VDC Research
Insights for the Connected World

Identity Technologies for IoT

A Comparison of Best Practices

Exclusive License to Distribute: Intel
By Steve Hoffenberg, Director, with Chris Rommel, Executive Vice President

TABLE OF CONTENTS

THE IMPORTANCE OF IOT DEVICE IDENTITY

As the Internet of Things expands the number of connected devices into the tens of billions, much of IoT functionality is contingent upon devices being able to reliably identify themselves to each other (M2M), to the edge (networks, gateways & fog servers), and to device management platforms (cloud or on-premises). to form an end-to-end authenticated security channel.

In the IoT, identity is a digital equivalent of the answer to the question “Who’s there?” when someone knocks on a door. In most—but not all—cases the answer is unique for each device, just as no two people have the same passport or driver’s license numbers, even though multiple people may have the same name. Authentication is the subsequent process of confirming the identity of a device seeking access to system resources, be they functional controls or data. Authentication verifies the truthfulness of the response to “Who’s there?”

In the context of IoT systems, however, knowing and verifying who’s knocking do not imply that a device has permission to enter even if its identity is confirmed. Permission is the role of authorization rather than identity or authentication, although it is reliant on both. Indeed it is impossible to know whether a device is authorized to access resources such as an IoT network or content without establishing at least certain aspects of its identity.

The authentication and authorization processes may also require attestation, in which the device must prove it’s in a trusted state, such as by verifying that a trusted execution environment, firmware, middleware and software application stack are running expected images and configurations. Such attestation can occur locally, or via connection to a remote system.

In many situations it is advantageous for identity to be established anonymously, that is, for a device to cryptographically prove it is a valid member of an authorized group, while keeping its actual identity private. Doing so can, for example, prevent hackers from uniquely identifying a specific device, or limit the extent of information disclosed by the device which might be associated with an individual user. With the upcoming European General Data Protection Regulation, the less personally identifiable information handled, the lower the risk that an organization might incur a hefty fine. A hospital might need to know which specific unit of a medical device is in which patient’s room, but the device’s manufacturer doesn’t need to know and may be better off not knowing.

Some of the uses for the identity, authentication, and authorization processes include:

