So, you now have crypto..

Planning for Third-Party Security Assurance and Penetration Tests as an OEM or Supplier

Dr Liz James

Contents

1. When Assurance meets Ceremony 3

2. From Checksums to Certificates: Complexity Crept In 4

2.1. The Age of Checksums 4

2.2. The rise of interdependency 4

2.3. The Slow Arrival of Cryptography 4

2.4. Testing Complexity That Grows with Every Safeguard 4

3. A Larger, Looser Ecosystem 5

3.1. Mixed maturity as the new normal 5

3.2. Distributed responsibility 5

3.3. The expanding test boundary 5

3.4. The human supply chain 5

3.5. Why it matters 6

4. From Component Pen Tests to System Assurance 6

4.1. When the box stopped being the boundary 6

4.2. Testing as participation 6

4.3. Assurance debt through isolation 7

4.4. Why system assurance matters 7

4.5. The Human Side of the Transition 7

5. Fit for Purpose: Making Cybersecurity Interface Agreements Work for Everyone 8

5.1. Responsibility without capability 8

5.2. Precision without practicality 8

5.3. The hidden cost of bespoke ecosystems 8

5.4. Making CIAs fit for purpose 9

5.5. Shared choreography 9

6. How We Are Responding 9

6.1. Developing capability ahead of demand 9

6.2. Why research matters now 9

6.3. Prototyping the next generation of test tools 10

6.4. R&D where it belongs 10

6.5. The goal: testability as an ecosystem property 10

7. From Penetration Testing to Participation 10

7.1. Consultancy as a Long-Term Partner 11

7.2. A Culture of Evidence 11

7.3. Choreography, Not Ceremony 11

8. Conclusion: Participation at Every Level 11

Abstract

As cryptographic standards such as AUTOSAR SecOC and SAE J1939-91C move from design papers into production vehicles, the task of validating security has become both technically and organisationally complex. What once involved handing an ECU to a test team for a quick penetration test now demands coordinated access to keys, certificates, and trust infrastructures spread across multiple suppliers and OEMs.

This paper explores how cryptography reshapes the relationship between OEMs, Tier 1 suppliers, and third-party security consultancies; how poorly conceived Cybersecurity Interface Agreements (CIAs) can create gaps in responsibility and evidence; and how NCC Group is adapting its methods and tooling to keep independent assurance feasible.

We argue that “having crypto” is not the end of the security journey, it is the beginning of a new form of assurance. Testing must evolve into participation within the system’s trust fabric, and assurance must start at concept, not at the end of development. Because in a world where every message is signed, ceremony without choreography is chaos, and that choreography begins at concept.

1. When Assurance meets Ceremony

Cryptography has finally reached the factory floor

Automotive, heavy-vehicle, and other cyber-physical systems are now embedding security properties, authentication, integrity, confidentiality, and non-repudiation, directly into their architectures. Standards such as AUTOSAR SecOC and SAE J1939-91C make those properties tangible, wiring cryptographic proofs into every message and every control loop.

From a security-engineering perspective, that’s progress. From a testing perspective, it’s a headache

In traditional IT systems, the industry has had decades to learn how awkward cryptography can be. Broken certificate chains, expired roots, mis-managed PKI, and hard-coded credentials have been a cause for countless sleepless nights and swimming pools of coffee. Even mature DevOps teams still struggle with expiring tokens and forgotten trust anchors.

Now that same complexity is arriving in embedded and safety-critical platforms, without the luxury of rapid patch cycles or centralised key management. The challenge is no longer only how to implement cryptography, but how to test it in environments where keys are proprietary, certificates are OEM-signed, and hardware security modules are sealed tighter than the engine bay.

The result is a collision between two worlds

Testing practices built for open networks are meeting architectures designed for closed trust domains. OEMs and Tier 1 suppliers must now coordinate cryptographic material, test environments, and evidence production across supply chains that were never designed for that level of integration.

This paper explores what that collision means in practice: how cryptography changes the way third-party testing works, why traditional penetration-testing approaches fall short, and what new forms of assurance are emerging to replace them. In a world where every message is signed, assurance is no longer about breaking in, it’s about proving that everyone inside the system is who they claim to be.

