trust yourself.

Decentralized PKI Solutions

Automotive Industry Focus

Version 1.0
2018-11-25
Gregor Jehle (gregor@p3ki.com)

Contents

1 Challenge Outline

Systems have moved and keep moving towards a distributed design, be it automotive solutions covering car-to-car and car-to-x communication, industrial control systems (ICS/SCADA) used to keep factories and the power grid operational, or the growing market around Internet of Things (IoT) applications.

The automotive industry has been aware of this for over a decade and put forward the requirements for a distributed PKI system in the form of a standard – the ISO 20828 – back in 2006. P3KI is now the first company to present a solution modeled after this standard, even improving upon it.

1.1 Technical Challenges

A common theme across all of these systems is that they are ***distributed but not decentralized***.

In short: they heavily rely on often very few or even a single intermediary who's in control, be it intentionally by design or unintentionally by how things have worked out over time. Control is, for instance, exerted via distributed devices depending on a central backend service for control and management. This is often the go-to business model for IoT applications and thus directly visible to all parties involved.

A more subtle but hugely important case when it comes to security considerations, especially for critical infrastructure, is the associated key management. Here the central control is not necessarily wanted but rather a mere fact of life dictated by the system itself because of limitations of the technology currently commonly available. While there are well-established solutions for handling PKIs, these solutions rely strongly on the idea that you're in control of all aspects, all the time.

The flaw in this thinking becomes apparent once you consider use-cases where the operator has lost control and has to react to compromise or an active attacker. Two common pitfalls in these scenarios are:

- A few central servers required for communicating certificate revocations (e.g. CRL and OCSP)
- The fact that revoking intermediates requires rolling out new certificates to every distributed device, even those that might be online only sporadically or not at all

1.2 Legal Challenges

Control implies responsibility. This is of special relevance if you're selling networked devices that rely on classic PKI approaches to Transport Layer Security (TLS) or authentication and authorization of communication peers.

On the one hand you require control during manufacturing to bring up the device for testing and initial configuration. However, you also might want to hand over responsibility for operating the device to your customer after the deal is closed. If you're still the one in control of the Root Certificate Authority (Root CA) all trust of the device is bound to, you're still in control and thus responsible. Furthermore, this handover should come with the minimum amount of work attached for your customer, otherwise user experience suffers greatly.

Rather than going with half-hearted workarounds to address this issue you'd instead want a system that gives you

- A full audit trail of changes and who approved them
- A no-gap handover procedure leaving no window for attack
- The ability to hand over all or a subset of responsibilities

2 Decentralized PKI by P3KI

P3KI develops a **novel Decentralized PKI** which solves the above challenges and can be integrated easily into your systems.

P3KI's approach addresses the following key issues:

Arbitrarily precise expression of permission levels

- Vastly improved risk management capabilities
- Easier damage assessment in forensic scenarios
- Ability to refine permission expressions while already in the field
- Evaluation of permissions via pure and provable mathematics
- Safe permission delegation semantics

Graceful degradation

- Offline capable
- No central infrastructure required
- Fully transport agnostic (seamless switching between central database, peer-to-peer data distribution, and opportunistic offline data exchange)

Separation of identity and permission delegation data

- Update intermediate permission delegations without affecting leaf nodes
- Update intermediate permission delegations at minimal cost and high frequency

Forward web-of-trust definition

- No inherently trusted root certificate authorities required
- Multi-path permission delegation links as native features

Easy & safe to use

- Off-the-shelf integrations and extensions for PAM, X.509, VPN solutions
- Background service for easy server-based deployments
- C-ABI multi-platform library targeting x86_64, ARM, MIPS, and more
- Easy to use, minimal API
- Trust data fully opaque to application- and transport layer
- Full-stack and full life-cycle consulting and support
  - Assessment
  - Planning & design
  - Customization & implementation
  - Rollout & operations

3 Example Scenarios

3.1 Car-to-Car & Car-to-X Communication

Key aspects: delegation, granularity, M2M communication, interoperability, safety, risk management.

The most obvious difference to existing certificate-based communication security approaches presented by P3KI's solution is the precision with which permissions, roles, and capabilities can be delegated between devices. No longer are we required to manage separate certificates for each role (a common workaround) but can express specific trust for specific operations.

3.1.1 Example: Communication for Trajectory Planning

For instance, P3KI can express and model that Car A trusts Car B to query A's trajectory planning system so B can safely pass A and merge in front of it. This trust can further be limited to work only under specific constraints like manufacturers, models, industry alliances, time frames, protocol version, geo-fencing, and many more. Furthermore, this trust is limited to ***querying*** the trajectory planning system. It does not include the unprovoked providing of input from B to A's system nor does it automatically imply that A is allowed to query B's trajectory system. All relationships are directional, meaning just because you trust someone with something, that someone does not automatically trust you back equally.

