Driving Forward with Tokenization and HCE

An SPA Position Paper

October 2014

1. Introduction

For a long time now, the retail payment market has been an exemplar of stability with well-established and socially accepted payment methods and little incentive for breakthrough innovation. However in the last five years, thanks to technological evolution and the impact of new market entrants, an unprecedented change in the way card payment instruments are issued and processed has been taking place.

Two key areas for driving this innovation are internet and mobile payments. Both have stimulated the emergence of new business models that have resulted in new players arriving on the retail payments scene. As a result, Tokenization by EMVCo and Google's Host Card Emulation (HCE) have become part of the maelstrom. More recently we've seen Apple Pay step into the ring with a proprietary payment solution that uses a Secure Element – which is good news from a user security perspective.

Tokenization sets out to prevent card data from being compromised when sent over public networks or stored in large databases held by retailers. Bound to a single transaction or a particular transaction context, tokens are of limited or no value to fraudsters. So, while a security breach of a token database may give fraudsters access to tokens used in past transactions, these can't be 'replayed'.

Host Card Emulation (HCE), on the other hand, makes it possible for applications residing in a mobile device to use a mobile device's NFC interface to communicate directly with a contactless terminal. This means application owners do not need to negotiate terms with issuers of the security element when deploying applications in NFC-enabled mobile devices.

At first sight, both these technologies appear to address different problems - Tokenization is all about security, while HCE is an NFC-enabler. But in practice these technologies complement one another rather well; for example, while HCE is not secure enough to store permanently static payment credentials, a token is a dynamic credential that's created for the purpose of a one-time payment. In other words, HCE is the perfect use case for tokenization.

The SPA believes there is a growing trend towards omni-channel payment methods, irrespective of the purpose of the payment (person-to-person, person-to-small business, person-to-business). In this context smart card technology, with different form factors that are adapted to different channels, provides the ideal central unifying technology foundation for the roll-out of safe payment applications.

In this paper the SPA clarifies the relationship between Tokenization and HCE, and evaluates the potential impact of Apple Pay. The SPA also sets out why smart card technology represents the perfect baseline for a new generation of payment instruments that fulfil the needs expressed by a variety of different stakeholders.

2. Is HCE an unconditional catalyst for NFC payments?

Host Card Emulation (HCE) technology enables application developers to eliminate the need to gain authorization by a third party to download applications to the mobile device. Obviously, this sounds attractive, both from a business model perspective and the point of view of delivering the freedom to manage mobile applications. A mobile device's communication channels can facilitate the provisioning of applications, while the NFC interface enables the execution of applications on proximity contactless readers. Thus, HCE might well become a primary driver for the deployment of NFC enabled terminals. The result of this would be a multiplication in the number of NFC payment transactions - along with a rise in the number of disputed transactions and chargebacks, should applications and/or data be compromised.

However, mobile devices are vulnerable computers. The standard mobile device architecture is not designed to resist physical attacks and tools are available today that make it easy to revert iOS and Android applications back to high-level source code. Other than malware threats, it's also worth remembering that a substantial number users are engaged in 'rooting' (gaining access to the administrative commands and functions of a mobile device's operating system) – an action which further weakens security software controls. By contrast, extracting data stored in a secure element requires dedicated laboratory equipment and many expert days of labor.

The SPA's proposed approach is different. Our customer-centric approach means that SPA members only market mobile technology that protects a user's financial assets. Customer-centric models are driven by trust and the mobile devices used to pay must be trusted by users. Which means that as a minimum; (1) these must provide a user interface that enables the customer to verify and authorize the payment data, and that (2) the mobile device under no circumstances will leak personal data that may be used for fraud purposes. Trusted displays and input devices, as well as isolated storage and execution environments with strict access control mechanisms, are essential for mobile financial services.

The value of HCE for users is that it's easy to provision applications and run these through the NFC interface. So, yes - HCE has the potential to dramatically stimulate growth in the number of NFC-initiated transactions. But this increase should be balanced against the potential risks for the users.

Without a secure hardware tamper-resistant device, HCE payment data are at real risk of exposure in the mobile device. But tokens offer a key security control in the HCE context; instead of real payment data, a token (in other words, a representation of these real data) could be used to pay.

3. Tokenization: it looks good but what about real-life implementations?

The idea behind tokenization is attractive - real card data for a transaction is replaced by a token, a temporary surrogate containing the same data structure as the original data. So, should the token be captured ‘on the fly’, only that specific transaction is compromised. Tokens can also feature attributes that makes them even less attractive for an attacker; for instance, they may be issued only to pay a particular merchant.

