Secure Data Synchronization Using the Relay Server and MobiLink Server Farms

A whitepaper from Sybase iAnywhere

Author: Joshua Savill, Sybase iAnywhere – Product Manager

Date: October 30th, 2008

This whitepaper was written in the context of SQL Anywhere 11.

SYBASE iAnywhere

CONTENTS

FIGURES

INTRODUCTION

With data being pushed to the forefront of company lines, demand for always-available data is at an all time high. Coupled with wireless and occasionally-connected environments, the need for systems to be available 24/7 is no longer a requirement but a necessity. Data transmission and data availability are integral components of corporate development and as a result, corporate data systems are being pushed to new limits.

MobiLink already provides a highly scalable session-based synchronization system that allows bi-directional data transfer between a main database, called the consolidated database, and many remote databases. The consolidated database can be one of several ODBC-compliant databases, including SQL Anywhere, Sybase Adaptive Server Enterprise, Oracle, Microsoft SQL Server, MySQL, IBM DB2 LUW and IBM DB2 Mainframe.

With today's heavy demands, MobiLink systems are being pushed further than ever. The introduction of SQL Anywhere 11 brings a new feature to leverage the MobiLink server into a server farm configuration for high availability and higher throughput systems. MobiLink server farms provide redundancy and concurrency in systems that are required to be available at all times. The MobiLink server farm ensures that failover and load balancing are automatic, providing availability to synchronizing clients even when hardware and system failures occur in the environment.

MobiLink server farms complement Database High Availability that was introduced in SQL Anywhere 10. Before reading this document, be sure to read our other documents related to security and high availability subjects in SQL Anywhere.

1. Configuring the Relay Server with Microsoft IIS using SSL: http://www.sybase.com/detail?id=1059277
2. MobiLink End-to-End Encryption: http://www.sybase.com/detail?id=1058723
3. MobiLink Synchronization High Availability: http://www.sybase.com/detail?id=1056536

SOFTWARE REQUIRED

- Microsoft Windows 2003 Server
  - Microsoft Internet Information Service 6.0
- SQL Anywhere with MobiLink 11.0.0, with the included RSA encryption option
- Code sample "SQL Anywhere 11.0.0 - MobiLink Server Farms synchronization with the Relay Server" found on the Sybase iAnywhere website here: http://www.sybase.com/detail?id=1060465

MOBILINK SERVER FARMS

A MobiLink server farm is an explicit group of MobiLink servers that all connect to a single consolidated database. The MobiLink server farm is designed for redundancy, load balancing, failover, and concurrent synchronization. Synchronizations from the same remote ID are automatically detected across the entire farm, removing the need for 3rd party load balancers to maintain stickiness during a single synchronization. Figure 1 shows the architecture of a MobiLink server farm.