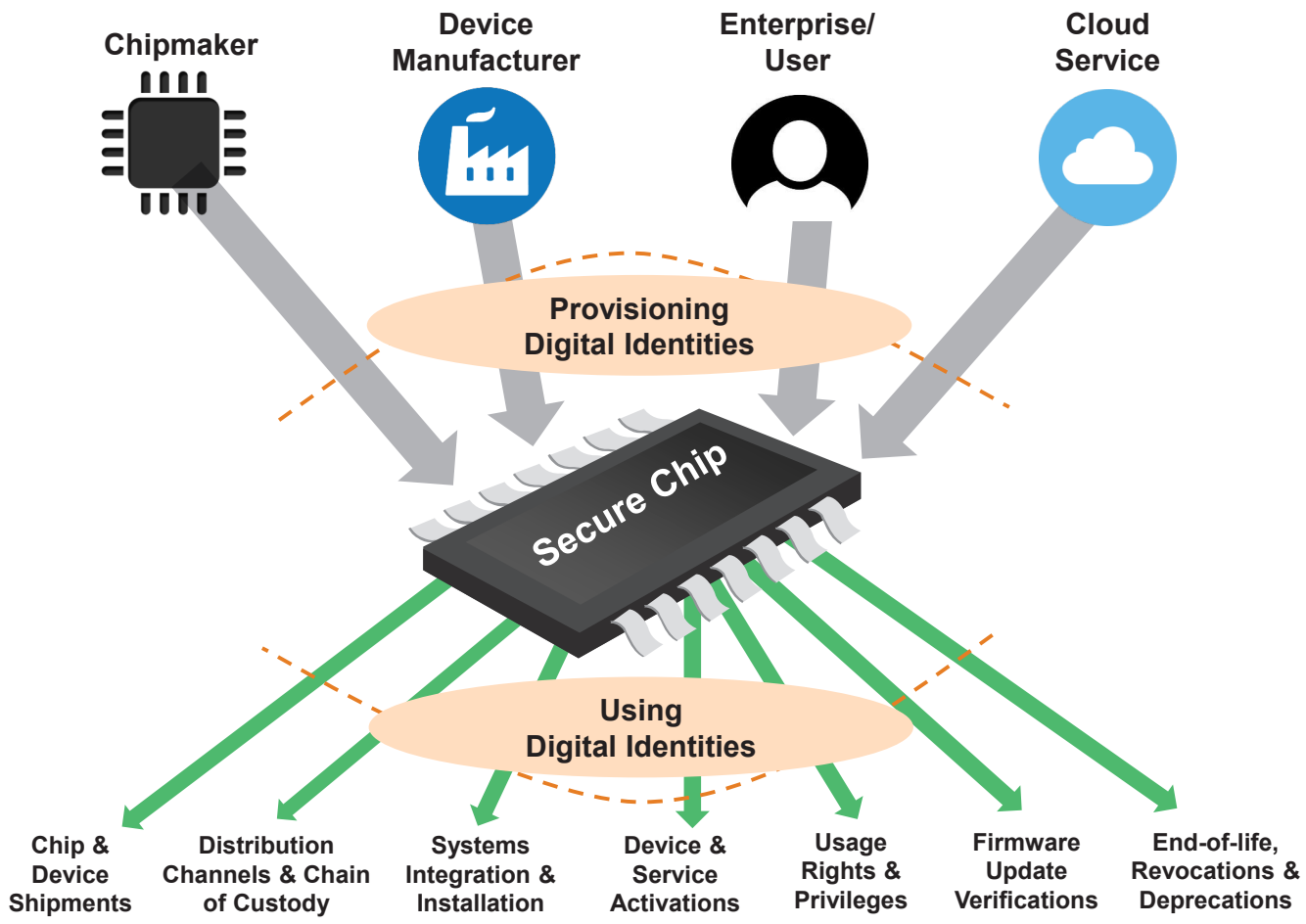
- protection of a device from unauthorized access to its resources, including those that might subvert its functionality (e.g. DDoS botnet), or be used to reveal the device’s ownership or location (e.g. IP address, GPS sensor, or other artifacts);
- protection of content from unauthorized use (e.g., digital rights management for entertain-ment media);
- protection of device data from unauthorized access (e.g., personally identifiable health care information);
- protection of device data from spoofing or tampering (e.g., electrical utility usage metering).

IoT devices may also have multiple identities, each of which is used in different contexts. For example, one identity might be used when sending sensor data to a local gateway, while another identity is used to anonymously onboard the device for management by an IoT cloud platform, and yet another identity is used to enable reception of firmware updates from the device manufacturer.

However, Internet-connected devices are inherently subject to possible monitoring or falsification of communications, even during the identity and authentication processes. To ensure integrity of devices and their data, both identity and authentication processes need to be cryptographically proven and encrypted. In certain breaches, such as Man-In-The-Middle attacks, even encrypted data can be compromised if the communications channel itself is compromised. In addition, hackers who gain access to insufficiently secure devices could potentially corrupt the devices' identity data, causing them to fail to function properly, or steal their identity data to enable unauthorized devices to spoof themselves as having authorized identities. Therefore, both stored identity data and the communications channel used for authentication need to be secure.

In this paper, we discuss methods for creating and using secure identities for IoT devices.

IDENTITY & DEVICE LIFECYCLE



There are many stages throughout the lifecycle of an IoT device where identity can be established and/or used, as described below.

Manufacturing

When a new device is produced, it can be pre-provisioned with a digital identity. This identity is often placed into either:

a) dedicated security hardware, such as a Trusted Platform Module (TPM) or a secure element (SE) chip; or
b) a "secure world" hardware area within a general purpose processor, such as one time programmable fuse, a virtual "on chip" TPM such as Intel Platform Protection Technology, or a trusted execution environment such as Intel® Software Guard Extensions (SGX) or ARM® TrustZone.