2. From Checksums to Certificates: Complexity Crept In

The story of modern assurance doesn't begin with cryptography. It starts with a much humbler ambition: making sure messages weren't corrupted in flight.

2.1. The Age of Checksums

As functional safety (FuSa) requirements became mainstream, control systems began wrapping their messages in checksums and sequence counters. The goal wasn't security—it was reliability. If a frame flipped a bit or arrived out of order, the receiver could reject it safely.

Those simple mechanisms had nothing to do with secrecy or authentication, but they already changed how testing worked. A tester could no longer send arbitrary traffic; the payload had to be correctly formed, counters synchronised, and timing precise. Miss a single increment and the ECU would quietly ignore you.

This was the first sign that the old "poke it until it responds" approach to testing was fading. Even without crypto, understanding how a system expected to be spoken to became essential.

2.2. The rise of interdependency

Next came plausibility checks and multi-source validation. Critical ECUs stopped trusting single sensors or isolated messages. To reach an operational state, a subsystem might now require consistent readings from multiple peers - a speed signal, a torque demand, and a mode indicator all arriving in the right order and timing window. From a safety and control perspective, that was progress. However, from a testing perspective, it meant that bench setups had to emulate whole ecosystems, not individual nodes. We saw this long before cryptography arrived: ECUs that would only start transmitting once a full network of believable signals surrounded them; diagnostic modes that refused activation unless half the car seemed alive. Testing became an exercise in systems theatre - you had to recreate the stage before the actor would perform.

2.3. The Slow Arrival of Cryptography

That layered choreography set the stage for the next evolution: cryptographic authentication. We're beginning to see it appear in production, but unevenly. High-impact domains such as gateways, telematics modules, and power-train controllers are adopting standards like AUTOSAR SecOC and SAE J1939-91C. Elsewhere, the network remains a patchwork of legacy checksum-protected signals and early crypto pilots. Iterative platform revisions make this transition messy by design. New ECUs may authenticate messages cryptographically while neighbouring systems still rely on rolling counters or CRCs.

For years to come, testers will face hybrids where one side speaks cryptographic ceremony, and the other speaks 2000s-era CAN. This mixed ecosystem poses practical questions:

- Which parts of the trust fabric truly need cryptography now?
- How do we test interfaces between secured and legacy domains?
- And how can evidence stay consistent across generations of vehicles?

2.4. Testing Complexity That Grows with Every Safeguard

At each stage, checksums, multi-signal dependencies, and now selective cryptographic authentication, the assurance task has grown more systemic. Every safeguard designed to improve integrity has also increased the coordination required to test it.

The industry didn't leap from cleartext to certificates; it climbed through layers of interdependence. By the time cryptography reached the production line, testers were already halfway inside the system's choreography. Now, participation in that choreography isn't just procedural, it's mathematical.

3. A Larger, Looser Ecosystem

Vehicle networks have never been tidy, but the coexistence of legacy buses, safety-enhanced platforms, and early cryptographic pilots has stretched the ecosystem into something far more sprawling than the classical OEM–supplier chain. Each generation of hardware inherits assumptions from the last while adding new trust mechanisms of its own.

3.1. Mixed maturity as the new normal

Walk through a modern architecture and you can see the layers in real time:

- A power-train gateway may already sign/authenticate its traffic using SecOC or J1939-91C.
- A neighbouring chassis controller still relies on counters and CRCs.
- Body and comfort domains, often sourced from different Tier 1s, might transmit completely unauthenticated CAN frames.

This heterogeneity isn't a failure; it's an artefact of long product lifecycles and incremental certification. Each platform revision inherits years of design and validation evidence that nobody can afford to discard overnight. New cryptographic features are added where the business case or safety argument is strongest, first in gateways and update channels, later in everything else.

For testers, this staggered adoption means no single set of assumptions holds. Every engagement starts by asking: Which layers of this network speak security, and which still speak convenience?

3.2. Distributed responsibility

The diffusion of technology has mirrored a diffusion of accountability. Where once an OEM might own the end-to-end design of a few ECUs, today's vehicles are assembled from components and subsystems built by dozens of suppliers, each with its own approach to security engineering:

- A Tier 1 delivers a gateway implementing the OEM's cryptographic policies.
- A Tier 2 provides the cryptographic library buried in its firmware.
- The OEM integrates everything, adds over-the-air provisioning, and relies on a cloud provider for certificate distribution.

