Encrypting Firebird databases

Alex Peshkov

Firebird Foundation
IbPhoenix
2016

Encrypting Firebird databases

- History of a feature
  - Existed (but closed with #ifdef) since IB 6.01.
    - No support for initial database encryption
    - Crypt key expected to be passed from client in DPB
  - Implemented from scratch in FB 3
    - SQL support for database encryption / decryption
    - Encryption on the fly using background server thread
    - Flexible crypt keys control including separate KeyHolder plugin

Encrypting Firebird databases

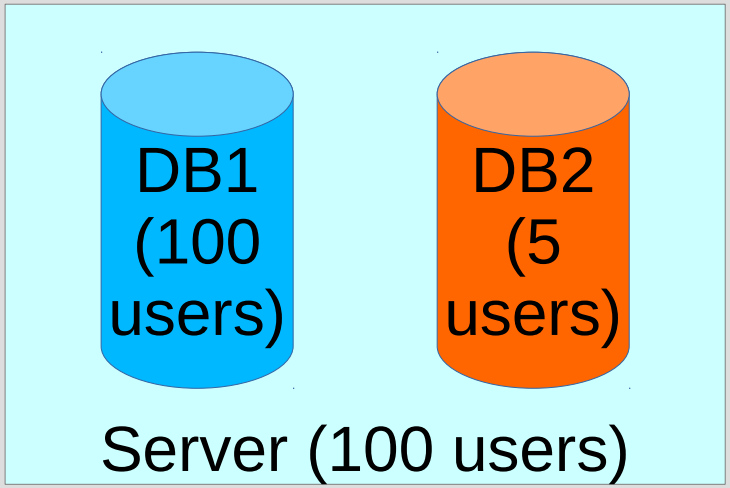
- When NOT use:
  - Protect database file from being copied over network
- Correct solution
  - Tune access rights in your network

  - Share \\server\c with full control
  - Everyone – Administrator or same access rights

  - Windows trusted authentication, mapping Domain Admins => SYSDBA (FB 2.1)
    - Everyone – SYSDBA?

Encrypting Firebird databases

- When NOT use:
  - Let only some users attach to specific database
- Correct solution
  - Use multiple security databases
- Pre-FB3 (Encrypt?)



Encrypting Firebird databases

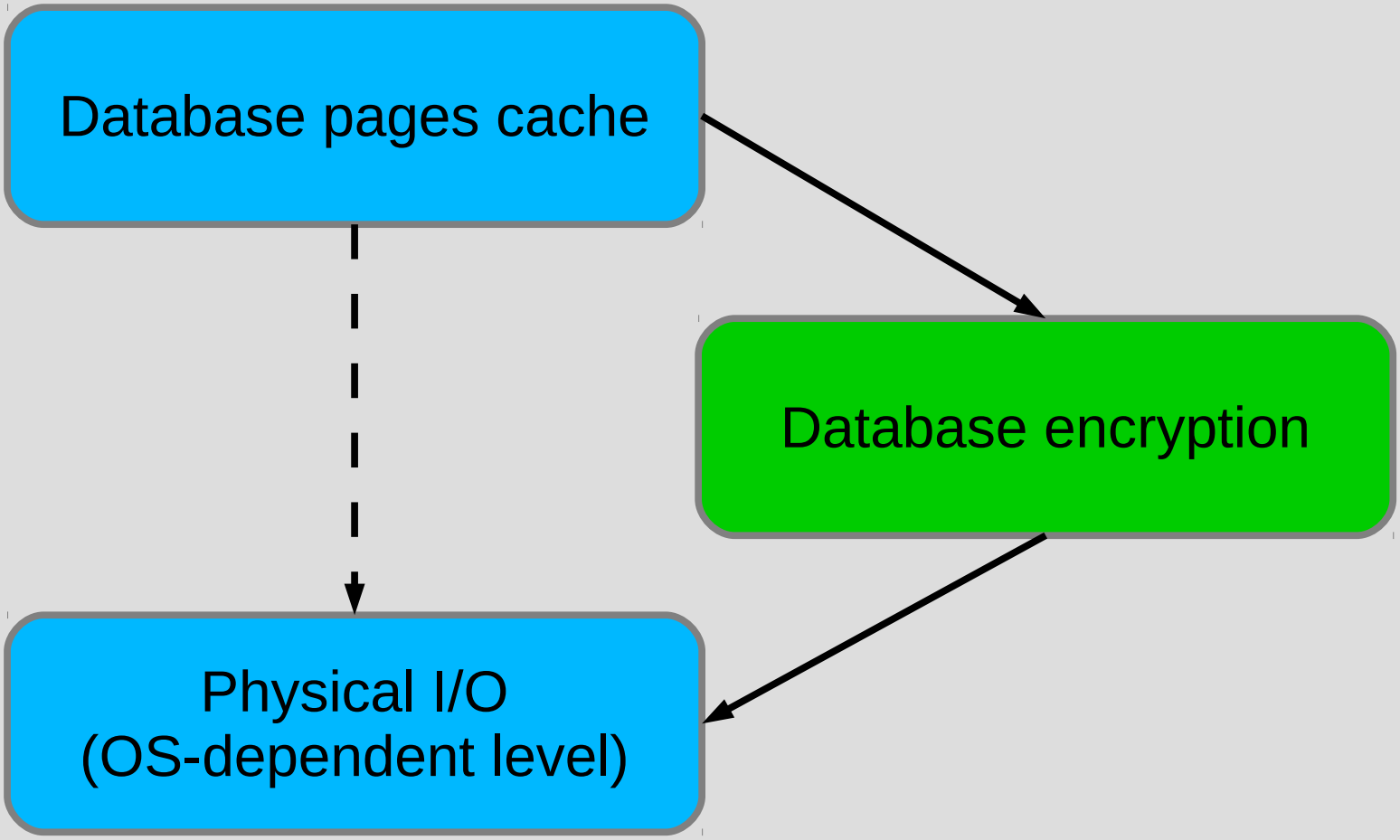
- When is it useful?
- Protecting databases distributed for fee
  - Filled with important data
  - With important business logic in metadata
- Protecting databases from being physically stolen (HDD or the entire server)

Encrypting Firebird databases

- Compared with use of encrypted disk
  - Useless when distributing databases
  - Requires offline period to copy database to encrypted disk
- What do we crypt
  - Data, blob and index pages (except header)
  - Subsidiary pages (PIP, TIP, etc.) left not encrypted
  - Key correctness is checked using hash providing zero-knowledge about a key
  - Sensitive data (hash, encryption flags, etc.) are protected by additional encrypted checksum

Encrypting Firebird databases

- When are pages encrypted / decrypted?