In these scenarios, an encrypted identity or a globally unique identifier (GUID), can be inserted during initial manufacture of the semiconductors, or subsequently during board/product assembly on the device production line at the OEM, ODM, or contract manufacturer. This pre-boot identity designed-in to the hardware has the advantage of secure inherent distribution ahead of potential field installation errors that open new security holes. For example, this identity could be used in conjunction with an automated authentication service to eliminate the need for hard-coded default usernames and passwords, such as those compromised in the Mirai botnet attacks. An ID at this level can also function as an immutable Root-of-Trust for other stages of the device's lifecycle. In most cases, a Hardware Security Module appliance (HSM) in the factory is used to generate the digital identities in a secure manner and create a record of those identities, although a cloud-based HSM-as-a-Service can also be used.

After the manufacturing stage, subsequent identities are often assigned via Public Key Infrastructure (PKI), with a private key functioning as the device ID.

Supply Chain

When devices are in transit, changing ownership from the manufacturer through distributors or other channel partners, digital identities (rather than a bill of lading paper trail) can be used to keep track of which entity has possession or various responsibilities for which units. These may be the same IDs (or GUIDs) that were pre-provisioned at the factory, or different IDs created and provisioned in the channel such as RFIDs. Additionally, a blockchain ledger could keep secure records of the devices' identity histories as they progress through the supply chain.

Installation & Onboarding

Devices can be designed so that when they are initially installed, they communicate with a server that authenticates their ID and provides additional credentials, such as more digital IDs or public key/private key X.509 pairs for subsequent use with cloud services during normal device operation. At this stage, for example, credentials may be established based on geographic location to comply with data residency requirements, and the manufacturer can keep track of which of its units have been installed. These ID provisioning steps may be conducted manually, in batch provisioning lots by the ODM, or automatically post boot, depending on how the devices and services are implemented.

Site Provisioning

In commercial or industrial facilities, network administrators at the site where a device is installed can use IDs to configure it for accessing local resources such as wireless networks and gateways, register it for IT (information technology) and OT (operational technology) systems, or other company-specific or site-specific services. Automated and trusted identity and configuration processes may reduce production downtime and potential friction between IT and OT departments. In a consumer home, this type of provisioning could include registering the device for authorized communications with a smart hub or other home controller. In both commercial and consumer environments, an automated process also may enable non-technically-oriented people to perform installations. Setup would only require access to the local network, not to secure resources (such as root level privileges) within the device.

Ongoing Operations & Management

During the useful life of a device, one or more of its IDs may be used to enable it to receive firmware updates, to give it access to new content, data, services, or third-party software, or even to update onboarding IDs with operational IDs from the customer's choice of PKI authority. The processes also need to be able to accommodate re-assignment of lost or corrupted identities, as well as revocation of credentials to decommission devices no longer authorized to access specific content or services due to policy changes, device resale, or end-of-life support termination by the manufacturer or service provider.

As we can see from the above lifecycle descriptions, device identity can be a complex task, involving multiple identities—both static and dynamic—as well as multiple business interests. In each instance, identity needs to be provisioned by a trustworthy party, and stored, retrieved, and communicated in a manner assuring trust to entities wishing to authorize the device for access to resources, data, content or services. For the IoT to successfully scale to billions of secure connected devices, all this identity provisioning and authentication needs to happen with little or no human intervention. And it needs to happen in a way that all interested parties can trust.

The Role of Public Key Infrastructure

Public Key Infrastructure, with its use of asymmetric cryptography, forms a cornerstone of today's Internet. PKI enables much of what we take for granted, for example, that emails and credit card transactions aren't openly exposed at every point in their journey across the Internet, even when communicating with people or sites with which we've never previously communicated. PKI also enables digital signatures which prove that a firmware or software release is a bit perfect replica of a reference version from the developer, and the signature matches one that could only be provided by a source in possession of the developer's private key.

