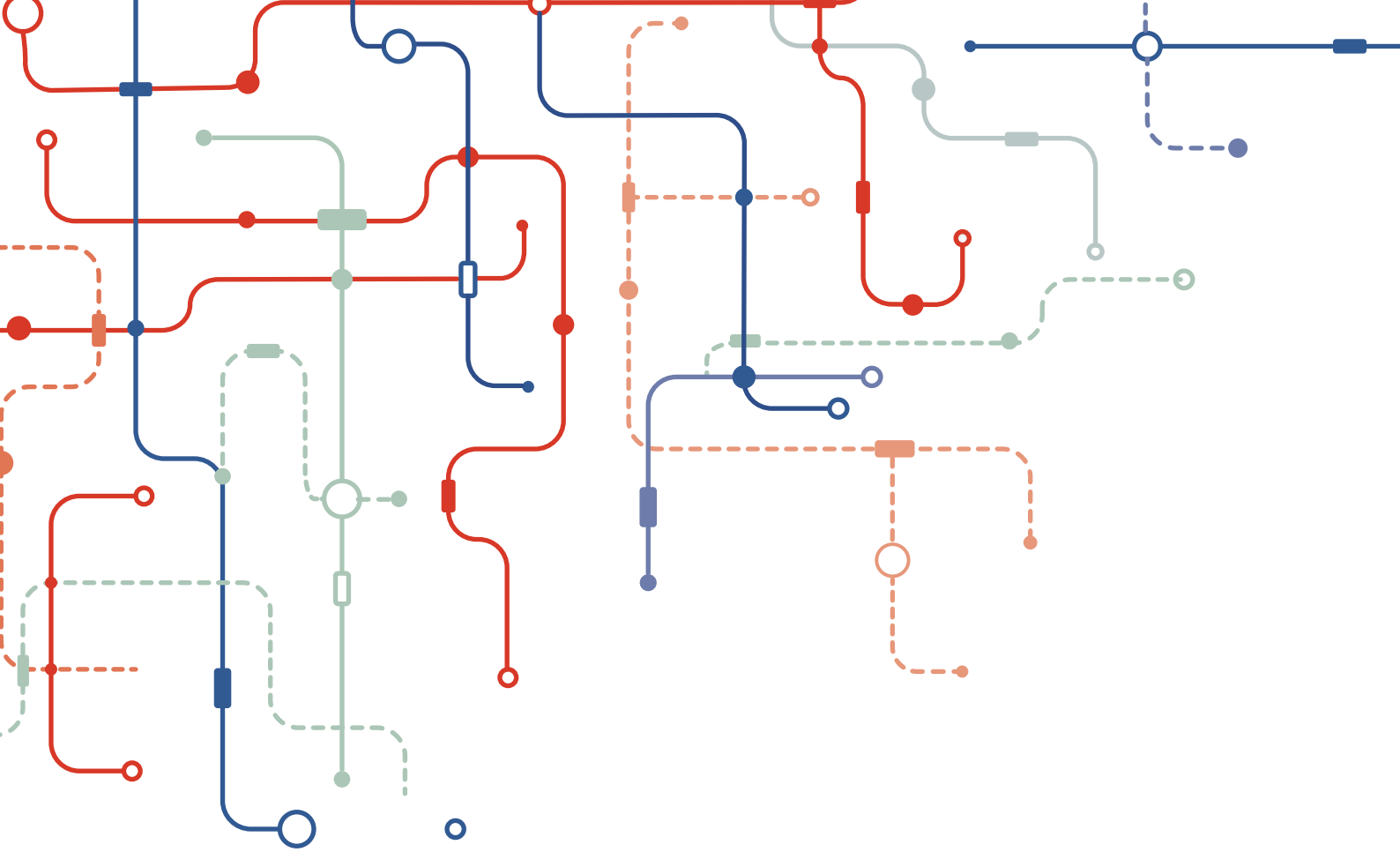


IoT Security Solutions

Connected Objects – Root of Trust

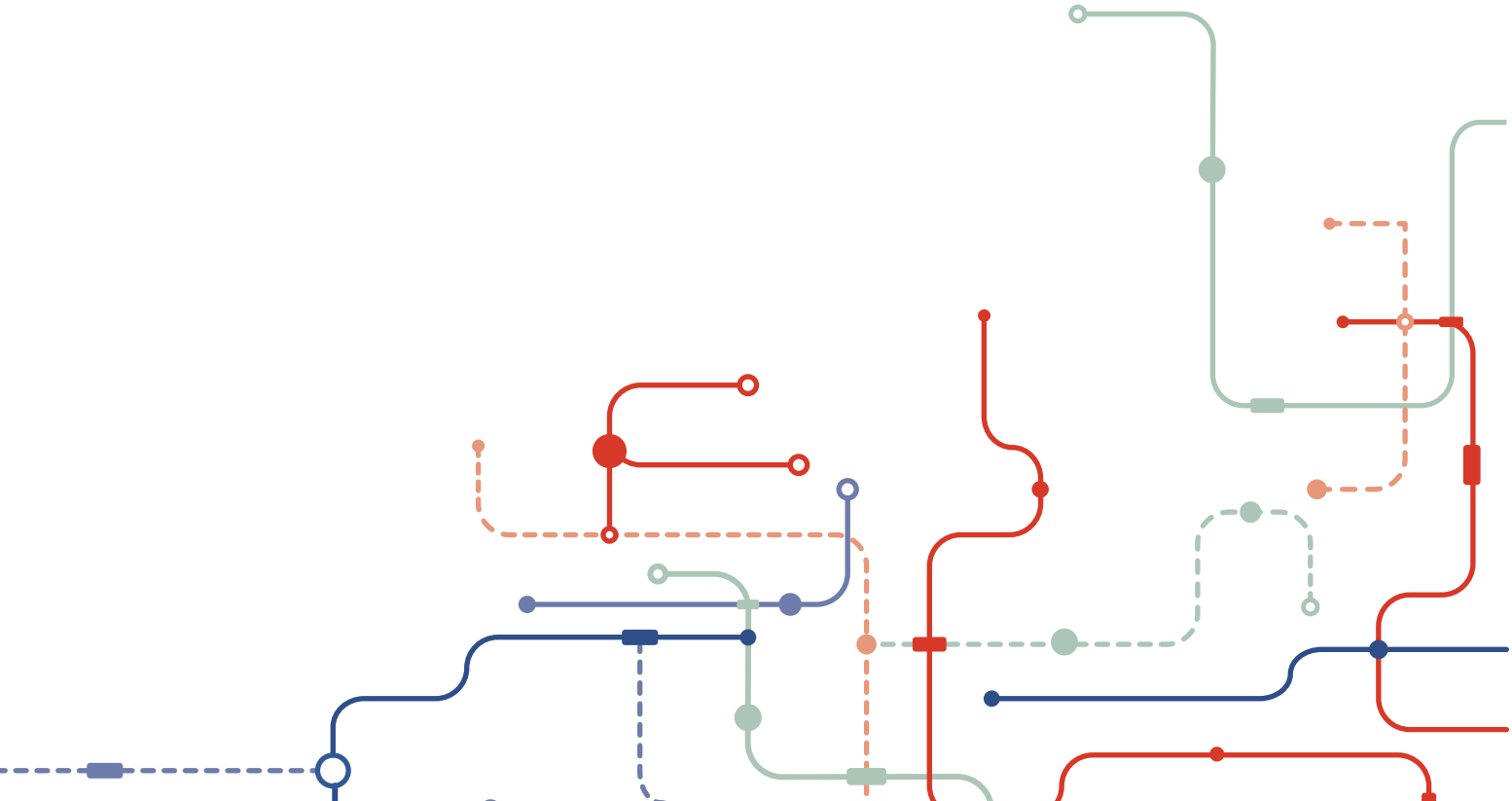


Table of Contents

Executive Summary

The development of IoT, based on the collection and management of large amounts of data, can only happen if data can be trusted. To achieve this, the most important security functions to secure the Internet of Things are:

