GBase 8s 传输加密 SSL 配置示例

南 大 通 用 数 据 技 术 股 份 有 限 公 司

General Data Technologies Co., Ltd.

天津总公司：天津市高新区华苑产业园区开华道 22 号普天创新园东塔 20-23 层

电　　话：022-58815678

传　　真：022-58815679

北京分公司：北京市朝阳区太阳宫中路 12 号太阳宫大厦 10 层 1008 室

电　　话：010-88866866

传　　真：010-88864556

http://www.gbase.cn

E-mail:info@gbase.cn

南大通用数据库 GBase 8s 传输加密 SSL 配置示例，南大通用数据技术股份有限公司

作者：廖晋清

通讯方式

南大通用数据技术股份有限公司

中国天津市高新区华苑产业园区开华道 22 号普天创新园东塔 20-23 层(300384)

电话：400-013-9696　　邮箱：info@gbase.cn

版本记录

日期	版本	变更说明	作者	审核
2022.04.11	V1.00	新建	廖晋清	

目 录

1. 传输加密介绍

传输加密，是用户数据在网上进行传输，为了防止数据被窃密、篡改和伪造，例如投资者网上证券交易的委托数据、通讯安全等保证信息在互联网上进行安全传输，从而使用的技术手段。

传输加密的方法主要是使用加密技术、数字签名技术、时间戳、数字凭证技术等，最常用的技术为安全套接字协议（SSL）。

加密技术

是电子商务采取的主要安全保密措施，是最常用的安全保密手段，利用技术手段把重要的数据变为乱码（加密）传送，到达目的地后再用相同或不同的手段还原（解密）。加密技术的应用是多方面的，但最为广泛的还是在电子商务和 VPN 上的应用，深受广大用户的喜爱。

数字签名技术

数字签名技术即进行身份认证的技术。是以电子形式存在于数据信息之中的，或作为其附件的或逻辑上与之有联系的数据，可用于辨别数据签署人的身份，并表明签署人对数据信息中包含的信息的认可。

时间戳（timestamp）

在电子商务交易文件中，时间是十分重要的信息。在书面合同中，文件签署的日期和签名一样均是十分重要的防止文件被伪造和篡改的关键性内容。其通常是一个字符序列，唯一地标识某一刻的时间。

数字凭证技术

数字凭证也叫数字证书、数字标识。它含有持有者的有关信息，以标识他们的身份。 证书包含的内容：证书拥有者的姓名；证书拥有者的公钥；公钥的有效期；颁发数字证书的单位；颁发数字证书单位的数字签名；数字证书的序列号。

SSL

SSL 协议向基于传输通信协议（TCP／IP）——互联网上的客户服务器应用程序提供了客户端和服务器的鉴别、数据完整性及信息机密性等安全措施，可在服务器和客户端同时实现支持。其目标是为用户提供互联网和企业内联网的安全通信服务。

2. 数据库传输加密 SSL 配置示例

针对网络中传输的数据进行加密，焦点就在连接上。对于 GBase 8s 体系下，总共有如下连接场景需要进行数据传输加密：

1) 客户端连接数据库服务端，包含：csdk 或 jdbc 连接 GBase 8s。

2) 客户端连接到连接管理器，包含：csdk 或 jdbc 连接连接管理器。无论连接管理器选择代理，还是重定向，客户端都会建立到连接管理器的连接。

3) 连接管理器连接数据库服务端。

4) 数据库服务端连接数据库服务端。对于高可用环境，数据库服务端之间是存在连接的。

对于连接而言，数据库服务端和连接管理器都存在组的概念。客户端可以选择连接数据库服务端的组或者连接管理器的组。经过调研，真正的连接不是建立在组上的。客户端代码，会根据 sqlhosts 的配置，在连接前选择合适数据库服务端或连接管理器进行连接。根据这个情况，对于组的连接不需要专门讨论。

