# **Trusted Platform Module**

[Trusted Platform Module 1](#_Toc149229956)

[Introduction 1](#_Toc1848845784)

[Brief history of TPM 2](#_Toc84441581)

[Current Implementations of TPM and Raspberry Pi 2](#_Toc1034117329)

[LetsTrust TPM 2](#_Toc935765343)

[Optiga™ IC 2](#_Toc102183069)

[TPM Architecture and Functionalities: 4](#_Toc1621461007)

[How a Raspberry Pi can utilise TPM 5](#_Toc1130459338)

[Known Security Issues in TPM Architecture 6](#_Toc933387842)

[Recommendations for Redback 7](#_Toc1163367985)

[Recommend the use of Infineon Optiga TPM SLB 9670 based TPM modules for Raspberry Pi. 7](#_Toc742336199)

[Recommend use of Raspberry Pi 4 for all projects for better support of TPM 7](#_Toc1630632520)

[Recommend use of Linux Kernel 4.19.50 & TPM 2.0 7](#_Toc527991730)

[Recommend use of LUKS open-source disk encryption specification for encrypting all storage drives including SD cards 8](#_Toc1717420889)

[Keys for encryption and decryption to be stored on TPM while using LUKS. 8](#_Toc1651021420)

[Conclusion 8](#_Toc1067663694)

[References 9](#_Toc962031583)

## **Introduction**

Trusted Platform Module (TPM) is a small piece of hardware that has many functions to increase security of a device. TPM is commonly found built into modern processors from Intel and AMD but can also be added on to supported operating systems through an add-in board into a dedicated slot [1]. It’s also an evolving standard that both industry [2] and international [3] standard specifications with wide international support.

More specifically, the physical TPM device is an inexpensive but powerful and flexible, cryptographic processor. It can create, manage, and use cryptographic keys, as well as store confidential data. It is closely tied to how a computer boots and runs, which means it is far more powerful and useful than a simple “smart-card on the motherboard” or hardware security module (HSM) which it’s often compared to [4].

In practical terms, platforms that utilise TPMs, “measure” and log the software that boots on the device. This boot-log can be used to verify that the platform is running verified, up to date software using a TPM feature called *attestation* or *quoting*. This boot-log can also be used to protect keys in disk encryption. This is a feature called *sealing*, it can be used to make sure that the encryption key is only disclosed to authorized software [5].

Other, more advanced TPM features include monotonic counters, secure clock, a non-volatile storage facility, and secure mechanisms for key management operations when importing and exporting keys [4].

## **Brief history of TPM**

TPM was first deployed back in 2003 as TPM 1.1b. This specification included only very introductory features compared to what is available in TPM 2.0. However, this version was still able to generate RSA keys, provide secure storage, authorisation, and attest to the health of a device [6].

The TPM 1.2 revision only offered small improvements in terms of features over TPM 1.1b which included a standardisation of the pin layout and software used to make utilising the TPM easier as this prevented vendor locking. TPM 1.2 also included preventative measures against dictionary attacks and a small amount of space located on the chip itself for storing the certificate used for validating the endorsement key [6]. The endorsement key is a hardware encoded encryption key that is typically embedded at the time of manufacturing the TPM [7]. TPM 1.2 also utilised a stronger encryption algorithm at the time of creation being SHA-1. This has since been changed in TPM 2.0 using numerous encryption algorithms including asymmetric, symmetric, and hashing algorithms.

The current revision of TPM is TPM 2.0 which is what this document will be focusing on.

## **Current Implementations of TPM and Raspberry Pi**

### **LetsTrust TPM**

LetsTrust TPM is a simple to utilise TPM solution for the Raspberry Pi. A hardware TPM can be utilised dynamically, including for confirmation/marks, putting away crypto keys and significantly more. The module can likewise be utilised as a True Hardware Random Number Generator (TRNG) on the off chance that you really want a decent wellspring of irregularity.

LetsTrust TPM utilises the SPI interface to have a connection with the Raspberry Pi [8]. It is compatible with all Raspberry Pi models - Model B, Model A, Pi Zero and Pi400 (with cord). It has a compact footprint which permits the excess GPIO pins to be utilised.

The Review for the product is between 3-4 stars as the product make/design scripts does not work (missing a few conditions) even with the use of Bullseye 32bit delivery on a Raspberry 4. The unit works - have the option to utilise the basic testing/indicative library and directions to show this [9]. Currently- for non-progressed clients at any rate - these isn't a lot of you can do on the unit - apart from creating irregular numbers, check registers and so on.

