IOTOPS
Internet-Draft
Intended status: Informational
Expires: 26 October 2023

B. Moran
Arm Limited

24 April 2023

A summary of security-enabling technologies for IoT devices
draft-moran-iot-nets-03

Abstract

The IETF has developed security technologies that help to secure the Internet of Things even over constrained networks and when targetting constrained nodes. These technologies can be used independenly or can be composed into larger systems to mitigate a variety of threats. This documents illustrates an overview over these technologies and highlights their relationships. Ultimately, a threat model is presented as a basis to derive requirements that interconnect existing and emerging solution technologies.

Table of Contents

1. Introduction

This memo serves as an entry-point to detail which technologies are available for use in IoT networks and to enable IoT designers to discover technologies that may solve their problems. This draft addresses.

Many baseline security requirements documents have been drafted by standards setting organisations, however these documents typically do not specify the technologies available to satisfy those requirements. They also do not express the next steps if an implementor wants to go above and beyond the baseline in order to differentiate their products and enable even better security. This memo defines the mapping from some IoT baseline security requirements definitions to ietf and related security technologies. It also highlights some gaps in those IoT baseline security requirements.

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Survey of baseline security requirements

At time of writing, there are IoT baseline security requirements provided by several organisations:

* ENISA's Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures ([ENISA-Baseline])

* ETSI's Cyber Security for Consumer Internet of Things: Baseline Requirements [ETSI-Baseline]

* NIST's IoT Device Cybersecurity Capability Core Baseline [NIST-Baseline]

4. Requirement Mapping

Requirements that pertain to hardware, procedure, and policy compliance are noted, but do not map to ietf and related technologies. NIST's requirements ([NIST-Baseline]) are very broad and already have mappings to ENISA baseline security recommendations.

4.1. Hardware Security

4.1.1. Identity

ENISA GP-PS-10: Establish and maintain asset management procedures and configuration controls for key network and information systems. NIST Device Identification: The IoT device can be uniquely identified logically and physically.

These requirements are architectural requirements, however [RFC4122] can be used for identifiers.

4.1.2. Hardware Immutable Root of Trust

ENISA GP-TM-01: Employ a hardware-based immutable root of trust.

This is an architectural requirement.

4.1.3. Hardware-Backed Secret Storage

ENISA GP-TM-02: Use hardware that incorporates security features to strengthen the protection and integrity of the device - for example, specialized security chips / coprocessors that integrate security at the transistor level, embedded in the processor, providing, among other things, a trusted storage of device identity and authentication means, protection of keys at rest and in use, and preventing unprivileged from accessing to security sensitive code. Protection against local and physical attacks can be covered via functional security.

NIST Data Protection: The ability to use demonstrably secure cryptographic modules for standardized cryptographic algorithms (e.g., encryption with authentication, cryptographic hashes, digital signature validation) to prevent the confidentiality and integrity of the devices stored and transmitted data from being compromised

This is an architectural requirement.

4.2. Software Integrity & Authenticity

4.2.1. Boot Environment Trustworthiness and Integrity

ENISA GP-TM-03: Trust must be established in the boot environment before any trust in any other software or executable program can be claimed.

Satisfying this requirement can be done in several ways, increasing in security guarantees:

1. Implement secure boot to verify the bootloader and boot environment. Trust is established purely by construction: if code is running in the boot environment, it must have been signed, therefore it is trustworthy.

2. Record critical measurements of each step of boot in a TPM. Trust is established by evaluating the measurements recorded by the TPM.

3. Use Remote Attestation. Remote attestation allows a device to report to third parties the critical measurements it has recorded (either in a TPM or signed by each stage) in order to prove the trustworthiness of the boot environment and running software. Remote Attestation is implemented in [I-D.ietf-rats-eat].

4.2.2. Code Integrity and Authenticity

ENISA GP-TM-04: Sign code cryptographically to ensure it has not been tampered with after signing it as safe for the device, and implement run-time protection and secure execution monitoring to make sure malicious attacks do not overwrite code after it is loaded.