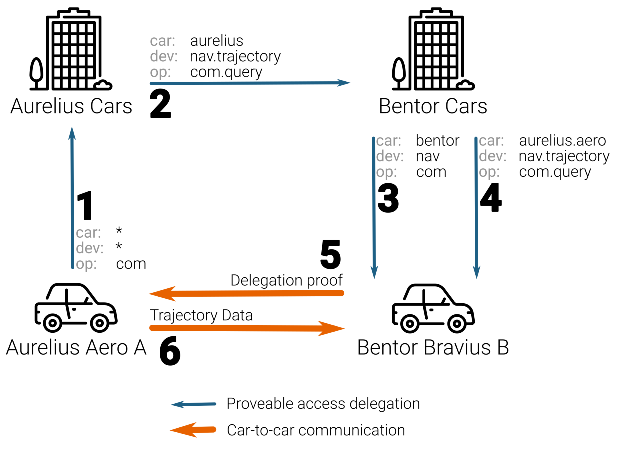

Figure 1: Communication and delegation relationships between two cars (Note: only showing delegations in a single direction for brevity)

3.1.2 Delegation and Scalability

Another key aspect in play with most practical scenarios is delegation. In the scenario at hand here it's not practical that Car A trusts Car B directly. This would require that every car knows every other car, which simply does not scale under any consideration.

Car A would trust its manufacturer (fig. 1, delegation "1") broadly and the manufacturer would then further delegate to third parties on Car A's behalf who to trust for exchanging certain information regarding trajectory planning. This can cover all aspects of trajectory planning, simple queries to A's system, or possibly active input of specified type into the car's systems. The intermediary (e.g. manufacturer or certification body) can only delegate at most this initial level of trust extended to it by the previous party further on to other parties (fig. 1, delegation "2": query trajectory planning for all Aurelius cars). These parties can include other cars of the same manufacturer or industry partners running part or fully compatible systems in their cars.

The example shown in figure 1 has a car manufacturer (Bentor Cars) as the third party. Bentor has a fully integrated line of cars that communicate freely with each other (s. broad delegation "3"). Additionally, the Bentor Bravius models are compatible with and allowed to query the trajectory planning system of the Aurelius Aero line of cars which is shown by the delegation marked "4".

The verification of whether Car B can query Car A's systems (fig. 1, response "6") is entirely made by Car A with delegation proof data that Car B can provide (fig. 1, request "5") together with the request combined with delegation proof data A either already knows or has received from arbitrary third parties (e.g. drive-by push updates from tollgates who themselves don't require any specific level of trust to be able to provide delegation proof data). The verification of the delegation proof and the following exchange of data can happen on the road, possibly inside a tunnel, or in a region outside cellphone coverage without any need for online reachable central infrastructure.

3.1.3 Safety and Security despite Interoperability

The fine-granular nature of delegations possible with P3KI's solution directly allows modelling of trust on a level directly informed by safety and security considerations and risk management. You can trust your own devices more than others yet still allow others some well-defined degree of access.

This makes P3KI an enabler of new technological development while at the same time ensuring integrity and safety of systems while also drawing clear upper bounds for delegation scopes required for risk management analysis.

3.2 Firmware Update

Key aspects: short-lived trust, task specific trust, granularity.

Protecting firmware update procedures via certificate based code signing is already the best practice. However, the question asked during authorization in such a scenario is usually "are you a developer?" or "are you a service technician?". The way these very coarse roles are delegated is usually via a certificate issued by the manufacturer to its employee. Such a certificate is usually bound to the longer lasting role of "being a technician" rather than the task of "install this firmware update" due to the complexity involved in having to have fine-granular permission levels as well as short-lived certificates with current approaches.

3.2.1 Per-Task & Scenario Appropriate Certificates

Issuing a more or less specific level of access delegation is the same operation for P3KI's system and it's both quick and effectively cost neutral. This enables you to both express (as in: level of trust) and issue (as in: duration of validity) access delegation proofs on a per-task basis.

For example: a car will trust its manufacturer to perform all firmware updates targeting all devices of its specific make and model. The manufacturer then delegates a subset of these permissions (updating certain uncritical controllers in certain models) to licensed repair workshops (fig. 2 delegation "1"). A given workshop can then delegate these permissions further to its employees who then perform the actual work on the car.