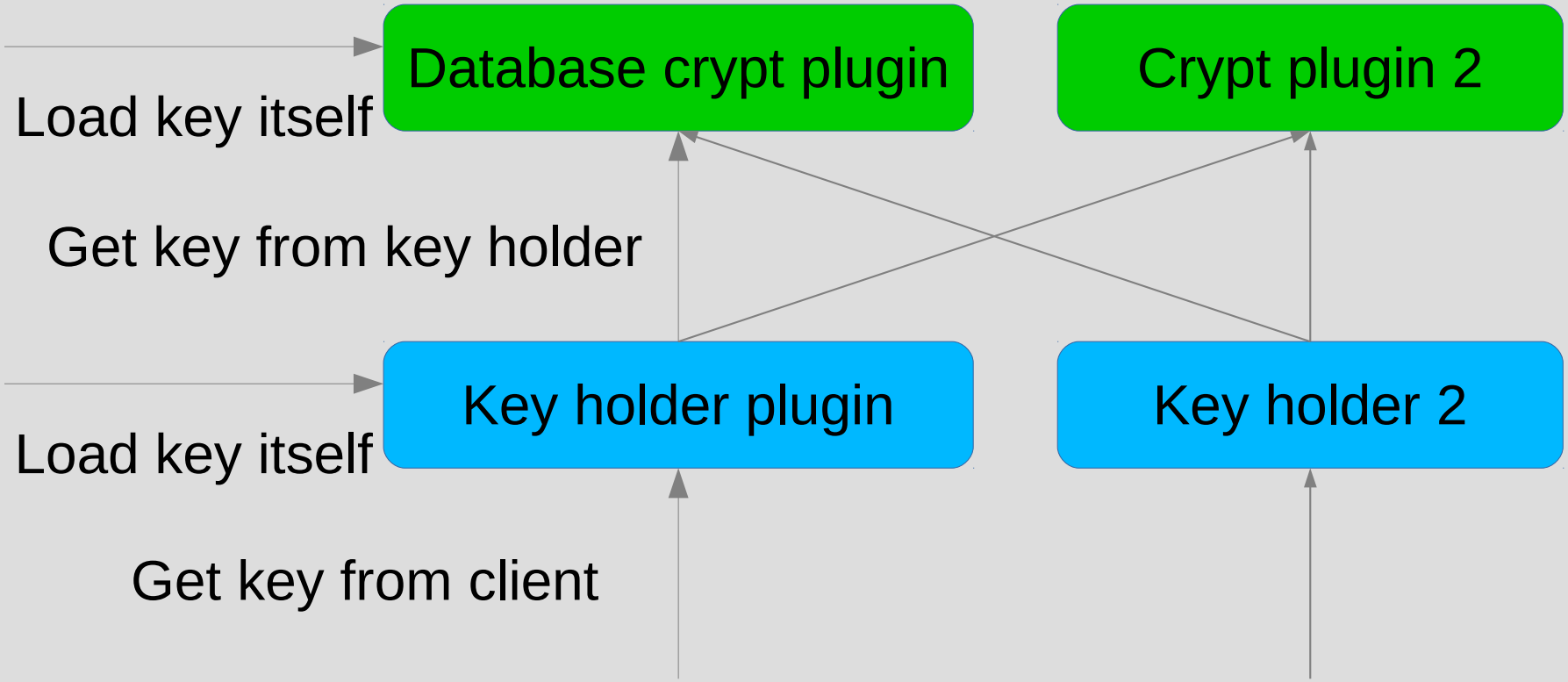


Encrypting Firebird databases

- How crypt key can be stored?
- Databases distributed for fee
  - In special client software
    - Database should be accessible only from that software
  - Support “developers mode”
- Databases protected from physical loss
  - In some secret place (host in security department)
    - Database should be accessible from any client, including generic purpose tools

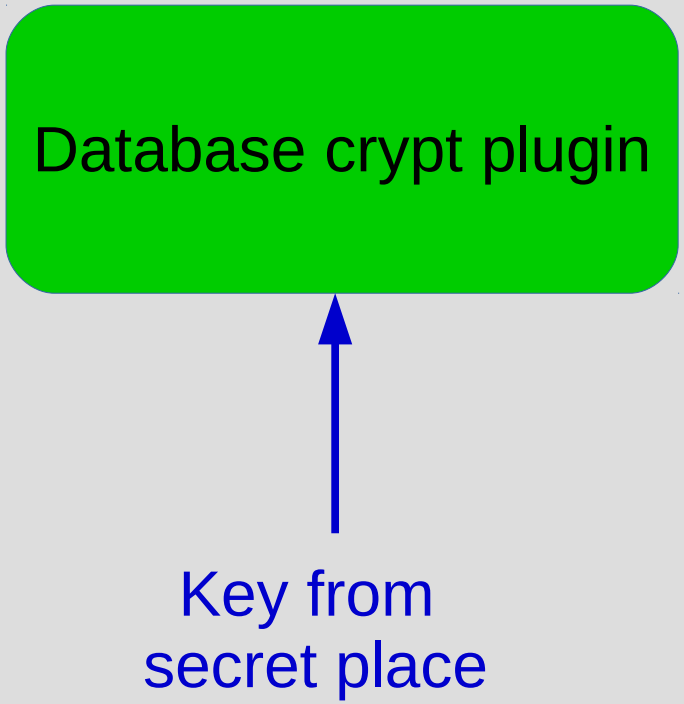
Encrypting Firebird databases

- How to store key?



