protegrity
The Tokenization Experts

Securosis

Tokenization Guidance:
How to reduce PCI compliance costs

By Adrian Lane, Analyst and CTO, Securosis

Who should read it

- Security experts, IT and business professionals interested in learning more about tokenization
- Anyone familiar with the PCI DSS Tokenization Guidelines
- Anyone who wants better security, lower risk and significant cost reductions for compliance efforts

What it's about

- Key principals to follow when selecting a token solution
- A candid look at the PCI DSS Tokenization Guidelines
- Recommendations for the auditors
- Advice on how to use tokenization within scope of the Payment Card Industry Data Security Standard

Contents

"I believe the most important takeaway is validation of the fact that storing tokens instead of PANs can help reduce the amount of cardholder data in the environment, potentially reducing the merchant's effort to implement PCI DSS requirements." - Ulf Mattsson, CTO, Protegrity

"The Tokenization Supplement is the first tokenization guidance from the council that merchants can rely on, and is further validation of this data security approach. It has positive implications for using tokenization, also beyond PCI." - Yigal Rosenberg, Vice President, Chief Architect, Protegrity

Introduction

On August 12, 2011, the PCI task force studying tokenization published an "Information Supplement" called the PCI DSS Tokenization Guidelines. Commonly known as the 'Tokenization Guidance' document, it discussed dos and don'ts of using token surrogates for credit card data. Tokenizing payment data holds the promise of improving security while reducing auditing costs, generating great demand amongst the merchant community. The introductory section of the supplement identifies the key principles to follow when selecting a token solution — ideas that can be distilled into the single question on the mind of every merchant, payment processor and auditor: How does tokenization alter PCI compliance?

The problem is that the document does not actually answer this central question, nor does it adequately address any of the key points raised in the introduction. For a set of guidelines, the supplement is sorely lacking in actual guidance. Even the section on "Maximizing PCI DSS Scope Reduction" is a collection of broad generalizations on security, rather than practical advice or definitive statements on scope. After spending the better part of the last two months with this wishy-washy paper, I have come up with a better title: "Grudging Acknowledgement of Tokenization Without Guidance".

So *this* paper will address merchant concerns left dangling by the PCI Council.

"We read the guidance but we don't know what falls out of scope!" is the universal merchant response to the tokenization information supplement. "Where are the audit guidelines?" is the second most common thing we heard. The tokenization guidelines provide an overview of the elements of a tokenization system, along with the promise of reduced compliance requirements, but they don't provide a roadmap to get there. And let's make one thing very clear right from the start: Our research shows a lot of interest in tokenization because it promises better security, lower risk and — potentially — significant cost reductions for compliance efforts. Merchants want to reduce the work they must do in order to comply with the PCI requirements — which means they are interested in tokenization technologies. Security and lower risk are benefits as well — albeit secondary in the minds of merchants who assume they are already doing what they can to keep payment data secure. But without a concrete idea of the actual cost reduction — or even an understanding of how they will be audited once tokenization is deployed — they are dragging their feet on adoption.

During the research process for this paper it became clear that many people closely associated with tokenization technologies in the payment space came to the same conclusions when reading the supplement. In fact we learned that the recommendations made by the tokenization special interest group were not adopted by the PCI council. Instead, they released 'guidelines' which are lacking in guidance. This white paper is the result of dozens of interviews; hundreds of hours of research; and a deep dive into the deployment, auditing, and scope reduction concerns people have regarding tokens. We have vetted our recommendations with as many qualified assessors as possible to ensure the advice will hold up to PCI requirements with minimal friction during the assessment process.

Tokenization is a simple technology with a clear value proposition, so there is no good reason to omit a basic cookbook for scope reduction. We will take the guesswork out and provide real guidance for evaluating tokenization, and clarify how to benefit from tokenization. This will be in the form of concrete, actionable steps for merchants

deploying tokenization, with checklists for auditors reviewing tokenization systems. We'll fill in the gaps from the PCI supplement, poke at the topics they found politically unpalatable to discuss, and specify what you can reasonably omit from the scope of your assessment.

This paper will not be uncontroversial, as we advise against using some technologies and deployment models that frankly should not have been lumped into the supplement, because they don't simplify and reduce risks in the way any merchant should be looking for. Further, because the PCI guidelines are vague, we are sticking to sensible choices. We understand that some of these recommendations will make many interested stakeholders quite angry. This is unavoidable — our guidance is geared toward making the lives of merchants who buy tokenization solutions easier, rather than avoiding conflict with vendor products or PCI Council politics. No technology vendor or payment provider ever endorses guidance that puts their product or service in a bad light, so not everyone will agree with our technology recommendations.

This paper assumes the reader has some familiarity with tokenization and PCI DSS. If not, there is **a lot** of research on the Securosis blog and within the Securosis Research Library — free of charge — covering tokenization, PCI guidelines and related subjects. If you are interested in learning more about tokenization in general, we suggest you review some our previous papers on tokenization. Most helpful will be Understanding and Selecting a Tokenization System, Tokenization vs. Encryption: Options for Compliance and FireStarter: an Encrypted Value is **Not** a Token!

PCI Supplement Highlights

Before we provide specific advice on how to use tokenization within scope of the Payment Card Industry Data Security Standard, it's important that we cover the PCI DSS Tokenization Guidelines Information Supplement — which we refer to as "the supplement" for the remainder of this document. Understanding where the supplement fails is important as it frames the remaining discussion on scope reduction and audit requirements. The supplement's not all bad: there are useful deployment architecture models and a description of common pitfalls provided. But the lack of advice and the contradictory statements regarding scope reduction hinder secure deployments more than they help. So before we go any further, let's examine the good and the bad of the PCI DSS Tokenization Guidelines.

