Velocity

V5 2026

Booz Allen

INSIGHTS FOR INNOVATORS

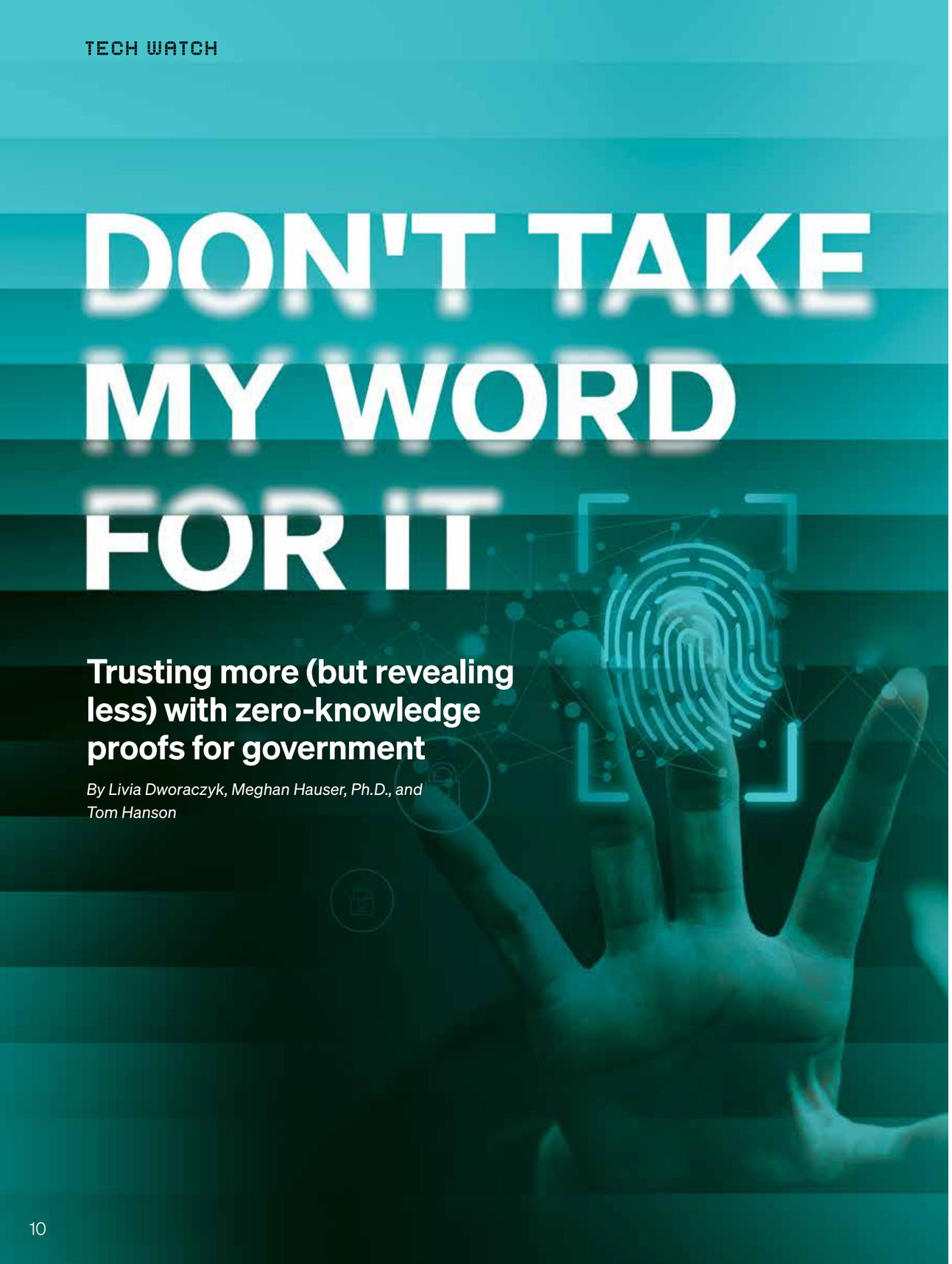
AI vs AI

REIMAGINING CYBER FOR A FASTER FIGHT

DON'T TAKE MY WORD FOR IT

Trusting more (but revealing less) with zero-knowledge proofs for government

By Livia Dworaczyk, Meghan Hauser, Ph.D., and Tom Hanson



When was the last time you had to prove that you were you? From filing taxes to checking in at the airport, it probably wasn't very long ago. Most of us are used to handing out personal data to simply participate in society.