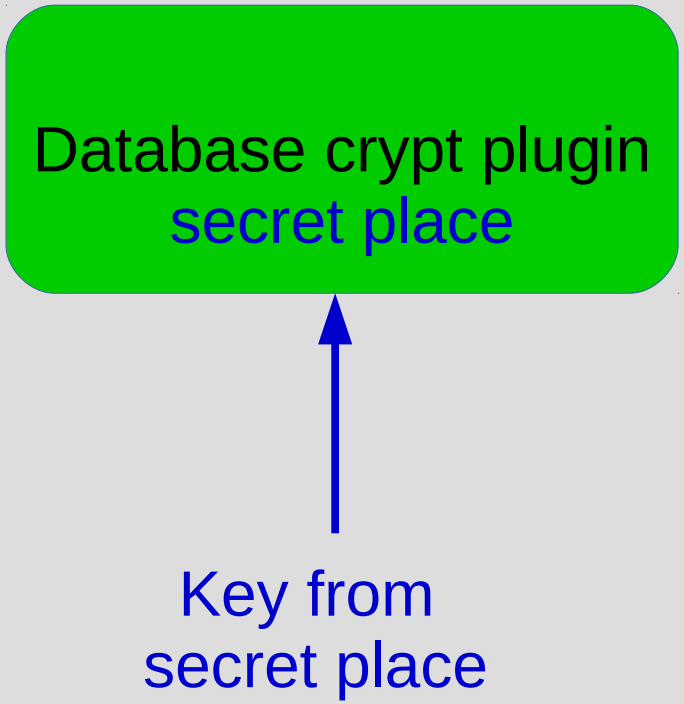
Encrypting Firebird databases

- Possible key sources



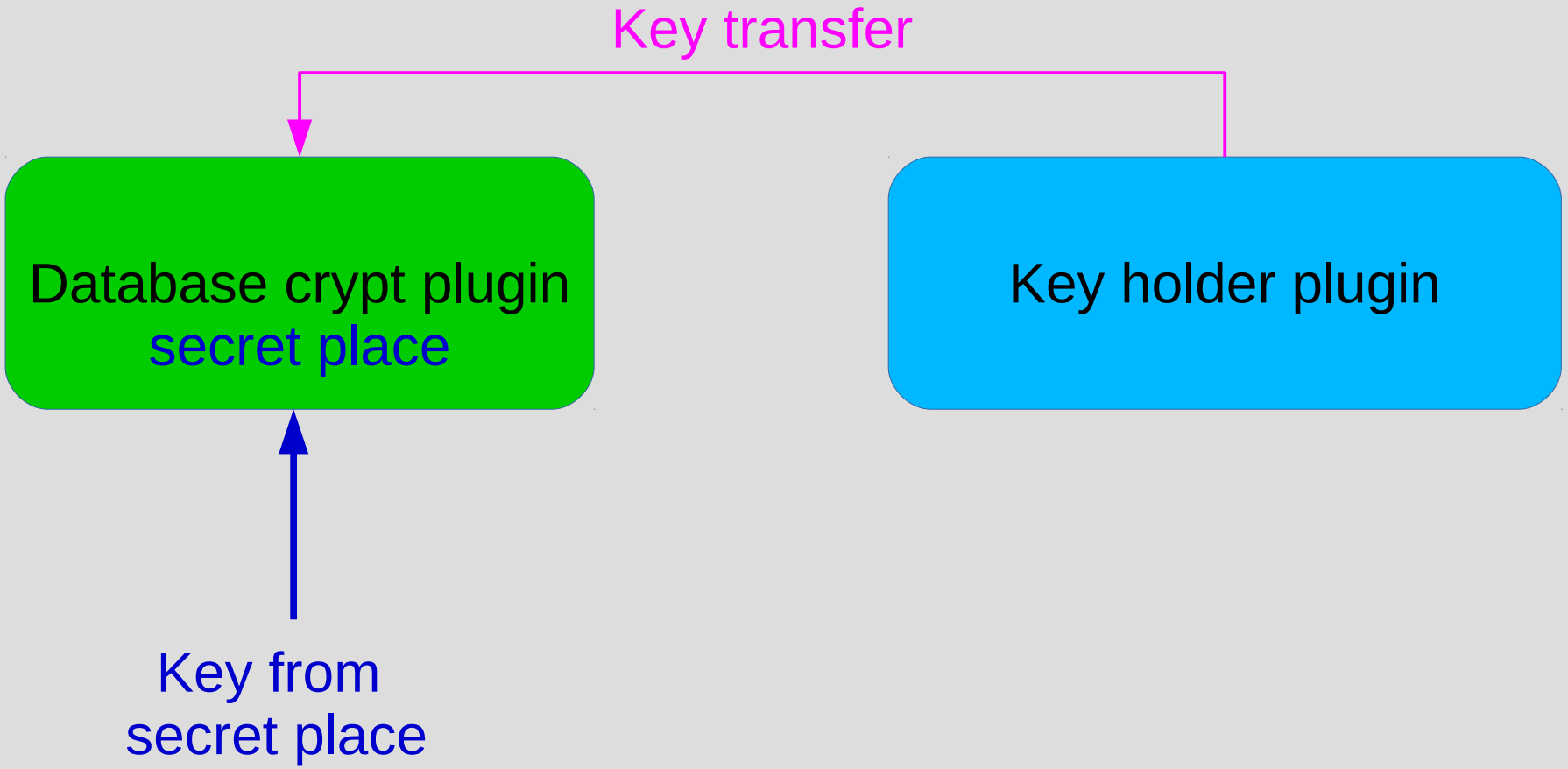
Encrypting Firebird databases

- Possible key sources



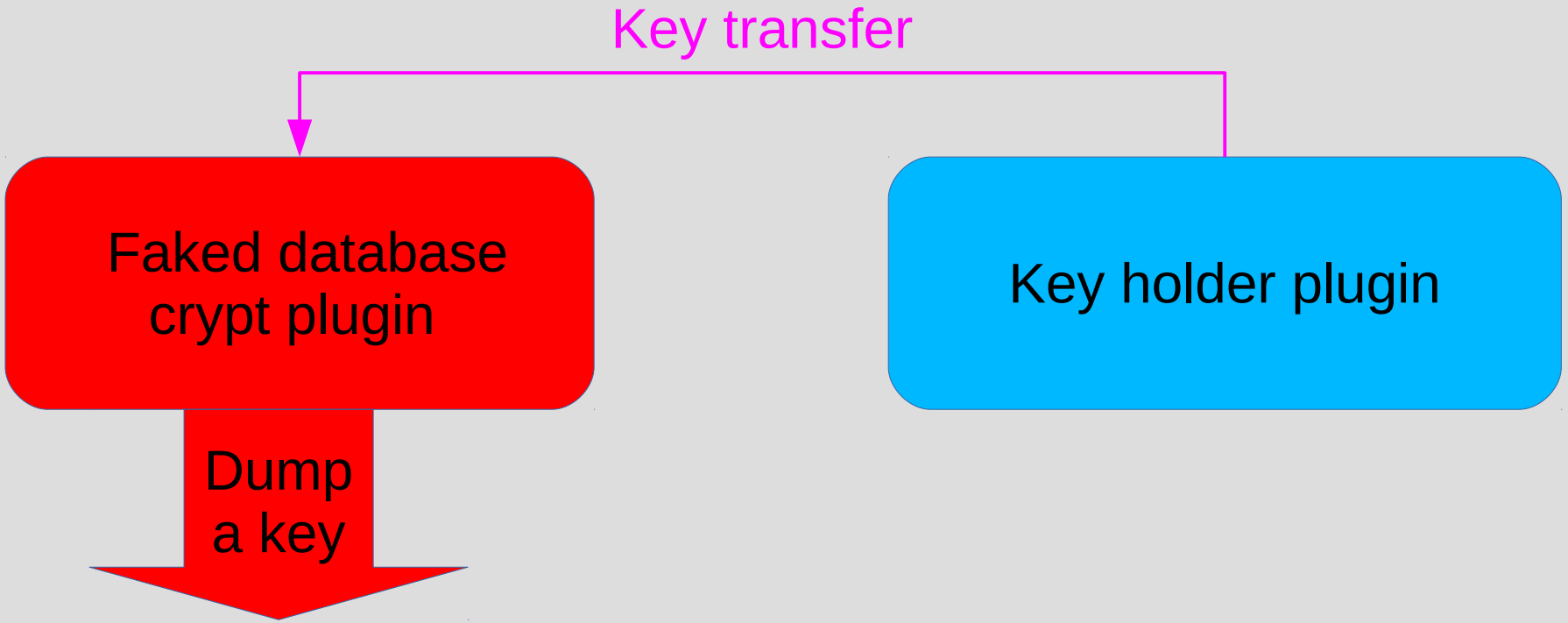
Encrypting Firebird databases

- Possible key sources



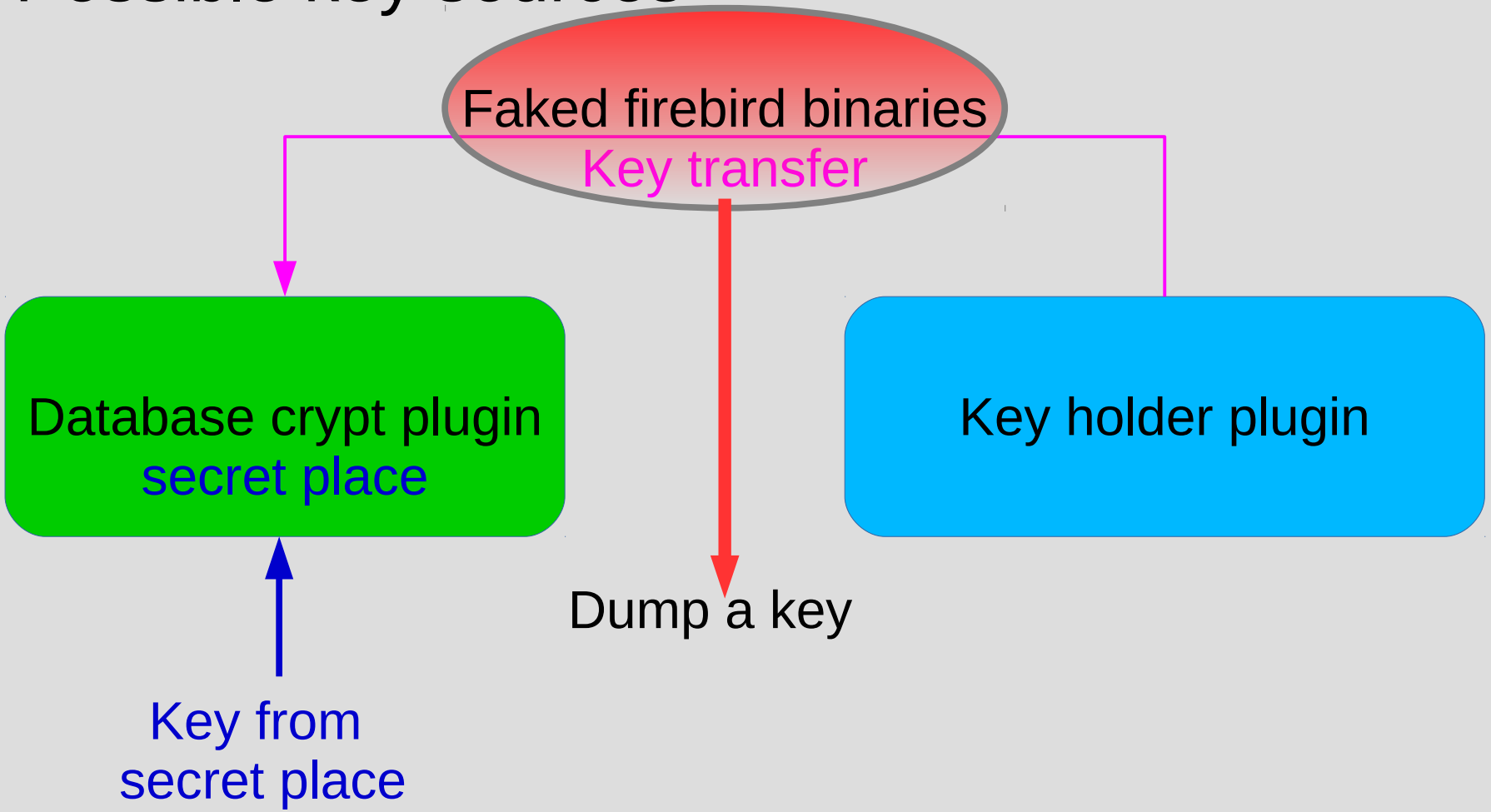
Encrypting Firebird databases

- Possible key sources



