PCI Requirements Checklist – General Data Protection

How EncryptRIGHT Helps with PCI Data Security Compliance

The Payment Card Industry (PCI) Data Security Standard (DSS) is a comprehensive security standard that includes requirements for security management policies, procedures, network architecture, software design, and other critical protective measures. While the standard was designed for bank card information security, the recommendations apply to any general data security needs. Any organization that stores or processes confidential data — and that includes just about everyone in the payments space — could use the PCI security requirements as a guide to establish their data protection policies and procedures.

EncryptRIGHT complies with version 3.1 of the PCI DSS, which recommends 12 broad PCI security requirements ranging from firewalls and physical access to software and data security systems, as well as documented security procedures. Of the 12 PCI security requirements, eight address encryption and key management, and EncryptRIGHT can help meet compliance for each of these eight requirements (**in bold below**):

1. Install and maintain a firewall configuration to protect cardholder data.
2. **Do not use vendor-supplied defaults for system passwords and other security parameters.**
3. **Protect stored cardholder data.**
4. **Encrypt transmission of cardholder data across open, public networks.**
5. Protect all systems against malware and regularly update anti-virus software or programs.
6. **Develop and maintain secure systems and applications.**
7. **Restrict access to cardholder data by business need-to-know.**
8. **Identify and authenticate access to system components.**
9. Restrict physical access to cardholder data.
10. **Track and monitor all access to network resources and cardholder data.**
11. Regularly test security systems and processes.
12. **Maintain a policy that addresses information security for all personnel.**

Following is a requirement-by-requirement overview of how EncryptRIGHT helps to address specific encryption and key management requirements:

PCI Requirement #2

Do not use vendor-supplied defaults for system passwords and other security parameters.

Requirements	EncryptRIGHT Compliance
2.1 Do not use vendor-supplied defaults for system passwords and other security parameters.	EncryptRIGHT does not have a default User ID or password. On installation a user ID and password must be created and given administrative rights before installation can continue.
2.2.3 Implement additional security features for any required services, protocols, or daemons that are considered to be insecure. **2.3** Encrypt all non-console administrative access using strong cryptography.	EncryptRIGHT provides strong cryptography and key management to protect your data. This includes strong hashing (SHA 2 and SHA 3 256/384/512) and encryption (Triple DES 2/3 key, and AES 128/192/256).

PCI Requirement #3

Protect stored cardholder data.

Requirements	EncryptRIGHT Compliance
3.3 Mask PAN when displayed (the first six and last four digits are the maximum number of digits to be displayed).	EncryptRIGHT provides the capability to define a group of users that only have rights to view a masked portion of a data field. This includes applications that decrypt the data, the EncryptRIGHT audit log, and EncryptRIGHT trace files.
3.4 Render PAN, at minimum, unreadable anywhere it is stored (including on portable digital media, backup media, in logs).	EncryptRIGHT provides strong cryptography and key management to protect your data. This includes strong hashing (SHA-2 and SHA-3 256/384/512) and encryption (Triple DES 2/3 key, and AES 128/192/256).
3.5 Protect cryptographic keys used for encryption of cardholder data against both disclosure and misuse.	EncryptRIGHT requires a User ID and password for all users. Each user is assigned to one (or more) groups. Each group is assigned the types of access within the EncryptRIGHT administration application. This includes the ability to view or maintain keys and other information. In addition EncryptRIGHT allows you to require a quorum of logged on users for administration functions. EncryptRIGHT, when used with an HSM, will store the KEK used to encrypt keys stored within the EncryptRIGHT database within the hardware system. Keys may optionally be stored on the server only and fetched when needed by client computers.
3.6 Fully document and implement all key-management processes and procedures for cryptographic keys used for encryption of cardholder data. **3.6.1** Generate of strong cryptographic keys.	EncryptRIGHT supports Triple DES 2/3 keys. AES 128, 192 and 256. RSA public keys from 1024 to 4096 bits.
3.6.2 Secure cryptographic key distribution.	Secure key distribution is provided in EncryptRIGHT when the client/server option is licensed. This provides for a server where keys are maintained, and one or more client computers connected through a TCP/IP network. Each client computer is registered to the server and the server automatically distributes key changes to the clients. All key distributions are double encrypted during transmission, once using an EncryptRIGHT client key, and once using TLS communications encryption.
3.6.3 Secure cryptographic key storage.	EncryptRIGHT hashes and encrypts all records within the EncryptRIGHT security database using a randomly generated local master key. The master key is protected by the software license which is restricted to licensed computers only. Additionally, if an HSM is in use by EncryptRIGHT, the LMK is encrypted using a KEK hardware key.
3.6.4 Cryptographic key changes for keys that have reached the end of their cryptoperiod (for example, after a defined period of time has passed and/or after a certain amount of cipher-text has been produced by a given key), as defined by the associated application vendor or key owner, and based on industry best practices and guidelines.	EncryptRIGHT allows you to change key values manually or automatically. Automatic key changes can be set to occur on a flexible schedule from 1-99 days, weeks, months, quarters, or years.

