Security Standards Council®

Guideline: Tokenization Product Security Guidelines

Version: 1.0

Date: April 2015

Author: PCI Security Standards Council

Tokenization Product Security Guidelines –
Irreversible and Reversible Tokens

Table of Contents

Introduction

With a rising demand for tokenization products, the PCI Security Standards Council (PCI SSC) believes it is imperative to build, test, and deploy products that provide strong support for compliance with the *PCI Data Security Standard* (PCI DSS). With this aim, the Council has produced these technical guidelines for evaluating tokenization products that replace the primary account number (PAN) with a surrogate value called a "token". The security and robustness of a tokenization system relies on many factors, including the configuration of different components, the overall implementation, and the availability and functionality of specific security features for each product. A tokenization product can be a hardware device, such as an appliance, a software application, and/or a service offering.

The security objective of a tokenization process is to ensure the resulting token has no value to an attacker (see Annex G – Formal Security Objective of a Tokenization Product). When evaluating a tokenization system, it is important to consider all elements of the overall tokenization implementation. These elements include the technologies and mechanisms used to capture cardholder data, how a transaction moves through the entity's environment, the transmission from the point-of-capture (e.g., point-of-sale system) to the authorization endpoint, how tokens are retained for use (e.g. in back office systems) and so on. The tokenization implementation should also address potential attack vectors against each component and provide the ability to confirm with confidence the mitigation of associated risks.

A token, as described in these guidelines, replaces a PAN with a surrogate value. The token can be stored in lieu of a PAN, reducing the risk of unauthorized disclosure of a PAN.

This document, *Tokenization Product Security Guidelines*, provides best practices for "acquiring tokens," which are defined as:

> Tokens created by the acquirer, merchant, or a merchant's service provider. This token is created after the cardholder presents their payment credentials. Acquiring tokens may be used as part of the authorization process, including card-on-file transactions.

The General Guidelines/Best Practices statements are intended for *all* types of token-generation methods, and there are also specific Guidelines/Best Practices for irreversible and reversible tokens. This document also describes different classifications of tokens (i.e., tokenization taxonomy), including their general use cases. This document is neutral to which approach is used by product developers and builders.

This document does not address scope of the cardholder data environment (CDE) or applicable PCI DSS requirements.

The launch of this document does not constitute a recommendation from the Council or obligate merchants, service providers, or financial institutions to purchase or deploy such products.

Intended Audience

The intended audience includes tokenization product developers, vendors and evaluators, as well as entities wishing to design and build tokenization systems and products, and entities using, or wishing to use, tokenization systems and products. These guidelines may also be applicable to any payment industry stakeholder (e.g., merchants, payment processors, acquirers, service providers, and assessors).

Intended Usage

Usage for different stakeholders may include:

- **Tokenization solution or product vendors:** May evaluate their tokenization offerings against these guidelines, allowing customers to obtain a degree of assurance about a purchase.
- **Organizations wishing to develop their own tokenization solution:** May use this document as best practices upon which they can base functional and non-functional requirements.
- **Organizations wishing to procure tokenization products and solutions:** May include these as requirements in their RFPs or other processes for evaluating tokenization products.
- **Organizations wishing to use tokenization products to reduce presence of cardholder data in their environment:** May use this document to evaluate that their tokens are truly independent of PANs and therefore represent a much smaller risk if compromised.
- **Independent evaluators of tokenization products (e.g., labs):** If a tokenization solution/product developer wishes to have an independent evaluation of their product/solution, the evaluator may use this document to evaluate.

Terminology

As stated above, the guidelines in this document are intended for use with acquiring tokens.

Tokenization as used within this document is a process by which a surrogate value, called a "token," replaces the primary account number (PAN) and, optionally, other data. The tokenization process may or may not include functionality to exchange a token for the original PAN ("de-tokenization"). The security of an individual token relies predominantly on the infeasibility of determining the original PAN knowing only the surrogate value (i.e., token).

The terms ***Informative*** and ***Normative*** are used to distinguish informational content from a security best practice or recommendation. For example: An annex marked as "informative" provides supporting material, such as samples, examples, or tutorial. An annex marked as "normative" provides further clarification of the security guidelines/best practices.

Further guidance of terms used throughout this document is provided in the Glossary, which follows the annexes at the end of the document.

Naming Convention for Guidelines/Best Practices

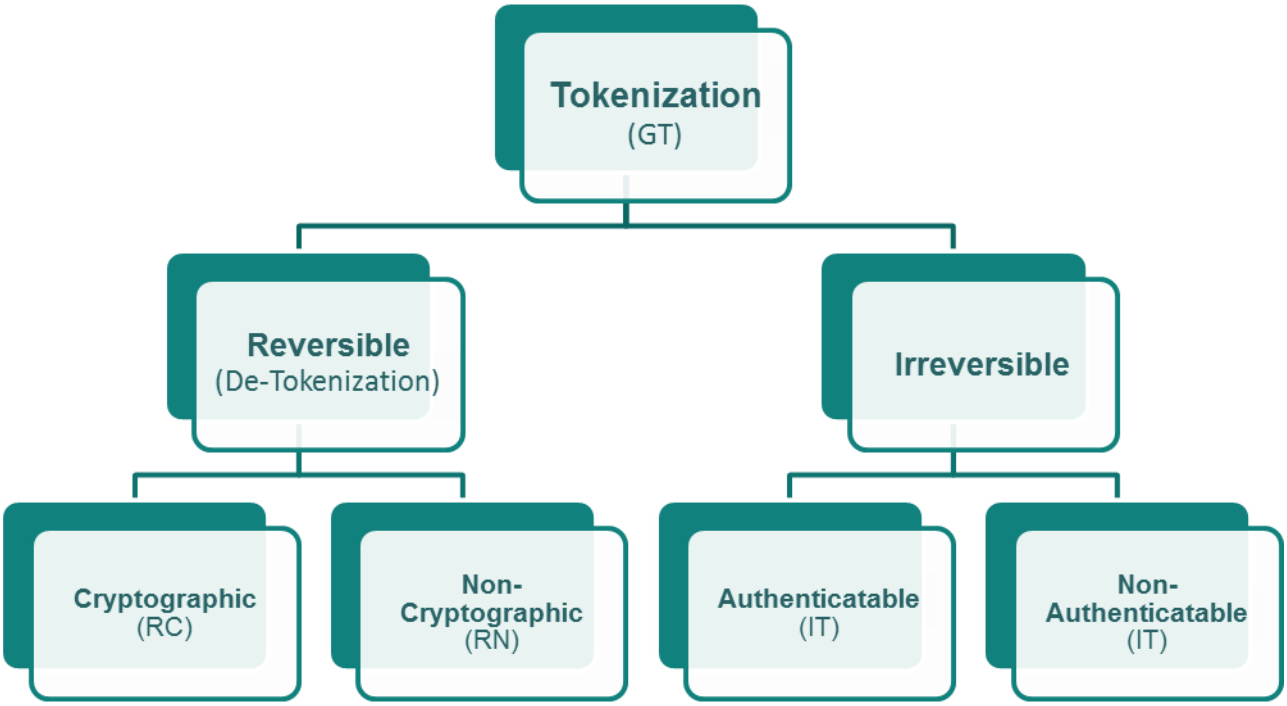
In order to logically arrange the guidelines/best practices and to reduce any ambiguity, the following naming conventions are used:

- General Tokenization guidelines have “GT” as a prefix.
- Guidelines for Irreversible Tokens have “IT” as a prefix.
- Guidelines for Reversible Cryptographic tokens have “RC” as a prefix.
- Guidelines for Reversible Non-cryptographic tokens have “RN” as a prefix.

Tokenization Classification

Figure 1 provides an overview of how the tokenization processes are classified in this document (i.e., tokenization taxonomy). Different types of tokens have differing use cases.

Figure 1: Tokenization Classification



As the figure above shows, different classes of tokens may exist; these are created through distinct mechanisms and may support different use cases. In general, tokens are either created by a mathematical process (e.g., cryptographic function) or by a non-cryptographic process (e.g., data look-up through a database function). However, this document does not preclude hybrid products using more than one classification.

Tokens and Tokenization

This section describes the different implementations of irreversible and reversible tokens. In any implementation, the token should be distinguishable from a valid PAN.

Irreversible Tokens

Irreversible tokens can never be converted back to the original PAN. It is ***not*** possible in any circumstance for ***any party*** to obtain a PAN from its irreversible token, either through analysis or from any kind of stored data extraction. Within this classification, tokens may be "authenticatable" or "non-authenticatable."

Authenticatable Irreversible Tokens

An authenticatable irreversible token is created mathematically through a one-way function that could be used to verify that a given PAN was used, but cannot be reversed to de-tokenize for the PAN. Annex E – Use Cases for Tokenization provides sample use cases.

Non-Authenticatable Irreversible Tokens

Irreversible tokens that are not authenticatable represent little to no risk for the disclosure of PAN. For instance, they can never be linked to a specific PAN, but they may be linked to a customer or account within the merchant. Annex E – Use Cases for Tokenization provides sample use cases.

Reversible Tokens

Reversible tokens provide the possibility for entities using or producing tokens to obtain the original PAN from the token. Reversible tokens have the potential to become a PAN again by the process of de-tokenization. Reversible tokens can be mapped to a unique PAN or multiple tokens may map back to the same PAN depending on technology used. If it is technically possible for a token to be de-tokenized, a product is considered to be a reversible tokenization product even if the entity producing the tokens does not intend to permit de-tokenization.

The security measures for the different approaches to reversible tokens (i.e., cryptographic and non-cryptographic) have some common high-level recommendations; at the detailed level, they require tailored criteria. For instance, regardless of whether the tokens are created cryptographically, a PAN is retrievable from its reversible token. An authorized user may obtain the original PAN from its token with a de-tokenization request through an appropriate access control mechanism.

Reversible Cryptographic Tokens

Reversible cryptographic tokens are tokens generated from PANs using strong cryptography. In this case, the PAN is never stored; only the cryptographic key is stored.[1] Annex E – Use Cases for Tokenization provides sample use cases.

[1] An exception to this is a hybrid product in which the cryptographic token is stored in a card data vault (CDV) associating it with its PAN. In this scenario, the guidelines/best practices for a non-cryptographic token also apply.

Reversible Non-Cryptographic Tokens

For reversible non-cryptographic tokens, obtaining the PAN from its token is only by a data look-up in a card data vault (CDV), which would then typically retrieve the PAN from a PAN-to-token table. For example, a PAN could be assigned to a token in a pre-generated table of random values. The only thing that has to be kept secret is the actual relationship between the PAN and its token. For this instance of tokenization, the token has no mathematical relationship with its associated PAN (i.e., for the purposes of this document, a look-up table or index is not considered a mathematical relationship between the token and PAN). However, in a hybrid product the cryptographic token has a mathematical relationship with its PAN, so the guidelines for reversible cryptographic tokens would also apply. Annex E – Use Cases for Tokenization provides sample use cases.

Tokenization Roles

This section summarizes the roles of stakeholders with direct responsibility for tokenization products or services.

Stakeholder Role	Responsibilities
Tokenization Product Vendor	A tokenization product vendor is a third-party vendor who provides a packaged tokenization product (e.g., tokenization appliance or software application) to a merchant or other end user of the product. This product vendor is also responsible for creation, distribution, and maintenance of a *Tokenization Installation Guide (TIG)* for their product.
Tokenization Service Provider	A tokenization service provider is a third-party entity (e.g., a processor, acquirer, or payment gateway) providing tokenization services to other entities (such as merchants).

Tokenization At-a-Glance

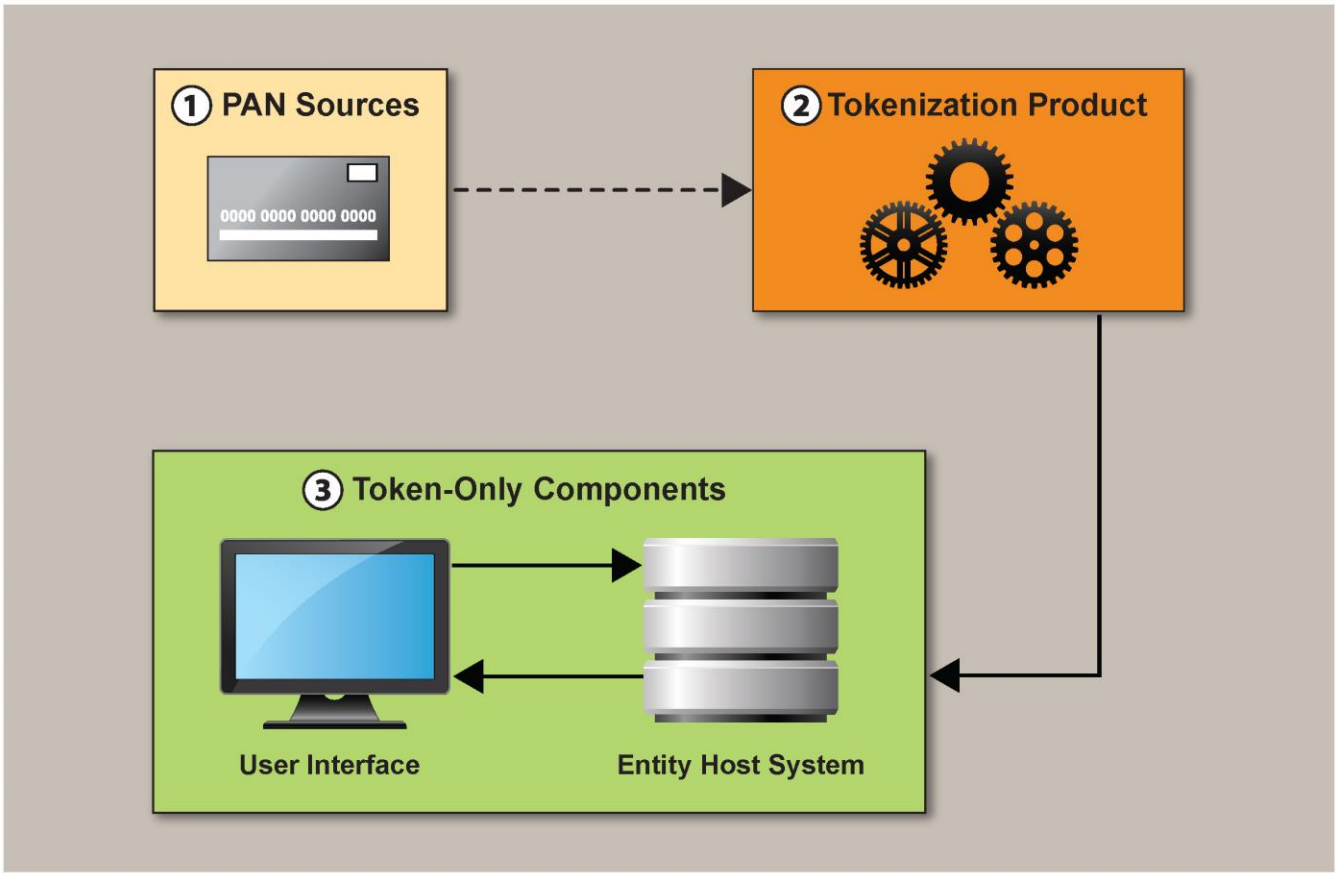
In accordance with the tokenization taxonomy, irreversible and reversible tokens have different security considerations in addition to the general guidelines that apply to all tokenization products. This section gives an overview and illustration of the data flow within four tokenization scenarios. Figures 2 and 3 are examples of irreversible tokenization scenarios. Figures 4 and 5 are examples of reversible tokenization scenarios. It is important to note that the tokenization product schematics in the following figures are meant to illustrate one of many possible scenarios.

Note: *These examples illustrate data flow within a hypothetical tokenization product implementation and do not describe any resulting cardholder data environment (CDE).*

Irreversible Tokens

Figure 2 shows generic PAN sources sending a PAN to a tokenization product. The tokenization product then sends an irreversible token back to the components requesting it along with any other transaction information, excluding PAN and sensitive authentication data (SAD). In this case, only the token is stored after authorization.

Figure 2: Irreversible Tokenization Scenario



Contains PAN within a secure channel.