Encrypting Firebird databases

- Possible key sources

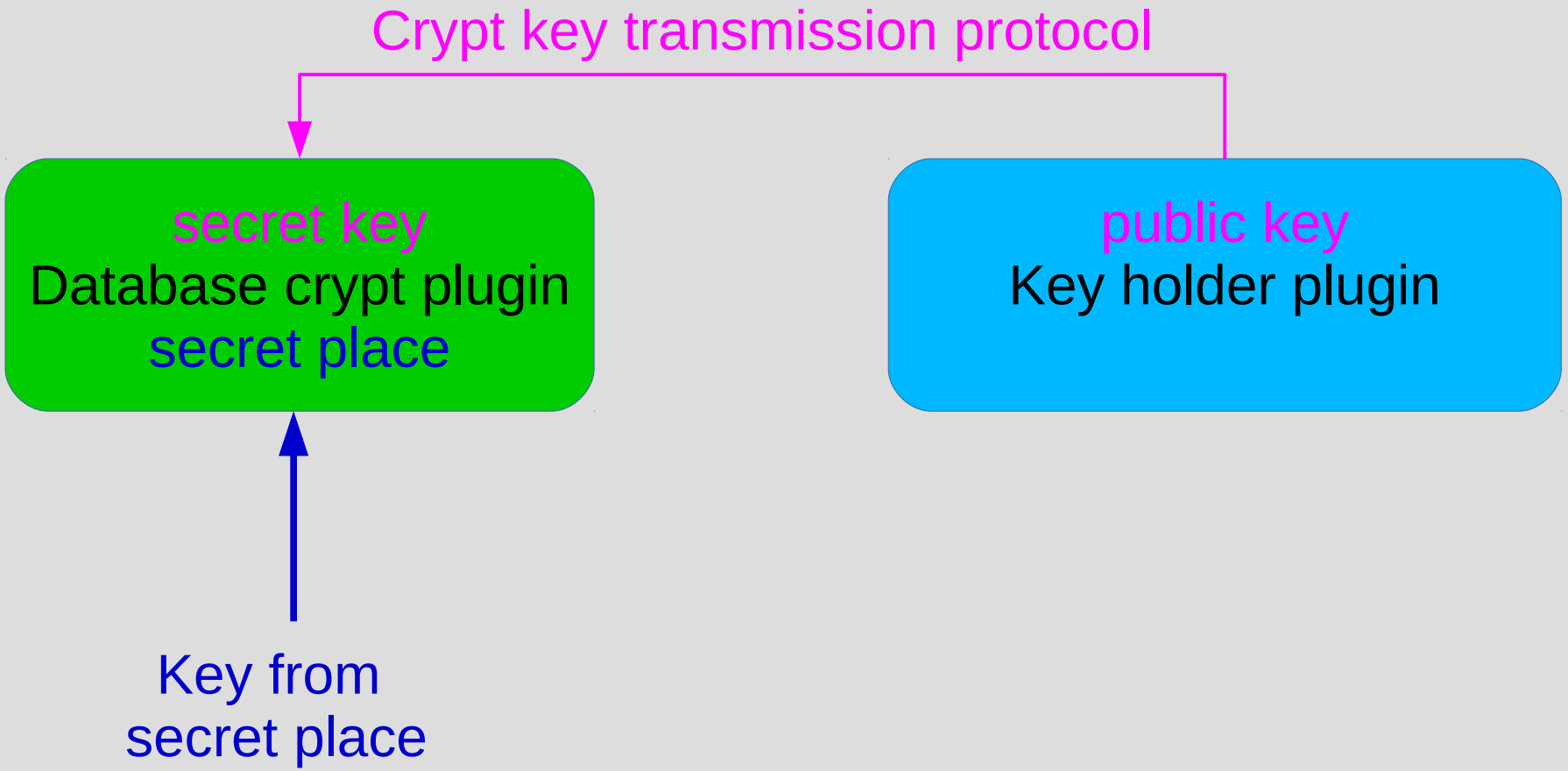


Encrypting Firebird databases

- Approximate authorization protocol
  - Crypt plugin => Key holder:
    - Send me a key
  - Key holder:
    - Encrypts a key
  - Key holder => Crypt plugin:
    - Encrypted key
  - Crypt plugin:
    - Decrypts a key
    - Ready to work

