Recovering from Lost Devices in FIDO

Trust a Private Recovery Service for Privacy and Availability

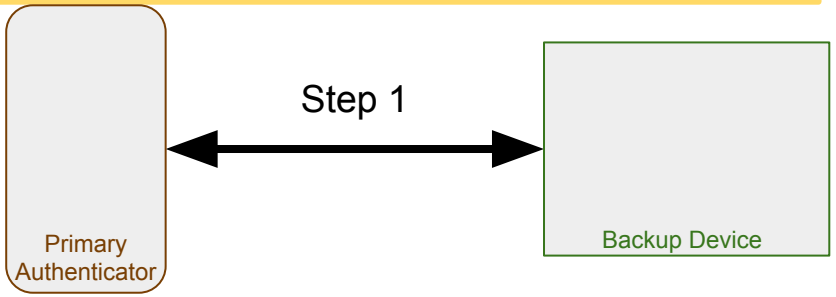
Alex Takakuwa, University of Washington

Private Recovery Service (PRS) Solution

- Assumptions:
  - Secure channel between backup device and primary authenticators
- Goals:
  - Recovery from lost primary authenticator
  - Users trust PRS for privacy (prevent linkability) and availability
  - Relying Party doesn’t need to trust (or know about) Third Party
  - Allow a single recovery device to recover many primary authenticators
  - Prevent security attacks by the PRS
  - Do not otherwise weaken existing FIDO scheme
  - Allow for multiple backup devices
  - Allow many types of backup (third party server, hardware device, key splitting)
  - Allow Transfer of Access from Primary Authenticators
- Open Problems:
  - Transfer to a new backup device

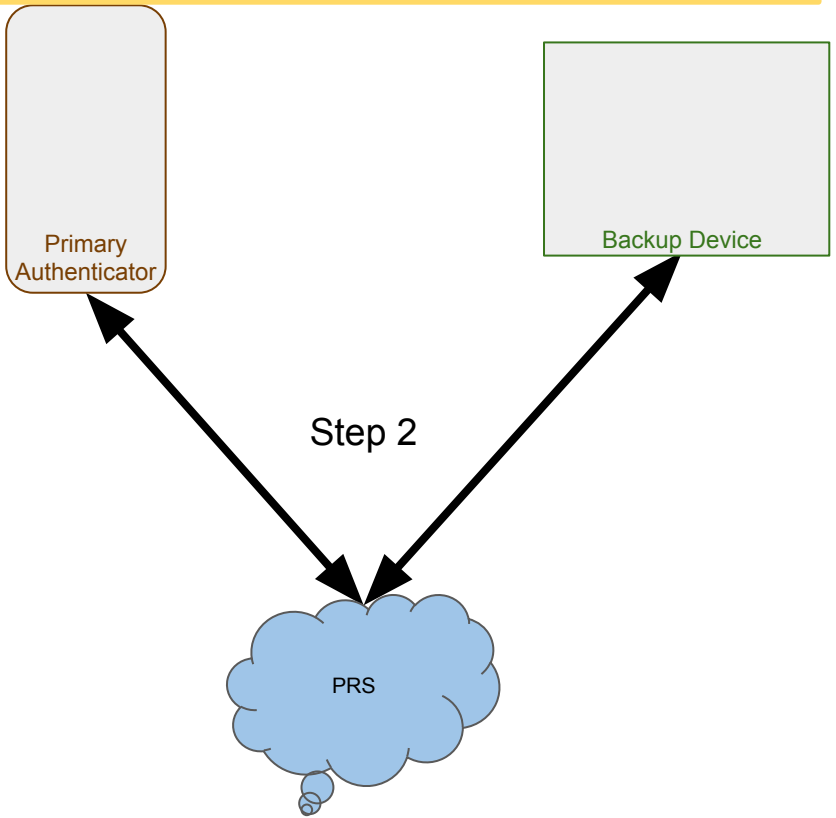
PRS Solution - Setup (User Experience)

- User needs a backup device
- User syncs backup device with primary authenticator (Step 1)
- User creates an account at the PRS and syncs each device with the PRS (Step 2)
- User uses primary authenticator for future authentications and registrations



PRS Solution - Setup (User Experience)

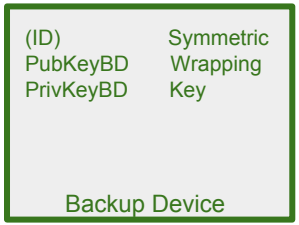
- User needs a backup device
- User syncs backup device with primary authenticator (Step 1)
- User creates an account at the PRS and syncs each device with the PRS (Step 2)
- User uses primary authenticator for future authentications and registrations



PRS Solution - Setup (Technical)

1. Primary authenticator and backup device create long-lived priv/pub key pair. The pub key is its long-term identifier.
2. User sets up local authentication on both devices
3. User creates secure channel between devices, who can exchange their identifiers.
4. User creates an account with PRS, registers both long-term identifiers with the PRS
5. Backup device generates (* some number) of key pairs, encrypts each private key with its wrapping key, and stores the pairs at the PRS. Each of these keys should be signed with PrivKeyBD (ID for the Backup Device) so that the Primary Authenticators can verify they are owned by a legitimate backup device.

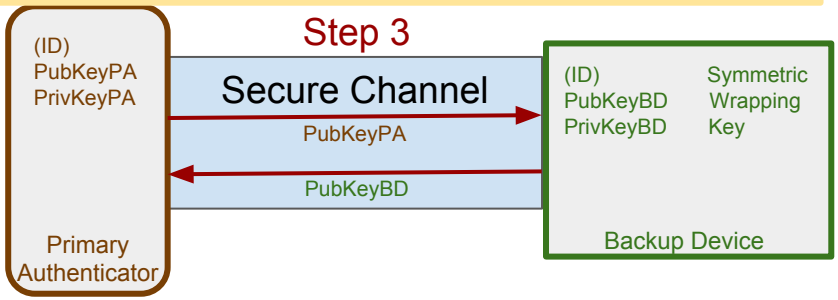
* The backup device should generate (at first setup) enough key pairs to last for all registrations from all authenticators backed up by this device. (num devices * lifetime num accounts). It does not store these locally.



PRS Solution - Setup (Technical)

1. Primary authenticator and backup device create long-lived priv/pub key pair. The pub key is its long-term identifier.
2. User sets up local authentication on both devices
3. User creates secure channel between devices, who can exchange their identifiers.
4. User creates an account with PRS, registers both long-term identifiers with the PRS
5. Backup device generates (* some number) of key pairs, encrypts each private key with its wrapping key, and stores the pairs at the PRS. Each of these keys should be signed with PrivKeyBD (ID for the Backup Device) so that the Primary Authenticators can verify they are owned by a legitimate backup device.

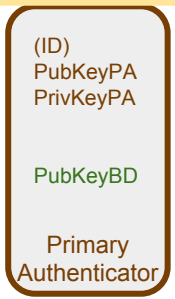
* The backup device should generate (at first setup) enough key pairs to last for all registrations from all authenticators backed up by this device. (num devices * lifetime num accounts). It does not store these locally.

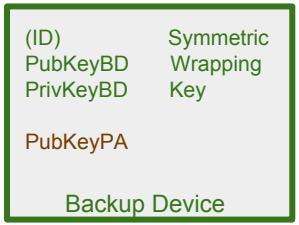


PRS Solution - Setup (Technical)

1. Primary authenticator and backup device create long-lived priv/pub key pair. The pub key is its long-term identifier.
2. User sets up local authentication on both devices
3. User creates secure channel between devices, who can exchange their identifiers.
4. User creates an account with PRS, registers both long-term identifiers with the PRS
5. Backup device generates (* some number) of key pairs, encrypts each private key with its wrapping key, and stores the pairs at the PRS. Each of these keys should be signed with PrivKeyBD (ID for the Backup Device) so that the Primary Authenticators can verify they are owned by a legitimate backup device.

* The backup device should generate (at first setup) enough key pairs to last for all registrations from all authenticators backed up by this device. (num devices * lifetime num accounts). It does not store these locally.