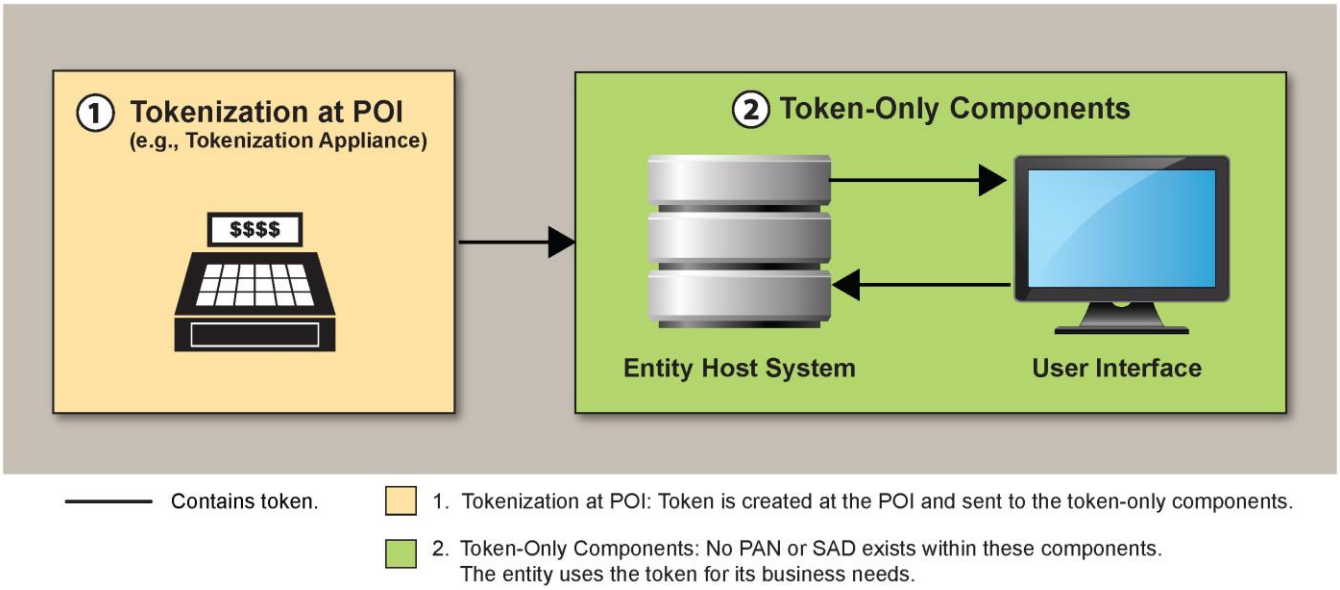
Contains Token.

1. PAN Sources: PAN is sent from here to the Tokenization Product within a secure channel.
2. Tokenization Product: Tokenization takes place and token is sent to the token-only components. No PAN is stored after authorization.
3. Token-Only Components: No PAN or SAD exists within these components. The entity uses the token for its business needs.

Note: *This example illustrates data flow within a hypothetical tokenization product implementation and does not describe any resulting CDE. Additionally, tokenization may occur pre-authorization or post-authorization.*

Figure 3 shows another irreversible tokenization scenario where the tokenization takes place at the point of interaction. The irreversible token is then sent to the token-only components along with any other transaction information, excluding PAN and SAD. In this case, after authorization only the token is stored.

Figure 3: Irreversible Tokenization Scenario with Tokenization at POI

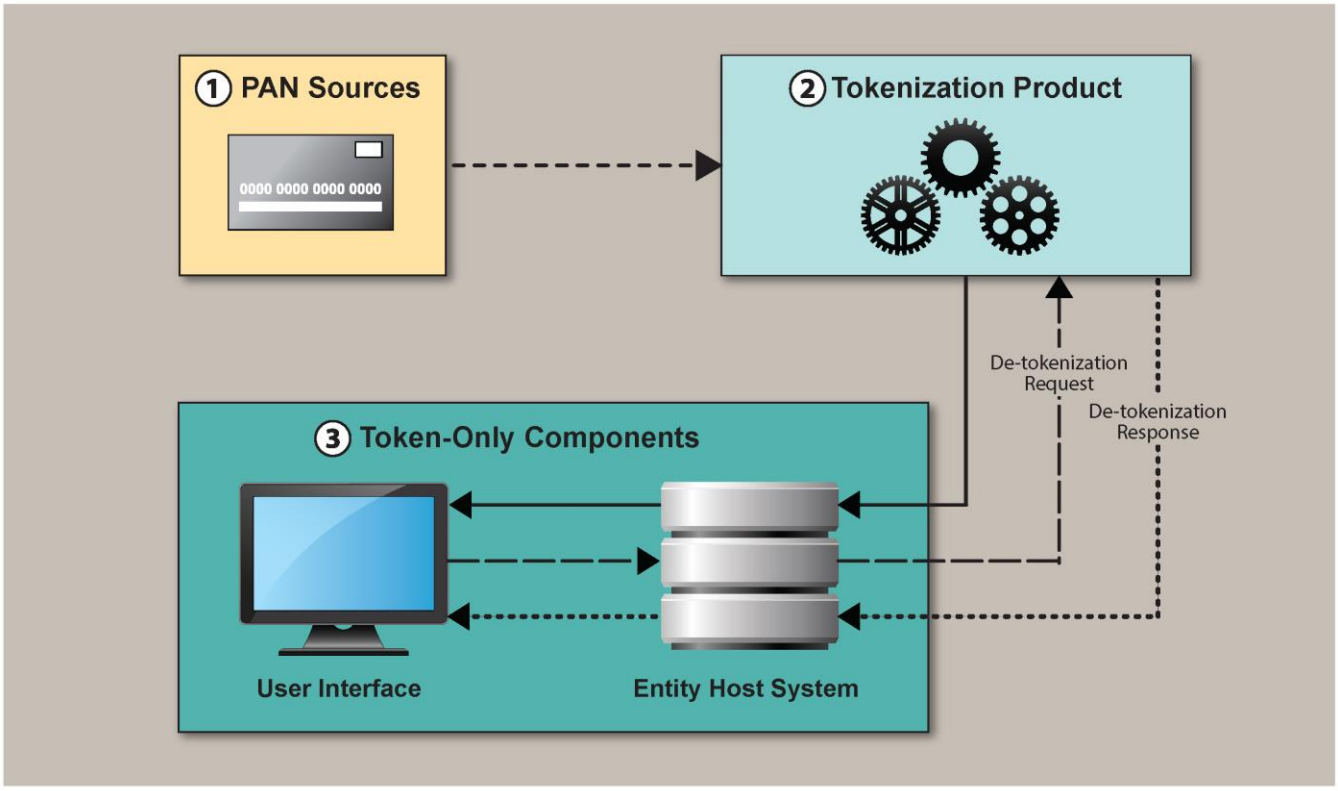


Note: *This example illustrates data flow within a hypothetical tokenization product implementation and does not describe any resulting CDE. Additionally, tokenization may occur pre-authorization or post-authorization.*

Reversible Tokens

Figure 4 shows a reversible tokenization implementation where a reversible token is sent back to the components requesting it. The figure shows that one portion of the environment has the capability for de-tokenization.

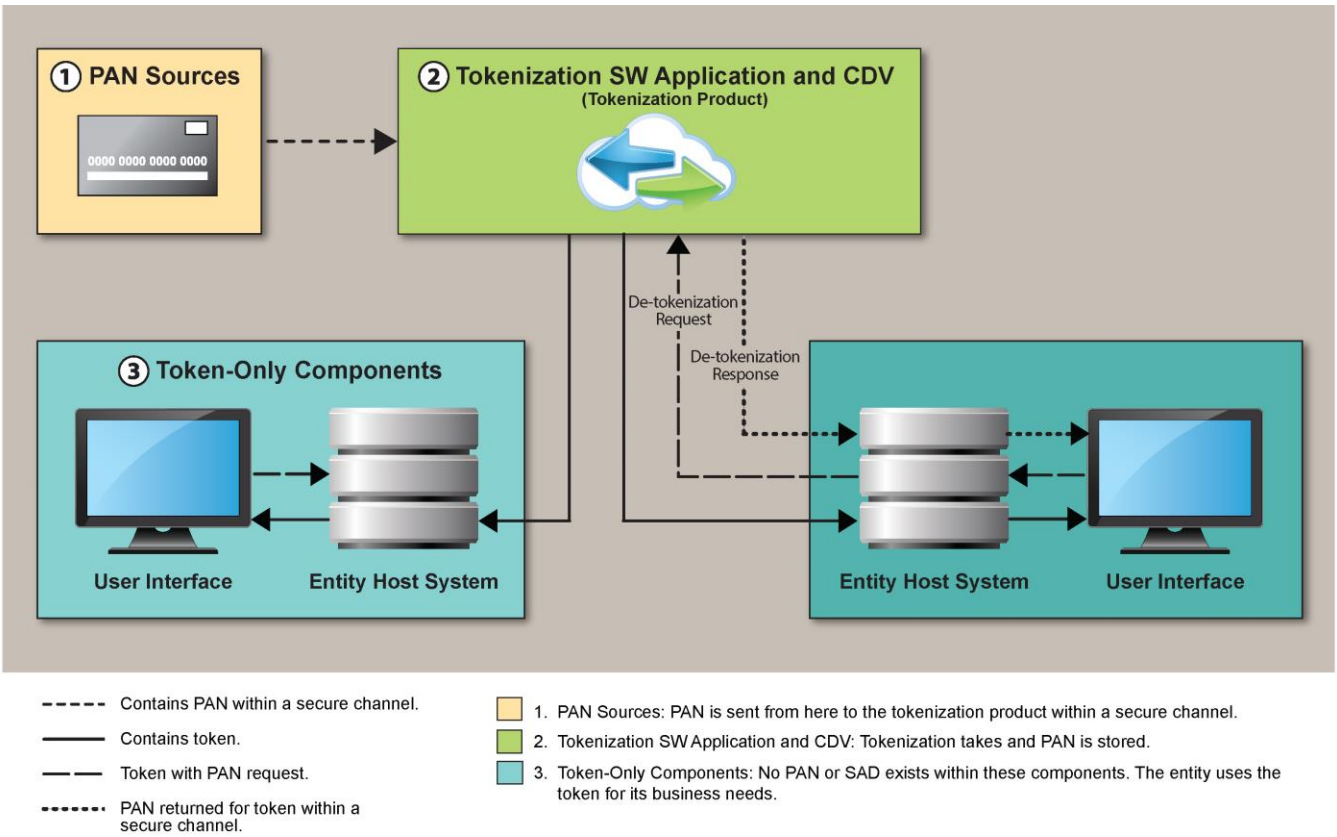
Figure 4: Reversible Tokenization Scenario



Note: *This example illustrates data flow within a hypothetical tokenization product implementation and does not describe any resulting CDE. Additionally, tokenization may occur pre-authorization or post-authorization.*

Figure 5 shows another reversible tokenization scenario where tokenization is performed with a software product.

Figure 5: Reversible Tokenization Scenario with a Tokenization Software Product



Note: *This example illustrates data flow within a hypothetical tokenization product implementation and does not describe any resulting CDE. Additionally, tokenization may occur pre-authorization or post-authorization.*

Detailed Tokenization Guidelines/Best Practices

The following defines the column headings for the *Tokenization Product Security Guidelines*:

- **Tokenization Guideline/Best Practice –** This column defines the Tokenization Product Security Guidelines against which tokenization products may be evaluated.
- **Evaluation Procedures –** This column shows processes to be followed to evaluate that the tokenization product has met the Guidelines/Best Practices.
- **Guidance –** This column describes the intent or security objective behind each Guideline/Best Practice.

Use of Secure Cryptographic Devices (SCDs)

If a tokenization product needs to protect cryptographic keys, the product should use an SCD as described in Annex A – Guidelines/Best Practices for Products using an SCD. Examples may include:

- Reversible cryptographic tokenization products, as these use a cryptographic key to create tokens;
- Tokenization products that encrypt part or all of the CDV to protect PAN, tokens, or the token/PAN relationship;
- An irreversible tokenization product using a cryptographic key.

General Guidelines/Best Practices

General Guidelines/Best Practices apply to tokenization products regardless of where the tokenization product falls within the taxonomy. These guidelines form the foundation for minimizing the potential for unauthorized disclosure of PANs through the use of tokens. Beyond these general Guidelines/Best Practices, additional considerations will apply depending on the nature of the tokenization product—e.g., irreversible or reversible.

General Guideline/Best Practice	Evaluation Procedures	Remarks
GT 1 If hardware products are used for tokenization, the hardware products should be validated to FIPS 140-2 Level 3, operate in FIPS mode, and be initialized to Overall Level 3 (or greater) per security policy. See Annex A – Guidelines/Best Practices for Products using and SCD. In addition, a PCI-listed HSM evaluation process may be used. ***Note**: In order for a product to be evaluated under FIPS 140-2, the vendor submitting the product would have had to submit the necessary protection profile for the evaluation. The vendor shall provide the protection profile and supporting documentation to the lab for the lab to assess relevance and suitability for tokenization. (Note that FIPS 140-2 permits Common Criteria with appropriate EALs as part of its evaluation process. A Common Criteria evaluation that met or exceeded the requirements for FIPS 140-2 is an acceptable international alternative.)*		Hardware products that have achieved a FIPS 140-2 Level 3 rating have undergone a rigorous qualification process to protect the cryptographic module and verify cryptographic algorithms. This provides an industry standard of assurance for evaluating cryptographic modules and algorithms. Additionally, the operation of an SCD in FIPS mode increases the overall security of the device and core functionality provided by the SCD.

General Guideline/Best Practice	Evaluation Procedures	Remarks
GT 3 If software products are used for tokenization, the software products should be validated to FIPS 140-2 Level 2, operate in FIPS mode, and be initialized to Overall Level 2 (or greater) per security policy. The FIPS validation should include any operating system that the software depends on.	**GT 3** Confirm that FIPS 140-2 Level 2 or higher certificate exists for the product or that the criteria are met.	Tokenization software products that have a FIPS 140-2 Level 2 rating have undergone a rigorous qualification process to protect the cryptographic module and verify cryptographic algorithms. This provides an industry standard of assurance for evaluating cryptographic modules and algorithms. Additionally, the use of FIPS mode for operating systems (OSs) and tokenization software products increases the security of the software cryptographic modules and boundaries.
GT 4 Access to multiple token-to-PAN pairs should not allow the ability to predict or determine other PAN values from knowledge of only tokens.	**GT 4.a** Verify that the tokenization mechanism generated PAN/token pairs are statistically independent of each other by design. **GT 4.b** Perform testing to verify that the output corresponds to what is expected from the analysis in GT 4.a above. If tokens are mathematically calculated from the PAN and/or (pseudo-) random numbers are used, test for statistical independence of PAN/token pairs by means of statistical tests.	The intent is to show that the creation of token-to-PAN pairs is independent of all other token-to-PAN pairs. Therefore, the evaluation should show that each instance of a token-to-PAN pair is statistically independent of all other instances of token-to-PAN pairs. Additionally, the randomness and security of the tokenization process should also be confirmed.
GT 5 The recovery of the original PAN should be computationally infeasible knowing only the token, a number of tokens, or a number of PAN/token pairs.	**GT 5** Verify that the product acts in accordance with the security model or formal proof (see Annex K – Security Models and Formal Proofs) and the recovery of the original PAN is computationally infeasible knowing only the token, a number of tokens, or a number of token/PAN pairs.	The intent is to ensure that it is computationally infeasible to recover the original PAN knowing only the token, a number of tokens, or token-to-PAN pairs that don't include the original PAN. If it is feasible, the tokenization system is not secure.

General Guideline/Best Practice	Evaluation Procedures	Remarks
GT 6 The tokenization product should implement monitoring to detect any malfunctions, anomalies, and suspicious behavior that might indicate irregular token-to-PAN or PAN-to-token mapping requests or the presence of unauthorized activity within the tokenization process, and implement a means to alert personnel. The tokenization product should provide a means to log all such events.	**GT 6** Perform exception testing to confirm detection and reporting of any anomalies, malfunctions, and/or suspicious behavior. Such testing should include all documented error conditions (for conformance with documentation), data fuzzing, and frequency threshold triggers (or such other testing as is appropriate to the product's documented mechanism for detection of suspicious behavior).	Monitoring controls ensure that suspicious events are identified in a timely manner. The product vendor would have to document what is the normal or expected behavior around requests for token-to-PAN or PAN-to-token mapping. Additionally, the product would need to have rules in place so that any deviations are recorded for future analysis and appropriate personal are notified.
GT 7 The tokenization product should include a mechanism for distinguishing between tokens and actual PANs. The mechanism or method for distinguishing between tokens and PANs for a particular tokenization product may be intrinsic (e.g., the resulting tokens are not in a format that could reasonably be interpreted as PAN or uses a specific BIN range) or extrinsic (e.g., a label logically bound[2] to the token). Tokenization product vendors should share the mechanism with the entities using that product. (See *Tokenization Installation Guide* (TIG).)	**GT 7.a** Verify that the asserted mechanism for distinguishing tokens from PANs is adequate to ensure consistent distinguishability of PANs from tokens. **GT 7.b** If the asserted mechanism is adequate, verify that it functions as described. **GT 7.c** Verify that the method to distinguish PANs and tokens is described in the TIG.	The intent is to verify that the tokenization product is functioning correctly. It is essential that the capability exists to distinguish a token from a PAN. Without this, you would have the problem of distinguishability and you wouldn't be able to distinguish tokens from PANs. Since the token generated could have the same or a different format as that of a PAN, there needs to be a way to determine which is a token and which is a PAN, as the security needs for tokens and PANs are different. The organization would need to ensure that they continue to protect and secure PANs.

