Tokens and tokenization

Tokenization is defined as any form of format-preserving data protection. OpenText Data Privacy and Protection Enterprise (Voltage) is a highly scalable enterprise-class solution that offers many tokenization methods, including reversible and irreversible cryptographic, and reversible non-cryptographic approaches, to protect any data type in any language.

Attributes of tokens

- Tokenization systems map each data element to a unique value
- Tokens may share characteristics with the original data elements, such as character set and length
- Tokenization systems are deterministic, such that repeatedly generating a token for a given value yields the same token
- Tokenized data can be searched by tokenizing the queried data elements and searching for those

Tokenization is a process by which PANs (primary account numbers), PHI (protected health information), PII (personally identifiable information), and other sensitive data elements are securely replaced by surrogate values referred to as **tokens**. Tokenization is a form of encryption, but the two terms are frequently used in distinct ways. Encryption is commonly meant as the encoding of human- readable data into incomprehensible text that can only be decoded with the right decryption key, while tokenization (or “masking” or “obfuscation”) is commonly meant as some form of **format-preserving data protection**; that is, the conversion of sensitive data elements into non-sensitive, replacement values—tokens—that are the same length and format as the original data elements.

Tokenization history

Digital tokenization was created by TrustCommerce in 2001 to help a client protect customer credit card information. Merchants were storing PANs on their own servers, which meant that anyone who had access to their servers could potentially view those customer credit card numbers.

TrustCommerce developed a system that replaced the PANs with a randomized number correlated with the PAN through a database. This allowed merchants to store and reference these tokens, instead of the PANs, when accepting payments. When processing the payments, TrustCommerce converted the tokens back to the original PANs. This isolated the risk to TrustCommerce, since the merchants no longer stored PANs in their systems.

As security concerns and regulatory require-ments grew, such first-generation solutions proved the technology's value and other vendors offered similar solutions. However, as discussed in the next section, problems with this approach soon became clear.

What are the types of tokenization?

There are two types of tokenization: **reversible** and **irreversible**.

Reversible tokenization means a process exists to convert the tokens back to their original values. In privacy terminology, data protection via a reversible process is called pseudonymization. Such tokens may be further subdivided into cryptographic and non-cryptographic, although this distinction is artificial, since tokenization really is a form of encryption.

- **Cryptographic** tokenization generates tokens using strong cryptography; the cleartext data elements are not stored anywhere—just the cryptographic key.

  NIST-standard FF1-mode AES, pioneered by Voltage Security (now OpenText) in the early 2000s, is an example of cryptographic tokenization.

  OpenText™ Data Privacy and Protection Enterprise (Voltage) ***Format-Preserving Encryption (FPE) and Embedded Format-Preserving Encryption*** (eFPE) are cryptographic tokenization methods.

- **Non-cryptographic** tokenization originally meant that tokens were created by randomly generating a value and storing the cleartext and corresponding token in a database, like the first-generation TrustCommerce offering. This database or vaulted approach is conceptually simple, but means that any tokenization or detokenization request must make a server request, adding overhead, complexity, and risk. It also does not scale well. Consider a request to tokenize a PAN. The server must first perform a database lookup to see if it already has a token for that PAN. If it does, it returns that token. If it doesn't, it must generate a new token, then do another database lookup to make sure that token has not already been assigned to a different PAN. If it has, it must generate another token, check that one, and so forth. As the number of tokens created grows, the time required for these database lookups increases and, worse, the likelihood of such collisions grows exponentially. Such implementations also typically use multiple token servers, for load-balancing, reliability, and failover, and these must perform real-time database synchronization to ensure reliability and consistency, adding further complexity and overhead.

The terms "tokenization," "masking," and "obfuscation" are all often used generically to mean any form of format-preserving data protection.

OpenText Data Privacy and Protection Enterprise (Voltage) offers reversible and irreversible cryptographic, and reversible non-cryptographic tokenization methods.

Current non-cryptographic tokenization focuses on a *"stateless"* or approach, using randomly generated metadata that is securely combined to build tokens. Such systems can operate disconnected from each other, and infinitely scale since they require no synchronization beyond replication of the original metadata.