### **Optiga™ IC**

The OPTIGA™ IC purposes a SPI interface as per the TCG specifications. Infineon gives driver programming to a straightforward variation to any standard microcontroller SPI interface [10].

The TPM is a protected regulator with added cryptographic usefulness as shown in [11]:

1. Top of the line security regulator with cutting edge cryptographic calculations carried out in equipment (for example) RSA-2048, ECC-256, SHA-256)
2. Normal standards (EAL4+) security affirmation
3. Adaptable combination with SPI interface support
4. Stretched out temperature range (- 40 to +105°C) for various applications
5. Simple to incorporate with wide reach open-source help
6. One of a kind key that recognizes each TPM

Review for the product by road testers points out that it has various enemy of alter components, and some portion of the memory can't be recovered externally. That regulator is modified with firmware that executes the TPM API [12].

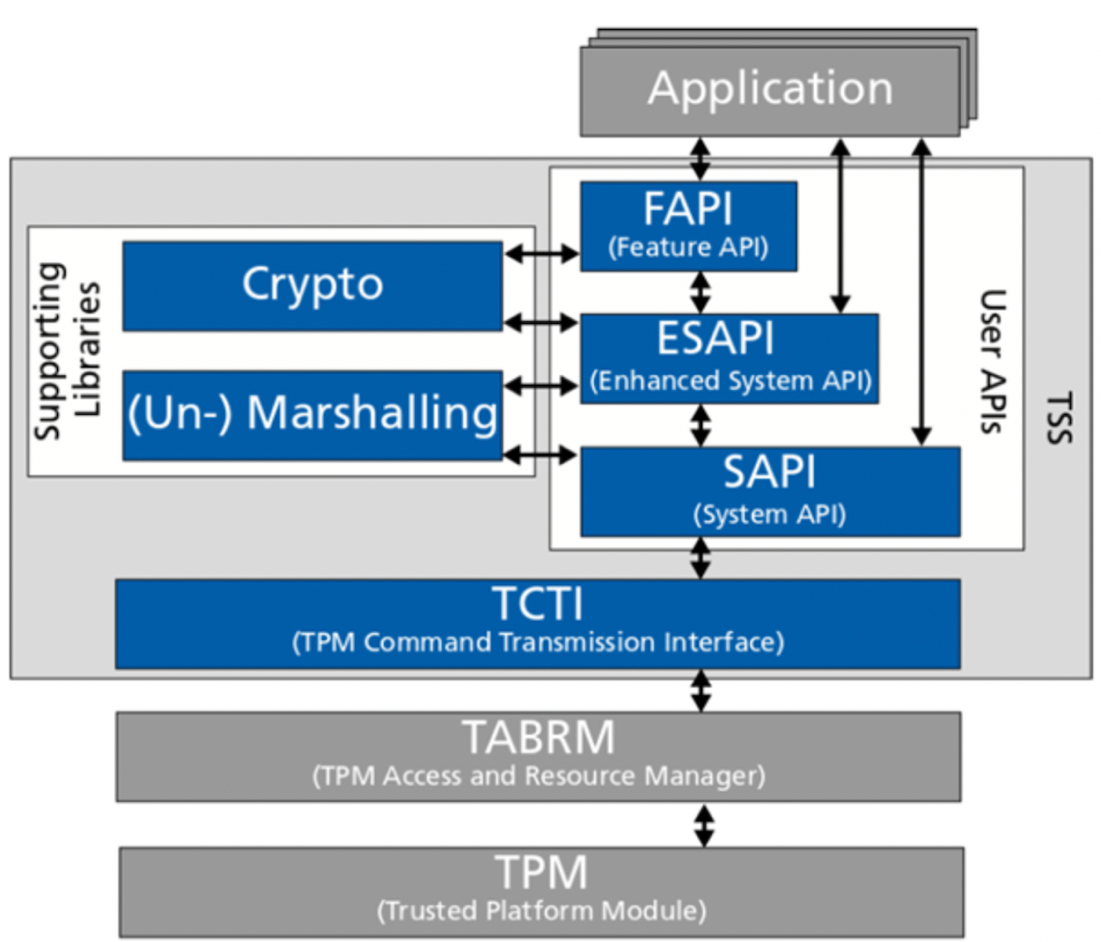
There is no known method for getting the root certificates out of the IC. The firmware API doesn't uncover it, when you attempt to open the bundle, it obliterates the substance. Measures are taken that you can't get from power profile or execution time that you have a critical match or mismatch [13].