- Authentication: confirming the identity of the communication peer;
- Secure communication: Protecting the data in transit;
- Secure Execution of code: Protecting the data in process;
- Secure storage: Protecting data at rest

The use of proven technologies coming from Information Technology and the security market, adapted to the IoT, offer the best solutions to secure the IoT infrastructure.

Introduction

IoT New opportunities… and new risks

Technology allows connections for People-to-Machines and Machines-to-Machines, creating new opportunities to improve our lives by optimizing processes and resources, reducing risk towards end users, improving end-user experience and creating or improving businesses. This is the Internet of Things.

But such enormous benefits also bring new security threats that cannot be ignored, such asfake objects interacting in networks, injection of misleading data in industrial setups, eavesdropping of confidential data to mention only a few.

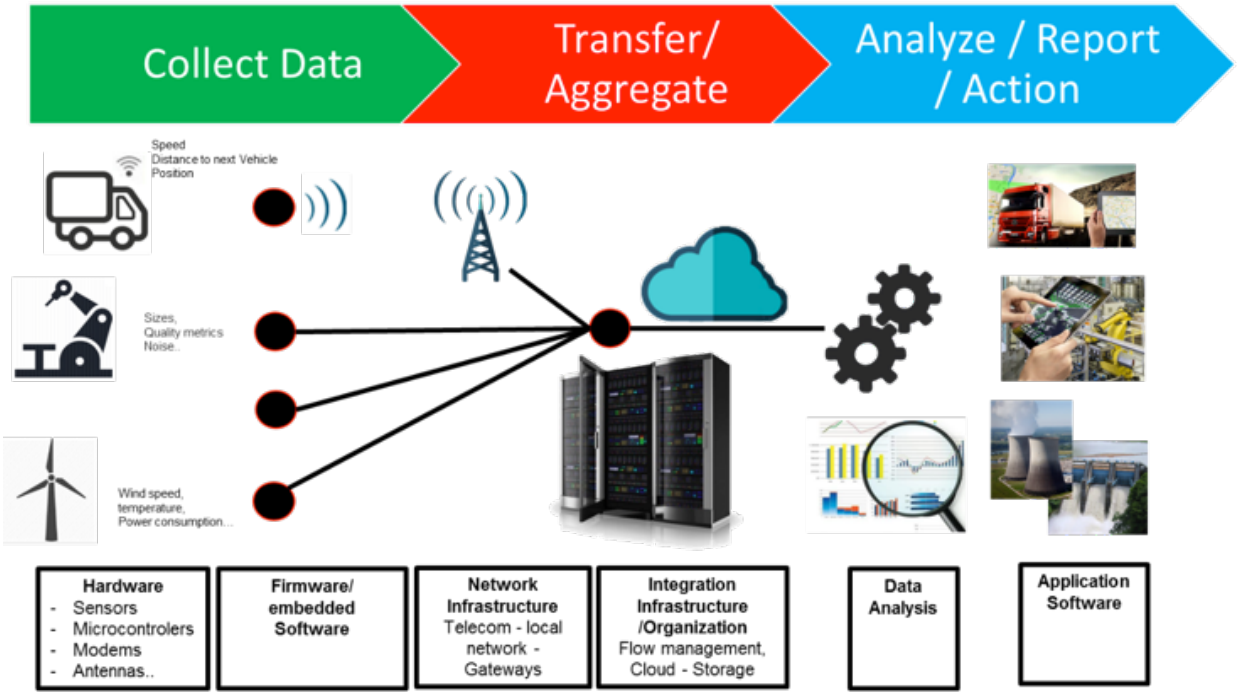


Figure 1: IoT Value Chain

We can simplify the IoT value chain as shown in Figure 1. It starts with the collection of data through a variety of sensors connected to edge devices. The data are transferred to a server, where they will be analyzed and used to create actions.

Recent news reports show that attacks on IoT are not science fiction but real: Distributed Denial of Service (DDoS) attacks -Europol reports that “Distributed Denial of Service (DDoS) attacks are becoming more widespread and more dangerous causing organizations billion dollar losses (The 2016 Internet Organized Crime Threat Assessment (IOCTA)), it demonstrates IoT devices could be the threat entry point for network attack; Men in the middle attacks on video surveillance cameras”; NY Times, November 2016: “Researchers demonstrated a foreseen potential attack switching OFF and ON wireless smart light bulbs, using a drone and exploiting a security flaw in wireless protocol enabling a worm propagating and infecting these LED lights”… and this is not an exhaustive list.

Hackers have different motivations (have fun, get money, terrorisism…) and different resources (material, collusion, expertise, …) to penetrate a system, but all IoT Systems will face hacking. In case of success, consequences can be heavy: stop or disturb services, affect individual’s privacy and safety, theft of intellectual property, damage to brand reputation, loss of revenue and job destruction and so on. The inclusion of security techniques in the IoT is critical to obtain all the benefits of IoT while controlling the risks.

Internet of Things

This white paper covers the security of the Internet of Things architecture: a network of physical objects or "things" embedded with electronics, software, sensors, and connectivity to enable objects to exchange data with the manufacturer, operator and/or other connected devices based on the typical infrastructure architecture.

The Internet of Things covers connected devices and objects over the Internet Protocol (IP) such as personal computing devices (laptop or desk computers, tablets, smartphones) and also devices that are connected to each other through non-IP protocols (e.g. Bluetooth, ZigBee, …).

IoT devices are typically thought of as "smart devices", such as networked home appliances (remotely monitored or controlled), "smart home" components (such as lighting, heating, or ventilation units with remote management/monitoring access), sensor networks for industrial automation, networked vehicle telematics sensors, and a multitude of other embedded devices that are network connected and computationally capable. IoT devices, or nodes, often operate without a screen or even without a user interface at all, may rely on battery power for operation, and are usually dedicated to a single task. We only mention here a few examples'. The range of devices and use cases that are seen as inclusive to the Internet of Things is extremely wide. It stretches from the seemingly trivial (such as toys or entertainment devices) to consumer devices (such as wearable smartwatches and personal health devices) to the clearly mission critical (such as smart energy grid and medical technology), leading to a large variety of architectures of edge devices connected to routers and gateways.