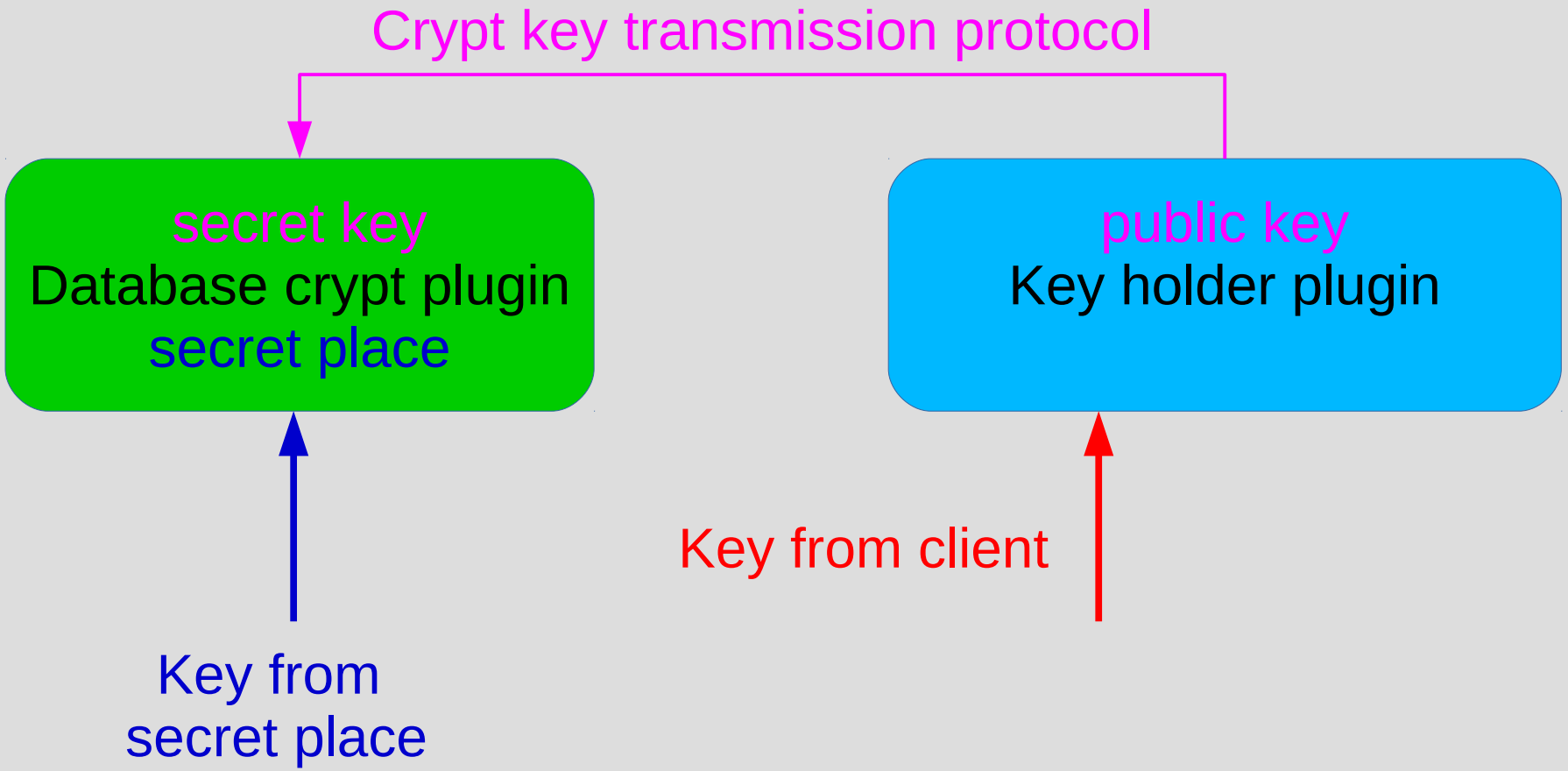
Encrypting Firebird databases

- Possible key sources



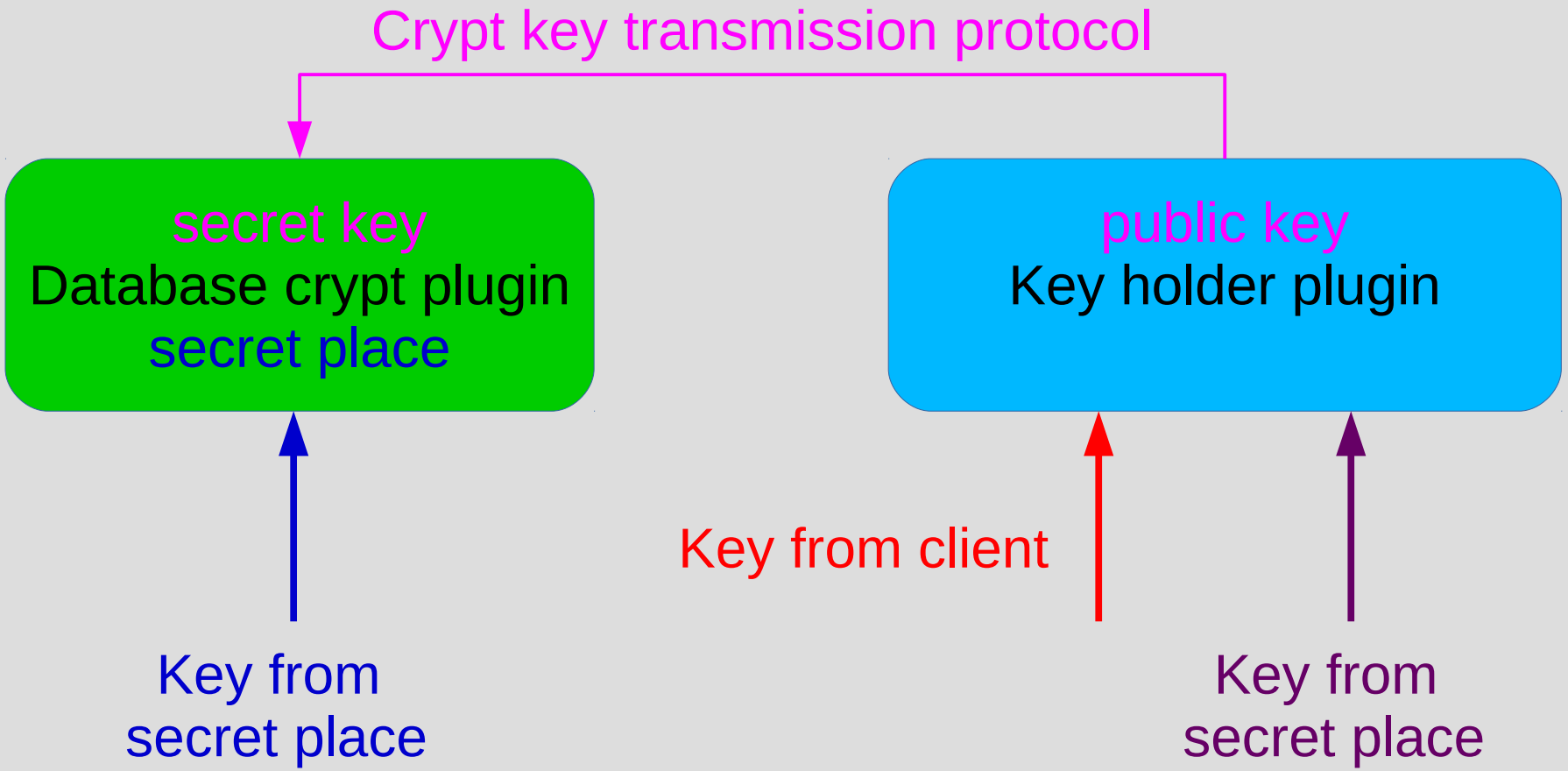
Encrypting Firebird databases

- Possible key sources



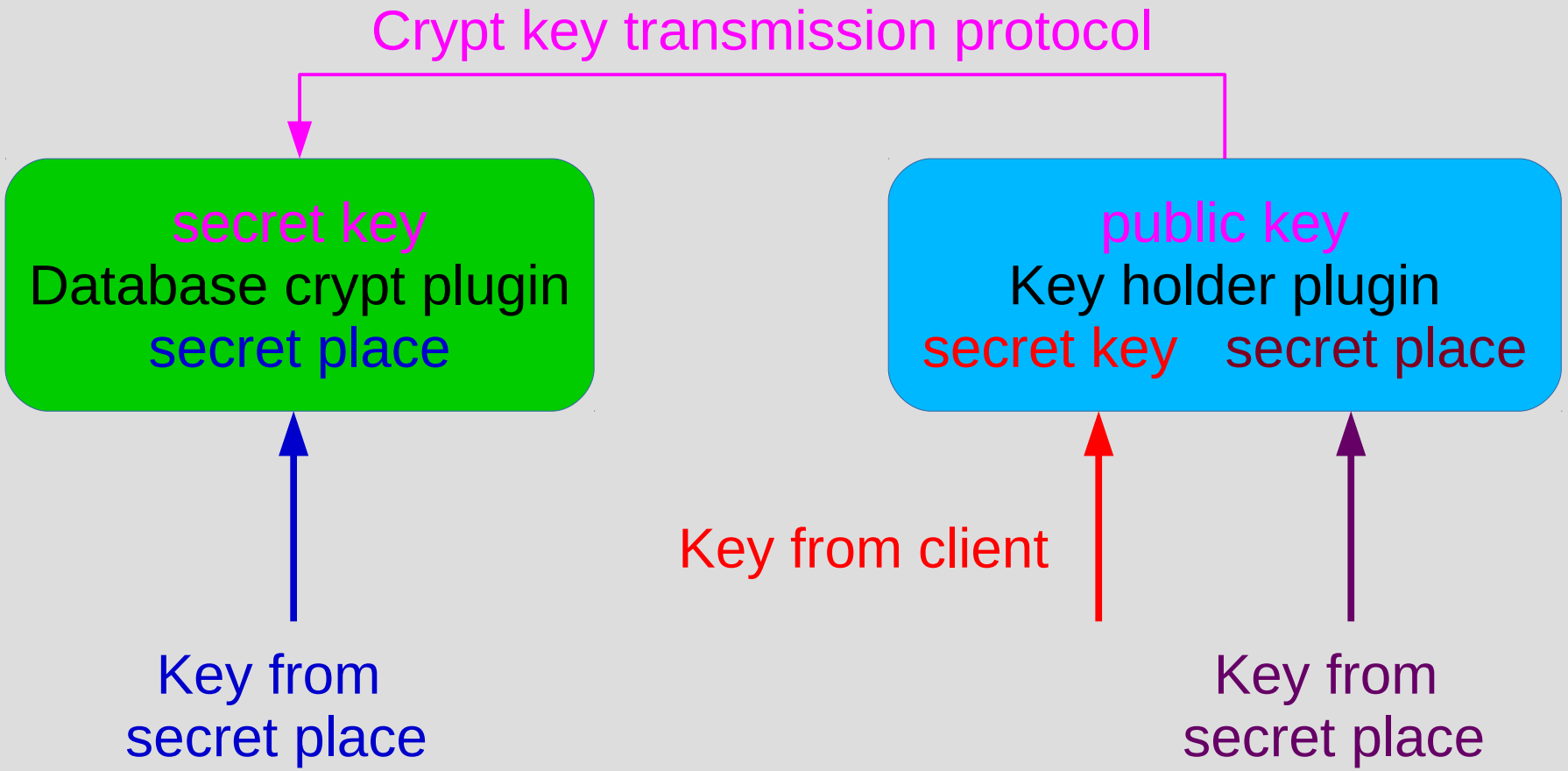
Encrypting Firebird databases

- Possible key sources



Encrypting Firebird databases

- Possible key sources



Encrypting Firebird databases

- Step 1 – select plugin to use
  - Not open source – problems with crypt keys
    - Write it yourself
    - Use trusted third party plugin

- Step 2 – install and check on database copy