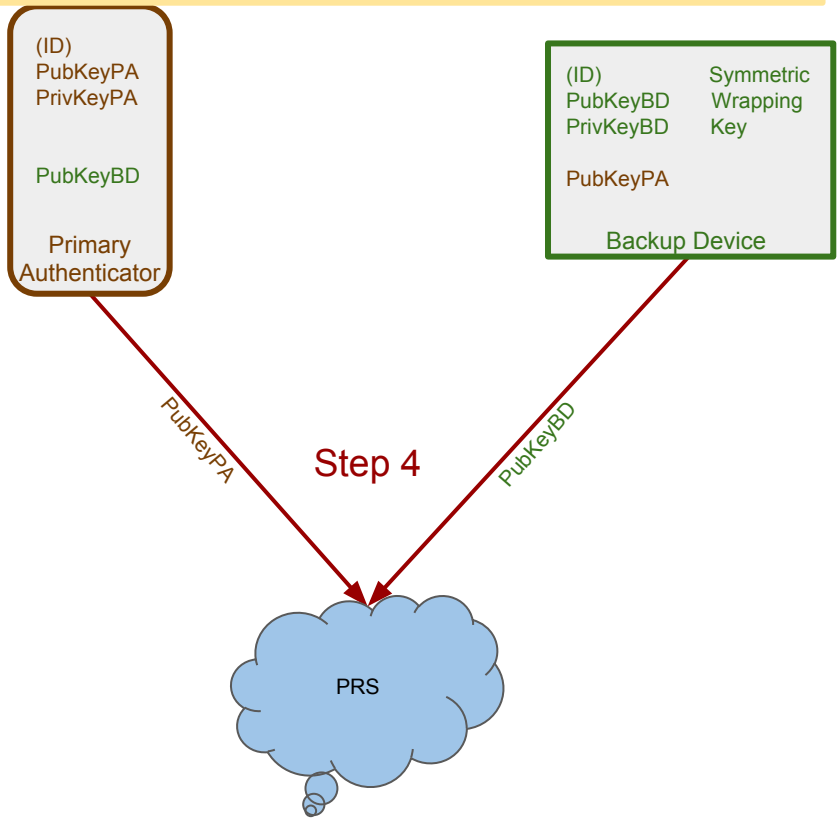




PRS Solution - Setup (Technical)

1. Primary authenticator and backup device create long-lived priv/pub key pair. The pub key is its long-term identifier.
2. User sets up local authentication on both devices
3. User creates secure channel between devices, who can exchange their identifiers.
4. User creates an account with PRS, registers both long-term identifiers with the PRS
5. Backup device generates (* some number) of key pairs, encrypts each private key with its wrapping key, and stores the pairs at the PRS. Each of these keys should be signed with PrivKeyBD (ID for the Backup Device) so that the Primary Authenticators can verify they are owned by a legitimate backup device.

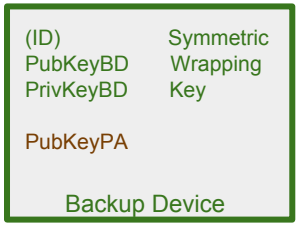
* The backup device should generate (at first setup) enough key pairs to last for all registrations from all authenticators backed up by this device. (num devices * lifetime num accounts). It does not store these locally.

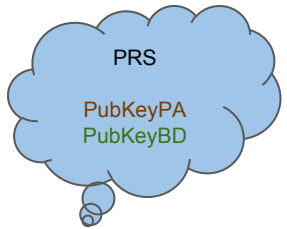


PRS Solution - Setup (Technical)

1. Primary authenticator and backup device create long-lived priv/pub key pair. The pub key is its long-term identifier.
2. User sets up local authentication on both devices
3. User creates secure channel between devices, who can exchange their identifiers.
4. User creates an account with PRS, registers both long-term identifiers with the PRS
5. Backup device generates (* some number) of key pairs, encrypts each private key with its wrapping key, and stores the pairs at the PRS. Each of these keys should be signed with PrivKeyBD (ID for the Backup Device) so that the Primary Authenticators can verify they are owned by a legitimate backup device.

* The backup device should generate (at first setup) enough key pairs to last for all registrations from all authenticators backed up by this device. (num devices * lifetime num accounts). It does not store these locally.

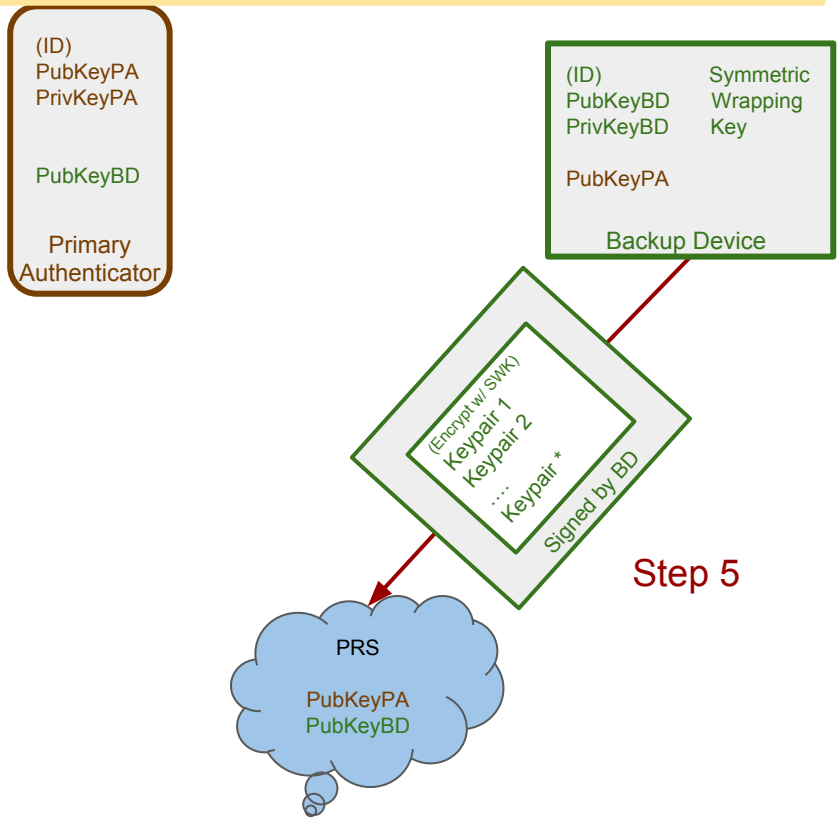




PRS Solution - Setup (Technical)

1. Primary authenticator and backup device create long-lived priv/pub key pair. The pub key is its long-term identifier.
2. User sets up local authentication on both devices
3. User creates secure channel between devices, who can exchange their identifiers.
4. User creates an account with PRS, registers both long-term identifiers with the PRS
5. Backup device generates (* some number) of key pairs, encrypts each private key with its wrapping key, and stores the pairs at the PRS. Each of these keys should be signed with PrivKeyBD (ID for the Backup Device) so that the Primary Authenticators can verify they are owned by a legitimate backup device.

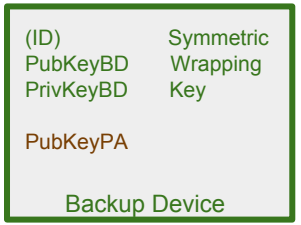
* The backup device should generate (at first setup) enough key pairs to last for all registrations from all authenticators backed up by this device. (num devices * lifetime num accounts). It does not store these locally.

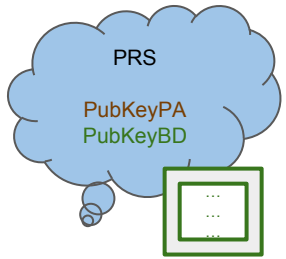


PRS Solution - Setup (Technical)

1. Primary authenticator and backup device create long-lived priv/pub key pair. The pub key is its long-term identifier.
2. User sets up local authentication on both devices
3. User creates secure channel between devices, who can exchange their identifiers.
4. User creates an account with PRS, registers both long-term identifiers with the PRS
5. Backup device generates (* some number) of key pairs, encrypts each private key with its wrapping key, and stores the pairs at the PRS. Each of these keys should be signed with PrivKeyBD (ID for the Backup Device) so that the Primary Authenticators can verify they are owned by a legitimate backup device.

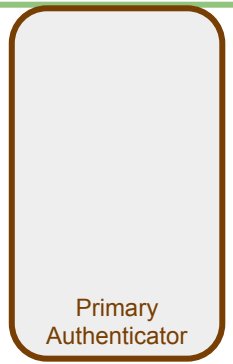
* The backup device should generate (at first setup) enough key pairs to last for all registrations from all authenticators backed up by this device. (num devices * lifetime num accounts). It does not store these locally.

