Trusted Platforms to secure Mobile Cloud Computing

Thinh LE VINH
CEDRIC Lab
Conservatoire National des Arts et Métiers - CNAM
Paris, France
le_vi2@auditeur.cnam.fr

Samia BOUZEFRANE
CEDRIC Lab
Conservatoire National des Arts et Métiers - CNAM
Paris, France
samia.bouzefrane@cnam.fr

Abstract— Taking full advantage of the availability of Cloud computing facilities, Mobile Cloud Computing (MCC) is becoming more and more popular as an element of mobile technology. This availability enables mobile users to use the cloud infrastructure to overcome the limitations of mobile technology; namely limited data storage, processing power and battery life. With the increasing use of mobile cloud computing, security issues remain a challenge. This paper investigates some available trusted platforms in order to pinpoint the question on how to secure the MCC under trusted platforms. This investigation comprises an overview of the security aspects of Trusted Platforms including Secure Elements (SE), Host Card Emulation (HCE), Trusted Execution Environment (TEE), and Trusted Platform Module (TPM). In addition, it shows how these platforms can be suitable for securing mobile Cloud computing.

Index Terms—mobile computing, trusted platforms, cloud computing, security.

I. INTRODUCTION

According to Cisco [1], “about 36 percent of worldwide mobile service revenues come from data; by 2016, it will increase to 46 percent. Similarly, the 1.3 exabytes per month of mobile data traffic generated today is expected to increase eight-fold by 2016”. Mobile devices will more and more deal with heavy computation tasks because principally of the huge size of data that will be more and more manipulated. Besides, secure elements growth will reach 64% at the end of 2014, according to EuroSmart1, with 435 millions of units, while smart secure contactless devices will exceed one billion of units.

With the emergence of new information and communication technologies, smart devices combined with communication infrastructures such as Internet and mobile networks need to provide reliable services.

For this purpose, many application domains with high social and business impact such as personal healthcare, home automation, mobile payment, may use mobile devices that rely on trusted environment. This trusted environment may be provided or based on many hardware or software platforms such as SIM cards, secure elements for Near Field Communication (NFC), Host Card Emulation (HCE), TEE (Trusted Execution Environment), or TPM (Trusted Platform Module). Even if these trusted platforms are extremely constrained in terms of energy, computational power and memory, they are designed to solve major constraints of mobile devices and to benefit from unlimited Cloud capacities such as storage. However, MCC must offer mechanisms to guarantee data protection and security in untrusted environment. The contribution of this paper is to look into the Trusted Platforms features. Section 2 provides a summary in short of Trusted Platforms. Section 3 deliveries the features of trusted platforms in comparison. Section 4 discusses the different architectures that may implement a MCC model. Section 5 discusses how MCC security issues can require the use of trusted platforms. Section 6 concludes the paper with some perspectives.

II. TRUSTED PLATFORMS

The emergence of a new generation of mobile devices leads to the growing number of security concerns demand attention. Mobile users use their smart-devices for a variety of application in an open environment. They are provided the reliable services to reduce the risks associated with the security threats. As a result, mobile devices need Trusted Platform to meet a proper implemented security. This Trusted Platform can be a Secure Element, a Trusted Execution Environment, a Host Card Emulation, or a Trusted Platform Module.

A. Secure Element

A smartcard is a tamper resistant micro-controller [2] whose security is enforced by multiple software and hardware countermeasures. A secure element (SE) is a smart card embedded in an NFC device such as NFC-enabled smartphone, able to store sensitive data such as PIN code and security keys and perform internal computation such as cryptographic processing when needed by the terminal (phone). A SE has the following physical features: 8, 16 to 32 bits of CPU, 32 to 64 KB of PROM that hosts the operating system, and 1 to 4 KB of RAM to execute programs, 1KB to 128 KB of EEPROM/Flash to store data and applications, while a dedicated crypto-processor is used for cryptographic computation. To interact with a SE, the terminal sends APDU commands and the SE responds using APDU responses.

The SE is generally a Java Card platform (see Fig. 1) composed of a Java Card Virtual Machine (JCVM), a Java Card API and a Java Card Runtime Environment (JCRT).

1 http://www.eurosmart.com/publications/market-overview
JCRE implements the Java Card mechanisms that are intrinsic to the smart cards such as transaction management. When the SE is a SIM card, it is enriched with Java Card packages that allow interaction with the Mobile Network Operator (MNO) such as sending SMS, generating a phone call, etc.

