Mobile Payment Security discussion paper
# Contents

**EXECUTIVE SUMMARY** 4

1. **INTRODUCTION** 5

2. **INTRODUCTION TO SECURITY** 6
   2.1 Different types of security 6
   2.2 Security objectives 7
   2.3 Types of security measures 7

3. **SECURITY IN MOBILE PAYMENT** 8
   3.1 Securing data storage 8
   3.2 Ecosystem vulnerabilities 11

4. **HARDWARE AND SOFTWARE SECURITY SOLUTIONS IN MOBILE PAYMENTS** 14
   4.1 Secure element and HCE 14
   4.2 Security assessment framework 16
   4.3 Security assessment 16

5. **SUMMARY** 18
Executive summary

Mobile payment services are evaluated to understand the level of risk they provide. This evaluation is used to provide a transparent level of risk that can be assessed, accepted and managed by issuing banks.

With host-based card emulation (HCE) offering new ways of implementing services, issuing banks are interested in the opportunity to launch services with more flexibility in a less complex ecosystem. There is, however, a security trade-off for moving from hardware to software-based solutions. While mobile payment solutions using mature hardware-based security measures, such as the tamper-proof secure element (SE), have been tested against the latest market attacks, newer solutions that use software-based security, such as HCE, have an unknown level of security. Security measures can be implemented to reduce the likelihood and impact of a successful attack based on current assumptions and use of logical and procedural security measures. However, the efficiency of a specific implementation of these measures will have to be verified and evaluated over time.

Mobile contactless payments have traditionally been secured by a tamper-proof hardware SE, inspired by chip-and-PIN cards, that controls the transactions’ security level through a well-known process. This requires the involvement of multiple players, in particular mobile operators when the SE is the SIM. HCE makes it possible for software-only payment applications to access the handsets’ contactless interface without using a hardware secure element. This interests banks, who are eager to deploy their applications with more flexibility. Without a secure element, however, HCE does not provide the tools to protect payment applications and additional security measures are needed to reduce the likelihood and impact of a successful attack.

This paper explores the typical vulnerabilities of a mobile payment system and the security measures applied to payment solutions with and without a hardware secure element, evaluating them against:

- Security constraints—to reach a comparable risk levels
- Complexity and cost—to implement the considered security measures
- Usability—impact on the user experience
- Auditability—ability to evaluate the risk level with a good level of confidence.

Four major conclusions are drawn:

- **Complexity** of the overall ecosystem may reduce for software-based security solutions in the absence of a hardware SE. At the same time, complexity shifts to the issuer, which will bear the burden of all security measures in its back-end, application and processes.

- **Cost** is impacted in both solutions: the complexity faced by the issuer to implement and manage new software-based security measures will incur unknown costs, while hardware-based SE needs significant investment into multi-player infrastructure, with costs distributed between players.

- **Usability** on a software-based solution will require the provisioning of temporary tokens on handsets to authorise transactions and perform regular security updates that require the user to re-authenticate. This is so the solution is able to access the sensitive data stored in the cloud. Careful design will be required to preserve the quality of the user experience.

- **Auditability** via formal processes is in place for the testing, certification and the security evaluation of hardware-based SE solutions. Similar processes are difficult to put into place for a large variety of non-standardised software-based solutions. For now, evaluation must be performed on a case-by-case basis, and the actual risk level will be evaluated more reliably when enough experience has been gathered in the market.
1. Introduction

As mobile handsets become more prevalent as a tool for payments and banking, the security of sensitive data used becomes increasingly important. Considerations, such as how the applications and transactions are secured and delivered, and how the user is securely identified, are key concerns of the mobile payment ecosystem. Payment solutions are being evaluated for risk management by banks investigating implementing mobile payment services.

Aimed at issuing banks, and written in collaboration with UL, this discussion paper provides a high-level and informational overview of the security technology involved in the implementation of mobile payment solutions. More specifically, the paper considers the security implications of hardware (e.g. secure element) and software (e.g. host card emulation) based mobile payment solutions.

