Expert insights: Over-the-air updates

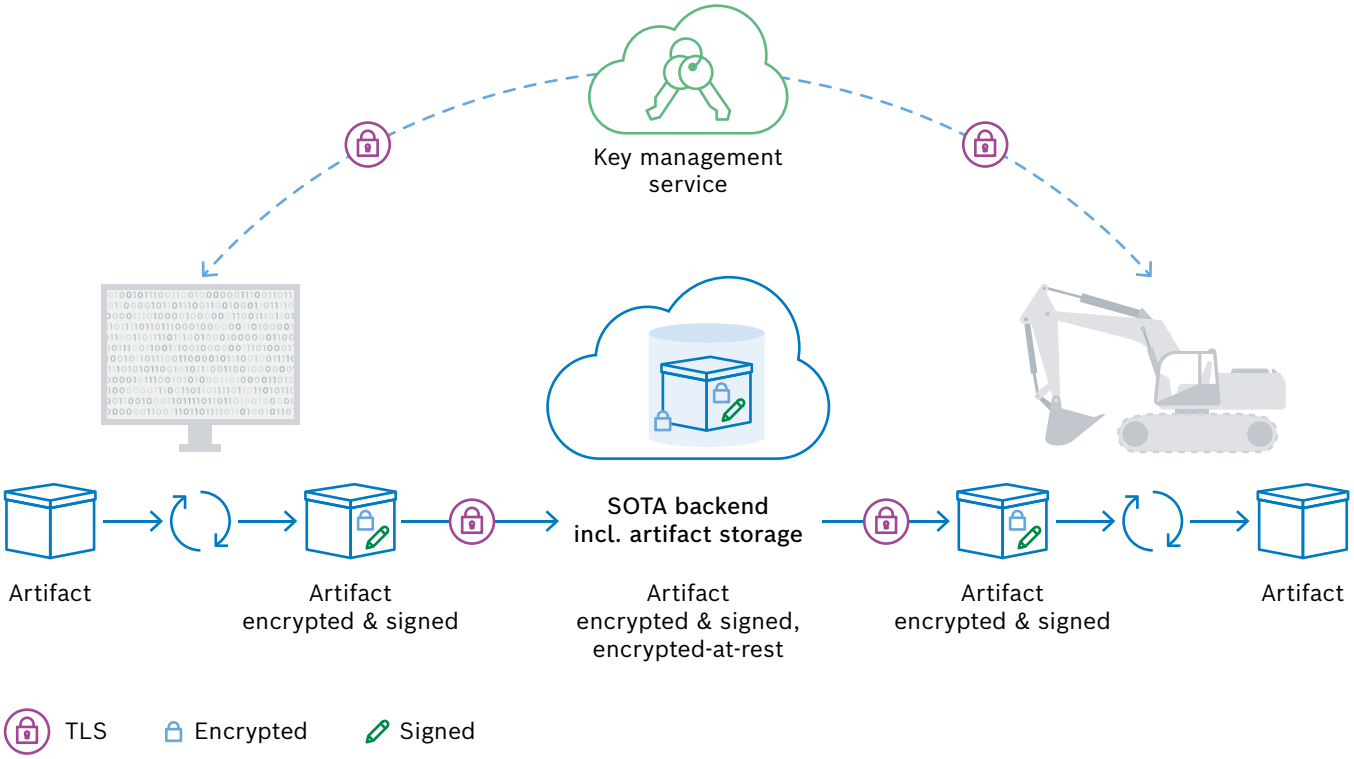
Key considerations to ensure a secure update process

By rolling out updates to devices, companies cannot only mend software issues quickly; more and more manufacturers also understand software updates as a means to prolong a product's life cycle by adding new features. This in turn can form the basis for implementing completely new business models.

Providing software updates over the air (SOTA) makes the whole process much more convenient and efficient. However, this does not mean that SOTA is a simple undertaking from the manufacturer's perspective. There are many steps to consider such as assigning updates to eligible devices, carefully managing large-scale software rollouts, and continuously monitoring update processes. One topic is particularly important – security.

How do companies ensure a secure software rollout from start to finish? Here are three key points to keep an eye on.

1. Security during the artifact life cycle



There are various ways in which a software artifact can be corrupted or compromised, be it technical issues during a file transfer or the interference of a malicious attacker. Companies have to find ways to safeguard the artifacts they provide over the air and ensure their confidentiality, authenticity, and integrity throughout their life cycle, starting with the software artifact's development all the way to its deployment on a device.

To achieve this, a trusted relationship has to be established between the authority that publishes the artifacts and the devices. This is where encryption and digital signatures come into play. These ensure end-to-end security, no matter if an artifact is in transit or at rest.