Satisfying this requirement requires a secure invocation mechanism. In monolithic IoT software images, this is accomplished by Secure Boot. In IoT devices with more fully-featured operating systems, this is accomplished with an operating system-specific code signing practice.

Secure Invocation can be achieved using the SUIT Manifest format, which provides for secure invocation procedures. See [I-D.ietf-suit-manifest].

To satisfy the associated requirement of run-time protection and secure execution monitoring, the use of a TEE is recommended to protect sensitive processes. The TEEP protocol (see [I-D.ietf-teep-architecture]) is recommended for managing TEEs.

4.2.3. Secure Firmware Update

ENISA GP-TM-05: Control the installation of software in operating systems, to prevent unauthenticated software and files from being loaded onto it.

NIST Software Update:

1. The ability to update the device's software through remote (e.g., network download) and/or local means (e.g., removable media)

2. The ability to verify and authenticate any update before installing it

3. The ability for authorized entities to roll back updated software to a previous version

4. The ability to restrict updating actions to authorized entities only

5. The ability to enable or disable updating

6. Configuration settings for use with the Device Configuration capability including, but not limited to:

7. The ability to configure any remote update mechanisms to be either automatically or manually initiated for update downloads and installations

8. The ability to enable or disable notification when an update is available and specify who or what is to be notified

Many fully-featured operating systems have dedicated means of implementing this requirement. The SUIT manifest (See [I-D.ietf-suit-manifest]) is recommended as a means of providing secure, authenticated software update. Where the software is deployed to a TEE, TEEP (See [I-D.ietf-teep-protocol]) is recommended for software update and management.

4.2.4. Configuration

NIST Device Configuration:

1. The ability to change the device's software configuration settings

2. The ability to restrict configuration changes to authorized entities only

3. The ability for authorized entities to restore the device to a secure configuration defined by an authorized entity

Configuration can be delivered to a device either via a firmware update, such as in [I-D.ietf-suit-manifest], or via a runtime configuration interface, such as [LwM2M].

4.2.5. Resilience to Failure

ENISA GP-TM-06: Enable a system to return to a state that was known to be secure, after a security breach has occured or if an upgrade has not been successful.

While there is no specificaiton for this, it is also required in [RFC9019]

4.2.6. Trust Anchor Management

ENISA GP-TM-07: Use protocols and mechanisms able to represent and manage trust and trust relationships.

EST (https://datatracker.ietf.org/doc/html/rfc7030) and LwM2M Bootstrap ([LwM2M]) provide a mechanism to replace trust anchors (manage trust/trust relationships).

4.3. Default Security & Privacy

4.3.1. Security ON by Default

ENISA GP-TM-08: Any applicable security features should be enabled by default, and any unused or insecure functionalities should be disabled by default.

NIST Logical Access to Interfaces:

1. The ability to logically or physically disable any local and network interfaces that are not necessary for the core functionality of the device

2. The ability to logically restrict access to each network interface to only authorized entities (e.g., device authentication, user authentication)

3. Configuration settings for use with the Device Configuration capability including, but not limited to, the ability to enable, disable, and adjust thresholds for any ability the device might have to lock or disable an account or to delay additional authentication attempts after too many failed authentication attempts

These are procedural requirements, rather than a protocol or document requirement.

4.3.2. Default Unique Passwords

ENISA GP-TM-09: Establish hard to crack, device-individual default passwords.

This is a procedural requirement, rather than a protocol or document requirement.

4.4. Data Protection

The data protection requirements are largely procedural/ architectural. While this memo can recommend using TEEs to protect data, and TEEP ([I-D.ietf-teep-architecture]) to manage TEEs, implementors must choose to architect their software in such a way that TEEs are helpful in meeting these requirements.

ENISA Data Protection requirements:

* GP-TM-10: Personal data must be collected and processed fairly and lawfully, it should never be collected and processed without the data subject's consent.

* GP-TM-11: Make sure that personal data is used for the specified purposes for which they were collected, and that any further processing of personal data is compatible and that the data subjects are well informed.

