SafeNet
The Foundation of Information Security

Encrypting Critical Data In Databases

An Overview of the
Database Integration Process

WHITE PAPER

Overview

As the incidence and severity of security breaches continues to grow, it is increasingly incumbent upon organizations to begin encrypting data inside the enterprise. SafeNet™ offers breakthrough solutions that make it practical to encrypt critical data and ensure that it is secured throughout an organization. With SafeNet DataSecure™ Platforms, organizations can better ensure that they are compliant with legislative and policy mandates for security, and eliminate the risks of a breach.

SafeNet DataSecure Platforms deliver comprehensive security capabilities, including:

- Encrypting critical data in Web servers, application servers, and databases
- Ensuring that all access to critical data is carefully managed, logged, and controlled
- Administration of keys and policies in a secure, centralized fashion
- Ensuring that security processing is highly scalable and reliable

SafeNet works seamlessly with leading databases—including IBM DB2, Microsoft SQL Server, and Oracle—delivering capabilities for securely and efficiently managing encryption. DataSecure encrypts data in the database at the column level, and can be used to secure information such as credit card numbers, social security numbers, passwords, account balances, and email addresses. DataSecure significantly streamlines the administrative tasks involved in database encryption—it automates much of the configuration and implementation process and can be deployed without any disruption to the applications tied to the database.

The SafeNet DataSecure Platform is comprised of three components:

- DataSecure appliance—a dedicated hardware system
- Network-Attached Encryption™ (NAE) Server—runs on the DataSecure appliance
- SafeNet NAE Connector—software that is installed on the Web, application, or database server

The SafeNet NAE Connector features standards-based cryptographic interfaces that allow the protection of user-defined data through integration of security functions at the business logic layer. These small software components initiate encrypt and decrypt operations, and are installed on each database that has a need to interface with the SafeNet appliance. This white paper will focus on how organizations can implement SafeNet DataSecure Platforms and employ them to encrypt critical data inside a database.

How It Works

Integrating the DataSecure platform into your existing database infrastructure is a straightforward, automated process. The NAE Connector component outlined above consists of complete code to manage seamless interaction between the database and the DataSecure platform.

The DataSecure appliance features a secure, Web-based user interface that steps administrators through the configuration process and, once parameters have been set, even automates the installation of the required NAE Connector software on the database. Once installed and configured, the NAE Connector dynamically generates all of the necessary stored procedures and functions to:

- Encrypt and decrypt data on demand from inside the database
- Migrate data from plaintext to ciphertext, and change the database schema to accommodate encrypted columns
- Rotate cryptographic keys
- Automate subsequent encrypt and decrypt operations
- Authenticate users so that only authorized users are able to access sensitive data

This transparent integration means that you can continue using your existing SQL statements without having to modify them. And, more importantly, you do not have to write any of the logic to perform encrypt or decrypt operations from your database. Following is a high-level diagram outlining how the solution is deployed.

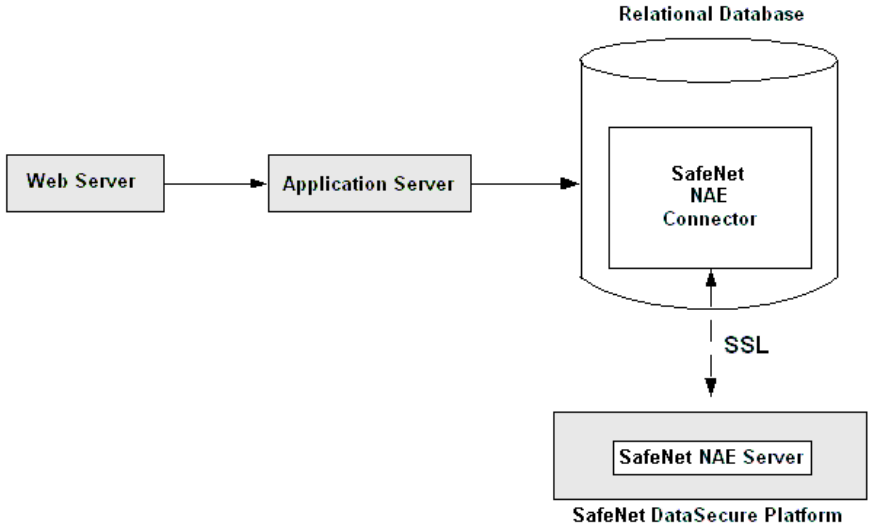

Figure 1: High Level View of Implementation with SafeNet DataSecure Platform

The previous diagram shows a Web server accessing an application server, which is making a call to a database to access sensitive data. The NAE Connector is installed in the database and the sensitive data is stored in encrypted format within the database. The user who requests the sensitive data must have permission within the database to make a request to the NAE Connector. That user must also have access to the requested key on the NAE Server. If either of the conditions above is not met, then the user is not given access to the sensitive data. If the user is authorized to use the requested key for decryption, then the NAE Server performs the decrypt operation on the sensitive data. The decrypted data is then passed back to the database.