The paper outlines the overall objectives, types and needs for security with a specific focus on mobile payments. It covers sensitive data storage, mobile security measures and vulnerabilities in the ecosystem, and provides an assessment framework that compares the available hardware- and software-based solutions in the market today.
2. Introduction to security

Security is often measured alongside risk and the total risk associated with payments. This chapter provides a broad overview of the different types and objectives of security, and explains the measures of risk mitigation.

2.1 Different types of security

Security can be represented as a triangle, with three linked security corner stones: physical, logical and procedural security. The total level of security is determined by the weakest link.

**Physical security**
Physical security is based on physical or hardware elements. The aim of physical security is to safeguard data by means of tamper-proof hardware elements where this sensitive data is stored and crucial operations are executed. This is discussed further in section 3.1

**Logical security**
Logical security is based on software safeguards placed in a system. While software needs hardware elements to operate, logical security does not use the hardware elements to provide the security. This is discussed further in section 3.1

**Procedural security**
Procedural security is based on safeguarding sensitive information through organisational procedures. Procedural security protects the confidentiality, integrity and/or availability of information by procedures and corresponding organisational security measures, which can be implemented either proactively or reactively. Procedural security is particular to a specific implementation, and needs to be considered on a case-by-case basis.

---

**SECURITY TRIANGLE**

![Security Triangle Diagram]

Figure 1
2.2 Security objectives

The overall goal of security is to protect relevant assets from threats in terms of confidentiality, integrity and availability via security measures.

**Assets**

Assets contain the user's sensitive data, such as the primary account number (PAN), use keys and tokens. These assets can be captured by attackers such as fraudulent users, fraudulent merchants and hackers.

**Threats**

A system is open to attack at different locations, defined as the points of vulnerability. Vulnerabilities are attacked when the assets are considered valuable to the attacker.

**Security measures**

Security measures can be applied to mitigate the risk of a threat to a valuable asset. Security measures may not be used, or may be reduced, to accelerate deployments or reduce costs (such as during pilot or testing phases).

2.3 Types of security measures

Security of an implementation is always evaluated to provide a good estimation of the overall level of risk. This will provide the issuing bank with a transparent level of risk that is acceptable and manageable. Security measures are a way to make it complex and expensive to hack, while reducing the value and visibility of an asset. There are two types of security measures to mitigate risk:

- **Reduce likelihood of successful attack**
  The likelihood of a successful attack is reduced through implementation of robust and adequate security. This could be done, for example, by storing sensitive data in tamper-proof elements. Also, hiding the information via cryptography can make the effort or expense of hacking too significant.

- **Reduce the impact of successful attack**
  The impact of a successful attack is decreased by reducing the value of the assets that can be obtained by an attacker, which reduces the appeal. For example, tokenisation reduces the impact since the assets (tokens) have limited value for an attacker. This strategy allows issuers to accept a lower level of security while maintaining an acceptable risk level.

**Market implementation: Apple Pay**

Apple Pay allows users to make NFC payments. Tokenisation is used to store payment card data. Also, a dynamic security code will validate each transaction. The user confirms the transaction with TouchID.

**Main security mechanisms:**

- SE: Stores the Unique Device Account number and the single use unique transaction numbers
- User and hardware verification: User verified with TouchID
- Tokenisation: Tokenised PAN stored on device, single use unique numbers used per transaction.
- Cryptography: Unique Device Account number cryptographically stored, single use unique numbers cryptographically generated.
3. Security in mobile payment

Issuing banks work with sensitive payment data, and mobile payments introduce additional security challenges for issuers.

Issuers have the challenge of authenticating the user and their device, as well as securely storing the primary account number (PAN), use keys and cryptograms locally on the mobile handset. The overall level of risk should remain manageable for issuers when implementing either hardware-based or software-based security in mobile payment solutions. These measures not only impact ecosystem size but also on the functionality of the solution. The impact needs to be considered alongside critical aspects like usability, costs, auditability and complexity. To manage assets, issuers can implement hardware or software-based security solutions.