Each actor controls part of the assurance puzzle; none see the whole picture. When something fails - expired certificates, incompatible key lifetimes, mismatched freshness handling - the root cause spans organisations rather than modules.

3.3. The expanding test boundary

In this environment, the "system under test" has no clean edge. A component's secure behaviour depends on certificates issued by the OEM, counters synchronised across suppliers, and timing parameters defined by the integration team. A single ECU can only be validated meaningfully when the surrounding ecosystem, virtual or physical, exists. For independent test houses, that means reconstructing parts of the vehicle's identity management system just to get the unit to talk.

The days of bench-testing an isolated ECU are fading; the assurance boundary now stretches from the manufacturing plant's provisioning scripts to the cloud service distributing updates.

3.4. The human supply chain

Assurance complexity isn't just technical, it's social. Testing now requires cooperation between cybersecurity engineers, safety specialists, procurement lawyers, and backend architects who may never have met and each speaks a slightly different dialect of "risk."

- A security engineer wants access to certificates.
- A compliance manager wants to avoid liability.
- A supplier wants clarity on testing costs.
- A tester wants a network that actually talks.

Aligning those interests has become its own branch of engineering: "diplomacy as a service".

3.5. Why it matters

The uneven rollout of cryptography means the industry will live with hybrid ecosystems for at least a decade. That period will decide whether the transition strengthens assurance or simply relocates uncertainty into the seams between systems and organisations. For now, testers must navigate both worlds: replaying checksum-guarded legacy messages one minute and joining authenticated sessions the next. OEMs must coordinate assurance evidence across suppliers whose maturity varies wildly. And regulators must interpret security claims that depend on a mixture of mathematical proof and procedural trust.

The ecosystem has never been larger, looser, or more dependent on shared choreography. How well it learns to dance together will determine how credible its assurances become.

4. From Component Pen Tests to System Assurance

For decades, the dominant testing rhythm in automotive was comfortably simple. An OEM or Tier 1 would ship an ECU to a lab, sometimes with a schematic, sometimes not, and expect a penetration-test report in return. The testers powered it up, watched the CAN traffic, and saw what they could make it do. That model worked because the system boundaries were clear and the networks were trusting. The ECU was a sealed appliance that either resisted external input or didn't. Security testing was about finding faults in isolation. However, the world we work in now no longer allows that simplicity.

4.1. When the box stopped being the boundary

Modern vehicle subsystems are not self-contained. They are nodes in a distributed security fabric that spans the entire vehicle, and often beyond it. Their behaviour depends on key material provisioned in manufacturing, certificates issued by an OEM authority, and signals synchronised with peers across the network. The assurance boundary has therefore shifted from the component to the system. Testing a single ECU tells you little unless it can complete its handshake with the rest of the trust chain.

In practice, what arrives on the bench today often looks inert:

- An ECU that refuses to transmit until it authenticates its peers.
- A bootloader that rejects unsigned firmware.
- A network gateway that silently discards packets from any unrecognised identity.

From a design point of view, that's excellent security hygiene. However, from a tester's point of view, it's an exceptionally quiet afternoon.

4.2. Testing as participation

To test such a system, we no longer attack from outside, we join the conversation. Testing becomes an act of participation: acquiring valid certificates, synchronising freshness counters, deriving session keys, and deliberately corrupting them to watch the results. The tester must act as a legitimate peer, not just a noisy neighbour.

That shift turns every engagement into what we call assurance by participation, verifying that the system's cryptographic ceremonies behave correctly under stress and failure, not trying to bypass them entirely.

Doing this requires:

- Test keys and certificates that mimic production but are cryptographically independent.
- Controlled access to configuration interfaces for injecting or rotating those keys.
- Simulation harnesses that act as credible peers or leaders in the protocol.

Without those, a "pen test" becomes a black hole of time: the device refuses communication and everyone mistakes silence for security.

4.3. Assurance debt through isolation

When teams stick to the old model, testing boxes in isolation, they inevitably accumulate assurance debt: results that look complete but lack relevance. A report might state that "penetration testing was performed" while noting that message authentication, certificate handling, and key rotation could not be tested due to unavailable artefacts. That report satisfies a process requirement but not an assurance one.