The DataSecure appliance is the physical device where cryptographic operations and key management operations are performed. Different hardware platforms provide varying capabilities for performance and FIPS compliance. The NAE Server and all cryptographic keys reside on this hardened security system.

How Does Data Get Encrypted?

Before implementing DataSecure in your enterprise, your sensitive data is most likely being stored as plaintext, so the logical question is—*how do you migrate plaintext data into encrypted format?* The process is straightforward, and, as mentioned above, SafeNet automates this process through the DataSecure appliance's GUI.

To illustrate the simplicity of the process, take social security numbers as an example. If you have a table called CUSTOMER that stores sensitive customer data like names, addresses, and social security numbers, you might want to encrypt the social security numbers. Your original CUSTOMER table might look like this:

CUSTOMER

Name	SSN	Address	City	State	Zip
Irwin M. Fletcher	123456789	411 Main Street	Santa Barbara	CA	93101
Josh Ritter	111122223	1801 21st Ave	San Francisco	CA	94122
Steve Garvey	987654321	123 First Ave	Brentwood	CA	90049

Figure 2: Sensitive Data is Stored in the Clear

The first step in the process of securing your sensitive data is to identify what data you want to secure and where that data resides. In this example, social security numbers are stored in a column called SSN. Once you have identified the sensitive data, you can configure SafeNet to automate the data protection process. During the first step, SafeNet renames the table in which the sensitive data resides—the table must be renamed so that a view can be created later with the same name as the original table. Notice in Figure 3 that the table is renamed to CUSTOMER_ENC, but the SSN column is not yet changed.

CUSTOMER

Name	SSN	Address	City	State	Zip
Irwin M. Fletcher	123456789	411 Main Street	Santa Barbara	CA	93101
Josh Ritter	111122223	1801 21st Ave	San Francisco	CA	94122
Steve Garvey	987654321	123 First Ave	Brentwood	CA	90049

CUSTOMER_ENC

Name	SSN	Address	City	State	Zip
Irwin M. Fletcher	123456789	411 Main Street	Santa Barbara	CA	93101
Josh Ritter	111122223	1801 21st Ave	San Francisco	CA	94122
Steve Garvey	987654321	123 First Ave	Brentwood	CA	90049

Figure 3: Rename CUSTOMER Table

In the next step, SafeNet creates a temporary table and exports the sensitive data to it. Notice in Figure 4 below that SSN is the only column exported to the temporary table from the original table. The Row_ID column is added automatically and used later when returning the encrypted data back to the original table. The values in the column that held the sensitive data in the CUSTOMER_ENC table (remember—the CUSTOMER table was renamed) are set to null to avoid any data conversion issues that might arise when changing the data type in a later step.

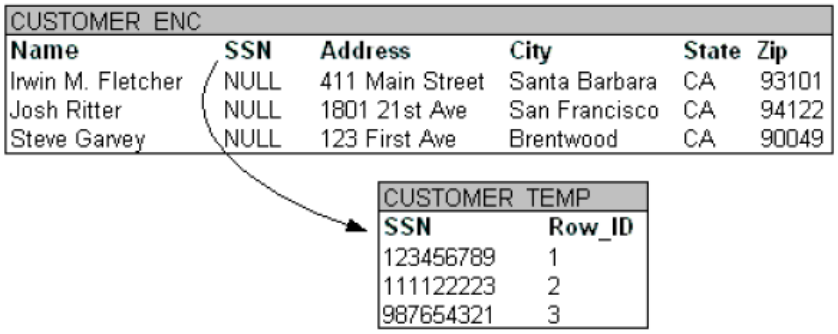

Figure 4: Export Sensitive Data to Temporary Table

Before SafeNet can populate the original column with encrypted data, it must modify the column size and data type because encrypted data is predictably larger than plaintext data and, most likely, your Social Security Numbers are stored as some sort of integer or character data type. Although SafeNet gives you the option to store your encrypted data in Base64 encoded format, it is recommended that you store your encrypted data in binary format (the default choice during configuration) since binary data is smaller than Base64 encoded data. There is also less overhead with binary data because the system does not have to encode and decode data with every encrypt or decrypt operation.[1]