At the point when the IC recognises that brute force endeavours are used, it makes itself unusable for some time.

Fundamental capabilities and utilizations as shown in [14]:

* Key supplier, store and approval
* Sign and encode/decode
* Irregular number generator
* Equipment Authenticator
* Equipment bound permit the executives, distinguish a gadget
* guaranteed occasion logging with timestamp

When the OS is set up, the following step is to construct the TSS otherwise called Trusted Software Stack, Software libraries and utilities. The diagram is readable from base to top which shows the picture what will be introduced when you follow the integration of Open Source TPM Software Stack 2.0 on a Raspberry Pi3 Linux environment with Optiga™ IC

[15]

## **TPM Architecture and Functionalities:**

The raspberry pi utilizes TPM for a variety of tasks such as the storage of cryptographic keys, generation of random numbers, and for the authorization and authentication of signatures.

I/O block is used to manage the flow of data among the communication bus. The data transmitted between internal and external buses is encrypted and decrypted by the I/O block and then it is pointed to the appropriate TPM component. Non-Volatile storage is used to store the platform keys. The endorsement key (EK) is stored in non-volatile memory [16].

Attestation Identity Key (AIK) is a public and private key pair that is used as an alternative for the Endorsement key (EK) as an identity. The private part is only used inside the TPM for limited TPM-defined operations such as signing. It is a part of a versatile memory [17].

Platform Configuration Registers (PCR) is one of the most important and essential features of TPM. PCRs are utilized to record the software state including the software running on a platform and its configuration data in a cryptographic manner. It uses a one-way hash method which makes the record irremovable. It is also a part of versatile memory. PCRs can also be used to restrict the utilization of other objects [18].

Most of the code is written (programmed) in the subset of the C programming language. TPM programming interface is elaborated in machine-readable tables. Open-source libraries and solutions are available which allow the development of TPM-based applications [19].

The TPM has an execution engine that runs program code and responds to external commands by choosing and running the necessary program code on the TPM. TPM includes a random number generator to guarantee the application's security. To boost entropy, random bit streams are utilized to seed a random number generator. The Secure Hash Algorithm SHA-1 is implemented by the SHA-1 message digest engine. This algorithm creates a 20-byte digest after hashing the input data. It also serves as the foundation for a Hash-Based Message Authentication Code engine and is utilized by the TPM in a number of cryptographic operations. The RSA key generation technique requires the generation of keys, which can be computationally demanding. TPM uses these keys often for authentication and safe storage, the standard requires that the TPM include a module expressly for this function. A TPM must be able to support keys with a 2048-bit modulus in order to comply with the specification. Additionally, some keys used with the TPM must have a modulus of at least 2048 bits. To perform the RSA algorithms for authentication, encryption, and decryption, TPM has a dedicated RSA engine [20].

There are several permanent and volatile flags that control the status of the TPM. If the TPM owner is present, his consent is required before these flags can be changed. Depending on the initial state in which the device was supplied, the TPM will transition through a number of stages during the process of gaining ownership. The Opt-In component's purpose is to offer safety and mechanisms for maintaining the TPM state by monitoring the status of these flags [21].

## **How a Raspberry Pi can utilise TPM**

There are numerous readily available TPM 2.0 add-in boards that work with Raspberry Pi, each offering a wide range of functionality. These include:

* Ability to store cryptographic keys
  + These keys can range from keys stored for secure applications to the key required to unlock the Raspberry Pi when it turns on. The TPM contains a key called the storage root key and it uses this key for encrypting and decrypting the keys used by various applications. These keys can be created under certain conditions and can only be used when these same conditions are met or can be created by the TPM alone. Both methods only allow the storage root key to be kept on the TPM itself. The process of encrypting the keys is called sealing and the process of decrypting is called unsealing [22].
* Authentication of hardware
  + When a specific hardware configuration is set, the TPM will check to make sure that this configuration is correct each time the Raspberry Pi is started. If this configuration has been changed, then the Raspberry Pi will not be considered trustworthy and may prevent certain functionality of the Raspberry Pi [23].
* Digitally sign and encrypt/decrypt drives
  + Having the ability to digitally sign the Raspberry Pi means that the platform can be attested, validating that the device has not been tampered with. Once this authentication process has been successfully undertaken, the device is then trusted and able to do whatever Redback Operations tasks it to do [24]. A TPM cannot encrypt and decrypt drives but can create and store the keys used for said encryption and decryption. When the correct conditions are met and trustworthiness of a system can be attested to, keys can be unsealed and released to the system for decryption of data while sealed keys can be used for encrypting data [22].
* Generate random numbers
  + The TPM can take a range of various inputs, from environment to physical inputs, in order to create random numbers that have such a high level of entropy [25]. Utilising these random numbers is essential when generating keys as it makes it extremely difficult for an attacker to brute force since the entropy is not defined by software which can be easily guessed [26].
* Hardware bound licence management
  + TPM does have the capacity to store licences from software on the TPM itself. Storing the licences here makes them only accessible to the specific program and prevents any sort of plagiarism or hacking of any kind as it is not generally accessible [27].
* Logging of events
  + The only operation that will be logged is the PCR Extend operation [28]. The Extend operation occurs when the current value is extended with a new value in accordance with the following equation:
    - PCR[N] = HASHalg( PCR[N] || ArgumentOfExtend ) [29]
  + This log is used by external entities for attestation purposes. Logs are reconstructed and compared against known values by external entities for remote attestation purposes [28].
* Integrate with networks for verification and trust of the device
  + Once hardware has been authenticated, meaning that it has not been tampered with and is in full working order as required, this can allow for specific functions to take place. Functions such as starting secure applications and connections that should not be accessible by any computer other than certified Raspberry Pi’s [30].

## **Known Security Issues in TPM Architecture**

A trusted platform module has a number of known but easily mitigated security vulnerabilities. The reason an attacker would attempt to exploit a TPM-based machine is to acquire the resources on the victim’s encrypted data drives.

TPM is known for implementing a multi-layered security protocol to make it more resilient to any sort of cyberattack. As these modules are assigned an RSA-based encryption key at the time of bootup, the initial bootup of a device that has TPM installed in it would securely function and communicate with the hardware and the OS once the device has started. The TPM would only protect the device if it has been powered on. Therefore, if the device is not able to be powered on or the user has not been logged in, there is no device security as the security protocol operates at a kernel level of the OS and would not run if the user session is not initiated. This process can be considered a vulnerability if the device is stuck in a boot loop, leaving the product vulnerable to potential attacks. To resolve this issue, a connected standby protocol needs to be implemented that would allow the governance of the software and hardware-based protocols present in the device [31].

A reported vulnerability found in the TPM architecture is these TPM chips can make automated DNS queries. TPM software patches and updates are automatically enabled, meaning the device can make automatic a DNS resolution request. Considering this can be easily poisoned, the attacker can direct the TPM request to any malicious pages. This means that TPM chips could potentially download malicious files by accessing the internet and enabling the propagation of the virus on the host machine [32].

A protocol vulnerability risk that is commonly found in TPM architecture is that the owners that have this architecture installed on their machines are relatively incapable of mitigating security risks. An attacker can easily take advantage of the strong encryption methods offered by TPM to hide the malware, cloaking itself as legitimate software on the encrypted hard disk. The malware cloaking is an effective technique as the malware is hidden under the layers of security offered by this architecture making it difficult for the users to detect the malicious code. As the TPM is not accessible easily, it further increases the difficulty of scanning and checking for viruses and other forms of malware [33].

A security vulnerability that existed in the TPM chipset for a long time now is the key strength that is automatically weakened on the devices that have BitLocker installed. BitLocker uses the seal and unseal features from the TPM architecture to protect the secrets that are present in the operating system space. Prior versions of TPM were mostly affected, but this vulnerability has since been patched. This vulnerability resulted in a weakened key that could easily be decrypted and effectively reducing the time required to bypass the disk encryption [34].

## **Recommendations for Redback**

### **Recommend the use of Infineon Optiga TPM SLB 9670 based TPM modules for Raspberry Pi.**

The Infineon Optiga TPM SLB 9670 chipset is inexpensive and widely used in TPM modules design to work with Raspberry Pi devices. These modules are readily available for a retail cost of between ~$20 AUD to ~$60 AUD with a much lower wholesale price expected. Given the solid documentation available for the chipset online [35], its wide use and relatively low cost, we believe this TPM 2.0 enabled chipset will be ideal for Redback and its IoT projects. The wide use of the product also means that any post sale security flaw or bug will be attended to by the manufacturer promptly with firmware or driver updates. The “SLB” version of this chipset is specified for standard security applications while other versions of the chip exist for automotive and industrial security applications.