[2] A data element or field is *logically bound* to its label if one or more of the following are true: (1) the label and datum are cryptographically bound such that an attempt to change the label is detectable (e.g., as in message authentication); (2) the label and datum are programmatically bound (i.e., they form a data object that the underlying programming language treats as a single object—e.g., a 2-tuple [label, datum]); or (3) the label is the logical representation of the data item such that a change in the label results in it ceasing to represent the data with which it was previously associated and such that the data item represented by the label cannot be changed or replaced except through use of the label.

General Guideline/Best Practice	Evaluation Procedures	Remarks
GT 8 The tokenization product vendor should develop and provide a *Tokenization Installation Guide* (TIG) to the entity to assist in the proper deployment, implementation, and use of the tokenization product.	**GT 8.a** Verify the existence of the TIG. **GT 8.b** Verify that all the information in the TIG is correct by testing it on a live installation. **GT 8.c** Follow TIG instructions to configure different functions on the product and observe system output to confirm the correctness of the TIG instructions.	Without proper instruction on how to install and use the tokenization product, an entity might configure and use the product in an insecure manner. Therefore, the intent is to ensure that all necessary information (see Annex B – Tokenization Installation Guide (TIG)) is provided by the vendor of the product to the entity implementing the product.
GT 9 Mechanisms should be in place to ensure the integrity of the token-generation process. Examples include the use of cryptographic authentication techniques (e.g., digital signatures, HMAC, and hashes) to ensure the integrity of the executable or the use of high-assurance programming techniques.	**GT 9.a** Verify that the vendor has documented the mechanisms to ensure the integrity of the token-generation process. **GT 9.b** Verify that the product includes and uses those integrity mechanisms. **GT 9.c** Assess the adequacy of the integrity mechanisms. ***Note:*** *The vendor should document integrity constructs and methods used for the irreversible token-generation process. Additionally, the vendor should reference and use approved cryptographic methods for integrity checks per Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives.*	The integrity of the tokenization product is critical to ensuring its proper functioning and reliability.
GT 9.1 Critical functions (e.g., the API code) within the tokenization application must be protected by integrity-checking mechanisms (e.g., cryptographic integrity techniques, independent parallel processing with comparisons, read-only memory, or other high-assurance techniques).	**GT 9.1.**a Verify the documented critical functions are consistent with the evaluation of what the critical functions should be. **GT 9.1.b** Verify the critical functions are protected by integrity-checking mechanisms. **GT 9.1.c** Verify the integrity-checking mechanisms are providing effective protection.	Using proper integrity-checking mechanisms on critical functions protects access to the tokenization and de-tokenization process.

General Guideline/Best Practice	Evaluation Procedures	Remarks
GT 10 Only authenticated users and system components should be allowed access to the tokenization system and tokenization/de-tokenization processes. In addition, the following authentication aspects should be addressed when evaluating a tokenization product: • **Identification –** Provides a unique identifier to the application, user, process, or system (i.e., subject) requesting access. • **Enrollment –** Associates a unique identity with a subject. • **Authentication –** Validates the alleged identity of the subject. The authentication method should categorize all endpoints, including but not limited to applications, people, processes, and systems to ensure the appropriate level of access is granted. ***Note:*** *For purposes of authentication, the mechanism should be at least as stringent as specified in PCI DSS Requirement 8.*	**GT 10.a** Identify all logical access-control points to the tokenization system, including but not limited to applications, people, processes, and systems. **GT 10.b** Verify that all logical access-control points function as specified in the documentation and as defined by GT 9. ***If the tokenization product relies on an external authentication mechanism:*** **GT 10.c** Verify that documentation provides detailed instructions for implementing authentication, and the instructions cover all access-control points identified for the tokenization solution (including but not limited to applications, people, processes, and systems). **GT 10.d** Verify that the authentication method(s) documented in the TIG include defining and enforcing the appropriate level of access. ***If authentication method(s) is provided with tokenization solution:*** **GT 10.e** Verify that the authentication mechanisms function as specified in the documentation.	The intent is to preserve the integrity of the tokenization and/or de-tokenization process by limiting access to the tokenization and/or de-tokenization system to only specifically authorized users and systems. This can be accomplished with a proper authentication process that is documented and validated to include all access-control points.

General Guideline/Best Practice	Evaluation Procedures	Remarks
	GT 10.f Verify that the authentication method(s) provided with the solution enforce the following for all logical access-control points. • **Identification** – Provides a unique identifier to the application, user, process, or system (i.e., subject) requesting access. • **Enrollment** – Associates a unique identity with a subject. • **Authentication** – Validates the alleged identity of the subject.	
GT 10.1 Mapping requests should go through an evaluated Application Program Interface (API) such that the application is able to control effectively all access attempts and uniformly apply access-control rules.	**GT 10.1.a** Verify actual APIs versus the documented APIs.	An API can be used to control specific activities such as access to the tokenization and/or de-tokenization requests. Therefore, it is essential that all critical APIs be evaluated to ensure they function properly (i.e., both correctly and securely). Additionally, an API is an entry point that could be used by unauthorized functions if not properly controlled and/or managed.
	GT 10.1.b Verify that each API can be effectively controlled based on access-control rules.	
GT 10.2 The tokenization product should have access and tokenization/de-tokenization logging functionality. This functionality should be securely configurable. ***Note:*** *Refer to PA-DSS Requirement 4 – Log Payment Application Activity.*	**GT 10.2.a** Verify that the identified access and transaction logging functionality is actually in place.	Logging mechanisms, the ability to track user activities, and tokenization/de-tokenization activities are critical in preventing, detecting, or minimizing the impact of a product failure or data compromise. The presence of logs in all environments allows thorough tracking, alerting, and analysis when something does go wrong—e.g., a failure of the tokenization function or process. Determining the cause of a product failure or compromise is very difficult, if not impossible, without product activity logs.
	GT 10.2.b Assess the sufficiency of the access and transaction logging functionality. For example, identify events that are not captured by this logging functionality.	
	GT 10.2.c Verify that the access and transaction logging functionality is securely configurable.	

General Guideline/Best Practice	Evaluation Procedures	Remarks
GT 10.2.1 Tokenization and de-tokenization requests should be logged and the logs should not contain PAN; however, PAN truncation is acceptable (see *PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms* for definition of *truncation*) **if** it does not contain different or more clear-text digits than those in the token. Alternatively, it may be acceptable if the log data is isolated from the tokenization data such that access (including unauthorized access) to one of these does not imply access to the other.	**GT 10.2.1.a** Verify that the product does not have fields in its log records that would contain PAN.	A properly designed and implemented tokenization product does not contain PAN outside of the token vault, including logs. It is essential that the necessary logs provide an accurate and unaltered record of what has taken place within the tokenization product (e.g., who did what, where, when, and how), but not facilitate the ability to map tokens to PANs in any form. For example, a log may contain a truncated PAN provided that knowledge of the token, together with the truncated PAN, does not facilitate a feasible attack on the PAN or increase the probability of correctly guessing a truncated PAN.
GT 10.3 Tokenization product should support multi-factor authentication for all user access[3] to the tokenization product, including administrative access, tokenization and de-tokenization requests, maintenance, vendor access, etc. ***Note:*** *Two or more of the same factor, for example two passwords do not qualify as MFA. There should be no fall back to a single factor in the event of a failure. MFA for remote client applications should not be vulnerable to malware on the client device.*	**GT 10.3.a** Identify all user-access mechanisms supported by the product. **GT 10.3.b** Verify that each mechanism supports multi-factor authentication.	Access to the tokenization product is high-risk as it contains highly sensitive data, information, and configuration settings, which if accessed or altered by unauthorized personnel—even if unintentional—could result in a product failure or data compromise. Multi-factor authentication (MFA) requires at least two independent methods of authentication for access to the tokenization product linked to a unique digital ID (see PCI DSS Requirement 8.2 or the *Glossary* of this document for a description of the three methods).
GT 10.4 Tokenization product should support mutual authentication for all system-access requests to the product, including tokenization and de-tokenization requests.	**GT 10.4.a** Identify all system-access paths supported by the product. **GT 10.4.b** Verify that each system-access path supports mutual authentication.	The intent is to protect system-level access, with the same or greater rigor as access to systems that contain PAN and other sensitive data, to the tokenization product since these systems have the ability to map tokens to PANs and vice-versa. As such, it is critical that all system-level access can be validated by the tokenization product as originating from a valid, authorized, and secure

[3] "User access" in this context means access by a human being.

General Guideline/Best Practice	Evaluation Procedures	Remarks
		source.
GT 10.5 Strong cryptography should be used for encryption of all non-console administrative (see Glossary) access to tokenization applications and/or appliances.	**GT 10.5.a** Identify all non-console administrative access.	If non-console (including remote) administration does not use secure authentication and encrypted communications, sensitive administrative or operational level information (like administrator's IDs and passwords) can be revealed in clear-text to an eavesdropper. A malicious individual could use this information to access the network, become administrator, access the CDV, and obtain data. To be considered "strong cryptography," industry-recognized protocols with appropriate key strengths and key management should be in place as applicable for the type of technology in use. (Refer to "Strong Cryptography" in the *PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms* and Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives of this document.)
	GT 10.5.b Verify that strong cryptography (see Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives) is used for all non-console administrative access.	

General Guideline/Best Practice	Evaluation Procedures	Remarks
GT 11 Converting from a token produced under one system (or cryptographic key or non-cryptographic process) to a token produced under another independent system (or cryptographic key or non-cryptographic process) should require an intermediate PAN state—i.e., invocation of de-tokenization. This assures that the old token is independent of the new token. (See Annex J – Token-to-Token Conversions.) ***Note:*** *1. This guideline/best practice does not prohibit the tokenization of a token (i.e., recursion, which is not the same as converting from one token under one system to another token under another independent system). See Annex I – Recursive Tokenization.* *2. Irreversible tokenization products will not be capable of such conversions.*	**GT 11** Confirm that no function exists that allows the conversion of Tokens produced under one mechanism (or cryptographic key) to another token produced under a different mechanism (or cryptographic key).	There are many reasons why it might become necessary to change from one tokenization basis to another. Examples include changing tokenization vendor, suspected security failure, regulatory change, transfer of assets (e.g., sale or merger), and migration to a new platform that requires a different product. As a result, organizations may find the need to convert tokens. To ensure the integrity of each of the tokenization systems, the conversion process will require the token to become PAN in order to perform the next tokenization process. (See Annex I – Recursive Tokenization and Annex J – Token-to-Token Conversions.)

General Guideline/Best Practice	Evaluation Procedures	Remarks
GT 12 The product vendor should implement measures to address common security vulnerabilities as identified in PA-DSS Requirement 5.2. Strategies by the developer to address these vulnerabilities may include: • Avoiding them by design (extreme example: using a programming language, which prevents buffer overflow by definition); • Finding them by adequate testing (for example, static code analysis or comprehensive fuzz testing); and/or • Mitigating them by techniques that include but are not limited to: Address Space Layout Randomization (ASLR), Data Execution Prevention (DEP), Harvard Architecture, and Stack Canaries. Applications also should make use of—and not disable—operating system-based memory protection such as Address Space Layout Randomization (ASLR), Data Execution Prevention (DEP), compilation flags, and other options to prevent unauthorized code execution.	**GT 12.a** Perform a source-code review of each interface and confirm that only documented commands are implemented and that secure defaults are provided for each interface. Detail the methods used to verify the length and content of each command before processing. Derive vulnerability-analysis models from source-code review and other available evidence to determine appropriate penetration testing. These evaluation activities should be targeted on relevant security-critical functionalities such as (but not restricted to): buffer overflows, unhandled exceptions, read-access violations, and denial-of-service conditions, etc., including factors that are specific to the SCD's OS, communications protocols, and source-code software language(s). **GT 12.b** Verify that the vendor has implemented appropriate measures to address common security vulnerabilities as identified in PA-DSS Requirement 5.2.	This is intended to address common software security vulnerabilities for tokenization products. The application layer is high-risk and may be targeted by both internal and external threats. Without proper security, PAN and the tokenization process can be exposed. PA-DSS outlines specific requirements that are the minimum controls that should be in place. This list is composed of the most common coding vulnerabilities at the time that this version of the PA-DSS was published. As industry-recognized common coding vulnerabilities change, vendor coding practices should likewise be updated to match.
GT 13 Where the vendor uses cryptographic primitives, those primitives should be based on published national or international standards—e.g., AES or ECC. If a cryptographic primitive is used (per Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives), the vendor shall provide the lab with a statistical validation document—e.g., NIST CAVP cryptogram validation document or similar.	**GT 13.a** Identify the cryptographic primitives used. **GT 13.b** Compare the primitives used against the applicable published standards and Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives.	The intent is to ensure that when cryptographic primitives are used, they are based on published national and international standards. Cryptographic primitives are well-known, low-level cryptographic routines that set the foundation for more complex cryptographic algorithms and protocols. Industry-recognized cryptographic primitives have been tested and proven to be reliable, efficient, and effective.

Overview of Security Domains for Tokenization

This section outlines the security domains for tokenization products. The guidelines/best practices that are unique to a specific tokenization class will be given in their corresponding sections of this document. The following is a brief overview of those domains.

Domain 1 – Token Generation

For each tokenization class, this domain defines considerations for securely generating tokens. For all token-generation processes, this domain covers all devices, processes, mechanisms, and/or algorithms used to create tokens. For example, in irreversible tokenization, this domain will specify that the process, mechanism, or algorithm used to create tokens is provably irreversible. As another example, for a reversible token, this domain may specify the cryptographic key strength used in the algorithm that creates the token.

Domain 2 – Token Mapping

Domain 2, which addresses the mapping of tokens to their original PANs, is only applicable to a reversible tokenization implementation. Among other things, this domain will encompass access controls and logging needs for tokenization and de-tokenization requests.

Domain 3 – Card Data Vault

Domain 3, which addresses the card data vault (CDV), is only applicable to a reversible tokenization implementation. This domain covers the mandatory encryption of the PAN and the access controls used to access the CDV.

Domain 4 – Cryptographic Key Management

Domain 4 defines proper cryptographic key management practices for all cryptographic key management operations performed by the tokenization product.

Applicability of Best Practices to Different Token types

The following table summarizes how the Guidelines/Best Practices in this document apply to different types of tokens.

Section / Domain	Applicability per Token Type			
	Irreversible (IT)	**Reversible Cryptographic (RC)**	**Reversible Non-Cryptographic (RN)**	**Hybrid**
GT – General Tokenization Guidelines	**Yes**	**Yes**	**Yes**	**Yes**
Domain 1 – Token Generation	**Yes**	**Yes**	**Yes**	**Yes**
Domain 2 – Token Mapping	**No**	**Yes**	**Yes**	**Yes**
Domain 3 – Card Data Vault	**No**	**Potentially**	**Yes**	**Yes**
Domain 4 – Cryptographic Key Management	**Yes**	**Yes**	**Yes**	**Yes**
Annex A – Guidelines/Best Practices for Products Using an SCD	**Yes, if an SCD is used**	**Yes, if an SCD is used**	**Yes, if an SCD is used**	**Yes**

Irreversible Tokens

These Guidelines/Best Practices for irreversible tokens are in addition to the General Guidelines/Best Practices. They apply only to tokenization products that qualify as irreversible.

Each domain has its own table that provides an overview of the domain. Each guideline/best practice is presented in detail following the table.

Domain 1: Token Generation

	Environments using Irreversible Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 1: Token Generation	▪ Token generated is irreversible. ▪ The creation of a table or "dictionary" of static tokens (see Glossary) should be infeasible at least to the extent that the probability of correctly guessing the PAN should be less than 1 in 10^6. (Where access to the associated partial PAN is possible—i.e., the masked PAN—the Luhn check process allows the calculation of any single missing digit, so the effective strength drops to 1 in 10^5.)	**IT 1A** The process/mechanism/algorithm used to create the token provably is not reversible.

Irreversible Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
IT 1A The process, mechanism or algorithm used to create the token provably is not reversible.		
IT 1A-1 The process for creating tokens classified as irreversible should ensure that the process/mechanism/algorithm used to create the token provably is not reversible. ***Note:*** *If a hash is used, the hash function should be a cryptographic primitive and use a secret key such that knowledge of the hash function does not by itself permit the creation of an oracle. See Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives.*	**IT 1A-1** Evaluate the process/scheme/algorithm used to create the token to determine whether it conforms to the proof provided by the vendor. Additionally: • If a hash function is used, confirm that an oracle cannot be created, and that an approved cryptographic primitive and secret key is used (per Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives). • If non-cryptographic means are used, the vendor should provide both a statistical validation document and security proof to validate irreversible token generation. • The vendor should also clearly state against what standard they are measuring for non-cryptographic methods and process. ***Note:*** *If a cryptographic primitive is used (per Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives), the vendor should provide a statistical validation document (NIST CAVP cryptogram validation document or similar) and security proof in order to validate irreversible token generation.*	The intent is to ensure that irreversible tokens are provably irreversible.