  - Use SQL statement:

    Alter database encrypt with “PluginName”
  - Or:

    Alter database encrypt with “PluginName” key “Name”

  Meaning of key name is plugin-dependent

Encrypting Firebird databases

- Step 3 – backup !!!

- Step 4 – choose off-peak load period and encrypt database
  - Do not backup database during encryption!
  - Use monitoring tables or gstat (may be in services API) to monitor encryption progress

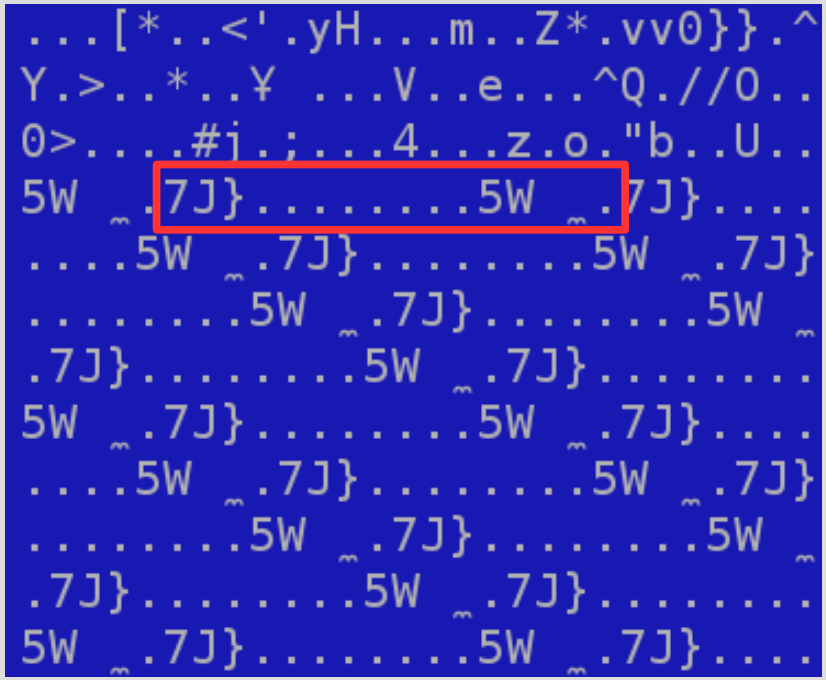
```
SQL: Select MON$CRYPT_PAGE * 100.0 /
  MON$PAGES as Percent from mon$database

gstat -e db_name
```