3.1 Secure data storage

An important factor in determining the level of security is where sensitive data or credentials are generated and stored. The security of the storage should be dependent on the sensitivity of the data, as each solution has an impact on risk. Issuers have a number of options for generating, processing and storing sensitive data.

**Host Central Processing Unit (CPU)**
Sensitive information is secured in the application running on the operating system (OS) on the mobile handset. The protection from possible attacks is low since the only default security mechanism is ‘sandboxing’. Sandboxing is a security mechanism which separates ‘unverified’ running programmes. In this instance, the Android OS isolates the assets at an OS level.

**Secure storage in the cloud**
Sensitive data is managed on a network server in the cloud. The mobile handset has to establish a secure connection to obtain this data. Potentially, the user must authenticate themself, the handset and the application. In software-based solutions, authentication of the handset is solved by ‘hiding’ use keys within the software using code obfuscation or white-box cryptography.

**Trusted execution environment**
The trusted execution environment (TEE) is a secure area in the CPU of mobile handsets. The TEE environment offers safe execution of ‘trusted applications’ and enforces access rights in addition to storing the sensitive data. The TEE runs its own OS which isolates its hardware and software security resources from the main OS. The TEE environment offers safe execution of applications and enforces access rights in addition to storing the sensitive data. Provisioning sensitive data to the TEE offers similar challenges to provisioning via a trusted service manager (TSM) or application server. The TEE usually lacks the tamper-resistance of a hardware-based SE. Each equipment manufacturer uses its own version of this technology, such as Samsung’s Knox (using ARM’s Trustzone) and Apple’s Secure Enclave.
LOGICAL SECURITY MEASURES FOR MOBILE PAYMENT SOLUTIONS

There are many logical security measures linked to mobile payment applications. These include:

A. User and hardware verification

User authentication can be completed by:
- what the user knows, such as username and password, PIN or pattern
- what the user has, such as unique device identifier or token
- how the user behaves, such as transaction location or transaction history
- what the user is (biometric identification), such as fingerprint, voice or facial recognition

Multiple methods can be combined in two-factor identifiers in the form of a device’s unique number (IMEI or ICCID). For example, hardware authentication could be used on top of user authentication to authenticate the user further.

B. Transaction constraints

Limiting transactions can reduce the impact of a breach of security. Examples of constraints include:
- low value payments and limit number of transactions in a certain timeframe
- allowing only online transactions for proximity payments
- constraints on location/country

These transaction constraints are signed by the issuer and stored in the SE or point of sale terminal (POS), making it very hard to adapt for a malicious user.

C. Operating system checks

Applications are able to retrieve the OS settings to verify if a device is ‘rooted’. The risk of rooted devices is that a user or hacker is able to take control of processes and operations controlled by the OS. A user or hacker can bypass security safeguards and obtains access to the secure storage location. A user is able to:
- alter identifiers of the device or application used in user and hardware verification
- obtain access to the code of applications where sensitive data is stored
- install malware applications which could perform fraudulent actions, such as man-in-the-middle attacks

D. White-box cryptography

The primary purpose of white-box cryptography is to implement a large set of lookup tables and hardcode the key within. All cryptographic operations are handled as a succession of lookup tables during an operation, which then hinders the opportunity to guess the correct key used during an operation. The assets remain secure since they are ‘hidden’ from an attacker; even when the attacker has access to all the resources. By hiding the key within the application, the distribution becomes more complex, as every application needs to be uniquely defined per user.