In addition, Java Card platforms are designed in respect to Global Platform specifications [3]. In other terms, each Java Card platform is composed of isolated E²PROM areas called security domains (SD) as in Fig. 2. A SD is assigned to one service provider (SP) and may contain one or several applications belonging to the same SP. Each Java Card application is composed of one or several Java Card applets, generally called Cardlets.

As in Fig.3, each NFC device has a NFC controller that allows a contactless communication with respect to three modes:

- **Read/write mode**: the application running on the mobile device is able to read and write data on RFID tags according to an NFC Data Exchange Format defined by NFC forum [4]. Since 2010, Google proposed an API to implement this mode.

- **Peer to peer mode**: called Android Beam in Android platforms, this mode allows a tap of NFC devices together to exchange data between them.

- **Emulation card**: unlike the other modes, the emulation card mode involves the use of a SE and allows communication between the mobile and a contactless reader. In case SE is a SIM card (see Fig.3), a Single Wire Protocol (SWP) connects the SIM card with the NFC controller to perform the communication with the contactless smartcard reader infrastructure.

**B. HCE**

“Card Emulation” has been initially defined by the NFC Forum [4] and integrated as a part of the core set of NFC specifications. HCE (Host Card Emulation) allows the software emulation of a smart card-based application. Prior to December 2013, HCE was deployed firstly in Blackberry OS. Since then, the feature has also been supported by Android OS, following the launch of version 4.4, codenamed “KitKat”. It provides an API set that helps developers to control the NFC interface and send commands to NFC-enabled devices. In this case, the data is routed to the host CPU on which Android applications are running directly, instead of routing the NFC protocol frames to a secure element. HCE operation and protocols are shown in figure 4 as in [5].

Applications could therefore be developed to emulate any classic contactless smartcard application using the ISO 14443-4 standard, such as loyalty, access control, ticketing or payment applications. In addition, HCE is simply an ordinary Android application without any specific hardware or software-based security services. Security concerns relating to HCE are further compounded by industry reports indicating that Android is, by some distance, the most attacked of all mobile environments.

According to [6], HCE is good for the NFC ecosystem as a whole; it will make NFC more accessible and versatile to developers, and more familiar to end-users, increasing mass...
market adoption as a result. It is also best suited to use cases where the user's stored credentials are of low value and where stringent security requirements are not mandatory and the emulated NFC application is not based on direct implementation of a current, pre-existing card application. However, service providers evaluating HCE for payment and other high-value NFC services should proceed with caution. HCE presents a new raft of challenges and has the potential to diminish both the transaction security and the end user’s NFC service experience. HCE remains in its infancy. Until more OS providers support the HCE model, deployments will remain vulnerable to the prevailing challenges associated with global Android OS utilization.

As a software solution, HCE is considered as less secure than a physical SE. For example, the HCE cannot handle counter-measures for physical attacks. Unlike secure devices, HCE concept does not target applications with high-level security, and relies on larger resources (memory & processor) because it uses the host OS. In addition, it is independent from SE providers.
in an ordinary, non-secure computing platform and convert them into trust environments. TPM implements mechanisms and protocols to ensure that a platform has loaded its software properly. Independent from the host OS, it stores secret keys to encrypt data files/messages, to sign data, etc.

Called Trusted Computing Platform Alliance until 2003, Trusted Computing Group (TCG) is a consortium of companies: Compaq, HP, IBM, Intel, Microsoft, AMD, etc. that design TPM specifications. The TCG requires three Roots of Trust in a trusted platform: Root of Trust for Measurement (RTM), Root of Trust for Storage (RTS), and Root of Trust for Reporting (RTR). It has recently released the TPM 2.0 library specification that provides a support for additional cryptographic algorithms, enhancements to the availability of the TPM to applications, enhanced authorization and management mechanisms. According to TCG, “specifications will detail how the TPM can be implemented in various platforms through TCG platform specific specifications. These future specifications include the TPM Software Stack specification (TSS) and separate specifications for PCs, mobile, embedded and virtualized platforms”.

III. COMPARISON OF TRUSTED PLATFORMS FEATURES

In previous sections, we presented the overview of available Trusted Platforms. In this section, we go to the comparison of the different features of Trusted Platforms. Table 1 summarizes our study of the Trusted Platforms.