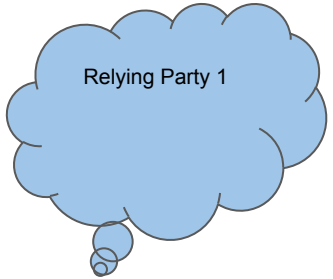




PRS Solution - Registration (User Experience)

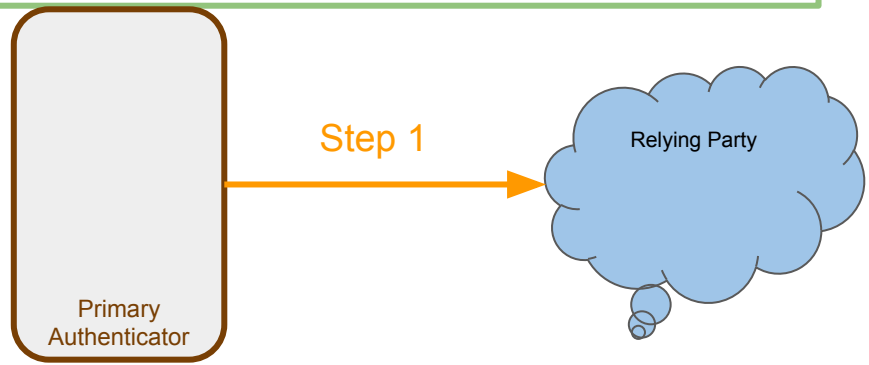
- User does a standard FIDO registration (completely unchanged) with the Relying Party (Step 1)
- User updates the Private Recovery Server through another channel (Step 2)
- Does not require the Backup Device
- Relying Party is unaware of the PRS





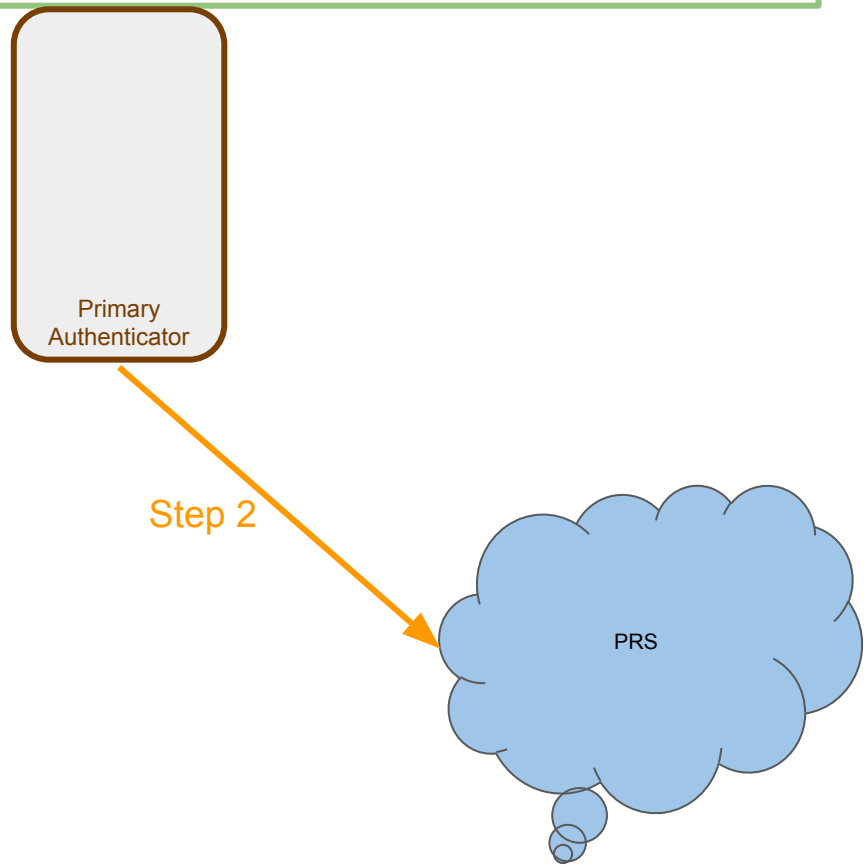
PRS Solution - Registration (User Experience)

- User does a standard FIDO registration (completely unchanged) with the Relying Party (Step 1)
- User updates the Private Recovery Server through another channel (Step 2)
- Does not require the Backup Device
- Relying Party is unaware of the PRS



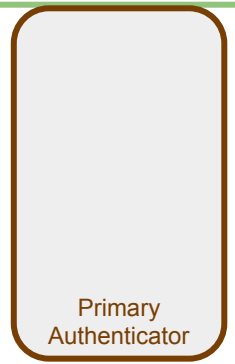
PRS Solution - Registration (User Experience)

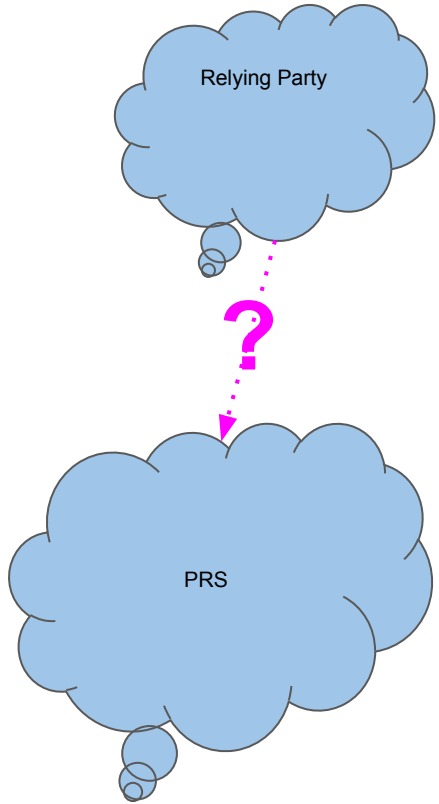
- User does a standard FIDO registration (completely unchanged) with the Relying Party (Step 1)
- User updates the Private Recovery Server through another channel (Step 2)
- Does not require the Backup Device
- Relying Party is unaware of the PRS



PRS Solution - Registration (User Experience)

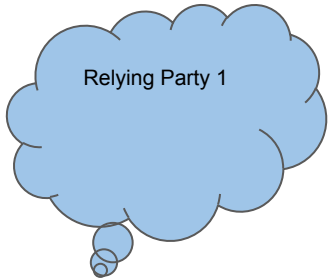
- User does a standard FIDO registration (completely unchanged) with the Relying Party (Step 1)
- User updates the Private Recovery Server through another channel (Step 2)
- Does not require the Backup Device
- Relying Party is unaware of the PRS





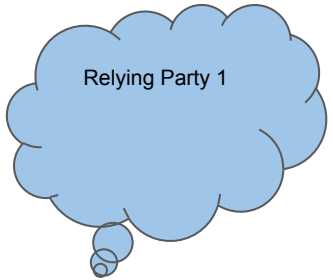
PRS Solution - Registration (Technical)

1. Standard Registration
   a. Primary Authenticator generates keys
   b. Registers public key with RP1
2. Primary Authenticator logs in to Private Recovery Service using ID key pair.
3. Primary authenticator provides information about the registration with RP1 and signs that information with its ID PrivKeyPA.
   a. PubKey
   b. Key Handle
   c. AppID
4. PRS selects an unused backup public key and presents it to the Primary Authenticator, along with the signature provided by the Backup Device.
5. The Primary Authenticator checks the signature to make sure it matches its stored PubKeyBD
6. The Primary Authenticator creates a delegation from its own PrivKeyRP1 to the selected backup public key and sends that delegation to the RPS.
7. The RPS moves that key pair out of the “unused” portion and associates it with the stored information about the registration with RP1.



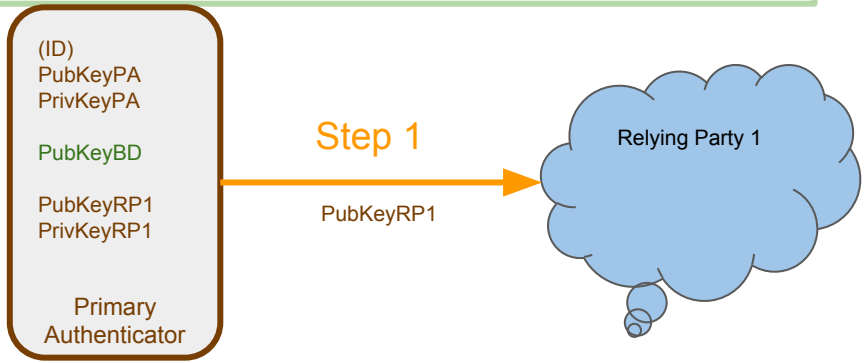
PRS Solution - Registration (Technical)

1. Standard Registration
   a. Primary Authenticator generates keys
   b. Registers public key with RP1
2. Primary Authenticator logs in to Private Recovery Service using ID key pair.
3. Primary authenticator provides information about the registration with RP1 and signs that information with its ID PrivKeyPA.
   a. PubKey
   b. Key Handle
   c. AppID
4. PRS selects an unused backup public key and presents it to the Primary Authenticator, along with the signature provided by the Backup Device.
5. The Primary Authenticator checks the signature to make sure it matches its stored PubKeyBD
6. The Primary Authenticator creates a delegation from its own PrivKeyRP1 to the selected backup public key and sends that delegation to the RPS.
7. The RPS moves that key pair out of the “unused” portion and associates it with the stored information about the registration with RP1.



