Protect Sensitive Data with Ada Keystore

Stéphane Carrez

FOSDEM 2020

Why Ada Keystore?

- Store sensitive parameters of an Ada server :
  - Database connection passwords
  - API secret keys
- Tool to store sensitive information :
  - Passwords and accounts
  - Encrypted documents without exposing their keys

Getting a secret (1)

- Open keystore with simple password (simple but less secure : password in code!)

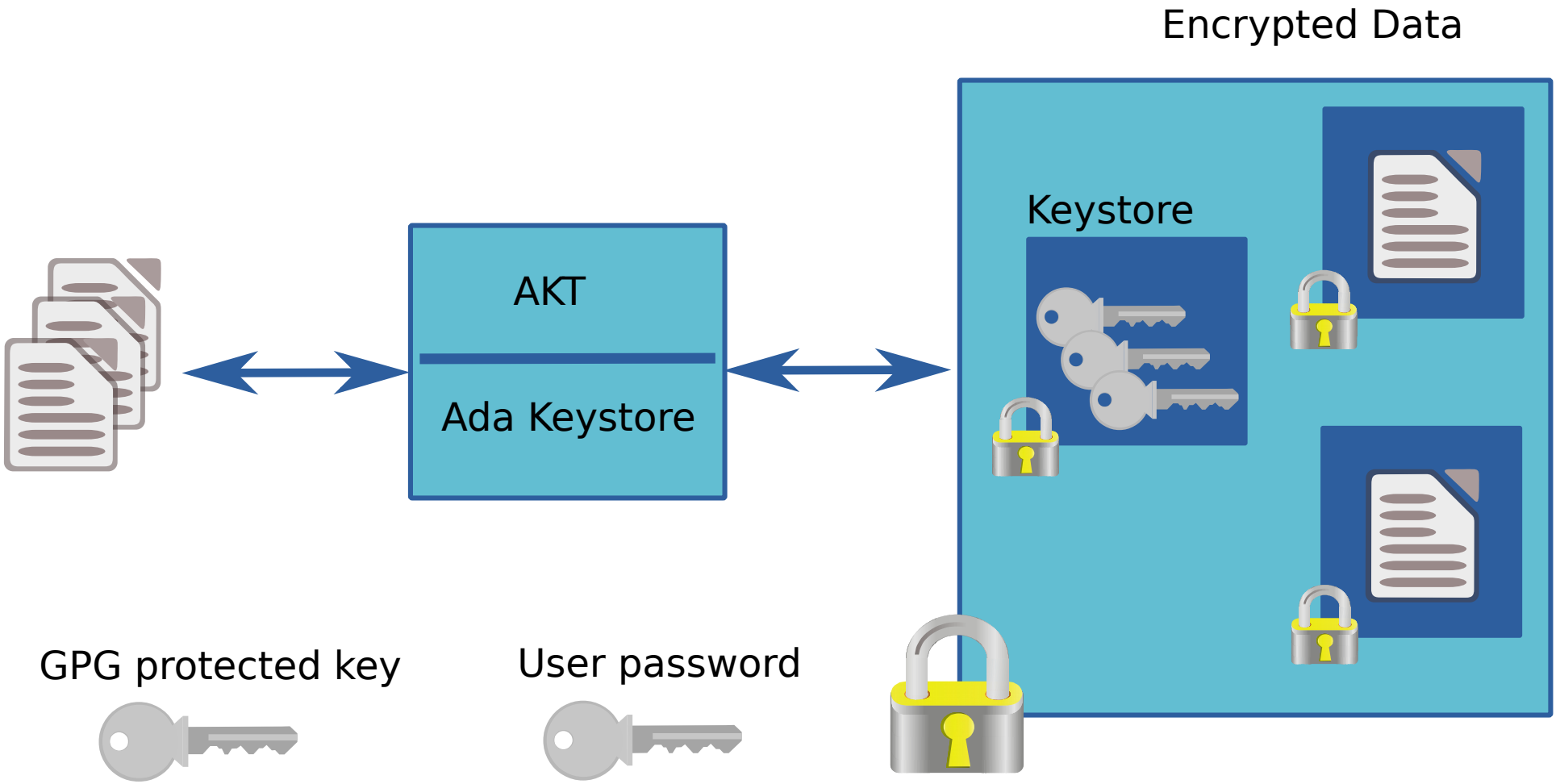
```
with Keystore.Files;
function Get_Secret return String is
   WS   : Keystore.Files.Wallet_File;
   Pass : Keystore.Secret_Key:= Keystore.Create ("...");
begin
   WS.Open (Pass, "secure.akt");
   return WS.Get ("api-key");
end Get_Secret;
```

Getting a secret (2)