Address. Birthday. Social Security Number.

Verification is exploding everywhere. Meanwhile, the world of cybersecurity is wrestling with a philosophical contradiction as two opposing priorities come to a clash: "We need more proof!" versus "We need more privacy!"

Individuals and corporations are seeking more data privacy for good reasons. In the World Economic Forum's Global Cybersecurity Outlook 2026, 73% of survey respondents reported that they or someone they know was personally affected by cyber fraud or attacks in 2025.

But beyond individual concerns, the discord between proof and privacy is playing out on a national and global scale. For example, intelligence agencies need to share data with coalition partners without compromising sources and methods. Intelligence officers may need to prove AI outputs are trustworthy without revealing classified models and logic.

The next big wave in security is to accommodate this contradiction by decoupling verification (proving that something is true) from disclosure (revealing the underlying information that makes it true). From the Defense Advanced Research Projects Agency to the National Institute of Standards, researchers and developers across U.S. government and industry partners are turning to a mathematical concept introduced over 40 years ago: zero-knowledge proofs (ZKPs).

This cryptographic method has scaled throughout the blockchain industry over the past decade, enabling private transactions on a public ledger. In the coming years, we expect the same methods to profoundly change what's possible in adversarial, data-sensitive environments at the national level. With analysis and insights from Booz Allen's Tech Scouting and Intelligence team, we examine where the market trends are heading and why ZKPs matter now for federal government leaders and the U.S. security industry.

Technology at a Glance

ZKPs are cryptographic methods that let you prove that you know something or meet a given requirement without showing exactly how. Let's illustrate this idea with a low-stakes example.

Imagine you're trying to rent a car. The rental company requires drivers to be over 21 years of age to avoid extra fees. But when you show your driver's license, you end up revealing a lot more than that: your full name, exact date of birth, home address, license number, height, eye color—basically a full identity profile. The agent doesn't need most of this information, yet now they've seen it, and you have no control over how the data is remembered, copied, or potentially misused.

A zero-knowledge version of this interaction would let you prove "I am over 21" without revealing anything else—not your name, not your birthday, not even how old you are, just that you meet the requirement. Instead of handing over a physical driver's license, it's possible to generate a ZKP that you possess a valid, DMV-issued credential and that the age encoded in that credential is over 21, without revealing any other personal details. The agent gets a cryptographic "yes" or "no," and you keep the rest of your personal data private.

Now consider zero-knowledge transactions for government missions, where information is often very sensitive. With ZKPs, it's possible to prove clearance levels without exposing personnel files, prove software integrity without revealing source code, prove compliance without sharing raw logs, and much more.

The shift to proof without disclosure, for illustrative purposes only.

How ZKPs Work

ZKPs mark the shift from disclosure-based verification to mathematics-based verification. Instead of sharing raw data, a prover generates a mathematical proof that a statement about the data is true.

There are three properties that fundamentally define ZKPs:

1. Completeness: If the statement is true and both parties follow the protocol correctly, an honest verifier will be convinced by the honest prover.
2. Soundness: If the statement is false, no dishonest prover can convince the honest verifier that it is true except with negligible probability.
3. Zero-knowledge: The verifier learns nothing beyond the fact that the statement is true, meaning no additional information about the underlying secret is revealed.

So, how do ZKPs work? Under the hood, ZKPs rely on cryptography to separate verification from disclosure. Here is the high-level sequence of this methodology in action: The prover commits to some secret information (e.g., your actual birthdate) in a way that is binding (you can't change it) but hidden (no one can see it). The proof encodes a logical statement about that hidden data (e.g., "this date is more than 21 years prior to today's date"). Finally, a verifier can check the proof using public parameters and cryptographic rules, but—critically—cannot reverse the proof to learn the underlying data.

Common Types of ZKPs

The most widely deployed ZKP systems today are Succinct Non-Interactive Arguments of Knowledge (zk-SNARKs) and Scalable Transparent Arguments of Knowledge (zk-STARKs). These categories are useful because they capture high-level system properties—such as proof size, verification cost, and setup requirements—and they are often how ZKPs are discussed in practice. However, they are not defined by a single underlying cryptographic assumption, and so they do not form a clean, mutually exclusive taxonomy.