本文基于数据库版本为 GBase 8s V8.8 3.0.0_1，操作系统为 CentOS 7.8 编写。

2.1. 准备证书

GBase 8s 配置传输加密功能，首先需要生成数字证书，后续将数字证书分别导入到服务端和客户端的密钥数据库（也称作密钥库）中。

附件：gbasedbt_certs.tar 包含了已经生成的私有数字证书。

数字证书的密钥是：11111111

gbasedbt_certs.tar

下载地址：https://gbasedbt.com/dl/GBase8s-Certs/gbasedbt_certs.tar

2.2. 安全连接配置

2.2.1. 数据库服务端配置

首先，Server 需要部署的内容包括如下部分：

1）onconfig 需要配置参数 encrypt VP 和相应的网络类型。示例如下：

```
DBSERVERALIASES gbase01_ssl

VPCLASS encrypt,num=1
NETTYPE socssl,1,150,NET
```

2）${GBASEDBTSQLHOSTS}.ext，指向数据加密传输的配置文件。

环境变量中必须要指定 GBASEDBTSQLHOSTS 配置参数指定的文件。

3）${GBASEDBTDIR}/certs 目录，保存证书、私钥和根证。

解压 gbasedbt_certs.tar 到${GBASEDBTDIR} 目录下，生成 certs。

4）$GBASEDBTSQLHOSTS 文件

GBASEDBTSQLHOSTS 文件中有一项配置是服务端实例或实例别名的连接类型。此次新增了一种提供传输加密功能的连接类型——socssl。要使用该功能的话，需要将对应实例的连接类型配置成 socssl。

```
gbase01_ssl  onsocssl  0.0.0.0 9089
```

5）GBASEDBTSQLHOSTS.ext 文件

GBASEDBTSQLHOSTS.ext 文件用于 ssl 的相应配置。

2.2.1.1. 准备服务端证书文件

本章节的所有操作都需要使用操作系统的 gbasedbt 用户完成。

需要通过${GBASEDBTSQLHOSTS}.ext 指定配置文件。

假定 Server 安装在/opt/gbase 中，在 sqlhosts 中指定 socssl 的实例名是 gbase01_ssl。

配置文件的示例如下：

```
# ${GBASEDBTSQLHOSTS}.ext
[config]
GBS_TYPE=gbasessl

# DBSERVERNAME or DBSERVERALIASES with onsocssl
[gbase01_ssl]
server=gbase_ssl_server
client=gbase_ssl_client

[gbase_ssl_server]
TLSCACertificateFile=/opt/gbase/certs/ca/ca-cert.pem
TLSCertificateFile=/opt/gbase/certs/server/server-cert.pem
```

```
TLSCertificateKeyFile=/opt/gbase/certs/server/server-key.pem
TLSCertificateKeyFilePasswd=DqOUlNa1c2OMcEEbvTG30w==
TLSVerifyCert=never

[gbase_ssl_client]
TLSCACertificateFile=/opt/gbase/certs/ca/ca-cert.pem
TLSCertificateFile=/opt/gbase/certs/dba1/dba1-cert.pem
TLSCertificateKeyFile=/opt/gbase/certs/dba1/dba1-key.pem
TLSCertificateKeyFilePasswd=DqOUlNa1c2OMcEEbvTG30w==
TLSVerifyCert=never
```

由于 Server 端也会部署 dbaccess，另外，Server 端也会安装 csdk。因此，需要在配置文件中写入 gbase_ssl_client。

2.2.1.2. 启用加密

修改$GBASEDBTDIR/etc/sqlhosts 配置文件， 将需要启用传输加密功能的实例的连接类型设置为 onsocssl；如下述示例中，两个数据库服务名称 gbase01、gbase01_ssl，服务名 gbase01 连接类型为 onsoctcp，为普通的 tcp 通信方式，服务名 gbase01_ssl 的连接类型为 onsocssl，启用了传输加密方式：

```
gbase01          onsoctcp    0.0.0.0        9088
gbase01_ssl      onsocssl    0.0.0.0        9089
```