- Open keystore protected with GPG

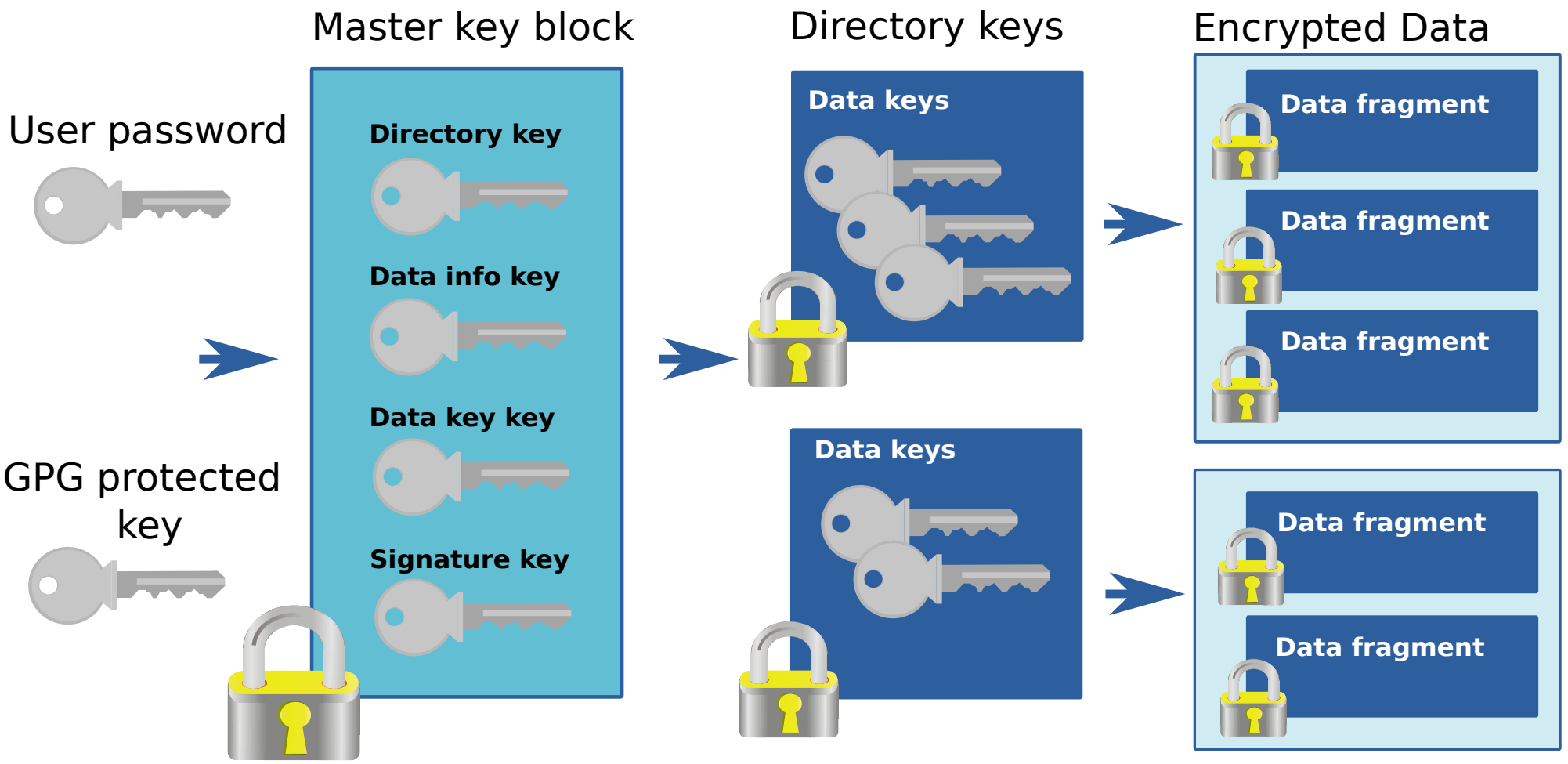
```
with Keystore.GPG;
with Keystore.Passwords.GPG;
with Keystore.Files;
function Get_Secret return String is
   WS   : Keystore.Files.Wallet_File;
   Ctx  : Keystore.Passwords.GPG.Context_Type;
begin
   Keystore.GPG.Open (WS, Ctx, "secure.akt");
   return WS.Get ("api-key");
end Get_Secret;
```

Ada Keystore: a secure storage



https://github.com/stcarrez/ada-keystore

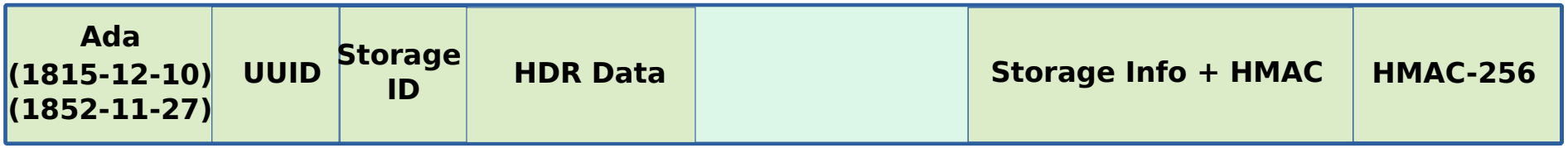
Ada Keystore: with multiple keys



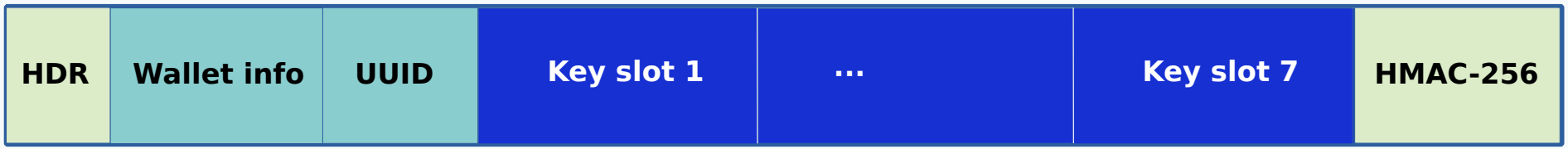
https://github.com/stcarrez/ada-keystore

Ada Keystore: with blocks (unchained)