The supplement is intended to address how tokenization *may* impact Payment Card Industry (PCI) Data Security Standard (DSS) scope. The supplement is divided into three sections: a discussion of the essential elements of a tokenization system, PCI DSS scoping considerations, and new risk factors to consider when using tokens as a surrogate for credit card numbers. It is aimed at merchants who process credit card payment data and fall under PCI security requirements. If you have not already downloaded a copy we recommend you do so now. Even with its deficiencies, the guidance provides plenty of good basic information, so we are not advising you throw it away. Our goal is to illuminate key questions left unanswered — then address those concerns.

The bulk of the PCI document covers tokenization systems as a whole: technology, workflow, security, and operations management. The tokenization overview does a good job of introducing what tokenization is, what tokens look like, and the security impact of different token types. Their diagrams do an excellent job of illustrating how token substitution fits within the payment processing flow, providing a clear picture of how an on-site tokenization system — or a tokenization service — works. The supplement stresses the need for authorization and network segmentation — the two essential security tools needed to secure a token server and reduce compliance scope.

The last section of the supplement helps readers understand the risks inherent in using tokens — which are new and distinct from the issues of credit card fraud. Remember, tokens are different than credit cards. While that sounds obvious, there is a risk your applications will use tokens as a logical credit card replacement for financial transactions, as opposed to a transaction reference. This is a concern as tokens then allow an attacker to commit fraud without the credit card number. Should hackers penetrate IT systems — even without credit cards — they can use tokens as financial instruments, then conduct transactions without needing to crack the token database. If the token can initiate a transaction, force a repayment, or be used as money, there is risk of fraud. This section of the supplement covers a couple of critical risk factors merchants need to consider; although this has little to do with the token service directly it does affect how merchants use tokens in their back office systems.

Those were the highlights of the supplement — now the lowlights. The section on PCI Scoping Considerations is convoluted and ultimately unsatisfying. If you wanted bacon, sorry, you're only getting half a piece of Sizzlean. Seriously, the supplement made me ask "Where's the beef?" Okay, we will stop mixing meats and metaphors. But to give credit where credit is due the supplement starts with great promise. They do a fantastic job answering the presales questions of tokenization buyers in section 1.3: simplification of merchant validation, verification of deployment, and risks unique to token solutions. The guidance does offer "scoping considerations," but it fails to provide advice or a definitive standard for auditing or scope reduction. That missing information is what triggered this research project, and so it is where we begin our analysis. Let's take a critical look at the supplement to spotlight the issues before we move on to our tokenization cookbook for merchants.

From the supplement:

"The security and robustness of a particular tokenization system is reliant on many factors including the configuration of the different components, the overall implementation, and the availability and functionality of the security features for each solution."

That sounded like information — but isn't. Ask yourself if that statement helps you in any way? Is this statement true for every software or hardware system? Yes, it is. Yet this ambiguous assertion reflects the core theme and embodies a bulk of their advice. The supplement does not help anyone answer the critical questions posed in the introduction, and as you wade through these pages, don't be surprised if you are lulled to sleep. Uselessly vague statements like this litter the whole document. Sadly, the first paragraph of the 'guidance' — a disclaimer repeated at the foot of each page, quoted from Bob Russo in the PCI press release — demonstrates the supplement's true nature:

"The intent of this document is to provide supplemental information. Information provided here does not replace or supersede requirements in the PCI Data Security Standard."

Tokenization does not supersede the PCI Data Security Standard but should replace some security controls and should reduce PCI DSS scope. It's not about layering — as Mr. Russo stated in the official PCI press release. Tokenization replaces one security model with another. Technically there is no need to adjust the PCI DSS specification to account for a tokenization strategy — they can happily co-exist — with tokenization removing systems from scope by removing access to PAN (Primary Account Number, i.e. a credit card number) entirely, and those which store payment data still subject to PCI DSS. But the lack of a clear definition of which is which, or what merchants are responsible for, are examples of the guidance's deficiency.

Strictly speaking, according to the supplement, PCI DSS scope can *never* be reduced with tokenization. For example,

section 2.2 emphatically states "If the PAN is retrievable by the merchant, the merchant's environment will be in scope for PCI DSS." Section 3.1, "PCI DSS Scope for Tokenization" starts from the premise that *everything* is in scope, including the tokenization server. But what falls out of scope and how is not clear in section 3.1.2, "Out-of-scope Considerations", where one would expect such information. Rather than define what is out of scope, it outlines many objectives to be met, apparently without regard for where the credit card vault resides or the types of tokens used. Section 3.2, "Maximizing PCI DSS Scope Reduction", states that "If tokens are used to replace PAN in the merchant environment, both the tokens, and the systems they reside on will need to be evaluated to determine whether they require protection and should be in scope of PCI DSS." How can anything then be out of scope? The merchant, and likely the auditor, must still review every system, which means there is no benefit of audit scope reduction.

Here's the deal: Tokenization — properly implemented — reduces security risks and **should** reduce compliance costs for merchants. Systems that have fully substituted PAN data with random tokens, and that have no method of retrieving credit card data, are out of scope. The Council failed to endorse tokenization as a recommended approach to securing data or securely performing financial transactions, or to provide more than a broad hand-wave for how this can happen.

If you are ready to begin your evaluation process, jump ahead to Merchant Guidance. If you are still trying to reconcile how the official supplement differs from our guidance, here are a few other topics to consider. While scope reduction is the biggest issue by far, there are additional areas where the PCI Council and tokenization task force punted that **must** be addressed:

- **Audit Guidelines:** How will merchants be judged? Provide guidelines so merchants know how to meet their obligations and guide auditors on how to approach compliance. Far too many questions are left open.
- **Update the self-assessment questionnaires:** The vast majority of merchants don't process enough transactions to warrant an on-site audit, but must complete the self-assessment questionnaire. There should be a series of clear questions, in layman's terms, to determine whether the merchant has implemented tokenization correctly.
- **Token Service Checklist:** What should you look for in a token service? The majority of merchants **will not** run their own tokenization server — what the PCI Council calls a Card Data Vault — instead they will buy this additional service from their payment providers. The migration process to replace PANs with tokens across your organization, and altering all applications to use tokens, can be complicated. A discussion of the impact on both software and process is needed, as this influences the selection of a token format and is likely to be a deciding factor when choosing between different solutions.
- **Provide Auditor Training:** Understanding how to audit a tokenization system and validate the migration from PAN to token storage is critical. Without specifications, Qualified Security Assessors (QSA's) are naturally concerned about what they can approve and worried about repercussions for accepting tokenization. If something goes wrong the first person who will be blamed is the auditor, so they keep everything in scope. Yes, it's CYA, but auditing is their livelihood.
- **Risks of Vendor Lock-in:** Tokenization affects the whole system, impacting data storage, key management systems, and business systems that rely on payment related data. A token system can *lock a merchant in with a specific payment processor!* Simply being aware of the issue up front allows merchants to take simple precautions during the initial deployment to minimize lock-in.