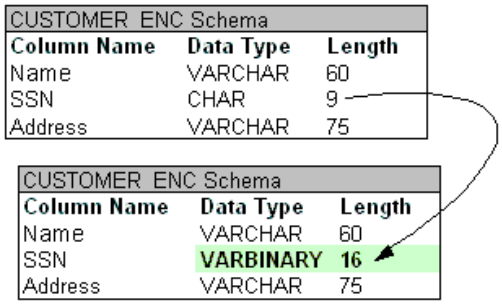

Figure 5: Modify Data Type and Column Size of the Encrypted Table

[1] The percentage by which encrypted data is larger than plaintext data varies depending on the encryption algorithm that is used to perform the encrypt operation. The examples in this document were created with 168-bit DESede ciphers in CBC mode with PKCS5 padding; as such, a nine-digit Social Security Number expands to 16 bytes of binary data.

Once the column is modified, SafeNet can migrate the sensitive data back into the CUSTOMER_ENC table. Before this happens, however, the NAE Connector sends the data to the NAE Server, where it is encrypted. The NAE Server returns the encrypted data to the NAE Connector, which then inserts it into the CUSTOMER_ENC table.

Figure 6: Encrypt Data and Insert into CUSTOMER_ENC Table

After encryption, the CUSTOMER_TEMP table is dropped, and the CUSTOMER_ENC table might look something like this[2]:

CUSTOMER_ENC

Name	SSN	Address	City	State	Zip
Irwin M. Fletcher	0xEED95DB775158895...	411 Main Street	Santa Barbara	CA	93101
Josh Ritter	0x21010B370F8752D5...	1801 21st Avenue	San Francisco	CA	94122
Steve Garvey	0xC5187FC3A3286B7F...	123 First Avenue	Brentwood	CA	90049

Figure 7: Sensitive Data is Stored in Encrypted Format

How Do You Automate Subsequent Updates and Inserts?

Once your table and column are able to accommodate encrypted data, SafeNet automates encryption and decryption of data by creating a view, triggers, and stored procedures that are generated during configuration to work with the NAE Connector. In this way, properly authenticated applications outside the database can continue to query and update the same database tables as before. The NAE Connector remains transparent to outside applications, and, more importantly, the number of code changes necessary to integrate the NAE Connector are minimal.

[2] The ciphertexts shown in this document are displayed in hex; as such, 32 characters are used to represent 16 bytes of binary data.

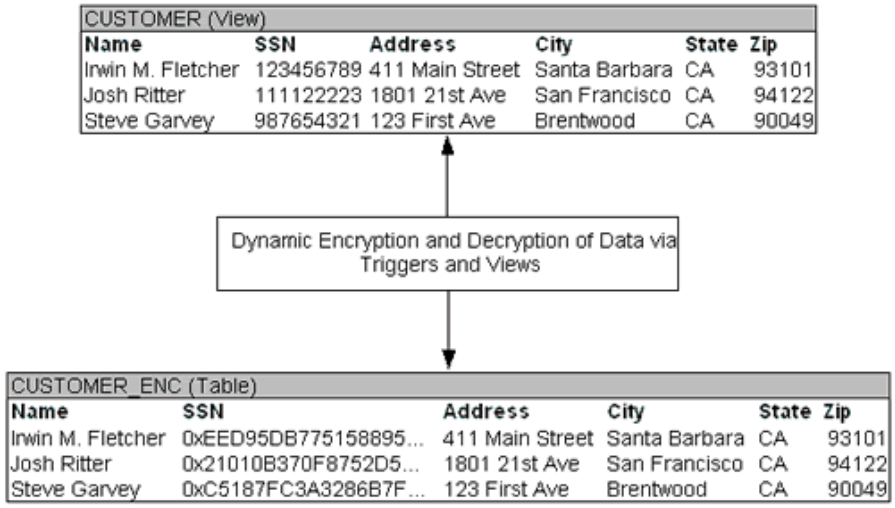

Figure 8: View is Automatically Instantiated

As you can see from Figure 8 above, when sensitive data is accessed, the view is instantiated by the database and populated with decrypted data from the CUSTOMER_ENC table. Because the view has the same name as the original table, all SQL statements that reference the encrypted data can function regularly without modification.

