TRUSTED® COMPUTING GROUP

GUIDANCE

Extended VRT10-11 Guidance

Revision 1
August 11, 2026

Contact: admin@trustedcomputinggroup.org

DISCLAIMERS, NOTICES, AND LICENSE TERMS

Contents

1 Purpose

Attestation in highly distributed environments can be especially challenging when considering vulnerable (or possibly-vulnerable) TPM implementations. Even once the patches are known and widely available, they must be rolled out to each endpoint. Attestation techniques (e.g., using the TPM) can help establish proof that patches reached particular endpoints, but when the patches fix bugs in the TPMs themselves, security engineers need to take extra caution.

This document provides extended guidance to system administrators and engineering teams related to two vulnerabilities (VRT10 and VRT11) that were disclosed in the TPM 2.0 reference code in 2026.

There is no one-size-fits-all solution. Each environment has its particular constraints and concerns. In this document, we provide context and case studies to help equip users and relying parties to best mitigate VRT10 and VRT11 in their environments.

2 Background

The Advisory for VRT10 [1] provides a high-level overview of VRT10:

- The Reference Code did not provide adequate separation between Objects (i.e., keys) and Hash Contexts, which are stored in the same area of TPM memory.
- A use-after-free bug caused stale Object state (i.e., the object Name) to be retained when the memory was reused for a Hash Context.
- The TPM2_Certify command attests that an Object with a particular Name exists (has its sensitive area protected) in a particular TPM.
- The TPM2_Certify command did not check that the handle being Certified was not a Hash Context.
- An attacker could exploit these bugs to get a TPM to Certify an external object and misrepresent it as being internally stored in a TPM.

The Advisory for VRT11 [2] provides a high-level overview of VRT11:

- The Reference Code did not implement RSA-OAEP decryption in secret-independent time.
- An attacker could exploit timing side channels while requesting the TPM to decrypt adaptively chosen ciphertexts in order to recover the plaintext for an arbitrary RSA-OAEP ciphertext (including an EK ActivateCredential challenge).

The TPM 2.0 errata [3] provides details on the affected locations in the reference code as well as minimal patches. Note that these patches are best-effort minimal fixes to the timing side channels, and a future version of the TPM reference code may be refactored to fully delegate all cryptographic operations (including RSA padding and padding checks) to the cryptographic library (e.g., OpenSSL).

3 Deploying Updates to TPMs

TPMs are typically designed to be field-upgraded, to allow for performance and security patches, but TPM firmware updates tend to be implementation- and platform-dependent, involving participation from both the TPM vendor (e.g., to provide the actual firmware update) and the platform vendor (e.g., to provide an updater suitable for a given platform). They also tend to be infrequent, which means that rolling out an important TPM firmware update at scale is especially challenging.

In some environments, the TPM firmware update may be performed in a trusted environment or by a trusted party (e.g., an IT support team physically interacting with a device).

In other environments, the TPM firmware update may be requested over the network and may rely on untrusted host software to actuate the update and report the results faithfully.

When the TPM firmware update is performed in an untrusted environment or by an untrusted party, it can be useful to use TPM attestation features in order to achieve confidence that the firmware update was actually applied.

Beginning with TPM 2.0 v184 [4], a TPM feature called “Firmware-Limited Keys” allows a supporting TPM to cryptographically attest to its own firmware, without implicitly trusting the firmware itself. Version 2.6 of the EK Credential Profile [5] provides a template for the TPM implementor to provide a Firmware-Limited EK certificate, which auto-rotates to a new key (certified by an embedded CA, e.g., in the TPM’s bootloader) when the TPM is patched.

When the TPM does not support Firmware-Limited EKs, the EK-Based Key Attestation with TPM Firmware guidance document [6] provides a protocol that can be used to ascertain the current firmware version of the TPM during AK attestation. This protocol is designed to be robust even in the presence of VRT10.

Administrators of heterogeneous fleets of TPMs can simplify the management of TPM patches by adopting TPM-firmware-version-aware attestation protocols (such as [6], or using Firmware-Limited EKs where supported by the TPM).

Protection against rollback of the TPM firmware is implementation-dependent. In environments where it is not known that every TPM prevents rollbacks, it is pragmatic to assume that a TPM could be updated to an old firmware version, and check the TPM for VRT10/VRT11 as often as practical (rather than assuming that a given TPM is trustworthy forever after taking the update once).

Maintaining tracking information for firmware versions of TPMs in the environment is not required in order to mitigate VRT10 and VRT11. However, it is a best practice that may facilitate response to TPM vulnerabilities in general.

4 Responding to VRT10

VRT10 involves a business-logic bug in a TPM implementation. Therefore, IT administrators and engineering teams are urged to work with their TPM and platform vendors to source and deploy TPM firmware updates to fix VRT10.

Because it takes time to deploy any update at scale, it is also recommended for owners or users of attestation systems to take precautions assuming that any given TPM in their environment may be affected by VRT10.

Specifically, this means that a given attestation statement from `TPM2_Certify` or attestation challenge for `TPM2_ActivateCredential` may be for an object that is *not* actually protected by a TPM. Instead, it may be for a so-called “external” object, where just the public key is loaded into the TPM.

It is possible to entirely mitigate this issue by migrating from the traditional `TPM2_ActivateCredential`-based AK attestation flow, to one based on `TPM2_Import` of a restricted HMAC key, as described by EK-Based Key Attestation with TPM Firmware [6]. In that flow, a restricted HMAC key is wrapped to the EK as an importable key blob, and used to Certify (`TPM2_Certify`) or CertifyCreation (`TPM2_CertifyCreation`) the subject AK.