- **State a definitive position:** It would be best if the PCI Council came out and endorsed tokenization as a way to remove the PAN from merchant sites. Reading the guidance, we get a clear impression that the PCI Council would prefer that payment gateways/banks and cardholders be the *only* parties with credit card numbers. This would remove all credit cards from merchant exposure, and we support this goal — which should have happened 15 years ago. That sort of roadmap would help merchants, mobile application developers, POS device manufacturers, encryption vendors, and key management vendors plan their products and efforts.

- **Tokenization for mobile payment:** Some of you are probably asking, "What?" While even basic adoption of mobile payment is still uncommon, dozens of large household names are building mobile and 'smart' payment options. When the players are ready they will come fast and furious, and most firms that consult us *want to embed tokenization into mobile systems for security purposes*. The market is moving too fast for PCI to passively watch this happen.

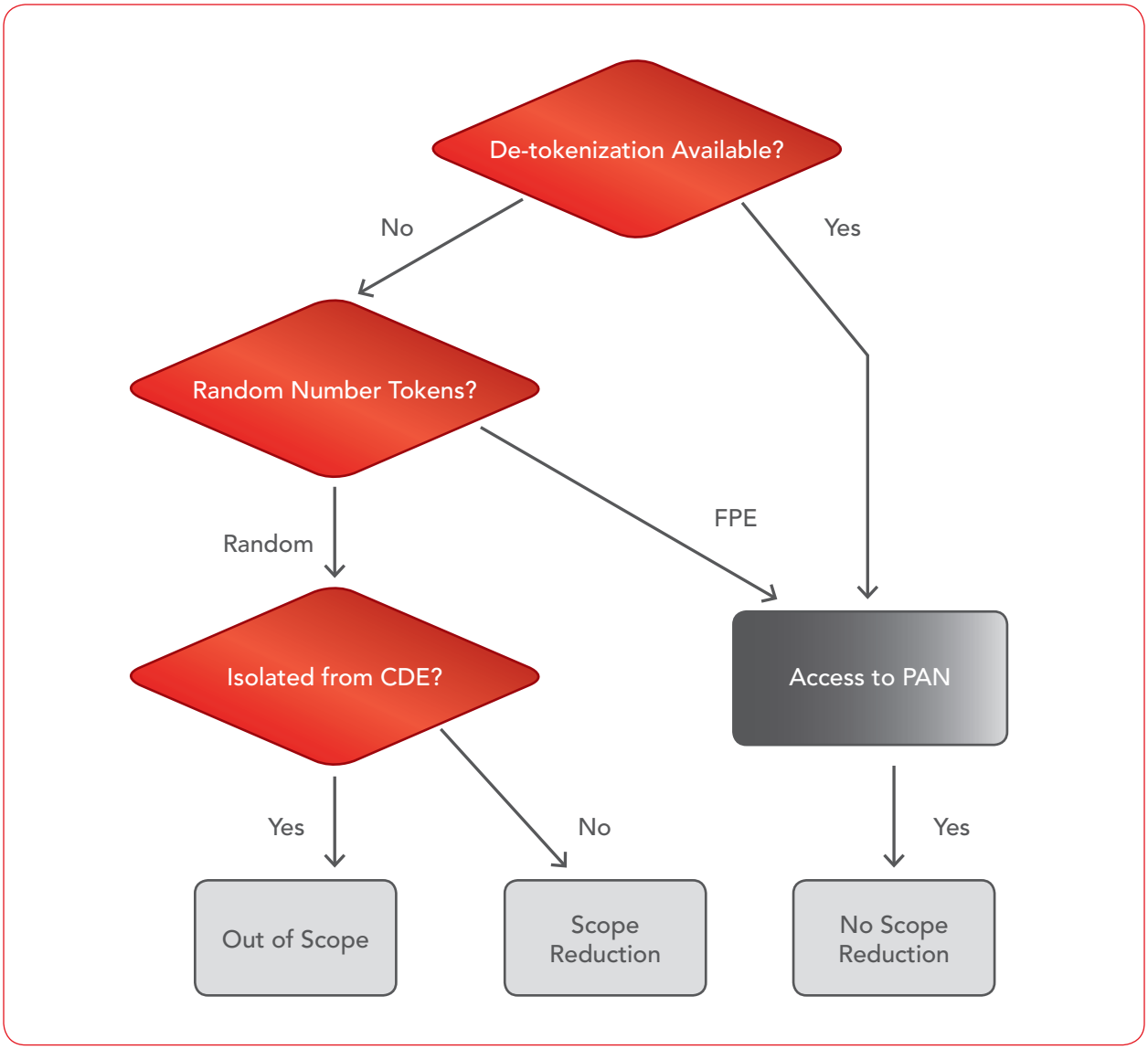
- **Take a stand on encryption:** Encrypted values are not tokens. Rich Mogull wrote an excellent article explaining An Encrypted Value Is **Not** a Token! Encrypted credit card numbers are just that — encrypted credit card numbers. A token is a random surrogate — pulled randomly out of thin air, obtained from a sequence generator, copied from a one-time pad, or even from a code book; these are all acceptable token generation methods. We can debate just how secure encrypted values are until the end of time, but in practice you simply cannot reduce audit scope with encrypted values if the merchant has access to the keys. Practically speaking, companies choose Format Preserving Encryption (FPE) or encrypted PANs because — at some point in time — they need to decrypt and access the original PAN/credit card number. They make this choice because they don't want the expense of altering their applications to use a token server, so they directly access the encryption key manager to decrypt PAN as needed. These systems — per PCI DSS — must be in scope. Unless all PAN data is *unavailable to the merchant* through an 'edge' tokenization solution or *true* POS-to-Processor encryption (P2PE) where the merchant never has PAN data, encrypted tokens are in scope!