2.2.1.3. 配置数据库参数

修改$GBASEDBTDIR/etc/$ONCONFIG 配置文件，配置参数。

1）使同一实例同时支持普通通讯方式和加密通讯方式：通过设置 DBSERVERALIASES 来增加服务的别名，这样可以在 sqlhosts 配置文件中设置当前服务支持多种类型的连接，如下示例中实例名 gbase01、gbase01_ssl 都为当前数据库服务实例名，但是在 sqlhosts 配置文件中可以配置为不同的连接类型，这样客户端就可以通过多种通信方式与数据库服务通信：

```
DBSERVERNAME      gbase01
DBSERVERALIASES   gbase01_ssl
```

2）配置支持通讯加密的 VP：数据库加密和解密操作将由 Encrypt VP 执行。通过 VPCLASS 来配置该类型 VP 的属性；如果未配置 VPCLASS，则 Server 会默认启动一个 Encrypt VP。VPCLASS 的配置请参考《GBase 8s 管理员参

考手册》，示例如下：

```
VPCLASS encrypt,num=1
```

3）配置支持通讯加密的连接方式：设置 NETTYPE 来配置数据库连接的轮询线程和每个线程的连接数。如果未配置轮询线程，则 Server 将启动一个轮询线程。NETTYPE 的配置请参考《GBase 8s 管理员参考手册》，示例如下：

```
NETTYPE socssl,1,50,NET
```

2.2.1.4. 重启服务

服务端证书配置好后，需要重启数据库服务来生效。

2.2.2. 配置 CSDK 的传输加密

这里特指只安装了 CSDK 的客户端

CSDK 需要部署的内容包括如下部分：

1）${GBASEDBTSQLHOSTS}.ext，指向数据加密传输的配置文件

2）certs 目录，保存证书、私钥和根证。

3）$GBASEDBTSQLHOSTS 文件

用来配置待连接的服务端实例或实例别名。和 server 端一样，需要将对应实例的连接类型配置成 socssl。

2.2.2.1. 准备客户端证书文件

客户端也需要通过${GBASEDBTSQLHOSTS}.ext 指定数据库配置文件。

假定 CSDK 客户端安装在/opt/csdk 中。客户端需要连接的实例名是 gbase01_ssl。配置文件示例，如下：

```
# ${GBASEDBTSQLHOSTS}.ext
[config]
GBS_TYPE=gbasessl

# DBSERVERNAME or DBSERVERALIASES with onsocssl
[gbase01_ssl]
client=gbase_ssl_client
```

```
[gbase_ssl_client]
TLSCACertificateFile=/opt/csdk/certs/ca/ca-cert.pem
TLSCertificateFile=/opt/csdk/certs/dba1/dba1-cert.pem
TLSCertificateKeyFile=/opt/csdk/certs/dba1/dba1-key.pem
TLSCertificateKeyFilePasswd=DqOUlNa1c2OMcEEbvTG30w==
TLSVerifyCert=never
```

2.2.2.2. dbaccess、odbc 之 sqlhosts 配置

dbaccess、odbc 通过修改$GBASEDBTDIR/etc/sqlhosts 配置文件来指定连接的启用传输加密的数据库服务实例，并将连接类型设置为 onsocssl 来启用客户端的传输加密功能，示例如下：

```
gbase01_ssl         onsocssl     h01.gbasedbt.com         9089
```

示例中 gbase01_ssl 为启用传输加密的数据库服务实例。

2.2.2.3. dbaccess 验证

配置完后通过 dbaccess 连接启用传输加密的数据库服务，如果可以正常连接并执行 SQL 语句，则 dbaccess 传输加密功能配置成功。

1）配置环境变量：

创建文件 gbase01_ssl.ksh，并通过 source 命令执行。文件内容如下：

```
export GBASEDBTDIR=/opt/csdk                                  # 安装目录
export GBASEDBTSERVER=gbase01_ssl                             # 服务名称
export GBASEDBTSQLHOSTS=${GBASEDBTDIR}/etc/sqlhosts           # SQLHOSTS 文件
export PATH=${GBASEDBTDIR}/bin:$PATH                          # 执行文件目录
```

注：需要保持与 SERVER 端一致的 DB_LOCALE、CLIENT_LOCALE 和 GL_USEGLU 等参数，应当一起配置。

示例中 GBASEDBTDIR 根据 dbaccess 为 Server 还是 CSDK 中的 dbaccess 来设置为 CSDK 或者 Server 的安装路径。