E. Obfuscation

Code obfuscation is a method to deliberately write the code or parts of the code in a difficult to understand way. This makes it difficult for hackers to reverse engineer the code and conceals critical or secret parts of the code. Code obfuscation comes in several forms:
- Abstraction transformations: Destroys module structure, classes, functions
- Data transformations: Replaces data structures with new representations
- Control transformations: Destroys conditional statements, such as if-, while-, do-
- Dynamic transformations: Makes the program change at runtime

F. Tokenisation

In tokenisation, sensitive data is substituted by a less sensitive equivalent, the token. The value of the asset is drastically reduced and it becomes less interesting for an attacker. Tokens can be readily generated and replaced without compromising the PAN. Two forms of tokenisation are:
- EMVCo tokenisation: A tokenised version of the PAN is linked to the user and stored on a unique device.
- Use key tokenisation: A token is generated for single or limited use that the user authenticates before setting-up the secure connection.
PHYSICAL SECURITY MEASURES FOR MOBILE PAYMENT SOLUTIONS

Secure element
The hardware-based SE is a tamper-proof element in mobile handsets and comes in the form of a universal integrated circuit card (UICC, or commonly referred to as a SIM) or embedded SE (eSE). These secure elements have been tested against the most recent attack methods and security requirements set by trusted authorities.

The main features of the SE are:
• The keys and sensitive data are stored on the SE.
• A secure application running on the SE is needed to actually process the sensitive data. Secure applications hold sensitive data and the combination of a UICC and payment applications are usually certified.
• When the application is running outside the handset OS environment, both the applications and the keys are more difficult to reach for malware.

Market implementation: Softcard
Formerly known as Isis, the Softcard wallet is a joint venture between three mobile operators providing a SE-based mobile wallet application. The wallet offers NFC proximity payments as well as loyalty cards. A virtualised version of the payment card is stored on the SIM of the mobile handset.

Main security mechanisms:
• SE: Storage of sensitive payment information on tamper-proof UICC (SIM)
• Transaction constraints: low value payments (no mobile PIN), high value payments (mobile PIN)
• User and hardware verification: mobile PIN to unlock the application + Device ID verification
3.2 Ecosystem vulnerabilities

A high-level analysis showing potential threats and vulnerabilities in the ecosystem is shown in Figure 2. Possible measures to mitigate the risk per threat are also given. Every implementation will require an individual risk analysis to identify specific vulnerabilities.

EXAMPLES OF ECOSYSTEMS VULNERABILITIES:

Cloud-based system
A successful attack on the cloud-based system would entail a hack that tampered with data or disclosed sensitive information. Possible threats to the cloud payment system include the interception of sensitive data by an attacker spoofing his identity, interception of data by malware and the redirection of application data to a different mobile handset. With these methods, an attacker can obtain assets such as user and card details, card verification method values, use keys, and even entire mobile payment applications. The biggest challenges are secure access and authenticating the user to the cloud. For a cloud-based system to handle tokenised payments, it needs to be able to manage these tokens, and tokenise and detokenise the sensitive data.

Mobile payment application
A successful attack on the software-based mobile payment application could consist of decompiling the source code, where the attacker obtains access to all assets hidden in the application (such as tokens and cryptographic keys). The integrity of an application can also be compromised by data tampering and cloned applications intercepting sensitive data. Another point of vulnerability is a merchant’s mobile POS, as a fraudulent merchant could tamper with the mobile application controlling the mobile POS. With these methods, an attacker can obtain assets such as user and card details, card verification method values, and use keys. Security mechanisms, such as white-box cryptography, reduce the likelihood of cloning and decompiling payment applications. Provisioning of secure data to the SE or delivery of a payment token is a point of vulnerability in mobile payment applications.

Mobile handset
Attacks on the mobile handset could address assets in the device: SE, near field communication (NFC) processor and mobile OS. The mobile OS has weak points through the debug port or rooting bugs, which can obtain access to the mobile payment application. Additionally mechanisms, such as a screen logger, can obtain key entry information from the user. Potential security measures are offline PIN, OS checks, white-box cryptography and TEE to secure the keyboard and screen.

Secure element
The impact of an attack on the SE is high, however, the likelihood of a successful attack is very low due to the high security level of the tamper-proof SE. The most critical security aspect is the access control to the SE, which defines access for mobile applications. Access to the SE is based on certificates implemented by the service provider, which are used to authenticate or sign the service provider application to the SE.