In general, almost security solutions for mobile computing rely primarily on SE. Nevertheless, in some cases, the solutions do not require very strict security requirements and the level of security required should be balanced to the importance of the context. In this paper, we discuss the following criteria.

A. Form and Dependency

Different from SE and TEE, HCE and TPM present only in the software and hardware form prospectively. The former is simply the android program where the user's stored credentials are of low value. The latter is a computer chip and considered as the cheapest way to enhance security in an ordinary, non-secure computing platform. In term of Dependency, TEE works as the framework for mobile-device security. It is also known as a layer between mobile OS and typical SE.

B. Hardware

SE, TPM and some implementations of TEE have their own limited hardware resource. In contrast, HCE and TEE have larger resources because they have access to the hot resources. Thus, the TEE provides a more powerful processing speed capability and greater accessible memory space than SE. Unlike the other platforms, HCE has at its disposal all the memory resources of the Android platform, but HCE is not a piece of hardware.

C. OS and API support

At the moment, HCE is not isolated from the device's operation system and supported by Android OS in version 4.4. However, in the software form, HCE is considered less secured than others.

D. Security features

In this aspect, the SE is the highest security level for the mobile computing. However, it also required extra cost for the implementation. In some case, the TEE is sufficient for most application. Moreover, the TEE can work with another trusted platform to enrich the security level. For the moment, TEE standardization is not yet finalized.

By contrast, HCE concept does not target application with high-level security. Security concerns relating to HCE are further compounded by industry report indicating that Android is, by some distance, the most attackable of all mobile environment [6].

E. Role

The role of HCE is a blur in comparison with other trusted platforms. Some drawbacks deal with HCE such as low power mode, roaming and no data connectivity scenarios and transaction speed. As stated in [6], HCE still remains in the beginning stage and needs a long way to go before it can establish a certification schema that is comprehensive and robust enough to gain the confidence of service providers and end-users.

F. Physical / Software attack

The SE’s physical protection depends on the type of chips. Most of them have strong protection. TEE, SE and TPM are designed to prevent from unauthorized software attack, work well in many use cases including e-commerce, citizen to government applications, online banking, and other fields where there is high security level required.

G. Privacy

In order to be against the increasing of the security threats, the TPM supports covering privacy protection and interoperability on multi-platforms including PCs and Mobile devices. Hardware manufacturers can flexibly implement the architecture of TPM chips. The owner can also flexibly deploy this architecture depending on the security solutions. Regarding the software implementation of available trusted platforms, HCE, supporting strongly by Google, has gained some attention in the NFC technology. With SE-less, HCE faces with many security risks in comparison to SE and also requires
more attention from the developer side. However, it enriches some factors; namely development costs, time to market and the need to cooperate with other parties.

In the next section, we investigate the different architectures of mobile cloud computing in order to show how to use the trusted platforms to secure them.

IV. MOBILE CLOUD COMPUTING ARCHITECTURES

To highlight the motivation for mobile cloud computing, different architectures in the literature [9, 10, 11, 12, 13] have been defined to cater to different use cases in mobile cloud computing. The major benefit of cloud computing for mobile devices is to enable running applications between resource-constrained devices and Internet-based Clouds. Hence, resource-constrained devices can outsource computation/communication/resource intensive operations to the cloud. Mobile cloud applications based on mobile cloud collaboration may deploy their components into different places including local smartphone, virtual machines in cloud and cloudlets. The principal motivation of offloading is to achieve less execution time and less energy consumption within mobile devices. In the following, we describe the possible architectures for mobile cloud computing.

A. Cloud computing with mobile devices

Commonly, mobile cloud computing means to prolong the capabilities of storage/computation-limited devices, and to provide seamless access to data/application on a remote resource rich server from anywhere. The network connectivity from the device to the cloud server needs to be optimized to ensure the quality of service and seamless handover.

B. Virtual cloud computing provider

Another approach [14] is to build up a cloud with peer-to-peer connected mobile devices for data storage and processing so that mobile devices are resource providers of a virtual cloud. Through Hyrax platform, the author in [15] demonstrates the possibility of building a cloud with mobile devices such as Android smartphones to provide basic functions and services.

