Tokenisation and Encryption in Digital Payments, FinTech

Feb, 2022

Tokenisation and Encryption

(Digital payments / Fintech)

① Tokenisation in Digital payments

(i) Tokenisation is the process of protecting sensitive data by replacing it with an algorithmically generated number called token.

(ii) Token will be diff's for each device.

(iii) Card credentials are secured and stored inside a vault, outside the device.

(iv) For Every transaction, the token must be coupled with a cryptogram.

② How it works in payment

① Card holder use a device that must support the digital wallet application and registers a card.

② send data to token requestor (TR). (Card details)

↓

③ Token server provide verify the data. (TSP)

(2)

④ TSP verify the issuer with Card issuer. (3)

⑤ TSP register the PAN and links it to a new token in a Secure database (Token Vault)

⑥ TR identifier, token Expire date, restrictions on the use of the new token on Certain channels, use by particular merchant, limitation on the number of permitted uses, and verification of the cryptogram.

⑦ TSP notifies the application of the newly generated token.

① Apple generated token as :-

- Device account Number (DAN)

② SAMSUNG

Digitized PAN (DPAN) in Secure ~~location~~ (Hardware) Element (SE) or Host Card Emulation (HCE)

It store all the info and it could be anywhere like mobile, pc.

(3) There could be two formats of token

(1) Format preserving tokens (16-digit)
* only numbers *

Purpose :- (1) If someone is a hacker, they will not be diff. between, is a card number or Random generated number.

(2) It could be designed of the system as well

(2) Non-Format preserving tokens

(1) It could be any thing without any restrictions (16, 12, 20-digit ...)

(4) How to add Card for Token

(1) manual entry → directly (Card, photo ..) add → Capture all details

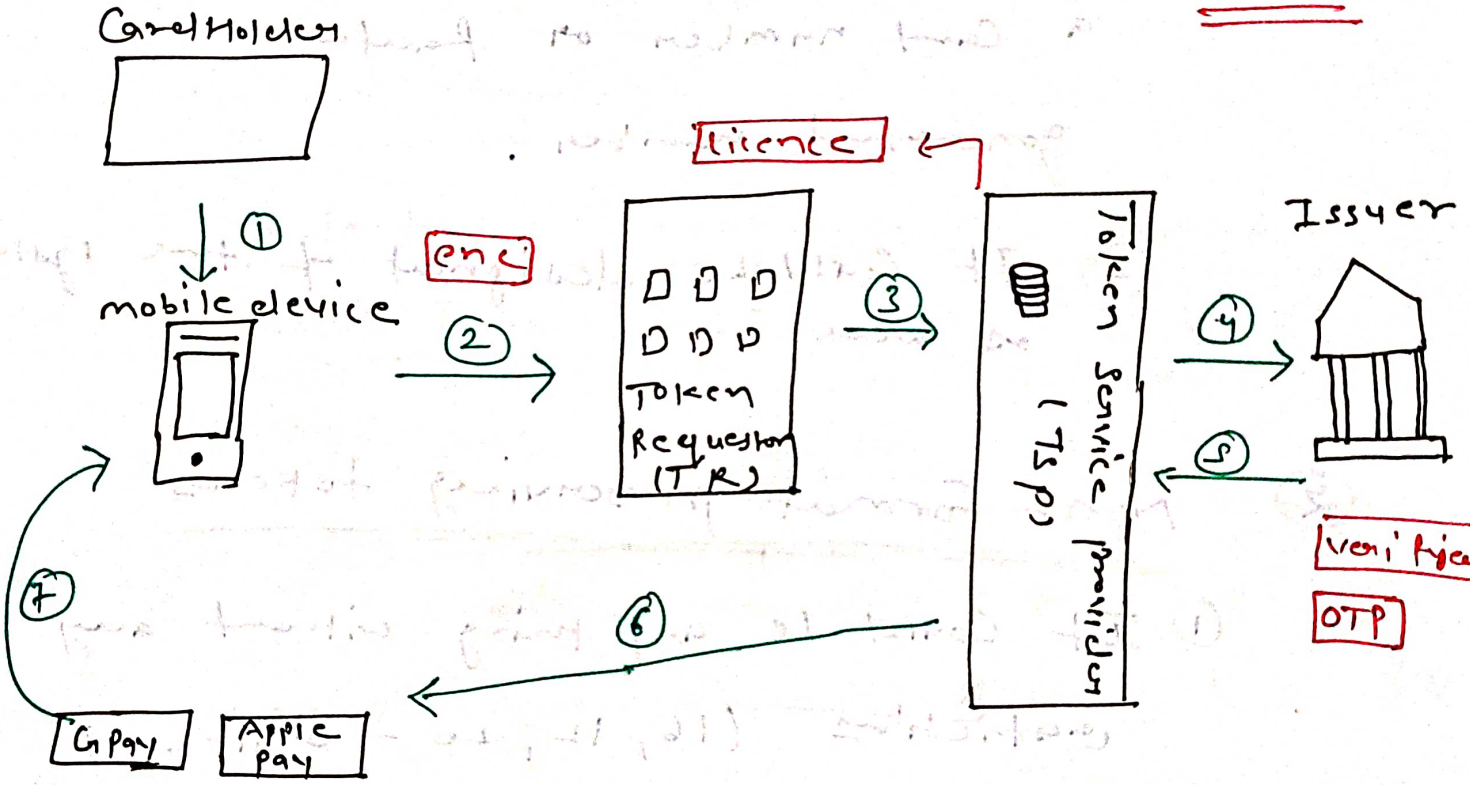
(2) Card-on-file → ~~after~~ already added in system, trying to add same Card