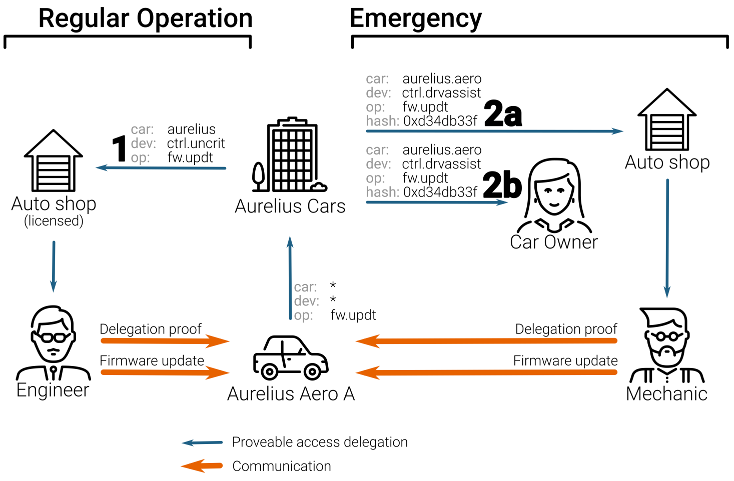

Figure 2: Firmware update in regular operation and emergency scenarios

Dealing With Emergencies Within this same framework it's now possible to flexibly and under perfect control sidestep the regular process and hierarchy to address extraordinary circumstances. Take for instance a critical motor controller update that needs to be rolled out to a certain model of car. The manufacturer can now issue additional delegation proofs alongside the firmware update to allow anyone with the right equipment – even unlicensed shops (fig. 2 delegation "2a") or tech-savy car owners (fig. 2 delegation "2b") – to install this specific update on specific controllers in specific models.

This can be expressed without affecting existing delegations as well as in a way so that the new delegation will automatically be invalid after a given time and generally not usable for incompatible models and/or controller combinations.

Unique Cases With Uniform Handling From the car's point of view all firmware update requests look the same except for additional delegation proof data provided by the updating party. This additional data can be used to authorize a given update even in the absence of third party verification instances (as in: entirely offline). The only delegation proof that needs to be present initially on the car is its trust towards its manufacturer. Anything beyond that is provable by third-party provided delegation proofs.

3.2.2 Smart vs. Simple Update Workflow

P3KI's solution supports modeling a smart update use-case where the device to be updated can verify actively and in-situ that the party performing the update operation is trusted to do this right now. This is enabled by a secure handshake protocol based on P3KI's cryptographic primitives and requires the updating party to have access to its personal identity (asymmetric cryptographic key pair) during the update.

If this is not an option, P3KI allows scenarios where anyone can install previously approved firmware updates using nothing more than a portable storage medium. At first glance this seems little different to existing signed firmware update approaches. However, P3KI's solution has several benefits:

- Federated approval of updates possible
- Restrict the set of possible targets of a firmware update using arbitrary precise policies
- Audit log including full chain of approval

Federated approval means all verification steps required to approve an update can be made locally without having to consult third parties central or otherwise. If third parties are available they can of course be included in such an evaluation to increase confidence even further, but they are not required.

Arbitrary precision policies mean you can introduce new permission levels or more specific permission levels at all levels at any time without having to update existing devices in already the field. It also means you can express restrictions and permissions in a multitude of dimensions like hierarchical expressions, value ranges, geo-locations, and time.

Full chain of approval for auditing means that every verification sees every detail of a chain of delegations: every intermediary, their respective delegation details, the version and age of that delegation, and finally the party that requested the operation. If questions arise later on, this information can be used to learn who did what and when at whose authority.

3.3 Ownership Handover

Key aspects: audit trail, granularity, zero-gap trust.

A fundamental implication of ownership and control is responsibility and ultimately liability. There are scenarios that require the manufacturer to fully hand over ownership of and control over devices to customers and/or third parties. This poses additional challenges when considering Public Key Infrastructures which are part of virtually all non-trivial systems that require networking and communications security. This becomes especially relevant in jurisdictions that require the ability to resell previously used devices at will by their current owner.

3.3.1 Handover Process

P3KI's technology easily enables controlled handover scenarios via several key aspects of its technology. Take the following workflow example for an ownership handover process based on P3KI's solution:

1. Car is initialized at manufacturing time with the knowledge that its manufacturer is its owner
2. After completing the sale to a customer, a suitable role is delegated to said customer for the specific car that was bought.
3. The customer now provides proof of ownership to the car and requests the ownership role to be transitioned to her. The car can verify this request without internet connectivity.
4. The car has transferred the role to the customer and at the same time removed the role delegation to its previous owner.