PRS Solution - Registration (Technical)

1. Standard Registration
   a. Primary Authenticator generates keys
   b. Registers public key with RP1
2. Primary Authenticator logs in to Private Recovery Service using ID key pair.
3. Primary authenticator provides information about the registration with RP1 and signs that information with its ID PrivKeyPA.
   a. PubKey
   b. Key Handle
   c. AppID
4. PRS selects an unused backup public key and presents it to the Primary Authenticator, along with the signature provided by the Backup Device.
5. The Primary Authenticator checks the signature to make sure it matches its stored PubKeyBD
6. The Primary Authenticator creates a delegation from its own PrivKeyRP1 to the selected backup public key and sends that delegation to the RPS.
7. The RPS moves that key pair out of the “unused” portion and associates it with the stored information about the registration with RP1.

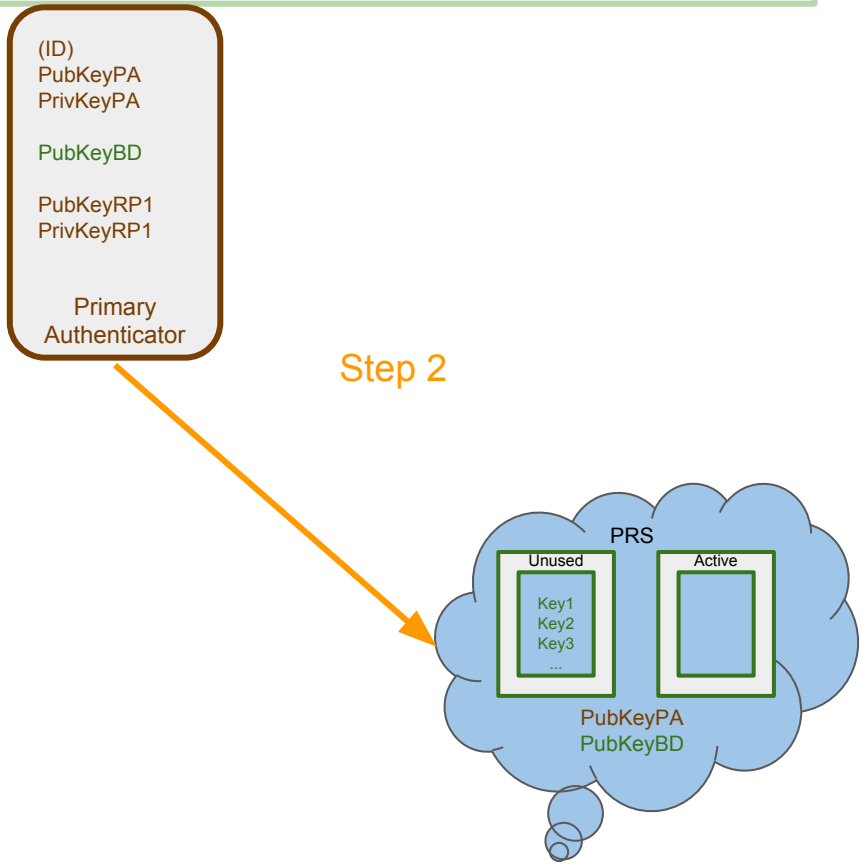


PRS Solution - Registration (Technical)

1. Standard Registration
   a. Primary Authenticator generates keys
   b. Registers public key with RP1
2. Primary Authenticator logs in to Private Recovery Service using ID key pair.
3. Primary authenticator provides information about the registration with RP1 and signs that information with its ID PrivKeyPA.
   a. PubKey
   b. Key Handle
   c. AppID
4. PRS selects an unused backup public key and presents it to the Primary Authenticator, along with the signature provided by the Backup Device.
5. The Primary Authenticator checks the signature to make sure it matches its stored PubKeyBD
6. The Primary Authenticator creates a delegation from its own PrivKeyRP1 to the selected backup public key and sends that delegation to the RPS.
7. The RPS moves that key pair out of the “unused” portion and associates it with the stored information about the registration with RP1.

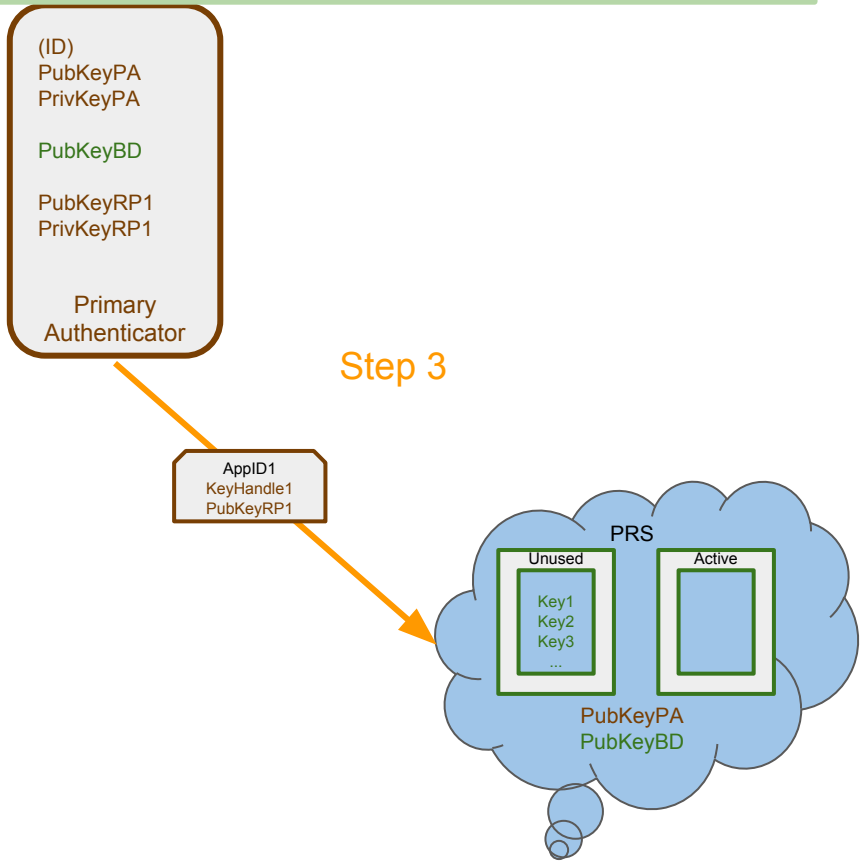
PRS Solution - Registration (Technical)

1. Standard Registration
   a. Primary Authenticator generates keys
   b. Registers public key with RP1
2. Primary Authenticator logs in to Private Recovery Service using ID key pair PubKeyPA.
3. Primary authenticator provides information about the registration with RP1 and signs that information with its ID PrivKeyPA.
   a. PubKey
   b. Key Handle
   c. AppID
4. PRS selects an unused backup public key and presents it to the Primary Authenticator, along with the signature provided by the Backup Device.
5. The Primary Authenticator checks the signature to make sure it matches its stored PubKeyBD
6. The Primary Authenticator creates a delegation from its own PrivKeyRP1 to the selected backup public key and sends that delegation to the RPS.
7. The RPS moves that key pair out of the “unused” portion and associates it with the stored information about the registration with RP1.



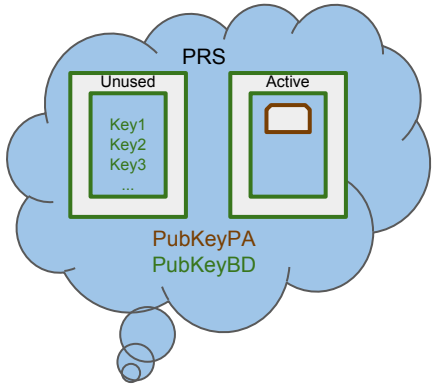
PRS Solution - Registration (Technical)

1. Standard Registration
   a. Primary Authenticator generates keys
   b. Registers public key with RP1
2. Primary Authenticator logs in to Private Recovery Service using ID key pair PubKeyPA.
3. Primary authenticator provides information about the registration with RP1 and signs that information with its ID PrivKeyPA.
   a. PubKey
   b. Key Handle
   c. AppID
4. PRS selects an unused backup public key and presents it to the Primary Authenticator, along with the signature provided by the Backup Device.
5. The Primary Authenticator checks the signature to make sure it matches its stored PubKeyBD
6. The Primary Authenticator creates a delegation from its own PrivKeyRP1 to the selected backup public key and sends that delegation to the RPS.
7. The RPS moves that key pair out of the “unused” portion and associates it with the stored information about the registration with RP1.