Proliferation of connected devices

Some noteworthy developments have made the roll-out of connected devices possible. The mobile Internet revolution introduced a number of wireless protocols that have now grown as the "over-the-air" extensions of the traditional wired Internet. These wireless protocols – GPRS, UTMS, LTE, WiFi, ZigBee, Bluetooth and new low bandwidth network such as LoRA, Sigfox, RPMA etc. – have spread the Internet coverage to be nearly omnipresent.

For a while, it seemed that only the shortage of addresses could slow down the rapid expansion of the Internet. This key shortage was overcome soon – a key enabling feature of IoT is the dramatic expansion of the IP addressing space brought upon by the introduction of IP version 6 (IPv6) that can guarantee an address for each node.

A second key development that paved the way for IoT is the readily and economically available computational power brought upon by the present level of miniaturization of integrated circuitry. Modern chip design combined with advanced semiconductor technology provides cost-effective and condensed computational capability with sufficiently low power requirements, enabling each addressed node to be capable of computing on a level that was in the past restricted to centralized specialized computers.

The control of connected devices along with their generated data is certainly one of the major challenges to solve during the expansion of the IoT. While the information that is processed by these IoT nodes seems innocuous in itself, the sheer volumes of the collected, processed and transferred information may have serious implications for both public safety and individual privacy. The security risks of the IoT deployments vary a great deal. An attacker that gains illegitimate access to the location data of a certain wearable computing device certainly invades the privacy of the wearer, while an attacker that compromises the security of a smart energy network may be a threat on a national level.

Digital Security (Overview)

Risk Assessment

The increasing number of devices, connected through a heterogeneous infrastructure, increases the risk of attacks. The potential attack of an IoT infrastructure (network or end devices) generates risks, including: losing control of the application, denial of service, off switch (e.g. black-out on smart grid application), losing user privacy, fraud, terrorism and more, with heavy social consequences such as loss of revenue, liability issues, brand damage, harm to people's health issues, job destruction and other tragedies.

Most of end users, you and ourselves, worry about information leakage, but have insufficient ideas about security.

When implementing an IoT infrastructure, a security risk analysis must be conducted in order to evaluate the effect of a successful attack, and define the security solution to be implemented.

As mentioned previously, hackers or malicious users are motivated by various considerations.

The hacker will put in perspective the reward of the hack versus the "cost" and the "risk" of the attack. Time spent to perform the attack, cost of equipment needed to perform the attack (economical barrier), expertise required to perform the attack are good examples of "cost". Some examples of 'risk' includes legal penalty if caught (e.g. fine, prison…). So the security must be sized relatively to the consequence of a hack, not to the value of the device.

The device that the hacker targets may have different levels of accessibility. Can the hacker have a physical access to the device? What level of collusion (level of information of the system) can the hacker get access to? Security must also be sized relatively to the environment where the device is running and its accessibility to the hacker.

It is in most cases faster and cheaper to implement the right security level when designing a system rather than trying to increase the security of an existing system already deployed.

Security is a chain and is only as strong as its weakest link. Hacker will always find the weakest point in the IoT architecture to start an attack. It is therefore mandatory to implement a consistent and well balanced security level in the entire IoT infrastructure. Some portions of the infrastructure may not appear as "critical", because they are not directly involved with critical data, but in fact have to be secured seriously as they could be the access entry point for a much more dangerous hack.

Security is a key business enabler and needs to be built into each solution from the start.

Attacks

IoT architecture and IoT devices are potentially open to a huge number of attacks. Attacks may be performed at different levels. At System Level, the hacker has only access to the network though the applications and or services offered by the IoT solution provider. In the recent past, Distributed Denial of Service (DDoS) attacks put in danger several IoT networks and associated services. At Device Level, the hacker has access directly to the device. He or she can perform some physicals attacks on the device. At Chip Level, the hacker can physically perform attacks (e.g. reverse engineering) on chips located in the device.

Due to the diversity of IoT devices, and the level of maturity of the market, there is no standard document describing the potential attacks or vulnerabilities for IoT. But some standards exist and can provide a good reference for IoT. Common Weakness Enumeration (CWE™) and the Common Attack Pattern Enumeration and Classification (CAPEC™) provide a presentation of software vulnerabilities and attack methods. FIPS 140-2 from US authorities or Common Criteria developed by the Smart Card industry also provide guidance from experience.

Some examples of attacks are:

Software attacks: the attacker will find and exploit vulnerabilities in protocols, crypto algorithms, or in their implementation to bypass security mechanisms. These attacks may be very efficient and usually do not require sophisticated equipment, and can sometimes be performed through the network without direct physical access to the device.

Exploitation of Test features: The attack path aims to enter the IoT device (or IC in the IoT devices) through using its test mode to provide a basis for further attacks (e.g. disclosure or corruption of memory content to retrieve user data like crypto keys or device configuration.

Attacks on RNG (Random Number Generator): Random Numbers are at the basis of cryptography. Reducing the entropy of a RNG will significantly reduce the security level of the system.

Side channel attacks: Attacks performed based on information (e.g. power consumption, electromagnetic radiations…) gained from a physical system (e.g. Integrated Circuit of an IoT device).

Fault Injection attacks: Intentionally causing error in a system (e.g. clock or power supply glitches…). With such attacks, hackers can disclose an AES key while the non-protected IOT device is encrypting or decrypting a file. To perform such attacks, equipment may cost less than $1000 and require low skill profile, as demo scripts are provided together with equipment available on internet.

Cryptography

Cryptography is the practice and study of hiding information. It is the science used to try to keep information secret and safe (Source: Wikipedia).

We cannot obviously be exhaustive on this subject here. You'll find some basic information about cryptography in this section. You'll also find additional information spread throughout this document to explain some security functions.

Cryptography can be divided in three main functions:

- Authentication: establish true identity;
- Encryption: keep information secret;
- Data Integrity: detect changes.

These three functions can be used individually or combined to create a solution.

Several algorithms have been developed and deployed to perform such functions, they can be divided in two groups:

Symmetric algorithms (sharing a single secret key between the two communicating peers): If the keys become public, the system collapses. The difficulty is the exchange of keys between the two peers. Such algorithms are quite efficient in computing power. DES and AES are typical standard algorithms, usually used to encrypt data, where AES provides a better resistance than DES.

Asymmetric algorithms (using a key pair, private and public). Only the public key needs to be transferred: These are usually much hungrier in computing power compared to symmetric algorithms. Usually used to transfer crypto keys or to perform a digital signature. Typical algorithms are ECC and ECDSA. The RSA algorithm requires longer keys and more computing power for the same [1] security level as ECC and will not be used in IoT.

Messages encrypted with a public key can only be decrypted using the associated private key. Thus only the private key owner is able to read the message.

Messages encrypted with a private key (typically called 'signed' with the private key) can be decrypted only by using the associated public key. Successful decryption of the message using the communication peer's public key assures that only the peer can have sent the message (unless of course the private key has leaked).

Some important messages related to Cryptography:

Cryptography is robust against brute force attacks only if the right standard algorithms are selected, and appropriate key lengths are used (e.g. NIST recommendations). The use of private algorithms is not proven and could lead to potential disaster if a weakness is found.