Tokens constitute a candidate security control for using HCE to pay without having to store card data in a mobile device. For example, a local HCE application would request the generation of several tokens by a Token Service Provider which are then stored in the mobile device for immediate or future use. Since these are stored in the mobile-rich OS which potentially puts means tokens in the mobile device at risk, the real card data is outsourced to the Cloud. This way, the only sensitive data stored on the device is the authentication data required for controlled access to the Cloud and/or the Token Service Provider (who may also be in the Cloud).

Assuming the token is properly implemented, this approach ensures financial risk is limited - even if an attacker captures the token, they won’t be able to access the original card data information. But the secure generation and transmission of tokens is not that easy. As Tokenization standards for NFC or remote payments are still at the infancy stage, real-life implementations are currently unable to guarantee security as commonly accepted evaluation and certification requirements still don’t exist. As a result of weak standards – and flawed token system implementations – attacks that have successfully retrieved a card number from a token have been well publicized in the press.

To avoid this scenario, a mobile device needs to be authenticated – and if the mobile device can't authenticate itself, then a third party must confirm the mobile device’s identity to the Cloud Service Provider (CSP). Next, a protected channel needs to be established between the CSP and the mobile device to support the transmission of session keys or generated tokens - and the CSP must only send tokens to a legitimate mobile device.

So, the conclusion for real-life implementation is that, yes – Tokenization is possible in mobile payment scenarios, provided that robust security controls are implemented in the mobile device. A complex key management system for authenticating all the parties involved in the transaction (mobile device, merchant payment service provider, terminal, cloud service provider, token service provider, and financial institutions) needs to be designed and certified.

As such, tokenization might well act as a key enabler for HCE for payments, provided that a secure certified infrastructure is built around the entire ecosystem.

4. How does Apple Pay fit into this ecosystem?

When it comes to resisting attacks, there’s no contest between a payment application that stores secrets on the main Android OS - such as a payment application using HCE - versus one implemented in a dedicated hardware device as a secure element. The SPA therefore welcomes the choice of a secure element by Apple to support its Apple Pay solution.

It’s worth noting that Apple Pay also supports tokenization, despite the fact that the payment static credentials are securely stored in the secure element. The SPA currently does not have critical technical details on how Apple Pay has been designed; specifically, the enrollment process and the card and payment data which is available for the customer to authenticate and pay.

But the secure element is a perfect place to generate a token because it already stores the original card data. Which means that in a remote payment context using a vulnerable network (for example, mobile commerce), a token may be transmitted instead of the real card data. This token may be electronically signed by the secure element, so that only the secure element issuer may identify the payment account the token is associated with. The online retailer also benefits, thanks to a reduction in the perimeter of the PCI-DSS certification.

5. The SPA suggests ‘safing’ - safe innovating with card technology

The SPA message is clear: users of mobile financial services must trust the applications and devices at his/her disposal. Mobile devices are vulnerable computers, so security-relevant functions should be performed in separate tamper-resistant hardware that has been adequately evaluated. But the security evaluation and certification of secure components hosting payment applications adds cost and should therefore be streamlined. In this regard, the SPA is actively working on new procedures to optimize the renewal of certifications for payment applications in a secure element.

But trust has to be built up with patience and can easily be lost. Liability shift alternatives – such as the ‘just use it and if something goes wrong we guarantee you'll be paid back’ - are not good and may incentivize the use of unsafe technology until the first incident. In any payment transaction, the security of the user must prevail over apparent convenience. Furthermore, payment transaction disputes are time-consuming and undermine the loyalty of the customer.

The use of card technology for mobile device payments offers a lot of advantages for the mobile financial service provider, including:

- the same level of security as existing EMV chip cards; malware installed in an application in the mobile device or in mobile rich-OS is not able to access the payment and/or authentication credentials stored in the SE
- the acquisition infrastructure is the same as that used for traditional EMV card payments
- well-known interoperability standards, implementation specifications, evaluation and secure certification processes.

As a second defensive line, HCE could make use of a Trusted Executed Environment (TEE) to store payment credentials and run payment applications. The TEE implements an intermediate security environment, which although not as equally strong as the Secure Element is by far preferable to pure software controls.

Finally, the TEE combined with a Secure Element adds the key functionality of a trusted path between the mobile user interface and the secure element. The SPA supports this configuration as being by far the most cost-effective solution to implement efficient security countermeasures to well-known mobile device vulnerabilities.