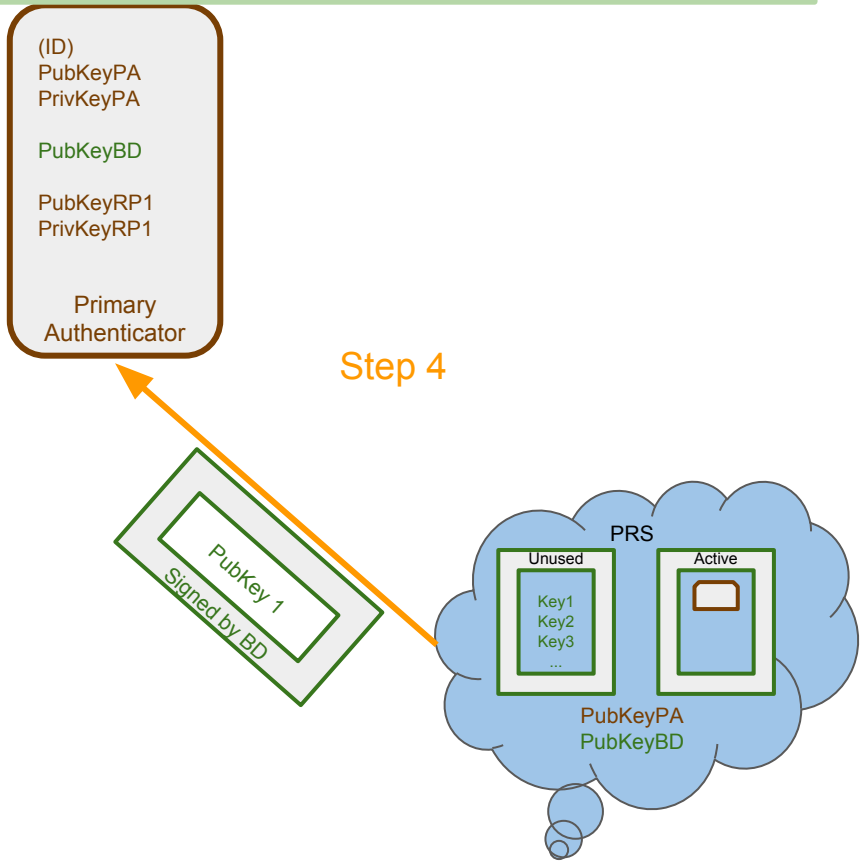
PRS Solution - Registration (Technical)

1. Standard Registration
   a. Primary Authenticator generates keys
   b. Registers public key with RP1
2. Primary Authenticator logs in to Private Recovery Service using ID key pair PubKeyPA.
3. Primary authenticator provides information about the registration with RP1 and signs that information with its ID PrivKeyPA.
   a. PubKey
   b. Key Handle
   c. AppID
4. PRS selects an unused backup public key and presents it to the Primary Authenticator, along with the signature provided by the Backup Device.
5. The Primary Authenticator checks the signature to make sure it matches its stored PubKeyBD
6. The Primary Authenticator creates a delegation from its own PrivKeyRP1 to the selected backup public key and sends that delegation to the RPS.
7. The RPS moves that key pair out of the “unused” portion and associates it with the stored information about the registration with RP1.



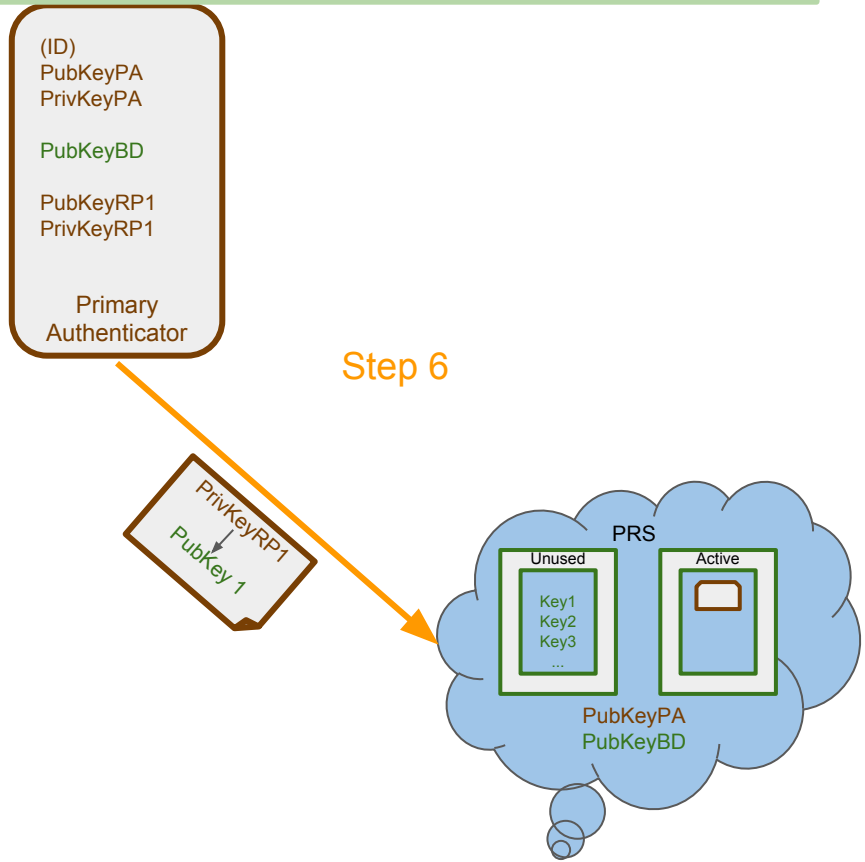
PRS Solution - Registration (Technical)

1. Standard Registration
   a. Primary Authenticator generates keys
   b. Registers public key with RP1
2. Primary Authenticator logs in to Private Recovery Service using ID key pair PubKeyPA.
3. Primary authenticator provides information about the registration with RP1 and signs that information with its ID PrivKeyPA.
   a. PubKey
   b. Key Handle
   c. AppID
4. PRS selects an unused backup public key and presents it to the Primary Authenticator, along with the signature provided by the Backup Device.
5. The Primary Authenticator checks the signature to make sure it matches its stored PubKeyBD
6. The Primary Authenticator creates a delegation from its own PrivKeyRP1 to the selected backup public key and sends that delegation to the RPS.
7. The RPS moves that key pair out of the “unused” portion and associates it with the stored information about the registration with RP1.



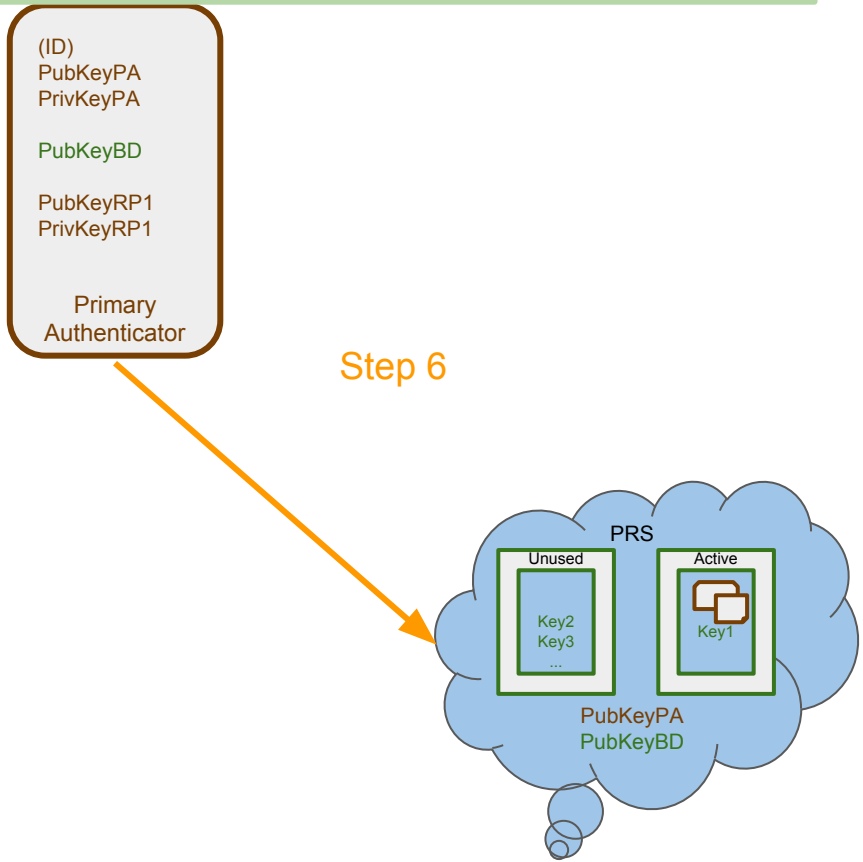
PRS Solution - Registration (Technical)

1. Standard Registration
   a. Primary Authenticator generates keys
   b. Registers public key with RP1
2. Primary Authenticator logs in to Private Recovery Service using ID key pair PubKeyPA.
3. Primary authenticator provides information about the registration with RP1 and signs that information with its ID PrivKeyPA.
   a. PubKey
   b. Key Handle
   c. AppID
4. PRS selects an unused backup public key and presents it to the Primary Authenticator, along with the signature provided by the Backup Device.
5. The Primary Authenticator checks the signature to make sure it matches its stored PubKeyBD
6. The Primary Authenticator creates a delegation from its own PrivKeyRP1 to the selected backup public key and sends that delegation to the RPS.
7. The RPS moves that key pair out of the “unused” portion and associates it with the stored information about the registration with RP1.