### **Recommend use of Raspberry Pi 4 for all projects for better support of TPM**

Given the first recommendation to use TPM modules with the Optiga SLB 9670, we further recommend the use of Raspberry Pi 4 for all projects where Raspberry Pi is a good fit. This is based on confirmed information from the manufacturer, Infineon, with respect to integrated TPM 2.0 driver support on the Raspberry Pi 4 and simplified TPM set up [35].

### **Recommend use of Linux Kernel 4.19.50 & TPM 2.0**

According to documentation from Infineon, the Linux Kernel 4.19.50 has integrated TPM 2.0 driver support which will simplify TPM set up on the Raspberry Pi4 [35]. TPM 2.0 is the latest version of TPM released in 2014 with the last revision in 2019. TPM 2.0 uses SHA-1 and SHA-256 algorithms for hashing, better public key cryptography using Barreto-Naehrig 256-bit curve and NIST P-256 and HMAC with 128-bit AES symmetric-key algorithms [36]. This makes TPM 2.0 far more secure than its predecessors and we do not recommend using TPM versions below 2.0.

### **Recommend use of LUKS open-source disk encryption specification for encrypting all storage drives including SD cards**

The Linux Unified Key Setup (LUKS) is recommended for encrypting all storage devices including SD cards on the Raspberry Pi devices. The LUKS disk encryption specification is open source and platform independent, which allows a standardized, on-disk encryption format that can be used across different programs [37]. LUKS is also well documented, and a knowledge of LUKS will allow Redback staff to work on encryption projects with Raspberry Pi as well as other Linux based systems. Open source and free guides are available on using TPM 2.0 modules with the LUKS framework [38] and on how LUKS can be used with Raspberry PI for the purpose of encrypting SD cards [39].

### **Keys for encryption and decryption to be stored on TPM while using LUKS.**

While encryption using LUKS doesn’t necessarily need a TPM for decryption and encryption functions [40], we recommend the use of a TPM 2.0 for key storage as a best practice across all Redback IoT operations with Raspberry Pi. We further recommend the use of TPM 2.0 on projects that involved any Linux based system so long as a TPM can be feasibly used for holding encryption keys.

## **Conclusion**

A Trusted Platform Module device and its associated standards are a worthwhile investment for Redback Operations despite some known security issues.

These include:

* Increased exposure if the device is stuck in a boot loop
* TPM chips can send automated DNS queries – useful in DDOS attacks.
* The TPM protocol isn’t easily accessed or modified. Malware that has bypassed a TPM will find it very easy to hide.

To mitigate these few known issues, we have made some recommendations for Redback to implement TPM in a safe and secure way. These include the use of:

* A connected standby protocol needs to be implemented to allow for governance if stuck in a boot loop.
* The use of Infineon Optiga TPM SLP 9670 based modules
* Raspberry Pi 4 wherever appropriate
* Linux Kernel 4.19.50 & TPM 2.0 for the latest, most secure versions of each
* LUKS open-source disk encryption specification for encrypting all storage drives including SD cards. The encryption and decryption keys should be stored on the TPM device itself.

## **References**

[1] W. Arthur, D. Challener and K. Goldman, “History of the TPM”, *A practical guide to TPM 2.0*, 1st ed. Apress open, 2015, p. 1.

[2] "TPM 2.0 Library | Trusted Computing Group", *Trusted Computing Group*, 2015. [Online]. Available: <https://trustedcomputinggroup.org/resource/tpm-library-specification/>.

[3] "Information technology — Trusted platform module library — Part 1: Architecture", *ISO/IEC 11889*, vol. 1, 2022. Available: <https://www.iso.org/standard/66510.html>.

[4] "Trusted Platform Module (TPM)", *microsoft.com*, 2022. [Online]. Available: <https://www.microsoft.com/en-us/research/project/the-trusted-platform-module-tpm/>.

[5] "Trusted Platform Module (TPM) fundamentals (Windows) - Windows security", *Docs.microsoft.com*, 2022. [Online]. Available: <https://docs.microsoft.com/en-us/windows/security/information-protection/tpm/tpm-fundamentals>.