2）dbaccess 连接

执行 dbaccess，然后执行 SQL，如果执行成功则说明 dbaccess 配置后可以正常运行。

```
[gbasedbt@localhost ~]$ dbaccess - -
```

```
> connect to 'testdb@gbase01_ssl' user 'gbasedbt';
   输 入 密 码 :

已 连 接 。
```

2.2.2.4. ODBC 验证

可以通过 unixODBC 程序来测试 ODBC 在配置传输加密功能后能否正常使用

1）配置环境变量：

创建文件 gbase01_ssl.ksh，并通过 source 命令执行。文件内容如下：

```
export GBASEDBTDIR=/opt/csdk                                    # 安装目录
export GBASEDBTSERVER=gbase01_ssl                               # 服务名称
export GBASEDBTSQLHOSTS=${GBASEDBTDIR}/etc/sqlhosts             # SQLHOSTS 文件
export ODBCINI=/home/gbasedbt/odbc.ini                          # ODBCINI 位置
export
LD_LIBRARY_PATH=${GBASEDBTDIR}/lib:$GBASEDBTDIR/lib/cli:${GBASEDBTDIR}/lib/esql
:${LD_LIBRARY_PATH}                                             # LD 库文件目录
```

2）配置 ODBC：

根据 ODBC 配置文件${GBASEDBTDIR}/etc/odbc.ini，按照实际环境创建 odbc.ini

```
;-------------------------------------------------------------------------
; GBase ODBC Sample File
;
; File:         odbc.ini
;
;-------------------------------------------------------------------------
[ODBC Data Sources]
gbase01_ssl=GBase ODBC DRIVER
;
; Define ODBC Database Driver's Below - Driver Configuration Section
;
[gbase01_ssl]
Driver=/opt/csdk/lib/cli/iclis09b.so
Description=GBase ODBC DRIVER
Database=testdb
LogonID=gbasedbt
pwd=GBase123
```

```
Servername=gbase01_ssl
CursorBehavior=0
CLIENT_LOCALE=zh_CN.utf8
DB_LOCALE=zh_CN.utf8
TRANSLATIONDLL=/opt/csdk/lib/esql/igo4a304.so
;
; UNICODE connection Section
;
[ODBC]
;uncomment the below line for UNICODE connection
;UNICODE=UCS-4
;
; Trace file Section
;
Trace=0
TraceFile=/tmp/odbctrace.out
InstallDir=/opt/csdk
TRACEDLL=idmrs09a.so
```

3）ODBC 连接测试：

通过 isql -v gbase01_ssl 连接到数据库， SQL 语句操作如果执行成功则说明 ODBC 配置后可以正常运行。

```
[gbasedbt@localhost ~]$ isql -v gbase01_ssl
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
SQL> select user from dual;
+--------------------------------+
|                                |
+--------------------------------+
| gbasedbt                       |
+--------------------------------+
SQLRowCount returns -1
1 rows fetched
SQL>
```

2.2.3. 配置 JDBC 的传输加密

JDBC 的传输加密功能独立于其他客户端的配置，在使用 JDBC 的程序中进行必要的设置。

2.2.3.1. JDBC 客户端配置证书

gbasedbt_certs.tar 解压到 jdbc 客户端。

例如/home/gbasedbt/certs 目录下。

2.2.3.2. JDBC 验证用例

本章节提供了一个验证 JDBC 传输加密功能的示例程序，连接数据库服务并获取数据库服务的版本信息，然后按照步骤编译运行即可。

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class SSLTest {

    public static void main(String[] args) {
        Connection conn = null;
        try {
            /* System properties for keystore */
            String propdir = System.getProperty("user.dir");
            /* keyStore */
            System.setProperty("javax.net.ssl.keyStore", propdir +
"/certs/sysdba/sysdba.keystore");
            System.setProperty("javax.net.ssl.keyStorePassword", "11111111");
            /* trustStore */
            System.setProperty("javax.net.ssl.trustStore", propdir +
"/certs/ca/ca.truststore");
            System.setProperty("javax.net.ssl.trustStorePassword", "11111111");

            try {
                Class.forName("com.gbasedbt.jdbc.Driver");
            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            }
```