Requirements	EncryptRIGHT Compliance
3.6.5 Retirement or replacement (for example, archiving, destruction, and/or revocation) of keys as deemed necessary when the integrity of the key has been weakened (for example, departure of an employee with knowledge of a clear-text key), or keys are suspected of being compromised.	EncryptRIGHT allows you to designate when a key can be destroyed. The choices are **never** (for long term data storage), **on expiration**, or **expiration plus** 1 to 99 days, weeks, months, quarters or years. Any key that becomes compromised or invalid can be marked as revoked. A new value can easily be created. Every production key value is assigned a unique key version number. When a specific key value is marked as revoked or is otherwise superseded, a new version is created. Both keys still exist in the database, so if an application attempts to use a revoked key, they will receive an indication that the key is revoked.
3.6.6 If manual clear-text cryptographic key management operations are used, these operations must be managed using split knowledge and dual control (for example, requiring two or three people, each knowing only their own key component, to reconstruct the whole key).	EncryptRIGHT provides the ability for up to 3 different people to enter key components. Additionally, you can require that the components must be entered by different users. You also have the ability to require a quorum of users to be logged on. For example you can require that two people with rights for key management be logged on at the same time to enter a key component, where one person enters the key and the other observes.
3.6.7 Prevent unauthorized substitution of cryptographic keys.	Key values cannot be substituted by unauthorized people. Every user should be assigned a unique user ID and password. Every key change is logged in the EncryptRIGHT secure audit log. When a key is selected for use, its check value is validated before using the key to secure or unsecure data.

PCI Requirement #4

Encrypt transmission of cardholder data across open, public networks.

EncryptRIGHT uses TLS 1.2 for secure communications between the EncryptRIGHT servers and client installations. For secure synchronization of keys, user information, data policies, etc., additional encryption of sensitive information is employed at the field level (such as key material and passwords). EncryptRIGHT also uses TLS 1.2 for secure browser sessions used with the EncryptRIGHT administration web-based application.

Although this requirement talks about using SSL and TLS security over networks, it is always better to also encrypt sensitive data from application to application so the data is not only protected during transmission, but also between transmission and the application that will process the data. For example, when you send an un-encrypted file through a secure network, the file may temporarily reside on the receiving computer until some process uses the data. During this time someone could access or even change the data without your knowledge. Use EncryptRIGHT to protect this data until the receiving application can process the data.

PCI Requirement #6

Develop and maintain secure systems and applications.