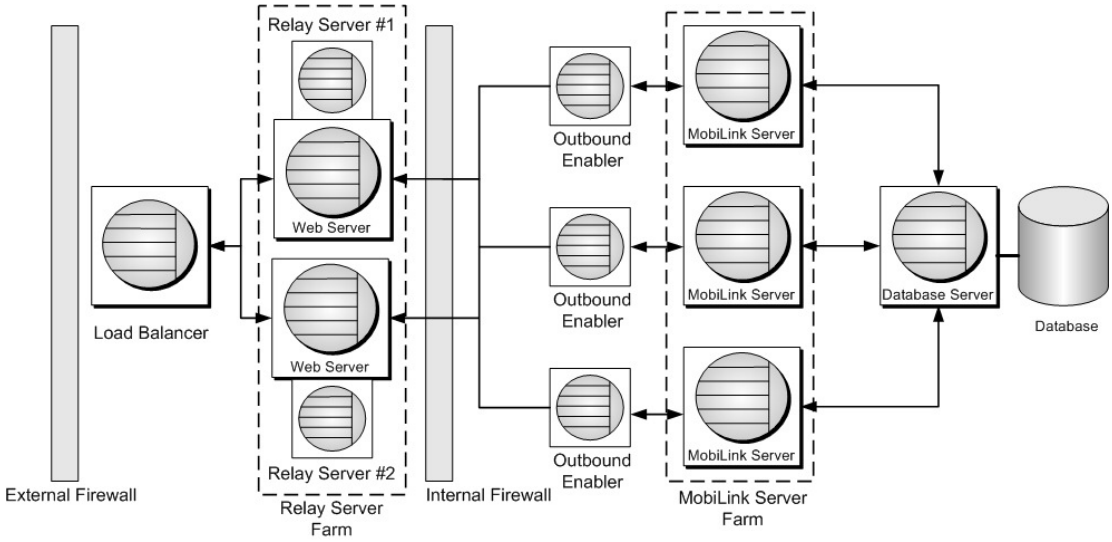

Figure 1 – MobiLink Server Farm Architecture

The MobiLink servers in a server farm all share state information that is stored in the MobiLink system tables located in the consolidated database. The relay servers perform load balancing using a round robin algorithm amongst the MobiLink servers. The relay servers ensure HTTP/S requests for a session go to the same MobiLink server for the entire synchronization. Failover is automatically supported and handled with communication between the relay server and outbound enabler.

If the relay server is not used, then a 3rd party load balancer is required for TCPIP, TLS, HTTP, and HTTPS communication with the MobiLink server farm. Figure 2 shows the architecture of a MobiLink server farm using a 3rd party load balancer.

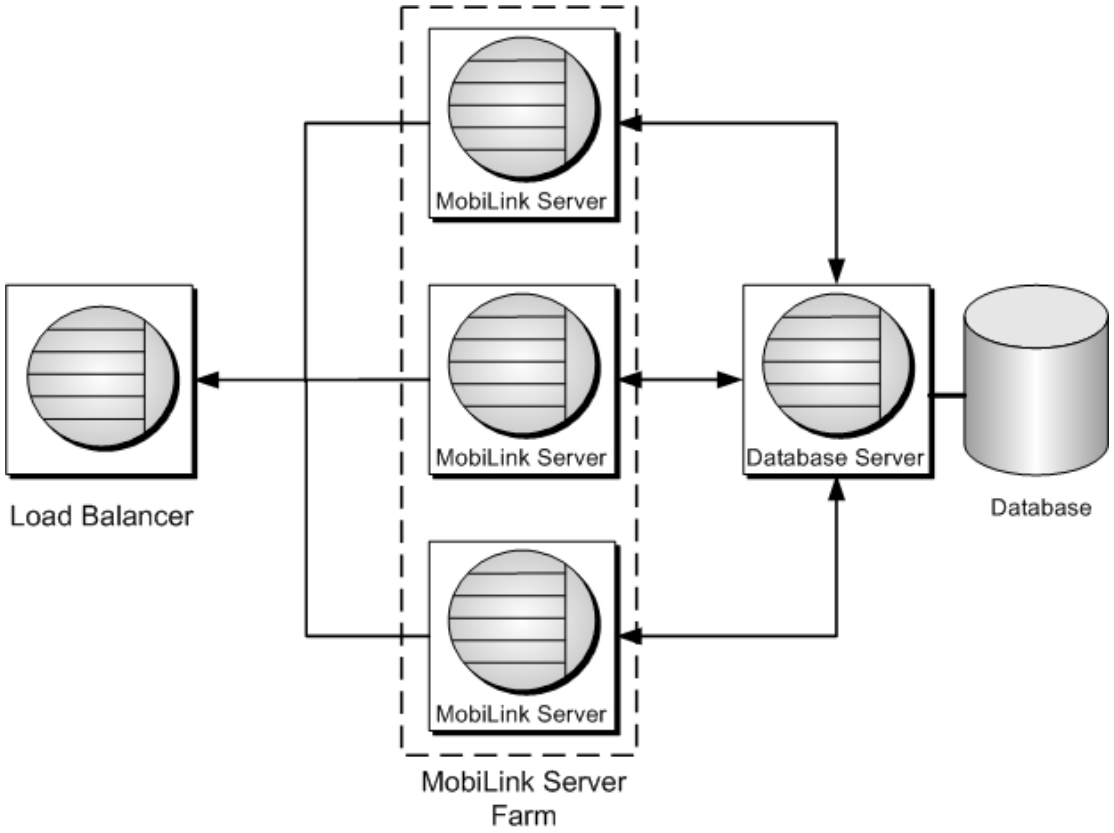

Figure 2 – MobiLink Server Farm Architecture with a Load Balancer

Performance tip: The load balancer does not need to maintain session affinity between the MobiLink client and MobiLink server after a synchronization is complete. Different synchronization sessions from the same MobiLink client may go through different MobiLink servers in the farm. Significant load balancer performance gain can be obtained by turning off the session affinity feature.