`TPM2_CertifyCreation` is not affected by VRT10 because it requires valid creation data. However, it does require retaining the creation data for the AK from when it was created, which may or may not be practical in all environments.

`TPM2_Certify` is affected by VRT10, but it can be mitigated by checking that the `TPMS_CERTIFY_INFO`’s *name* field is not equal to *qualifiedName*.

Within a given `TPMS_CERTIFY_INFO`, *name* is not equal to *qualifiedName* if (and only if) the object is internal.

By using the EK-Based Key Attestation with TPM Firmware flow with either `TPM2_CertifyCreation`, or `TPM2_Certify` with the *name* != *qualifiedName* check, an AK attestation service can be resilient to the presence of VRT10 on any given TPM and thus protect itself (and its customers) from the possible presence of VRT10 on any given TPM in its environment.

Note: the EK-Based Key Attestation with TPM Firmware flow is specific to decryption EKs. If a TPM is equipped with a signing EK, it is sufficient to directly use that EK to produce a signature over an AK with either `TPM2_CertifyCreation`, or `TPM2_Certify` with the *name* != *qualifiedName* check.

5 Responding to VRT11

VRT11 involves a cryptographic side-channel vulnerability in a TPM implementation. Therefore, IT administrators and engineering teams are urged to work with their TPM and platform vendors to identify if their TPMs are vulnerable and, if needed, source and deploy TPM firmware updates to fix VRT11.

Because it takes time to deploy any update at scale, it is also recommended for owners or users of attestation systems to take precautions assuming that any given TPM in their environment may have been affected by VRT11.

VRT11 only affects RSA keys (e.g., RSA EKs). Therefore, the most trivial way to become resilient to VRT11 in any environment is to fully migrate away from RSA to ECC, or at least migrate away from RSA-OAEP decryption (which is the only way to use an RSA EK). ECC support throughout the TPM 2.0 ecosystem is mature (having been required for PC Client TPMs since 2015 [7], with available ECC EK certs having been required since 2020 [8]).

In some environments, it may not be practical to fully migrate from RSA-OAEP. Some additional approaches to mitigating VRT11 in these environments may be possible.

5.1 Leap-of-Faith to ECC EKs

For TPMs that support ECC but do not have an ECC EK cert, it may be useful to perform a one-time attestation of an ECC EK (using any template including the typical ECC EK templates in the EK Credential Profile [9]) using the RSA EK. This may either require a leap-of-faith that an attacker exploiting VRT11 was not present at that time, or it may be practical to perform the leap-of-faith in a trusted environment (for example, by an IT department during the deployment of the patched TPM firmware).

Note that support for new post-quantum cryptography (PQC) algorithms introduced in TPM 185 [10] means that cryptographic agility is a continual process, not a one-time step. The migration from RSA to ECC as part of a VRT11 response is similar to the migration from RSA and ECC to ML-KEM that will be part of the PQC transition.

5.2 Deadline-based Mitigation

Exploiting VRT11 requires around one observed RSA private-key operation attempt per bit of plaintext recovered. This means that in order to recover N bits of OAEP padded plaintext, the attacker needs to request and observe the timing side channels for N private-key operations with the target key.

If the performance characteristics of an environment's TPMs are known, they can be used to set strict RPC deadlines for the EK-based attestation protocol.

For example, if the fastest VRT-11-affected TPM in a given environment takes 100ms to attempt one RSA-2048 private key operation, then a deadline of 100 seconds would make it infeasible to recover over half the bits of padded plaintext before the deadline. If an even longer RPC deadline is desired, multiple EK challenges can be issued at the same time. Since TPM operations are typically serialized, the cost of the attack can be driven arbitrarily high.

References

[1] “VRT10 Advisory.” Trusted Computing Group, Aug. 2026.

[2] “VRT11 Advisory.” Trusted Computing Group, Aug. 2026.

[3] “TPM 2.0 v1.83 Errata.” Trusted Computing Group, Aug. 2026.

[4] “Trusted Platform Module 2.0 Library version 184.” Trusted Computing Group, Mar. 2025. Available: https://trustedcomputinggroup.org/resource/tpm-library-specification/

[5] “TPM 2.0 EK Credential Profile version 2.6.” Trusted Computing Group, Dec. 2024. Available: https://trustedcomputinggroup.org/resource/http-trustedcomputinggroup-org-wp-content-uploads-tcg-ek-credential-profile-v-2-5-r2_published-pdf/

[6] “EK-Based Key Attestation with TPM Firmware.” Trusted Computing Group, Jul. 2025. Available: https://trustedcomputinggroup.org/resource/ek-based-key-attestation-with-tpm-firmware/

[7] “TCG PC Client Platform TPM Profile (PTP) Specification revision 0.43.” Trusted Computing Group, Jan. 2015. Available: https://trustedcomputinggroup.org/wp-content/uploads/TCG_PC_Client_Platform_TPM_Profile_PTP_2.0_r43.pdf

[8] “TCG PC Client Platform TPM Profile Specification for TPM 2.0 version 1.04.” Trusted Computing Group, Feb. 2020. Available: https://trustedcomputinggroup.org/wp-content/uploads/PC-Client-Specific-Platform-TPM-Profile-for-TPM-2p0-v1p04_r0p37_pub-1.pdf

[9] “TCG EK Credential Profile For TPM Family 2.0; Level 0.” Trusted Computing Group, Dec. 2024. Available: https://trustedcomputinggroup.org/resource/http-trustedcomputinggroup-org-wp-content-uploads-tcg-ek-credential-profile-v-2-5-r2_published-pdf/

[10] “Trusted Platform Module 2.0 Library version 185.” Trusted Computing Group, Mar. 2026. Available: https://trustedcomputinggroup.org/resource/tpm-library-specification/

