Authentic 2 - Bug #66436

auth_oidc: en l'absence d'un set de clé la réception d'un idtoken chiffré avec RSA lève une trace

21 juin 2022 11:00 - Benjamin Dauvergne

Statut:	Nouveau	**Début:**	21 juin 2022
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

https://sentry.entrouvert.org/entrouvert/publik/issues/61792/?environment=test

```
AttributeError: 'NoneType' object has no attribute 'get_key'
  File "django/core/handlers/exception.py", line 34, in inner
    response = get_response(request)
  File "django/core/handlers/base.py", line 115, in _get_response
    response = self.process_exception_by_middleware(e, request)
  File "django/core/handlers/base.py", line 113, in _get_response
    response = wrapped_callback(request, *callback_args, **callback_kwargs)
  File "authentic2/decorators.py", line 40, in f
    return func(request, *args, **kwargs)
  File "django/views/generic/base.py", line 71, in view
    return self.dispatch(request, *args, **kwargs)
  File "django/views/generic/base.py", line 97, in dispatch
    return handler(request, *args, **kwargs)
  File "authentic2_auth_oidc/views.py", line 131, in get
    response = self.handle_authorization_response(request)
  File "authentic2_auth_oidc/views.py", line 175, in handle_authorization_response
    return self.handle_code(request, provider, nonce, code)
  File "authentic2_auth_oidc/views.py", line 288, in handle_code
    user = authenticate(
  File "authentic2/utils/misc.py", line 1311, in authenticate
    return dj_authenticate(request=request, **kwargs)
  File "django/contrib/auth/__init__.py", line 73, in authenticate
    user = backend.authenticate(request, **credentials)
  File "authentic2_auth_oidc/backends.py", line 41, in authenticate
    return self._authenticate(
  File "authentic2_auth_oidc/backends.py", line 52, in _authenticate
    id_token.deserialize(provider)
  File "authentic2_auth_oidc/utils.py", line 113, in deserialize
    decoded = parse_id_token(self._encoded, provider)
  File "authentic2_auth_oidc/utils.py", line 64, in parse_id_token
    key = provider.jwkset.get_key(kid=kid)
```

En fait OIDCProvider.idtoken_algo n'est pas vraiment utilisé partout parce que l'idtoken est décodé deux fois, une fois via utils.parse_id_token() et une fois directement (toujours avec la classe JWT) dans la méthode authenticate() du backend, il y a quelque chose à simplifier ici, je pense que idtoken_algo pourrait être retiré et l'algo déduit implicitement à partir d'un choix plus simple, autoriser les algorithmes hmac ou pas et déduire les algorithmes RSA/ECDSA en fonction des clés disponibles.

Si un idtoken RSA/ECDSA est reçu sans clé disponible on pourra indiquer le messaque adéquat.