Among other attributes, PKI relies on trusting that a Certificate Authority (CA):

- properly generates public/private key pairs
- keeps private keys secret
- correctly provides public keys with low latency
- maintains and applies certificate revocation lists
- only trusts other CAs known to it through established links or hierarchies.

PKI has its limitations for the IoT. It requires that identities be provided to and retained by third parties (usually) in centralized databases which include potentially lengthy revocation lists. Although those databases are encrypted so their contents aren't likely to be divulged—at least until quantum computing becomes viable—the databases could potentially be corrupted or destroyed by accident or hacking incident.

An organization can establish and operate its own Certificate Authority, essentially a privately run PKI, to self-sign digital certificates and issue encryption keys. Doing so in a secure manner requires considerable resources, such as purchase of multiple Hardware Security Module appliances, which can cost tens of thousands of dollars or more. And use of the keys outside the organization would still require other entities to trust the CA.

Alternatively, security-as-a-service providers specializing in identities and provisioning, such as Device Authority, can manage these processes for manufacturers and customers. Major cloud service providers are also offering identity provisioning and management services, although procuring all cloud and identity services from the same vendor may create a single point of failure as well as restrict the ability of a customer to choose individual best-of-breed identity services. A distributed trust model does not rely on any single vendor.

The Blockchain Trust Model

Although the PKI system in use today is distributed across numerous CA's, any one CA is a single node that exclusively holds its database. Each CA database, therefore, is centralized and represents a single point of failure. This is in contrast to a blockchain, in which the entire ledger is distributed across its nodes. Blockchain nodes are not single points of failure, and the contents of any node can be verified by comparing it to the contents of other nodes, forming a

consensus about the validity of any information or transaction. Because of the consensus processes, individual nodes of a blockchain do not need to be trusted. The trust resides in the architecture of the blockchain system.

Ironically, a blockchain relies on digital signatures using private/public key pairs issued from PKI to encrypt and decrypt its data. A lost or disclosed private key might risk the contents of individual blocks of data, but the integrity of the overall blockchain remains intact because the ledger is immutable.

Various efforts are underway to apply blockchain's decentralized trust model to PKI itself.

For IoT devices, blockchain technology offers the opportunity to add a layer of trust into the processes of manufacturing, distributing, selling, installing, and servicing products.

INTEL® EPID: A CASE FOR SEAMLESS IOT IDENTITY FROM ONBOARDING TO OPERATION

Back in 2009, a pair of researchers at Intel published an article detailing a new identity technology they had developed called Enhanced Privacy ID (EPID).[1] EPID could enable device manufacturers to remotely authenticate that a device in the field was a genuine article, i.e. not counterfeit, and Intel had already begun using in its own processor chips.

A common method devised prior to EPID to accomplish a similar objective embeds in the device a digital certificate signed using the manufacturer's private key. When the manufacturer needs to verify the device, the manufacturer sends a verification request—again signed with its private key—then the device uses its own private key to add its digital signature to the original certificate and sends it back. That process allows both parties to establish that each other are genuine by using public keys to verify their digital signatures. However, this type of remote authentication allows the manufacturer to uniquely identify each individual device unit, which is not always desirable for users or necessary for the manufacturer.

The crucial attribute of EPID distinguishing it from conventional public key cryptography in the above example is that with EPID, multiple private keys correspond to the same public key. When a device returns a certificate signed with its private key, the receiving party can use the public key to authenticate that the device is a member of an authorized group of devices, but it won't know which specific device signed the certificate. Thereby, EPID enables mutual authenti-cation to occur between the device and other parties, while maintaining anonymity for individual devices.

The crucial attribute of EPID distinguishing it from conventional public key cryptography is that with EPID, multiple private keys correspond to the same public key.

A technique called Direct Anonymous Attestation (DAA) has properties similar to EPID and had been developed previously by a team including one of the same researchers. EPID is essentially a new variation of DAA with expanded capabilities to revoke private keys.

