DelphiDay italian conference

PIACENZA EDITION

2019 - XVIII Edizione

CR1TTOGR4F14

Firebird 4.x

wintech italia

embarcadero reseller partner

Fabio Codebue

P-SOFT

P-Soft

www.p-soft.biz

www.firebirdsql.it

@fcodebue

linkedin.com/in/fcodebue

@delphiforce

f.codebue@p-soft.biz

AGENDA

- ➔ Cryptographic function
- ➔ Backup & encryption
- ➔ Encrypt it!
- ➔ Simple encryption plugin
- ➔ Thirrd party encryption plugin

Cryptographic function

Last update firebird 4.0.0 beta 1

Built-in Cryptographic Functions

8 new built-in functions supporting cryptographic tasks

- ***ENCRYPT() and DECRYPT()***
- ***RSA_PRIVATE() and RSA_PUBLIC()***
- ***RSA_ENCRYPT() and RSA_DECRYPT()***
- ***RSA_SIGN()***
- ***RSA_VERIFY()***
- ***BASE64_ENCODE() and BASE64_DECODE()***
- ***CRC32()***

Cryptographic Functions

ENCRYPT() and DECRYPT()

For **encrypting/decrypting data using a symmetric cipher**

```
{ENCRYPT | DECRYPT}
        ( <string | blob> USING <algorithm> [MODE <mode>] KEY <string>
```

```
select encrypt('897897' using sober128 key 'AbcdAbcdAbcdAbcd' iv
'01234567') from rdb$database;
```

```
select decrypt(x'0154090759DF' using sober128 key 'AbcdAbcdAbcdAbcd' iv
'01234567') from rdb$database;
```

Cryptographic Functions

RSA_PRIVATE()

Returns an RSA private key of specified length (in bytes) in PKCS#1 format as a VARBINARY string.

```
RSA_PRIVATE ( <smallint> )
```

```
select rdb$set_context('USER_SESSION', 'private_key’,  rsa_private(256))
from rdb$database;
```

Cryptographic Functions

RSA_PUBLIC()

Returns the RSA public key for a specified RSA private key. Both keys are in PKCS#1 format

```
RSA_PUBLIC ( <private key> )
```

```
select rdb$set_context('USER_SESSION', 'public_key',

rsa_public(rdb$get_context('USER_SESSION', 'private_key'))) from
rdb$database;
```



Cryptographic Functions

RSA_ENCRYPT()

Pads data using OAEP padding and encrypts it using an RSA public key. Normally used to encrypt short symmetric keys which are then used in block ciphers to encrypt a message.

```
RSA_ENCRYPT ( <string> KEY <public key> [LPARAM
              <string>] [HASH <hash>] )
```

select rdb$set_context('USER_SESSION', 'msg', rsa_encrypt('Some message'

key rdb$get_context('USER_SESSION', 'public_key'))) from rdb$database;

Cryptographic Functions

RSA_DECRYPT()

Decrypts using the RSA private key and OAEP de-pads the resulting data.

```
RSA_DECRYPT ( <string> KEY <private key> [LPARAM
              <string>] [HASH <hash>] )
```

KEY should be a value returned by the RSA_PRIVATE function. LPARAM is the same variable passed to RSA_ENCRYPT. If it does not match what was used during encoding, RSA_DECRYPT will not decrypt the packet.

```
select rsa_decrypt(rdb$get_context('USER_SESSION', 'msg')

key rdb$get_context('USER_SESSION', 'private_key')) from
rdb$database;
```

Cryptographic Functions

RSA_SIGN()

Performs PSS encoding of the message digest to be signed and signs using the RSA private key.

```
RSA_SIGN ( <string> KEY <private key> [HASH <hash>]
           [SALT_LENGTH <smallint>] )
```

```
select rdb$set_context('USER_SESSION', 'msg',
        rsa_sign(hash('Test message' using sha256)
        key rdb$get_context('USER_SESSION', 'private_key')))
from rdb$database;
```

Cryptographic Functions

RSA_VERIFY()

Performs PSS encoding of message digest to be signed and verifies its digital signature using the RSA public key

```
RSA_VERIFY ( <string> SIGNATURE <string>
             KEY <public key>
             [HASH <hash>] [SALT_LENGTH <smallint>] )
```

```
select rsa_verify(hash('Test message' using sha256)
         signature rdb$get_context('USER_SESSION', 'msg')
         key rdb$get_context('USER_SESSION', 'public_key'))
from rdb$database;
```

Cryptographic Functions

BASE64_ENCODE() and BASE64_DECODE()

These two functions are for encoding and decoding input data between string and BASE64 representation.