After this process, the manufacturer has no power anymore over the car whereas the customer holds full control.

This handover can be performed completely (as presented above) or partially. It would for instance be possible to only allow certain levels of base trust to be updated to allow the customer to establish control over the device but still allow the manufacturer to retain a well-defined but lesser amount of control (e.g. performing some maintenance operations). This in turn could also be used to implement lawful interception (LI) interfaces in a safe, secure, and most importantly auditable fashion.

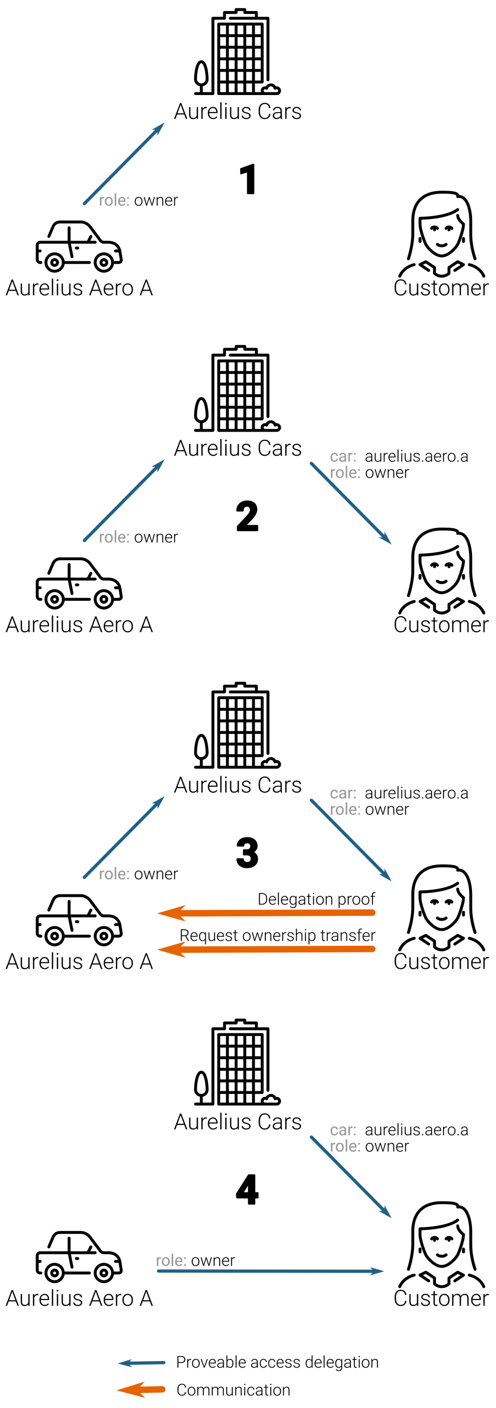

Figure 3: Outline of an ownership handover workflow

4 Solution Background Information

4.1 Arbitrary Precision Access Delegations

To express permission delegations, roles, and capabilities we designed a concept called ***Trust Policy Languages*** or TPL. TPLs are scenario specific, highly adaptable, arbitrarily specific or coarse, and most importantly formally proven to behave correctly and reliably in all situations.

Underlying all TPLs is a mathematical concept that ensures delegated permissions can be at most equal to what one was trusted with to begin with or smaller. Interpretation of TPLs is fully handled by P3KI's software unburdening the application and business logic from this task to ensure correctness across all platforms and implementations.

Since TPLs can express permissions, roles, and capabilities in an arbitrarily sub-dividable manner, it's possible to design a TPL that solves the exact task at hand without having to fall back on pre-defined permission levels, roles, or capabilities. It also allows to be as precise in expressing these as needed. This in turn allows risk management to gauge the impact of individual nodes getting compromised in never before seen precision. In terms of roles and permissions it is for instance easily possible to perfectly differentiate between permissions granted to a CEO and a director.

Another effect of this feature is vastly improved protection against lateral movement of attackers. Instead of compromising a generic certificate enabling the attacker to effectively move freely in your network, the attacker only gains the exact permissions, roles, or capabilities the compromised device is trusted with and only towards those devices that trust it to begin with.

In real world terms, loosing a classic certificate is similar to using a janitor's master key. A lowly role with superior access. This scenario can easily be averted in trust models designed using a proper TPL.

4.2 Decentralized and Offline-Capable

P3KI's solution enables you to build systems that offer actual graceful degradation. This is possible because our solution does not rely on any specific storage or communication network to exchange delegation data relevant to determine who trusts whom and to what degree.