OpenText SecureData Enterprise's ***Secure Stateless Tokenization*** (SST) is a non-cryptographic tokenization method, offering optimal performance, security, and scalability.

Despite the fact that United States Export Administration Regulations consider both approaches to be cryptography, the Payment Card Industry (PCI) Data Security Standard (DSS) continues to distinguish the two. When the PCI DSS was released in 2008, it required merchants to "Render a [credit card Primary Account Number, or PAN], at minimum, unreadable anywhere it is stored…[using] index tokens and pads…[or] strong cryptography." Index tokens and pads are what people usually call tokenization, i.e., replacing a value with a randomly generated value. More current versions of PCI DSS provide more detail:

"An index token is a cryptographic token that replaces the PAN based on a given index for an unpredictable value. A one-time pad is a system in which a randomly generated private key is used only once to encrypt a message that is then decrypted using a matching one-time pad and key."

Analysis of this reveals what any cryptographer will tell you: tokenization ***is*** a form of encryption. Index tokens are a one-time pad: that's the "unpredictable value." When this is implemented using first-generation database tokenization, a random number generator is used to create that "unpredictable value."

It is critical to realize that, in this scenario, the database ***is an encryption key***. Just because it is not 128 bits or 256 bits or some other common size does not mean it is not an encryption key. Given an algorithm and some secret information (in this case, the database), a cleartext PAN can be consistently converted to a token and vice versa. And that is precisely the definition of encryption. Nevertheless, some PCI QSAs (qualified security assessors) still prefer "non-cryptographic" tokens over cryptographic tokens when performing PCI DSS assessments.

Irreversible tokenization means the tokens cannot be converted back to their original values. In privacy terminology, data protection via an irreversible process is called *anonymization*. Such tokens are created through a one-way function that anonymizes the data elements for various use cases, such as third-party analytics, and replacing production data for test and development in less secure environments.

OpenText Data Privacy and Protection Enterprise's ***Format-Preserving Hash*** (FPH) is an irreversible cryptographic tokenization method.

Sensitive data type	Sensitive data	Tokenized data
Credit card number	1111-2222-3333-4444	1111-2287-9581-4444
US Social Security number	**999-88-7654**	740-36-7654
Address	1234 Maple Street	7321 Uqhap Fbzira
Phone number	415-555-1234	415-555-1234

Resources

OpenText Data Privacy and Protection Foundation

Learn more ›

Tokenization vs. traditional encryption

Traditional encryption products and services are frequently based on a non-format- preserving AES mode, and are generally used to protect unstructured or semi- structured data, such as whole files. This is because they generate tokens with a different format from the inputs:

- Different characteristics from the original data elements, such as character set and length
- Significantly larger than the original data elements
- May be probabilistically created, such that repeatedly encrypting a given value yields different outputs

OpenText Data Privacy and Protection Enterprise's ***Identity-Based Symmetric Encryption*** (IBSE) and asymmetric ***Identity-Based Encryption*** (IBE) are traditional encryption methods.

A common use of traditional encryption for structured data is to protect it at the disk level. This is appealing because it is transparent to users and applications: data is automatically encrypted when written to disk and automatically decrypted when accessed. However, this approach does not protect the data against modern attack vectors, such as malware, insider threats, and phishing approaches, that take advantage of that automatic decryption "feature." All it really protects against is access attempts following physical theft of the disk drives. For example, laptops and mobile devices often use whole- disk encryption technologies, such as Windows BitLocker, Apple FileVault, or Symantec Endpoint Encryption, because these devices are prone to getting lost or stolen.

Protecting data at the field level with one of OpenText Data Privacy and Protection Enterprise's tokenization methods removes any requirement for "transparent" or "whole disk" encryption approaches, as the data is already secure before it is written to the disk and it will only be accessible to authorized users.

What are the OpenText differentiators?

OpenText Data Privacy and Protection Enterprise is a highly scalable, highly performant enterprise-class solution that offers a variety of tokenization methods, including reversible and irreversible cryptographic, and reversible non-crypto- graphic approaches, to protect any data type in any language. Further, it's stateless technologies avoid the imposition of operational and managerial burdens from key and vault management requirements in competing solutions through the use of OpenText innovations that avoid the storage of encryption keys and token tables.

opentext™