They operate with character strings and BLOBs.

```
BASE64_ENCODE( <binary data> )
BASE64_DECODE( <base64 data> )
```

```
select base64_encode(public_key) from clients;
```

Cryptographic Functions

CRC32()

Accepts an argument than can be a field, variable or expression of any type recognised by DSQL/PSQL and returns a CRC-32 code calculated from the input data with the polynomial 0x04C11DB7.

A constant, such as a file name passed as a string, returns the CRC for only the input.

```
CRC32( <any value> )
```

```
select crc32(job_title) from job;
```

```
select crc32('Firebird-4.0.0.716-Alpha1.amd64.tar.gz')
from rdb$database;
```

Backup & encryption

Backup and Restore with Encryption

- With an encrypted database, sooner or later it will need to be backed up and restored.
- It is not unreasonable to want the database backup to be encrypted as well.
- If the encryption key is delivered to the plug-in by some means that does not require input from the client application, it is not a big problem.
- However, if the server expects the key to be delivered from the client side, that could become a problem
- The introduction of keys to gbak in Firebird 4 provides a solution.

New Switches for Encrypted Backups & Restores

They are case-insensitive

-KEYHOLDER

- This is the main switch necessary for gbak to **access an encrypted database.**

To backup an encrypted db

```
gbak -b -keyholder MyKeyHolderPlugin host:dbname backup_file_name
```

To restore a database that was previously backed up encrypted:

```
gbak -c -keyholder MyKeyHolderPlugin backup_file_name host:dbname
```

New Switches for Encrypted Backups & Restores

-KEYHOLDER sample

- restore the database from a backup file made using non-default Crypt and Keyholder plug-ins, using the same Keyname as was used for the backup;
- restore a non-encrypted backup as an encrypted database

```
gbak -c -keyholder MyKeyHolderPlugin -crypt MyDbCryptPlugin
  -keyname SomeKey non_encrypted_backup_file host:dbname
```

- To make an encrypted backup of a non-encrypted database:

```
gbak -b -keyholder MyKeyHolderPlugin -crypt MyDbCryptPlugin
   -keyname SomeKey host:dbname encrypted_backup_file
```

- To create a compressed, encrypted backup:

```
gbak -b -keyholder MyKeyHolderPlugin -zip host:dbname backup_file_name
```

New Switches for Encrypted Backups & Restores

They are case-insensitive

-KEYNAME

Available **to name the key explicitly**, in place of the default key specified in the original database (when backing up) or in the backup file (when restoring).

Cryptographic Functions

-CRYPT

- Available to **name the plug-in to use to encrypt** the backup file or restored database in place of the default plug-in.
- It can also be used in combination with the -KEYNAME switch to **encrypt the backup of a non-encrypted database** or **to encrypt a database restored from a non-encrypted backup**.

Cryptographic Functions

-ZIP

- Only for a backup, to **compress the backup file before encrypting it.**
- The switch is necessary because the usual approach of compressing the backup file with some favoured compression routine after gbak, perhaps using pipe, does not work with encrypted backups because they are not compressible.
- The **–ZIP switch is unnecessary for a restore** because the format is detected automatically

Encrypt it!

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

Performance (desktop)

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

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 (**DB size > 320 Mb**)

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 (DB size > 6Gb)

operations / minute

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

Encryption Plugin


Mmm
Crypto
plugin….

Third party encryption plugin

Third party encryption plugin

IBPHOENIX

IBPhoenix
YOUR PREMIER SOURCE OF FIREBIRD SUPPORT

- Compatible with AES128 or AES256, AES128 is considered to be a "business-grade security", whilst AES256 is "military".
- The plugin is available for Windows (32/64bit), Linux (32/64bit) and MacOSX (32/64bit) currently.
- However if you would like to test, please contact me
- https://www.ibphoenix.com/products/software/encryptionplugin