* GP-TM-12: Minimise the data collected and retained.

* GP-TM-13: IoT stakeholders must be compliant with the EU General Data Protection Regulation (GDPR).

* GP-TM-14: Users of IoT products and services must be able to exercise their rights to information, access, erasure, rectification, data portability, restriction of processing, objection to processing, and their right not to be evaluated on the basis of automated processing.

NIST Data Protection:

1. The ability to use demonstrably secure cryptographic modules for standardized cryptographic algorithms (e.g., encryption with authentication, cryptographic hashes, digital signature validation) to prevent the confidentiality and integrity of the device's stored and transmitted data from being compromised

2. The ability for authorized entities to render all data on the device inaccessible by all entities, whether previously authorized or not (e.g., through a wipe of internal storage, destruction of cryptographic keys for encrypted data)

3. Configuration settings for use with the Device Configuration capability including, but not limited to, the ability for authorized entities to configure the cryptography use itself, such as choosing a key length

4.5. System Safety and Reliability

Safety and reliability requirements are procedural/architectural. Implementors should ensure they have processes and architectures in place to meet these requirements.

ENISA Safety and Reliability requirements:

* GP-TM-15: Design with system and operational disruption in mind, preventing the system from causing an unacceptable risk of injury or physical damage.

* GP-TM-16: Mechanisms for self-diagnosis and self-repair/healing to recover from failure, malfunction or a compromised state.

* GP-TM-17: Ensure standalone operation - essential features should continue to work with a loss of communications and chronicle negative impacts from compromised devices or cloud-based systems.

4.6. Secure Software / Firmware updates

Technical requirements for Software Updates are provided for in SUIT ([I-D.ietf-suit-manifest]) and TEEP ([I-D.ietf-teep-protocol]). Procedural and architectural requirements should be independently assessed by the implementor.

ENISA Software Update Requirements:

* GP-TM-18: Ensure that the device software/firmware, its configuration and its applications have the ability to update Over-The-Air (OTA), that the update server is secure, that the update file is transmitted via a secure connection, that it does not contain sensitive data (e.g. hardcoded credentials), that it is signed by an authorised trust entity and encrypted using accepted encryption methods, and that the update package has its digital signature, signing certificate and signing certificate chain, verified by the device before the update process begins.

* GP-TM-19: Offer an automatic firmware update mechanism.

* GP-TM-20: (Procedural / Architectural) Backward compatibility of firmware updates. Automatic firmware updates should not modify user-configured preferences, security, and/or privacy settings without user notification.

4.7. Authentication

4.7.1. Align Authentication Schemes with Threat Models

ENISA GP-TM-21: Design the authentication and authorisation schemes (unique per device) based on the system-level threat models.

This is a procedural / architectural requirement.

4.7.2. Password Rules

ENISA applies the following requirements to Password-based authentication:

* GP-TM-22: Ensure that default passwords and even default usernames are changed during the initial setup, and that weak, null or blank passwords are not allowed.

* GP-TM-23: Authentication mechanisms must use strong passwords or personal identification numbers (PINs), and should consider using two-factor authentication (2FA) or multi-factor authentication (MFA) like Smartphones, Biometrics, etc., on top of certificates.

* GP-TM-24: Authentication credentials shall be salted, hashed and/or encrypted.

* GP-TM-25: Protect against 'brute force' and/or other abusive login attempts. This protection should also consider keys stored in devices.

* GP-TM-26: Ensure password recovery or reset mechanism is robust and does not supply an attacker with information indicating a valid account. The same applies to key update and recovery mechanisms.

As an alternative, implementors are encouraged to consider passwordless schemes, such as FIDO.

4.8. Authorisation

4.8.1. Principle of Least Privilege

ENISA GP-TM-27: Limit the actions allowed for a given system by Implementing fine-grained authorisation mechanisms and using the Principle of least privilege (POLP): applications must operate at the lowest privilege level possible.