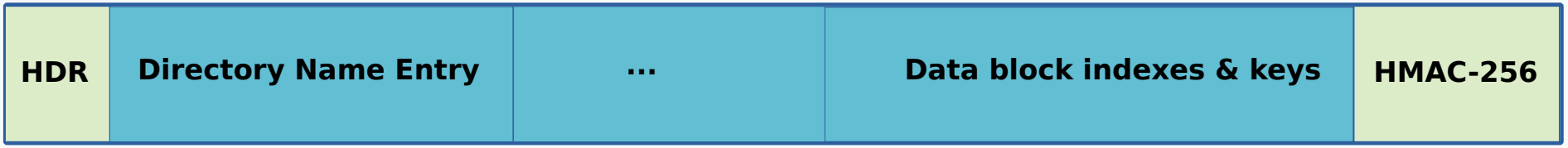
Header block 4K



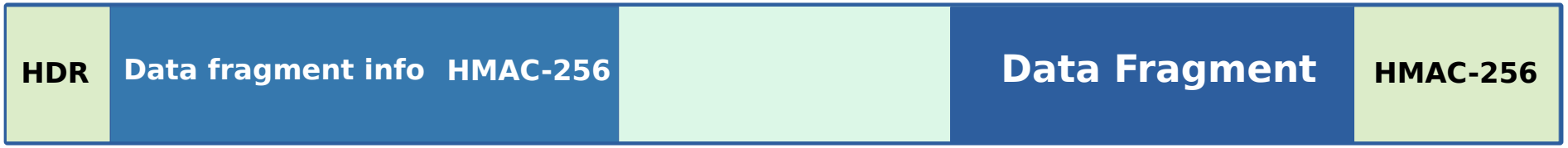
Master key block



Directory block

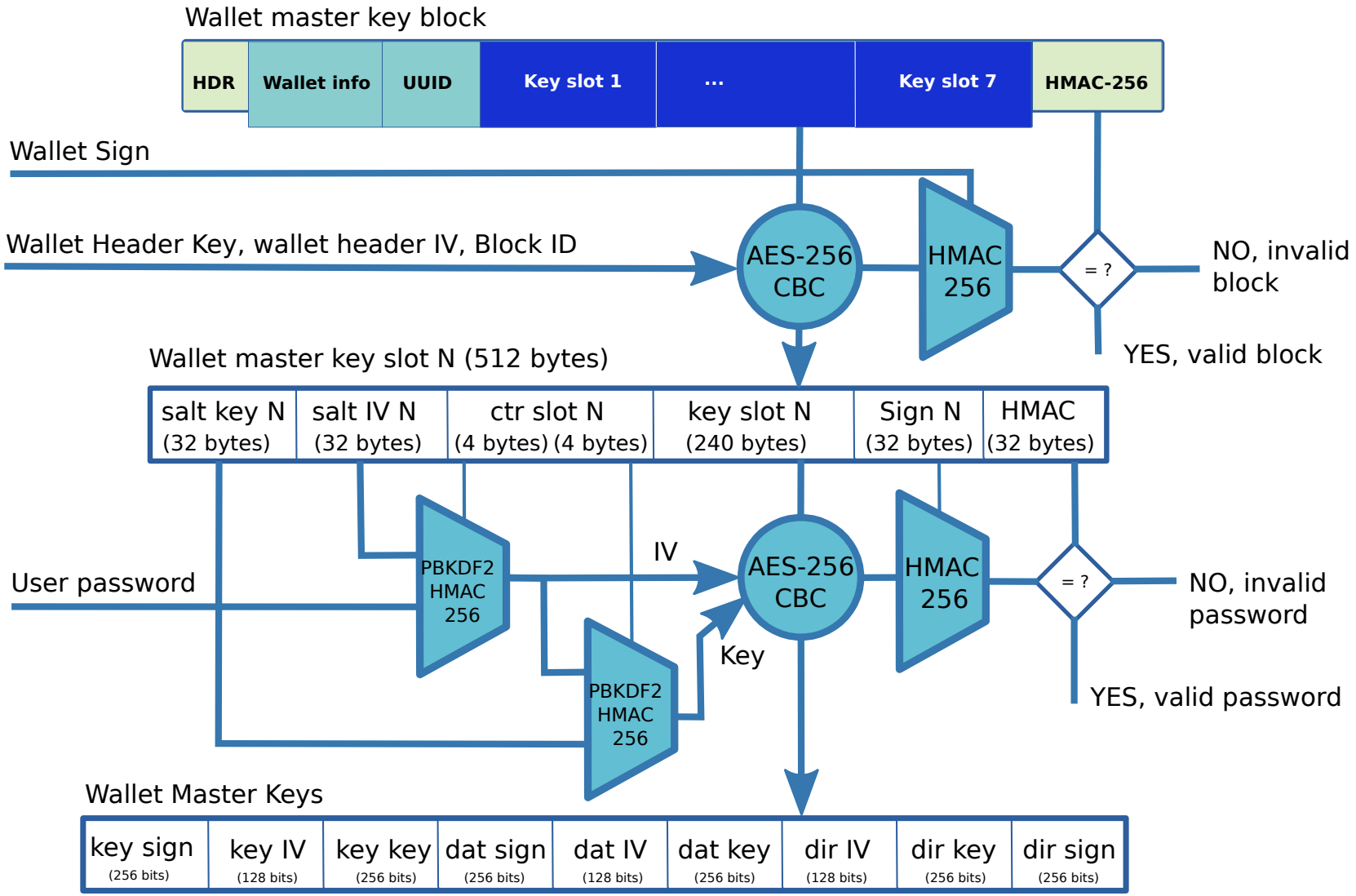


Data block



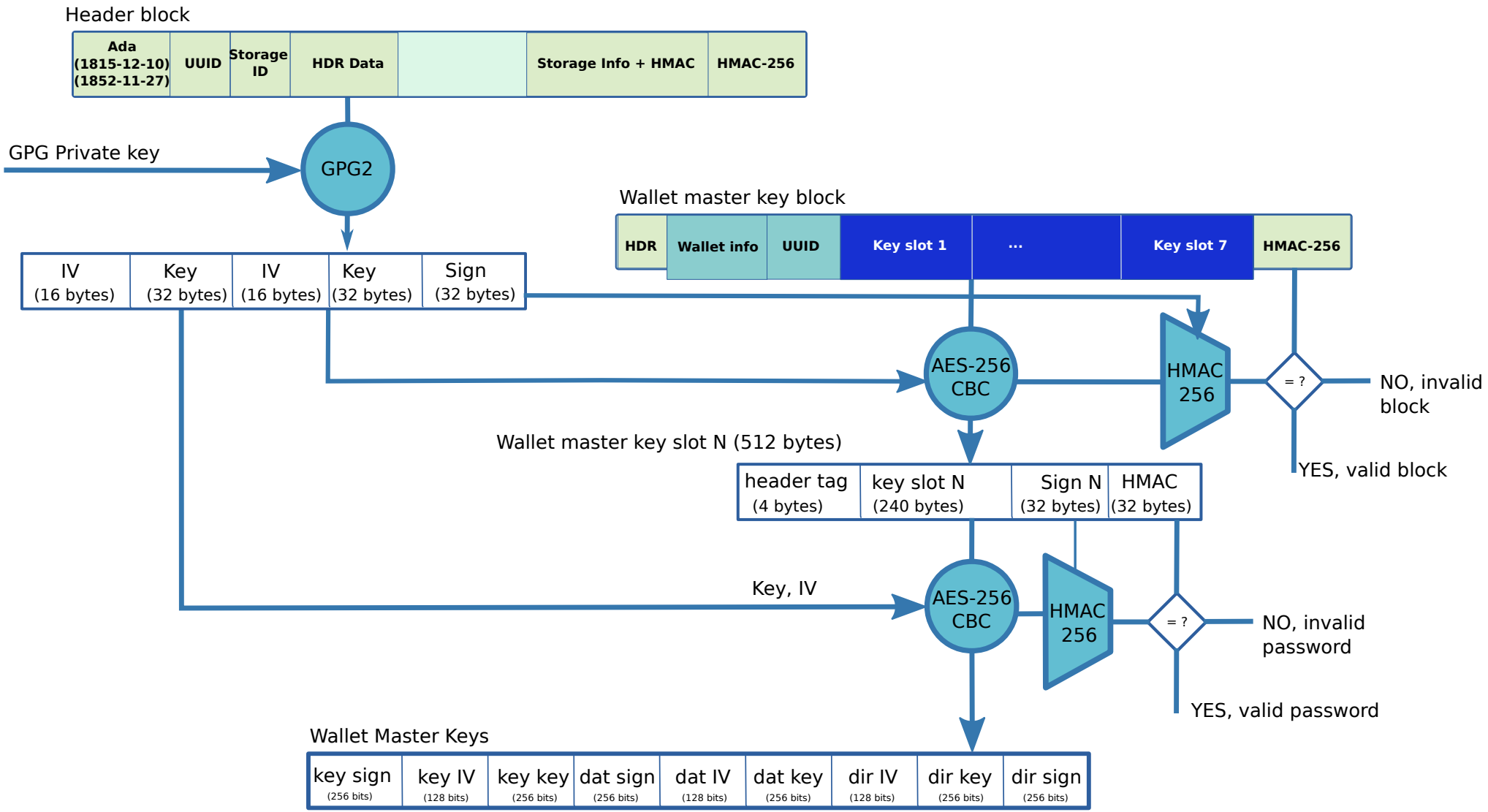
https://github.com/stcarrez/ada-keystore