Integration later exposes misalignments: different key lifetimes, inconsistent freshness handling, and nobody has the evidence needed to explain why. Each supplier tested what they could reach, but nobody validated how those parts fit together. The evidence trail looks busy yet shallow - a pattern every auditor eventually recognises.

4.4. Why system assurance matters

Modern regulation demands system-level reasoning. UNECE R155 and its associated standard ISO 21434 both require traceable evidence that security objectives are met throughout the lifecycle, not merely within a single component. That means testing must contribute to a larger assurance case:

- Defining the system of interest (vehicle, fleet, or platform).
- Identifying interoperating systems that exchange trusted data.
- Mapping the enabling systems that provision, update, and maintain those trust relationships.

Only when those are explicit can individual component tests make sense in context. Otherwise, assurance collapses into paperwork disconnected from system reality.

4.5. The Human Side of the Transition

Culturally, this shift is uncomfortable for almost everyone involved. Penetration testing has always prized independence: arrive late in the project, operate at arm's length, and uncover surprises. System assurance is the opposite: it rewards early collaboration, context, and traceability.

For testers, that feels alien. For engineers, it feels risky: inviting outsiders into design conversations seems to blur accountability. For procurement and compliance, it complicates neat boundaries between "development" and "validation". But the world we're moving into doesn't respect those old categories. Independence can no longer mean ignorance.

The tester who joins late and knows little can only scratch the surface of a system built on cryptographic interdependence. True independence now comes from perspective: the ability to see the system whole, understand its claims, and verify them objectively from within. This is also where the security-consultancy industry itself must mature. Much of our profession grew up around product testing and adversarial audits - engagements measured in findings, not in assurance coverage. That model worked when vulnerabilities were visible and systems were loosely coupled. However, it struggles when assurance is about verifying trust chains, lifecycle controls, and safety dependencies. Security consultancies need to integrate more deeply with security-engineering disciplines:

- Learning the language of requirements, verification, and evidence.
- Participating in architectural reviews rather than post-release fire drills.
- Building relationships that last through a platform's lifecycle instead of a single report.

It's a mindset shift from penetration testing as a service to assurance as a partnership. When consultancies adapt in that way, they stop being peripheral auditors and start becoming the connective tissue between engineering intent and regulatory evidence.

5. Fit for Purpose: Making Cybersecurity Interface Agreements Work for Everyone

Cryptography was meant to bring certainty to vehicle networks, instead, it has exposed how uncertain our relationships are. The shift from open buses to authenticated messaging didn't just alter protocols; it redefined how trust, accountability, and evidence flow through the automotive ecosystem. At the centre of that new web sits the Cybersecurity Interface Agreement (CIA) - the document intended to define who does what, who holds which keys, and who is responsible for proof. But in practice, many CIAs remain conceptually elegant and operationally untestable.

5.1. Responsibility without capability

A common CIA clause still reads:

“The supplier shall implement and verify cryptographic mechanisms in accordance with OEM requirements.”

On paper, that splits the task neatly. In reality, it creates the trust gap that has come to define so many programmes: the OEM owns the certificate authority and production keys, while the supplier must verify behaviours it can't fully exercise. The CIA may appear “fit for purpose” from a contractual viewpoint, but not from a technical one and the result is accountability without capability - a pattern now familiar across the supply chain.

5.2. Precision without practicality

Most ineffective CIAs fail for the same reason: they describe responsibilities precisely but impractically. They specify what shall be done but never how it will be possible.

Missing details include:

- Who provides test keys and when.
- Which CAs issue them, and under what governance.
- How test environments maintain independence.
- What constitutes acceptable evidence for each assurance claim.

Without those definitions, every project starts with the same question: who owns the testability problem?

5.3. The hidden cost of bespoke ecosystems

Suppliers working across multiple OEM programmes face an invisible tax. Each customer's PKI, certificate policy, and key-rotation scheme differs subtly. Replicating those environments for testing means maintaining parallel infrastructures, staff training, and validation cycles - work rarely acknowledged in contracts. When that cost isn't planned for, assurance becomes performative. Teams run just enough tests to keep milestones green, not to build confidence.