Encrypting Firebird databases

- Working with encrypted database
  - API fully functional
  - Utilities fully functional – except gstat
  - Limited gstat functionality – only -e / -h switches
  - Backup database
    - gbak: encrypt copy (file.gbak) manually
    - nbackup: needs full (level 0) copy after encryption

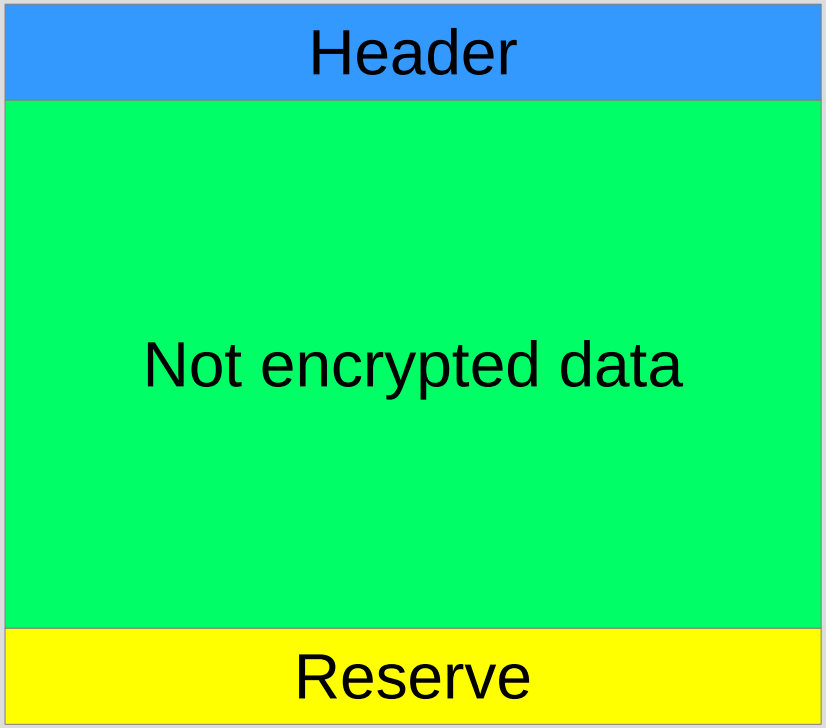
Encrypting Firebird databases

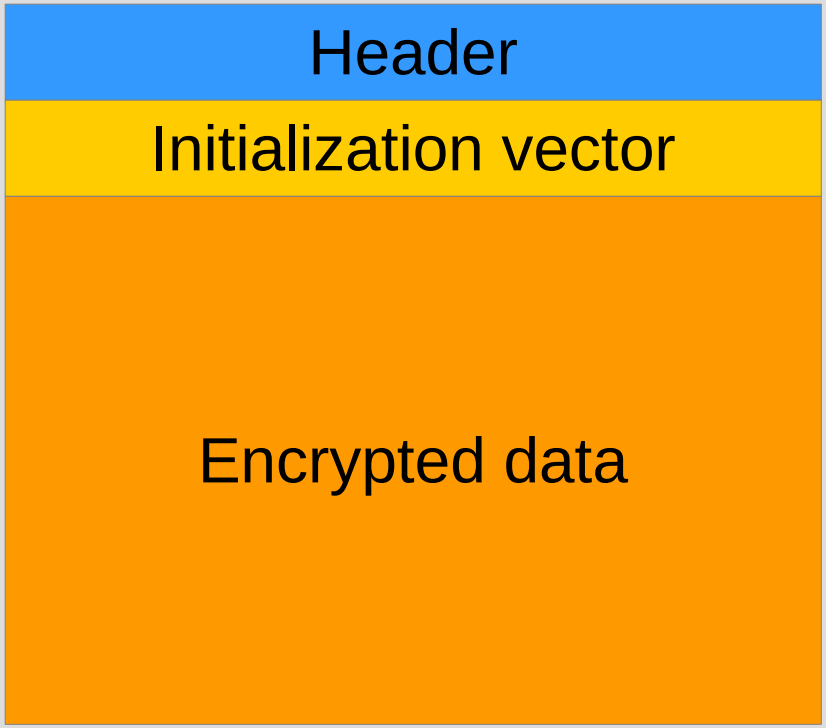
- Known issue
  - Encrypted size == initial size
  - Use of ECB mode in AES
  - Visible repeating sequences on some pages



Encrypting Firebird databases

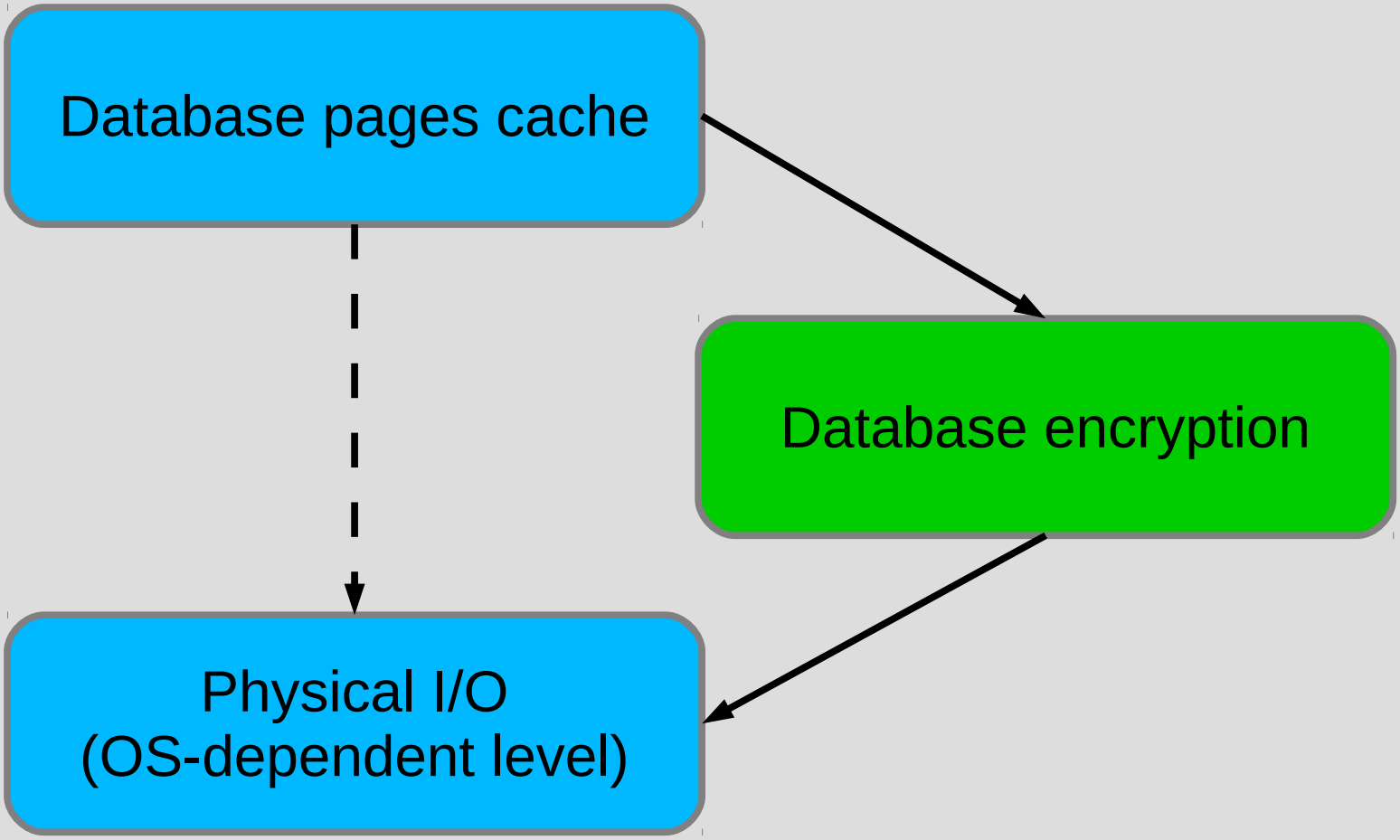
- Possible solutions
  - Use other cipher (RC4)
  - Reserve space on pages for IV at database creation time