Irreversible Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
IT 1A-2 Tokens should not contain clear-text digits of the original PAN, except by chance. ***Note:*** *For any acquiring token intended to be irreversible, no clear-text digits of the original PAN may be copied over to the token.*	**IT 1A-2** Verify that the token does not contain clear-text digits of the PAN, except by chance. ***Note:*** *The vendor should provide an original PAN and irreversible token sample for each method or process used to create irreversible tokens, to determine whether the token contains clear-text digits of the original PAN.*	Clear-text digits from the original PAN, if transferred to the "irreversible" token, reduce the space for guessing and increase the ability to build a dictionary, thereby risking the token becoming reversible.
IT 1A-3 The creation of a table or "dictionary" of static tokens (see Glossary) should be infeasible at least to the extent that the probability of correctly guessing the PAN should be less than 1 in 10^6. (This is the same probability of guessing a truncated PAN under current rules without recourse to a Luhn check.)	**IT 1A-3.a** Verify that the asserted mechanism prevents the creation of PAN/token pairs. Determine whether the truncated PAN contains digits not found in the token (only applicable where the token contains clear-text digits from the original PAN). If so, fail. If applicable, confirm that the token does not contain clear-text PAN digits that are not already contained in the truncated PAN, except by chance. ***Note:*** *The vendor should provide documentation to confirm whether any irreversible tokens leverage truncation, FPE, or any other mechanism for tokenization. It should be demonstrated that a creation of a table or "dictionary" of static tokens is infeasible at least to the extent stated here.*	The intent is to set a floor for the probability of guessing a PAN from the token. Since this is for irreversible tokens, the security principle is that no token or set of tokens (in a given context) should provide sufficient information to permit a guess of its associated PAN with better than 1 chance in 1,000,000. If the token contains clear-text digits and additional clear-text digits are available from an associated truncated PAN, the number of remaining digits will be fewer than 6 (probably fewer than 5) making a guess more likely than 1 chance in 1,000,000. (This is even worse if Luhn checking is available.) Without this test, a table of partial PAN and associated tokens would be feasible that will allow increasingly accurate guesses as it expands with correct guesses.

Irreversible Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
	IT 1A-3.b Verify that the coexistence of a truncated PAN and a token does not provide a statistical advantage greater than the probably of correctly guessing the PAN based on the truncated value alone. Based on the mechanism used to produce the irreversible token, assess whether the truncated value would be sufficient to permit a known-plaintext attack (which might occur off line). If so, would a successful attack compromise the mechanism (e.g., cryptographic primitive) or only one PAN/token pair? If the former, fail. If the latter, note weakness and determine whether a control exists that would effectively prevent or detect the acquisition of multiple token/truncated PAN pairs. (Such a control is unlikely to exist, so this would also generally fail.) ***Note**: If a cryptographic mechanism is used as a component of an irreversible token (truncated PAN, FPE, or any other reference to PAN values are in the nomenclature of the token), then the vendor provides documentation to validate the security strength (see Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives) for each respective mechanism. The vendor should provide a truncated PAN and irreversible token sample for each.*	The intent is to ensure that the probability of guessing a token for a given PAN is no greater than that of guessing the PAN based on a truncated PAN under current rules without recourse to a Luhn check.

Irreversible Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
IT 1A-4.1 If an authenticatable irreversible token is used, the authentication process should not leak information sufficient to test PANs except by PAN-space exhaustion. Controls should be in place to detect attempted PAN-space exhaustion.	**IT 1A-4.1.a** Verify that mechanisms are in place to prevent information leakage. **IT 1A-4.1.b** Verify that the controls in place to prevent attempted PAN-space exhaustion are effective and function as document by the vendor. ***Note:*** *The vendor should provide both a statistical validation document (NIST CAVP cryptogram validation document or similar), and security proof to validate for cryptographic and non-cryptographic methods. Apply this to the application processes and methods to ensure data leakage and PAN-space exhaustion are not possible and function as documented by the vendor. Additionally, the vendor should clearly state against what standard they are measuring for non-cryptographic methods and processes.*	This intent is to ensure that no data leakage from the tokenization process gives any increased probability of guessing either an associated PAN or a future token.

Domain 2: Token Mapping

	Environments using Irreversible Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 2: Token Mapping	▪ Not applicable.	***Domain 2 has no applicable Guidelines/Best Practices for this irreversible token scenario***.

Domain 2 has no applicable Guidelines/Best Practices for the irreversible token scenario.

Domain 3: Card Data Vault

	Environments using Irreversible Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 3: Card Data Vault (CDV)	▪ Not applicable.	***Domain 3 has no applicable Guidelines/Best Practices for this irreversible token scenario***.

A CDV is not permitted for irreversible token implementations. Therefore, Domain 3 has no applicable Guidelines/Best Practices for the irreversible token scenario.

Domain 4: Cryptographic Key Management

	Environments using Irreversible Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 4: Cryptographic Key Management	▪ Follow industry standards (e.g., NIST SP800-57 and ISO/IEC 11770) and other PCI standards that address proper cryptographic key-management practices that may apply.	**IT 4A** Proper cryptographic key-management practices should be followed.

Irreversible Tokens Domain 4 Guidelines/Best Practices	Evaluation Procedures	Guidance
IT 4A Proper Cryptographic Key-management practices should be followed.		
IT 4A-1 The tokenization key should have a key life-cycle policy as described in ISO/IEC 11568-1. (Refer to Annex D – Cryptographic Key Management Life Cycle.)	**IT 4A-1** If applicable, verify the existence and adequacy of documentation on the intended key life cycle.	Proper management of tokenization keys is essential for effective and secure operation of the tokenization product. A documented policy outlines requirements and provides assurance that keys are adequately protected throughout its life cycle from generation, loading, conveyance and destruction.
IT 4A-2 The key lifetime policy should include a description of the active cryptoperiod of the tokenization key. (Refer to Annex D – Cryptographic Key Management Life Cycle.)	**IT 4A-2** If applicable, verify the existence and adequacy of documentation on the active cryptoperiod.	Old, static keys may be more susceptible to attack since it allows more time for criminals to compromise them. Defining a cryptoperiod or timespan for the allowable active life of a key, mitigates this risk.

Irreversible Tokens Domain 4 Guidelines/Best Practices	Evaluation Procedures	Guidance
IT 4A-3 The vendor should incorporate a feature that permits the zeroization/destruction of its cryptographic keys without requiring the device to be tampered or opened.	**IT 4A-3.a** Verify that the vendor asserted mechanism exists that would zeroize/destroy the cryptographic keys without requiring the device to be tampered.	The ability to render a device inoperable through zeroization/destruction of its cryptographic keys allows an organization to quickly respond to a bad situation that could lead to the compromise of PAN. It should be noted that access to this function should be strictly limited and incorporates logging and alerts.
	IT 4A-3.b Verify that the mechanism zeroizes/destroys the cryptographic keys without requiring the device to be tampered.	

Reversible Cryptographic Tokens

The Guidelines/Best Practices for reversible cryptographic tokens are in addition to the General Guidelines/Best Practices. They apply only to tokenization products that qualify as reversible cryptographic.

A cryptographic tokenization system is where the secret information consists of a cryptographic key of at least 128 bits of strength. In addition, such systems should meet all the Guidelines/Best Practices for tokenization in this section. These characteristics make cryptographic tokens qualitatively different from encrypted PANs.

Each domain has its own table that provides an overview of the domain. Each guideline/best practice is presented in detail following the table.

Domain 1: Token Generation

	Environments using Reversible Cryptographic Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 1: Token Generation	▪ Key generation. ▪ Regardless of the encryption method used, a PAN is retrievable from its token. ▪ The probability of guessing the token should be less than 1 in 10^6. (Where access to the associated partial PAN is possible—i.e., the masked PAN—the Luhn check process allows the calculation of any single missing digit, so the effective strength drops to 1 in 10^5.)	**RC 1A** Cryptographic key management should be secure. (See Domain 4: CKM.) **RC 1B** The probability of guessing a token to PAN relationship should be less than 1 in 10^6. **RC 1C** Tokens that are based on the entire PAN should not be stored if the tokenization product (including any dependent systems) also stores their corresponding truncated PANs.

Reversible Cryptographic Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
RC 1A Cryptographic key management should be secure. (See Domain 4: CKM).		
RC 1A-1 The cryptographic keys used to generate tokens should not be able to be exported in plaintext from the tokenization product.	**RC 1A-1.a** Verify that the product conforms to vendor-provided documentation with regard to cryptographic key storage and use, including their form. **RC 1A-1.b** Verify that the product does not make the cryptographic key available in plaintext form outside of the secure decryption environment. **RC 1A-1.c** Confirm that nothing in the TIG would require or allow plaintext cryptographic keys outside of the secure decryption environment.	It is essential that cryptographic keys are secured and protected at all times for their entire life cycle within the product—see Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives and Annex D – Cryptographic Key Management Life Cycle. Further, the cryptographic keys used to generate tokens must not be exported from a state of higher security to a state of lower security. There is no mechanism in the device that would allow the outputting of a private or secret clear-text key, the encryption of a key under a key that might itself be disclosed, or the transfer of a clear-text key from a component of high security into a component of lesser security.
RC 1A-2 The cryptographic key used to generate tokens should be generated from a source with at least 128 bits of entropy. See Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives.	**RC 1A-2.a** Verify that documentation exists describing the entropy sources used to generate the cryptographic keys. **RC 1A-2.b** Verify that the entropy sources used to generate cryptographic keys have at least 128 bits of entropy. ***Note:*** *Vendors should demonstrate via their documentation that, whatever their source of entropy, it provides at least 128 bits of entropy.*	The intent is to set the floor for the cryptographic key strength. Since you cannot have more bits of security than you do bits of entropy, 128 bits of entropy is the minimum necessary to meet the minimum key strength. Multiple sources of randomness and uniqueness can support random number generators (RNG) and the creation of cryptographic keys that support tokenization.
RC 1A-3 The cryptographic key used to generate tokens, or any derivative of that key, should not be used for any other purpose.	**RC 1A-3.a** Verify that documentation exists detailing how the cryptographic keys used to generate tokens, or any derivative of that key, is not used for any other purpose.	The intent is to ensure that the cryptographic keys associated with the generation and use of reversible tokens is only used for a single purpose.

Reversible Cryptographic Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
	RC 1A-3.b Verify that the key used to generate tokens, or any derivative of that key, may not be used for any other purpose. *For example, if the system generates a random key that is only stored for a single purpose and not otherwise retained, the random value could not be loaded as some other type of key. Any counter example found would indicate that this test fails.*	
RC 1B ***The probability of guessing a token to PAN relationship should be less than 1 in 10^6. (The token should not give any advantage to an attacker trying to guess the corresponding PAN. This is the same probability of guessing a truncated PAN under current rules without recourse to a Luhn check.)***	**RC 1B** Verify that the products function in accordance with the security model or formal proof. (See Annex K – Security Models and Formal Proofs.) ***Note:*** *If a cryptographic primitive is used (per Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives), the vendor should provide both a statistical validation document (NIST CAVP cryptogram validation document or similar), and security proof in order to validate reversible token generation as implemented in their token application. See Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives for approved primitives and methods.*	The intent is to set a floor for the probability of guessing a PAN from the token. This is the same probability of guessing a truncated PAN under current rules without recourse to a Luhn check.

Reversible Cryptographic Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
RC 1B-1 For a given PAN, all matching token values should be equivalently likely—i.e., the tokenization product should not exhibit a probabilistic bias as it would expose it to a statistical attack.	**RC 1B-1.a** Verify that documentation exists for a security model or formal proof used to demonstrate that all matching token values are equivalently likely for a given PAN.	This is intended to ensure that there is no bias in the generation of tokens. That is, each token from the set of possible tokens is equally likely for every PAN submitted to the tokenization product. For example, when a PAN is presented to the tokenization product, the product will generate a token where that token is just as likely to be produced as any other possible token.
	RC 1B-1.b Verify that the product functions in accordance with the security model or formal proof. See Annex K – Security Models and Formal Proofs.	
RC 1B-2 The tokenization method should be shown to act as a family of random permutations from the space of actual PANs to the token space.	**RC 1B-2.a** Verify that documentation exists that describes the tokenization methods that act as a family of random permutations from the space of actual PANs to the token space.	The intent is to ensure that the tokens are indistinguishable from a random permutation over the space of actual PANs. The probability of any token mapping to any PAN should be equal.
	RC 1B-2.b Verify that the product functions in accordance with the documented tokenization method.	
RC 1B-3 Changing the tokenization key should change the token mapping.	**RC 1B-3.a** Verify that documentation exists describing how a change in the tokenization key changes the token mapping.	The intent is to ensure that a change to the "tokenization key" will result in a different token being generated for a particular PAN (except by chance), thus there will be a new PAN-to-token pair for that key. ***Note:*** *A token mapping is the relationship that a given token has to its associated PAN (or vice versa). Because reversible cryptographic solutions may include those that use a cryptographic primitive, but are not a direct encryption, a "tokenization key" change might not result in a new mapping of PAN/tokens. This best practice provides an assurance that it will.*
	RC 1B-3.b Verify that a change in the tokenization key changes the token mapping.	

Reversible Cryptographic Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
RC 1B-4 Changing the clear digits of the PAN should change the token mapping. ***Notes:*** • *Outside of the token-generation process, there are some cases where a new PAN may need to update a CDV associated with an existing token.* • *As an exception, if a PAN is being replaced (e.g. reissued), the replacement PAN can be mapped to same token as the previous PAN.*	**RC 1B-4.a** Verify that documentation exists describing how a change in the clear digits of the PAN changes the token mapping. **RC 1B-4.b** Verify that a change in the clear digits of the PAN changes the token mapping.	The intent is to ensure that all PANs map to a different token. Thus, if there is a change in the clear-text digits of the PAN, then there should be a change to the token mapping. Additionally, it is not possible for a token to map to different PANs.
RC 1B-5 The vendor should provide a means for the practical verification of digit randomization—e.g., refer to NIST SP 800-90A. See Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives.	**RC 1B-5** Verify that documentation exists for practical verification of digit randomization.	Verification is necessary to ensure the digits are properly randomized. NIST SP 800-90A provides guidance in producing randomization of digits.

Reversible Cryptographic Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
RC 1C Tokens that are based on the entire PAN should not be stored if the tokenization product (including any dependent systems) also stores their corresponding truncated PANs. ***Note:*** *In a hybrid solution, the only place where these can be stored together is within the card data vault. Details for securing the tokenization vault are in the Non-Cryptographic Token section.* *Nowhere in a payments ecosystem should a truncated PAN and token be stored, outside of a CDV.*	**RC 1C** Confirm, by documentation verification and testing, that the product does not store both tokens based on the entire PAN (include any dependent systems) and their corresponding truncated PANs.	The intent is to prevent correlating the truncated PAN and the tokenized PAN.
RC 1C-1 The system storing tokens should not have truncated PANs that contain any plain PAN digits that are not present in the generated token (or vice versa). For examples, see Annex H – Examples of Tokens.	**RC 1C-1** Confirm that the product does not produce tokens that contain any plain-text PAN digits that would not otherwise already be present in the truncated PAN. ***Alternatively:*** Confirm that the tokens produced by the product do not contain any digits from the original PAN, except by chance.	This best practice addresses the security vulnerability of having both the token value and its corresponding truncated PAN stored in the same location—namely; the token and the PAN could be correlated. Furthermore, the entropy of the missing digits is significantly reduced because of the non-truncated digits.

Domain 2: Token Mapping

	Environments using Reversible Cryptographic Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 2: Token Mapping	▪ The equivalent of token mapping for a cryptographic token is the process of decryption.	**RC 2A** There should be access controls in place for tokenization and de-tokenization requests.

Reversible Cryptographic Tokens Domain 2 Guidelines/Best Practices	**Testing Procedures**	**Guidance**
RC 2A There should be access controls in place for tokenization and de-tokenization requests.		
RC 2A-1 All application requests for tokenization or de-tokenization should be authenticated and tested against internal access controls.	**RC 2A-1.a** Verify that documentation exists describing the authentication mechanisms for tokenization and de-tokenization requests.	Authentication of all tokenization or de-tokenization requests ensures that only permitted requests are granted access to the tokenization or de-tokenization system.
	RC 2A-1.b Verify that application requests for tokenization and de-tokenization are authenticated and tested against internal access controls.	
	RC 2A-1.c Assess that the authentication mechanisms are adequate.	

Reversible Cryptographic Tokens Domain 2 Guidelines/Best Practices	Testing Procedures	Guidance
RC 2A-2 Role-based access controls (RBACs) should be required to obtain the PAN from its associated token—e.g., ANSI INCITS 359.	**RC 2A-2.a** Verify that documentation exists that describes the RBACs used when obtaining a PAN for its associated token.	Using RBACs can ensure that the de-tokenization request is limited to only those individual users with authorization to make those requests.
	RC 2A-2.b Verify that the RBACs function as described in the documentation.	
	RC 2A-2.c Assess whether the RBACs are adequate when obtaining a PAN for its associated token.	