Likewise, triggers trap all of the inserts and updates executed on the view. If an insert statement is detected, a new insert statement is generated based on the original insert values. The Social Security Number, in this case, is encrypted before insertion into the base table. Similarly, when an update statement is executed, a new update statement is generated to update the base table (CUSTOMER_ENC) as opposed to the view referenced in the call from the outside application (CUSTOMER).

In Summary

As outlined above, the process to migrate plaintext data to encrypted format is quite simple when using the NAE Connector. Furthermore, the process can be completely automated through the use of triggers, views, and stored procedures, all of which are created and installed by the DataSecure appliance during configuration. What's most important is that the integration is completely transparent to applications that interface with your sensitive data. Before deploying DataSecure, your sensitive data sits in the clear in your databases, as shown below.

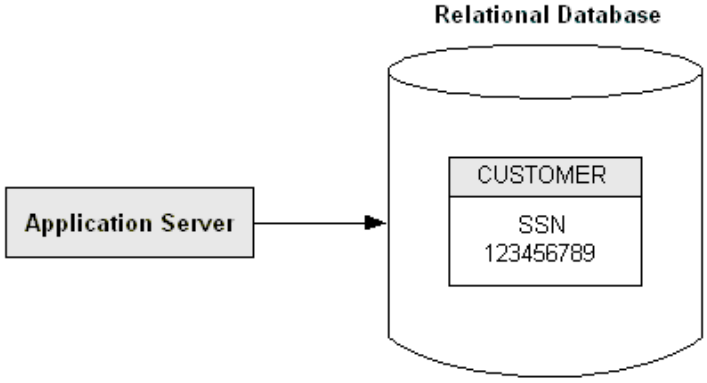

Figure 9: App Server Interacts with Plaintext Data before Deploying DataSecure

As the figure below illustrates, after deploying DataSecure, your sensitive data is encrypted and applications can continue interacting with sensitive data using the same SQL statements. However, instead of interacting directly with that sensitive data, the application servers are actually interacting with a view of the data.

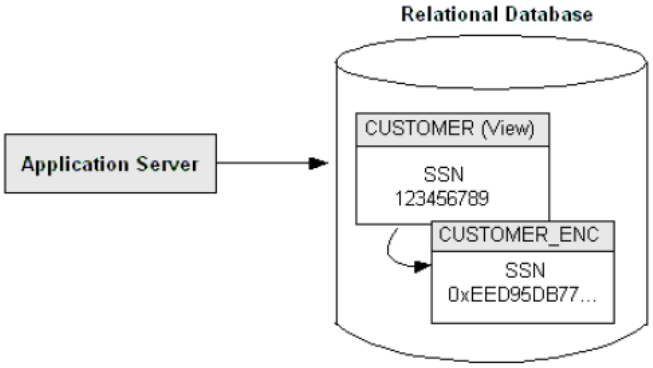

Figure 10: App Server Interacts with Plaintext View of Encrypted Data after Deploying DataSecure

About SafeNet, Inc.

SafeNet, Inc. is a global leader in information security. Founded 25 years ago, the company provides complete security utilizing its encryption technologies to protect communications, intellectual property and digital identities, and offers a full spectrum of products including hardware, software, and chips. UBS, Nokia, Fujitsu, Hitachi, Bank of America, Adobe, Cisco Systems, Microsoft, Samsung, Texas Instruments, the U.S. Departments of Defense and Homeland Security, the U.S. Internal Revenue Service, and scores of other customers entrust their security needs to SafeNet. In 2007, SafeNet was taken private by Vector Capital. For more information, visit http://www.safenet-inc.com.

Contact Information

Corporate Headquarters
4690 Millennium Drive, Belcamp, Maryland 21017 USA
Tel.: +1 410 931 7500 or 800 533 3958, Fax: +1 410 931 7524
E-mail: info@safenet-inc.com

EMEA Headquarters
Tel.: + 44 (0) 1276 608 000
E-mail: info.emea@safenet-inc.com

APAC Headquarters
Tel: +852 3157 7111
E-mail: info.apac@safenet-inc.com

For all office locations and contact information, please visit
www.safenet-inc.com/company/contact.asp

06/2008