Let's start by looking at zk-SNARKs, which emerged earlier. Many widely used constructions such as Groth16 and PLONK rely on bilinear pairings over elliptic curves. In these systems, security is grounded in hardness assumptions from elliptic-curve cryptography, including discrete logarithm-type problems and pairing-based assumptions. These proofs are extremely small and fast to verify, which has made them attractive for blockchain applications. In the driver's license example, this is what makes the interaction practical: A user can prove they hold a valid DMV-issued credential and are over 21 with a succinct proof that the verifier can check instantly. However, pairing-based SNARKs typically require a trusted setup (though there are variants with universal or updatable setups) and are not believed to be resistant to quantum attacks—a limitation that will become increasingly important as quantum computing advances.

A BRIEF HISTORY OF ZERO-KNOWLEDGE PROOFS

1980s

The first ZKPs were introduced as a theoretical and academic breakthrough.

1990s

Niche security applications were developed, but they were computationally expensive and impractical at scale.

2010s

Succinct non-interactive ZKPs emerged. The blockchain boom forced real-world engineering; zk-SNARKs and zk-STARKs matured under economic pressure.

2020s

Tooling and performance have become viable beyond crypto. What was once niche cryptography is now production-grade verification infrastructure.

SEQUENCE OF VERIFICATION WITH ZKPS

Prover and verifier agree on the rule to be proven

Before anything happens, the prover and verifier agree on:

- The statement to be proven (for example, "This AI model ran within approved constraints")
- The rules or policy that define what "correct" means
- The verification method to be used

No data is shared at this stage.

Prover commits to the sensitive information

The prover takes the sensitive inputs (w) and creates a cryptographic commitment, which:

- Prevents (w) from being changed later
- Hides (w) from the verifier
- Ensures the prover cannot cheat by swapping inputs after the fact

Think of this as creating a tamper-evident seal on the information.

Prover generates a ZKP

Using (w) and the agreed-upon rules, the prover runs a proof-generation process that produces a ZKP, which mathematically demonstrates that:

- (w) satisfies the rule
- The prover knows (w)
- No additional information about the data is revealed

Importantly, the proof does not contain (w) itself, the model logic, or internal system details.

Prover sends the proof to the verifier

The prover sends:

- The ZKP
- A reference to the rule or statement that was proven

The prover does not send the underlying sensitive information (w).

Verifier checks the proof

The verifier uses a verification algorithm to mathematically check the proof. This check:

- Does not require trust in the prover
- Does not depend on the prover's hardware, software, or environment

If the proof verifies, the statement is true; if it does not, the statement is false or invalid.

Verifier acts on the result

Based on the result, the verifier can:

- Grant or deny access
- Accept or reject data
- Approve or fail an audit
- Allow or block a system interaction

All without ever seeing or storing (w).

KEY TERMS:

Prover: The system or party that has the sensitive information and wants to prove something about it.

Verifier: The system or party that wants assurance the claim is true, but does not want—or is not allowed—to see the sensitive information.

(w): The sensitive inputs (e.g., data, model parameters, logs, identity attributes).

By contrast, zk-STARKs are designed to be transparent and avoid trusted setup entirely. Constructions such as Aurora and Ligero rely on hash-based cryptography and techniques from error-correcting codes. Their security ultimately depends on the collision resistance of cryptographic hash functions—a standard, conservative assumption in modern cryptography. These functions have been extensively studied and are widely believed to make it computationally infeasible to find two distinct inputs that produce the same hash output, even for quantum-capable adversaries. This makes such zk-STARK constructions plausibly secure against quantum adversaries. STARKs also scale well to very large computations, though they tend to produce larger proofs and have higher verification costs compared to pairing-based SNARKs.

As alluded to above—where we introduced the underlying assumptions behind common SNARKs and STARKs—a more fundamental way to classify ZKPs is by these underlying hardness assumptions rather than by system-level properties like "SNARK" or "STARK." From this perspective, most practical systems fall into the following categories:

- Pairing-based/elliptic-curve-based, which underpin many SNARKs
- Hash-based, which underpin STARKs and other transparent proof systems
- Discrete-log–based, which include protocols built directly from sigma protocols and related techniques

STARKs also scale well to very large computations, though they tend to produce larger proofs and have higher verification costs compared to pairing-based SNARKs.