Domain 3: Card Data Vault

	Environments using Reversible Cryptographic Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 3: Card Data Vault (CDV)	▪ Applicable only if used.	***Domain 3 may have no applicable Guidelines/Best Practices if the PAN is not being stored in card data vault. However, if it is stored, the Domain 3 Guidelines/Best Practices of Reversible Non-Cryptographic Tokens apply***.

Domain 3 has no applicable Guidelines/Best Practices because the PAN is not being stored in a card data vault.

Domain 4: Cryptographic Key Management

	Environments using Reversible Cryptographic Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 4: Cryptographic Key Management (CKM)	▪ Key generation ▪ Key storage ▪ Key strength ▪ Key lifecycle	**RC 4A** All cryptographic key management (CKM) operations should be performed in an approved SCD (e.g., HSM). **RC 4B** CKM should be performed in accordance with ISO/NIST Standards—e.g., NIST Special Publication 800-57, ISO/IEC 11770, and NIST Special Publication 800-130. **RC 4C** The effective key strength should be at least 128 bits. **RC 4D** If cryptographic keys are used to produce multiple, different PAN/token sets—e.g., to separate merchants—each key should be statistically independent.

Reversible Cryptographic Domain 4 Guidelines/Best Practices	Evaluation Procedures	Guidance
RC 4A ***All cryptographic key management operations should be performed in an approved SCD. For example, any one of the following is acceptable:*** • ***PCI-listed SCD (e.g., HSM)*** • ***FIPS 140-2 Level 3 (validated to FIPS 140-2 Overall Level 3, operated in FIPS mode, and initialized to Overall Level 3 per security policy) or above*** • ***Independently validated to ISO 13491-1***		All symmetric keys (except ephemeral keys that are one-time use) must exist only in one of the approved forms—i.e., within an approved SCD, in full-length cryptographic key shares, or encrypted under another cryptographic key of equal or greater effective key strength. For asymmetric keys, the private key must be equivalently protected.
RC 4A-1 The cryptographic key used to perform tokenization operations should be generated within an approved SCD (e.g., HSM), and the encryption operations associated with the tokenization method should be performed inside the SCD. The cryptographic key should not be available in plaintext form outside the SCD. If using a software product, it is permissible for the key to temporarily exist in plaintext within the memory of the secured host computer while necessary for the cryptographic operation.	**RC 4A-1.a** Verify that documentation exists that describes all cryptographic tokenization methods as outlined here.	It is essential that cryptographic keys be strongly protected, because those who obtain access will be able to decrypt data. Documented standards provide an operational overview; however, it is important to verify that the devices are operating as intended to ensure clear-text data is not being processed, stored or transmitted instead of ciphertext.
	RC 4A-1.b Verify that the cryptographic key(s) used for tokenization operations are generated from an approved SCD or HSM.	
	RC 4A-1.c Verify that the encryption operations associated with the tokenization method are performed inside an SCD or HSM.	
	RC 4A-1.d Verify that the cryptographic key used for tokenization operations is not available in plaintext from outside the HSM or SCD, except within memory of a secure host computer while necessary for the cryptographic operation.	

Reversible Cryptographic Domain 4 Guidelines/Best Practices	Evaluation Procedures	Guidance
RC 4B CKM should be performed in accordance with ISO/NIST Standards—e.g., NIST SP 800-57, NIST SP 800-130, and ISO/IEC 11770. ***Note:*** *Vendor documentation should be provided to support this.*	**RC 4B.a** Verify that documentation exists that describes all cryptographic key management operations.	In order to help ensure that CKM is performed securely, ensure it is done in accordance with the appropriate industry standards and the vendor provides the relevant supporting documentation.
	RC 4B.b Verify that CKM is performed in accordance with ISO/NIST Standards—e.g., NIST SP 800-57, NIST SP 800-130, and ISO/IEC 11770.	
RC 4B-1 The tokenization key should have a key life-cycle policy as described in ISO/IEC 11568-1. See Annex D – Cryptographic Key Management Life Cycle.	**RC 4B-1** If applicable, verify the existence and adequacy of documentation on the intended key life cycle.	Defining a life cycle for cryptographic keys ensures the process is repeatable and predicted that helps control quality and delivery schedule of keys.
RC 4B-2 The key lifetime policy should include a description of the active cryptoperiod of the tokenization key (refer to Annex D – Cryptographic Key Management Life Cycle).	**RC 4B-2** If applicable, verify the existence and adequacy of documentation on the active cryptoperiod.	As part of the cryptographic key management life cycle, a period of time needs to be defined to determine the span of time in which a key will be or remain valid.
RC 4B-3 The vendor should incorporate a feature that permits the zeroization/destruction of its cryptographic keys without requiring the device to be tampered.	**RC 4B-3.a** Verify that a vendor-asserted mechanism exists that would zeroize/destroy the cryptographic keys without requiring the device to be tampered.	The ability to render a device inoperable through zeroization/destruction of its cryptographic keys allows an organization to quickly respond to bad situation that could lead to the compromise of PAN. It should be noted that access to this function should be strictly limited and incorporates logging and alerts.
	RC 4B-3.b Verify that the mechanism zeroizes/destroys the cryptographic keys without requiring the device to be tampered.	

Reversible Cryptographic Domain 4 Guidelines/Best Practices	Evaluation Procedures	Guidance
RC 4C The effective key strength should be at least 128 bits.		
RC 4C-1 The tokenization key should have an effective key strength of at least 128 bits. (Refer to NIST SP 800-57 Recommendation for Key Management – Part 1: General (Revision 3), Table 2.)	**RC 4C-1.a** Verify that documentation exists describing the key strength of the tokenization key.	In cryptographic systems, the length of the key directly relates to the security of the system. The larger effective key strength increases the difficulty of a brute-force attack.
	RC 4C-1.b Confirm that the product actually uses only keys that have an effective key strength of at least 128 bits.	
RC 4C-2 Any cryptographic keys used to protect or to derive the tokenization key should have equal or greater effective key strength.	**RC 4C-2** Verify that documentation exists requiring any keys used to protect or to derive the tokenization key should have equal or greater effective key strength.	To ensure the strength of the key is fully observed, any key protecting a key should be the same or greater strength. A strong key becomes weaker when protected by a key of lesser strength.
RC 4D If cryptographic keys are used to produce multiple, different PAN/token sets (e.g., to separate merchants), each key should be statistically independent.	**RC 4D.a** Verify that documentation exists that describes the security model or formal proof used to show that if cryptographic keys are used to produce multiple, different PAN/token sets, each key is statistically independent.	Keys that are not statistically independent are more likely to be compromised since the formula used to produce the keys is known. Proper key generation using approved random number generators ensures the uniqueness of the keys.
	RC 4D.b Verify that cryptographic keys used are statistically independent.	

Reversible Non-Cryptographic Tokens

For reversible non-cryptographic tokens, obtaining the PAN from its token is only by data look-up within the card data vault (CDV). For instance, PANs could be assigned to a token in a pre-generated table of random values. The only thing that needs to be kept secret is the actual relationship between the PAN and its token. In this instance, the token should have no mathematical relationship with its associated PAN. (For the purposes of this standard, a look-up table or index is not considered a mathematical relationship between the token and PAN.)

These Guidelines/Best Practices for reversible non-cryptographic tokens augment the General Guidelines/Best Practices. They apply only to tokenization products that qualify as reversible non-cryptographic.

Each domain has its own table that provides an overview of the domain. Each guideline/best practice is presented in detail following the table.

Domain 1: Token Generation

	Environments using Reversible Non-Cryptographic Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 1: Token Generation	▪ While data elements used in the process may be secret—e.g., the seed to a random number generator—the process of generating a token is not a secret. ▪ The probability of guessing a PAN from its token should be less than 1 in 10^6. (Where access to the associated partial PAN is possible—i.e., the masked PAN—the Luhn check process allows the calculation of any single missing digit, so the effective strength drops to 1 in 10^5.)	**RN 1A** The generation of a token should be performed independently of its PAN and the relationship between a PAN to its token would only be contained within the CDV. **RN 1B** The probability of guessing a PAN from its token should be less than 1 in 10^6. **RN 1C** The token-generation process should ensure an unbiased distribution of tokens, i.e., the probability of any given PAN/token pair should be equal. **RN 1D** If multiple, different PAN/token CDVs are used—e.g., to separate merchants—each instance should be statistically independent. (This is analogous to the concept of using different cryptographic keys in a cryptographic token model.)

Reversible Non-Cryptographic Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
RN 1A ***The generation of a token should be performed independently of its PAN, and the relationship between a PAN to its token would only be contained within the CDV.***	**RN 1A.a** Verify that documentation exists describing how the token is generated independently from its PAN.	The intent is to ensure that tokens are generated independently of their corresponding PANs and that their relationship only exists within the CDV. If the token and the PAN are not independent, then they have a relationship or are considered a cryptographic token.
	RN 1A.b Verify that the token is generated independently of its PAN as described in the documentation. ***Note:*** *The vendor should provide documentation that the process to generate tokens is statistically independent from PAN.*	
	RN 1A.c Verify that the relationship between the PAN and its token is only stored within the CDV. ***Note:*** *The vendor should document that the relationship pairing the resulting token values and PANs only exists within the card data vault.*	

Reversible Non-Cryptographic Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
RN 1B ***The probability of guessing a PAN from its token should be less than 1 in 10^6. (This is the same probability of guessing a truncated PAN under current rules without recourse to a Luhn check.)***	**RN 1B.a** Verify that the product functions in accordance with the security model or formal proof provided by the vendor. (See Annex K – Security Models and Formal Proofs.) ***Note:*** *If a cryptographic primitive is used (per Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives), the vendor should provide both a statistical validation document (NIST CAVP cryptogram validation document or similar), and security proof in order to validate reversible token generation. (See Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives for approved primitives and methods.)*	The intent is to set a floor for the probability of guessing a PAN from its token. It is also essential that the vendor clearly document how they measure and achieve the probability of guessing a PAN from its token.
	RN 1B.b Assess the validity of the vendor-asserted model or proof (See Annex K – Security Models and Formal Proofs). ***Note:*** *The vendor should provide both a statistical validation document and security proof to validate reversible token generation using non-cryptographic means within their application.*	

Reversible Non-Cryptographic Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
RN 1B-1 For a given PAN, all matching token values should be equivalently likely—i.e., the tokenization product should not exhibit a probabilistic bias as it would open it up to a statistical attack.	**RN 1B-1.a** Verify that documentation exists for a security model or formal proof used to demonstrate that all matching token values are equivalently likely for a given PAN. ***Note**: The vendor should provide documentation that the security model and formal proof of PAN and token relationships have a normal statistical distribution without any bias.*	This is intended to ensure that there is no bias in the generation of tokens. That is, each token from the set of possible tokens is equally likely for every PAN submitted to the tokenization product. For example, when a PAN is presented to the tokenization product, the product will generate a token where that token is just as likely to be produced as any other possible token.
	RN 1B-1.b Verify that the product functions in accordance with the security model or formal proof. ***Note:** The tester should sample PAN and token pairs to ensure the application meets a normal statistical distribution without bias via the documented security model or proof.*	
RN 1B-2 The tokenization method should be shown to act as a family of random permutations from the space of actual PANs to the token space.	**RN 1B-2.a** Verify that documentation exists that describes the tokenization methods that act as a family of random permutations from the space of actual PANs to the token space.	The intent is to ensure that the tokens are indistinguishable from a random permutation over the space of actual PANs. The probability of any token mapping to any PAN should be equal.
	RN 1B-2.b Verify that the product functions in accordance with the documented tokenization method.	
RN 1B-3 The tokenization method should include parameters such that a change of these parameters will result in different token mappings. For example, different installations or instances of the process should be able to produce a different sequence of tokens, even when presented with the same sequence of PANs.	**RN 1B-3.a** Verify that documentation exists that describes how a change in the parameters of the tokenization parameters will result in a change in the token mapping.	This is intended to ensure that different instances (if, there are any) of a tokenization product produce different tokens for the same PAN. Additionally, it is not possible for a token to map to different PANs. ***Note:** This parallels RC 1B-3.*
	RN 1B-3.b Verify that a change in the parameters of the tokenization method changes the token mapping.	

Reversible Non-Cryptographic Tokens Domain 1 Guidelines/Best Practices	Evaluation Procedures	Guidance
RN 1B-4 Changing the clear digits of the PAN should change the token mapping. ***Notes:*** • *Outside of the token-generation process, there are some cases where a new PAN may need to update a CDV associated with an existing token.* • *As an exception, if a PAN is being replaced (e.g. reissued), the replacement PAN can be mapped to same token as the previous PAN.*	**RN 1B-4.a** Verify that documentation exists describing how a change in the clear digits of the PAN changes the token mapping. **RN 1B-4.b** Verify that a change in the clear digits of the PAN changes the token mapping.	The intent is to ensure that all PANs map to a different token. Thus, if there is a change in the clear-text digits of the PAN, there should be a change to the token mapping.
RN 1B-5 The product vendor should provide a means for the practical verification of digit randomization—e.g., refer to NIST SP 800-90A.	**RN 1B-5** Verify that documentation exists for practical verification of digit randomization.	Verification is necessary to ensure the digits are properly randomized. NIST SP 800-90A provides guidance in producing randomization of digits.
RN 1C The token-generation process should ensure an unbiased distribution of tokens, i.e., the probability of any given PAN/token pair should be equal.	**RN 1C.a** Verify that documentation exists that describes a tokenization process that produces an unbiased distribution of tokens. **RN 1C.b** Verify that the tokenization process produces an unbiased distribution of tokens.	The intent is to ensure that the creation of tokens is performed in an unbiased manner and the assignment of a token to a PAN is indistinguishable from a random assignment. As a result, each PAN-to-token pair is equally likely.
RN 1D If multiple, different PAN/token CDVs are used—e.g., to separate merchant—each instance should be statistically independent. (This is analogous to the concept of using different cryptographic keys in a cryptographic token model.)	**RN 1D.a** Verify that documentation exists that describes the different PAN/token CDVs that are used and how they are independent. **RN 1D.b** Verify that the product conforms to the vendor's documentation.	Using the same PAN/token CDVs for multiple customers increases the potential to compromise all customers serviced.

Domain 2: Token Mapping

	Environments using Reversible Non-Cryptographic Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 2: Token Mapping	▪ Obtaining a PAN from its token should be performed by data look-up within the CDV.	**RN 2A** The mapping of a token to its PAN should be performed by data look-up within the CDV. **RN 2B** Role-Based Access Controls (RBACs) should be required to obtain the PAN from its associated token within the CDV—e.g., ANSI INCITS 359.

Reversible Non-Cryptographic Tokens Domain 2 Guidelines/Best Practices	Evaluation Procedures	Guidance
RN 2A The mapping of a token to its PAN should be performed by data look-up (or an index) within the CDV.	**RN 2A.a** Verify that documentation exists that describes the mapping of a token to its corresponding PAN.	The de-tokenization of the token to the full PAN value can only be performed via a data look-up or index within the CDV only and not via a cryptographic method.
	RN 2A.b Verify that the mapping operates as indicated in the vendor documentation.	

Reversible Non-Cryptographic Tokens Domain 2 Guidelines/Best Practices	Evaluation Procedures	Guidance
	RN 2A.c Verify the mapping of a token to its PAN is performed by data look-up (or an index) within the CDV. ***Note:*** *If the CDV is not part of the application submitted for evaluation, the cryptographic security measures and tokenization functions should be accomplished external to the database system and internal to the application.*	
RN 2A-1: The PAN and the token value should be provably independent. ***For example,*** *if you have a table of sorted PANs and you are using an index as the token, then they are not independent. Further, any token based on a logical arrangement of PANs (e.g., a logical tree structure) is an example that fails to meet this independence criterion.*	**RN 2A-1.a** Verify that documentation exists that describes the security model or formal proof used to show that the tokens and their corresponding PANs are independent. **RN 2A-1.b** Verify that the PANs and their corresponding tokens are independent.	A logical pattern or method, such as a mathematical formula, is not to be used to tokenize the PAN and/or to de-tokenize the token. This ensures true independence between the PAN and the token.
RN 2B Role-Based Access Controls (RBACs) should be required to obtain the PAN from its associated token within the CDV—e.g., ANSI INCITS 359.	**RN 2B.a** Verify that documentation exists that describe the RBACs used when obtaining a PAN for its associated token within the CDV. **RN 2B.b** Verify that the RBACs functions as described in the documentation. **RN 2B.c** Assess whether the RBACs are adequate when obtaining a PAN for its associated token within the CDV.	In order to limit access to the CDV, only those individuals who need such access should be defined using a role-based access-control system—e.g., system administrator, security administrator or key administrator. Individual access can be granted according to their job classification and function by using an already created role.

