SQrazorLoc

SQrazorLoc is a product designed to protect sensitive data, such as credit card numbers or national identification numbers (for example, U.S. social security numbers), stored in SQL databases. SQrazorLoc allows clients to encrypt sensitive data inside client applications, which provides an additional level of security since the data is encrypted in all messaging between the client and the SQL server. Also, if the SQL server is managed by a party different than the party who owns the data, such as in a cloud application, SQrazorLoc provides separation between those who own the data and can view it, and those who manage the data but should have no access.

SQrazorLoc separates the ownership of the configuration used for the database encryption from the database and thus provides the means to delegate authority for database protection. This separation also provides another layer of protection by forcing any attacker to simultaneously hit multiple targets at multiple locations in order to gain access to all that is needed to illicitly access the protected database.

SQrazorLoc also ensures that the protected data is still able to be searched. Embedded into SQrazorLoc is a CYPHYX patented technology called DARE (Dynamic And Random Encryption). DARE is the process by which industry accepted encryption algorithms are used to implement encryption and yet maintain a keyless environment and to ensure that the encryption cannot be defeated and remains in tact regardless of how it is attacked. When SQrazorLoc encrypts data, DARE ensures that no two encryptions ever use the same key value in that process. As such each time the same identical piece of data is encrypted using DARE the resulting product is different. This makes it impossible to use for a search value since encrypting a value of “123” multiple times would result in multiple different encrypted results. Because of this SQrazorLoc implements a method called “tokenization” to create a non-reversable, always consistent value as a companion to the encrypted value. This means that the token for a given value would always compute to the same result. This provides the means to take the value being searched, tokenize it, and then use it to search the existing tokens for the column in question to find the result.

SQrazorLoc makes encryption transparent to applications. An SQrazorLoc-enabled driver installed on the client computer achieves this by automatically encrypting and decrypting sensitive data in the client application. The driver encrypts the data in sensitive columns before passing the data to the Database Engine, and automatically rewrites queries so that the semantics to the application are preserved. Similarly, the driver transparently decrypts data, stored in encrypted database columns, contained in query results.

The management of the driver installed at each client computer is made simple through the use of the CYPHYX Customer Portal.

CYPHYX Customer Portal

The CYPHYX Customer Portal (Portal) is the gateway to many services provided by our product including the configuration of subscriptions for products such as SQrazorLoc. The Portal is used to create a customer record, purchase and configure product subscriptions, and to access the configuration backups, product performance data, product alerts, product logs, and product troubleshooting data.

At the Portal the customer can also check on alerts, such as illicit access attempts to SQrazorLoc or attempted connections from illicit clients that are not recognized as part of the customers installed base of SQrazorLoc installations. The Portal is also the location for performance data to gauge how well clients are able to access the database and which applications are making the heaviest demands. Another performance measurement available at the Portal is the individual response time measurements for each encryption, decryption, and tokenization that is taking place on a given client connecting to a specific database. All of this information is able to be viewed for a specific client, group of clients, or all clients and can also be unified into averages across an entire database.

Also, at the portal is where the customer configures the products purchased and can assign which elements in a database are protected. The protection can be activated or deactivated for a given element. In defining the protection for a given element the customer can specify if the element is searchable and thus requires tokenization. They can also indicate if the element has specific configuration parameters predefined or if those parameters are left to random selection. The embedded technology DARE utilizes many components in the configuration to construct random value processing steps utilized to reconstruct an environment when performing decryption. Since these random values are in flux the configuration used to construct them becomes a critical component for processing. This is also why the configuration is backed up and why the customer is urged to backup and download for storage in a secure device or location a copy of the configuration in use.

The Portal is also where the customer goes to download the specific product packages, such as SQrazorLoc, as part of the installation process. These packages contain the necessary software that needs to be installed on each client that will access the database as part of the subscription. The package has two parts, the infrastructure component and the product component. The infrastructure component only needs to be installed once for a given client and then can be used by all CYPHYX products installed on that client. The product component is separate and for SQrazorLoc only needs to be installed once for a given client no matter how many protected databases that client will access. Since the SQrazorLoc product is an SQL driver on the client, the client applications that will access databases can utilize this same SQrazorLoc driver to access all databases for which the client computer has been configured at the Portal during the installation process.

Installation Process

During the installation process we ask for basic information about who the customer is and where are they located. This provides the basis from which users are assigned and subscriptions for different products are created. A subscription for SQrazorLoc can be selected and this leads to a link from which the SQrazorLoc package can be downloaded. The same SQrazorLoc package downloaded here can be installed on each for the clients that are intended to be included under this subscription.

Once the SQrazorLoc package is installed on a client computer the package then logs identifying information such as the client computer name, IP address, and operating system parameters that assist with identifying this client for the customer. Also, as part of installing the SQrazorLoc package, the SQrazorLoc driver should be identified in the Data Source list for the use by the applications replacing the existing driver for access to the SQL database. The SQrazorLoc driver supports existing databases without protection and handles all SQL messages from the application the same as the standard Microsoft driver. This allows the SQrazorLoc driver to be staged into place on the client between the application and the database prior to protection being activated at a later point.