By default, the MobiLink server is not enabled for use in a farm environment. To enable a MobiLink server for use in a farm, the MobiLink server must have a unique name, which is specified using the –zs option on the mlsrv11 command line. The MobiLink server is also required to run in shared server state mode, which is specified with the –ss option on the mlsrv11 command line.

MobiLink server farms also provide support for the server-initiated synchronization Notifier. This is supported by making one Notifier primary and all other Notifiers in the farm secondary. The primary Notifier is in control of notification, either directly or indirectly, via the secondaries. The secondary Notifiers also route Listener information to the primary Notifier. This is configured using the –lsc option on the mlsrv11 command line.

MobiLink - Server Administration »
Using MobiLink Server Technology »
MobiLink server options »
-lsc option

Online:
http://dcx.sybase.com/index.php#http%3A%2F%2Fdcx.sybase.com%2F1100en%2Fmlserver_en11%2Fml-syncserver-s-7767735.html

RELAY SERVER AND MOBILINK SERVER FARMS DEMONSTRATION

The sample demonstration uses both transport layer security and end-to-end encryption to secure the MobiLink communication stream. The MobiLink clients and relay sever communicate using SSL with RSA encryption through a Microsoft IIS web server. The outbound enabler and relay server also communicate using the same SSL encryption through the Microsoft IIS web server.

The MobiLink client and MobiLink server use end-to-end encryption, using RSA, for the entire length of the communication stream. Figure 3 shows the sample demonstration architecture, including each component of the encrypted stream. End-to-End encryption is highlighted in yellow, and SSL communication with the Microsoft IIS web server is highlighted in red.

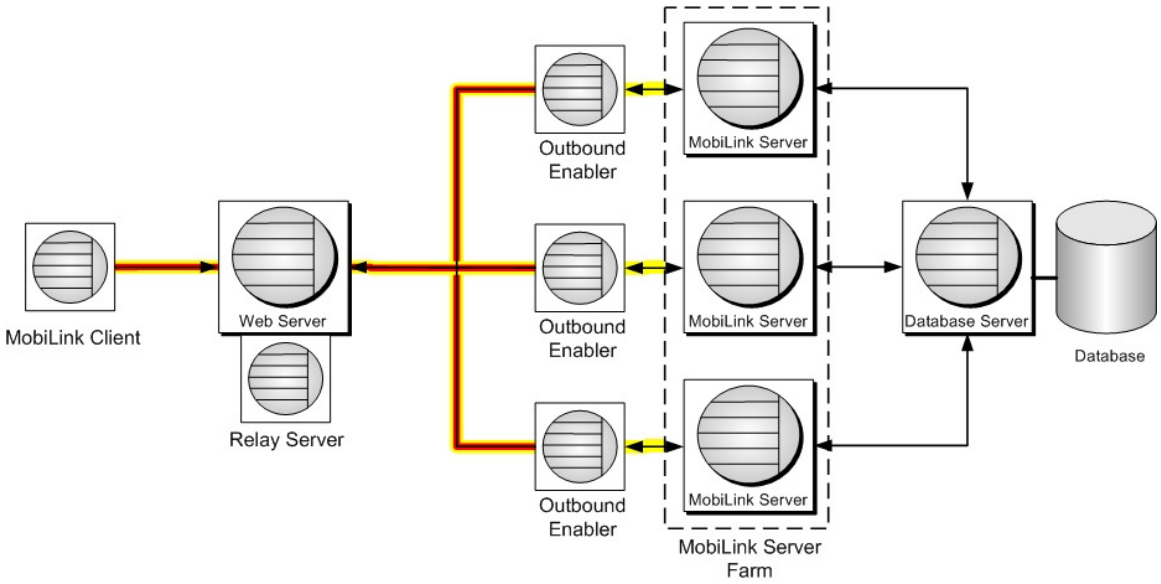

Figure 3 – Relay Server and MobiLink Server Farm Demonstration Architecture

All code samples for the relay server and MobiLink server farm demonstration can be found on the Sybase iAnywhere website here: http://www.sybase.com/detail?id=1060465

For more information on configuring the relay server with a Microsoft IIS web server using SSL communication, see the following white paper: http://www.sybase.com/detail?id=1059277

For more information on MobiLink with end-to-end encryption, see the following: http://www.sybase.com/detail?id=1058723

CONFIGURING THE RELAY SERVER

The relay server configuration file is used to define all relay servers, MobiLink servers, and MobiLink server farms. The configuration file, *rs.config*, is divided into four sections: options, relay server, backend farm, and backend server.