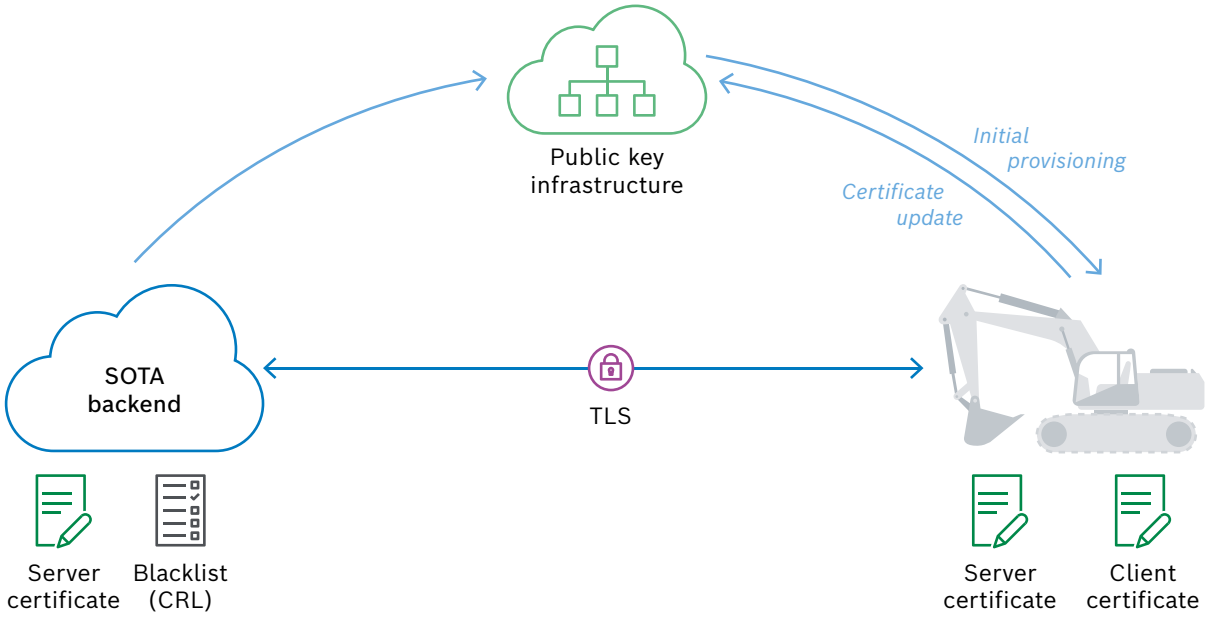
Encryption is used to guarantee **communication confidentiality** and ensures that the messages exchanged can only be read by the sender and the intended recipient – and not by third parties.

Most commonly, communication between devices and the backend application is encrypted using an asymmetric encryption scheme. This approach involves a public key to encrypt messages and a private key to decrypt them. Granted, using symmetric encryption with generally shorter encryption keys makes the whole process quicker, less resource-consuming and more straightforward. Nevertheless, this approach is more vulnerable to security risks due to the nature of keeping the shared key as a secret on both ends of the communication – in particular, this applies to devices that an attacker can physically access to extract the secret.

Looking at the devices, a **digital signature** helps to verify the **integrity of the artifacts** they receive. For this, encryption principles as described above are applied to validate the signature. After successful validation, the package's content can be accessed by applying the decryption procedure.

Especially in large-scale software rollouts, software artifacts do not travel directly from a developer to the device, but are buffered in an artifact storage. To protect artifacts at rest in the backend, the persistence also has to be encrypted to prevent illicit access. In addition, the upload to and the download from the storage must be secured. This is where standard mechanisms such as **transport layer security (TLS)** come into play to ensure **in-transit security**.

2. Secure communication and authentication between devices and the SOTA backend



A central task of a SOTA system is to ensure secure communication between the backend and the devices. It should also enable key management. What do companies have to consider with regard to the technical implementation?

Encryption: public key infrastructure (PKI) and key management system (KMS)
A public key infrastructure is a hierarchical system or concept that defines the certificate life cycle, private/public key management as well as the processes, protocols, and guidelines to distribute them.

The public key infrastructure provides public keys to enable encryption for IoT devices, as well as hierarchical digital certificates. These are used for secure communication, commonly in the form of TLS, or for device authentication at the SOTA backend.

By issuing and governing these digital certificates, the public key infrastructure also addresses a major challenge that comes with asymmetric (and symmetric) encryption: ensuring that the private key actually belongs to the application or device it is supposed to – and not to a man-in-the-middle attacker. Certificates assign devices or applications to specific keys to confirm the identities of devices and backend applications.