Domain 3: Card Data Vault

	Environments using Reversible Non-Cryptographic Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 3: Card Data Vault (CDV)	▪ The PAN should be stored encrypted in the CDV. (See Domain 4: CKM.)	**RN 3A** The PAN should be encrypted with a cryptographic key that has the strength of at least 128 bits. **RN 3B** RBACs should be required for access to the CDV (e.g., ANSI INCITS). **RN 3C** All copies (e.g., backups, load balancing, or distributed) should be equivalently protected.

Reversible Non-Cryptographic Tokens Domain 3 Guidelines/Best Practices	Evaluation Procedures	Guidance
RN 3A The PAN should be encrypted with a cryptographic key that has the strength of at least 128 bits (SP 800-57 Recommendation for Key Management-Part 1: General [Revision 3] Table 2). Refer to Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives.	**RN 3A.a** Verify that documentation exists describing the key strength of the key used to encrypt the PAN. **RN 3A.b** Confirm that the product actually uses only keys that have an effective key strength of at least 128 bits.	The intent of strong cryptography is that the encryption be based on an industry-tested and accepted algorithm—not a proprietary or "home-grown" algorithm—with strong cryptographic keys.

Reversible Non-Cryptographic Tokens Domain 3 Guidelines/Best Practices	Evaluation Procedures	Guidance
RN 3B Role-Based Access Controls (RBACs) should be required for access to the CDV—e.g., ANSI INCITS 349).	**RN 3B.a** Verify that documentation exists that describe the RBACs used for accessing the CDV.	In order to limit access to the CDV, only those individuals who need such access should be defined using a role-based access-control system—e.g., system administrator, security administrator, or key administrator. Individual access can be granted according to their job classification and function by using an already created role.
	RN 3B.b Verify that the RBACs function as described in the vendor documentation.	
	RN 3B.c Assess whether the RBACs are adequate when accessing the CDV.	
RN 3C All copies—e.g., backups, load balancing, or distributed—should be equivalently protected.	**RN 3C** Verify that documentation exists that describes how copies are to be equivalently protected.	Documented procedures identify controls that have been established for protecting backup copies. These procedures allow for recreating steps to ensure consistency of methods.

Domain 4: Cryptographic Key Management

	Environments using Reversible Non-Cryptographic Tokens	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Domain 4: Cryptographic Key Management (CKM)	▪ Cryptographic key management is required if elements of the CDV are cryptographically protected.	**RN 4A** All cryptographic key management operations should be performed in an approved SCD (e.g., HSM). **RN 4B** All CKM should be performed in accordance with NIST/ISO Standards—e.g., NIST Special Publication 800-57, ISO/IEC 11770, and NIST Special Publication 800-130.

Reversible Non-Cryptographic Tokens Domain 4 Guidelines/Best Practices	Evaluation Procedures	Guidance
RN 4A ***All cryptographic key management operations should be performed in an approved SCD. For example, any one of the following is acceptable:*** • ***PCI-listed SCD—e.g., HSM.*** • ***FIPS 140-2 Level 3 (validated to FIPS 140-2 Overall Level 3, operated in FIPS mode, and initialized to Overall Level 3 per security policy) or above.*** • ***Independently validated to ISO 13491-1***	**RN 4A.a** Verify that documentation exists that describes the CKM operations that are performed within an approved SCD or HSM. **RN 4A.b** Verify that all CKM operations are performed within an HSM or SCD.	Hardware products that have achieved a FIPS 140-2 Level 3 rating have undergone a rigorous qualification process to protect the cryptographic module and verify cryptographic algorithms. Since key-management functions are fundamental to the security of the tokenization product, use of approved SCDs provides reasonable assurance of secure operations.
RN 4B ***All CKM should be performed in accordance with NIST/ISO Standards—e.g., NIST SP 800-57, ISO/IEC 11770, and NIST SP 800-130. See Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives and D.***	**RN 4B.a** Verify that documentation exists that describes all cryptographic key management operations. **RN 4B.b** Verify that CKM is performed in accordance with ISO/NIST Standards—e.g., NIST SP 800-57, NIST SP 800-130, and ISO/IEC 11770.	Documented procedures identify controls, methods and steps that ensure security operations. Documented procedures inform key custodians and stakeholders of approved and allowed practices.

Annex A – Guidelines/Best Practices for Products Using an SCD (*Normative*)

The Guidelines/Best Practices in this annex are normative if you are using an SCD as part of the tokenization product.

	Products using SCDs	
Domain	**Characteristics**	**Summary of Tokenization Guidelines/Best Practices**
Device Management	▪ If Secure Cryptographic Devices (SCDs)—e.g., tokenization appliance, POI, or HSM—are used, they should be securely managed throughout their life cycle.	**A 1A** If Secure Cryptographic Devices (SCDs) are used: • SCDs should be secured throughout their life cycle; • Secure device-management processes should be implemented.

Products using SCDs	Evaluation Procedures	Guidance
A 1A [Conditional] If secure cryptographic devices are used:		
A 1A-1 Product vendor should maintain inventory control to track accurately SCDs in their possession. This should include documented procedures for monitoring the inventory of SCDs.	**A 1A-1** Verify that documentation exists that describes the procedures for monitoring the inventory of SCDs.	The intent is to ensure that all the SCDs in the vendor's possession are accounted for and reflect where they are being stored. This ensures that the vendor can detect any lost or stolen devices in a timely manner.
A 1A-2 Product vendor should physically secure SCDs in their possession at all times, including when not deployed or in use, and provide related instructions (refer to Annex B – Tokenization Installation Guide (TIG)) to the entity implementing the tokenization product.	**A 1A-2.a** Verify that documentation exists that describes the procedures for physical security of SCDs in the possession of the vendor. **A 1A-2.b** Verify that the vendor has produced the TIG and contains the related instructions.	The intent is to ensure that the vendor stores SCDs in a secure facility to prevent them from being lost or stolen. Additionally, the vendor will provide explicit guidance to the entity implementing the tokenization product on how to securely store this product within their facility.

Products using SCDs	Evaluation Procedures	Guidance
A 1A-3 Product vendor should have procedures to prevent and detect the unauthorized alteration or replacement of SCDs in their possession prior to and during deployment, and provide related instructions (see Annex B – Tokenization Installation Guide (TIG)) to the entity implementing the tokenization product.	**A 1A-3.a** Verify that documentation exists that describes the procedures to prevent and detect the unauthorized alteration or replacement of SCDs. **A 1A-3.b** Verify that the vendor has produced the TIG and contains the related instructions.	The intent is to ensure that processes or controls are in place to ensure that the SCDs are not tampered. This can be accomplished by monitoring the inventory, developing a check-in or check-out process, and reviewing the inventory periodically to ensure that all devices are accounted for at various stages of their life cycle.
A 1A-4 Product vendor should prevent unauthorized physical access to devices undergoing repair or maintenance while in their possession, and provide related instructions (see Annex B – Tokenization Installation Guide (TIG)) to the entity implementing the tokenization product.	**A 1A-4.a** Verify that documentation exists for procedures that prevent unauthorized physical access to devices undergoing repair or maintenance while in their possession. **A 1A-4.b** Verify that the vendor has produced the TIG and contains the related instructions.	Although there might be formal processes in place to track SCDs that are yet to be deployed, test devices or devices that are being repaired might not have the same level of rigor. If such a device is compromised, a malicious user might be able to tamper with and/or obtain sensitive information from it, which could impact the security whenever it is deployed in the field later.
A 1A-5 Product vendor should securely maintain devices being returned, replaced, or disposed of, and provide related instructions (see Annex B – Tokenization Installation Guide (TIG)) to entities implementing the tokenization product.	**A 1A-5.a** Verify that documentation exist for procedures for securely maintaining devices being returned, replaced, or disposed of. **A 1A-5.b** Verify that the vendor has produced the TIG and contains the related instructions.	It is important for the vendor to document and maintain the state of various devices so that inventory accurately reflects them. It is essential to prevent these devices from being tampered with and redeployed elsewhere without appropriate security safeguards.
A 1A-6 Devices should be configured by default to immediately fail closed (that is, stop, shut down, go offline, or otherwise cease all processing) if tokenization mechanism fails, until the tokenization mechanism is restored.	**A 1A-6.a** Verify that the mechanism functions as described in the documentation provided by the vendor, for all failure modes. **A 1A-6.b** Assess the adequacy of the mechanism.	PANs become exposed if the tokenization mechanism fails. Having the devices default to immediately fail closed if the tokenization mechanism fails removes that exposure. Simulating various types of failures can confirm whether the device does default to immediately fail closed.

Products using SCDs	Evaluation Procedures	Guidance
A 1A-7 Product vendor should restrict access to devices in its possession to authorized personnel.	**A 1A-7.a** Verify that documentation exists that describes the procedures for restricting access to devices in their possession to authorized personnel.	Having documented procedures that restricts access of the devices to only authorized personnel of the product vendor ensures proper custody of those devices. Unauthorized personnel may effect changes to the devices knowingly or unknowingly before it gets to the end user.
	A 1A-7.b Provide evidence of an independent assessment (or audit) of the procedures in their environment.	
A 1A-8 The product vendor should protect SCDs from known vulnerabilities and implement procedures for secure updates to devices, including:		
A 1A-8.1 The product vendor should have secure update processes in place for all firmware and software updates, including: • Integrity-check of update. • Authentication of origin of the update.	**A 1A-8.1.a** Verify that the secure update processes for all firmware and software updates operate in accordance with vendor documentation.	Security updates on all firmware and software is critical in addressing vulnerabilities.
	A 1A-8.1.b Assess the adequacy of the controls.	
A 1A-8.2 The product vendor should maintain an up-to-date inventory of SCD system builds and conduct vulnerability assessments against all builds at least annually and upon any changes to the build.	**A 1A-8.2** Verify the documentation exists for the maintaining of an up-to-date inventory of SCD system builds and of their policy to conduct vulnerability assessments.	An up-to-date list of SCDs and its associated firmware and software helps preserve the product's integrity throughout the product's life cycle. Annual vulnerability tests help to ensure vulnerabilities are kept to a minimum or are non-existent.
A 1A-8.3 The product vendor should develop and deploy patches and other device updates in a timely manner.	**A 1A-8.3** Verify that documentation exists for the development and deployment of patches and other device updates in a timely manner.	Timely firmware or software patches are critical to maintaining the integrity of the SCD while reducing the risk of the device being susceptible to exploit.

Products using SCDs	Evaluation Procedures	Guidance
A 1A-8.4 The product vendor should deliver updates in a secure manner with a known chain-of-trust. Security patches should be distributed in a manner that prevents malicious individuals from intercepting the updates in transit, modifying them, and then redistributing them to unsuspecting customers.	**A 1A-8.4.a** Verify that documentation exists for the delivery of updates in a secure manner with a known chain-of-trust.	A secure process for delivery of SCD firmware and software is essential to ensure that the integrity of the firmware, software and device are preserved.
	A 1A-8.4.b Verify the manner in which the updates are delivered is adequate and prevents malicious individuals from intercepting the updates in transit, modifying them, and then redistributing them to unsuspecting customers.	
A 1A-8.5 The product vendor should maintain the integrity of patch and update code during delivery and deployment.	**A 1A-8.5.a** To the extent that the product is integral to its own update process, verify that it maintains the integrity of patch and update code during delivery and deployment.	The intent is to ensure that the integrity of the patch and update code is maintained during delivery and deployment. For instance, upon completion of the product development and deployment, the software should undergo a validity and verification check, such as, a checksum test.
	A 1A-8.5.b If the product is not integral to its own update process, verify that any ancillary process integral to the update process maintains the integrity of patch and update code during delivery and deployment. ***Note:*** *If the ancillary process is not a product of the vendor, testing is not required.*	
A 1A-9 The product vendor should implement secure processes for handling account data when troubleshooting. Processes should include securely delete any PAN or SAD used for debugging or troubleshooting purposes. These data sources should be collected in limited amounts and collected only when necessary to resolve a problem, encrypted while stored, and deleted immediately after use. Alternatively, the vendor attests that they never collect account data for troubleshooting/maintenance purposes.	**A 1A-9.1** Confirm that the documented procedures include steps for securely deleting PAN or SAD used for debugging or troubleshooting purposes.	Adequately securing account data and production information is always of prime importance. When this information is used for debugging or troubleshooting, it should be adequately protected with the same level of controls as in production environments and this is especially important when debugging or troubleshooting occurs in non-production environments. Use of scrubbed or masked techniques may be considered.

Products using SCDs	Evaluation Procedures	Guidance
A 1A-10 Product vendor should implement tamper-detection mechanisms for devices and provide related instructions to entities implementing the tokenization product.	**A 1A-10.a** Verify that procedures exist for the detection of tampered devices in the vendor's possession.	Tamper-detection controls provide notification of physical alternation or damage of the device.
	A 1A-10.b Verify that the related instructions exist in the TIG.	

Annex B – Tokenization Installation Guide (TIG) (*Normative*)

This annex is ***normative***. The entity that produced, manufactured, and/or developed the tokenization product should provide a Tokenization Installation Guide (TIG) that describes how the product or device (e.g., tokenization appliance) should be installed and configured to ensure an appropriate level of security. The TIG should be provided to the implementer of the tokenization product—e.g., merchant, acquirer, or service provider.

Recommended Content for Tokenization Installation Guide

Tokenization products will have both hardware and software components. For example, tokenization appliances will have firmware and software components, and similarly, tokenization software applications may have dependent hardware devices and SCDs. The tokenization product vendor should include relevant information in the TIG that covers all components—both hardware and software—of the tokenization product, including dependent SCDs (e.g., an HSM) that are required by the tokenization product.

Some of the following TIG content may not be applicable for certain tokenization products. If this is the case, the vendor should be able to provide justification within the TIG.

TIG – I: Tokenization Installation Guide – Recommended Content for all Tokenization Products
1 Tokenization product details: ▪ Tokenization product name ▪ Tokenization Vendor name ▪ Product Model Name/Number ▪ Firmware Version Number (if applicable) ▪ Application Version Numbers (if applicable) ▪ All hardware and software components of the tokenization product, including any dependent components that are separate to the product and which are necessary for functionality of the tokenization product. ▪ SCD Manufacturer (if applicable) ▪ Description of the environment in which the product is intended to operate—e.g., attended, unattended, physically secure, publicly accessible, or mobile.

TIG – I: Tokenization Installation Guide – Recommended Content for all Tokenization Products
2 Include the following as applicable to the tokenization product: ▪ Describe the logical and physical technical architecture of all of the solution components including typical integration points with existing infrastructure—e.g., web proxy, email gateway, etc. ▪ Provide details on hardware and OS infrastructure requirements, software or appliance, web server, application server, database requirements, tiered logical architecture, application dependencies—e.g., third-party software or open source usage, etc. ▪ Attach logical and physical architecture diagrams depicting how the solution components would be implemented. ▪ Method to distinguish PANs and tokens. (See GT 7.)
3 A description of the vendor's published versioning methodology for the tokenization product—i.e., both hardware and software components of the product—including: ▪ Details of versioning scheme, including the format of the version scheme (number of elements, separators, character set, etc.). ▪ Details of how security-impacting changes will be indicated by the versioning scheme. ▪ Details of how other types of changes will affect the version. ▪ Details of any wildcard elements that are used, including that they will never be used to represent a security-impacting change.
4 Details of the tokenization product's functions, including: ▪ A description of the purpose and tokenization methods for all tokenization functions performed by the product.
5 Instructions on how to install and set up the tokenization product for correct functioning of all tokenization functions.
6 Description of the environment in which the product is intended to operate—e.g., attended, unattended, physically secure, publicly accessible, or mobile.
7 A detailed description and data flow diagram of how the tokenization product stores, processes and/or transmits PAN.
8 If the tokenization product stores PAN for any tokenization process, outside of the CDV, describe the methodology or processes used by the product to securely delete PAN upon completion of processing.
9 Details about how the application outputs clear-text PAN, including: ▪ Description of any tokenization product functions that allow for the output of clear-text PAN—for example, through the use of "whitelisting" BIN ranges. ▪ Instructions for configuring the tokenization product to permit only authorized personnel to access functions that allow for output of clear-text PAN data.
10 Instructions for configuring secure authentication—including administrative, assigning privileges, adding user identifiers, passwords, etc.