[6] W. Arthur, D. Challener and K. Goldman, "History of the TPM", *A Practical Guide to TPM 2.0*, pp. 2-5, 2015. Available: <https://link.springer.com/content/pdf/10.1007/978-1-4302-6584-9.pdf>. [Accessed 16 August 2022].

[7] Microsoft, "TpmAttestation interface", *Microsoft Docs*. [Online]. Available: <https://docs.microsoft.com/en-us/javascript/api/azure-iot-provisioning-service/tpmattestation?view=azure-node-latest>. [Accessed: 16- Aug- 2022].

[8] P. Kissinger, "LetsTrust", *Letstrust.de*, 2022. [Online]. Available: <https://letstrust.de>. [Accessed: 18- Aug- 2022].

[9]John. P, "LetsTrust TPM for Raspberry Pi", *The Pi Hut*, 2021. [Online]. Available: <https://thepihut.com/products/letstrust-tpm-for-raspberry-pi>. [Accessed: 18- Aug- 2022].

[10] I. AG, "OPTIGA™ embedded security solutions - Infineon Technologies", *Infineon.com*, 2022. [Online]. Available: <https://www.infineon.com/cms/en/product/security-smart-card-solutions/optiga-embedded-security-solutions/>. [Accessed: 19- Aug- 2022].

[11] “LetsTrust TPM for Raspberry Pi,” *buyzero.de*. [Online]. Available: <https://buyzero.de/products/letstrust-hardware-tpm-trusted-platform-module>. [Accessed: 20-Aug-2022].

[12]Jan**.** Cumps, 2021. [Online]. Available: <https://community.element14.com/products/roadtest/rt/roadtests/529/infineon_trust_platf#pifragment-4100=4&pifragment-4106=9>. [Accessed: 19- Aug- 2022].

[13] “OPTIGATM TPM – security forum,” *Infineon.com*. [Online]. Available: <https://community.infineon.com/t5/OPTIGA-TPM/bd-p/OptigaTPM?_ga=2.209230994.1697558163.1661643991-1488785146.1661643991&intc=CSS_Cmty_Bnr>. [Accessed: 20-Aug-2022].

[14]“Infineon trust platform module + raspberry pi 3 B - element14 community,” *Element14.com*. [Online]. Available: <https://community.element14.com/products/roadtest/rt/roadtests/529/infineon_trust_platf>. [Accessed: 20-Aug-2022].

[15] *Farnell.com*. [Online]. Available: <https://www.farnell.com/datasheets/2869244.pdf>. [Accessed: 23-Aug-2022].

[16] M. Huerrera, "incibe-cert," 2015. [Online]. Available: <https://www.incibe-cert.es/en/blog/tpm-en>. [Accessed 13 8 2022].

[17] J. D. Clercq, "Trusted Platform Module (TPM) Key Attestation," itprotoday, 19 Mar 2015. [Online]. Available: <https://www.itprotoday.com/mobile-management-and-security/trusted-platform-module-tpm-key-attestation>. [Accessed 22 8 2022].

[18] V. Bael, "TPM," 2006. [Online]. Available: <https://www.webopedia.com/definitions/trusted-platform-module/>. [Accessed 13 8 2022].

[19] A. Tomlinson, "Introduction to the TPM," in Smart Cards, Tokens, Security and Application, Boston, Springer, 2008, pp. 155-167.

[20] B. S. R. M. I. L. Shohreh Hosseinzadeh, "Recent trends in applying TPM to cloud computing," Cyber Trust research program; DIMECC , Finland, 2019.

[21] M. D. James, "A Reconfigurable Trusted Platform Module," Brigham Young University, Brigham, 2017.

[22] Dansimp et al., "TPM Fundamentals", *Microsoft Docs*, 2022. [Online]. Available: <https://docs.microsoft.com/en-us/windows/security/information-protection/tpm/tpm-fundamentals>. [Accessed: 23- Aug- 2022].

[23] Trusted Computing Group, "Trusted Platform Module (TPM) Summary", *Trusted Computing Group*. [Online]. Available: <https://trustedcomputinggroup.org/resource/trusted-platform-module-tpm-summary/>. [Accessed: 21- Aug- 2022].

[24] Gilles, "TPM and Remote attestation", *Cryptography Stack Exchange*, 2018. [Online]. Available: <https://crypto.stackexchange.com/questions/59965/tpm-and-remote-attestation>. [Accessed: 22- Aug- 2022].