Ada Keystore: key protection



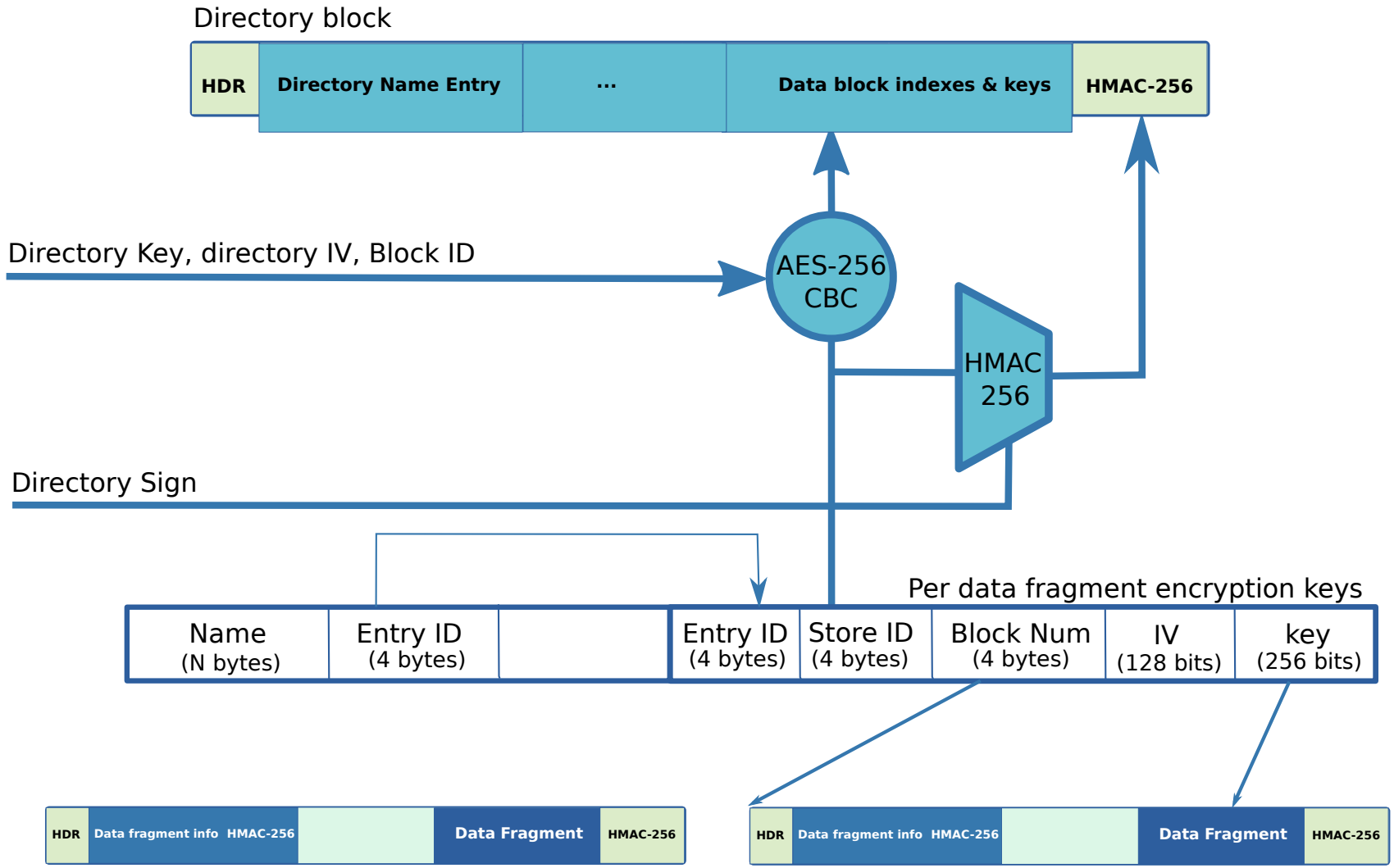
https://github.com/stcarrez/ada-keystore