- **Address the token identification problem:** PCI calls it 'Token Distinguishability' and section 2.3.4 of the supplement talks about it. In simplest terms, how can you tell if a 'token' really is not a credit card number? We will not cover all the nuances here, but we do want to point out that this is a problem only the PCI Council can address. Merchants, QSAs, and payment providers can't distinguish tokens from credit card numbers with any certainty, especially considering that most large merchants want several digits of each token to reflect the original credit card number. But PCI *requires* 'distinguishability'! Honestly, there is no good solution here given existing PCI guidelines — only the PCI Council can sort this issue out. For now it's better to acknowledge this unpleasant fact and move away from tokens that preserve digits from the original credit card values.

- **Liability and the merchant:** Section 2.4.2 says "The merchant has ultimate responsibility for the tokenization solution." Most merchants buy tokenization as a service, which makes the PCI's position look absurd. And in cases where Point to Point Encryption (P2PE) or 'edge' tokenization is used, it's hard to justify putting the onus on the merchant. This blanket assignment of liability is entirely unjustified, and merchants must work with their third-party providers to specify roles and responsibilities.

The list of issues with the supplement is pretty long, and while we do not want to be completely negative, it is not useful to be fuzzy or dance around the elephant in the room. We all need clear and actionable guidance, and the supplement failed in this regard. The above deficiencies are addressed in the next three sections.

Merchant Guidance



The goal of tokenization is to reduce the scope of PCI security assessment. This statement rubs security people the wrong way, but it's true. The merchant must comply with PCI requirements, and tokenization is designed to reduce the time, cost, and complexity of compliance auditing. That's the motivation for adoption! We want to remove the need to inspect every component for security settings, encryption deployments, network security, and application security, as much as feasible. And whenever possible to remove entire systems from scope. For smaller merchants tokenization can make self-assessment much more manageable. For large merchants paying third-party auditors to verify compliance, the cost savings is huge.

However, PCI DSS still applies. Every system in the logical and physical network associated with the payment transaction systems, de-tokenization, and systems that store credit cards — what the payment industry calls "primary account number," or PAN. For many merchants this includes a major portion — if not an outright majority — of information systems. The PCI documentation refers to these systems as the "Cardholder Data Environment," or CDE. The primary goal of tokenization is to reduce the number of systems encompassed by the CDE. The secondary goal is to reduce the number of relevant checks which must be made. Systems that store tokenized data need fewer checks to ensure compliance with PCI DSS. Even if not fully isolated logically and/or physically from the token server, if they don't have the capability to de-tokenize, they don't risk exposing PAN data.

The Ground Rules

So how do we know when a server is in scope? Let's lay out the ground rules, first for systems that always require a full security analysis:

- **Token server:** The token server is always in scope **if** it resides on premise. If the token server is hosted by a third party, the systems that ask for tokens — and the API's — are subject to inspection per PCI DSS.
- **Credit card/PAN data storage:** Anywhere PAN data is stored, encrypted or not, is in scope. This may be separate from the token server.
- **Tokenization applications:** Any application or system that requests tokenized values in exchange for credit card numbers, is in scope.
- **De-tokenization applications:** Any application platform that can make de-tokenization requests is in scope.

In a nutshell, anything that touches PAN or can request de-tokenized values is in scope. It is assumed that administration of the token server is limited to a single physical location, and not available through remote network services. Also note that PAN data storage is commonly built into token servers, however, some vendors separate storage from the service. If PAN data storage is separate from the token server that provides token generation and de-tokenization services, then both systems are in scope. Always.

Determining System Scope

How can you determine whether tokenization can remove remaining systems from the audit scope? For each **system** ask these questions:

The first thing to check is the capability to make requests to the token server: If it can, it's in scope. The focus here is on de-tokenization, because it is assumed that every other system that has access to the token server or its server API is passing credit card numbers and fully in scope. If this capability exists — through user interface, programmatic interface, or any other means, the PAN is accessible and the system is in scope. It is critical to minimize the number of people and programs that can access the token server or service, both for overall security and to reduce scope.

The second test concerns use of random tokens. Suitable token generation methods include random number generators, sequence generators, one-time pads, and unique code books. Any of these methods can create tokens that cannot be reversed back to credit cards without access to the token server. We're leaving hash-based tokens off

this list because they are relatively insecure (reversible) because providers routinely fail to *salt* hash values or salt with ridiculously guessable values (*i.e.*, the merchant ID).

Vendors and payment security stakeholders are busy debating encrypted card data versus tokenization, so it's worth comparing them again. Format Preserving Encryption (FPE) was designed to secure payment data without breaking applications and databases. Application platforms were programmed to accept credit card numbers, not huge binary strings, so FPE was adopted to improve security with minimum disruption. FPE is entrenched at many large merchants, who don't want the additional expense of moving to tokenization, and so are pushing for acceptance of FPE as a form of tokenization. The supporting encryption and key management systems are accessible — meaning PAN data is available to authorized users, so FPE cannot legitimately remove systems from the audit scope.