[25] tomoveu, "tpm.dev.tutorials/Random\_Number\_Generator/README.md", *GitHub*, 2021. [Online]. Available: <https://github.com/tpm2dev/tpm.dev.tutorials/blob/master/Random_Number_Generator/README.md>. [Accessed: 21- Aug- 2022].

[26] Ebrary.net, "Random Number Generator", *Ebrary.net*. [Online]. Available: <https://ebrary.net/24721/computer_science/random_number_generator>. [Accessed: 21- Aug- 2022].

[27] D. Previtali, "License Management with TPM powered with CodeMeter", *Trusted Computing Group*, 2018. [Online]. Available: <https://develop.trustedcomputinggroup.org/2018/03/20/license-management-with-tpm-powered-with-codemeter/>. [Accessed: 22- Aug- 2022].

[28] IBM, "IBM Documentation", *IBM*, 2021. [Online]. Available: <https://www.ibm.com/docs/en/power9?topic=POWER9/p9ia9/p9ia9_tpm_event_logs.htm>. [Accessed: 21- Aug- 2022].

[29] Dansimp et al., "Understanding PCR banks on TPM 2.0 devices", *Microsoft Docs*, 2022. [Online]. Available: <https://docs.microsoft.com/en-us/windows/security/information-protection/tpm/switch-pcr-banks-on-tpm-2-0-devices>. [Accessed: 21- Aug- 2022].

[30] J. Cumps, "Infineon Trust Platform Module + Raspberry Pi 3 B - Review", *element14*, 2021. [Online]. Available: <https://community.element14.com/products/roadtest/rv/roadtest_reviews/1514/infineon_trust_platf>. [Accessed: 21- Aug- 2022].

[31] B. Parno, "Bootstrapping Trust in a “Trusted” Platform," Carnegie Mellon University, Pittsburgh, 2018.

[32] D. R. Omar, "How Trusted Platform Module (TPM) Is Used Today," 2017. [Online]. Available: <https://www.linkedin.com/pulse/how-trusted-platform-module-tpm-used-today-sbeit-black-belt/>. [Accessed 13 8 2017].

[33] D. GOODIN, "Trusted platform module security defeated in 30 minutes, no soldering required," 2021. [Online]. Available: <https://arstechnica.com/gadgets/2021/08/how-to-go-from-stolen-pc-to-network-intrusion-in-30-minutes/>. [Accessed 13 8 2022].

[34] , A. Louca, "TPM Vulnerability: Bitlocker Full Disk Encryption impacted," 2017. [Online]. Available: <https://www.softcat.com/news/tpm-vulnerability-bitlocker-full-disk-encryption-impacted>. [Accessed 13 8 2022].

[35] Integration of an OPTIGA™ TPM SLx 9670 TPM2.0 with SPI Interface in a Raspberry Pi® 4 Linux environment. Munich, Germany: Infineon, 2019 [Online]. Available: <https://www.infineon.com/dgdl/Infineon-OPTIGA_SLx_9670_TPM_2.0_Pi_4-ApplicationNotes-v07_19-EN.pdf?fileId=5546d4626c1f3dc3016c3d19f43972eb>. [Accessed: 20- Aug- 2022]

[36] M. Stanojevic, "TPM 1.2 vs 2.0: Here's everything you need to know", Windows Report - Error-free Tech Life, 2021. [Online]. Available: <https://windowsreport.com/tpm-1-2-vs-2-0/>. [Accessed: 20- Aug- 2022]

[37] "LUKS: Disk Encryption", Guardian Project. [Online]. Available: <https://guardianproject.info/archive/luks/>. [Accessed: 20- Aug- 2022]

[38] "Trusted Platform Module - ArchWiki", Wiki.archlinux.org. [Online]. Available: <https://wiki.archlinux.org/title/Trusted_Platform_Module>. [Accessed: 20- Aug- 2022]

[39] "LUKS on Raspberry Pi", LUKS-on-Raspberry-Pi, 2021. [Online]. Available: <https://rr-developer.github.io/LUKS-on-Raspberry-Pi/>. [Accessed: 20- Aug- 2022]

[40] "Raspbian Stretch Luks Encrypt [solved] - Raspberry Pi Forums", Forums.raspberrypi.com, 2018. [Online]. Available: <https://forums.raspberrypi.com/viewtopic.php?t=219867>. [Accessed: 20- Aug- 2022]