Ada Keystore: GPG protection



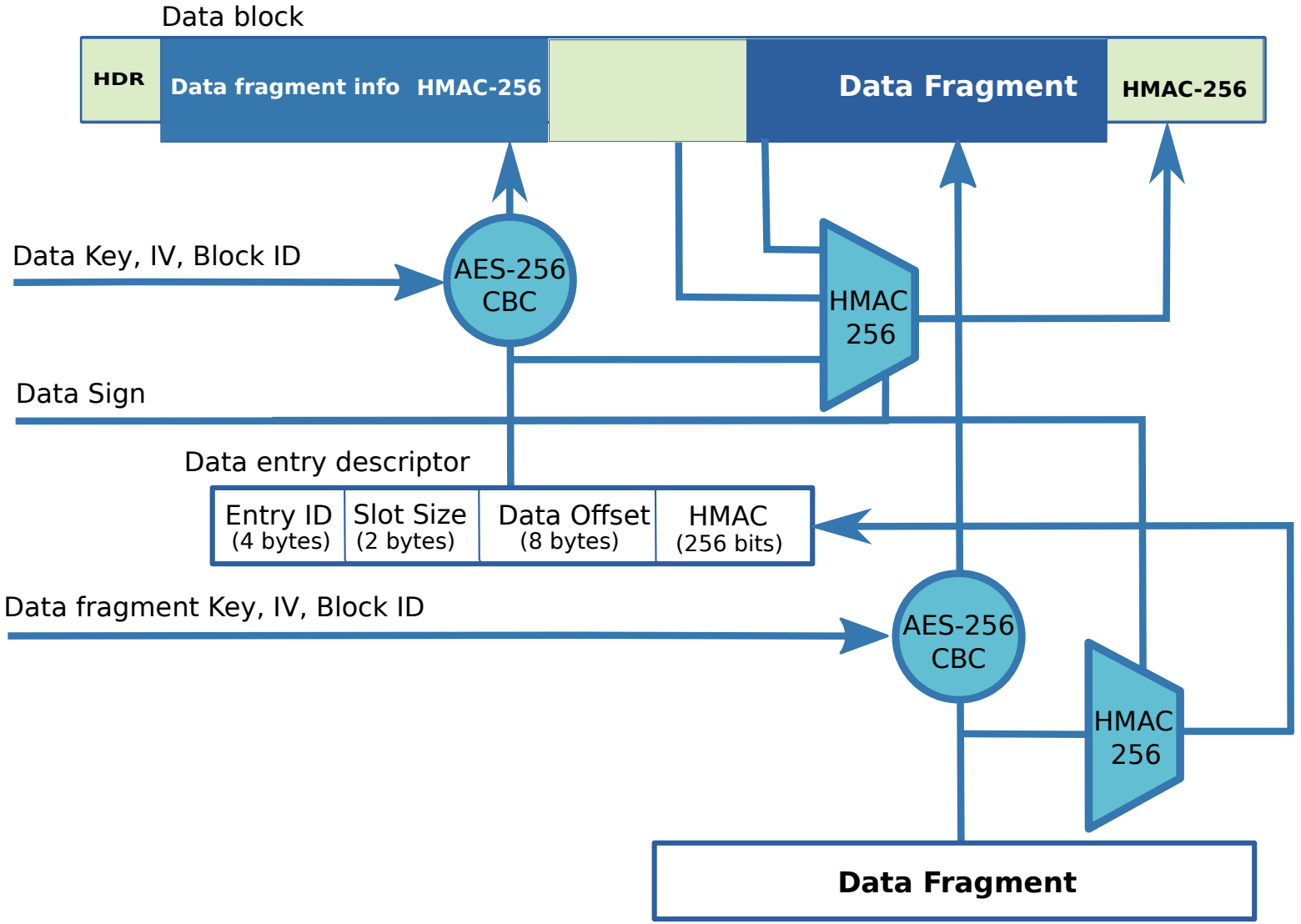
https://github.com/stcarrez/ada-keystore

Ada Keystore: directory encryption



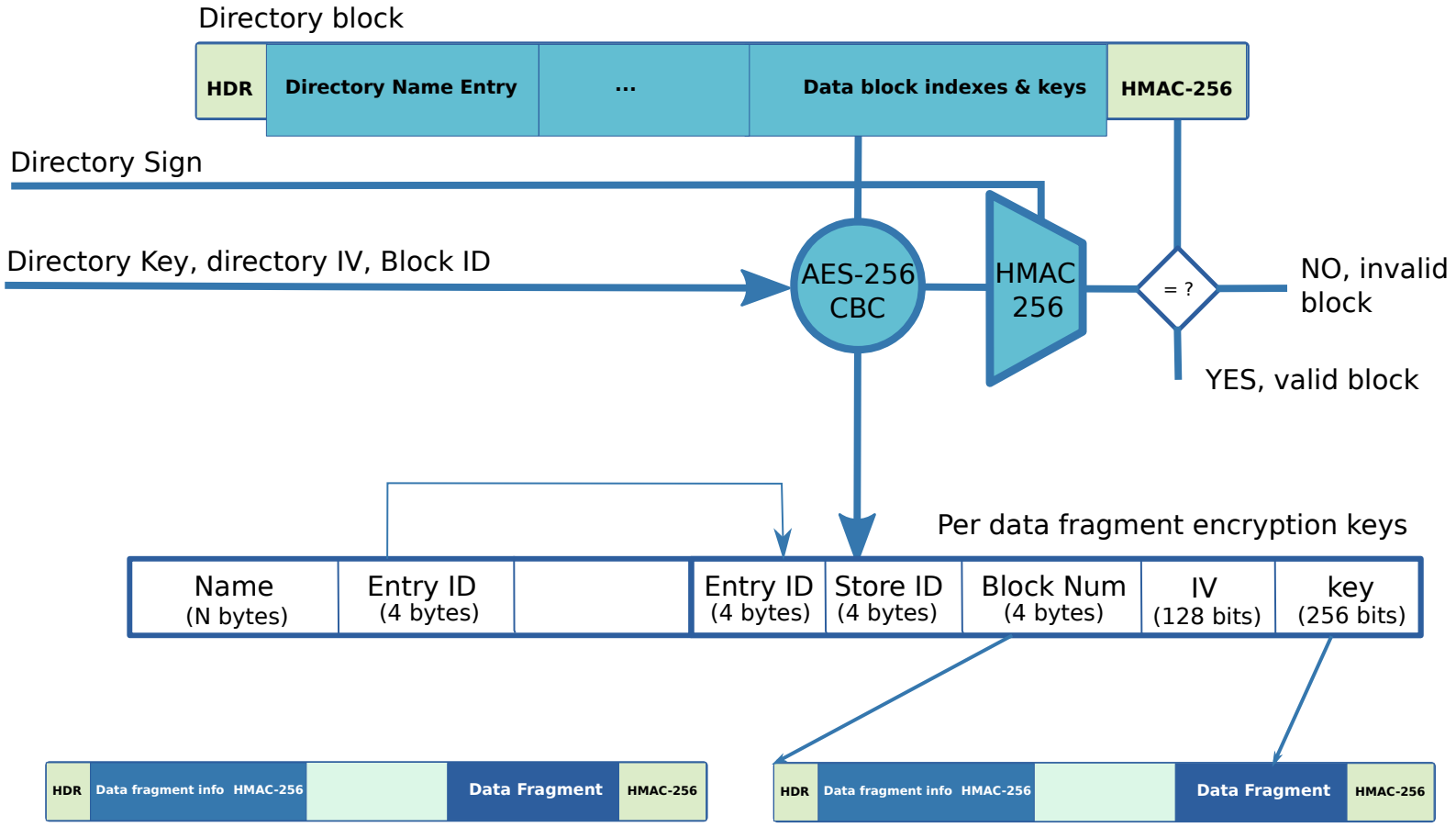
https://github.com/stcarrez/ada-keystore