OPTIONS SECTION

The options section is used to specify properties that apply to each relay server and how the relay server is started. For this demonstration, the relay server is configured to start as a Windows service:

```
#---------------------------------------
# Relay server with no auto start option
#---------------------------------------
[options]
start = no
verbosity = 2
```

The following command sets up the relay server as an auto-start Windows service:

```
dbsvc -y -as -s Automatic -w rs
"c:\Inetpub\wwwroot\ias_relay_server\server\rshost.exe" -q -qc
-f "c:\Inetpub\wwwroot\ias_relay_server\server\rs.config" -o
"c:\temp\myrshost.log"
```

Tip: It is considered best practice to start the relay server as a Windows service.

RELAY SERVER SECTION

The relay server section is used to define each relay server in the environment. Each entry in the relay server section identifies a unique relay server. If there is a relay server farm in the environment, each relay server in the farm needs to be identified in the relay server section. For this demonstration, there is only one relay server. The configuration is as follows:

```
#--------------------
# Relay server peers
#--------------------
[relay_server]
enable          = yes
host            = localhost
http_port       = 80
https_port      = 443
description     = Relay Server Definition
```

BACKEND FARM SECTION

The backend farm section specifies the properties of the MobiLink server farm. If there are multiple MobiLink server farms in the environment, there will be multiple entries in this section. A sample of the section is below:

```
#---------------
# Backend farms
#---------------
[backend_farm]
enable          = yes
id              = MobiLink.Server
client_security = on
backend_security= on
```

```
description     = MobiLink server farm is 3 MobiLink servers
```

The configuration above shows that the MobiLink server farm is identified as *MobiLink.Server*. This MobiLink server farm identification is specified to the relay server by the MobiLink client when synchronizing.

BACKEND SERVER SECTION

The backend server section defines a backend server connection to the machine running the MobiLink server. Each entry in the backend server section defines a single MobiLink server. The definition information is used by the outbound enabler when it connects to the relay server. This sample demonstration uses 3 MobiLink servers in the MobiLink server farm:

```
#-----------------
# Backend servers
#-----------------
[backend_server]
enable   = yes
farm     = MobiLink.Server
id       = MLServer1
mac      = 00-0C-29-7A-C2-AB
token    = 7b2493b0-d0d4-464f-b0de-24643e1e0feb

[backend_server]
enable   = yes
farm     = MobiLink.Server
id       = MLServer2
mac      = 00-0C-29-7A-C2-AB
token    = 7b2493b0-d0d4-464f-b0de-24643e1e0fec

[backend_server]
enable   = yes
farm     = MobiLink.Server
id       = MLServer3
mac      = 00-0C-29-7A-C2-AB
token    = 7b2493b0-d0d4-464f-b0de-24643e1e0fed
```

Each MobiLink server in the farm has a unique identification, namely MLServer1, MLServer2, and MLServer3. Each entry also references the farm *MobiLink.Server* to establish with the relay server that all 3 MobiLink servers are running in the same MobiLink server farm. The MAC address and token, specified for added security and authentication, are used by the outbound enabler when establishing the connection between the MobiLink server and the relay server.

Note: After changing the *rs.config* file, you must perform one of the following two options for the changes to take effect:

1) The relay server (*rshost.exe*) needs to be updated with the changes by executing `rshost -u -f rs.config` on the command line.

2) The Default Web Site and relay server (*rshost.exe*) need to be stopped and restarted.

CREATING THE SQL ANYWHERE CONSOLIDATED DATABASE

The SQL Anywhere consolidated database can be created using the *setup.bat* file located in the *MLServerFarm_EEE_Encryption|cons* directory.

