità dell'OP, sul quale è pubblicato il metadata.

METADATA CLIENT

I Relying Party (RP) possono utilizzare sia il "jwks" che il "jwks_uri".

REGISTRO SPID OPENID CONNECT

Con apposito atto di AGID sarà prevista l'adozione di "Federation OIDC SPID 1.0".

CIFRATURA E SIGILLO

Per tutte le richieste e le risposte HTTP deve essere utilizzando il protocollo TLS nella versione più recente disponibile.
La firma e la cifratura devono essere prodotte in accordo a quanto indicato negli standard JWS (https://tools.ietf.org/html/rfc7515) e JWE (https://tools.ietf.org/html/rfc7516), utilizzando chiavi RSA almeno a 2048 bit, si consiglia 4096, e algoritmo di digest SHA-256 o superiore.

Algoritmi di firma per i JWS da supportare:

1. RS256
2. RS512

Algoritmi di cifratura (Key Encryption Algorithms - alg) per i JWE da supportare:

1. RSA-OAEP
2. RSA-OAEP-256

Algoritmi di cifratura (Content Encryption Algorithms - enc) per i JWE da supportare:

1. A128CBC-HS256
2. A256CBC-HS512

Quando nelle Linee Guida OpenID Connect in SPID viene indicato che il contenuto deve essere firmato e cifrato è necessario prima firmare e poi cifrare.

AUTHORIZATION REQUEST

L'implementazione del valore "verify" per il parametro "prompt" è sospesa fino a indicazione contraria di AGID.

AUTHORIZATION CODE

L'Authorization Code deve essere realizzato nella forma di UUID (https://tools.ietf.org/html/rfc4122).
L'Authorization Code deve avere un periodo di validità pari a 5 minuti e bisogna verificare che non sia stato usato precedentemente.

PKCE

Il "code verifier" e il "code challenge" devono essere realizzati in base a quanto stabilito nel RFC 7636, "Proof Key for Code Exchange by OAuth Public Clients" (https://datatracker.ietf.org/doc/html/rfc7636).

REQUEST TOKEN

La differenza tra "iat" e "exp" è a discrezione del Client.

TOKEN ENDPOPINT RESPONSE

Il parametro "expires_in" nella Response Token non deve essere superiore a 300 secondi.

ID TOKEN

Il parametro "exp" nell'ID Token deve essere pari a "iat" + 5 minuti
L'ID Token deve avere un periodo di validità pari a 5 minuti e bisogna verificare che non sia stato usato

precedentemente.
Il parametro "exp" nell'ID Token rilasciato a seguito di richiesta di refresh deve essere pari a "iat" + 30 giorni – tempo dell'autenticazione originaria.
[Vedere schema per il flusso con refresh]

ACCESS TOKEN

Il parametro "exp" nell'Acces Token deve essere pari a "iat" + 15 minuti.
L'Access Token deve avere un periodo di validità pari a 15 minuti e deve essere riutilizzabile fino alla scadenza.

RESPONSE USERINFO

Nel JWT contenuto nella response del userinfo endpoint è necessario anche il parametro:

cty	(Content Type) Deve essere valorizzato con JWT

REFRESH

La differenza tra "iat" è "exp" può essere di massimo 30 giorni.
La validità del refresh token deve essere calcolata a partire dall'autenticazione originaria, salve eccezioni stabilite da AGID in casi specifici.
I casi d'uso previsti sono:

CU 1 (senza rotazione)
- t1 : RP effettua autenticazione con offline_access, quindi ottiene refresh_token RT1 (30gg)
- t2 = t1 + 4gg : dopo 4gg da t1 RP fa richiesta a /token presentando RT1. OP rilascia nuovo access_token e lo stesso RT1 con validità 30gg da t1 (26gg)

CU 2 (con rotazione)
- t1 : RP effettua autenticazione con offline_access, quindi ottiene refresh_token RT1 (30gg)
- t2 = t1 + 4gg : dopo 4gg da t1 RP fa richiesta a /token presentando RT1. OP rilascia nuovo access_token e nuovo refresh_token RT2 con validità 30gg da t1 (26gg)
- t3 = t1 + 27gg : dopo 27gg da t1 RP fa richiesta a /token presentando RT2. OP rilascia nuovo access_token e nuovo refresh_token RT3 con validità 30gg da t1 (3gg)

CU 3 (con rotazione ed eccezione per RP X)
- t1 : RP X effettua autenticazione con offline_access, quindi ottiene refresh_token RT1 (30gg)
- t2 = t1 + 4gg : dopo 4gg da t1 RP X fa richiesta a /token presentando RT1. OP riconosce che la richiesta proviene da X e rilascia nuovo access_token e nuovo refresh_token RT2 con validità 30gg da t2 (30gg)
- t3 = t1 + 27gg : dopo 27gg da t1 RP fa richiesta a /token presentando RT2. OP riconosce che la richiesta proviene da X e rilascia nuovo access_token e nuovo refresh_token RT3 con validità 30gg da t3 (30gg)

Al momento, viene richiesta solo la CU2, ma gli IdP devono garantire anche CU1 e CU3 in base a specifiche richieste di AGID.
[vedere schema per il flusso con refresh token].