TIG – I: Tokenization Installation Guide – Recommended Content for all Tokenization Products
11 Procedures for the secure disposal of devices, including how to render sensitive data irrecoverable prior to device disposal.
12 List of protocols, services, and ports used by the tokenization product.
13 Instructions for use of secure communication methods, consistent with the product's communication interfaces: ▪ A list of the tokenization product's external communication methods. ▪ A description of what each external communication method is used for by the tokenization product. ▪ Instructions for how to configure each of the tokenization product's external communication methods for secure functioning.
14 For all configurable options provided with the tokenization product, provide necessary instructions for the appropriate security settings.
15 Specific instructions for installing and connecting tokenization appliances to maintain the integrity of tokenization product, including any permitted connections to other devices.
16 If the tokenization product shares resources, include: ▪ A list of shared resources. ▪ A description of how the device connects to and/or uses shared resources. ▪ Instructions for configuring the tokenization product for secure integration with shared resources.
17 Instructions for performing pre-installation inspection procedures, including physical and functional tests and visual inspection, to verify tokenization product components have not been tampered with or compromised. Also, provide instructions on what to do if tampering or compromise is discovered.
18 A description of how tokenization product enforces secure application installations, upgrades, and updates.
19 Instructions for backing out or uninstalling applications and application updates.
20 Instructions for rendering cryptographic keying material irretrievable, to include: ▪ Detailed procedures for rendering cryptographic material irretrievable. ▪ Instructions on how to re-encrypt historic data with new keys, including procedures for maintaining security of clear-text data during the decryption/re-encryption process. ▪ Instructions on how to transition data to the updated version prior to destruction of previous version keys.
21 Instructions on how the tokenization application enforces strong authentication for any authentication credentials—for example, user identifiers, passwords—that the application generates or manages. Refer to GT 9 and, as applicable, RC 2A-2, RN 2B, and RN 3B.

TIG – I: Tokenization Installation Guide – Recommended Content for all Tokenization Products
22 Detailed instructions on how to physically secure tokenization devices to prevent unauthorized removal or substitution, including specific examples of how devices can be physically secured.
23 Detailed procedures for performing physical inspections of tokenization devices to detect tampering or modification, including: ▪ Description of tamper-detection mechanisms. ▪ Guidance for physical inspections, including photographs or drawings of the device illustrating what to inspect—for example, missing or altered seals or screws, extraneous wiring, holes in the device, or the addition of labels or other covering material that could be used to mask damage from device tampering. ▪ Details of device weight and/or how to determine the correct weight of the device for a given configuration.
24 Instructions for secure remote access to the tokenization product, including: ▪ Description of the multi-factor authentication mechanisms supported by the application. ▪ Instructions on how to configure the application to support multi-factor authentication.
25 If the tokenization product facilitates non-console administrative access, include instructions on how to configure the application to use strong cryptography (such as SSH, VPN, or TLS) for encryption of all non-console administrative access to tokenization product.
26 Who to contact/or what steps to take if product fails.

The vendor may include additional information in the TIG that the vendor considers useful to the entity implementing the tokenization product. For example, consider the following:

- ***Provide benchmarking details of the capacity and performance, scalable and load balancing, synchronization and backups.***
- ***Define the memory and disc requirements for each of the solution components.***
- ***Provide document encryption export restrictions—e.g., export licenses findings or classification from Department of Commerce.***

Annex C – Minimum Key Sizes and Equivalent Key Strengths for Cryptographic Primitives (*Normative*)

This annex is ***normative.***

Wherever the tokenization process depends on the use of cryptographic primitives, the effective security strength of any keying material should meet that defined in Table C-1 below. Any reliance on a cryptographic hash should be in accordance with Table C-2 under "Secure Hash Algorithms." Any tokenization process that uses random or deterministic random numbers should be in accordance to the section below on Random Number Generators.

Cryptographic Algorithms

The following are the minimum key sizes and parameters for the algorithm(s) in question that should be used in connection with key transport, exchange, or establishment and for data protection in a tokenization product that uses encryption:

Algorithm	TDEA	AES	RSA	Elliptic Curve	DSA/D-H
Minimum key size in number of bits:	Not Allowed	128	3072	256	3072/256

A key-encipherment key should be at least of equal or greater strength than any key it is protecting. This applies to any key-encipherment key used for the protection of secret or private keys that are stored or for keys used to encrypt any secret or private keys for loading or transport. The following algorithms and bits of security are considered equivalent for this purpose:

Table C-1

Bits of Security	Key Lengths				
	Symmetric key algorithms	RSA	Elliptic Curve	D-H	
112	3TDEA [168-bit key]	2048	224-225	2048/224	Not Allowed
128	AES-128	3072	256-383	3072/256	
192	AES-192	7680	384-511	7680/384	
256	AES-256	15360	512+	15360/512	

3TDEA refers to three-key triple DEA keys exclusive of parity bits. The RSA key size refers to the size of the modulus. The Elliptic Curve key size refers to the minimum order of the base point on the elliptic curve; this order should be slightly smaller than the field size. The DSA key sizes refer to the size of the modulus and the minimum size of a large subgroup.

For implementations using Diffie-Hellman (DH) or Elliptic Curve Diffie-Hellman (ECDH):

- **DH implementations** – Entities should securely generate and distribute the system-wide parameters: generator *g,* prime number *p* and parameter *q,* the large prime factor of (*p* - 1). Parameter *p* should be at least 3072 bits long, and parameter *q* should be at least 256 bits long. Each entity should generate a private key *x* and a public key *y* using the domain parameters (*p, q, g*).
- **ECDH implementations** – Entities should securely generate and distribute the system-wide parameters. Entities may generate the elliptic curve domain parameters or use a recommended curve (See *FIPS186-4*). The elliptic curve specified by the domain parameters should be at least as secure as P-256 (or P-384). Each entity should generate a private key *d* and a public key *Q* using the specified elliptic curve domain parameters. (See *FIPS186-4* for methods of generating *d* and *Q*).
- Each private key should be statistically unique, unpredictable, and created using an approved random number generator as described in this document.
- Entities should authenticate the DH or ECDH public keys using DSA, ECDSA, a certificate, or a symmetric MAC (see *ISO 16609 – Banking –Requirements for message authentication using symmetric techniques*). One of the following should be used: MAC algorithm 1 using padding method 3, MAC algorithm 5 using padding method 4.

Note that TDEA should not be used in tokenization products.

The following table lists the approved modes of operation for each algorithm:

Algorithm	Modes
AES	CTR, OCB, CBC, OFB, CFB, FFn (ECB should not be used if encrypting more than one block)
RSA	RSAES-OAEP
ECC	ECDH, ECMQV or ECDSA (for key negotiation), ECIES
D-H	DHE, EHD

Secure Hash Algorithms

Current popular hashes produce hash values of length n = 128 (MD4 and MD5) and n = 160 (SHA-1), and therefore can provide no more than 64 or 80 bits of security, respectively, against collision attacks. To avoid introducing security weakness via any hash function used, the hash function should provide at least as many bits of security as does the cryptographic algorithm used, and in no case less than 128-bits. Table C-2 lists standardized hash algorithms and associated effective bits of security.

Table C-2

Bits of Security	Hash Algorithm
128	SHA-256
128	SHA3-256 (SHA-3 family, a.k.a., Keccak)
192	SHA3-384
256	SHA-512
256	SHA3-512

Random Number Generators

The proper generation of random number is essential to the effective security for cryptographic key generation and is an essential primitive for non-cryptographic tokenization products. Where deterministic random number generators are used, the requirements of *NIST Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators* apply, except for the Dual_EC_DRBG algorithm, which should not be used.

The number of bits of entropy should be equal to or greater than the required number of bits of security.

Annex D – Cryptographic Key-Management Life Cycle (*Informative*)

This annex is intended to be ***informative*** only. It is intended to describe the steps typical of the management life cycle for cryptographic keys or keying materials. While cryptographic keys (or analogous materials that must remain secret to be effective) may have long lives in tokenization products, they still have a life cycle. For an illustration of common life-cycle elements see ISO 11568-1.

Cryptographic Key-Management Life Cycle Process Definitions

Process	Definition
Generation	Key generation involves the creation of a new key for subsequent use.
Storage	Key storage involves the holding of a key in one of the permissible forms.
Backup	Key backup occurs when a protected copy of a key is kept in storage during its operational use.
Distribution and loading	Key distribution and loading is the process by which a key is manually or electronically transferred into a secure cryptographic device.
Use	Key use occurs when a key is employed for the cryptographic purpose for which it was intended.
Replacement	Key replacement occurs when one key is substituted for another when the original key is known or suspected to be compromised or the end of its operational life is reached.
Destruction	Key destruction ensures that an instance of a key in one of the permissible key forms no longer exists at a specific location. Information may still exist at the location from which the key may be feasibly reconstructed for subsequent use.
Deletion	Key deletion is the process by which an unwanted key, and information from which the key may be reconstructed, is destroyed at its operational storage/use location. A key may be deleted from one location and continue to exist at another—e.g., for archival purposes.
Archive	Key archive is the storage process for a key that is no longer in operational use at any location.
Termination	Key termination occurs when a key is no longer required for any purpose and all copies of the key and information required to regenerate or reconstruct the key have been deleted from all locations where they ever existed.

Operational Life of a Key

The operational life of a key depends on many factors in a tokenization product including:

- The effective cryptographic strength of the underlying algorithm for a given key length.
- Whether the key or related keying material is suspected of compromise.
- Change in vendor support of product or need to replace product.
- Technological advances that make previously infeasible attacks feasible (i.e., the risk equation changes for the worse).
- Change of ownership where a change of keys is associated with a change in assignment of liability.
- Regulatory requirements, contractual requirements, or policy (cryptoperiod) that mandates a maximum operational life.

Because these and other factors may force an end-of-key-life, any organization developing a tokenization product that depends on cryptographic materials (or equivalent secrets) should include a mechanism for supporting the cryptographic key-management life cycle.

Annex E – Use Cases for Tokenization (*Informative*)

This annex is intended to be ***informative*** only. The purpose of this section is to illustrate a business use case for each type of token that has been discussed within this document. It is important to note that these use cases do not preclude other implementations of a particular tokenization process. The use cases are examples and are intended to be illustrative only.

Irreversible Tokens

Authenticatable Irreversible Tokens

An authenticatable irreversible token could be used to support warranty enforcement where the presentation of the payment card that was allegedly used for the purchase could be verified as the one used. This situation may happen when a customer has lost their receipt and needs a way to prove the transaction.

Non-Authenticatable Irreversible Tokens

A non-authenticable irreversible token might be used to support legacy applications that require a validly formatted, generally unique value in the PAN data field. While this value cannot be used to obtain the original PAN, this could be an alternative to a costly system replacement that may be required to implement another form of tokenization.

Reversible Cryptographic and Non-Cryptographic Tokens

Reversible cryptographic and non-cryptographic tokens have very similar if not identical use cases. A reversible cryptographic or non-cryptographic token implementation may support fraud investigations or situations wherein:

- Merchant needs PAN for other entities with which they interact.
- Merchant needs PAN for follow-on transactions.
- Acquirer needs PAN for anti-money-laundering operations.

A typical process in these scenarios may include the business unit that needs the original PAN submitting the token for de-tokenization. Then, after proper authentication, a PAN is returned in a secure manner (e.g., encryption), and when no longer needed, the PAN is deleted or destroyed.

Hybrid Tokenization Products

A hybrid tokenization product may, for example, generate a cryptographic token where the cryptographic key is either ephemeral or disposed of once the token is created. This token is then stored in a CDV with the mapping to its corresponding PAN. As a result, a hybrid tokenization product may need to meet the criteria for both Cryptographic and Non-Cryptographic Tokens. A hybrid product may have components that require separate evaluation (e.g., CDV and the appliance that generates tokens). Another example is where the tokens are based on a deterministic random number generator (DRNG, also known as a pseudo RNG (PRNG)), which is based on cryptographic primitives.

Annex F – Illustration of Tokenization and P2PE (*Informative*)

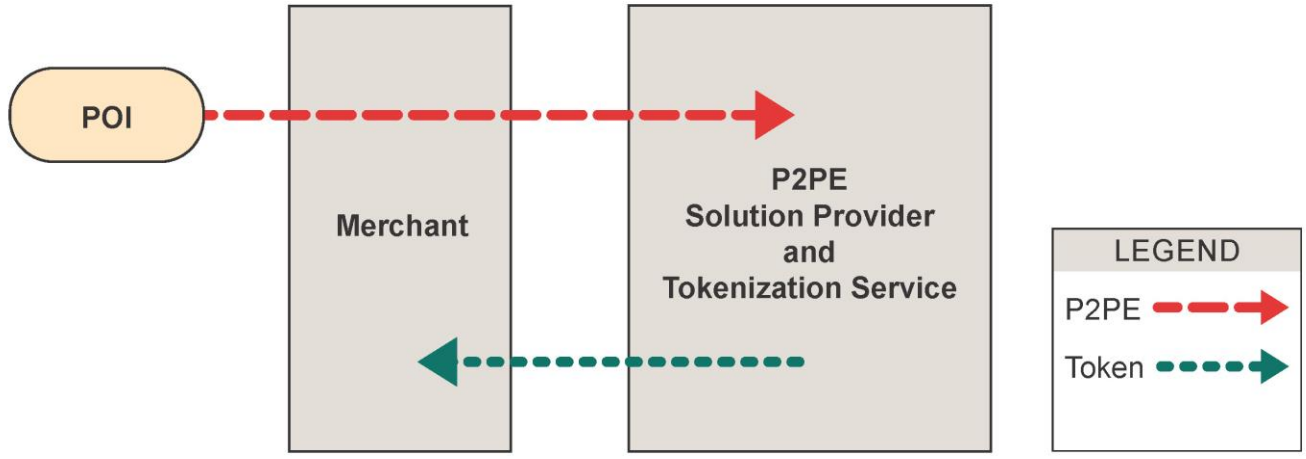
This annex is intended to be ***informative*** only and meant to illustrate a hypothetical implementation.

Tokenization may be used in conjunction with Point-to-Point Encryption (P2PE) to provide additional capabilities to merchants. The tokenization might occur at the POI or after processing by the P2PE solution provider—e.g., if the tokenization service provider is the same party as the P2PE solution provider or a separate entity.

The PCI P2PE standard provides a mechanism for potential scope relief independent of any tokenization. For a current list of P2PE solutions, please refer to PCI Security Standards Council website.

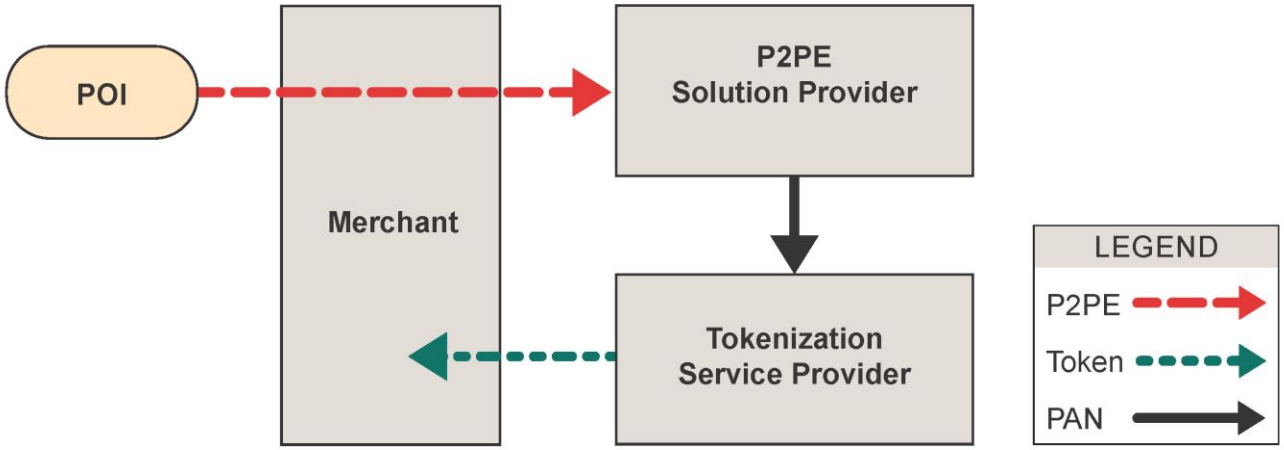
In Illustration 1, a typical P2PE transaction occurs. This solution provider is both the P2PE solution provider and the tokenization service provider. In its role as tokenization service provider, it produces the token and provides the token to the merchant.