C. Cloudlets as Intermediate Offload Elements

Similar to the concept of hotspots, the cloudlet concept proposed by Satyanarayanan [16] is another approach to mobile cloud computing. Cloudlet approach is an alternative solution to remote cloud server which would present latency and bandwidth issues. Indeed, a cloudlet is viewed as a multi-core computer installed in the public infrastructure with connectivity to remote cloud servers. Hence, the cloudlet is used by the mobile device to offload its workload while ensuring low delay and high bandwidth.

Simanta et al. in [17] proposed an example of implementation of the concept of cloudlets with the objective of enhancing processing and conserving battery power in the mobile devices notably in hostile environments where networks are unreliable. The key feature of the architecture is that offload elements are stateless. Communication between the cloudlet and the cloud core is done only during setup and provisioning. Once the cloudlet is provisioned, it works in a disconnection mode with the core and in a connection mode with the mobile device. When a mobile device is connected to a cloudlet, an application overlay is offloaded from the mobile device to the cloudlet. An application overlay represents the difference between a base VM with only an operating system installed and the same VM with the application installed. The cloudlet may host multiple VMs.

V. SECURITY ISSUES IN MOBILE CLOUD COMPUTING

MMC not only takes advantage of the cloud computing facilities but also inherits the security threats of conventional cloud computing. Using the cloud for mobile devices raises a challenge of security and trust issues. The current challenges of MCC can be listed here such as malware, virus, privacy, financial fraud, content protection, enterprise data and secure space. Due to this matter, there is a rise of awareness and an acceleration of adoption of security solutions. In this section, to address the role of Trusted Platforms in MMC, we discuss two security concerns such as Security of the mobile devices and Security of the Cloud.

A. Securing the mobile devices

In terms of Security for mobile users, the available Trusted Platforms have played their security role well in mobile device environment. In addition, these trusted platforms can work together to meet the security requirement. For example, TEE can serve as a complementary security countermeasure and integrating nucleus for service that depends on partial identities distributed across SEs, such as Active Stickers, Secure Micro SD Cards, UICCs, and Embedded SEs. The TEE combines these into integrated solutions which assure seamless interaction and security of processes that are executed in the periphery of the respective [3]. The combination of TEE and TPM Mobile is another typical example. Because of the advantage in on-processor technologies such as the TEE combining with the flexibility of the TPM protocols, this means it is possible to implement the TMP as an integrated solution or in firmware. As a result, the TEP Mobile functionality is implemented as a Trusted Application in the TEE.

Securing mobile devices means to secure mobile apps and data. Recent work has been devoted to design trust environment for mobile devices as in the following.

- Security of mobile apps

Since apps are executed in untrusted mobile devices, trusted platforms are essential to overcome security constraints. Depending on the security level that is required, apps may rely on physical or software trusted platforms. For example, Roland developed, in [18], a prototype that emulates secure elements within Android platforms. Such an open emulator is used for debugging and rapid prototyping of secure element applications. The emulator is a complete Java Card API implemented within Android. Comparable to HCE of Google, the emulated secure element allows security and trust, and is proposed to replace the physical SE only for long-term testing and for showcasing of applications.
- **Security of Data**

Confidentiality and integrity are required to secure data stored on the mobile device and to allow transfer through the Internet. Well-established protocols such as SSL/TLS should be used for data transfer while confidentiality and integrity may be computed by the trusted platforms of the mobile device since they integrate a crypto-processor with trust storage, and implement cryptographic algorithms. Depending on the data sensitivity, data may be encrypted or not and may be stored in a distinct storage memory. In fact, if the data is sensitive like a PIN code, a generated security key, or a password, it can be stored in the Flash memory or the E\textsuperscript{2}PROM of the trusted platform (SE or TEE). If the data size exceeds the memory available in trusted platforms, data can be stored in the mobile device memory that is protected by the trusted platform like in TEE, or even outsourced if the Cloud provides a trusted environment. In case data is not sensitive, it can be stored in the untrusted mobile device or on the Cloud.

B. **Securing the Cloud**

- **Security of the apps and VMs**

To secure apps, the authors, in [19], propose the concept of a Cloud of Secure Elements (CoSE) where the secure services are hosted by servers rather than by smartphone Secure Elements. They discuss the use of CoSE for mobile payments, and illustrate how an NFC smartphone may be efficiently used as a bridge between an NFC reader and an Internet server of secure microcontroller that hosts EMV applications. The advantage of this solution is to assign dynamically a physical secure element on the cloud for a sensitive application.

