outpayce from amadeus

PAYMENTS SECURITY
CARD DATA TOKENIZATION

1. Description

1.1 This solution replaces payment cards data sensitive information with non-sensitive data (tokens), and vice-versa.

1.2 We store the sensitive data on behalf of Customer. Customer stores non-sensitive data without compromising security.

1.3 This solution leverages on our NOX engine, a patented internal application PCI DSS certified, that stores payment cards data with a prominent level of security.

1.4 Security measures of our NOX engine include encrypted database, special hardware for encryption, secure keys, special firewall, limited and monitored access.

1.5 Our NOX engine is based in the principles of separation and isolation. Our NOX engine and Customer's application do not yield the complete payment card number. Therefore, each database taken separately is worthless as no database contains real card numbers. Furthermore, no communication between our NOX engine and Customer's application contains any real card number.

2. How does this solution work?

2.1 Tokenization process.

2.1.1 Customer's application that needs to store a PAN (Primary Account Number) calls the solution.

2.1.2 This solution splits the payment card number into two (2) parts: the apparent part (usually the first 6 digits and 4 last digits) and the concealed part (the middle part).

2.1.3 This solution generates a token in real time and sends the concealed part of the payment card to our NOX engine. Then, a hash is computed on the whole payment card number.

2.1.4 Our NOX engine assigns a unique identifier (NOX ID) for the information received. If the payment card has already been stored in our NOX engine, the NOX ID is reused.

2.1.5 Our NOX engine replies with the NOX ID.

2.1.6 This solution stores the NOX ID, the apparent part of the payment card number (usually the first 6 digits and 4 last digits) and the token.

2.1.7 This solution transmits the token to Customer's application.

2.2 Detokenization process.

2.2.1 Customer's authorized application that needs to detokenize calls the solution.

2.2.2 This solution retrieves the NOX ID associated to the token and the apparent part of the payment card number (usually the first 6 digits and 4 last digits).

2.2.3 The tokenization module sends the NOX ID to NOX.

2.2.4 NOX retrieves the concealed digits from the database.

2.2.5 NOX replies to the tokenizer with the concealed digits.

2.2.6 This solution returns the full payment card number (PAN) to Customer.

3. Key features

3.1 This solution is PCI DSS certified / compliant.

3.2 This solution can tokenize payment cards with a structure of 12 to 19 digits.

3.3 Tokens are created by our NOX engine.

3.4 We create a unique token for each payment card, which is an alphanumeric string of the same length as the payment card number.

3.5 In general, the first 6 digits and the last 4 digits of the token are the same as in the payment card number.

3.6 The middle part of the token contains at least one letter, which allows differentiating a token from a payment card number and cannot be used to process payments transactions.

3.7 This solution generates tokens that are limited to eight (8) calendar years since their creation.,

3.8 CCVs (Card Security Codes) cannot be stored for extended periods.

3.9 This solution generates CVV tokens when the CCV is a numeric string of 3 or 4 numbers (same size as the CCV token), with a lifetime of ten (10) minutes to serve the card authorization purpose.

3.10 Upon expiration of this period, we purge these tokens from our databases. The consumer must enter their CVV every time they make a purchase.

3.11 The tokenization service is available through XML web service. Alternatively, there is a graphical user interface available, which allows employees of the Customer to enter card numbers on a web page manually. Once Customer's Personnel clicks on 'tokenize', the payment card number is sent for tokenization, and the generated token is displayed in the 'token value' field. A specific limit can be enforced for the number of detokenization performed on the graphical user interface or by a specific agent to avoid fraud.

3.12 Customer can send the payment card data to the solution in clear or encrypted form.

3.13 If Customer wants to use encryption, Customer must call our encryption configuration service.

3.14 If Customer calls this flow, Customer's systems call our solution with the encrypted payment card PAN and/or encrypted CVV.

3.15 This solution will decrypt the message, tokenize the PAN and/or CVV and return a payment card token and/or CVV token to Customer.

3.16 Tokenization operations are restricted to fully identified Customer's Personnel. Customer's Personnel must have LSS (Local Security Settings) ACL (Access Control Lists) access. Customer must define user accounts and set permissions and roles. The security is handled by LSS with user authentication. Our tokens are associated with the Customer with the rights on it.

3.17 Tokenization and detokenization requests are different transactions accounted for individually. For clarification purposes, the following are examples of basic tokenization and detokenization transactions:

3.17.1 Storage of payment card data: tokenization transaction.

3.17.2 Form of payment creation: detokenization transaction.

3.17.3 Payment authorization request: detokenization transaction.

3.18 Depending on the payment flow implemented by Customer, additional tokenization and detokenization transactions may apply (for example, authentication, fraud screening and card capture would be detokenization transactions).

3.19 Customer sees the number of monthly tokenization and detokenization transactions through the relevant invoices.

4. Disclaimers and limitations

4.1 This solution does not tokenize or detokenize fidelity numbers or bank account numbers.

4.2 The tokens we generate are not network tokens. We do not need any issuer agreement to create such tokens.

4.3 We renew the lifetime of the token with each usage. Nonetheless, if the tokens are not used and their lifespan expires, these tokens cannot be detokenized or used to process payments transactions.

5. Integrations

5.1 This solution can be integrated with our Checkout SDK and Checkout UI solutions as well as with the PNR.

5.2 Other integrations with Customer's front and back ends may be available, subject to further scoping.