Robustness of cryptography is based on a secret ("Secure Key"). If the secret is found the security is compromised. This mean that this "Secure Key" must be generated, loaded and managed securely through personalization, or provisioning during the device's entire life cycle. It also means that this "Secure Key" must be stored securely inside the device, to avoid being captured by a hacker. Therefore it is recommended to store this "Secure Key" inside a tamper resistant chip. A tamper resistant component will ensure that all the secure assets including crypto keys, application data and personal data, are stored securely, and will not be revealed at rest or in use (during the execution of cryptographic algorithms).

A tamper resistant environment is mandatory if the device is used in a hostile environment or could easily be cloned to introduce other devices.

Cryptography is a robust and proven technology if it is used with standard algorithms and appropriate key sizes. Some national agencies provide recommendations of algorithms and key sizes*.

It is highly recommended to use standard and proven algorithms as opposed to private ones. The reason is that the standard algorithms have been extensively analyzed and tested. The use of private algorithms expose the user to potential vulnerability that is easy to exploit.

On chip key pair generation with a key generator using high quality True Random Number Generator (TRNG) Secure key storage inside the security module (please refer to the specific Secure Data Storage section in this document)

Security functions for Internet of Things:

Regardless of the use case, energy/smart grid, industrial automation, connected home, wearable computing, etc., the IoT nodes share common basic security needs which stand as pillars of security:

- **Authentication:** confirming the identity of the communication peer
- **Secure Communication:** Protecting the data in transit
- **Secure Execution of code:** Protecting the data in process
- **Secure storage :** Protecting data at rest
- **Non repudiation:** A service that provides proof of the integrity and origin of data.

Cryptography is at the foundation of these pillars. Cryptography is used to authenticate, encrypt/decrypt information or sign and check integrity of data

Hardware and software technologies are available to protect the data at rest in process and in transit.

Authentication

Authentication is the process in which the communicating peers identify each other and assure each other of their identity. The IoT

1 The NIST provides guidelines for selecting different crypto algorithms and associated key sizes within the SP800-175B publication.

deployments imply a vast number of interconnected and distributed endpoints that need to communicate, which highlights the importance of the strength, reliability and scalability of the authentication methods used. Each end point must be controlled to make sure each peer is genuine and to avoid insertion of fake devices into the network.

The most common (and simple) form of authentication is based on shared secrets using symmetric algorithms. But remember, this methodology is highly vulnerable, because if the shared secret is compromised the entire security collapses.

IoT and its billions of connected objects is bringing another problem that symmetric crypto cannot solve: how to securely store shared secret keys. These shared secret keys have to be different to overcome the issue of the secret being disclosed (a Key revealed by a hacker on one specific device should not compromise the security of other devices). Keys have to be stored locally (e.g. a gateway communicating with a number "n" of IOT devices has to store "n" secret keys) or remotely (e.g. gateway has to request the IoT device secret key from a key repository, but the key exchange has to be protected as well). Key storage generates a security flaw. Asymmetric cryptography is solving this issue.

Public Key Infrastructure (PKI)

"A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption. The purpose of a PKI is to facilitate the secure electronic transfer of information for a range of network activities such as e-commerce, internet banking and confidential email. It is required for activities where simple passwords are an inadequate authentication method and more rigorous proof is required to confirm the identity of the parties involved in the communication and to validate the information being transferred.

In cryptography, a PKI is an arrangement that binds public keys with respective identities of entities (like persons and organizations). The binding is established through a process of registration and issuance of certificates at and by a certificate authority (CA). Depending on the assurance level of the binding, this may be carried out by an automated process or under human supervision." (Source Wikipedia)

A cryptographically secure authentication method is the use of asymmetric, public key cryptography. Public key authentication is commonly used in internet connected servers and devices to provide strong authentication, and an excellent solution for IoT: the digital certificate will be used to authenticate devices during the entire device life cycle. Only trusted devices can connect to secure networks.

Digital certificates, for instance SSL certificates, can also be used to secure communication channels from devices to gateways/routers, and from gateways/routers to servers.

PKI is applicable to a wide variety of secure communication protocols (IPsec, TLS / SSL, DLTS) in a standards compliant way.

Authentication and Root Keys

The challenge in communication is how to create an initial trust, i.e. how can you trust that the received public key belongs to the intended communication peer?

A peer needs to trust that a public key it uses to verify a signature:

- belongs to the device it communicates with, and the device possesses the related private key

This invariably requires the storage of some public key, in the device. The key must be immutable – it must not be possible for an attacker to change the key, or cause the device to use another key instead. This key is referred to as the 'root of trust' for the specific key hierarchy. Sometimes it is needed to combine the public key of a device/user, with other information to identify the device (IP address, domain name, real name and address, etc.). That's what digital certificates are for.

Authenticated Key Exchange

Digital Signature Algorithms (DSA) are an application of a public key crypto system to allow authentication and integrity detection of messages. DSAs require high quality randomness to remain secure and devices should have control over what they sign.

Digital Signatures for Device Authentication

Digital Signatures are always related to both communicating parties showing proof that they have or know a particular secret.

A 'secret' can be a symmetric key (e.g. HMAC key) allowing both parties to sign a challenge that requires possession of the secret HMAC key to complete. Typically symmetric keys are used in cases where public-key cryptography is considered too complex. This provides a system where there is no 'individual' authentication of a device or entity (both or all peers share the same key). The shared key is used to show 'membership' of a common ecosystem. If the secret is ever compromised, all devices using the same key are compromised.

More often, the 'secret' is an asymmetric private key, which is used to sign a challenge. The associated public key is then used to

verify the signature. More complex arithmetic, requiring more time and more power, provides a finer grain security framework. Each device has its own unique private key.

Digital Signatures can be used for offline authentication (e.g. by signing boot code) and for Online Authentication (signing challenges).

Protecting data in Process

The code executed by an IoT device must behave as expected, not changed, modified or corrupted. This is even more important when the code is manipulating security assets, like cryptographic keys, or application-related data. Performing cryptographic algorithms thus requires strong protection against attacks. The key must remain secret and unchanged during the full process of cryptography (including the loading of the key), and the code executing the crypto function must behave as expected.

The code can be protected in two ways: either it is executed within a secure or trusted environment, or can be protected using software tools.

Secure or trusted execution environment can be provided either by a security chip or by an isolated trusted or secure environment built-in the IoT device Host SoC/CPU. Figure (2) provide a summary of those different security implementations.