(3) In-app-provisioning → System creates a virtual Card, generated in real time.

④ Token is very much device specific

→ generated in Hardware / Secure element

⑤ Card Registration and Token - First Time



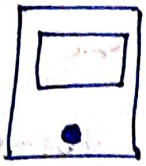
TR → Just ask and Create the token

NOT Store

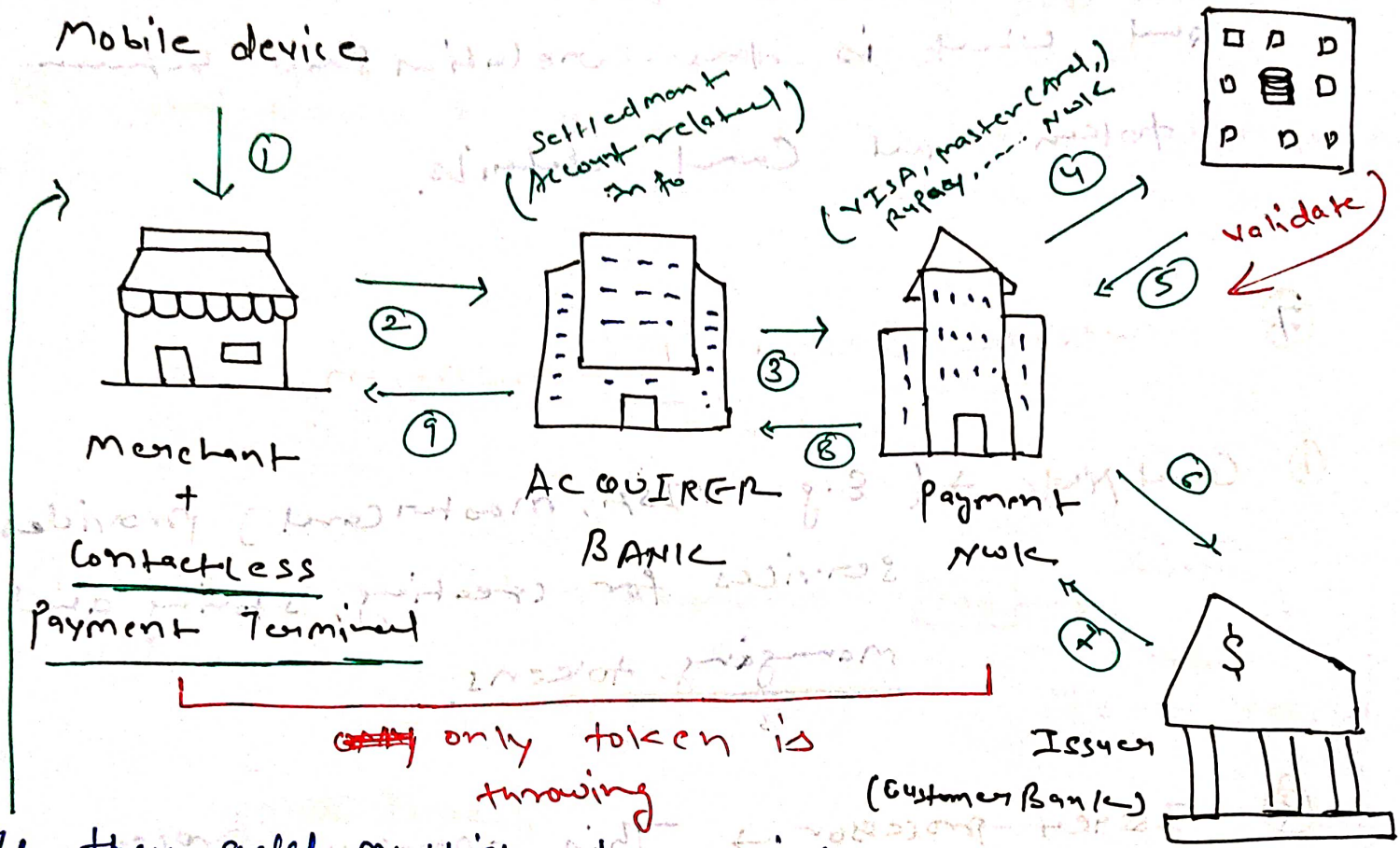
⑥

⑥ Tokenised Transaction

G-Pay, Apple-Pay



(Here token already generated)



① If they add multiple transaction's then they can choose that from which card they want to make this payment.