Requirements	EncryptRIGHT Compliance
6.3 Develop internal and external software applications (including web based administrative access to applications) securely.	EncryptRIGHT uses TLS 1.2 for secure browser sessions used with the EncryptRIGHT administration web-based application. For secure protection of user interface selections, each session has a unique session number and session key. This key is used to encrypt HTML selection field information to tie each to the specific user, session and item being maintained.
6.5.3 Insecure cryptographic storage.	EncryptRIGHT uses shared memory on computer systems to increase throughput and efficiency. All sensitive information is encrypted (user IDs, passwords, names, descriptions, key material). In addition data in shared memory is authenticated using the database LMK to detect unauthorized changes.
6.5.4 Insecure communications.	EncryptRIGHT uses TLS 1.2 for all communications.
6.5.5 Improper error handling.	EncryptRIGHT audit logs and trace files do not contain sensitive information (key values or passwords). In addition, customer data policy fields may be marked as sensitive. Sensitive field data will not be displayed in trace files.

PCI Requirement #7

Restrict access to cardholder data by business need-to-know.

EncryptRIGHT provides the ability to define the level of data access for each user. Access can be specified at the record and/or data field level. The type of access that can be defined is no access, masked access (specifying the mask character, the position and length of the clear data), read only or read/write. By default no access is allowed for anyone when creating record definitions. In order to use a definition you must provide access definitions.

Requirements	EncryptRIGHT Compliance
7.2.3 Default “deny all” setting.	At the moment of installation of EncryptRIGHT using the web-based administration application, access is restricted to the first IP address used. From there you may expand access as needed. The installation user will define the first administrative User ID and password for the installation. Access from the web-based administration application is initially restricted to only the installation directory and the temp directory on the host computer system. Additionally, directories may be added as needed.

PCI Requirement #8

Assign a unique ID to each person with computer access.

Requirements	EncryptRIGHT Compliance
8.1.1 Assign all users a unique ID before allowing them to access system components or cardholder data.	EncryptRIGHT allows for all users to have a unique User ID and password. This is used not only for access to the EncryptRIGHT database, but also for access to cryptography for your defined user data records and fields.

Requirements	EncryptRIGHT Compliance
8.1.2 Control addition, deletion, and modification of user IDs, credentials, and other identifier objects.	Only user administrators can add, delete and change users. In addition this can be further controlled by requiring a quorum of logged on users so no single person can maintain user information without supervision.
8.1.3 Immediately revoke access for any terminated users. **8.1.4** Remove/disable inactive user accounts at least every 90 days. **8.1.5** Enable accounts used by vendors for remote maintenance only during the time period needed.	Users can be temporarily disabled or entirely deleted by user administrators at any time.
8.1.6 Limit repeated access attempts by locking out the user ID after not more than six attempts. **8.1.7** Set the lockout duration to 30 minutes or until administrator enables the user ID. **8.1.8** If a session has been idle for more than 15 minutes, require the user to re-authenticate to re-activate the session. **8.2.3** Require a minimum password length of at least seven characters. Use passwords containing both numeric and alphabetic characters. **8.2.4** Change user passwords at least every 90 days. **8.2.5** Do not allow an individual to submit a new password that is the same as any of the last four passwords he or she has used.	EncryptRIGHT provides a password policy that can specify the following: • Maximum and minimum password age. • Minimum password length (default is 7 characters). • Password uniqueness. This includes how many passwords to remember, the maximum characters that can be repeated from the previous password, and a password dictionary. • Required special characters, upper/lower case and numbers (default is both numeric and alphabetic characters). • Bad attempts lockout count, and the duration of the lockout (default is 6 attempts, 30-minute lockout, and 30 minutes to reset). EncryptRIGHT allows you to set the automatic timeout period for the web-based administration application. The default is 15 minutes.
8.3 Incorporate two-factor authentication for remote network access.	EncryptRIGHT allows you to use two-factor authentication applications supporting TOTP (Time-based One Time Passwords) and HOTP (HMAC-based One Time Passwords), such as Google Authenticator.
8.4 Render all passwords unreadable during transmission and storage on all system components using strong cryptography.	Passwords are stored self-encrypted. This means that they are encrypted using a key derived from the password itself. These passwords cannot be decrypted. The only way to verify a password is to enter the same password, which is encrypted and then compared to the original encrypted password.
8.5.1 Control addition, deletion, and modification of user IDs, credentials, and other identifier objects.	Only user administrators can add, delete and change users. In addition this can be further controlled by requiring a quorum of logged on users so no single person can maintain user information without supervision.
8.5.2 Verify user identity before performing password resets.	Passwords can only be changed by a user if the user also enters the current password.
8.5.3 Set first-time passwords to a unique value for each user and change immediately after the first use.	EncryptRIGHT defaults to forcing the user to change their password anytime an administrator sets or changes a user's password.