This is a procedural / architectural requirement, however at the network level, this can be implemented using Manufacturer Usage Descriptions (see [RFC8520]).

4.8.2. Software Isolation

ENISA GP-TM-28: Device firmware should be designed to isolate privileged code, processes and data from portions of the firmware that do not need access to them. Device hardware should provide isolation concepts to prevent unprivileged from accessing security sensitive code.

Implementors should use TEEs to address this requirement. The provisioning and management of TEEs can be accomplished using TEEP (see [I-D.ietf-teep-architecture]).

4.8.3. Access Control

ENISA GP-TM-29: Data integrity and confidentiality must be enforced by access controls. When the subject requesting access has been authorised to access particular processes, it is necessary to enforce the defined security policy. ENISA GP-TM-30: Ensure a context-based security and privacy that reflects different levels of importance.

These requirements are complex and require a variety of technologies to implement. Use of TEEs can provide a building block for these requirements, but is not sufficient in itself to meet these requiremnents.

4.9. Environmental and Physical Security

ENISA defines the following physical security requirements. These are hardware-architectural requirements and not covered by protocol and format specifications.

* GP-TM-31: Measures for tamper protection and detection. Detection and reaction to hardware tampering should not rely on network connectivity.

* GP-TM-32: Ensure that the device cannot be easily disassembled and that the data storage medium is encrypted at rest and cannot be easily removed.

* GP-TM-33: Ensure that devices only feature the essential physical external ports (such as USB) necessary for them to function and that the test/debug modes are secure, so they cannot be used to maliciously access the devices. In general, lock down physical ports to only trusted connections.

4.10. Cryptography

ENISA makes the following architectural cryptography requirements for IoT devices:

* GP-TM-34: Ensure a proper and effective use of cryptography to protect the confidentiality, authenticity and/or integrity of data and information (including control messages), in transit and in rest. Ensure the proper selection of standard and strong encryption algorithms and strong keys, and disable insecure protocols. Verify the robustness of the implementation.

* GP-TM-35: Cryptographic keys must be securely managed.

* GP-TM-36: Build devices to be compatible with lightweight encryption and security techniques.

* GP-TM-37: Support scalable key management schemes.

4.11. Secure and Trusted Communications

4.11.1. Data Security

GP-TM-38: Guarantee the different security aspects -confidentiality (privacy), integrity, availability and authenticity- of the information in transit on the networks or stored in the IoT application or in the Cloud.

This Data Security requirement can be fulfilled using COSE [RFC8152] for ensuring the authenticity, integrity, and confidentiality of data either in transit or at rest. Secure Transport (see Section 4.11.2) technologies can be used to protect data in transit.

4.11.2. Secure Transport

ENISA GP-TM-39: Ensure that communication security is provided using state-of-the-art, standardised security protocols, such as TLS for encryption. ENISA GP-TM-40: Ensure credentials are not exposed in internal or external network traffic.

This requirement is satisfied by several standards:

* TLS ([RFC8446]).

* DTLS ([RFC9147]).

* QUIC ([RFC9000]).

* OSCORE ([RFC9203]).

4.11.3. Data Authenticity

ENISA GP-TM-41: Guarantee data authenticity to enable reliable exchanges from data emission to data reception. Data should always be signed whenever and wherever it is captured and stored.

The authenticity of data can be protected using COSE [RFC8152].

ENISA GP-TM-42: Do not trust data received and always verify any interconnections. Discover, identify and verify/authenticate the devices connected to the network before trust can be established, and preserve their integrity for reliable solutions and services.

Verifying communication partners can be done in many ways. Key technologies supporting authentication of communication partners are:

* RATS: Remote attestation of a communication partner (See [I-D.ietf-rats-architecture]).

* TLS/DTLS: Mutual authentication of communication partners (See [RFC8446] / [RFC9147]).

* ATLS: Application-layer TLS for authenticating a connection that may traverse multiple secure transport connections.

* Attested TLS: The use of attestation in session establishment in TLS (See [I-D.fossati-tls-attestation]).