Besides anonymity, EPID has the advantage over conventional PKI in that EPID certificate issuing authorities do not need to maintain databases of private/public key pairs. They only need to keep the public keys, which by virtue of being public can be readily distributed to multiple nodes. An EPID algorithm enables any node to determine whether a given

[1] Brickell, Ernie and Li, Jiangtao, Intel Technology Journal, Volume 13, Issue 2, 2009, pp. 96-111.

digital signature corresponds to the public key of a particular group. Revocation lists still need to be maintained, but they can be based on EPID signatures created with revoked private keys, without necessarily including the actual private keys.

EPID is compliant with ISO/IEC standards 20008/20009, as well as the Trusted Computing Group (TCG) standard for TPM 2.0 authentication.

EPID Use Cases

Intel began provisioning EPID identities in its Xeon CPUs, starting back in 2008, and added them to its Core processors in 2011 and Atom processors in 2014. The company has used EPID to support firmware updates and multi-factor authentication.

Another common use case for EPID is in digital rights management (DRM) for video content protection, and in that context EPID is an integral component of the UltraViolet high definition streaming video service[2] used in PCs, game consoles, smart TVs, connected Blu-ray players, and other streaming video boxes.

Other large potential markets include: government issued identity cards, medical devices; and anonymous reporting of weather and environmental data.

Intel maintains its own services to generate and provision EPID identities and verify EPID signa-tures from deployed devices. To date, Intel has issued more than 2.7 billion EPID identities, with that number likely to rise rapidly in the near future.

To date, Intel has issued more than ***2.7 billion*** *EPID identities.*

EPID Licensing

Intel has openly published the EPID specification and licenses its use under RAND-Z terms, that is, reasonable and non-discriminatory with zero royalty.

In 2015, semiconductor manufacturers Atmel and Microchip announced they were licensing EPID for inclusion in their microcontrollers. (Microchip has since acquired Atmel.) More recently, a growing list of non-Intel MCU providers have all licensed EPID.

Licensees can set up their own systems to generate, provision, and verify EPID identities, although Intel is now offering a publicly available EPID key generation service and an EPID based onboarding service called the Intel® Secure Device Onboard (Intel® SDO) solution, which is described next.

Intel® Secure Device Onboard (Intel® SDO) Solution

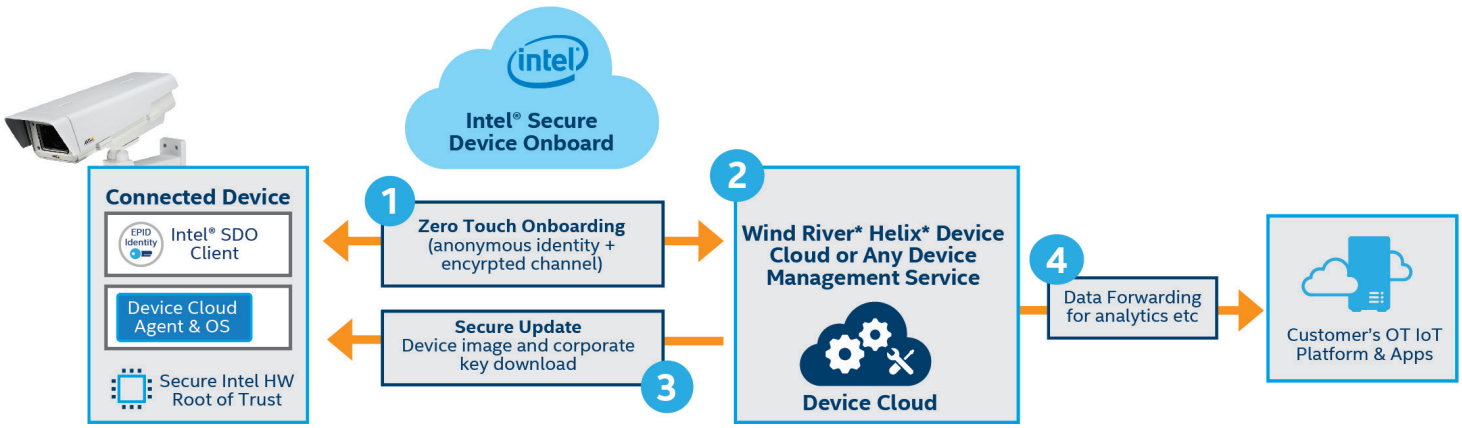
Most product manufacturers don't want to deal with the complexity of setting up and operating their own infrastructures to generate and manage digital identities. Key management services for IoT devices are available from many PKI vendors, such as DigiCert, Gemalto, GlobalSign, and Symantec. However, one of the hurdles that has held back increased use of digital identities in connected devices has been the need for multiple manual steps at installation time to register devices and provision certificates for specific network or cloud services. The device maker can't always know at the time of manufacturer the supply chain through which every device will be delivered, where it will be installed, and for which services it will need to be provisioned.