Requirements	EncryptRIGHT Compliance
8.5.4 Immediately revoke access for any terminated users. **8.5.5** Remove/disable inactive user accounts at least every 90 days. **8.5.6** Enable accounts used by vendors for remote maintenance only during the time period needed.	Users can be temporarily disabled or entirely deleted by user administrators at any time.
8.5.9 Change user passwords at least every 90 days. **8.5.10** Require a minimum password length of at least seven characters. **8.5.11** Use passwords containing both numeric and alphabetic characters. **8.5.12** Do not allow an individual to submit a new password that is the same as any of the last four passwords he or she has used. **8.5.13** Limit repeated access attempts by locking out the user ID after not more than six attempts. **8.5.14** Set the lockout duration to 30 minutes or until administrator enables the user ID.	EncryptRIGHT provides a password policy that can specify the following: • Maximum and minimum password age. • Minimum password length. • Password uniqueness. This includes how many passwords to remember, the maximum characters that can be repeated from the previous password, and a password dictionary. • Required special characters, upper/lower case and numbers. • Bad attempts lockout count, and the duration of the lockout.
8.5.16 Authenticate all access to any database containing cardholder data. This includes access by applications, administrators, and all other users.	EncryptRIGHT can optionally require a User ID and password for API use. This means that in order for an application to use the EncryptRIGHT library to encrypt any data, a User ID and password has to be given to the library for verification. With this, EncryptRIGHT fetches the access rights the user has. When a data record definition in EncryptRIGHT is used to encrypt or decrypt data, the user's access rights are checked to determine if they can decrypt data, and if so, do they only see a masked version of the data, or can the view or update the data.

PCI Requirement #10

Track and monitor all access to network resources and cardholder data.

Requirements	EncryptRIGHT Compliance
10.1 Implement audit trails to link all access to system components to each individual user.	EncryptRIGHT allows for all users to have a unique User ID and password.
10.2 Implement automated audit trails for all system components to reconstruct access events.	EncryptRIGHT provides the ability to create a secure audit log entries for: • Access to data down to the field level. • Invalid access attempts. • All changes made to definitions within EncryptRIGHT. • User logons and logoffs (optional).

Requirements	EncryptRIGHT Compliance
10.3 Record at least the following audit trail entries for all system components for each event: user identification, type of event, date and time, success or failure indication, origination of event, and identity or name of affected data, system component or resource.	The EncryptRIGHT audit log contains the User ID, the type of message, date and time, the EncryptRIGHT subsystem name and the success or failure message.
10.5 Secure audit trails so they cannot be altered.	The EncryptRIGHT audit log is secure. Each entry contains a sequence number and the entire entry is hashed and encrypted. When viewing the log, each entry hash and sequence number is verified. The ability to view the audit log can be assigned to groups of users.

PCI Requirement #12

Maintain a policy that addresses information security for all personnel.

Requirements	EncryptRIGHT Compliance
12.10.4 Include alerts from security monitoring systems.	EncryptRIGHT may be configured to provide automatic notifications for various events. These notifications are via a script the customer writes, which may include sending an e-mail or SMS message. Events include: • Bad password attempts (configurable threshold). • Client update synchronization errors. • EncryptRIGHT database scan errors. • Report of keys about to expire. • EncryptRIGHT API function errors (in customer applications). • A system heartbeat message • User defined notifications via an API call.

PCI-ER-20200212