PRS Solution - Registration (Technical)

1. Standard Registration
   a. Primary Authenticator generates keys
   b. Registers public key with RP1
2. Primary Authenticator logs in to Private Recovery Service using ID key pair PubKeyPA.
3. Primary authenticator provides information about the registration with RP1 and signs that information with its ID PrivKeyPA.
   a. PubKey
   b. Key Handle
   c. AppID
4. PRS selects an unused backup public key and presents it to the Primary Authenticator, along with the signature provided by the Backup Device.
5. The Primary Authenticator checks the signature to make sure it matches its stored PubKeyBD
6. The Primary Authenticator creates a delegation from its own PrivKeyRP1 to the selected backup public key and sends that delegation to the RPS.
7. The RPS moves that key pair out of the “unused” portion and associates it with the stored information about the registration with RP1.

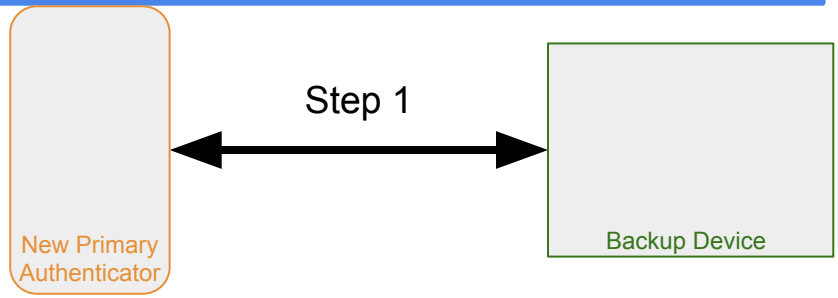


PRS Solution - Authentication

- Unchanged from Standard FIDO

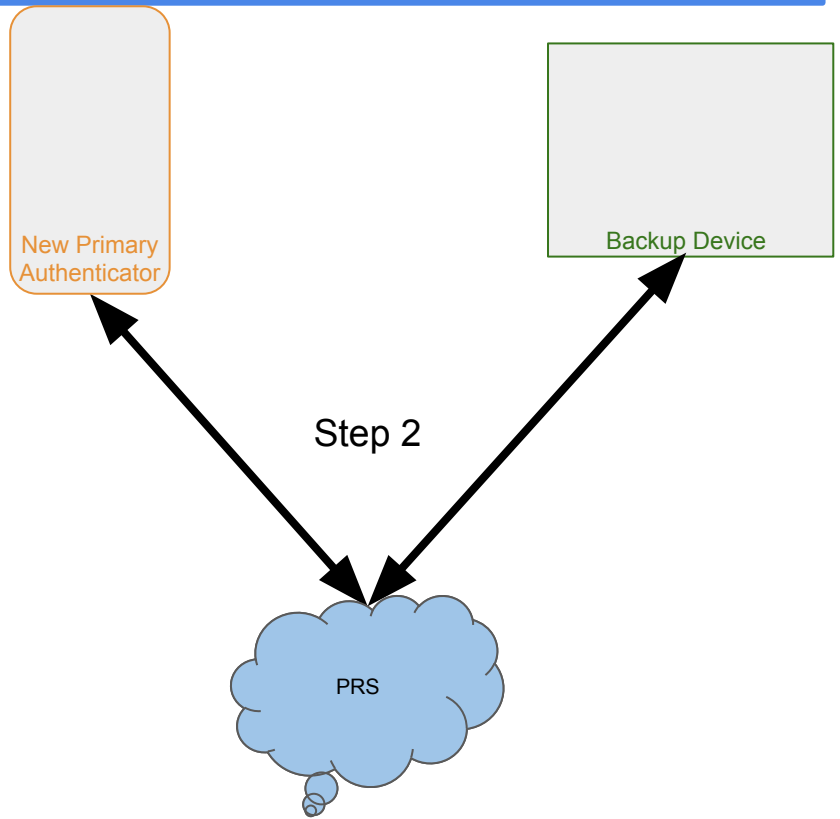
PRS Solution - Recovery (User Experience)

- User retrieves the backup device
- User sets up local authentication on the new Primary Authenticator
- User syncs backup device with new primary authenticator (Step 1)
- User syncs both devices with the PRS simultaneously (Step 2)
- User tells the PRS which device the New Primary Authenticator is replacing (Step 3)
- User uses new primary authenticator for future authentications and registrations



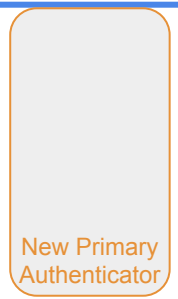
PRS Solution - Recovery (User Experience)

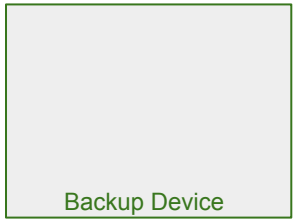
- User retrieves the backup device
- User sets up local authentication on the new Primary Authenticator
- User syncs backup device with new primary authenticator (Step 1)
- User syncs both devices with the PRS simultaneously (Step 2)
- User tells the PRS which device the New Primary Authenticator is replacing (Step 3)
- User uses new primary authenticator for future authentications and registrations



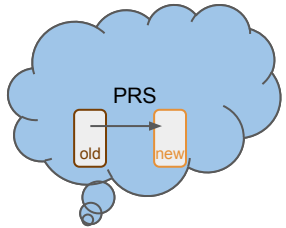
PRS Solution - Recovery (User Experience)

- User retrieves the backup device
- User sets up local authentication on the new Primary Authenticator
- User syncs backup device with new primary authenticator (Step 1)
- User syncs both devices with the PRS simultaneously (Step 2)
- User tells the PRS which device the New Primary Authenticator is replacing (Step 3)
- User uses new primary authenticator for future authentications and registrations



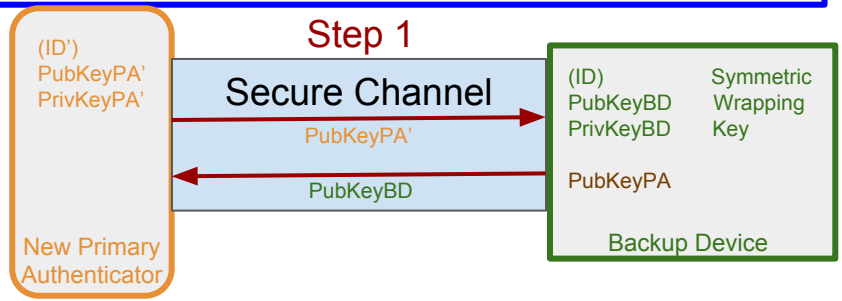


Step 3



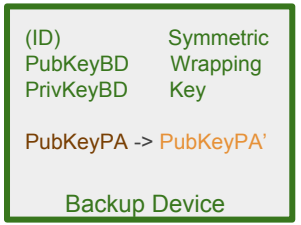
PRS Solution - Recovery (Technical)

1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
    a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
    a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)