Proponents of FPE claim they can segregate the encryption engine and key management from the tokens, therefore it's just as secure as random numbers. This is one of this single greatest points of contention in the security community because we are comparing the impossible to reverse engineer (tokenization) with the highly improbable to reverse (encryption). But the problem is a small mistake in FPE configuration, deployment, programming, network setup, random number generation or some such part of the system and suddenly the improbable is possible. In practice, with FPE deployments for back-office token storage, the key servers are accessible. This nullifies the argument of equivalence. In cases where FPE is used for end-to-end PAN encryption — or what's commonly called P2PE in PCI lingo — FPE is an acceptable solution for the *conveyance* of PAN data as the merchant does not have access to decryption keys. When it comes down to it there are too many moving parts in FPE systems to remove them from scope and they need to be fully vetted, obviating the advantage for use in scope reduction.

Finally, it is important to place systems containing tokenized data outside the Cardholder Data Environment using network segmentation. If they are in the CDE they are in scope for PCI DSS — if only because they provide an attack point for access to other PAN storage, transaction processing, and token servers. You want to limit direct communication between CDE and non-CDE systems in order to limit scope, so configure firewalls, network configuration, and routing to logically separate these areas. Systems that are physically and logically isolated from the CDE, provided they meet the ground rules and use random tokens, are completely removed from audit scope. While the supplement is cloudy on this subject, there is no deviation from the DSS here because you've removed PAN data and you are outside the CDE.

Keep in mind that you will always need at least one on-premise server that maintains tokens to serve as transaction receipts. If you keep tokens in many different locations, it's critical you keep one system as your master reference copy of events, and that the integrity of this server be guaranteed. An accurate record of events is critical for dispute resolution and fraud detection.

Under these conditions tokenization is a big win, but there are additional advantages...

Determining Control Scope

As mentioned above, a fully isolated system with random tokens can be removed from the audit scope. Consider those platforms which have historically stored credit card data but do not **need** it: customer service databases, shipping & receiving, order entry, etc. This is where you can take advantage of tokenization. For all systems which can be removed from audit scope, you can save effort on: identity management (and review), encryption, key

management, patching, configuration assessment, etc. Security services such as anti-virus, monitoring, and auditing are no longer mandatory on those systems for PCI compliance. This saves time, reduces licensing costs, and cuts audit expenses. You still need basic platform security to ensure tokens are not swapped, replaced, or altered — so monitoring, hash verification, and/or audit trails are recommended. But in this model the token is purely a reference to a *transaction* rather than PAN, so it's much less sensitive and the danger of token theft is reduced or removed entirely.

This raises the question: can we further reduce scope on systems that use tokenization but are not isolated from credit card processing? The short answer is 'yes.' Systems that are not fully isolated but only use random number tokens don't require full checks on encryption or key management (you did choose real tokens, right?). Further, you don't need to monitor access to tokens, and enforce separation of duties as rigorously. Yes, the PCI guidance on tokenization states that the full scope of PCI DSS applies, but these controls are ridiculous in the absence of cardholder data — as is the case with random number tokens, but not with FPE. Our research indicates that the PCI Council waffled on their tokenization scoping guidance, to avoid political infighting and limit their liability. But this additional scope reduction make a **huge** difference to merchants for cost reduction — and tokenization is for merchants, not the PCI Council. Merchants, as a rule, don't really care about other organizations' risk — only about maximizing their own savings while meeting requirements.

Remember that other controls are still required. Access control, anti-virus, firewalls, and the like remain on your checklist. And you need to verify network segmentation and that de-tokenization interfaces are not accessible outside tightly controlled circumstances. And you still need to monitor system and network usage for anomalous activity. Everything else remains as it was — especially if it is in scope under the ground rules above. This means systems that perform de-tokenization requests (whether they store PAN or not) must completely satisfy the PCI DSS standard. When reading the PCI's tokenization supplement, keep in mind that tokenization does not modify the standard — instead it has the potential to render some required checks irrelevant and/or remove systems from the assessment scope.

One final comment: Notice that this decision tree does not distinguish between in-house tokenization platforms and third-party tokenization services. If you use a third-party tokenization service the token server is automatically off-premise, further reducing audit complexity. Onsite storage repositories, even using encryption, complicate the assessment process as every facet of the token system needs to be fully vetted. Onsite token servers provide dozens of cost and performance benefits, but the best way to get as much as possible out of scope is to use a third-party service. The PCI Council claims you are ultimately responsible for token services in Requirement 12, but you can still place much of the burden of proof on the service provider.

Between scope and control reduction, a 50% reduction in compliance costs can really happen.

Audit Advice

In this portion of our tokenization guidance we offer advice to auditors. This section is applicable to both internal auditors going through a self-assessment questionnaire, as well as external auditors (QSAs) validating adherence to PCI requirements. For the most part auditors will continue to follow PCI DSS for the systems that process credit card information just as they always have. But we will discuss how tokenization alters the environment and how to adjust the investigation in those areas where tokenization systems supplant PAN processing. At the end of this paper, we will work through the PCI DSS requirements section by section and discuss specifics, but here we limit our comments to a high-level overview.

So what does the auditor need to know? How does tokenization change discovery processes? We have already set the "ground rules" for which types of systems are audited as before: anywhere PAN data is stored, applications that make tokenization or de-tokenization requests, and all on-premise token servers require thorough analysis. For those systems here is what to focus on:

- **Interfaces & APIs:** At the integration points (APIs and web interfaces) for tokenization and de-tokenization, you need to review security and patch management — regardless of whether the server is in-house or hosted by a third party. The token server vendor should provide details of which libraries are installed and how the systems integrate with authentication services. But not every vendor is great about producing documentation — especially proper use from a security standpoint — so ask for this data if they fail to provide it. And merchants need to document all applications that communicate with the token server. This encompasses **all** communication, including token-for-PAN transactions, de-tokenization requests, and administrative functions.

- **Tokens:** You need to know what kind of tokens are in use — each type carries different risks. Hashed values and encrypted values each present risks. Random numbers, number sequences, one time pads and code books present slightly different — albeit reduced — risks.