In cloud computing, the service provider must achieve auditability, for attesting whether security setting of applications has been tampered or not. As stated in [20], auditability can be achieved using remote attestation techniques that require a trusted platform module (TPM). The TPM contains an endorsement private key that uniquely identifies the TPM. At boot time, the host computes its system state (boot sequence, namely the BIOS, the bootloader, and the software implementing the platform) as a sequence of hashes securely stored within the TPM. A remote party can use this sequence of hashes to authenticate the host and to trust the host platform.

In traditional systems, the operating system is linked to the hardware machine regarding the attestion process. However, in a virtualized environment like the clouds, VMs can dynamically migrate from one location to another, making the remote attestation mechanism not sufficient. Santos et al. [21] propose to secure the hardware layer using hardware TPM, and secure VMs using a secure virtual machine monitors. VM migration is possible if both source and destination servers are trusted. In [21, 22], the authors designed efficient protocols for trust establishment and management.

Patidar et al., in [23], propose a system in which cloud computing system is combined with trusted computing platform using a TPM. Security mechanisms like authentication, confidentiality and integrity are provided to illustrate the TPM use.

- **Security of Data**

Many authors propose security mechanisms for Cloud storage. For example, Hsueh et al. [24] proposed hybrid cryptography, and digital signatures to provide security requirements for data storage of mobile phones. Their solution is used to avoid malicious attackers from illegal access and to share desired information with targeted friends by distinct access rights. Unlike this kind of solution, the hardware-based encryption can take advantage of the TPM and does not require user intervention or impact system performance. For instance, [25] have published a Trusted Storage specification to provide a manageable, enterprise-wide means for implementing full-disk encryption using TPM. This trusted storage notion may be extended to target the storage provided by the data centres of the Cloud.

In terms of digital right management (DRM), Zou et al. [26] propose Phosphor, a cloud-based mobile DRM scheme with a subscriber identity module (SIM) card in mobile phone, to avoid illegal distribution and piracy of digital contents (like video, audio, e-book, etc.). The SIM card hosts a Licence State Word (LSW) protocol that decrypts each digital content received from the Cloud using the decryption key stored on the SIM card. The drawback of this solution is the sharing of a secret key between the Cloud and the SIM card. In other words, the Cloud has to maintain a key for each user.

C. **Authentication**

In the cloud environment, authentication and access control are more crucial than ever since the cloud services and the cloud storage may be accessible to anyone over the Internet. The TPM can easily provide stronger authentication than username and password. In [27], the authors present a protocol to authenticate a nearby Cloudlet by a mobile device that handles NFC applications. For authentication, the mobile device uses a SE and involves the Mobile Network Operator and a trusted service management that belongs to a Cloud computing.

D. **Data privacy**

In addition to data protection as presented above, privacy is a means to allow the user having a complete control on his personal data hosted on the Cloud computing. In some MCC frameworks, like in MobiCloud, Huang et al. [28] introduce the notion of Extended Semi-Shadow Image (ESSI) that is not exactly the same as a virtual image because an ESSI could be an exact clone, a partial clone, or merely an image that has extended functions of the physical device. CloneCloud [29] is another MCC platform that uses offloading technique to perform computing on a resourceful server. For this purpose, the authors use device clones, and assume that the clone VM is by default a trusted one, and mention establishing trust as future work. The drawback of the MobiCloud and CloneCloud platforms that rely on device clones on the Cloud, is that no privacy means is provided to the user to let him choose the personal data to outsource or not on the device clone. If a TEE
The personal data that the user does not want to outsource can be stored in a secure manner by the TEE that has larger memory storage than a SE.

E. Trusted platforms based MCC architectures

In this paragraph, we investigate the use of trusted platforms regarding each MCC model. In the virtual cloud that is based on peer-to-peer mobile devices, the trusted environment can be built on the trusted platforms (like SE, TEE and HCE) of the mobile devices. Depending on the required security level, the trust can rely on SE, SE combined with TEE, or HCE. These trusted platforms can be used to guarantee the integrity and the confidentiality of the information exchanged between the devices and can be used by the mobile devices for authentication and trust.