5.4. Making CIAs fit for purpose

A CIA that works is one that treats assurance as a shared responsibility rather than a delegated chore. It defines:

- Which artefacts (test keys, dummy certificates, simulated roots) may be shared.
- How trust domains are separated between production and test.
- How results are accepted as evidence in each party's assurance case.

This requires maturity on all sides. OEMs must design their PKIs with testing in mind; suppliers must build controllable interfaces; and security consultancies must act as the connective tissue - neutral facilitators who can create test harnesses, manage temporary trust domains, and ensure both independence and practicality. That's the consultancy's evolving role: not gatekeeper, but assurance enabler.

5.5. Shared choreography

When CIAs are fit for purpose, cryptographic assurance becomes a cooperative performance. The OEM defines intent, the supplier implements, the tester verifies, all to a shared rhythm. Failures happen when that rhythm is missing: when CIAs reduce assurance to checklists or treat testing as an afterthought. The solution is not more paperwork but better choreography: agreements that enable each participant to do what they are accountable for, with the artefacts and authority they need. When that balance is achieved, a CIA stops being a contract and starts becoming a mechanism of trust.

6. How We Are Responding

If cryptography is reshaping how systems behave, it's also reshaping how we prepare to test them. The automotive sector isn't yet saturated with authenticated messaging, but it's coming. We're already seeing early implementations appear within suppliers, sometimes for pilot programmes, sometimes hidden deep in the next platform revision. The question isn't whether third-party security testing will have to adapt, but when. And by the time the first cryptographically authenticated components arrive on the bench, it will be too late to start learning how to talk to them.

6.1. Developing capability ahead of demand

We have begun a research and development stream focused on testability in cryptographic environments. This isn't billable client work; it's groundwork: building understanding, tooling, and processes before the commercial need becomes routine. Our aim is to avoid the scramble that often happens when a new technology suddenly enters the assurance pipeline. When the first SecOC or J1939-91C components arrive, we want to already have the foundations:

- Conceptual models for how test harnesses will authenticate and exchange keys,
- Simulation frameworks to recreate representative network states, and
- Design patterns for how testing can operate safely within OEM or supplier trust boundaries.

It's preparation, not production. We're not waiting for demand to dictate capability.

6.2. Why research matters now

Building these tools in advance is awkward precisely because we can't yet test them against live targets. Every prototype is a hypothesis: "this is probably how the cryptographic handshake will behave in production." but research has to lead implementation, not trail it. We've seen the same pattern before - functional-safety testing frameworks that only emerged years after the first ISO 26262 programmes launched, costing everyone time and rework. We'd rather not repeat that.

By investing early, we can refine our methods in a neutral, exploratory space, experimenting with key-exchange flows, certificate structures, and diagnostic entry points without the pressure of customer deadlines or contractual constraints. That breathing room is essential if the resulting methods are to be robust enough for industry use.

6.3. Prototyping the next generation of test tools

We're currently exploring the design of a J1939-91C Network Leader and Follower state machine: a test harness that could perform key exchanges, manage freshness counters, and observe authentication flows across ECUs. It hasn't landed in our laps for testing yet. We're treating it as a research artefact: a way to experiment with state synchronisation, certificate lifecycles, and fault injection in a controlled, non-OEM environment.

The intent is to make future testing practical and ethical: to verify cryptographic behaviour without demanding production keys, and to generate reusable assurance evidence rather than ad-hoc test reports.

6.4. R&D where it belongs

This work doesn't fit neatly into a commercial project plan. You can't charge one OEM for building tools that will ultimately benefit an entire sector, so we're handling it as internal R&D - shared learning rather than client-specific development. That means open dialogue: we're speaking with suppliers, tool vendors, and standards bodies about what testability hooks might look like in upcoming architectures. If the community designs for assurance early, everyone benefits later. This is about ensuring that by the time authenticated messaging becomes normal, independent assurance is still possible.

6.5. The goal: testability as an ecosystem property

The outcome we're working toward isn't a single product or framework; it's a new way of thinking about how cryptographic systems can be tested. If every OEM and supplier assumes that their implementation will one day face independent verification, then testability becomes a first-class design requirement, not an afterthought. Our research is trying to show how that can be achieved safely, efficiently, and repeatably. Assurance isn't just something done to a system; it's a property of how the system is built and shared.