Ada Keystore: data encryption



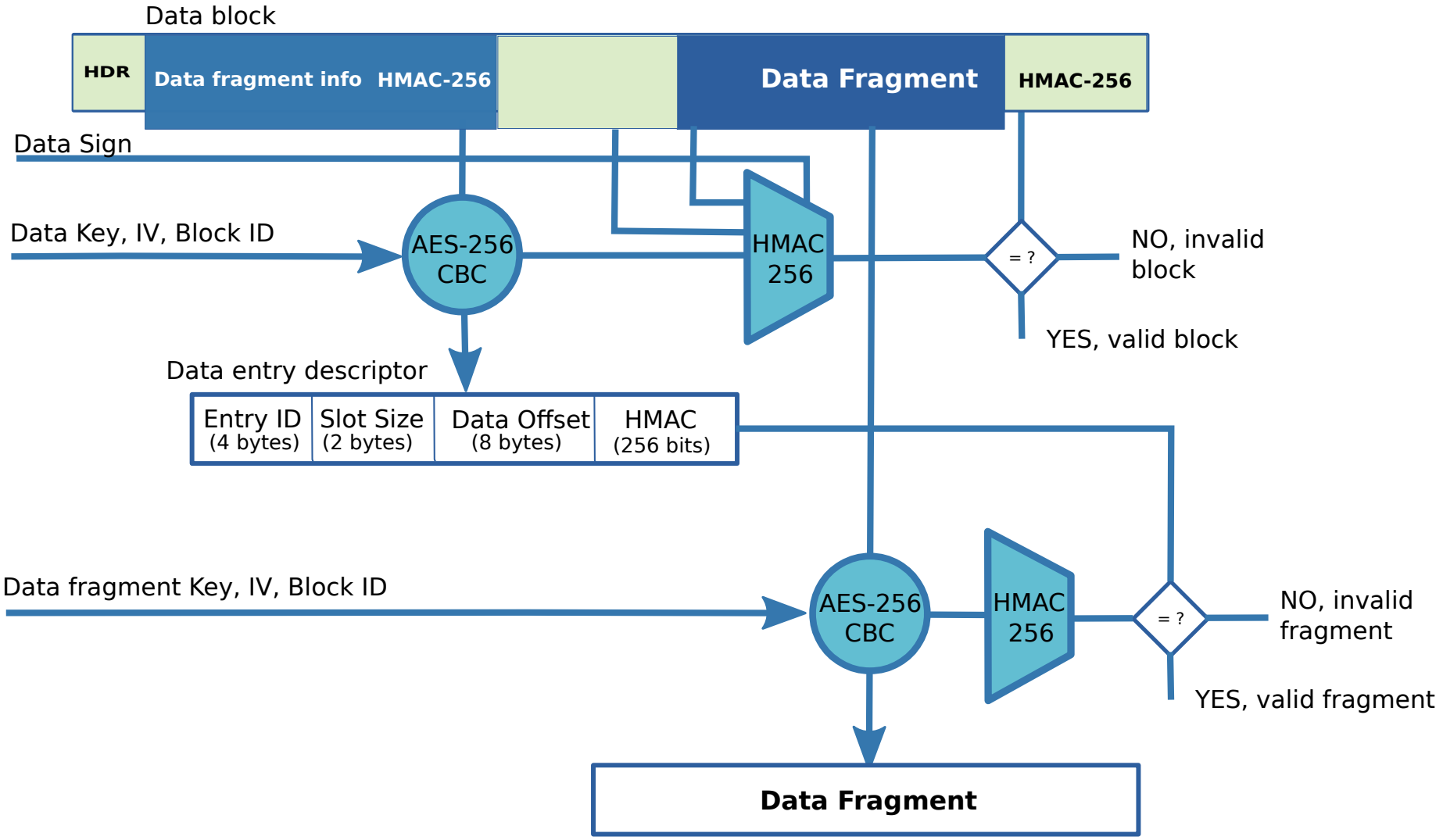
https://github.com/stcarrez/ada-keystore