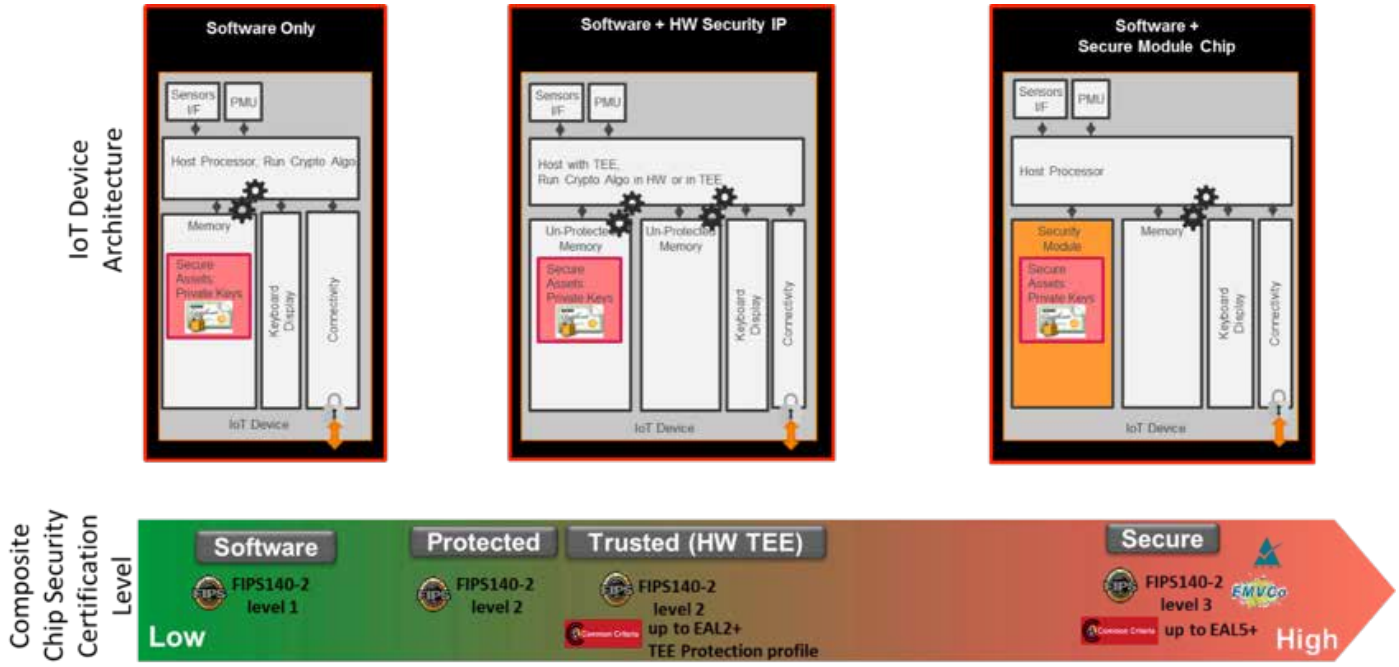


Figure (2) Different security implementations.

i/ Pure Software Solution

The first solution consists in adding security without additional hardware features. It can start by the use of cryptographic mechanisms (algorithms, PKI…) and keys implemented in the firmware of the device. The vulnerability of such a solution is critical. It is possible to protect the code of the IoT device and data using software obfuscation mechanisms. The code must be developed following good security practices (see OWASP Secure Coding Practices), and the execution code must be processed through a software development tool that provides improved resistance against different types of attacks. For example, some tools provide protection against debuggers and memory dumpers (referred to as software obfuscation) and an additional layer of security by hiding, partitioning, and re-arranging data as it is handled by software.

In the IoT node, software obfuscation can be used to protect security components that are executed within the general purpose operating system of the host. This will provide an additional layer of security and add some defenses, but will not provide the same security level as a hardware solution.

ii/ Solution based on Trusted Execution Environment

Some of the standard microcontrollers or System on Chip IC (SOC) used in IoT devices feature a Trusted Execution Environment (TEE). The TEE is a hardware feature to create two environments (standard and secure) and an isolation between these two. This

isolation provide an additional level of security compared to “pure software implementation”, depending on the robustness of the isolation.

iii/ Solution based on companion chip: Secure Element

Security Chips (also called Secure Elements or Security Modules) usually have designs derived from the Smart Card Industry and can be inserted as a companion chip in IoT device architectures. They offer state of the art security and can be certified according to Smart Card industry standards such as Common Criteria. This security implementation can offer a good solution for medium size volume applications as they are standard products and do not require any specific IC development.

It has to be noted that when a secure element is used to secure an IoT device, it is preferable to create a secure binding between this secure element and the host CPU of the IoT device, in order to conceal the data that flows between the two.

Secure Bootstrapping

Another important aspect of protecting data in execution, and to ensure that the device runs the software in the way its manufacturer or deploying organization has intended it, is by assuring a “secure boot” or “secure bootstrapping” mechanism.

A protected device will be allowed to boot only software images or accept data that is endorsed or signed by the manufacturer (or a trusted 3rd party). In general terms this can be realized by a hardware assisted boot process in which the images are verified by hashes (data integrity checks) and digital signatures prior to executing them at boot time. Implementing secure boot as part of a device architecture is closely connected to other device management and maintenance tasks, and should be considered as a part of a bigger picture. The secure booting procedure needs to be compatible, for example, with remote (and possibly “over the air”) device firmware upgrade mechanisms, to allow device software images to be altered post-deployment.

Secure Communications – Protecting Data in Transit

As their name implies, the IoT devices leverage existing Internet technologies and protocols. These protocols provide a selection of proven secure communication solutions. The actual use case of a node typically dictates the criteria for selecting the leveraged secure communication protocol. For example, in Smart Grid deployments (where the communication model typically follows the client-server model) protocols such as SSL/TLS or DLTS are used, while in full mesh-type networks a network level security protocol (such as IPsec) is more applicable.

All three security protocols that are mentioned above can use the earlier introduced and proposed PKI authentication in a standardized way. This guarantees the IoT nodes’ seamless integration and interoperability with the existing standard network infrastructure.

While the data rates to and from an individual node are typically limited in IoT deployments, the constrained nature of the host devices may pose implementation challenges in the context of cryptographically secured communications. To limit the amount of CPU cycles used for cryptography on the general purpose processing unit of the host, it is likely beneficial to offload the symmetric and asymmetric cryptographic algorithms to specialized silicon. When executed on hardware the common symmetric encryption algorithms (such as AES and 3DES) and asymmetric algorithms (such as RSA and ECDSA) are faster and more power efficient (important in power constrained and battery powered devices) compared to software implementations.

Secure storage - Protecting Data at rest

While data are protected during their manipulation, stored data must also be protected.

There are two main categories of data that must be protected during storage:

- Encryption key (the private key of an asymmetric key pair) and unique device identification. This key is used as a trust anchor for a secure system. From this root of trust, derivative and session keys will be generated to authenticate and securely communicate between peers. If these keys are divulged, clones of the device can be created and communication can be decrypted.
- Personal data that belongs to the application that must be protected to warrant its privacy.

In an IoT device these data are stored in variety of media, but essentially in memory devices offered by the semiconductor industry (OTP, ROM, RAM, Flash, EEPROM…). Data can be protected by encryption, by being stored in a tamper resistant device, or by a combination of both. In any case the access to it must be carefully controlled and only granted to authorized persons, machines or processes.