4.11.4. Least Privilege Communication

ENISA GP-TM-43: IoT devices should be restrictive rather than permissive in communicating.

This Requirement can be enabled and enforced using Manufacturer Usage Descriptions, which codify expected communication (See [RFC8520])

ENISA GP-TM-44: Make intentional connections. Prevent unauthorised connections to it or other devices the product is connected to, at all levels of the protocols.

This requirement can be satisfied through authenticating connections (TLS / DTLS mutual authentication. See [RFC8446] / [RFC9147]) and declaring communication patterns (Manufacturer Usage Descriptions. See [RFC8520])

Architectural / Procedural requirements:

* ENISA GP-TM-45: Disable specific ports and/or network connections for selective connectivity.

* ENISA GP-TM-46: Rate limiting. Controlling the traffic sent or received by a network to reduce the risk of automated attacks.

4.12. Secure Interfaces and network services

ENISA Architectural / Procedural requirements:

* GP-TM-47: Risk Segmentation. Splitting network elements into separate components to help isolate security breaches and minimise the overall risk.

* GP-TM-48: Protocols should be designed to ensure that, if a single device is compromised, it does not affect the whole set.

* GP-TM-49: Avoid provisioning the same secret key in an entire product family, since compromising a single device would be enough to expose the rest of the product family.

* GP-TM-50: Ensure only necessary ports are exposed and available.

* GP-TM-51: Implement a DDoS-resistant and Load-Balancing infrastructure.

* GP-TM-53: Avoid security issues when designing error messages.

4.12.1. Encrypted User Sessions

ENISA GP-TM-52: Ensure web interfaces fully encrypt the user session, from the device to the backend services, and that they are not susceptible to XSS, CSRF, SQL injection, etc.

This requirement can be partially satisfied through use of TLS or QUIC (See [RFC8446] and [RFC9000])

4.13. Secure input and output handling

Architectural / Procedural requirements:

ENISA GP-TM-54: Data input validation (ensuring that data is safe prior to use) and output filtering.

4.14. Logging

Architectural / Procedural requirements:

ENISA GP-TM-55: Implement a logging system that records events relating to user authentication, management of accounts and access rights, modifications to security rules, and the functioning of the system. Logs must be preserved on durable storage and retrievable via authenticated connections.

NIST Cybersecurity State Awareness

1. The ability to report the device’s cybersecurity state

2. The ability to differentiate between when a device will likely operate as expected from when it may be in a degraded cybersecurity state

3. The ability to restrict access to the state indicator so only authorized entities can view it

4. The ability to prevent any entities (authorized or unauthorized) from editing the state except for those entities that are responsible for maintaining the device’s state information

5. The ability to make the state information available to a service on another device, such as an event/state log server

Certain logs and indicators of cybersecurity state can be transported via RATS: See [I-D.ietf-rats-eat]. Where associated with SUIT firmware updates, logs can be transported using SUIT Reports. See [I-D.ietf-suit-report].

4.15. Monitoring and Auditing

Architectural / Procedural requirements:

* ENISA GP-TM-56: Implement regular monitoring to verify the device behaviour, to detect malware and to discover integrity errors.

* ENISA GP-TM-57: Conduct periodic audits and reviews of security controls to ensure that the controls are effective. Perform penetration tests at least biannually.

5. Security Considerations

No additional security considerations are required; they are laid out in the preceeding sections.

6. Normative References

[ENISA-Baseline]
ENISA, "Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures", n.d., <https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot/>.

[ETSI-Baseline]
ETSI, "Cyber Security for Consumer Internet of Things: Baseline Requirements", n.d., <https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf>.

[FDO] FIDO Alliance, "FIDO Device Onboarding", n.d., <https://fidoalliance.org/specs/FDO/FIDO-Device-Onboard-RD-v1.0-20201202.html>.