```
            /* useSSL=true;sslConnection=true */
            String url =
"jdbc:gbasedbt-sqli://h01.gbasedbt.com:9088/testdb:gbasedbtserver=gbase01_ssl;" +
            "DB_LOCALE=zh_CN.utf8;CLIENT_LOCALE=zh_CN.utf8;useSSL=true;sslConnection=true";
            String username = "gbasedbt";
            String password = "GBase123";
            conn = DriverManager.getConnection(url, username, password);
            if(conn != null) {
                System.out.println(" Successfully connected to GBasedbt database using SSL
Connection");
                System.out.println(" Database version  ...: " +
conn.getMetaData().getDatabaseProductName());
            }
        } catch (Exception e) {
            System.err.println("Error Message : " +e.getMessage());
            if(e instanceof SQLException)
                System.err.println("Error Code : " +((SQLException)e).getErrorCode());
            e.printStackTrace();
        } finally {
            if(conn != null)
                try {
                    conn.close();
                } catch (SQLException e) {}
        }
    }

}
```

编译，需要将 gbasedbtjdbc.jar 加入到 CLASSPATH 中：

```
javac -cp .:gbasedbtjdbc_3.3.0_2_36477d.jar SSLTest.java
```

运行

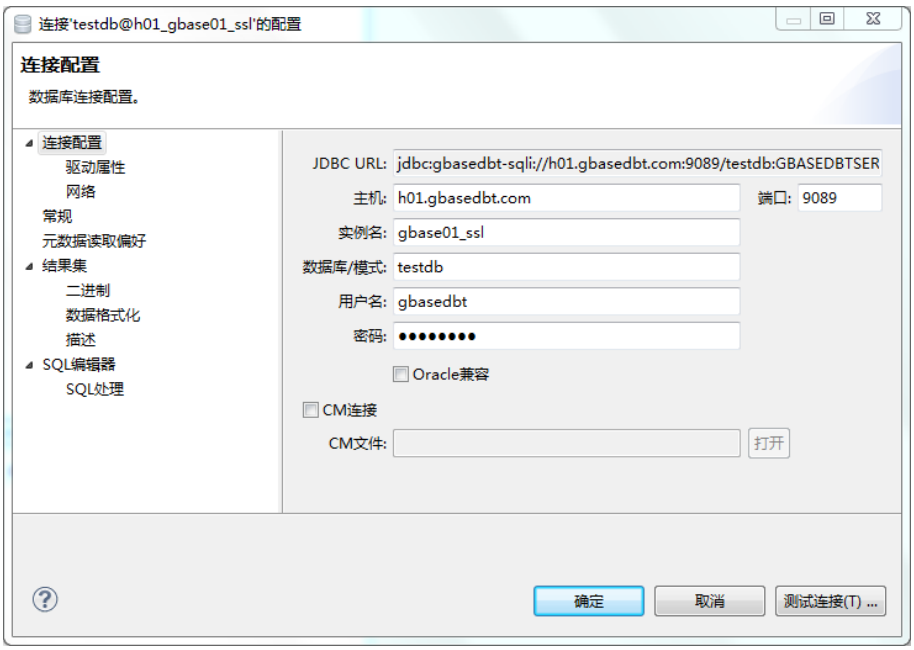
```
java -cp .:gbasedbtjdbc_3.3.0_2_36477d.jar SSLTest
```

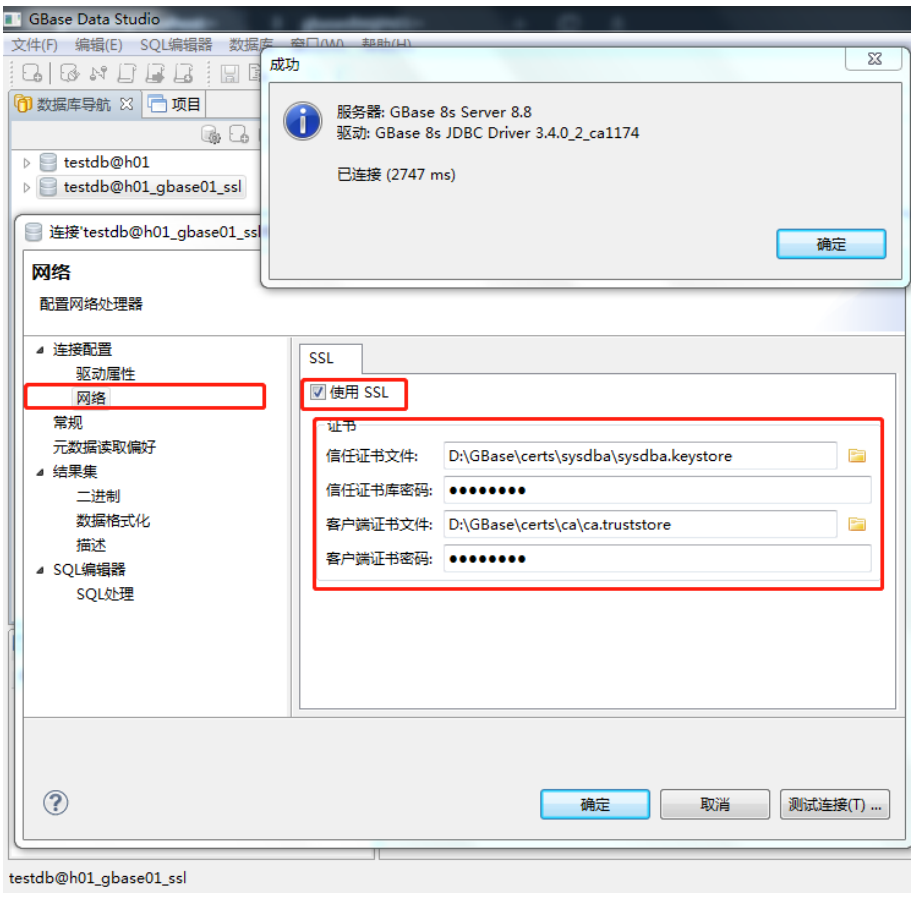
结果如下，表示使用 ssl 连接成功