WISeKey security solutions for IoT

WISeKey is a leading cybersecurity company and has been selected as a World Economic Forum Global Growth Company. WISeKey is currently deploying large scale Internet of Things (“IoT”) digital identity ecosystems and has become a pioneer of the “4th Industrial

Revolution" movement launched in 2016 at the World Economic Forum in Davos. WISeKey's Swiss based cryptographic Root of Trust ("RoT") serves as a common trust anchor, which is recognized by the operating system and applications, to ensure the authenticity, confidentiality and integrity of on-line transactions. With the cryptographic RoT embedded on the device, the IoT product manufacturers can use digital certificates to secure interactions among objects and between objects and people. WISeKey has patented this process in the USA as it is currently used by many IoT providers.

As a main player of the Internet security market and a leading security technology vendor WISeKey is in a unique position to offer a solution that can cover all aspects of a secure IoT infrastructure. WISeKey solution components are mature, robust, and field-proven in some of the most demanding markets. This experience and expertise is readily available for the IoT market.

The methods, technologies, and architectures described in this overview document can be realized with the offering, expertise, and experiences of WISeKey engineering.

WISeKey Solution for IoT: WISeTrustIoT

WISeKey provides an end-to-end scalable security framework, to be integrated into IoT platforms, based on PKI technology to protect the data at rest, in transit or in process. Digital Certificates will be used to authenticate devices during the entire device life cycle. Only trusted devices can connect to secure networks.

WISeKey is a provider of trusted cryptographic root keys and proposes products and services to use and manage digital certificates. WISeKey WISeTrustIoT framework can be "on premises" or offered as a Cloud Service.

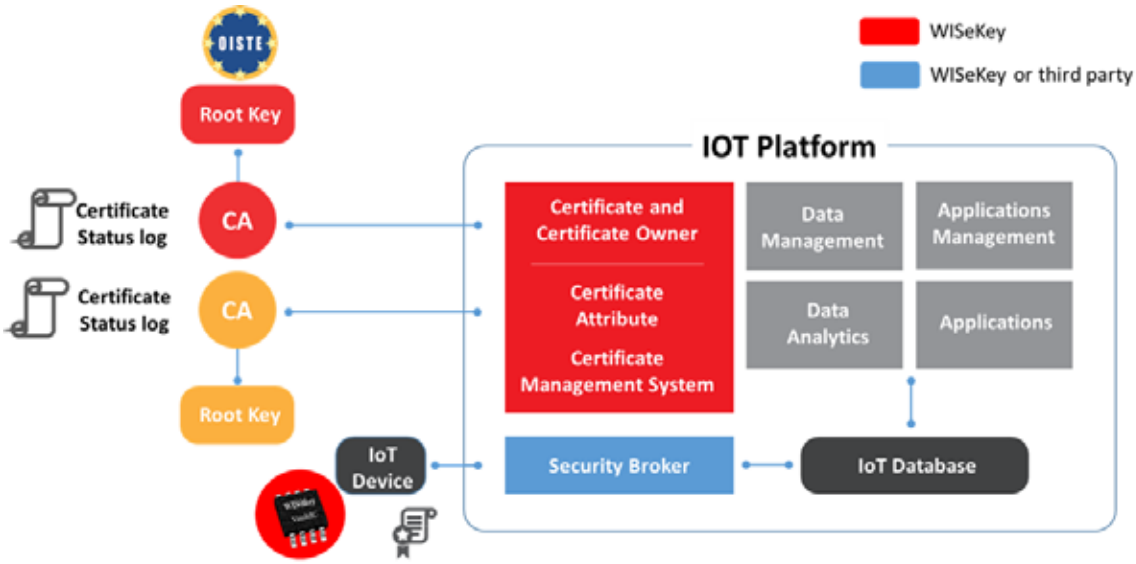


Figure (3) WISeKey WISeTrustIoT security Framework

Root of Trust

The Root of Trust (RoT) serves as a common trust anchor, which is recognized by the operating system (OS) and applications, to ensure the authenticity, confidentiality and integrity of on-line transactions. The combination of Cryptographic Root of Trust embedded in the device, together with its own unique identity, can secure the interactions among objects and between objects and other entities.

At the heart of this strategy is the OISTE-WISeKey Cryptographic Root of Trust which has been actively used since 1999 by over 2.6 billion desktop, browsers, mobile devices, SSL certificates and Internet of Things' devices. The OISTE WISeKey Cryptographic Root of Trust is ubiquitous and universal, and a pioneer in the identification of objects.

WISeKey has a unique positioning to offer a consistent security system from the RoT to back-office. WISeKey is the trusted operator

of the International Organization for the Security of Electronic Transaction (OISTE) Global Root. The OISTE Foundation is working with the UN and International Organizations since 1999 in line with the United Nation's Sustainable Development Goals which include giving everyone a legal identity by 2030. Swiss neutrality, security, and privacy laws allow operations without geopolitical or governmental constraints. The OISTE RoT is the only one available outside NATO. It is located in a military grade bunker deep in the Swiss Alps.

Certificate Authority

Based on Trusted Root Keys, the Certificate Authority (C.A.) creates a digital certificate for each individual device which uniquely and strongly identifies the device. A C.A. can also generate digital certificates to secure communications or control applications.

WISeKey can support a corporate Sub C.A. under WISeKey's Root of Trust model

The WISeKey Certificate Management System is also compatible with third party C.A. based on Microsoft or Enterprise Java Beans Certificate Authority (EJCBA) open source C.A.

Certificate Authority and Certificate management

WISeKey provides an enhanced technology platform that combines our experience in PKI and certificate management solutions with innovative features that address the new needs of the Internet of Things. WISeKey provides open interfaces that allow integrating the manufacturing process of the objects with the issuance of their identities, along with programming interfaces that simplify the usage of the new identities when authenticating the connected objects and the data that is transmitted across the network. The full lifecycle of the objects, their identities and the data can be easily managed with the new IoT platform provided by WISeKey.

The WISeKey Certificate Management System (CMS) is a software tool with a user-friendly interface that allows the complete management of the digital certificates during the entire life of the IoT device, from manufacturing to the field.

The status of the certificates can be checked or changed at any time by the administrator. They can, for instance, be remotely activated, revoked, or frozen.

The optional message broker allows for authenticating and validating messages coming from different IoT devices, and transfer only trusted messages to the IoT platform of our customer. The WISeTrustIoT framework can be integrated into a customer's IoT platform easily, as the customer does not need to implement additional security mechanisms. When the Message Broker is installed on customer premises, the solution offers end-to-end security from the IoT device to the customer's back-office.

WISeKey CMS can be installed on customer premises, but for the ones not willing to deploy their own infrastructure, WISeKey can provide trusted services from any of its local secure datacenters in Switzerland, USA, India or China. The managed platform can be accesses through a browser and a web-service API.

Provisioning

WISeKey also offers solutions to control a device's firmware integrity at initial stage (bootloader) and during upgrades in the field.

The WISeKey CMS also includes secure provisioning solutions to help maintain consistently high system security, even when the IoT device is in an unsecured environment, such as at contract manufacturers or in the field. Device configurations and firmware upgrades are made easy and secure at any time.

Implementation of digital certificates and security assets in IoT devices

Implementation in Security Modules/Secure Element

This is the recommended implementation to reduce the risk of success of potential hacks.