As each client is installed and identified in the Portal client page, the customer can then perform a verification to provide access to the client for the SQrazorLoc product and to assign the client to the subscription. At this point the customer only needs to identify the database for which the subscription will be used and assign it to the subscription. The next step is to configure the protection for the database, stage the configuration, and then select to implement it, which is all part of the Database Conversion Process.

Database Conversion Process

With the subscription assigned to a database and the SQrazorLoc package installed on all clients that will use this database, we are ready to configure the protection and begin the database conversion to protected mode. During the assignment of the database to the subscription, in the background, the schema of the database is retrieved to build configuration data to provide the selection of those elements that will be protected and other configurable features of that protection.

Individual elements can be configured to have protection turned on or off, made searchable or non-searchable, performance tracked made active or inactive, and access metrics on or off. This can be done manually at the element level by a user or defined as part of an existing template that configures all the necessary elements for a known application's database schema.

As data elements are being configured the Portal will also draw the customer's attention to any stored procedures, foreign keys, Indexes, or other Database Objects that involve data elements for which protection has been configured. For instance, in regards to stored procedures suggested updates to those procedures will be made to handle the change in columns where the data will be stored. Additionally, the driver configuration will be updated to handle normal processing for times when the application will call those stored procedures.

Once all data elements for which protection is desired are configured and all other database objects that involve those data elements have been addressed, the customer can then activate the protection via the subscription main settings. On activation, an automated conversion process begins for those elements that have protection configured where two columns are added; one to contain the protected (encrypted) version of the data and the other to contain the tokenized version of the data for those elements configured as searchable. This is done while leaving the original column for this data element in place to provide the needed space for conversion and if desired at a later time, deconversion.

Social Security #
344-77-6545

Social Security #	Social Security # Protected	Social Security # Tokenized
344-77-6545	324E6B76AA8772	FF6567EA342769

In the current version of SQrazorLoc this conversion process is done while the database is held offline to ensure a quick and efficient process. In the next version of SQrazorLoc there will be a choice to perform the conversion process while the database remains available and this is possible because the SQrazorLoc driver is already handling the database requests prior to the conversion process.

As the conversion completes the original unprotected column for a data element is cleared leaving the original data encrypted in the Protected column and the tokenized in the Tokenized column. At completion of the conversion the database is brought back online and an alert is logged to indicate the completion and availability of the database so that normal processing can start.

Normal Processing

During normal operation the applications that access the database make requests just as they did prior to the installation of SQrazorLoc and the responses the applications receive back also look as they did before installation. This is possible because the SQrazorLoc driver performs a rewrite of the SQL commands as they are received from authenticated applications prior to the request going to the SQL server. For example, those data elements in the request that are protected are redirected to the protected columns and if a protected data element is in the WHERE clause condition it is redirected to the token column and the value used for the search is tokenized to encode it for use when comparing to the values in the token column.

The same occurs on the response from the SQL server with those data elements that were protected being decrypted and redirected back to the original data element as expected by the applications. This occurs for INSERT commands, UPDATE commands, and DELETE commands as well.

During UPDATE commands, when protected data elements are part of the command, the data is encrypted by the SQrazorLoc driver and even if the same value that is already in this specific data element is used the protected value created for the data element will be different. This is due to how DARE functions in the SQrazorLoc driver.

More importantly, this means that key changes, as are done with competing products and mandated by certain regulatory requirements such as PCI, HIPAA, and others, are constantly occurring as protected data is updated back into the database. This resolves the need to perform quarterly key changes and allows the customer to be ahead of the curve as regulations are currently changing to require these key changes to occur monthly and eventually weekly. With SQrazorLoc keys are constantly in flux and so key changes no longer need to be planned or performed.

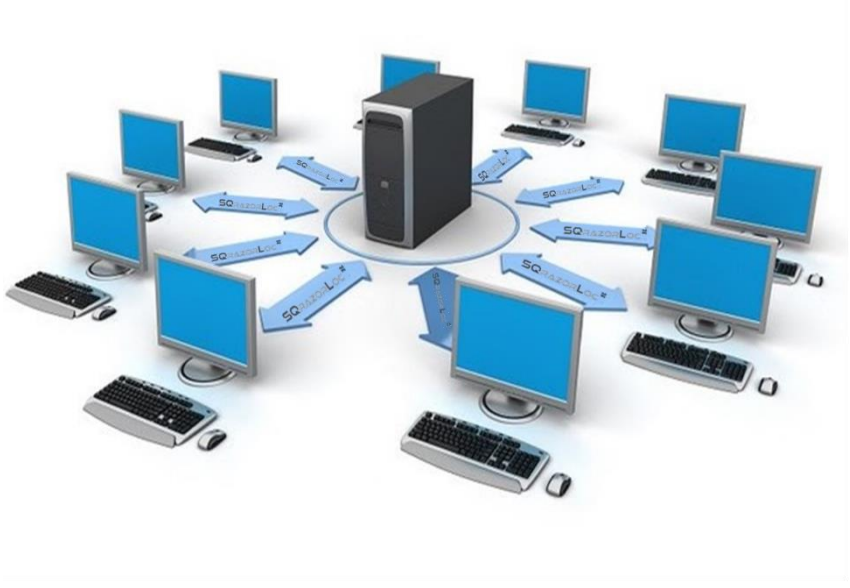
During normal processing statistics and product status updates are logged during each operation, which can be viewed on the Portal. Additionally if an illicit attempt is made to access the protected database by an unknown application that was not authenticated as having access by the customer, an alert is logged and all details about the attempt are saved to provide necessary information if the customer wishes to investigate the application that made the attempt. As always, no customer data is involved in any logging or statistical measurements that are uploaded to the Portal. Absolute security is always the first concern for CYPHYX.