Encrypting Firebird databases

- Performance



Encrypting Firebird databases

- Performance (desktop)
  - 8 CPU cores (AMD FX-8120)
  - RAM 8 Gb
  - Slow SATA
  - 4 connections, 1 minute (TPCC)
  - AES, using OpenSSL
  - Default cache (16 Mb < DB size)

(tpmC, TPC-C Throughput)

Forced writes	Not encrypted	Encrypted	Performance loss
On	984	740	25%
Off	27062	18453	32%

Encrypting Firebird databases

- Performance (desktop)
  - 8 CPU cores (AMD FX-8120)
  - RAM 8 Gb
  - Slow SATA
  - 4 connections, 1 minute (TPCC)
  - AES, using OpenSSL
  - Default cache (320 Mb > DB size)

(tpmC, TPC-C Throughput)

Forced writes	Not encrypted	Encrypted	Performance loss
On	1036	882	15%
Off	27793	19170	31%

Encrypting Firebird databases

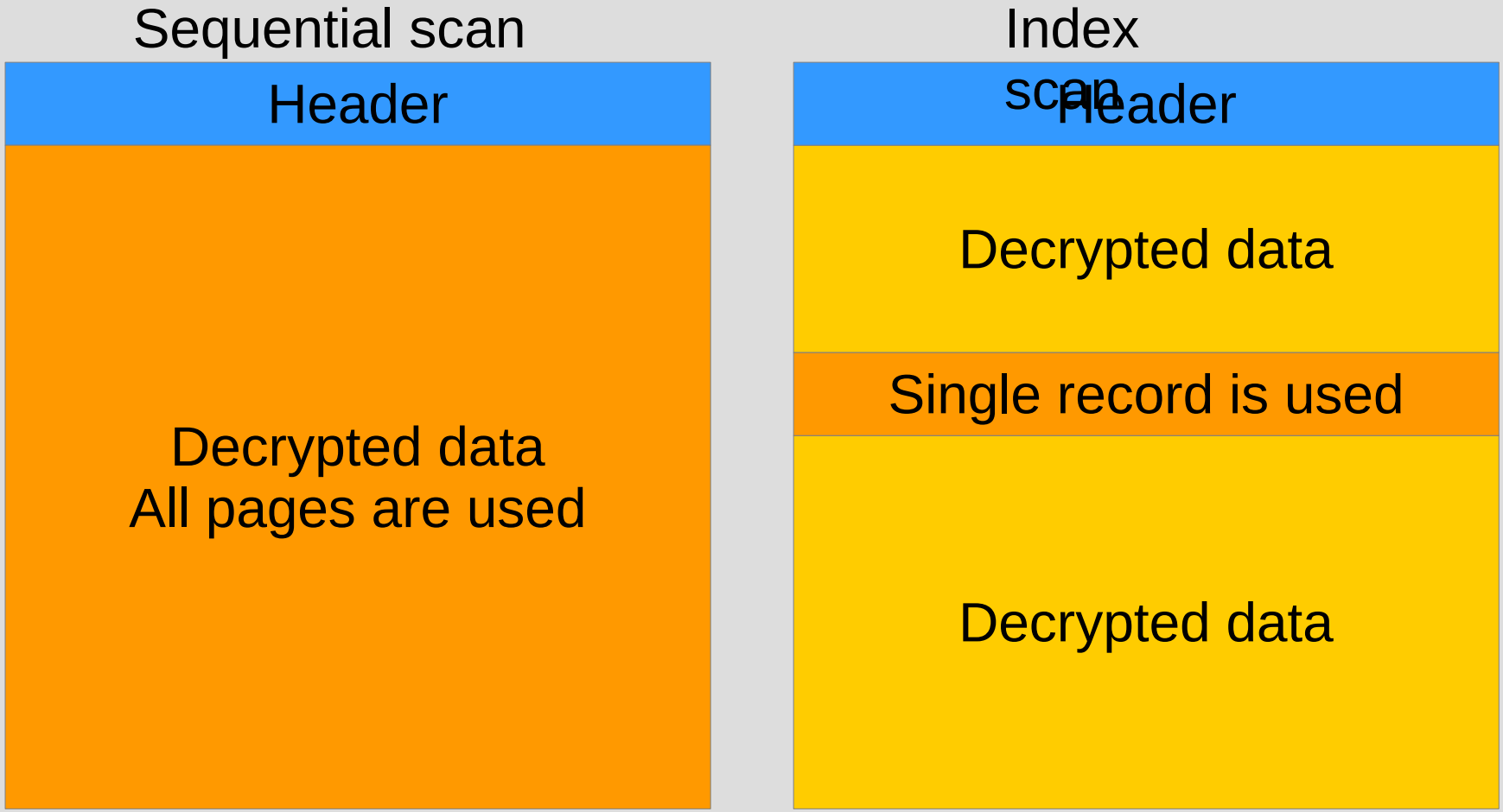
- Performance (dedicated server)
  - 24 (12 with HT) CPU cores
  - RAM 32 Gb
  - SSD
  - 100 connections, 90 minute
  - AES, using OpenSSL
  - DefaultDbCachePages = 768K (6Gb > DB size)

(operations / minute)

Forced writes	Not encrypted	Encrypted	Performance loss
On	4491	4152	8%
Off	4346	4183	4%

Encrypting Firebird databases

- Performance – sequential vs. index scan



Encrypting Firebird databases

- Initial encryption performance (desktop)
  - 8 CPU cores (AMD FX-8120)
  - RAM 8 Gb
  - Slow SATA
  - AES, using OpenSSL
  - Default cache (16 Mb < DB size)
  - Dedicated use of database

Pages (8k) / second

Forced writes	Encryption
On	3964
Off	6378

Thanks for your attention!