For the best security, digital certificates can be loaded in tamper-resistant secure chips on WISeKey's secured premises. These chips are then integrated in the IoT devices to become the concrete hardware foundation upon which the security subsystem is built.

VaultIC is a product family of tamper-resistant chips which can be used as a companion chip to the IoT-device host processor.

A secure element like VaultIC is physically separated from the general purpose computing resources and peripherals, and exists only for the purpose of performing its pre-programmed security routines.

Once key material is generated and enrolled into a PKI with the CMS platform, the secure element protects the key material (secrets, private keys, entity and CA certificates) from compromise and modification. Access to keys is restricted, and the applications on the host node are provided access to the security functions over standardized APIs (for example the multi-platform PKCS#11) that provide applications with the functionality needed, but never disclose the sensitive keys outside the secure confines of the Secure Element hardware.

Due to their nature as “mission critical”, some IoT use cases (automotive, medical, or smart energy) face stringent regulatory requirements. In these use cases, the quality of the implementations is typically demonstrated with public certifications. The VaultIC is highly tamper-resistant to avoid logical and physical attacks and cloning. It is certified to FIPS 140-2 Level 3 for the complete product, and the hardware alone is verified to Common Criteria EAL4+/5+. The VaultIC low-power consumption profile makes it a viable solution to meet the limited power budgets of the embedded IoT nodes.

VaultIC embeds configurable cryptographic tool boxes for Authentication, Confidentiality and Integrity executed in a secure environment.

The VaultIC security module includes a rich set of pre-programmed cryptographic services and functions that are executed in a hardened environment. The VaultIC cryptographic services provide:

- The low level cryptographic methods and algorithms including NIST recommended algorithms, needed for authentication and data encryption/decryption:
  - Symmetric algorithms : 3DES, AES with chaining modes and modes of operation
  - Asymmetric algorithms : RSA, ECC
  - Security functions:
    - Signature generation, verification : DSA, ECDSA
    - Key agreement (use case example TLS initiation): ECDH
    - Message authentication: MAC, CMAC, GMAC
    - Secure channel establishment
  - High Quality True Random Number Generation to guarantee the entropy needed for high-level cryptographic services (not achievable in software).
    - The importance of generating randomness of high quality cannot be over-stressed in the context of embedded device security. Some very prominent cryptosystem failures have been traced back to poor random number generation implementation. True random number generation on embedded systems is a genuinely difficult task, yet true randomness is vital for high-quality generation of encryption keys. The strength and quality of the random number generation is in direct relation to the strength and quality of the cryptographic security system.
  - On-chip Asymmetric key generation provides the highest security level for key storage, as the private key is known by VaultIC only and never exposed outside the VaultIC.
- The secure storage capabilities that allow hardware secured storage of essential data items (private keys, CA and user certificates, user credentials, and configuration data).

Secure communication

VaultIC comes with middleware which includes secure boot, secure firmware update for IoT devices and a secure communication (SSL/TLS) stack.

The SSL stack provided by WISeKey is a compact, low-footprint SSL library for embedded devices. For IoT nodes, the SSL Stack provides an extremely well optimized secure sockets layer (SSL) implementation that requires a fraction of the code and memory required by OpenSSL, the most common implementation of this protocol. The SSL stack is interoperable with any standards-compliant SSL server, and has a modular design, so that customers have the freedom to deploy only the desired extensions to the base SSL standard.

Conclusion

The Internet of Things is a new and emerging technology market, with a large scope of requirements and needs in term of security. The security requirements are similar to those in networked devices, embedded devices or security tokens and Smart Card devices but needs to be combined together to match each specific use case. The security design of an IoT node can greatly benefit from the experience gathered in the fields of platform, network security, and highly sensitive applications such as payment or financial transactions, government related applications and content protection.

WISeKey is the only player in the market that can offer the end-to-end solution from Root of Trust generation to implementation and management of digital certificates in IoT devices or servers.

WISeKey helps its customer to become more competitive, providing robust and proven software and security chip products that help to reduce time and cost for product launch and accelerate the business growth.

Glossary and References

AES	Advanced Encryption Standard: Cryptographic Symmetric Algorithm
CA	Certificate Authority, entity that issues digital certificates
CPU	Central Processing Unit
DES	Data Encryption Standard: Cryptographic Symmetric Algorithm
DDoS	Distributed Denial of Service (DDoS). Please refer to dedicated WISeKey White Paper
DH	Diffie–Hellman. Specific method of securely exchanging cryptographic keys
ECDSA	Elliptic Curve Digital Signature Algorithm. Cryptographic Algorithm based on ECC used for digital signature.
ECC	Elliptic Curve Cryptography: is an approach to public-key cryptography
EEPROM	electrically erasable programmable read-only memory. A type of non-volatile memory.
EJCBA	Enterprise Java Beans Certificate Authority
ESTI	European Telecommunications Standards Institute
FIPS 140-2	Federal Information Processing Standard, Publication 140-2. A U.S. government computer security standard used to approve cryptographic modules.
GPRS	General Packet Radio Service. A mobile data service on the second and third generation cellular communication defined by ETSI
HMAC	keyed-hash message authentication code (HMAC) is a specific type of message authentication code (MAC) involving a cryptographic hash function (hence the 'H') in combination with a secret cryptographic key.
IoT	Internet of Things
IPv6	Internet Protocol version 6 is the most recent version of the Internet Protocol (IP), http://www.rfc-editor.org/rfc/rfc2460.txt
LTE	Long-Term Evolution. A standard for high-speed wireless communication for mobile phones and data terminals defined by ETSI
NIST	The National Institute of Standards and Technology. A USA measurement standards laboratory, issuing the FIPS 140 Publication Series
OITSTE	Organization for the Security of Electronic Transaction https://oiste.org/
OTP	One Time Programmable Memory. A type of non-volatile memory.
OWASP	Non profit foundation. https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide
PKI	Public Key Infrastructure https://en.wikipedia.org/wiki/Public_key_infrastructure RAM Random Access Memory. Volatile Memory.
PKCS#11	http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-11-cryptographic-token-interface-standard.htm
RNG	Random Number Generator. See appendix of this document
ROM	Read Only Memory. A type of non-volatile memory
ROT	Root of Trust. The foundation for cryptography.
RSA	Stands for Ron Rivest, Adi Shamir and Leonard Adleman: Initial of inventors of this Cryptographic Asymmetric Algorithm
SOC	System on Chip
SSL	Secure Sockets Layer. Secure transportation protocol replaced by TLS
TEE	Trusted Execution Environment
TLS	Transport Layer Security. A secure transportation protocol
TRNG	True Random Number Generator. See appendix of this document
UMTS	Universal Mobile Telecommunications System. The third generation cellular system defined by ETSI
VaultIC	https://www.wisekey.com/products/vaultic/

Appendix - Cryptography Primer

Applications of Cryptography

Confidentiality - Transform data to a form that hides and does not reveal anything of the original form

Integrity - Ensure that any modification of data can be detected

Authentication - Ensure that the communication peer is truly the one it claims to be

Non-repudiation - Ensure that if an event happened between two parties, neither of the parties can deny that it happened.