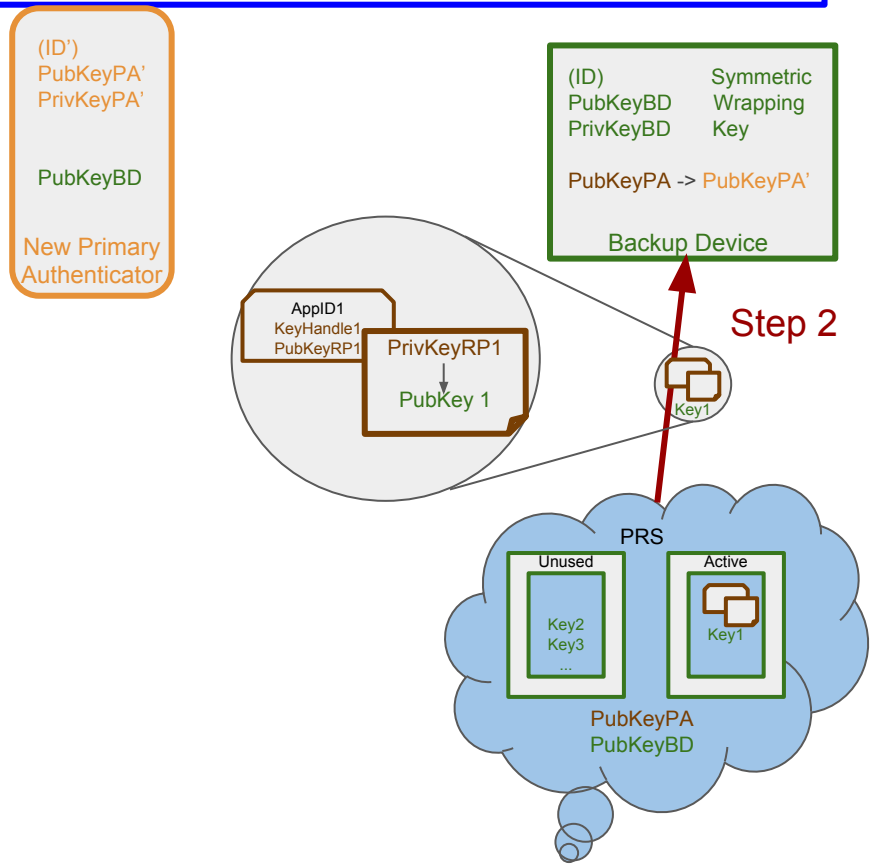
PRS Solution - Recovery (Technical)

1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
    a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
    a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)



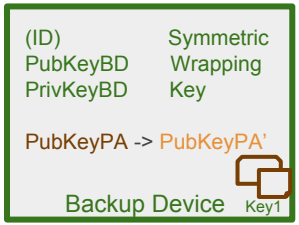
PRS Solution - Recovery (Technical)

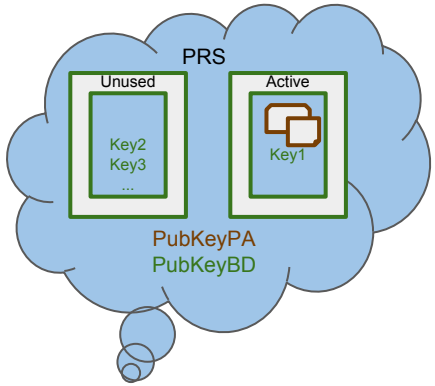
1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
   a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
   a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)



PRS Solution - Recovery (Technical)

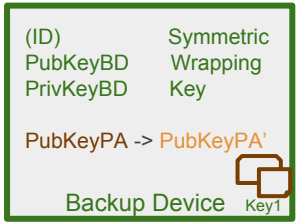
1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
   a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
   a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)

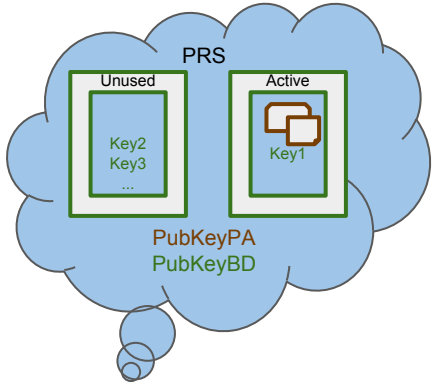




PRS Solution - Recovery (Technical)

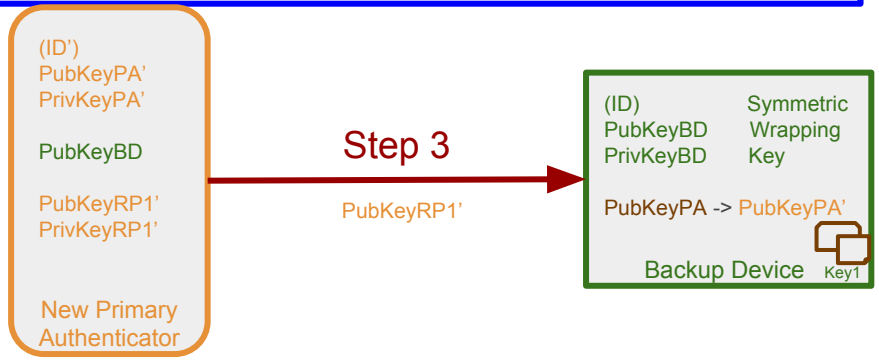
1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
   a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
   a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)

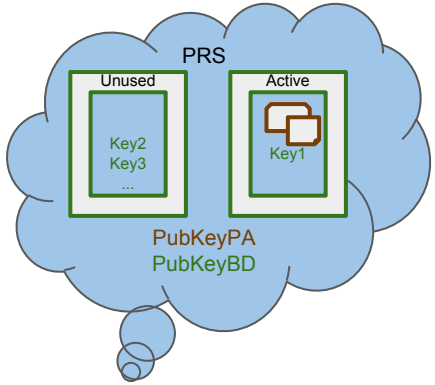




PRS Solution - Recovery (Technical)

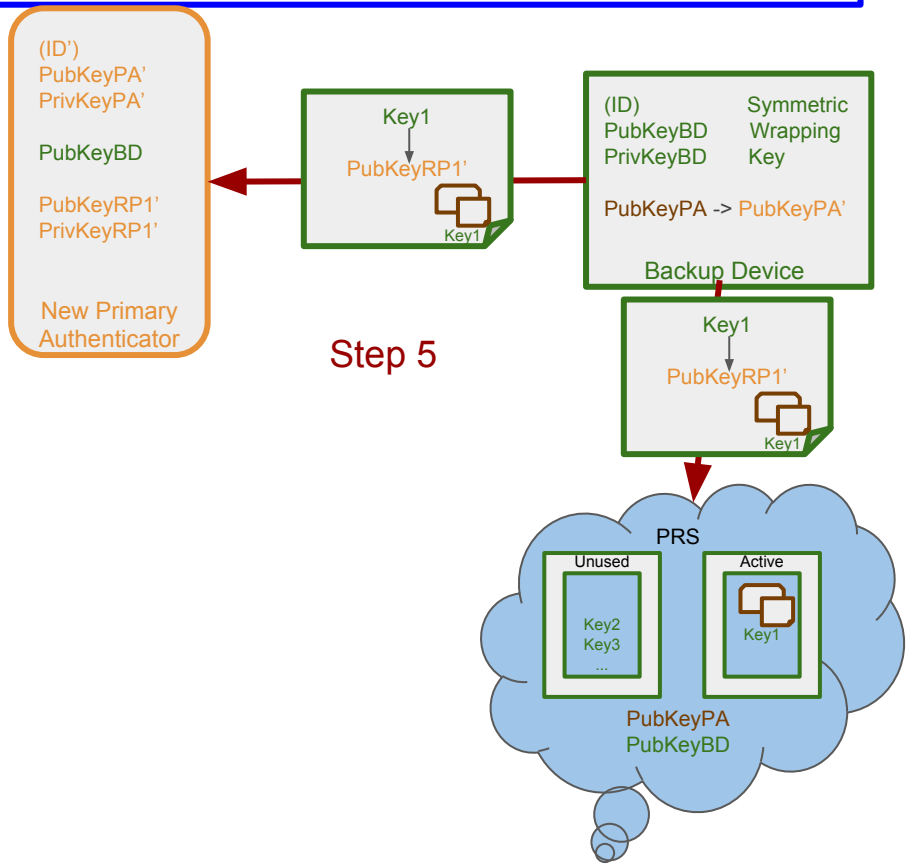
1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
    a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
    a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)





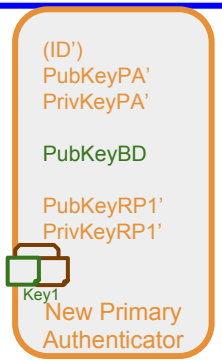
PRS Solution - Recovery (Technical)

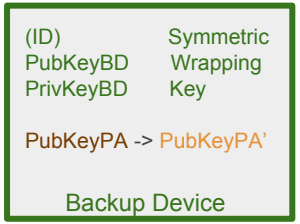
1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
    a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
    a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)