② ACQUIRER :- Will not any sense of token, Because neither they have generated token, neither they have loss of token

③ TSP → This is repository for all the token.

(6)

(3)

Token is only valid and valuable to the Eco system, which it's get generated. which means only the Tsp will be make sence of what is this token and what is the relationship between token and Card details.

(7) Network Based Tokenisation

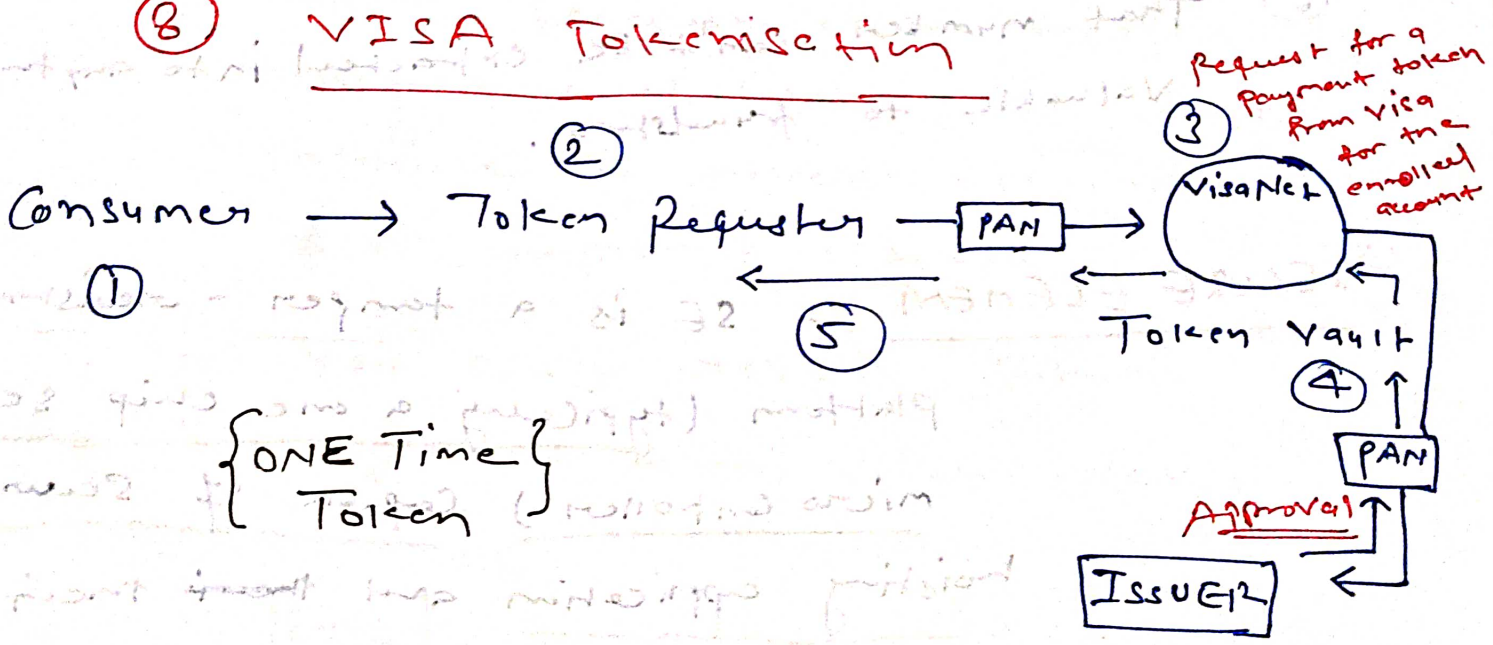
(1) Card Network → (E.g - VISA, MasterCard) provides services for creating, storing and managing tokens.