Cryptographic Algorithms

- **Symmetric Ciphers** are typically used to provide Confidentiality.
  - e.g. AES, DES, 3DES
- Asymmetric Ciphers are typically used to provide Authentication and Key-wrapping.
  - e.g. RSA, ECC
- Stream Ciphers are typically used to provide Arbitrary Length Confidentiality.
  - e.g. RC4, Salsa20, SNOW
- Key Exchange algorithms are used to provide Authorization and for establishing common secrets.
  - e.g. Diffie-Hellman (DH), ECDH
- Cyptographic Digest is used to provide data Integrity.
  - e.g. SHA, MD5
- Message Authentication Code or in other words Keyed Cryptographic Digest is used for Authenticated Integrity.
  - e.g. CBC-MAC, HMAC, CMAC, GMAC

Cryptographic Key Sizes and Key Lifetime

Symmetric	Asymmetric (RSA)	Asymmetric (ECC)
128 bits	3072 bits	256 bits
192 bits	7680 bits	384 bits
256 bits	15360 bits	521 bits

Asymmetric key strength relative to symmetric cryptography keys (source: NIST)

Lifetime	Symmetric	Elliptic Curve	RSA	Example Application
Short	64 bits	128 bits	700 bits	Session key
< 10 years	80 bits	160 bits	1024 bits	Mobile phone
10+ years	128 bits	256 bits	3072 bits	Automotive

What size key is required to protect information for it's full lifecycle

Elliptic Curve Diffie–Hellman (ECDH)

Anonymous key agreement protocol that allows two parties, each having an elliptic curve public–private key pair, to establish a shared secret over an insecure channel, using Elliptic Curve Cryptography. ECDH is a NIST SP 800-56A recommended key establishment scheme.

Authenticated Elliptic Curve Diffie–Hellman (ECDH) - Authentication of the key agreement provides protection against Man-in-the-Middle attacks. Public / Private key pairs are used for signing (ECDSA - Elliptic Curve Digital Signature Algorithm) the key agreement messages.

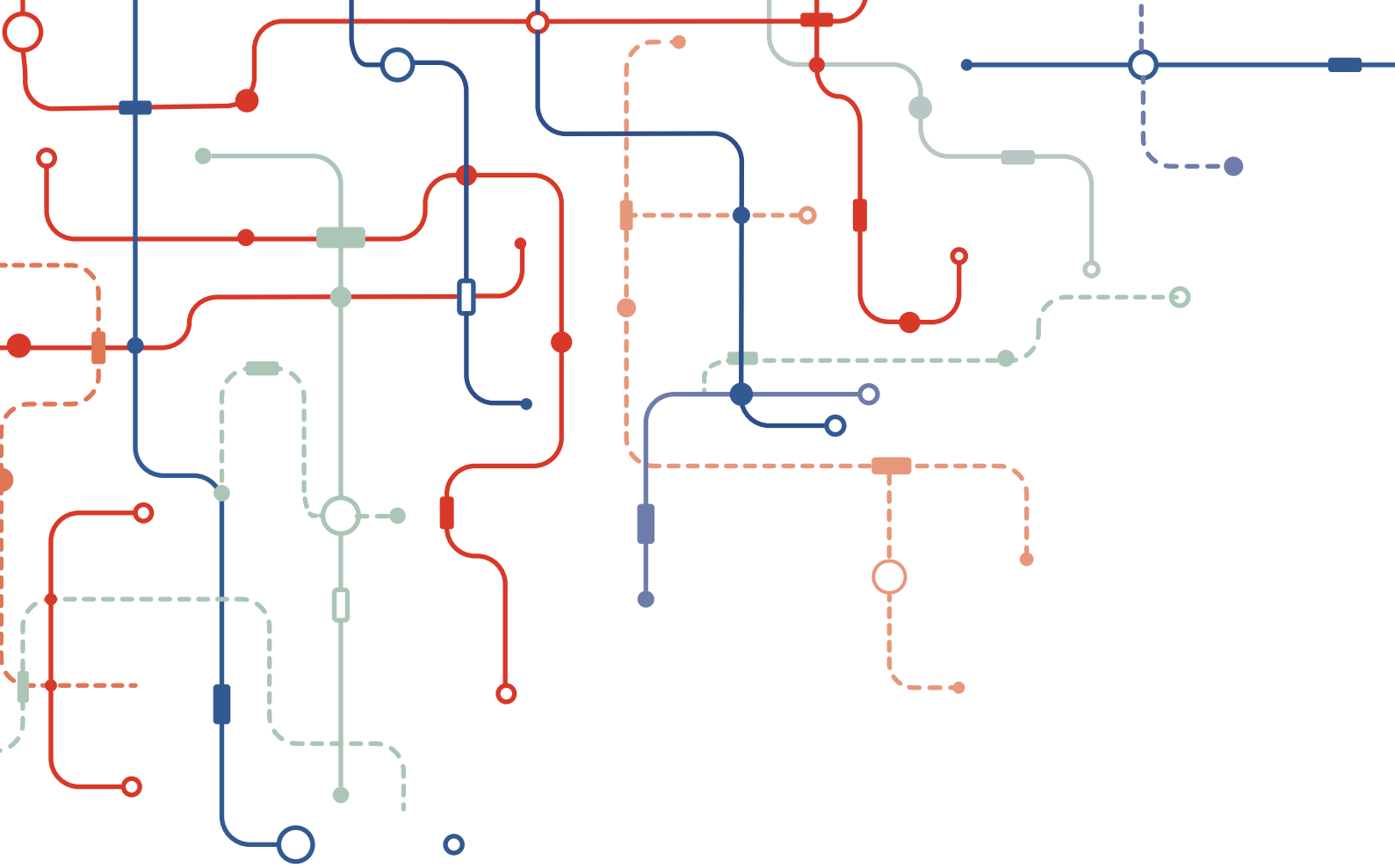
True Random Number Generator (TRNG)

A TRNG cannot be implemented in software and even on hardware it can be challenging to get enough entropy (with acceptable die area).. A truly random number stream is indistinguishable from noise, which means that all numbers in the value space have an equal probability of being generated. And that there is no correlation between the numbers generated by the TRNG.

Random numbers are used e.g. for Nonces (number used once), Key generation (symmetric and public), and Initialization vectors (IV) to initialize an encryption mode (e.g. AES-CBC) so that the first encrypted block is different for identical plaintext.

Even the slightest linearity in RNG compromises its security

“The generation of random numbers is too important to be left to chance” (Robert Coveyou, Oak Ridge National Laboratory)



Contact

WISeKey SA
WTC II
29, route de Pré-Bois · CP - 853
1215 Geneva · Switzerland
Tel: + 41 (0)22 594 30 00
Fax: +41 (0) 22 594 30 01

WISeKey Semiconductors
Arteparc Bachasson · Bât A
Rue de la carrière de Bachasson
13590 Meyreuil · France
Tel : +33 (0)4 42 370370
Fax : +33 (0)4 42 370 024

www.wisekey.com
Stay connected with @WISeKey