In practice this means that if all is well, you can use a central service or database to exchange delegation data. This is simple and allows updated relationships to be communicated instantly. Further, it's possible to double-check every piece of delegation data to verify it has not yet been superseded by a newer version.

However, whether delegation data of any party is cached locally in your network or stored centrally makes no difference to our system. All delegation data is verifiable at all times and if you're presented with two different versions of a given party's delegation data, you can always decide which one is newer and ignore the older one.

This in turn enables you to fall back to communicating delegation data through other channels, for example via a peer-to-peer network should a central service become unavailable. This might have some impact on corner case performance (as in blind searching for relationships) but has very little performance impact in common cases like verifying a piece of delegation data is of the newest version known to the network.

In extreme cases delegation data can even be communicated opportunistically every time any two parties meet or using store and forward networks where you can leave delegation data for others to discover.

Either way, all delegation data – regardless of its origin or how it was communicated – is safely verifiable and can be incorporated into proofs and verifications by anyone at all times without any central infrastructure or connectivity being required. This is true even offline, without any internet connectivity at all.

4.3 Platform: Embedded, Mobile, Backend

P3KI's decentralized PKI solution is developed using the Rust programming language. This language offers a great many safety and security guarantees over traditional languages like C, C++, or Java, eliminating whole classes of possible bugs and issues.

Rust is also a language targeting a similar level of abstraction as C and C++ do. This means Rust enables low-level programming and targeting of embedded platforms including micro-controllers.

It also integrates into existing environments and development processes exceptionally well since it offers support for interfacing with existing C and C++ code via an industry standard C ABI.

Using Rust we're able to target embedded systems, mobile platforms such as Google Android and Apple iOS, consumer computers, and server hardware with a single codebase.

4.4 Architecture

Adding decentralized PKI support to your system using P3KI's solution can be easily done in three different ways. This gives you excellent flexibility throughout all levels of your project lifetime spanning from initial evaluation to integration and finally rollout and operation.

4.4.1 Command Line Interface (CLI)

This is the perfect interface for direct interaction and control of individual devices and identities. The interface is both straight forward to use directly from the command line and indirectly by integrating it into your own scripted solutions.

It's the preferred solution for evaluation and small scale deployments as well as doing maintenance and operations tasks.

4.4.2 Standalone Service

Our standalone service offers similar functionality to the CLI but has several added features. A key difference is its REST interface, making it very easy to integrate into existing micro architectures and web applications. Applications already using the command line interface can be adapted to use the standalone service very easily.

4.4.3 Native Library

Our library offers the same general functionality as the CLI with the added benefit of closer control and the ability to write compact, native applications targeting any platform from embedded systems to powerful server hardware. It's the perfect solution if you want to seamlessly integrate our solution into your product.

5 Glossary

Delegation proof data	Data providing proof that a specific level of delegation across an arbitrarily definable number of intermediaries exists at a specific point in time and did so cryptographically provable. This data can be provided by anyone via arbitrary transport since it's cryptographically protected from being tampered with by anyone that's not the issuer of the data. Several individual delegation proofs can be combined arbitrarily into larger proofs by anyone who participates in the trust network. All proofs can be verified in offline scenarios without having to involve third parties.
Delegation data	see ***delegation proof data***
REST	Representational State Transfer (REST) is a software architectural style that defines a set of constraints to be used for creating web services. Web services that conform to the REST architectural style, or RESTful web services, provide interoperability between computer systems on the Internet.
X.509	In cryptography, X.509 is a standard defining the format of public key certificates. X.509 certificates are used in many Internet protocols, including TLS/SSL, which is the basis for HTTPS[1], the secure protocol for browsing the web.
PAM	Linux Pluggable Authentication Modules (PAM) provide dynamic authentication support for applications and services in a Linux or GNU/kFreeBSD system.
VPN	A virtual private network (VPN) extends a private network across a public network, and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network. Applications running across a VPN may therefore benefit from the functionality, security, and management of the private network.
Firmware	In electronic systems and computing, firmware is a specific class of computer software that provides the low-level control for the device's specific hardware.

6 About P3KI GmbH

P3KI is a Berlin-based Startup grown out of Recurity Labs' – a Berlin-based IT Security consultancy – research branch. We develop a novel, fully distributed PKI for access delegation, solving several hard problems like resilience and graceful degradation with applications for IoT and beyond. P3KI offers the first real distributed, decentralized, and federated authorization and authentication solution including full offline verification capabilities. We enable autonomous systems to make local decisions and be resilient to attack.

Contact us by email via contact@p3ki.com or phone via +49 (0)711 22051252 (Stuttgart office) or +49 (0)30 695399933 (Berlin office). For more details please visit our website at www.p3ki.com.