(2) Issuer-processor → This approval process require Integration and Certification with tokenization services of the Card Network. Basically Issuer should be approved and verified the details.

③ Digital wallet → (G-pay, apple-pay) requests and store tokens, it also require certificates in order to utilize Node tokenization services for request allowing and make purchase with tokens.

④ Cardholder → Card → digital wallet → request a token for card.

⑧ VISA Tokenisation



① Consumer enrolls their Visa account with a digital payment service (such as online ~~retailer~~ retailer or mobile wallet)

⑨ Apple-Pay Tokenisation

1. Take picture of credit card.
2. Apple send the details to the card's NWK.
3. Card NWK validates the info with issuing bank.
4. NWK → TSP → token created (DAN) (customer key) with token
5. Random number (token) is sent back to Apple.
6. That number can't be extracted into anything valuable to fraudsters.

SECURE ELEMENT :- SE is a tamper-resistant platform (typically a one chip secure micro controller) capable of securely hosting application and their confidential and cryptographic data in accordance with the rules and security requirements set forth by a set of inter-identified trusted Authorities.

Encryption and ~~Tokens~~ Tokenization

→ Encryption with reversible cryptographic algorithms was the preferred method of protecting sensitive data.

→ Encrypts → card → decrypt

→ Tokenization replaces sensitive cardholder details with a stand-in token. Because of the random assignation of tokens, it's almost impossible to reverse-engineer or compromise a token.

→ You can't reverse token (Need token vault access)

ENC	DEC
① Mathmetically transform plain text to cipher text using algo and key.	① Randomly generate a token for plain text and store the mapping in DB.
② Scale to large volume with just enc key to decrypt data.	② Difficult to scale securely and maintain performance as db in size.

(10)

(3) Structured fields, as well as unstructured such as entire files.

(4) System or third party can decrypt the data who have enc key

(5) Format preserving ~~issues~~ issues

(6) Entire file move

(3)

(3) used for structured data fields such as payment card or Social Security Num

(4) difficult to exchange data because token vault access required.

(5) format can be maintain.

(6) Relation ship of token resides, token does not move.

(11) Risk and security

(1) cyber attack target could shift to token databases. (where the relation maps between token and PAN).

(2) use both, first encryption and then tokenisation.

③ ~~[redacted]~~ LUHN Algo

(London unbeatable House Nancy)

↳ system which creates token or encryption

④ User → Card Number Enter → should be encrypted immediately for protection

⑫ HSM with Encryption and Decryption

Hard security Module → is a physical/cloud computing device or servers that manage digital keys, perform enc and dec functions for digital signatures, string authentication and other cryptographic functions.

→ PIN number generate ~~Entirely~~ Randomly by HSM

MA → Magnetic Stripe Card

Back side in 1/04 debit/credit card

Chip

(13) PCI DSS tokenisation

① single use (represent specific single transaction
↓
Depends on parameter

multiuse (g-pay, apple-pay)

Bank → PAN

→ Each & Every device will have may be a separate token.

(14) Components of PCI DSS

① Tokenization system common components

→ mathematically irreversible cryptographic function based on strong cryptographic algorithm and strong cryptographic key.

① Non-reversible (only generated)

② Token Mapping

① Token assign to PAN

② Both stored in card-data valut

(3) Card Data Vault

→ Central repository for PANs and tokens, used to store token - mapping process.

→ attractive target for attackers.

(4) Cryptographic key management.

(14) Merchant's Responsibility

(1) protection of Cardholder data is properly scoped and enforced.

(2) Verify the adequacy of any segmentation Controls, are not part of supplied solution.

(3) perform risk ~~assess~~ assessment as part of the due diligence when selecting Tsp.

(4) Ensure proper contractual agreements Tsp, Cardholder, stored...

(5) maintain and implement policies of tokenization services.

(6) security policies requirements, merchants PCI DSS.

(7) Including but not limited to:-

- Data retention and Disposal
- Access control and authentication
- usage policies
- vulnerability mgmt
- logging, monitoring and alerting.

(7) Disaster recovery plans, in TSP on cancel value delete or loss

(15) OAuth

(1) Access Token :- are the things that application use to make API requests on behalf of a user.

→ The access token represents the authorization of a specific application to access specific part's of user's data.

→ Limited time period, does not give unlimited access.

(1) The access token mean they are just getting an access from third party application

(2) Refresh Token :- If access token expired, without informing the customer.

→ client_Id and client_secret key.

It likes that this is still valid and authorized to the access token

This is by used to the true access token

* OAuth → open Authorization

is for machine logging into machines on behalf of humans.

→ It's authorization and not authentication.

During auth, observe the address bar

OAuth