In the Cloud server model, the trust environment can be built using the trusted platforms (SE/TEE) of the mobile device combined with trusted platforms on the Cloud such as the Cloud of SE (CoSE) proposed by [19]. However, this kind of trusted environment may rely on a global security strategy so that the involved trusted platforms hosted on the mobile and the Cloud may interact properly to achieve the same goal.

In the Cloudlet architecture, we need to design two trust environments: one trust environment based on the Cloudlet and the Cloud and a second environment based on the mobile device and the Cloudlet. Using a TPM on the Cloudlet and SE/TEE on the mobile device allows implementing different security mechanisms and attesting the authentication of the different involved platforms. The TPM of the Cloudlet can also contribute to define a trust environment with the Cloud. Regarding the service model of cloud computing, trusted platforms like TPMS can be used to ensure trusted VM launch in an untrusted IaaS environment like in [30].

VI. CONCLUSION

There is considerable consensus that the trusted platforms are helpful in Cloud security. In this paper, we have presented several Trusted Platforms architectures and described how the MCC security gains the support from the Trusted Platforms. By introducing the overview architecture of several available Trusted Platforms and Mobile Cloud Computing and making the comparison of these Trusted Platforms, we have presented our approach which is to build security and trust in this paper. The former focuses on either the security of mobile devices or the security on cloud enabled by trusted platforms. The latter concentrates to distributed trust environment for mobile devices and Cloud side including Cloud server model and Cloudlet model. We plan to design a complete trusted environment based on trusted platforms for mobile cloud computing and implement a fully functional prototype based on our design and evaluate its performance in the near future.

REFERENCES


Table 1. Comparison of features of trusted platforms

<table>
<thead>
<tr>
<th>Criteria</th>
<th>SE</th>
<th>TEE</th>
<th>TPM</th>
<th>HCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>Comprised of software and tamper resistant hardware</td>
<td>Made up of software and hardware</td>
<td>Hardware-Embedded in a host platform</td>
<td>Software-Android Applications</td>
</tr>
<tr>
<td>Dependency</td>
<td>Autonomous entity (like a little PC)</td>
<td>a secure area that resides in the main processor mobile devices</td>
<td>Bounded with the host platform</td>
<td>User defined (I don’t know if it is user defined or OS Library defined)</td>
</tr>
<tr>
<td>Hardware</td>
<td>Processor, crypto-processor, EEPROM, RAM</td>
<td>Processor, crypto-processor, EEPROM, RAM. The memories may be shared with the host device</td>
<td>crypto-processor, EEPROM, RAM</td>
<td>Not available</td>
</tr>
<tr>
<td>OS</td>
<td>Within the chip</td>
<td>Separate and run in parallel with the host OS</td>
<td>Bootstrap in the TPM, OS in the host platform</td>
<td>Operate in the host OS</td>
</tr>
<tr>
<td>API support</td>
<td>Limited</td>
<td>Limited</td>
<td>Limited</td>
<td>Rich API support</td>
</tr>
<tr>
<td>Security features</td>
<td>Stores sensitive data (PIN code, IMSI), secret keys, generate session keys</td>
<td>Provides a framework for security within the device</td>
<td>Trust root, generates asymmetric key pair, binding, identity attestation, protected objects</td>
<td>Security is dependent on device OS</td>
</tr>
<tr>
<td>Role</td>
<td>Used for identification/authentication, for data encryption during communication</td>
<td>Used for robust, hardware-backed, scalable-consistent, OS-independent security.</td>
<td>Used to secure the host platform, to remote communication</td>
<td>Used for the software emulation of a smart card-based application</td>
</tr>
<tr>
<td>Physical attacks</td>
<td>Many studies (the chip is endowed with counter-measures)</td>
<td>Depending on hardware features of hosting platform</td>
<td>No study</td>
<td>No counter-measures for physical attacks</td>
</tr>
<tr>
<td>Software attacks</td>
<td>Many studies (the chip is endowed with counter-measures)</td>
<td>Resistant to software attacks</td>
<td>Resistant to software attacks</td>
<td>Low resistance to software attacks</td>
</tr>
<tr>
<td>Privacy</td>
<td>Controlled by the user</td>
<td>Controlled by the user</td>
<td>No control by the user</td>
<td>Controlled by the user</td>
</tr>
</tbody>
</table>