New Features
Replication Server® version 12.6, ESD #5

Document ID: DC00344-01-1260-02

Last revised: July 2005

This document describes new features available for Replication Server 12.6, ESD #5.

1. Encrypted columns in Replication Server

Sybase Adaptive Server version 12.5.3a enables the encryption of sensitive user data at the column level. Data encryption requires an encryption key and password. You can set up encrypted columns when you create or alter a table. See *New Features Adaptive Server Enterprise 12.5.3a* for more information.

Note If you want to downgrade Adaptive Server 12.5.3a to an earlier version, see the *Adaptive Server Enterprise 12.5.3a Release Bulletin* for important instructions *before* performing the downgrade. Some preliminary steps are necessary to ensure a successful downgrade.

Only Adaptive Server users with special permission can access and decrypt encrypted columns. Only the System Security Officer can grant permission to create and manage encryption keys.

Replication Server replicates encrypted columns to replicate and warm standby databases. Encrypted data is replicated as varbinary data in encrypted form, which safeguards the replicated data while Replication Server processes it in stable queues on disk.

- Replication Server cannot decrypt and does not distinguish encrypted columns in any way. Any Replication Server can replicate encrypted columns to any Adaptive Server, but the columns can be accessed and decrypted only in Adaptive Server 12.5.3a and later.
- Replication Server users—even System Administrators—cannot decrypt encrypted columns.
- Encrypted columns should be replicated in their entirety. They cannot be searched or translated.
- Encrypted columns should not be used as primary keys if the encryption key specifies an initialization vector or random padding.
- Both the primary and replicate servers must use the same charset and language so that decrypted data can be interpreted properly at the replicate site.
- Data is encrypted in canonical form for integers (MSB) and floats (IEEE and MSB) so that it can be decrypted successfully at the replicate data server in heterogeneous system environments.

The replication of encrypted columns is supported by function strings. Replication Server applies the class-level function string rs_set_ciphertext when a connection is established at each database, except the RSSD. Replication Server issues a command to indicate to the target data server that the encrypted data is in ciphertext form. If the target Adaptive Server is earlier than 12.5.3, this command is ignored.

1.1 Setting up replication of encrypted columns

The sequence of steps necessary to set up encrypted columns at the primary database must be repeated at each database where encrypted columns are to be replicated. This includes enabling encrypted columns, setting the system encryption password, duplicating the primary site's encryption key and associated key values, and encrypting the specified columns.

See *New Features Adaptive Server Enterprise 12.5.3a* for detailed information about setting up encrypted columns on Adaptive Server.

You can synchronize encryption information at the primary and replicate sites in two ways:

- By allowing Replication Server to replicate the DDL commands and procedures that set up encryption information at the primary site, or
- By manually reentering setup information at the replicate site.

1.1.1 Replicating encryption information

If your system replicates DDL commands to the MSA (multi-site availability) replicate or warm standby site, Replication Server copies all necessary encryption setup information from the primary to the replicate.

These commands and stored procedures, which create and manage encrypted columns, are replicated:

- alter table...encrypt
- create encryption key
- create table...encrypt
- drop encryption key
- grant create encryption key
- grant decrypt
- grant select on *encryption_key_name*
- revoke create encryption key
- revoke decrypt
- sp_encryption system_encr_passwd

1.1.2 Manually synchronizing encryption information

If DDL commands are not replicated to the replicate site, you must manually enable and set up encryption at both the primary and replicate sites.

In this example, the primary database is seattle_hr and the replicate database is shanghai_hr. The ssn (social security number) column in the employee table is to be encrypted at both sites.

This example encrypts a column in an existing table. You can also encrypt columns when you create a table.

At the primary site:

1 Enable encryption:

```
sp_configure 'enable encrypted columns', 1
```

2 Set the system encryption password:

```
sp_encryption system_encr_passwd, password
```

This password allows Adaptive Server to protect encryption keys in the current database.

3 Create a new encryption key.

For example, enter:

```
create encryption key ssn_key for AES
```

See *New Features Adaptive Server Enterprise 12.5.3a* for detailed instructions.

4 Encrypt the specified columns using the new key. You must supply the new key name; otherwise, Adaptive Server searches for the database's default key.

In this step, the ssn column of the employee table is being encrypted with the ssn_key created in step 3. The owner of employee must first have been granted select permission on ssn_key.

```
alter table employee modify ssn
encrypt with ssn_key
```

5 Grant permission for selected users to decrypt the ssn column. For example, to grant decrypt permission to the hr_manager_role and the hr_director_role, enter:

```
grant decrypt on employee(ssn) to hr_manager_role,
hr_director_role
```

6 Use the DDLGen utility to create a variant of the create encryption key statement that includes the new key name and key information. This generated statement will be used at the replicate site.

Enter:

```
ddlgen -Uusername -Ppassword -Sseattle_hr
-I$SYBASE/interfaces -TEK -Nseattle_hr.dbo.ssn_key
-XOD
```

In this example:

- -S – specifies the server name,
- -I – specifies the interfaces file,
- -T – specifies the DDL object—the variant of the create encrypted key statement—to be created,

- -N – specifies the primary database and key name,
- -X – specifies an extended option “OD” for generating key value attributes of encrypted keys.

The DDL object you create appears on the console window. For example:

```
create encryption key seattle_hr.dbo.ssn_key for AES
with keylength 128
passwd 0x0000E3AD330E909EC8A601
init_vector random
keyvalue 0x23AF23138DD85A40FAE1AF155B7440086C012EF
   500638BDB62E2A16473B12C4C01
keystatus 32
```

In this example, DDLGen displays the encrypted password and keyvalue. See the Adaptive Server *Utility Guide* for more information about syntax options for DDLGen.

At the replicate site:

1 Enable encryption:

```
sp_configure 'enable encrypted columns', 1
```

2 Set the system encryption password:

```
sp_encryption system_encr_passwd, password
```

This password allows Adaptive Server to protect the encrypted key in the current database. This command must repeat *exactly* the command and password entered at the primary database.

3 Edit the DDL object created in step 5 at the primary, substituting the name of the replicate database. Execute the edited DDL object. In this example, execute:

```
create encryptionkey shanghai_hr.dbo.ssn_key for AES
with keylength 128
passwd 0x0000E3AD330E909EC8A601
init_vector random
keyvalue 0x23AF23138DD85A40FAE1AF155B7440086C012EF
   500638BDB62E2A16473B12C4C01
keystatus 32
```

4 Use alter table...encrypt with *keyname* to encrypt the specified columns at the replicate database. The owner of employee must first be granted select permission on ssn_key.

Repeat step 4 as executed at the primary. For example, enter:

```
alter table employee modify ssn
encrypt with ssn_key
```

5 Grant permission for selected users to decrypt the ssn column. To grant decrypt permission to the hr_manager_role and the hr_director_role, enter:

```
grant decrypt on employee(ssn) to hr_manager_role,
hr_director_role
```