TEMPI DI VALIDITA' DEI TOKEN

I Token devono rimanere validi fino alla loro scadenza
La scadenza di un token non incide sulla scadenza dei token collegati.
In caso di rinnovo a seguito dell'utilizzo del refresh token, i precedenti token devono essere resi inutilizzabili, se non già scaduti.

TOLLERANZE

La tolleranza in eccesso o in difetto sui valori "iat" e "exp" deve essere massimo di 3 minuti.

TEMPI DI RISPOSTA

I vari tempi di risposta devono essere compatibili con il tempo totale di risposta previsto dagli SLA allegati alla Convenzione.

TRACCIATURE

- TRACCIATURE OPENID PROVIDER

Ai fini della tracciatura l'OpenID Provider dovrà mantenere un Registro delle transazioni contenente i tracciati delle richieste di autenticazione servite negli ultimi 24 mesi. L'unità di memorizzazione di tale registro dovrà rendere persistente per ogni transazione l'identificativo dell'identità digitale (spidCode) interessata dalla transazione e tutti i messaggi scambiati con il rispettivo RP. Al fine di consentire una facile ricerca e consultazione dei dati di tracciature potrebbe essere opportuno memorizzare in ogni record informazioni direttamente estratte dai suddetti messaggi. A titolo non esaustivo, i messaggi memorizzati nel registro sono i seguenti:

- <AuthenticationRequest>
- <AuthenticationResponse>
- <TokenRequest>
- <TokenResponse>
- <UserInfoRequest>
- <UserInfoResponse>

e, per ogni messaggio, ove applicabili, potrebbero essere indicizzate ai fini di ricerca e consultazione le seguenti informazioni:

- authorization code
- client_id
- jti
- iss
- sub
- iat
- exp

- TRACCIATURE RELYING PARTY

Il comma 2 dell'articolo 13 del DPCM obbliga i fornitori di servizi (relying party) alla conservazione per ventiquattro mesi delle informazioni necessarie a imputare alle singole identità digitali le operazioni effettuate sui propri sistemi. A tal fine un RP dovrà mantenere un Registro delle transazioni contenente i tracciati delle richieste verso i diversi endpoint servite negli ultimi 24 mesi. L'unità di memorizzazione di tale registro dovrà rendere persistente per ogni transazione tutti i messaggi scambiati con il rispettivo OP. Al fine di consentire una facile ricerca e consultazione dei dati di tracciature potrebbe essere opportuno memorizzare in ogni record informazioni direttamente estratte dai suddetti messaggi. A titolo esemplificativo e non esaustivo i messaggi memorizzati nel registro potrebbero essere i seguenti:

- <AuthenticationRequest>
- <AuthenticationResponse>
- <TokenRequest>
- <TokenResponse>
- <UserInfoRequest>
- <UserInfoResponse>

e, per ogni messaggio, ove applicabili, potrebbero essere indicizzate ai fini di ricerca e consultazione le seguenti informazioni:

- authorization code
- client_id
- jti
- iss
- sub
- iat
- exp

- MANTENIMENTO TRACCIATURE

Le tracciature devono essere mantenute nel rispetto del codice della privacy sotto la responsabilità titolare del trattamento dell'Identity Provider. e l'accesso ai dati di tracciatura deve essere riservato a personale incaricato. Al fine di garantire la confidenzialità potrebbero essere adottai meccanismi di cifratura dei dati o impiegati sistemi di basi di dati (DBMS) che realizzano la persistenza cifrata delle informazioni. Per il mantenimento devono essere messi in atto meccanismi che garantiscono l'integrità e il non ripudio.

PARAMETRI COMUNI DEI JWT (FIRMATI)

L'header è costituito dalle seguenti informazioni:

Parametro	Descrizione
typ	OPZIONALE. Ove fosse assente viene considerato "JWT" come definito da https://datatracker.ietf.org/doc/html/rfc7519#section-5.1
alg	valorizzato con l'identificativo JWA dell'algoritmo crittografico utilizzato.
kid	ID della chiave utilizzata per sigillare il token

Il payload è costituito dalle seguenti informazioni:

Parametro	Descrizione
jti	identificativo unico del token.
iss	deve corrispondere al soggetto mittente.
aud	deve corrispondere al soggetto destinatario
iat	istante di generazione del JWT codificato come NumericDate come indicato in RFC 7519 – JSON Web Token (JWT)
exp	istante di scadenza del JWT codificato come NumericDate come indicato in RFC 7519 – JSON Web Token (JWT)

Non sono vietati ulteriori parametri necessari, sia nell'header sia nel payload, con valori opachi.

LOGOUT

I token rilasciati da OP sono validi fino all'istante di scadenza indicato nel parametro exp.
La revoca anticipata di un token e la conseguente chiusura della sessione sull'OP può essere richiesta dal RP tramite l'endpoint di Revocation.