[I-D.birkholz-rats-corim]
Birkholz, H., Fossati, T., Deshpande, Y., Smith, N., and W. Pan, "Concise Reference Integrity Manifest", Work in Progress, Internet-Draft, draft-birkholz-rats-corim-03, 11 July 2022, <https://datatracker.ietf.org/doc/html/draft-birkholz-rats-corim-03>.

[I-D.fossati-tls-attestation]
Tschofenig, H., Sheffer, Y., Howard, P., Mihalcea, I., and Y. Deshpande, "Using Attestation in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", Work in Progress, Internet-Draft, draft-fossati-tls-attestation-03, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-fossati-tls-attestation-03>.

[I-D.ietf-rats-architecture]
Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", Work in Progress, Internet-Draft, draft-ietf-rats-architecture-22, 28 September 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-architecture-22>.

[I-D.ietf-rats-eat]
Lundblade, L., Mandyam, G., O'Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", Work in Progress, Internet-Draft, draft-ietf-rats-eat-19, 19 December 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-eat-19>.

[I-D.ietf-sacm-coswid]
Birkholz, H., Fitzgerald-McKay, J., Schmidt, C., and D. Waltermire, "Concise Software Identification Tags", Work in Progress, Internet-Draft, draft-ietf-sacm-coswid-24, 24 February 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-sacm-coswid-24>.

[I-D.ietf-suit-manifest]
Moran, B., Tschofenig, H., Birkholz, H., Zandberg, K., and O. Rønningstad, "A Concise Binary Object Representation (CBOR)-based Serialization Format for the Software Updates for Internet of Things (SUIT) Manifest", Work in Progress, Internet-Draft, draft-ietf-suit-manifest-22, 27 February 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-suit-manifest-22>.

[I-D.ietf-suit-report]
Moran, B. and H. Birkholz, "Secure Reporting of Update Status", Work in Progress, Internet-Draft, draft-ietf-suit-report-05, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-suit-report-05>.

[I-D.ietf-teep-architecture]
Pei, M., Tschofenig, H., Thaler, D., and D. M. Wheeler, "Trusted Execution Environment Provisioning (TEEP) Architecture", Work in Progress, Internet-Draft, draft-ietf-teep-architecture-19, 24 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-teep-architecture-19>.

[I-D.ietf-teep-protocol]
Tschofenig, H., Pei, M., Wheeler, D. M., Thaler, D., and A. Tsukamoto, "Trusted Execution Environment Provisioning (TEEP) Protocol", Work in Progress, Internet-Draft, draft-ietf-teep-protocol-12, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-teep-protocol-12>.

[IoTopia] "Global Platform Iotopia", n.d., <https://globalplatform.org/iotopia/mud-file-service/>.

[LwM2M] NIST, "LwM2M Core Specification", n.d., <https://csrc.nist.gov/Projects/Software-Identification-SWID/guidelines>.

[NIST-Baseline]
NIST, "IoT Device Cybersecurity Capability Core Baseline", n.d., <https://www.nist.gov/publications/iot-device-cybersecurity-capability-core-baseline>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <https://www.rfc-editor.org/rfc/rfc4122>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/rfc/rfc7030>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/rfc/rfc8152>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

[RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <https://www.rfc-editor.org/rfc/rfc8520>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/rfc/rfc8995>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/rfc/rfc9000>.

[RFC9019] Moran, B., Tschofenig, H., Brown, D., and M. Meriac, "A Firmware Update Architecture for Internet of Things", RFC 9019, DOI 10.17487/RFC9019, April 2021, <https://www.rfc-editor.org/rfc/rfc9019>.

[RFC9147] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022, <https://www.rfc-editor.org/rfc/rfc9147>.

[RFC9203] Palombini, F., Seitz, L., Selander, G., and M. Gunnarsson, "The Object Security for Constrained RESTful Environments (OSCORE) Profile of the Authentication and Authorization for Constrained Environments (ACE) Framework", RFC 9203, DOI 10.17487/RFC9203, August 2022, <https://www.rfc-editor.org/rfc/rfc9203>.

Author's Address

Brendan Moran
Arm Limited
Email: brendan.moran.ietf@gmail.com