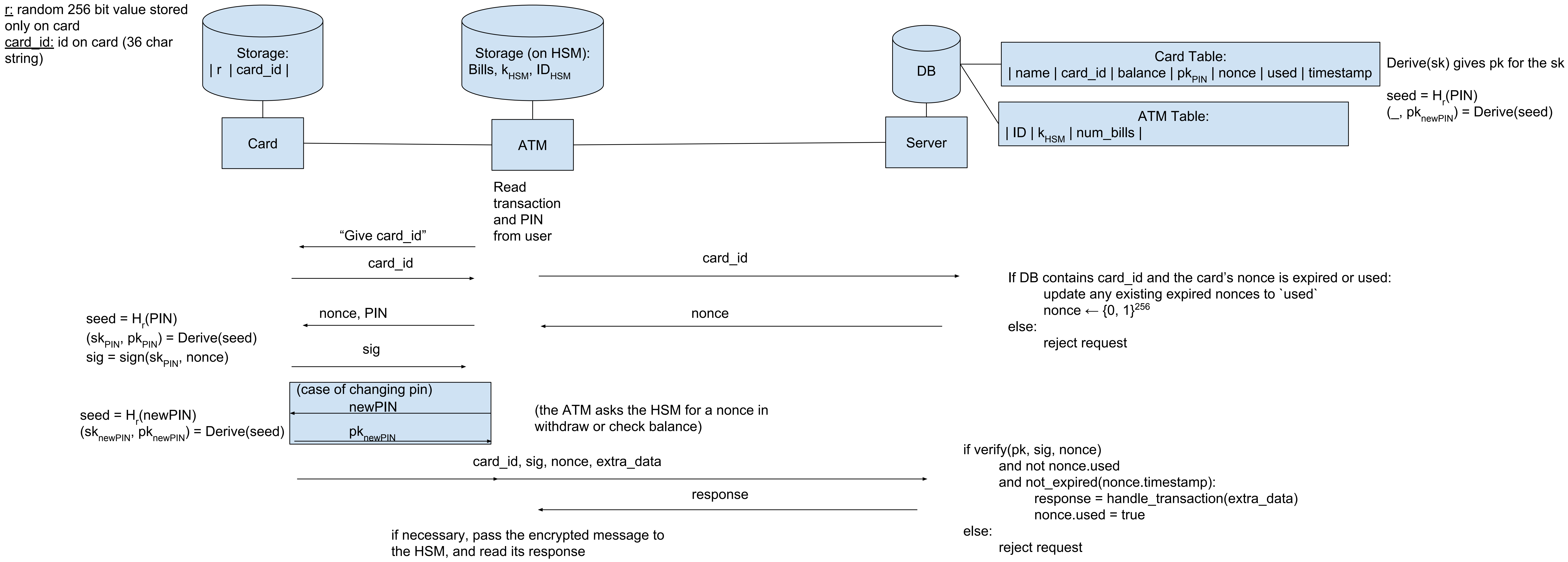

<u>r:</u> random 256 bit value stored only on card
<u>card_id:</u> id on card (36 char string)

Storage:
| r | card_id |
Card
Storage (on HSM):
Bills, k_{HSM}, ID_{HSM}
ATM
DB
Server
Card Table:
| name | card_id | balance | pk_{PIN} | nonce | used | timestamp
ATM Table:
| ID | k_{HSM} | num_bills |

Derive(sk) gives pk for the sk

seed = H_r(PIN)
(_, pk_{newPIN}) = Derive(seed)

Read transaction and PIN from user

```
If DB contains card_id and the card's nonce is expired or used:
    update any existing expired nonces to `used`
    nonce ← $\{0, 1\}^{256}$
else:
    reject request
```

seed = H_r(PIN)
(sk_{PIN}, pk_{PIN}) = Derive(seed)
sig = sign(sk_{PIN}, nonce)

seed = H_r(newPIN)
(sk_{newPIN}, pk_{newPIN}) = Derive(seed)

(the ATM asks the HSM for a nonce in withdraw or check balance)

```
if verify(pk, sig, nonce)
    and not nonce.used
    and not_expired(nonce.timestamp):
        response = handle_transaction(extra_data)
        nonce.used = true
else:
    reject request
```

if necessary, pass the encrypted message to the HSM, and read its response

<u>Contents of extra_data:</u>

Check Balance:	ID_{HSM}, $nonce_{HSM}$
Change PIN:	pk_{newPIN}
Withdraw:	ID_{HSM}, $nonce_{HSM}$, amount

<u>Contents of response:</u>

Check Balance:	"ERROR" or Enc_{kHSM}(CHECK_BALANCE | $nonce_{HSM}$ | balance)
Change PIN:	"OKAY" or "ERROR <some error message>"
Withdraw:	"ERROR" or Enc_{kHSM}(WITHDRAW | $nonce_{HSM}$ | amount)