CODICI DI ERRORE

I codici di errore relativi al Cap. 6 Par. 2 delle Linee Guida OpenID Connect in SPID sono i seguenti:

Scenario	Codice errore
L'OP ha negato l'accesso a causa di credenziali non valide o non adeguate al livello SPID richiesto.	**access_denied**
Il client_id indicato nella richiesta non è riconosciuto.	**invalid_client**
La richiesta non è valida a causa della mancanza o della non correttezza di uno o più parametri.	**invalid_request**
Sono stati richiesti degli scope non validi.	**invalid_scope**
L'OP ha riscontrato un problema interno.	**server_error**
L'OP ha riscontrato un problema interno temporaneo.	**temporarily_unavailable**

Il codice di stato http deve essere quello indicato nel RFC di riferimento indicato nelle LLGG.

I codici di errore relativi al Cap. 7 Par. 4 delle Linee Guida OpenID Connect in SPID sono i seguenti:

Scenario	Codice errore
Il client_id indicato nella richiesta non è riconosciuto.	**invalid_client**
Il parametro **grant_type** contiene un valore non corretto.	**unsupported_grant_type**
I parametri **grant_type**, **code**, **code_verifier**, **access_token** non sono validi.	**invalid_grant**
La richiesta non è valida a causa della mancanza o della non correttezza di uno o più parametri.	**invalid_request**
L'OP ha riscontrato un problema interno.	**server_error**
L'OP ha riscontrato un problema interno temporaneo.	**temporarily_unavailable**

Il codice di stato http deve essere quello indicato nel RFC di riferimento indicato nelle LLGG.

I codici di errore relativi al Cap. 9 Par. 3 delle Linee Guida OpenID Connect in SPID sono i seguenti:

Scenario	Codice errore
Il client_id indicato nella richiesta non è riconosciuto.	**invalid_client**
La richiesta non è valida a causa della mancanza o della non correttezza di uno o più parametri.	**invalid_request**
L'OP ha riscontrato un problema interno.	**server_error**
L'OP ha riscontrato un problema interno temporaneo.	**temporarily_unavailable**

Il codice di stato http deve essere quello indicato nel RFC di riferimento indicato nelle LLGG.

USER EXPERIENCE OPENID CONNECT IN SPID

In attesa delle Linee Guida User Experience SPID (Linee Guida UX SPID) resta valida la User Experience già prevista dalle vigenti regole tecniche SPID.

INTROSPECTION

Se il parametro "active" è valorizzato con "false" non è necessario restituire anche gli altri parametri.

SINGLE SIGN ON

Fino all'emanazione di apposite regole tecniche non è previsto il SSO a livello uno di SPID.

GESTIONE SESSIONI OIDC-SAML

Fino all'emanazione di apposite regole tecniche, non è necessario che OIDC e SAML debbano condividere le sessioni.

PAIRWISE IDENTIFIER

L'OP può individuare autonomamente un valore ricollegabile al Client per il "sector_identifier_uri" in caso non sia stato possibile usando i criteri indicati nella specifica OIDC (https://openid.net/specs/openid-connect-core-1_0.html#PairwiseAlg.).

REVOCHE DEI TOKEN

La revoca di un token da parte dell'utente determina anche la revoca dei token collegati.

CIFRATURA HTTPS

Vale quanto già previsto nell'Avviso SPID n. 1

INDICAZIONE DEI LIVELLI IN ACR

L'OP ha facoltà di effettuare l'autenticazione ad un livello superiore a quello richiesto dal RP, come per SAML.
Il RP deve richiedere sempre il solo livello minimo richiesto per l'autenticazione.
Ad es. con la seguente richiesta: "acr_values=https://www.spid.gov.it/SpidL2" l'OP può effettuare l'autenticazione sia con il livello 2 sia con il livello 3.
Nel caso di "offline_access" il RP deve sempre indicare il solo livello minimino richiesto per l'autenticazione, seguito dal livello 1, ai fini della creazione e gestione delle sessioni lunghe revocabili.
Ad es. con la seguente richiesta: "acr_values=https://www.spid.gov.it/SpidL2 https://www.spid.gov.it/SpidL1" l'OP può effettuare l'autenticazione sia con il livello 2 sia con il livello 3, considerando il livello 1 solo ai fini della creazione e gestione delle sessioni lunghe revocabili.
Se il livello minimo richiesto per l'autenticazione è il Livello 1, il RP può indicare:
"acr_values=https://www.spid.gov.it/SpidL1 l'OP può effettuare l'autenticazione sia con il livello 1 sia con il livello 2 sia con il livello 3, considerando ,l'indicazione del RP anche ai fini della creazione e gestione delle sessioni lunghe revocabili.
Se il RP indica nella "authentication request" con "off line access" prima il livello 1 e poi il livello minimino richiesto per l'autenticazione
Ad es.: con la seguente richiesta: "acr_values=https://www.spid.gov.it/SpidL1 https://www.spid.gov.it/SpidL2", l'OP restituisce l'errore "invalid_request".

Il Responsabile del progetto SPID

FRANCESCO PAORICI

Firmato digitalmente da FRANCESCO PAORICI
Data: 2022.05.06 09:09:36 +02'00'

OPENID CONNECT AUTHORIZATION CODE FLOW CON REFRESH TOKEN
Caso d'uso CUR (CU2)