How do companies put the idea behind a public key infrastructure into practice? This is where a key management system comes into play. As part of the SOTA solution, this system helps manage encryption keys that are shared between the backend and the device.

Secure communication: transport layer security (TLS)
When establishing an encrypted connection using transport layer security (TLS), the SOTA backend authenticates itself to the IoT device via a certificate. This certificate contains a public key from the backend server, which serves to create a shared secret for encrypted communication.

A device has to make sure that it can trust the backend certificate. Only then will it know that it is communicating with the expected server and not, for instance, with a man-in-the-middle attacker. A certificate authority (CA) has to sign the server certificate to this end. The device then checks if the hostname of the certificate matches the SOTA backend, if the validity period is still in effect, and if it has been signed by a trusted CA.

Trusted server: certificate pinning
The idea behind certificate pinning is to make it harder for bad actors to use certificates in attacks or spoofs and to help reduce the risk of misuse, CA compromise, or man-in-the-middle attacks. In addition, security standards such as OWASP Mobile Application Security Verification require a device-side certificate validation for devices that handle sensitive data (like health data) to grant level-2 certification.

Certificate pinning allows IoT devices to restrict which SOTA backend certificates are considered valid and trusted. A device can then refuse any connection to a server that is not using that certificate. By drawing upon this mechanism, devices do not have to use a trust store for root certificates.

How does this work? The TLS certificate that the server is expected to have is stored on the device during development. Instead of allowing the use of any trusted certificate, IoT devices pin the CA issuer, public keys, or even end-entity certificates of their choice. Clients connecting to that server will treat all other certificates as invalid and refuse to establish an HTTPS connection.

Depending on the use case, certificate or public key pinning takes multiple forms. An IoT device can directly pin the leaf certificate or the public key hash as the only trusted source. This means that connections with this key are established exclusively. The problem with this approach is that server certificates can expire, be exchanged, or revoked, and then the device can no longer connect to the SOTA backend.

A much less error-prone approach is pinning the certificate authority rather than just the server certificate. To do so, intermediate or root certificates within the certificate chain are pinned. This approach makes it easier to deal with certificate changes, while still being secure, as only a trusted CA issues certificates.

Trusted clients: device authentication
Using X.509 certificates is the way to go when it comes to ensuring that only eligible devices connect to the SOTA backend. For authentication, a device sends a complete (self-contained) certificate chain along with the request to the backend, where it is validated. The certificate chain can contain multiple certificates, for example, a target-specific client certificate, an intermediate certificate, and a root certificate.

Communication with a content delivery network
For more efficient packet availability and distribution, companies can leverage a content delivery network (CDN) instead of central artifact storage. In this case, companies should consider using signed URLs. Typically, they appear in a cryptic form to avoid attacks based on their naming structure. Signing the URLs renders authentication at the CDN obsolete because signed URLs contain the authentication information in their query strings, allowing devices without credentials to perform the download. Moreover, a signed URL comes with limited permissions and expires after a defined period.

3. Access management: role-based access control

Managing backend access is a crucial consideration for any software update system, protecting it against unauthorized access. It prevents:

- abuse, theft, or unauthorized removal of data
- misuse of software
- improper alteration or disclosure of information

Going further, role-based access control helps manage users and share responsibilities. It addresses questions such as who has access to the SOTA backend? What are the given users allowed to do? What areas do they have access to?

Thanks to a fine-grained permission scheme, administrators are able to manage the access rights of users and define how they interact with the various features of a SOTA solution. Admins can set up dedicated roles for:

- managing update targets (e.g., by a deployment admin)
- creating and uploading software artifacts (e.g., by a repository admin)
- defining and starting update campaigns (e.g., by a rollout admin)
- viewing reports, investigating customer support cases, and drilling down to troubleshoot problems with failed updates (e.g., by a support agent)

Splitting responsibilities into different roles also makes it easy to implement approval workflows for everything from creating software artifacts and defining campaigns to rolling out the software to the IoT devices.

Conclusion

Security is a multifaceted topic in the IoT and companies have to take a holistic approach to ensure a secure software rollout from start to finish. By drawing on principles outlined above, companies are able to cope with this complexity, reach a high degree of security, and minimize the risk of security threats that can result in breaches and data leaks.

The Bosch IoT Suite is our IoT platform that inherently covers all these principles – complemented by additional approaches that we address in close cooperation with Bosch's security-focused branch Escrypt.

Together with you, we will find the right security approach that fits the needs of your use case. Anything goes, from standard implementations to tailor-made solutions.

Would you like to discuss your use case with us? Contact our SOTA & IoT experts