

Prover (Client role)

Verifier (Server role)

Shared information : cipher suite, *ProverId*, *VerifierId*, and Context

Prover 'key pair' (*w0*, *w1*) derived from password

Registration record (*w0*, *L*) derived from password

`psa_pake_setup()` with key (*w0*, *w1*)  
`psa_pake_set_role(PSA_PAKE_ROLE_CLIENT)`  
`psa_pake_set_user(ProverId)`  
`psa_pake_set_peer(VerifierId)`  
`psa_pake_set_context(Context)`

Generate key share *X*

`psa_pake_output()` for *shareP* = *X*

(*shareP*)

`psa_pake_setup()` with key (*w0*, *L*) or key (*w0*, *w1*)  
`psa_pake_set_role(PSA_PAKE_ROLE_SERVER)`  
`psa_pake_set_user(VerifierId)`  
`psa_pake_set_peer(ProverId)`  
`psa_pake_set_context(Context)`

Validate *shareP*

`psa_pake_input()` for *shareP*

Generate key share *Y*

`psa_pake_output()` for *shareV* = *Y*

Compute *K\_shared*, *confirmP'* and *confirmV*

`psa_pake_output()` for *confirmV*

(*shareV*, *confirmV*)

Validate *shareV*

`psa_pake_input()` for *shareV*

Compute *K\_shared*,  
*confirmP* and *confirmV'*  
Verify *confirmV'* = *confirmV*

`psa_pake_input()` for *confirmV*

`psa_pake_output()` for *confirmP*

(*confirmP*)

`psa_pake_get_shared_key()` to extract *K\_shared*

Verify *confirmP'* = *confirmP*

`psa_pake_input()` for *confirmP*

`psa_pake_get_shared_key()` to extract *K\_shared*