Point of sale – NFC interface
The NFC POS interface could suffer from relay attacks for low-value payments. The NFC interface can be misused by a fraudulent merchant simulating ‘no PIN’ payments. Tampering with data could also be done by an attacker collecting use keys. Malware in the NFC processor is a possible entry to redirect sensitive data. The impact of such attacks can be reduced by implementing security mechanisms, such as tokenisation.
**THREATS**
- Tampering by decompiling the application. Attacker obtains user and card details, CVM values, use keys.
- Spoofing of identity/phishing fake/created application and captures sensitive data (use keys, mobile PIN)
- Tampering by injection of malicious code into application. Attacker obtains sensitive data.

**POSSIBLE MEASURES**
- TEE (Sec 3.1)
- White-box cryptography (D)
- Operating system checks (C)
- Obfuscation (E)
- User verification and hardware verification (A)

---

**THREATS**
- Tampering by getting access to sensitive data through device debug port or rooting bugs
- Tampering by running in simulated environment to find vulnerabilities
- Attacker finds vulnerabilities and captures user and card details, CVM values, use keys

**POSSIBLE MEASURES**
- Operating system checks (C)
- White-box cryptography (D)
- Obfuscation (E)
- TEE (Sec 3.1)
- User verification and hardware verification (A)

---

**THREATS**
- Tampering with access control between OS and SE, getting access to sensitive data
- Phishing which captures sensitive data (use keys, mobile PIN)
- Intercept application

**POSSIBLE MEASURES**
- Tamper-proof secure element (Sec. 3.1) (covers most threats)
- User verification (A)
This chapter presents a security assessment framework to help make a comparison between hardware and software-based security solutions for mobile payment. Typical hardware- and software-based solutions are assessed in Figure 4.3.

4.1 Secure element and HCE solutions

This section discusses one hardware and one software-based solution of similar security level including relevant security measures for risk mitigation.

Figure 4.1 looks at a hardware security solution based on the SE. This example has the payment card stored as a virtual card in the SE. The assumptions for this example are:

- All payment card data is located on and retrieved from the SE
- Low-value payments may be allowed with reduced security measures, such as no mobile PIN

<table>
<thead>
<tr>
<th>FIGURE 4.1  SECURE ELEMENT-BASED HARDWARE SECURITY SOLUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECURITY ASPECT</td>
</tr>
<tr>
<td>Security measure</td>
</tr>
<tr>
<td>Type of security</td>
</tr>
<tr>
<td>Risk mitigation</td>
</tr>
</tbody>
</table>

In a hardware-based solution, the main security comes from the SE (such as the SIM or UICC). To provision the secure applications and personalise the payment application, a secure connection is established with the SE. Issuing banks are able to fully control their assets within the SE through the GlobalPlatform standards.1

Figure 4.2 looks at a cloud-based software solution based on HCE. In this example, the payment card data is stored in the host CPU. The mobile application running on the host CPU manages the payment card data and connects to a cloud-based payment system to authenticate the user and approve transactions. The assumptions for this example are:

- Tokenisation is used by the application to generate use keys. These will contain any used sensitive data in tokenised form and provide user authentication
- White-box cryptography is used to protect sensitive data
- Offline or online PIN is triggered, depending on the transaction

<table>
<thead>
<tr>
<th>SECURITY ASPECT</th>
<th>HCE BASED EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security measure</td>
<td>1. White box cryptography</td>
</tr>
<tr>
<td></td>
<td>2. Tokenisation</td>
</tr>
<tr>
<td></td>
<td>3. User authentication</td>
</tr>
<tr>
<td>Type of security</td>
<td>Logical</td>
</tr>
<tr>
<td>Risk mitigation</td>
<td>Reduced likelihood of successful attack</td>
</tr>
<tr>
<td></td>
<td>Reduced impact of successful attack</td>
</tr>
<tr>
<td></td>
<td>Reduced likelihood of successful attack</td>
</tr>
</tbody>
</table>

In a software-based security solution, a connection is made directly from the application, via NFC, with the network. To enable payments, the user will need to authenticate to the cloud to obtain payment tokens that will have to be provisioned securely, involving a token issuer. Tokenisation is used to reduce the impact of a successful attack and these tokens will have to be stored on the mobile handset. The software-based solution may involve fewer players, but also needs more software security measures to compensate for the lack of an SE, such as the three assumptions in Table 2.