- **Token Storage Locations:** You need to be aware of where tokens are stored, and merchants need to designate at least one storage location as the 'master' record repository to validate token authenticity. In an on-premise solution this is the token server; third-party tokenization providers need to keep accurate records within their environment for dispute resolution. The system needs to comply fully with PCI DSS to ensure tokens are not tampered with or swapped.

- **PAN Migration:** When a tokenization service or server is deployed for the first time, the *existing* PAN data must be removed from where it is stored and replaced with tokens. This can be a difficult process for the merchant and *may not be 100% successful!* You need to know what the PAN-to-token migration process was, and review the audit logs to find any issues during the replacement process. If you have the capability to distinguish between tokens and real PAN data, audit some of the tokens as a sanity check. If the merchant hired a third party firm — or the vendor — then the service provider will supply the migration report.

- **Authentication:** This is **critical**; any attacker is likely to target the authentication service, as it is a critical gateway for de-tokenization requests. As with the 'Interfaces' point above: pay careful attention to separation of duties,

least privilege principle, and limiting the number of *applications* that can request de-tokenization.

- **Audit Data:** Make sure that the token server, as well as any API or application that performs tokenization/de-tokenization, complies with PCI Requirement 10. This is covered under PCI DSS but these log files become a central part of your daily review so this is worth repeating here.
- **Deployment & Architecture:** If the token server is in-house — or managed on-site by a third party — you need to review the deployment and system architecture. Understand what happens in the environment if the token server goes down, and how token data is synchronized across multi-site installations. Weaknesses in the communications, synchronization, and recovery processes are all areas of concern; the merchant and/or vendors must document these facilities and the auditor needs to review them.
- **Token Server Key Management:** If the token server is in-house — or managed on-site by a third party — you need to review key management facilities because every token server encrypts PAN data. Some solutions offer embedded key management while others use external services, but you need to ensure it meets PCI DSS requirements.

For non-tokenization usage and systems that store tokens but do not communicate with the token server, auditors need to conduct basic checks to ensure the business logic does not allow tokens to be used as currency. Don't use tokens to initiate financial transactions! Make certain that tokens are merely placeholders or surrogates, and don't function as credit card numbers internally. Make sure tokens don't initiate business processes or act as currency themselves. Repayment scenarios, chargebacks, and other monetary adjustments are all good places to check. The token is to be used for transactional reference — not currency or as a credit proxy. These uses lead to fraud, and in the event of a compromised system might be used to initiate fraudulent payments without credit card numbers.

The depth of these checks varies — merchants filling out self-assessment questionnaires tend to be more liberal in interpreting the standard than top-tier merchants who have external auditors combing through their systems. But these points are the focus for either group.

Tokenization Audit Checklist

This is our checklist for how PCI requirements apply to systems which use tokenization:

PCI Requirement	Recommendation
1.2 Firewall configuration	Token server should restrict all IP traffic both to and from systems specified under the 'ground rules', specifically: • Payment processor • PAN storage server (if separate) • Systems that request tokens • Systems that request de-tokenization This is no different than PCI requirements for the CDE, but it's recommended that these systems communicate only with each other. If the token server is on site, Internet and DMZ access should be limited to communication with the payment processor. Finally, remember to 'air-gap' systems with tokenized data to remove them from scope.
1.2.2 Network diagram	You will need to update your network and data flow diagrams to include token servers and systems that make de-tokenization requests.
2.1 Defaults	Implementation for most of requirement 2 will be identical, but section 2.1 is most critical in that there should be no 'default' accounts or passwords for the tokenization server. This is especially true for systems, which are remotely managed or have remote customer care options. All PAN security hinges on effective identity and access management, so establishing unique accounts with strong passphrases is essential. For on-site systems it is critical that the token server(s) run on dedicated systems.
2.2.1 Single function servers	2.2.1 bears mention both for security and to avoid vendor lock-in. For security, consider an on-premise token server as a standalone function, separate and distinct from applications which make tokenization and de-tokenization requests. To avoid vendor lock-in, make sure the token service API calls or vendor supplied libraries used by your credit card processing applications are sufficiently abstracted to facilitate switching token services without significant modifications to your PAN processing applications.

PCI Requirement	Recommendation
2.3 Encrypted communication	You will want to encrypt non-console administration per the specification, but also all API calls to the token service. When using multiple token servers to support failover, scalability, or multiple locations, be sure that all synchronization occurs over encrypted channels — preferably a dedicated VPN with mutual authentication.
3.1 Minimize PAN storage	The beauty of tokenization is that it's the most effective solution available for minimizing PAN storage. By removing credit card numbers from every system other than the central token server, you reduce the scope of your PCI audit. Look to tokenize or remove every piece of cardholder data you can, keeping it *all* in the token server. This addresses business, legal, and regulatory requirements, with a substantial improvement to real security.
3.2 Authentication data	Tokenization does not circumvent this requirement; you must still remove sensitive authentication data per subsections 3.2.X.
3.3 Masks	Technically you are allowed to preserve the first six (6) digits and the last four (4) digits of the PAN. However, we recommend that you examine your business processing requirements to determine if you can fully tokenize the PAN, or at minimum only preserve the last four digits for customer verification. The number of possible tokens you can generate with the remaining six digits is too small for many merchants to generate quality random numbers. Please refer to ongoing public discussions on this topic for more information. When using a token service from your payment processor, ask for single-use tokens to avoid cross-vendor fraud. Issuing a new token for each transaction increases overhead, so your vendor may not provide this as a default option.

