



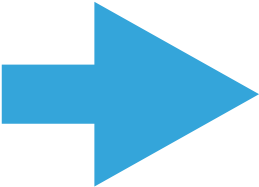




JENNER

+r1.id +r1.ver







DERIVEREORDERKEYS





**IV**



**IV**



**IV**

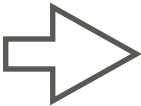
**MAKE SURE THE MASTER KEY HAS ENOUGH ENTROPY FOR DERIVED KEY AND DERIVED IV**





**Problem:** Use different keys for different records, only store master key.

**Solution:** Use key derivation to derive per-record keys.

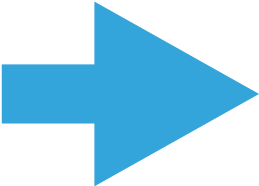


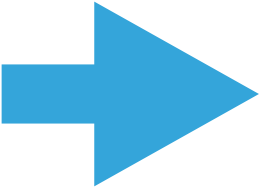












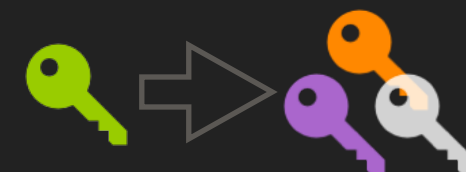




+r.n.id +r.n.ver

**IMPORTANT: NEVER USE THE SAME KEY/IV TO ENCRYPT DIFFERENT DATA**

## SOLUTIONS FOR DERIVING KEY(S)



```
// Input:
//  Master_key and
//  (DB) record_id target record DB id
// Output:
//  AES-Key and
//  salt for encrypting target record

// AES-Key and salt for target record. "||" concatenates
// AES-CBC uses 128 bit IV. AES-GCM uses a 96 bit IV
byte[32] keyAndIV = derive_key( master_key ||
                                record_id || record_version, 256 bit)

byte[16] derived_iv    = keyAndIV[0..15]
byte[16] derived_key   = keyAndIV[16..31]
```

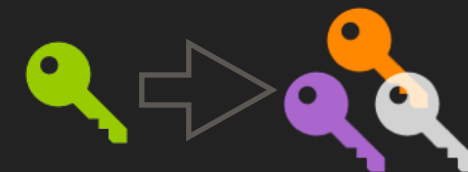
- ▶ `derive_key` needs an additional *installation specific* salt of  $\geq 128$  bit. PBKDF2 with HMAC sha256 is an example of `derive_key`, as is `sCrypt` or [argon2](#).
- ▶ Use same process for decryption.
- ▶ No need to store the *generated* IV value.

**IMPORTANT: NEVER USE THE SAME KEY/IV TO ENCRYPT DIFFERENT DATA**

**MAKE SURE THAT THE MASTER KEY HAS ENOUGH ENTROPY FOR DERIVED KEY AND DERIVED SALT**

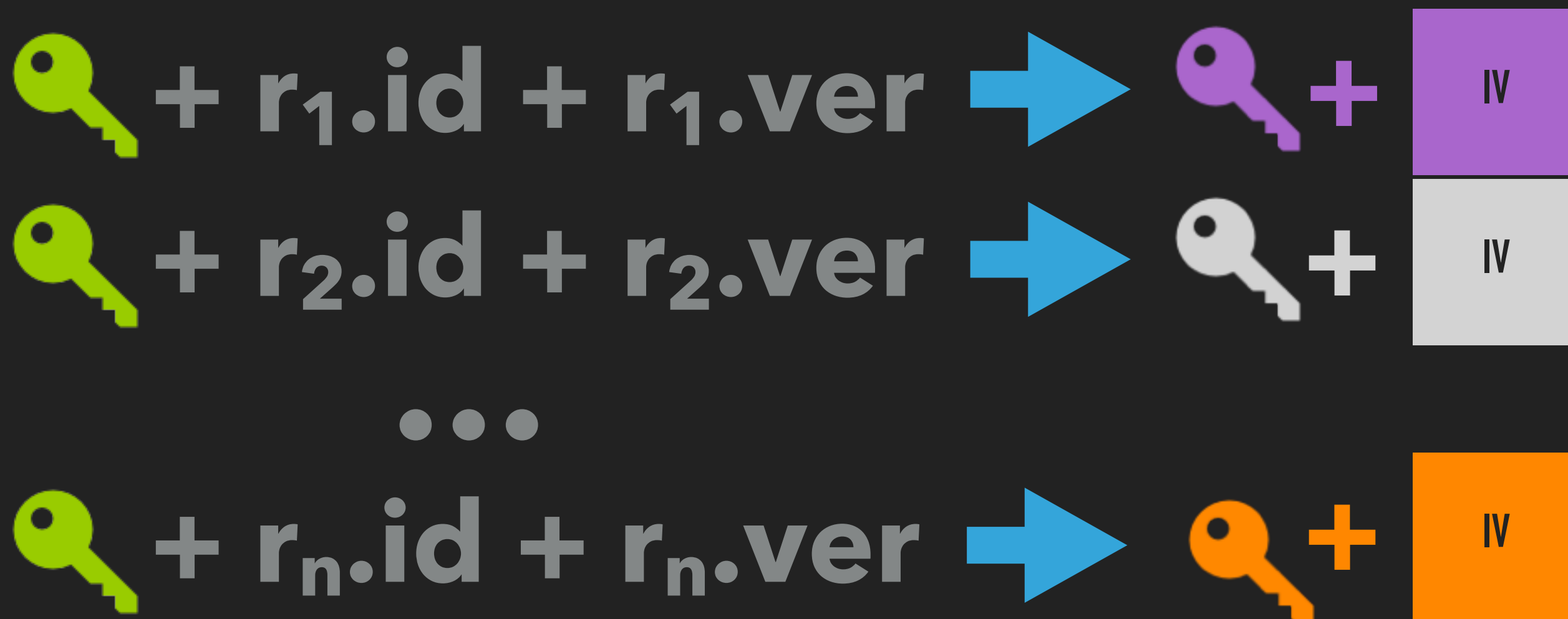


## DERIVE PER RECORD KEYS



**Problem:** Use different keys for different records, only store master key.

**Solution:** Use key derivation to derive per-record keys.



**IMPORTANT: NEVER USE THE SAME KEY/IV TO ENCRYPT DIFFERENT DATA**

**MAKE SURE THAT THE MASTER KEY HAS ENOUGH ENTROPY FOR DERIVED KEY AND DERIVED IV**