**Market implementation: BankInter**

This is a software-based mobile payment application offering eCommerce transactions (remote payments) as well as HCE-based payments at POS. Payment cards are stored on and linked to a mobile handset as mobile virtual cards. No mobile connectivity needed for NFC transactions, only for replenishing partly calculated credentials onto the mobile handset.

**Main security mechanisms:**
- Tokenisation: In the form of Mobile Virtual Card creation on the handset and single use cards
- Cryptography: EMV Cryptogram
- Transaction constraints: Time constraint of 60 seconds for single use card
- Validity
- User and hardware verification: Mobile Virtual Card PIN + Device ID verification
4.2 Security assessment framework

To compare solutions, a security assessment framework based on five aspects—security, usability, costs, auditability, and complexity—is used. Other factors, such as the specific means of set-up, deployment, and the feasibility of risk mitigation measures, are not considered in this framework, as they are specific to individual implementations.

4.3 Security assessment

The security assessment framework presents a high-level comparison based on the five aspects—security, usability, costs, auditability, and complexity. A full comparison would require a full risk analysis of two specific implementations. Note, in practice, the evaluation of a software-based solution's risk level will have to be completed over a period of operational monitoring before business rules (certification, risk status of transactions, provisioning for financial risk) can be fully established. Instead a high-level comparison is given which takes into account the general features of hardware- and software-based solutions.