Ada Keystore: directory decryption



https://github.com/stcarrez/ada-keystore

Ada Keystore: data decryption

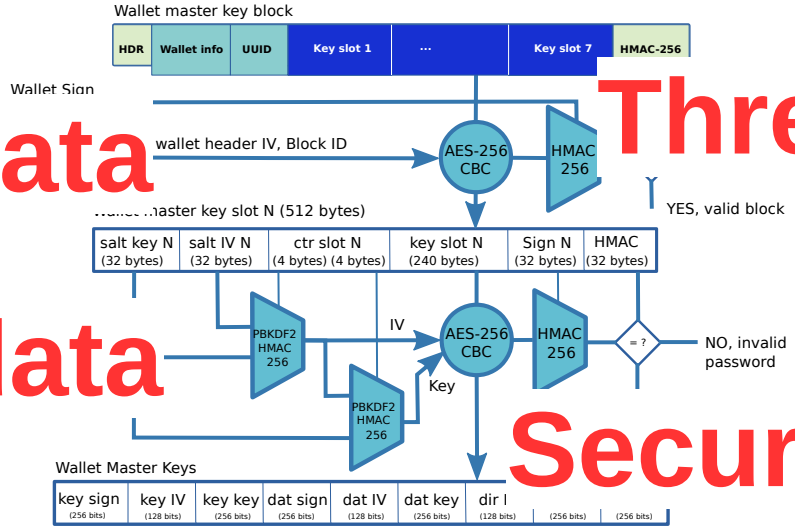


https://github.com/stcarrez/ada-keystore

Ada Keystore: more complexity

Insert data

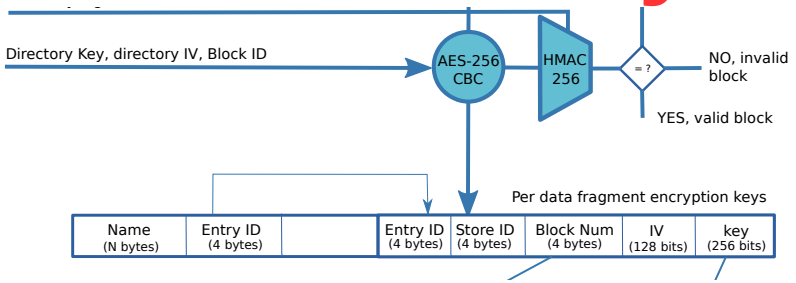
Thread safe API

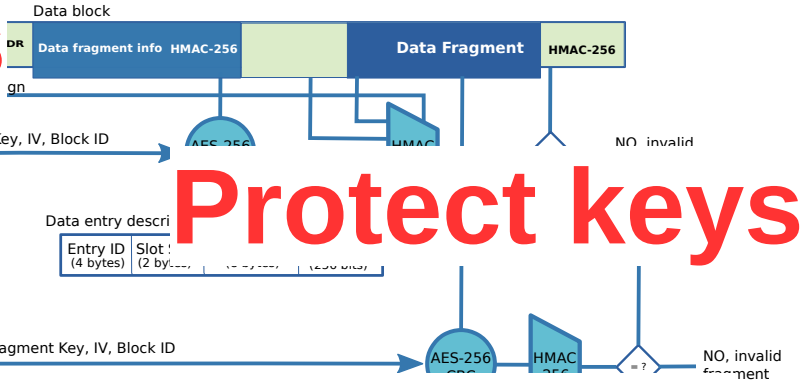


Update data

Secure data erase

Prevent security holes





Protect keys

Multi-task encryption/decryption

https://github.com/stcarrez/ada-keystore

Ada benefit : limited types

- Use limited record :
  - Encryption keys cannot be copied
  - Secret key content visible only to AES operations
- Use Ada.Finalization.Limited_Controlled :
  - Erase encryption keys when object is released

```
type Secret_Key (Length : Key_Length) is limited private;
```

Ada benefit : subtypes

- Use subtype to constraint index computation :
  - Detect index errors when they are computed not when we access the data

```
subtype Buffer_Size is Stream_Element_Offset range 0 .. 4064;
subtype Block_Index is Stream_Element_Offset range 1 .. 4064;
subtype Block_Type is Stream_Element_Array (Block_Index);
```

```
Key_Pos := Key_Header_Pos - Key_Slot_Size (Iterator.Key_Count);
```

➡ **better to raise Constraint_Error here**

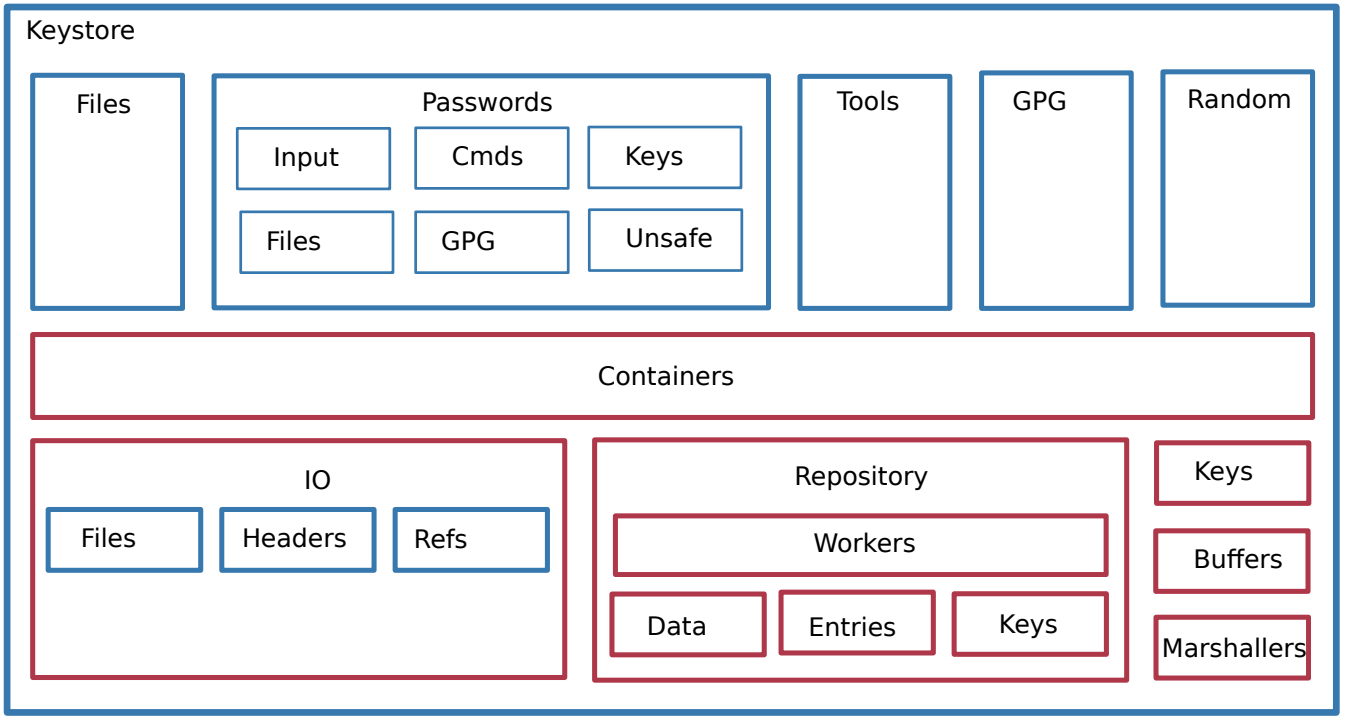
```
Buf.Data (Key_Pos .. Key_Pos + Key_Size - 1) := ...;
```