The *setup.bat* file does the following:

- Initializes a new SQL Anywhere database *MLcons.db*
- Starts the *MLcons* database
- Executes the *syncsa.sql* script found in the *%SQLANY11%|MobiLink|setup* directory to create the MobiLink system objects in the consolidated database
- Executes the *cons_schema.sql* file to create the consolidated database schema
- Loads test data into the database
- Starts the MobiLink server farm and outbound enablers

STARTING THE MOBILINK SERVERS FOR A MOBILINK SERVER FARM

For this demonstration, there are 3 MobiLink servers that are configured in a MobiLink server farm. Each MobiLink server is started individually, listening on a unique port, and specifies the –ss option to enable a shared state across all servers. The *setup.bat* file in the *MLServerFarm_EEE_Encryption|cons* directory executes the following statements to start the MobiLink servers:

```
mlsrv11 -ss -v+ -o mlsrv1.out -zs MLServer1 -c
"DSN=https_cons" -dl -zu+ -x
http{port=2439;e2ee_type=RSA;e2ee_private_key=..\certs\eee_prv
.key;e2ee_private_key_password=pwd}

mlsrv11 -ss -v+ -o mlsrv2.out -zs MLServer2 -c
"DSN=https_cons" -dl -zu+ -x
http{port=2440;e2ee_type=RSA;e2ee_private_key=..\certs\eee_prv
.key;e2ee_private_key_password=pwd}

mlsrv11 -ss -v+ -o mlsrv3.out -zs MLServer3 -c
"DSN=https_cons" -dl -zu+ -x
http{port=2441;e2ee_type=RSA;e2ee_private_key=..\certs\eee_prv
.key;e2ee_private_key_password=pwd}
```

- **mlsrv11** – is the MobiLink server executable
- **–ss** – enables participation in a MobiLink server farm, sharing state information across all MobiLink servers in the farm
- **–v+** – enables verbose logging for the MobiLink server. This is not recommended for production systems.
- **–o mlsrv1.out** – specifies that the output of the MobiLink server console be directed to the file *mlsrv1.out*
- **–zs MLServer1** – specifies a unique name for the MobiLink server. This must match an entry in the backend server section of the *rs.config* file. All unique names for the MobiLink servers must be specified in backend server section.
- **–c "DSN=https_cons"** – specifies the connection information for communication with the consolidated database
- **–dl** – specifies that all MobiLink server messages be displayed in the console. This is not recommended for production systems.
- **–zu+** – specifies that unrecognized MobiLink user names be automatically added to the ml_user table on first synchronization. This is not recommended for production systems.
- **–x http{}** – specifies that the communication stream for synchronization be HTTP

- **port=2439** – specifies that the MobiLink server listen for new connections on port 2439. This is the default port for the MobiLink server.
- **e2ee_type=RSA** – specifies the use of RSA encryption for end-to-end encryption
- **e2ee_private_key=..\certs\eee_prv.key** – specifies the private key for end-to-end encryption
- **e2ee_private_key_password=pwd –** specifies the private key password for end-to-end encryption

Note: The end-to-end encryption keys are generated using the Key Pair Generator utility (createkey). For information on how to generate end-to-end encryption keys, see the following:

SQL Anywhere Server – Database Administration »
Administering Your Database »
Database administration utilities »
Key Pair Generator utility

Online:
http://dcx.sybase.com/index.php#http%3A%2F%2Fdcx.sybase.com%2F1100en%2Fdbadmin_en11%2Fda-dbutilities-s-4065300.html

STARTING THE OUTBOUND ENABLERS

The use of 3 MobiLink servers in this sample demonstration requires that 3 outbound enablers be used. The outbound enabler is designed like a double client connector between the relay server and the MobiLink server. The outbound enabler establishes a connection with the relay server, using HTTPS in this sample demonstration, and the MobiLink server using HTTP. The outbound enabler facilitates the communication and requests between the relay server and MobiLink server.

The outbound enablers are started in the *setup.bat* file, located in the *MLServerFarm_EEE_Encryption|cons* directory, with the following commands:

```
rsoe -cr
"host=localhost;port=443;https=1;url_suffix=/ias_relay_server/
server/rs_server.dll/;trusted_certificates=..\certs\rsa_root.c
rt" -cs "host=localhost;port=2439" -f MobiLink.Server -id
MLServer1 -t 7b2493b0-d0d4-464f-b0de-24643e1e0feb -v 2 -o
Outbound1.txt

rsoe -cr
"host=localhost;port=443;https=1;url_suffix=/ias_relay_server/
server/rs_server.dll/;trusted_certificates=..\certs\rsa_root.c
rt" -cs "host=localhost;port=2440" -f MobiLink.Server -id
MLServer2 -t 7b2493b0-d0d4-464f-b0de-24643e1e0fec -v 2 -o
Outbound2.txt

rsoe -cr
"host=localhost;port=443;https=1;url_suffix=/ias_relay_server/
server/rs_server.dll/;trusted_certificates=..\certs\rsa_root.c
rt" -cs "host=localhost;port=2441" -f MobiLink.Server -id
```

```
MLServer3 -t 7b2493b0-d0d4-464f-b0de-24643e1e0fed -v 2 -o
Outbound3.txt
```