Illustration 1



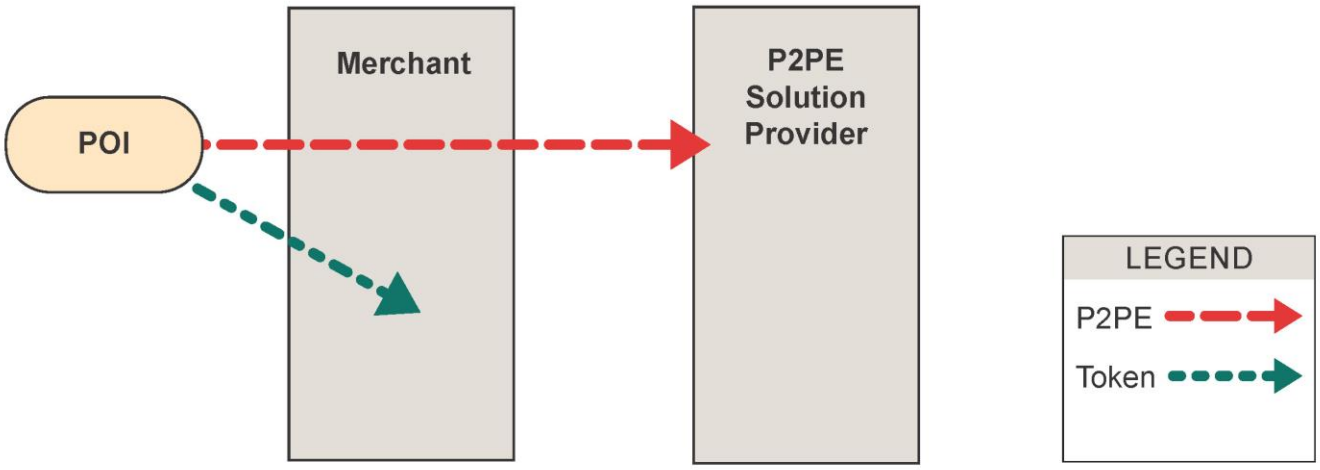
In Illustration 2, a typical P2PE transaction occurs. This P2PE solution provider securely transmits the PAN to the tokenization service provider. The tokenization service provider produces the token and provides the token to the merchant.

Illustration 2



In Illustration 3, a typical P2PE transaction occurs. In parallel, the POI device produces a token, which goes directly to the merchant.

Illustration 3



***Note**: The security of these implementations depends on many factors that are outside the scope of this document.*

Annex G – Formal Security Objective of a Tokenization Product (*Informative*)

This annex is intended to be ***informative*** *only*.

The security objective of any tokenization process is to remove the value of any digits not taken from the original PAN (if any) in the resulting token. This can be stated more precisely as the following set of circumstances:

1. There are two hypothetical attackers.
2. The goal of each attacker is to guess one or more digits of a PAN given a token.
3. The first attacker is given a collection of tokens.
4. The second attacker is given a collection of token and PAN pairs, where the second attacker can choose a PAN and get the corresponding token, or choose a token and get the corresponding PAN.

For any token that neither the first nor second attacker has seen previously, the second attacker should have no advantage over the first attacker in guessing any digits of the corresponding PAN.

Annex H – Examples of Tokens (*Informative*)

This annex is ***informative*** only and describes nine examples of tokens. In particular, this annex shows example formats of tokens and not any specific techniques used for their creation. Further, regardless of their format, all tokens should meet all applicable tokenization Guidelines/Best Practices in this document. Table H-1 is not intended to be all-inclusive, nor is it intended to preclude any other implementation of tokens.

Table H-1: Example Tokens

Examples	PAN	Token	Comment
A	3124 005917 23387	7aF1Zx118523mw4cwl5x2	This example shows a token that consists of alphabetic and numeric characters and contains more digits than the original PAN.
B	4959 0059 0172 3389	729129118523184663129	This example shows a token that consists of only numeric characters and contains more digits than the original PAN.
C	5994 0059 0172 3383	599400x18523mw4cw3383	This example shows a token that consists of a truncated PAN (first 6, last 4 of PAN are retained) with alphabetic and numeric characters replacing the middle digits. Also, the resulting token has several more characters than the original PAN.
D (FP)	3124 005917 23387	1234 5098765 6574	This is an example of a format-preserving (FP) token implementation. Here, the token is identical to PAN in structure and character set (Luhn check could even hold).
E	3124 005917 23387	T3245 918234 4251	This example shows a token that is almost identical in structure and character to the PAN except for a character indicating that it is a token.
F (FP)	4959 0059 0172 1234	12345 736251 1234	This is an example of a format-preserving (FP) token. In this example, the first 12 digits of the PAN are tokenized and the resulting token also retains the last four digits of the PAN.
G	3124 005917 23387	312400 F1Zx7a 3387	Token retains the first 6 and last 4 digits of the original PAN. This example shows the resulting token that retains the first 6 and last 4 digits of the original PAN and the middle six digits are tokenized.

Examples	PAN	Token	Comment
H (FP)	4959 0059 0172 1234	4123 0000 3405 7897	This example shows a format-preserving token that retains the first digit of the original PAN and the Luhn check is valid.
I (FP)	3124 005917 23387	3124 006843 43387	This is an example of format-preserving (FP) token. In this example the first 6 and last 4 digits were retained from the card. No new alpha-numeric characters were introduced. Luhn check may or may not be preserved.

Annex I – Recursive Tokenization (*Informative)*

This annex is intended to be ***informative*** only to illustrate the tokenization of a token—i.e., recursive tokenization. Figure 6 is intended to illustrate a token (Token 1) being submitted to a tokenization product, which then tokenizes the token (Token 1) and outputs a new token (Token 2).

Figure 6: Tokenization of a token

Annex J – Token-to-Token Conversions (*Informative)*

This annex is intended to be ***informative*** only.

GT 11 states the following:

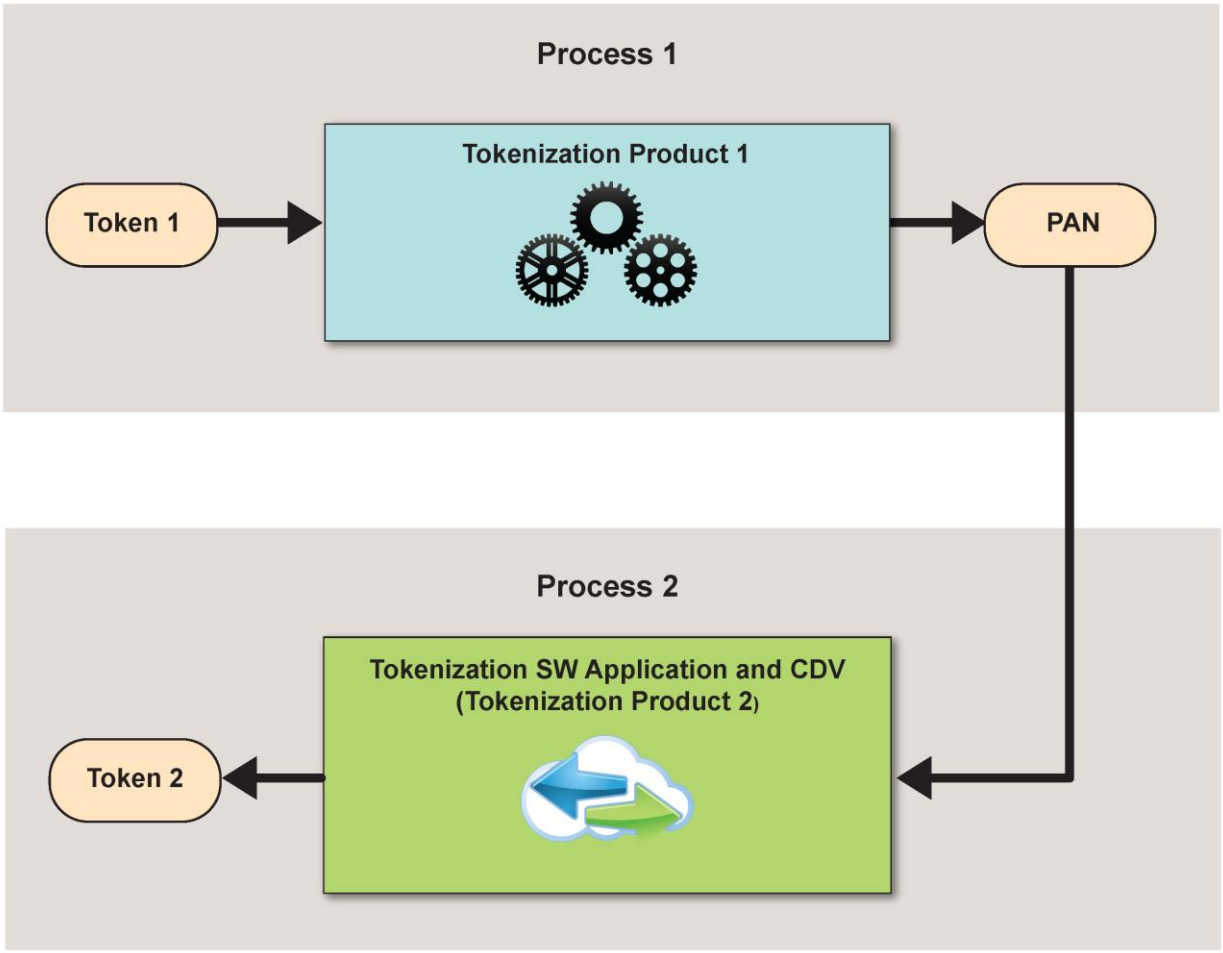
> Converting from a token produced under one system (or cryptographic key or non-cryptographic process) to a token produced under another system (or cryptographic key or non-cryptographic process) should require an intermediate PAN state—i.e., invocation of de-tokenization. This assures that the old token is independent of the new token. (See Annex J – Token-to-Token Conversions.)

Note:

- *The tokenization of a token is permitted. (See Note 1 of GT 11 and Annex I – Recursive Tokenization.)*
- *Irreversible tokenization products will not be capable of such conversions.*

This annex is intended to illustrate how that process may work. Figure 7 shows a process (Process 1) that converts the token into a PAN (i.e., de-tokenization). Then the PAN goes through another tokenization process (Process 2), which is completely separate from the first. It is important to note that the two processes should be separate.

Figure 7: Token-to-Token Conversion



Annex K – Security Models and Formal Proofs (*Informative*)

This annex is ***informative***.

Many formal security models exist that may be useful in defining the security policy of the tokenization product. Products that are designed and architected in conformance with a security policy that is then codified through an appropriate security model are more easily validated and are less likely to contain unintended access paths.

Some well-known security models include the following:

- Bell—LaPadula Confidentiality Model
- Biba Integrity Model
- Brewer—Nash (Chinese Wall)
- Clark—Wilson Integrity Model
- Graham—Denning Model
- Harrison—Ruzzo—Ullman Model

Department of Defense Trusted Computer System Evaluation Criteria (DoD TCSEC) [DoD 5200.28-STD] is an historic document based on an even earlier government effort [CSC-STD-00l-83, l5 Aug 83] that formalized evaluating information security in the context of a formal model. The *Common Criteria for Information Technology Security Evaluation* (CC) [(ISO/IEC 15408] is the modern prodigy of this [https://www.commoncriteriaportal.org/cc/].

Formal security proofs use mathematics to model the behavior of a system. Statements about the behavior of the system can then be evaluated. These hypotheses can be proven or disproven. By demonstrating that the tokenization product acts in accordance to the model, the security proof can, by analogy, be extended to the system the model represents.

Automated tools are often used to assist in developing formal security proofs for software. One such tool is Coq. Coq is a formal proof-management system. It provides a formal language to write mathematical definitions, executable algorithms and theorems, together with an environment for semi-interactive development of machine-checked proofs. [http://coq.inria.fr/] As another example, Isabelle (proof assistant) is an interactive theorem-prover that has been used to formalize theorems including correctness of security protocols. [http://isabelle.in.tum.de/]

Glossary

Term	Definition
Acquiring token	Tokens created by the acquirer, merchant, or a merchant's service provider. This token is created after the cardholder presents their payment credentials. Acquiring tokens may be used as part of the authorization process, including card-on-file transactions.
Adequate	The technical ability to meet the requirement. The intent is to permit the assessor flexibility in making this judgment.
Application program interface (API)	A set of routines, protocols, and tools for building software applications. (For security guidance on coding of API, refer to CERT Coding Standards (www.securecoding.cert.org) and to guidance specific to the operating system or programming environment.)
Bespoke	Software that is specially developed for the entity to the entity's custom requirements by, for example, an in-house software development group or an external software development company.
Card data vault (CDV)	The central repository of cardholder data that is used by the token mapping process.
Cardholder data environment (CDE)	See *PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms.*
Computationally infeasible	The principle that the best-known cryptanalytic attack cannot succeed within a practical length of time (e.g., decades) because it requires excessive computational resources—e.g., "zillions" of bytes of memory or computer cycles.
Cryptographic primitive	Cryptographic algorithms that are frequently used to build cryptographic protocols for computer security systems. These routines include, but are not limited to, one-way hash functions and encryption functions. A taxonomy of cryptographic primitives may be found in Figure 1.1 of the *Handbook of Applied Cryptography* [Menezes, Alfred J., Paul C. van Oorshot, and Scott A. Vanstone. CRC Press, Boca Raton, 1997, p5].
De-tokenization	The process of obtaining the PAN from its associated token.
Evaluated API	APIs that have been evaluated against secure coding standards to ensure they function properly.
Irreversible tokens	A token created such that no feasible mechanism exists to re-associate it with the original PAN.

Term	Definition
Logically bound	Describes one or more values or fields tightly associated within a system by cryptographic means (e.g., digital signature, secure hash or message authentication code (MAC)) or by system-enforced association (e.g., explicit field attributes).
Multi-factor authentication (MFA)	Method of authenticating a user whereby two or more factors are verified. These factors include something the user has (such as a smart card or dongle), something the user knows (such as a password, passphrase, or PIN), or something the user is or does (such as fingerprints, other forms of biometrics, psychometrics, etc.).
Non-console administrative access	Refers to logical administrative access to a system component that occurs over a network interface rather than via a direct, physical connection to the system component. Non-console administrative access includes access from within local/internal networks as well as access from external, or remote, networks.
Primary account number (PAN)	See *PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms.*
PAN space exhaustion	The PAN space is the set of all possible PAN (per ISO definitions that would be from 13 to 19 digits with some restrictions based on what values are valid for a given sub-field of the PAN). PAN space exhaustion is the process of trying every probable value until you find the right one. It assumes the existence of an oracle (i.e., a means for testing each value).
Reversible token	A token for which a mechanism exists that permits obtaining its associated PAN.
Secure cryptographic device (SCD)	A set of hardware, software, and firmware that implements cryptographic processes (including cryptographic algorithms and key generation) and is contained within a defined cryptographic boundary. Examples of secure cryptographic devices include host/hardware security modules (HSMs) and point-of-interaction devices (POIs) that have been validated to PCI PTS.
Sensitive authentication data (SAD)	See *PCI DSS and PA-DSS Glossary of Terms, Abbreviations, and Acronyms.*
Static token	Any token that has a one-to-one relationship with a given PAN such that the tokenization process for that PAN always results in the same token.
Token	For purposes of the *Tokenization Product Security Guidelines*, the term "token" means a value that replaces a PAN (and optionally other CHD).

Term	Definition
Token mapping	Token mapping is the relationship between the token and the PAN. For instance, a PAN may be mapped to a token by encryption with a secret key or by a data look-up process within a CDV (where the PAN/token relationship is secret).
Tokenization appliance	A device (that is, a PCI-listed SCD, FIPS 140-2 Level 3 [validated to FIPS 140-2 Overall Level 3, operated in FIPS mode and initialized to Overall Level 3 per security policy or above], a device Independently validated to ISO 13491-1 or self-contained product (e.g., package hardware and software tokenization product)) used for tokenization functions, de-tokenization functions, or any functions involving a CDV.
Token-only components	Components that contain the token and do not contain PAN or SAD.
User	Any individual (i.e., person) that is not a consumer (e.g., vendor personnel, administrators, contractors, or merchant personnel).

Related Publications

The following American National Standards, International Standards, European Payment Council, NIST, and PCI standards are applicable and related to the information in this document.

Standard/Resource	Source
ANSI X9.24: Retail Financial Services Symmetric Key Management	ANSI
ANSI X9.119-2012 Retail Financial Services —Requirements for Protection of Sensitive Payment Card Data — Part 1: Using Encryption Methods	ANSI
ANSI INCITS 359: American National Standard for Information Technology – Role Based Access Control	ANSI
SEPA Cards Standardisation (SCS) "Volume" – Book of Requirements [Chapter 5]	EPC
ISO/IEC 7813 Information Technology – Identification Cards – Financial Transaction Cards	ISO
ISO/IEC 11568-1 Banking – Key Management (Retail) – Part 1: Introduction to Key Management	ISO
ISO/IEC 11770 Information Technology – Security Techniques – Key Management	ISO
ISO/TR 14742:2010 Financial services — Recommendations on cryptographic algorithms and their use	ISO
ISO/IEC 18031:2011 Information Technology – Security Techniques – Random bit generation	ISO
ISO 13491-1:2007 Banking – Secure cryptographic devices (retail) – Part 1: Concepts, requirements and evaluation methods	ISO
NIST Special Publication 800-57 Recommendation for Key Management. July 2012.	NIST
NIST Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators	NIST
NIST Special Publication 800-130 – A Framework for Designing Cryptographic Key Management Systems. August 2013	NIST
Information Supplement: PCI DSS Tokenization Guidelines.	PCI SSC
Payment Card Industry Data Security Standard (PCI DSS)	PCI SSC
Payment Card Industry Payment Application Data Security Standard (PA-DSS)	PCI SSC
PCI PTS POI Modular Security Requirements	PCI SSC