➡ **bounds still verified**

https://github.com/stcarrez/ada-keystore

Ada benefit : private child package

- Use private child package :
  - Restrict scope of operations
  - Helps in refactoring decisions



https://github.com/stcarrez/ada-keystore

Ada benefit : precondition

- Use Preconditions :
  - Nice for API constraints
  - Helpful to detect internal errors earlier

```
function Contains
 (Container : in Wallet;
  Name      : in String) return Boolean is abstract with
    Pre'Class => Container.Is_Open;


procedure Put_Unsigned_32
 (Into  : in out Marshaller;
  Value : in Interfaces.Unsigned_32) with
    Pre => Into.Pos <= Block_Type'Last - 4;
```

https://github.com/stcarrez/ada-keystore

Ada benefit : postcondition

- Use Postconditions :
  - Nice for API clarification
  - More complex to write

```
function Get_Header
 (From  : in out Marshaller) return Unsigned_32 with
     Post => From.Pos = Block_Type'First + 3;

procedure Get_Keys
 (From : in Key_Provider;
  Key  : out Secret_Key;
  IV   : out Secret_Key;
  Sign : out Secret_Key) is abstract with
  Post'Class => Key.Length = 32 and IV.Length = 16
     and Sign.Length = 32;
```

https://github.com/stcarrez/ada-keystore

Ada benefit : protected types

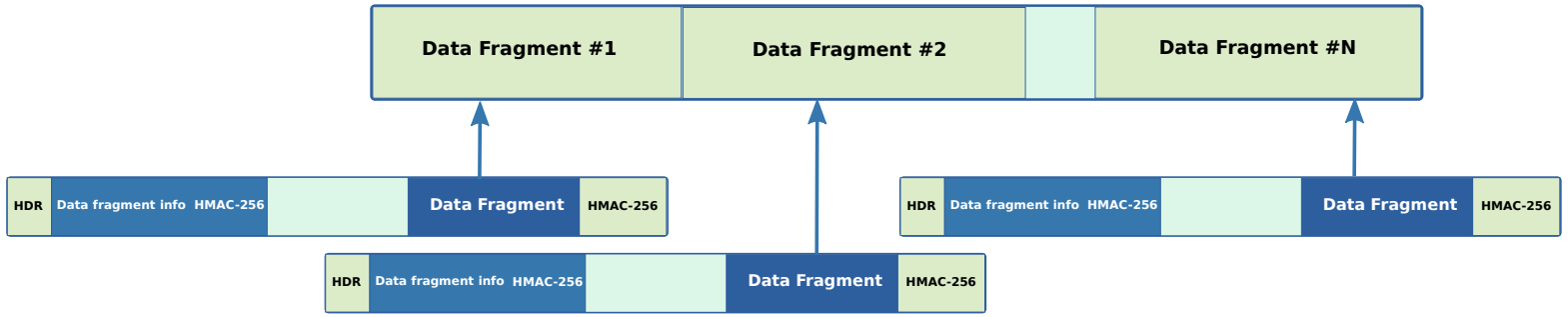
- Use protected types :
  - Concurrent access to keystore objects
  - Holds de keystore internal data structures

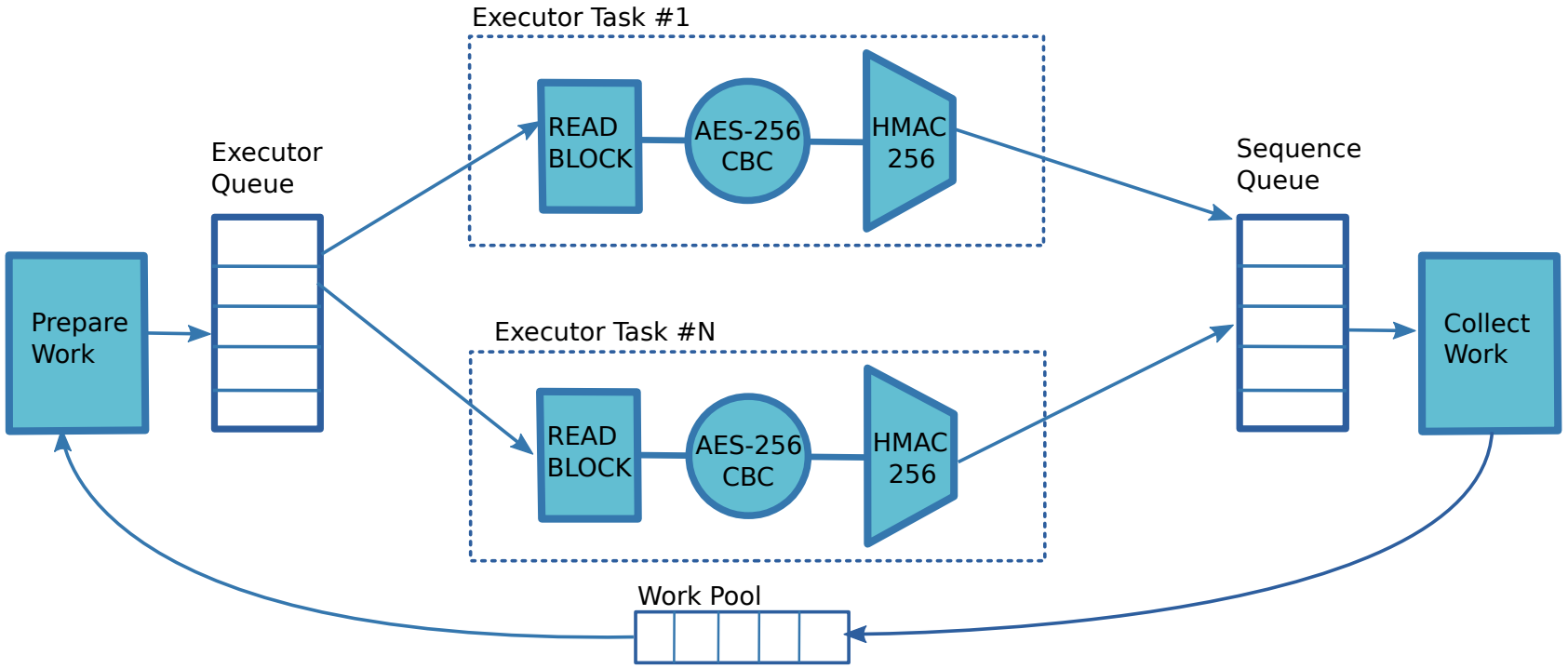
```
protected type Wallet_Container is
   procedure Open (...);
   procedure Create (...);
   procedure Add (...);
   ...
end Wallet_Container;
```

Ada benefit : tasks

- Use tasks for encryption/decryption process:
  - parallelize encryption and decryption
  - allow to configure the number of tasks
  - pre-allocation of tasks

Ada Keystore Work Queues





https://github.com/stcarrez/ada-keystore

Conclusion

- Writing a secure keystore is hard :
  - insert, update, remove data entries that look random
  - don’t leak secret keys
- Project succeeded thanks to Ada :
  - many errors prevented by the language
  - no buffer overflow, runtime error detection
- Secure storage made simple for Ada :
  - simple Ada API (Open, Create, Add, Get, Remove)
  - https://ada-keystore.readthedocs.io/en/latest/

Questions

?