**FIGURE 4.3 FRAMEWORK COMPARISON**

<table>
<thead>
<tr>
<th>FRAMEWORK</th>
<th>HARDWARE-BASED SOLUTION</th>
<th>SOFTWARE-BASED SOLUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECURITY</td>
<td>Reduces the likelihood of a successful attack by providing robust physical risk mitigation as the issuing bank's and user's assets are securely stored on the SE.</td>
<td>A software-based solution does not match the security of a hardware-based solution. To improve the security of a software-based solution, white box cryptography and user verification methods are required which impacts on the other framework aspects.</td>
</tr>
<tr>
<td></td>
<td>The security is provided by the tamper-proof SE, which has been produced and field-tested for more than 10 years.</td>
<td>The strategy to mitigate risk mainly focuses on reducing the impact of a successful attack through tokenisation and associated transaction constraints (such as only allowing low value payments without a PIN), to achieve a comparable overall risk level to hardware-based solutions.</td>
</tr>
<tr>
<td></td>
<td>Provision of data is securely managed and directly sent to the SE, not routed through the mobile OS.</td>
<td>As previously mentioned, these solutions are a recent development and there is a lack of field experience to fully assess the actual long-term level of risk.</td>
</tr>
<tr>
<td></td>
<td>Security has been certified by card schemes against the most recent known attacks.</td>
<td>Note, in practice, the evaluation of a software-based implementation's risk level will have to be assessed on a given implementation. Within the framework, the risk estimation is performed according to the presented ecosystem, with the points of vulnerability and their possible risk mitigations weighted by the impact of the possible fraud. Storage location of sensitive data is an important factor to determine the overall level of security. Factual security is assessed in this section as user perceived security varies by market and changes over time.</td>
</tr>
<tr>
<td>USABILITY</td>
<td>The usability of a hardware-based solution is good since it offers the possibility to perform low value payments without a PIN and payments when the mobile handset is turned off, which is comparable to a physical payment card.</td>
<td>The device needs to be turned on to make transactions.</td>
</tr>
<tr>
<td></td>
<td>When the SE is a UICC (SIM), users can potentially transfer it between handsets.</td>
<td>Mobile connectivity is regularly needed to connect to the cloud-based token server to replenish the payment tokens.</td>
</tr>
<tr>
<td></td>
<td>The issuer does not need to contract external parties such as TSMs and mobile operators or SE managers. However, other players such as a token service provider and player providing/updating the software security measures may be needed. Additional costs to manage new complexities (see Complexity) will occur costs, the extent of which are not yet known.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This is robust as there is an established process in place for many years for the testing, certification and formal security evaluation.</td>
<td>The application still has to be provisioned securely and personalised with keys or tokens which could be done internally or an external party.</td>
</tr>
<tr>
<td>COSTS</td>
<td>To securely provision and personalise virtual payment cards on mobile handsets, parties like TSM vendors are needed and mobile operators might charge a fee for using the UICC (SIM) as an SE.</td>
<td>Auditable is not as mature as hardware-based solutions as formal processes are under development (including by Visa and MasterCard).</td>
</tr>
<tr>
<td></td>
<td>Significant investment is needed into multi-party infrastructure: usually costs for the TSM set-up and management are covered by the SE owner and the connection to the TSM is covered by the user.</td>
<td>Processes are purposely obscure or cryptographically protected, making them less transparent and, consequently, less auditable.</td>
</tr>
<tr>
<td>AUDITABILITY</td>
<td>This is robust as there is an established process in place for many years for the testing, certification and formal security evaluation.</td>
<td>The complexity of an SE-based mobile payment solution comes from the large ecosystem and the number of external parties involved.</td>
</tr>
<tr>
<td></td>
<td>The strategy to mitigate risk mainly focuses on reducing the impact of a successful attack through tokenisation and associated transaction constraints (such as only allowing low value payments without a PIN), to achieve a comparable overall risk level to hardware-based solutions.</td>
<td>The total ecosystem may have less players, as the role of mobile operators and TSMs is replaced by token service providers and software security measure managers. The complexity at the issuing bank may be increased. The issuing banks' back-end systems will need to be adapted by incorporating a cloud-based payment platform that manages the mobile payment application and is capable of authenticating users and handling this type of transaction. As such, significant changes have to be made in the back-end systems of an issuing bank.</td>
</tr>
<tr>
<td>COMPLEXITY</td>
<td>The complexity of an SE-based mobile payment solution comes from the large ecosystem and the number of external parties involved.</td>
<td>When the application incorporates cryptographic algorithms, another party is needed to deliver keys or security updates. Another example is tokenisation where a token manager may be required.</td>
</tr>
<tr>
<td></td>
<td>These parties need to integrate and operate together to launch a successful solution. Integration and interoperability issues arise due to the number of interfaces and the complexity of on-boarding additional issuing banks and mobile operators.</td>
<td>The complexity at the issuing bank may be increased. The issuing banks' back-end systems will need to be adapted by incorporating a cloud-based payment platform that manages the mobile payment application and is capable of authenticating users and handling this type of transaction. As such, significant changes have to be made in the back-end systems of an issuing bank.</td>
</tr>
</tbody>
</table>
5. Summary

Similar risk levels can be obtained between solutions, however the impact on the usability, costs, auditability and complexity varies, as shown in Figure 5.1.