PCI Requirement	Recommendation
3.4 Render PAN unreadable	A principal benefit of tokenization is that it renders the PAN unreadable. Tokenization environments require two particular auditing changes. Verify that PAN data is actually swapped for tokens in all systems. For on-premise token servers, verify that the token server internally encrypts PAN, or offers an equivalent form of protection such as not storing PAN data*. We recommend that you use sensitive number finder or discovery tools to ensure all PAN information has been substituted for tokens. Regarding token type, we do not recommend hashing because it offers poor PAN data protection. Many vendors store hashed PAN values in the token database to speed token lookup, but this is a poor security choice. Our recommendation is to encrypt PAN data; if it is stored in a database we recommend table, column, or row level encryption within the token database. Use of full database or file layer encryption can be highly secure, but most such solutions offer no failsafe protection when database or token administrator credentials are compromised. We understand our recommendations are unusual, but experience has taught us to err on the side of caution for PAN storage. **Some solutions offer one-time pad and codebook options that don't require PAN storage.**
3.5 Key management	Token servers encrypt the PAN data stored internally, so you need to verify the supporting key management system as best you can. Some token servers offer embedded key management, while others are designed to leverage existing key management services. Very few people can adequately evaluate key management systems to ensure they are really secure, but you can confirm that the vendor is using industry standard components or has validated their implementation with a third party. Just be sure they are not storing the keys in the token database unencrypted. It happens.
3.6 Document procedures	Document the token server's use of key management facilities. If it's not documented it's not in place, as far as PCI is concerned.
4.1 Strong crypto for Internet communications	As with requirement 2.3, when using multiple token servers to support failover, scalability and/or multi-region support, ensure that all synchronization occurs over encrypted channels — preferably a dedicated VPN with mutual authentication.

PCI Requirement	Recommendation
6.3 Secure development	On-site and third-party token servers both introduce new libraries and API calls into your environment. It is critical that your development process includes verification that what you put into production is secure. You can't take the vendor's word for it – you need to verify that all defaults, debugging code, and API calls are secured. Tokenization changes release management, and you will need to update 'pre-flight' checklists to accommodate it. This includes patching, Anti-virus, and web application firewall settings to protect the entire CDE.
6.5 Secure code	With 'home-grown' tokenization applications, all of section 6.5 applies. When using a third-party product, ensure that your credit card processing applications correctly integrate with – and validate – third-party libraries and API calls. Ensure suitable abstraction between your code and what the vendor provides to avoid lock-in and painful migration, should you need to switch providers in a hurry. Make sure you have reasonable testing procedures to detect SQL injection, memory injection, and error handling issues commonly used to subvert systems.
7. Restrict access	Requirement 7, along with all its sub-requirements, applies fully with tokenization. While tokenization does not modify this requirement, we recommend you pay special attention to separation of duties around the three token server roles (admin, tokenization requestor, and de-tokenization requestor). We also stress that the token server security model hinges on access control for data protection and Separation of Duties (SoD), so spend extra time on this requirement. System administration and access to the token vault are two separate roles. Different people, each with suitably isolated roles, must perform these tasks.
8. Uniquely identify users	Hand in hand with requirement 7, you need to ensure you can uniquely identify each user – even when using generic service accounts. And if you have third-party support for tokenization interfaces or token server administration, make sure these administrators are uniquely identified. Your vendor should provide this information on request.
9. Physical security	You need to ensure the physical safety of the token server, as a disgruntled employee walking out with the server would be disastrous. Backup media for the token server must conform to physical security requirements, as before.

PCI Requirement	Recommendation
10.2 Monitoring	Monitoring is a critical aspect of token and token server security, so whether your token server is managed internally or by a third party, onsite or offsite, you need to log all – and we mean *all* – administrative actions and de-tokenization requests. These log entries should be as detailed as possible without leaking sensitive data. Most likely you will log each tokenization request as well, as these operations typically map one-to-one against payment processing transactions and should be available for forensic audits and dispute resolution. But normal log entries don't require the same degree of detail as administrative activity. Create symmetric logs for client applications as well as those on the token server, to enable cross-referencing and validation of events. This is especially important if the token server is provided as a service. Finally, include the token server and tokenization applications in your File Activity Monitoring scans.
10.5 Log data security	Some token servers provide options for secure log creation — including time stamps, transaction sequences, and signed entries to prevent tampering. Non-repudiation features like this are ideal, and we recommend them as an easier way to demonstrate secure logs. But transaction volume is often too great to use these features, so you may need to lock down access to the log files and/or stream them to secure log management facilities to ensure they are not tampered with.
11. Testing	It is not entirely clear what testing should be conducted to validate a token server. Tokenization for payment is a relatively new use case, and there is no such thing as a 'common' attack; nor is there a list of common vulnerabilities. This is an area where we hoped for more guidance from the PCI Council. To fill this void, our recommendations — over and above the basic requirements — are as follows: • The interfaces to the system are simple, and only support a handful of functions, but you should conduct a penetration test against the system and interfaces. • Deployment of the server can be complex — with several encryption, storage, and management functions that need to be tested. Focus your tests on how these services communicate with each other. • Test how different components handle service interruptions. • Test system reliance on DNS and whether the token server can be fooled or subverted through DNS poisoning.

PCI Requirement	Recommendation
12. Policies	Tokenization fundamentally changes procedures and business workflow. You will need to update some information security and operational policies to reflect these changes. Further, you will want to augment section 12.6 with tokenization awareness training —especially for personnel who handle PAN's for re-payment or in customer service situations.
12.8 Service providers	When using a token server through a payment processor or other third party, you will be required to follow the requirements for verifying your provider's compliance with PCI DSS. This is not a deviation from the requirements, more of a reminder that work needs to be completed prior to an assessment. The good news is the provider may help with this work.
Appendix A. Hosting Providers	Validating service providers is different with tokenization, and you will have a lot of work to do here. The good news is that your provider – likely a payment processor – offers tokenization to many different merchants and understands their obligations in this area. They have information concerning the compliance of their servers and services, along with their personnel policies. It will be up to you to dig a little deeper to find out what type of tokens they use, whether the tokens are stored in a multi-tenant environment, how they secure payment data, etc. Risk and responsibility is a grey area. The PCI Council's token guidance supplement says you are ultimately responsible for the security of PAN data, but that's not a reasonable position if you outsourced all PAN processing, and only store tokens, in order to shift much of the risk to the provider. After all, part of the value you receive when paying for a tokenization service is the transfer of risk and reduction of audit requirements. Still, you are responsible for due diligence in understanding your payment processor's security measures and documenting their controls so they are available for review.