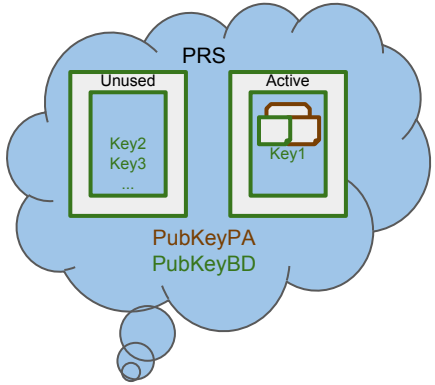


PRS Solution - Recovery (Technical)

1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
   a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
   a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)

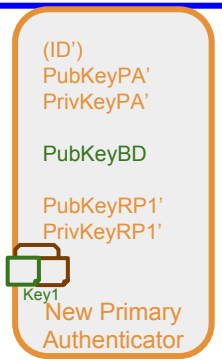


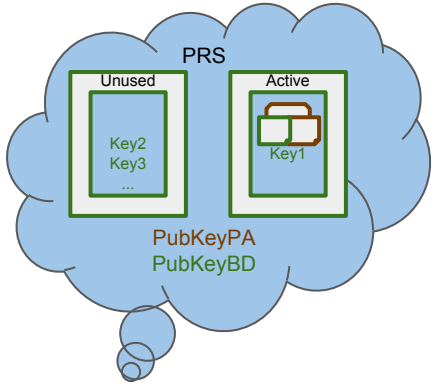




PRS Solution - Recovery (Technical)

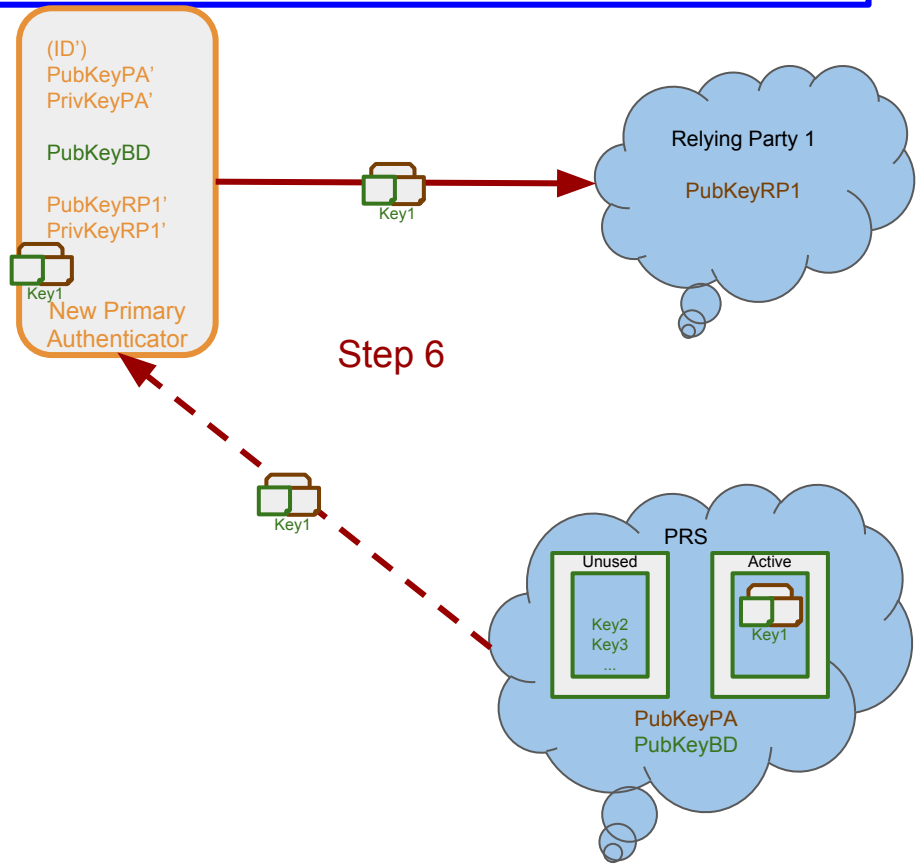
1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
    a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
    a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)





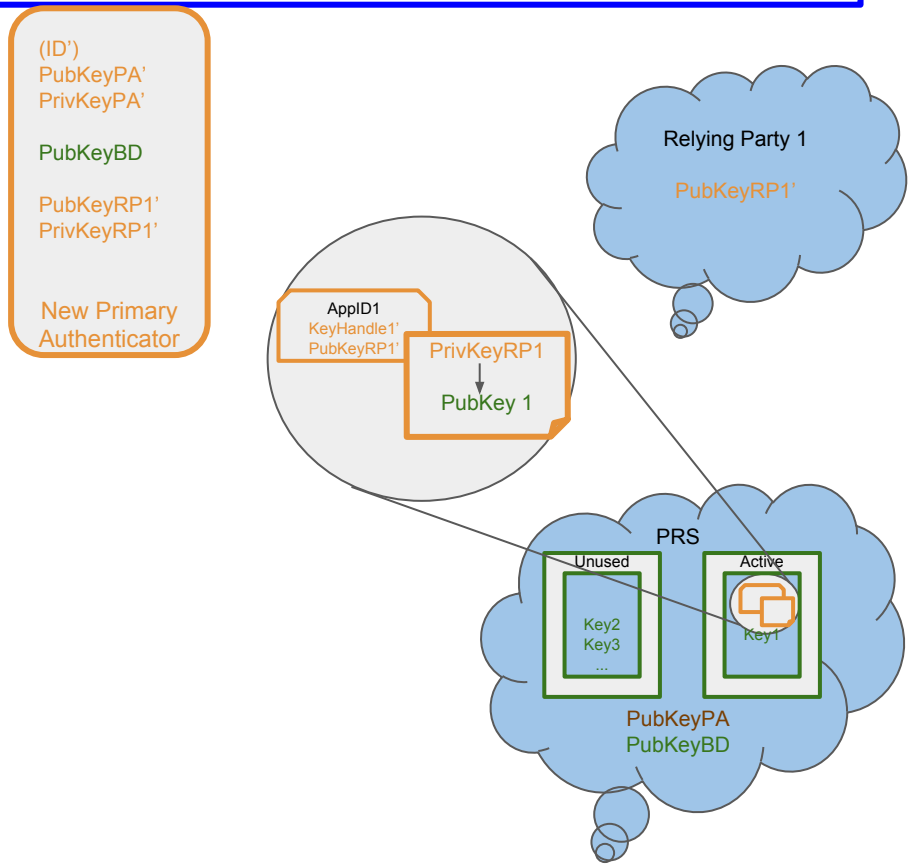
PRS Solution - Recovery (Technical)

1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
    a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
    a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)



PRS Solution - Recovery (Technical)

1. Create a secure channel between New Primary Authenticator and Backup Device so the devices can exchange IDs.
2. The PRS sends each credential assigned to the old Device (PubKeyPA) to the Backup Authenticator
    a. Includes *public key* and *metadata* associated with the registration with the RP, the *Active Backup public key* and *Active Backup private key* (encrypted with the wrapping key), and the *delegation* from the old device's PubKeyPA -> Active Backup Key
3. The Backup Authenticator gets a new key from the New Primary Authenticator (signed with PrivKeyPA')
    a. The New Primary Authenticator should probably sign the metadata as well.
4. The Backup Authenticator decrypts the private key with its wrapping key and uses it to create a delegation from that private key to the newly generated public key PubKeyRP1'
5. The Backup Authenticator delivers the delegation to the New Primary Authenticator and/or the PRS.
6. At next login, the New Primary Authenticator retrieves the certificate (can look up by key handle, since it has it stored) and delivers the chain PrivKeyRP1 -> Active Backup Key -> PubKeyRP1'
7. The Relying Party removes access from the old Authenticator and adds access for the New Authenticator credentials.
8. The PRS updates its Active Credentials with the new Registration Information, and the New Primary Authenticator Provides a credential delegating from its new credential to the Active Backup Key for future recoveries. (PrivKeyRP1' -> Active Backup Key)



PRS Solution - Transfer Access

- Transfer Access should be straightforward, as in the Transfer Access Protocol
- Old Key copies the IDs of its backups to the new key
- The New Key needs to generate IDs
- The Old Key needs to sign a transfer from the Old ID to the New ID (so that the backup device can verify the new device is trusted)
- Transfer Access proceeds for all existing key pairs on the old device
- *The new device must sign delegations from its new key pairs to the respective Active Backup Public Key, sign in to the PRS and deliver those delegations.*

PRS Solution - Revocation

- The backup key should be able to revoke access for any device at any RP by looking at all registrations for that device.
- Revocation would simply include delivering to the RP the transfer from the selected primary authenticator to the active backup public key and indicating to the RP that no new key should be registered.
    - This is dangerous if there are no other keys on the account.