7. From Penetration Testing to Participation

The traditional penetration-testing model: outsiders attacking black boxes, is poorly suited to systems that refuse to talk without credentials. Assurance by participation replaces that with collaboration: testers join design discussions, help define test hooks, and act as informed adversaries within the system's trust boundaries. But penetration testing itself is only one slice of a much larger assurance landscape.

Modern programmes also need:

- **Design-stage security analysis** – verifying that architectures meet security and safety objectives before implementation.
- **Source-code and binary review** – assessing cryptographic libraries, protocol handling, and memory safety directly.
- **Configuration and integration validation** – ensuring that secure defaults persist through build pipelines and OTA updates.
- **Vulnerability and threat modelling** – identifying misuse cases early and tracking their mitigation through design evolution.
- **Continuous evidence gathering** – embedding verification steps into CI/CD and validation workflows so assurance grows alongside the product.

These activities don't replace penetration testing; they contextualise it. A well-timed penetration test becomes confirmation that the broader assurance machinery works, not a desperate attempt to discover what was missed. Assurance by participation therefore isn't just a new testing method, it's a reframing of how assurance itself is practised.

7.1. Consultancy as a Long-Term Partner

For consultancies, this shift is existential. We can no longer define success purely by the number of vulnerabilities found. Our value lies in building confidence, in creating conditions where security, safety, and reliability claims can be proven through coherent evidence. That means working alongside engineering teams throughout the lifecycle: helping specify assurance requirements, supporting design reviews, and later performing validation within those same frameworks. It's a slower, steadier kind of independence - one based on transparency rather than surprise.

7.2. A Culture of Evidence

Ultimately, assurance is about storytelling with data. Each test, log, or simulation trace is a piece of evidence in a broader narrative that explains why the system can be trusted. Security, safety, and dependability are not competing objectives, they are interdependent claims supported by shared artefacts. When every stage of development produces those artefacts intentionally, final evaluation becomes confirmation, not discovery. Regulators stop auditing paperwork and start reviewing proof.

7.3. Choreography, Not Ceremony

Cryptography has introduced a lot of ceremony into engineering: certificates, signatures, lifetimes, counters, but ceremony alone doesn't guarantee assurance. What matters is choreography: knowing when and how each participant acts so that the whole system remains coherent. That choreography begins at concept - if OEMs, suppliers, and testers learn to move in step, each bringing their expertise at the right time, then the cryptographic future can be both secure and testable. Assurance by participation isn't just a methodology; it's an attitude. It treats trust as something we build together, not something we audit after the fact.

8. Conclusion: Participation at Every Level

Cryptography has changed more than just protocols; it has changed the way trust itself must be proven. Checksums and counters could be verified in isolation. Cryptographic authentication, by contrast, binds every participant: engineers, suppliers, OEMs, auditors, and regulators, into one continuous assurance system. That interconnectedness is both the challenge and the opportunity. We can no longer treat security testing as a discrete task or outsource assurance as if it were a commodity. It is now a shared responsibility that depends on visibility, cooperation, and the willingness of all parties to plan for testability from the start:

- **For OEMs**, this means taking ownership of assurance integration: ensuring that evidence from suppliers and third parties forms a coherent narrative that regulators and customers can trust.

- **For suppliers**, it means building transparency and controllability into their components so that independent testing is not only possible but productive.

- **For consultancies**, it means maturing beyond penetration testing into long-term partnership - investing in research, helping shape architectures, and acting as translators between engineering intent and regulatory proof.

Most importantly, for everyone involved, it means recognising that assurance is no longer a compliance checkbox; it is an organisational behaviour. It thrives only when security, safety, and reliability teams work as one system, producing continuous, verifiable evidence that a design behaves as claimed. Cryptography will continue to tighten the links between those systems and the people who build them. If we treat that tightening not as constraint but as choreography, then the next decade of automotive and cyber-physical innovation can be secure and provable.

Assurance by participation isn't just a testing philosophy, it is the recognition that trust, once distributed, can only be maintained together.