Conclusion

The vast majority of the recommendations in this guidance document are in line with the standard PCI DSS requirements. While there are a handful of places where this document contradicts the PCI supplement on tokenization, that is due to contradictory information in the supplement as opposed to our interpretation of PCI DSS. The remaining differences are areas where the supplement was lacking answers to common questions. Given these differences we have done our best to vet this information with Qualified Security Assessor (QSAs) and feel that the recommendations will hold up to scrutiny. It is not unusual to receive guidance on how to comply with PCI requirements from organizations other than the PCI Council, task forces, and/or steering committees. There are many different customer environments, use cases, and issues around PCI compliance that don't fit any cookie-cutter mold. What is unusual is that the PCI Council totally fumbled in providing guidance for a technology as simple as tokenization. Given that tokenization offers better security, simplifies data storage security, and makes audits easier, it's baffling that the Council failed to offer clear advice that would encourage adoption. Discovering what's missing from the official guidelines — usually only discovered *after* purchasing a product and coming to grips with how it *really* works — is no recipe for success.

We hope you find our research helpful in reducing your compliance scope and cutting costs. We deliberately skipped areas which are not relevant to common use cases, in order to keep this paper short. So we recommend engaging your QSAs to hash out the grey areas of compliance. If you are filling out self-assessment questionnaires, document the ambiguities and explain your choices. And remember: when in doubt about which way to set up a particular system or service, choose the most secure option — you can't go wrong with that approach. Good luck with your selection process — we are happy to help if you run into questions during your evaluation. Feel free to drop us a note at info@securosis.com, and we'll do our best to help out.

About the Analyst

Adrian Lane, Analyst and CTO

Adrian Lane is a Senior Security Strategist with 24 years of industry experience. He brings over a decade of C-level executive experience to the Securosis team. Mr. Lane specializes in database architecture and data security. With extensive experience as a member of the vendor community (including positions at Ingres and Oracle), in addition to time as an IT customer in the CIO role, Adrian brings a business-oriented perspective to security implementations. Prior to joining Securosis, Adrian was CTO at database security firm IPLocks, Vice President of Engineering at Touchpoint, and CTO of the secure payment and digital rights management firm Transactor/Brodia. Adrian also blogs for Dark Reading and is a regular contributor to Information Security Magazine. Mr. Lane is a Computer Science graduate of the University of California at Berkeley with post-graduate work in operating systems at Stanford University.

Author's Note

The content in this report was developed independently of any sponsors. It is based on material originally posted on the Securosis blog www.securosis.com, but has been enhanced, reviewed, and professionally edited. And just in case we've annoyed someone in this lawsuit happy world — we are making several controversial statements in this paper — it should be noted that the views expressed here don't necessarily reflect the views of our sponsors.

Special thanks to Chris Pepper for editing and content support.

About Securosis

Securosis, L.L.C. is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services.

Our services include:

- The Securosis Nexus: The Nexus is an online environment to help you get your job done better and faster. It provides pragmatic research on security topics, telling you exactly what you need to know, backed with industry-leading expert advice to answer your questions. The Nexus was designed to be fast and easy to use, and to get you the information you need as quickly as possible. Access it at www.nexus.securosis.com.
- Primary research publishing: We currently release the vast majority of our research for free through our blog, and archive it in our Research Library. Most of these research documents can be sponsored for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements and conform with our Totally Transparent Research policy.
- Research products and strategic advisory services for end users: Securosis will be introducing a line of research products and inquiry-based subscription services designed to assist end user organizations in accelerating project and program success. Additional advisory projects are also available, including product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessment.
- Retainer services for vendors: Although we will accept briefings from anyone, some vendors opt for a tighter, ongoing relationship. We offer a number of flexible retainer packages. Services available as part of a retainer package include market and product analysis and strategy, technology guidance, product evaluation, and merger and acquisition assessment. We maintain our strict objectivity and confidentiality. More information on our retainer services (PDF) is available.
- External speaking and editorial: Securosis analysts frequently speak at industry events, give online presentations, and write and/or speak for a variety of publications and media.
- Other expert services: Securosis analysts are available for other services as well, including Strategic Advisory Days, Strategy Consulting Engagements, and Investor Services. These tend to be customized to meet a client's particular requirements.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors.

Additionally, Securosis partners with security testing labs to provide unique product evaluations that combine in-depth technical analysis with high-level product, architecture, and market analysis. For more information about Securosis, visit our website: www.securosis.com.

About Protegrity

Protegrity is the leading global security software company providing high performance, infinitely scalable, end-to-end data security solutions. Protegrity customers centrally develop, manage and control data security policy that protects sensitive information across the enterprise in databases, applications and file systems from the point of acquisition to deletion. Protegrity's solutions give corporations the ability to implement a variety of data protection methods, including tokenization, strong encryption, masking and monitoring to ensure the protection of their sensitive data and enable compliance for PCI-DSS, HIPAA and other data security requirements.

Protegrity's award winning software products and innovative technology are backed by 14 industry patents, all of which differentiate the Protegrity Data Security Platform from point solutions. Protegrity employees are security technology specialists with deep industry expertise in data security approaches and techniques, including key management, forensics reporting and security policy definition and management. Protegrity has more than 200 enterprise customers worldwide who use its comprehensive data security solutions to fulfill their obligations to protect their sensitive data, brand, and business reputation. Protegrity is headquartered in Stamford, Connecticut USA with offices in the United Kingdom and Germany.

For more information
Telephone: 203.326.7200
Email: info@protegrity.com
www.protegrity.com

Contributors

The following individuals contributed significantly to this report through comments on the Securosis blog and follow-on review and conversations:

Walt Conway
'Brian'
Steve Sommers
Stephen Tihor