- **rsoe** – is the relay server outbound enabler executable
- **–cr** – specifies information for establishing a connection with the relay server
- **host=localhost** – specifies the machine name or IP address where the relay server is running
- **port=443** – specifies port 443 is used for communication. This is the default port for HTTPS.
- **https=1** – specifies that HTTPS be used for communication
- **url_suffix=/ias_relay_server/server/rs_server.dll/** – specifies the location of the relay server server extension that facilitates communication
- **trusted_certificates=..\certs\rsa_root.crt** – specifies the root certificate for SSL communication with the relay server
- **–cs** – specifies information for establishing an HTTP connection with the MobiLink server
- **–f MobiLink.Server** – specifies the name of the MobiLink server farm. This must match the entry in the *rs.config* file.
- **–id** MLServer1 – specifies the name for the MobiLink server in the MobiLink server farm. This must match an entry in the *rs.config* file and the –zs switch on the MobiLink server command line.
- **–t 7b2493b0-d0d4-464f-b0de-24643e1e0feb** – specifies the security token used for authentication with the relay server. This must match the entry in the *rs.config* file.
- **–v 2** – specifies a verbose logging level of 2 for the outbound enabler
- **–o Outbound1.txt –** specifies that verbose logging is directed to the *outbound1.txt* file

Note: The –cs switch defaults to host=localhost;port=80 if it is not specified on the outbound enabler command line.

CREATING THE SQL ANYWHERE REMOTE DATABASE

The SQL Anywhere remote database can be created using the *setup.bat* file located in the *MLServerFarm_EEE_Encryption|rem* directory.

The *setup.bat* file does the following:

- Initializes a new SQL Anywhere database *MLrem.db*
- Starts the *MLrem* database
- Executes the *rem_schema.sql* file to create the remote database schema
- Loads test data into the database
- Starts the MobiLink client (dbmlsync) to perform a synchronization

STARTING THE MOBILINK CLIENTS

The MobiLink client can be started by executing the *sync.bat* file located in the *MLServerFarm_EEE_Encryption|rem* directory.

The *sync.bat* file starts the MobiLink client as follows:

```
dbmlsync -c "DSN=https_rem" -mp sql -v+ -o rem.txt -e
"ctp=https;adr='host=localhost;port=443;url_suffix=/ias_relay_
server/client/rs_client.dll/MobiLink.Server/;trusted_certifica
tes=..\certs\rsa_root.crt;e2ee_type=RSA;e2ee_public_key=..\cer
ts\eee_pub.key'"
```

- **dbmlsync** – is the MobiLink client executable

- **–c "DSN=https_rem"** – specifies information for establishing a connection with the SQL Anywhere remote database
- **–mp sql** – specifies the password for the MobiLink user being synchronized
- **–v+** – specifies verbose logging for the MobiLink client
- **–o rem.txt** – specifies that verbose logging be directed to the *rem.txt* file
- **–e ""** – specifies extended options for MobiLink client during the synchronization
- **ctp=https** – specifies that HTTPS be used for synchronization
- **adr="** – specifies information for establishing an HTTPS connection with the relay server
- **host=localhost** – specifies the machine or IP address of the machine where the relay server is running
- **port=443** – specifies that port 443 be used for communication. This is the default port for HTTPS.
- **url_suffix=/ias_relay_server/client/rs_client.dll/MobiLink.Server/** – specifies the location of the relay server client extension that facilitates communication. The *MobiLink.Server* name is specified to indicate the MobiLink client needs to communicate with the *MobiLink.Server* MobiLink server farm.
- **trusted_certificates=..\certs\rsa_root.crt** – specifies the root certificate for SSL communication with the relay server
- **e2ee_type=RSA** – specifies that RSA encryption be used for end-to-end encryption
- **e2ee_public_key=..\certs\eee_prv.key** – specifies the public key for end-to-end encryption

SUMMARY

This document has outlined how to configure synchronization with a MobiLink server farm through the relay server. After finishing this document, you should understand how to set up a complete synchronization system, using SSL and end-to-end encryption, between the MobiLink clients and a MobiLink server farm using the relay server and a Microsoft IIS web server.