```
 Successfully connected to GBasedbt database using SSL Connection
 Database version  ...: GBase 8s Server
```

2.2.3.3. JDBC 客户端（GBase DataStudio）连接

在连接配置->网络 中配置 SSL，参考 JDBC 验证用例，配置相应的证书文件和相应的密钥。


连接'testdb@h01_gbase01_ssl'的配置
连接配置
数据库连接配置。
连接配置
驱动属性
网络
常规
元数据读取偏好
结果集
二进制
数据格式化
描述
SQL编辑器
SQL处理
JDBC URL: jdbc:gbasedbt-sqli://h01.gbasedbt.com:9089/testdb:GBASEDBTSER
主机: h01.gbasedbt.com
端口: 9089
实例名: gbase01_ssl
数据库/模式: testdb
用户名: gbasedbt
密码: ••••••••
Oracle兼容
CM连接
CM文件:
打开
确定
取消
测试连接(T) ...


GBase Data Studio
文件(F) 编辑(E) SQL编辑器
数据库导航
项目
testdb@h01
testdb@h01_gbase01_ssl
成功
服务器: GBase 8s Server 8.8
驱动: GBase 8s JDBC Driver 3.4.0_2_ca1174
已连接 (2747 ms)
确定
连接'testdb@h01_gbase01_ssl
网络
配置网络处理器
连接配置
驱动属性
网络
常规
元数据读取偏好
结果集
二进制
数据格式化
描述
SQL编辑器
SQL处理
SSL
使用 SSL
证书
信任证书文件: D:\GBase\certs\sysdba\sysdba.keystore
信任证书库密码: ••••••••
客户端证书文件: D:\GBase\certs\ca\ca.truststore
客户端证书密码: ••••••••
确定
取消
测试连接(T) ...
testdb@h01_gbase01_ssl