<table>
<thead>
<tr>
<th>FRAMEWORK ASPECT</th>
<th>IMPACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Hardware-based solution: Proven and well-understood level of security. Software-based solution: Similar levels of security can be achieved, but an implementation needs time in the field to assess and verify the level of risk.</td>
</tr>
<tr>
<td>Usability</td>
<td>Hardware-based solution: Usability is good. Once the payment product is provisioned to the user, the usability is at the same level as a physical payment card. Software-based solution: Usability can be at a comparable level, but is dependent on the specific software-based security measures an issuing bank will employ. These measures can hamper the usability, and could add cost factors and complexity for issuing banks.</td>
</tr>
<tr>
<td>Costs</td>
<td>Hardware-based solution: Involvement of many parties like TSMs and mobile operators, to securely provision the payment product to the user, raises costs in this multi-party ecosystem. Software-based solution: For a similar level of security, tokenisation and cryptography measures are required, adding cost factors such as a tokenisation service provider and a license fee to update the software security. The back-end system of an issuing bank should be adapted, which is also a cost factor.</td>
</tr>
<tr>
<td>Complexity</td>
<td>Hardware-based solution: Complexity lies in the multi-party ecosystem. Software-based solution: Complexity lies with the issuing bank.</td>
</tr>
</tbody>
</table>
Overall, risk can, in principle, be kept at a comparable level when implementing a software-based **security** solution by implementing logical security measures such as tokenisation, transaction constraints and white-box cryptography. A large array of logical security measures should be implemented to compensate for the lack of a hardware-based SE.

To support the provisioning of tokens and software security updates, strong user authentication might be required for secure access from the handset to the secure storage in the cloud, which impacts the **usability** of a software-based solution. Despite this, the user experience could be kept at a comparable level to a hardware-based application, depending on the specific implementation of the security measures.

**Cost** of launching and operating a software-based solution could be reduced by the lack of dependency on external partners, however, additional costs to the issuing bank are likely as technical complexity is absorbed by the issuing bank. Either the solution is developed in-house or by additional players, such as a tokenisation service manager.

**Auditability** via formal processes is in place for the testing, certification and the security evaluation of SE-based payment applications, however, for a software-based solution processes are under development and will be determined over time.

**Complexity** shifts to the issuer with a software-based solution, and is linked to the time-to-market. By choosing a software-based solution, there is no need for the large scale integration of many parties in the ecosystem. However, the complexity on the side of the issuer is increased in terms of back-end integration, which increases time-to-market in addition to the implementation of software-based security measures.

The launch of HCE-based payment applications has only recently become possible. Visa and MasterCard have launched requirements for cloud-based payments to licensees and a test and security evaluation process is under development to certify this new type of mobile payment solution – but the definitive answers regarding the level of risk will come from the operational work of the pioneers and, until then, risks will need to be evaluated and assessed on a case-by-case basis.
ABOUT UL

UL is a premier global independent safety science company that has championed progress for 120 years. Its more than 10,000 professionals are guided by the UL mission to promote safe working and living environments for all people. UL uses research and standards to continually advance and meet ever-evolving safety needs. UL partners with businesses, manufacturers, trade associations and international regulatory authorities to bring solutions to a more complex global supply chain.

UL’s Transaction Security division guides companies within the mobile, payments and transit domains through the complex world of electronic transactions. UL is the global leader in ensuring security, compliance and global interoperability. Offering advice, test and certification services, security evaluations and test tools, during the full life cycle of your product development process or the implementation of new technologies. UL’s people pro-actively collaborate with industry players to define robust standards and policies. Serving you locally whilst acting globally. UL has accreditations from industry bodies including Visa, MasterCard, Discover, JCB, American Express, EMVCo, PCI, GCF, ETSI, GSMA, GlobalPlatform, NFC Forum and many others.

For more information visit www.Ul.com

ABOUT THE GSMA

The GSMA represents the interests of mobile operators worldwide. Spanning more than 220 countries, the GSMA unites nearly 800 of the world’s mobile operators with 250 companies in the broader mobile ecosystem, including handset and device makers, software companies, equipment providers and Internet companies, as well as organisations in industry sectors such as financial services, healthcare, media, transport and utilities. The GSMA also produces industry-leading events such as Mobile World Congress and Mobile Asia Expo.

The GSMA’s Digital Commerce programme works with mobile operators, merchants, banks, payment networks, transport operators and services providers to support the deployment of mobile commerce services. By fostering the ecosystem to encourage and facilitate collaboration, the programme collaborates with the mobile ecosystem to develop specifications and guidelines for technical implementation and build value propositions for adjacent sectors.

For more information visit www.gsma.com