Third party encryption plugin

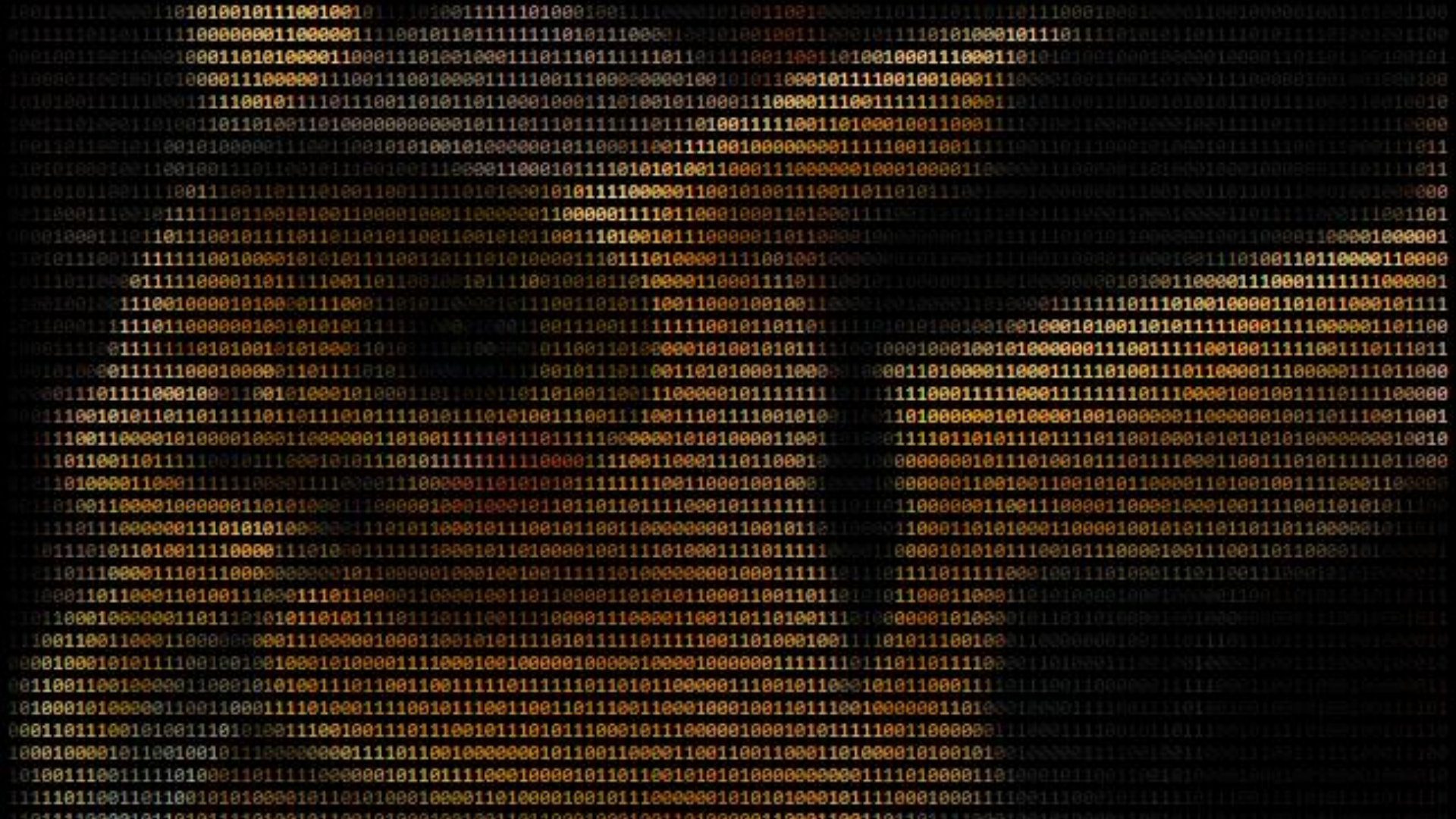
IBSURGEON

- Transparent and strong (AES256)
- Pre-built binaries for Firebird 3.0.3+, Windows 32/64 and Linux 32/64
- **Full sources included**
- Unlimited license for redistribution with all business applications of your company
- Detailed examples of the implementation in Delphi, PHP, etc
- Implementation support - our engineers will help you to implement and integrate encryption
- However if you would like to test, please contact me

Third party encryption plugin

DBEncryption Plugin for Firebird 3.0

- Available for Windows (32/64 bit) and Linux (32/64 bit).
- Latest version of IBExpert includes the 32-bit embedded version, free to use in embedded mode.
- Server versions (32 bit and x64) require the IBExpert server tools.
- https://www.ibexpert.net/ibe/pmwiki.php?n=Doc.DBEncryptionPluginFB3



GRAZIE…

Sostienici, diventa un associato …
https://firebirdsql.org/en/membership/

Fabio Codebue

P-Soft

www.p-soft.biz

www.firebirdsql.it

@fcodebue

linkedin.com/in/fcodebue

@delphiforce

f.codebue@p-soft.biz