[2] See https://www.myuv.com.

To alleviate these issues Intel has introduced the Intel® SDO.The solution consists of a series of hardware, software, and cloud-based service elements, with the goal being what Intel calls “Zero-Touch Onboarding.” that dramatically reduces ODM pre-loading and can onboard to any IoT platform. Although the sequence and extent of steps can vary, in a typical case:

- The chipmaker includes the EPID identity in a “secure world” area of protected hardware, such as a one-time writeable secure fuse or immutable memory. The group keys are insert-ed on the chip production line.
- The board maker or device manufacturer integrates Intel® SDO client software into the device’s boot code & trusted execution environment.
- The IoT platform provider approved to handle the device integrates an Intel® SDO API into its platform.
- The Intel® SDO service’s digital signature tools can optionally keep track of ownership credentials of a group of devices as it passes through distributor and retailer channels, and notify the IoT platform when devices are delivered & powered on.
- When a device is installed and booted for the first time, the Intel® SDO client software estab-lishes an anonymous connection with the Intel® SDO service and provides the device’s signed digital certificate.
- After verifying the device’s identity, the service helps broker a rendezvous URL to hand off communication to the IoT platform, which registers the device as a being an active member of a particular EPID group.
- Through the secure anonymous channel established by EPID au-thentication, the platform can perform additional authentication, such as verify a device is within a particular group or that it is trusted for a given purpose. Then the device and platform can establish identities (known to just the two parties) in a secure session for access to spe-cific networks or resources.
- Note that the specific configuration of these services does not need to be known to the device maker, Intel® SDO, or IoT platform at the time of device manufacture, only at the time the device is installed, and may vary, for example, based on the geographic location corre-sponding to the device network’s IP address.
- If the firmware or services for the device need to change, such changes are made in the IoT platform, which can use Intel® SDO API calls to request and verify the original EPID identity from the device prior to sending new firmware, services keys, etc.
- Intel® SDO can onboard to any IoT platform service and has been pre-integrated to work with Intel Helix Device Cloud device management solution. (see pic).

All this can take place with no manual action on the part of the device installer or user, unless ownership of the device later changes, and for example, its IoT data needs to be assigned to a different account.



Learn more at www.intel.com/securedeviceonboard

IOT IDENTITY MODELS COMPARISON

	Key Strength	Provisioning	Key Distribution	Cost Model	Trust Model	Privacy	Scalability	Static vs. Dynamic	HWROT
X509 RSA under PKI (1024-bit to 2048 bit)	80-112	SW	TLS	$/KEY	HW + SW	○	1/Device	S	N
X509 ECDH in HWRoT (256-bit to 384-bit)	128-192	HW/FW	TLS (ATTEST depends on HW)	$/KEY	HW	○	1/Device	S	Y
Physically Uncloanable Functions (PUFs)	Variable	HW		$/DEV	HW + ???	◖	1/Device		Y
e-SIM Card	Variable 80-192	FW	TLS/IKE	$/DEV	HW+ SW	◖	1/SIM	S	Y
Device Identity Composition Engine (DICE) using SHA-256/SHA-384	128-192	FW	ATTEST	$/DEV	HW + SW	○	1/Device	S	Y
Enhanced Privacy ID (EPID 2,0)	128	HW	ATTEST	$/DEV	HW	●	1/Group Group = millions of devices	D	Y

Note- As described previously, EPID delivers a new private onboarding capability that is effectively a new best practice for IOT. EPID is compatible with usage of other identity types listed in the table, post onboarding event.

