



class Auth::SCRAM

Authentication using SCRAM

Table of Contents

- 1 [Synopsis](#)
- 2 [Methods](#)
- 2.1 [new](#)

```
unit package Auth;  
class SCRAM { ... }
```

Synopsis

See documentation of [Auth::SCRAM::Client](#) and [Auth::SCRAM::Server](#).

Methods

Auth::SCRAM has some methods which are mostly used by the client or server roles and have not much use by the caller directly and are therefore not explained.

new

Client side BUILD is defined as

```
multi submethod BUILD (
  Str :$username!,
  Str :$password!,
  Str :$authzid,
  Bool :$case-preserved-profile = True,

  Callable :$CGH = &sha1,
  :$client-object!,
)
```

Initialize the process. The Cryptographic Hash function `$CGH` is by default set to `&sha1` from the `OpenSSL::Digest` module. The authorization id (`$authzid`) is needed when you want things done using the privileges of someone else. The `$client-object` object is an object performing client side tasks. The methods in this object are called by the methods in the `SCRAM::Client` Role.

Usernames and password (and maybe the authorization id) must be normalized. Older versions of the scram process made use of SASLprep. This module will use the PRECIS framework defined by rfc7564 and crystalized in module `Unicode::PRECIS`. There are several classes and profiles in that framework. For usernames there is a profile to map case to lowercase and one to

preserve it. This is controlled by the boolean `$case-preserved-profile` and by default set to `True`. There is only one type of profile for passwords so no control needed there.

Server side BUILD is defined as

```
multi submethod BUILD (  
    Bool :$case-preserved-profile = True,  
    Callable :$CGH = &sha1,  
    :$server-object!,  
)
```

The `$server-object` object is an object performing server side tasks. The methods in this object are called by the methods in the `SCRAM::Server` Role. Username, password and authorization id are not needed when a server side object is given because it will be provided by the client via an account registration mechanism and the clients first message will provide the username and authorization id to work with.

For specific client and server information look for the roles `SCRAM::Client` and `SCRAM::Server`.