Also, during normal processing, SQrazorLoc is constantly journaling any changes to the environment within the driver that is used for processing the protection of data elements. This journaling is also backed up to the Portal to maintain a secure off-premises copy that can be used for recovery if ever needed.

Configuration Protection and Recovery

The architecture of SQrazorLoc and CYPHYX products in general provides a diversified and secured setup that makes an attack a difficult and ultimately fruitless venture. This same architecture also provides protection against hardware faults making them much easier to handle and providing backed up configuration at an off-premises location that is protected from the event. For most competing database protection products, a hardware fault that corrupts the client computer or database server results in a recovery that is resource intensive.

With the architecture of SQrazorLoc there are a number of advantages that make recovery a much easier process. Each client computer utilizes its own installed copy of the SQrazorLoc driver to connect to the database. As such the client computers are operating outside the database server and thus if corruption or a malware attack occurs at any client computer or group of client computers the data remains safe and since SQrazorLoc has off-premises backup of the configuration that was in use at any client, the recovery is only a matter of restoring the client computer and then reinstalling SQrazorLoc to it in order for the client to be brought back online to the database. The data is not able to be corrupted since even if the attack or faults that causes the corruption occurs during an encryption or decryption the database request would not occur and thus the data in the protected data element in the database would remain untouched and accessible when access by the client is restored.



If the corruption or attack occurs at the database server then the only issue is access by the client computers and whether the database server is able to respond to requests. CYPHYX strongly recommends regular database backups as protection from server faults or database server attacks in order to aid in recovery. For SQrazorLoc the encryption and decryption occur at the clients and not at the database server, so if the database server is attacked or crashes the data remains in a consistent state and only the damage done by the attack or crash is in need of recovery from the most recent backup. If the encryption for a given data element were to be in question, SQrazorLoc maintains a journal that would allow for reconstruction of the environment used for encryption to allow for recovery of that data if it is only corruption of the encryption and not of the data itself.

The journaling that SQrazorLoc performs is a multi-part configuration just as was in use during the encryption at the time and requires components from the original client that are backed up at the Portal and from the Portal configuration for the subscription itself. This ensures that no single point attack would allow someone to illicitly obtain the needed information to remove encryption from a data element's individual piece of data. Each item within a given data element, when protected, is encrypted separately and would require access to multiple pieces of configuration data to allow someone to remove encryption as would be done during a special recovery process, so an attack in this manner is not feasible or possible to accomplish and secure the journaling feature of SQrazorLoc. As always, no customer data is ever involved in this journaling and no customer data ever leaves the database server.

Database Deconversion and Uninstallation Process

At CYPHYX we understand that there could be situations and moments when protection needs to be removed whether for a customer specific need or because the customer is moving in a new direction and no longer needs the product. Because of this fact we have a fully automated process for deconverting the database and for uninstalling SQrazorLoc.

SQrazorLoc deconversion is initiated through the Portal and the deactivation of protection for all data elements for which protection has been configured. The customer can deactivate the protection all at once or go through and individual change each protected data element to unprotected. Once the change to the configuration is complete the customer can then initiate the deconversion through the subscription main page just as was done when converting to use of SQrazorLoc.

At present the deconversion occurs while the database is taken offline and is inaccessible until the deconversion completes. Once the deconversion is complete the customer is notified of the database returning to an online state and is immediately useable by the applications.

If the customer wishes to completely uninstall the SQrazorLoc product it is only a matter of uninstalling the SQrazorLoc driver from each client computer and returning to use of the standard Microsoft driver for database access. Lastly, the removal of the additional columns that were added to the tables in the database can be done through a table recreation and copy procedure that will restore the tables to their original form prior to the installation of SQrazorLoc and without the loss of any data. It is recommended that this procedure be done with the database offline until it completes.

If you have any questions, please contact us

With SQrazorLoc we realize that much of what we are doing will change the way database protection is sold and used, so if anything contained here has raised questions, please let us know and we will gladly work to answer them.

CYPHYX
12870 Trade Way Four, Suite 107 #665
Bonita Springs, Fl. 34135
(888) 871-3273

Email: sqrazorloc@cyphyx.com