- **Key Strength:** Number of bits of security, normalized to symmetric key strength, where 128 bits is considered minimum security
- **Provisioning:** How are the secrets/keys initially provisioned into the device; HW Manufacturing (HW), Device System Firmware (FW), Software/After-Market (SW)
- **Key Distribution:** Mechanism and complexity of distributing keys during an IAA (identity-authentication-authorization) exchange; mechanisms include standards [TLS, IKE, ATTEST], and complexity from 1 (simple) to 5 complex
- **Cost Model:** Based on the type of expenditure –cost per exchange ($/EXC), cost per key in use ($/KEY), or a cost per device ($/DEV)
- **Trust Model:** What must be (implicitly) trusted to trust the use of the key/mechanism for IAA; hardware manufacture, device firmware and software; all solutions include some kind of infrastructure (PKI or attestation infrastructure), and all include some type of cryptographic proof/attestation (RSA, elliptic curve, EPID, symmetric encryption or hash.
- **Privacy:** What privacy is afforded the user/device: None ○, Partial ◖, Full ●.
- **Scalability:** What is the impact to backend infrastructure and verifiers? How are keys/secrets tracked by backend? How many secrets are required to be kept by the infrastructure to perform IAA with the devices?
- **Static vs Dynamic:** Is solution static (S) meaning keys never change; or is solution dynamic (D) meaning keys are/can be regenerated during operation?
- **Onboarding:** Can be used to onboard/provision a system; What protocols/service is used?
- **HWRoT:** Is/Can be tied to a hardware root-of-trust? (Y/N)
- **References:** "Commercial National Security Algorithm Suite and Quantum Computing FAQ", January 2016. https://www.iad.gov/iad/library/ia-guidance/ia-solutions-for-classified/algorithm-guidance/assets/public/upload/CNSA-Suite-and-Quantum-Computing-FAQ.pdf

ABOUT THE AUTHORS

Steve Hoffenberg

Steve Hoffenberg is a leading industry analyst and market research professional for Internet of Things technology. He has more than two decades of experience in market research and product management for technology products and services. Prior to joining VDC, he spent 10 years as Director of Consumer Imaging and Consumer Electronics Research at the firm Lyra Research, where he led industry advisory services providing extensive market research on consumer technology trends, user adoption, market sizing, marketing strategy, and competitive analysis for major consumer electronics manufacturers. Previously, he worked in product management for electronic design companies that developed and licensed embedded digital imaging and audio products. Steve holds an M.S. degree from the Rochester Institute of Technology and a B.A. degree from the University of Vermont.

Contact Steve:
shoffenberg@vdcresearch.com

Chris Rommel

Chris Rommel is responsible for syndicated research and consulting engagements focused on development and deployment solutions for intelligent systems. He has helped a wide variety of clients respond to and capitalize on the leading trends impacting next-generation device markets, such as security, the Internet of Things, and M2M connectivity, as well as the growing need for system-level lifecycle management solutions. Chris has also led a range of proprietary consulting projects, including competitive analyses, strategic marketing initiative support, ecosystem development strategies, and vertical market opportunity assessments. Chris holds a B.A. in Business Economics and a B.A. in Public and Private Sector Organization from Brown University.

ABOUT VDC RESEARCH

Founded in 1971, VDC Research provides in-depth insights to technology vendors, end users, and investors across the globe. As a market research and consulting firm, VDC's coverage of AutoID, enterprise mobility, industrial automation, and IoT and embedded technologies is among the most advanced in the industry, helping our clients make critical decisions with confidence. Offering syndicated reports and custom consultation, our methodologies consistently provide accurate forecasts and unmatched thought leadership for deeply technical markets. Located in Natick, Massachusetts, VDC prides itself on its close personal relationships with clients, delivering an attention to detail and a unique perspective that is second to none.

VDC Research
Insights for the Connected World