Other families—such as lattice-based, isogeny-based, and code-based ZKPs—are also active areas of research, particularly in the context of post-quantum cryptography, although they are generally less mature for large-scale, general-purpose proving.

In short, "SNARK" and "STARK" describe how a proof system behaves, while hardness assumptions explain why it is secure. The two lenses overlap but are not interchangeable, so understanding both is key to navigating the design space of modern ZKPs.

Spotlight Application: Zero Trust

Zero trust (ZT) security frameworks assume that networks are hostile, identities can be compromised, and no request should be implicitly trusted based on location or prior authentication. Every access decision must be continuously verified and limited to the least privilege necessary.

However, traditional verification mechanisms often require over-disclosure of sensitive information. To prove eligibility, compliance, or identity, users and systems typically transmit full records, credentials, logs, or configuration data. This creates unnecessary data exposure and expands the attack surface.

If ZT means never trust and always verify, then the next logical question for security teams is: Can we verify without disclosing?

From Identity Assertions to Cryptographic Proofs

ZKPs provide the cryptographic backbone for the next phase of ZT: enabling verification without disclosure, continuous assurance without surveillance, and mission execution in environments where no single node or operator can be fully trusted. By shifting from data verification to mathematics, systems don't need to evaluate user attributes but can still prove those attributes satisfy enterprise policies.

With ZKPs, no raw attributes are transmitted. No centralized identity query is required. No additional metadata is revealed. And verification can happen locally, to minimize exposure.

Beyond identity and access control, ZKPs extend ZT principles to the computation itself. For example, in untrusted or outsourced environments such as cloud services or third-party processing systems, ZKPs could soon enable verifiable computation, allowing a provider to prove that a workload was executed correctly according to previously agreed-upon rules. This shifts trust from infrastructure and contractual assurances to cryptographic guarantees at the application layer.

Taken together, these capabilities align directly with ZT's core requirement: constant verification under conditions of assumed compromise. ZKPs provide the cryptographic primitive that allows such verification to occur without expanding exposure. They enable proof without revelation, reduce centralized data risk, support privacy-preserving identity, and allow both compliance and computation to be verified in adversarial environments. In the event of a data breach, ZKPs can reduce the "blast radius" by minimizing the amount of data collected and stored, so that any incident exposes only limited or unusable information rather than full sensitive records. For modern distributed systems, particularly those spanning multiple organizations or regulatory domains, this capability makes ZKPs a powerful enabler of next-generation ZT architectures.

THE ARCHITECTURAL SHIFT OF ADDING ZKPS TO THE ZT TOOLBOX

Current ZT Model Verification Flow	ZT+ ZKP Verification Flow
1. User requests access	1. User wants access
2. System queries identity provider	2. User locally generates a proof
3. Identity provider returns attributes	3. User sends only the proof
4. Policy engine evaluates attributes	4. Policy engine verifies the proof
5. Access granted or denied	5. Access granted or denied

Use Cases on the Horizon

Within the United States and across governments around the world, ZKPs are actively in development for privacy-preserving digital identity, credential verification, and regulatory compliance, allowing users and institutions to prove attributes or satisfy requirements without exposing underlying personal or sensitive data.

Promising research and prototypes already emerging, with mission-aligned applications such as:

Proof of Human. ZKPs can be combined with biometrics and behavioral signals to let individuals prove they are a unique, real human—rather than a bot or synthetic identity—without revealing the underlying personal data used to verify that claim.

Cyber Defense Against Deepfakes. With the rise of agentic AI and high-quality deepfakes in 2026, ZKPs are being used to help authenticate digital content. For example, news organizations and government bodies could eventually use ZKPs to sign metadata, proving a video was captured by a specific trusted camera at a specific time without revealing the GPS coordinates of a sensitive location.

Verifiable AI and Automated Decisions. Government systems may gain the ability to generate proofs that algorithms and models are executed correctly and according to preapproved rules.

Secure Inter-Agency Computation. Agencies and partners could verify insights derived from sensitive data without sharing the raw datasets or revealing sources and methods.

Privacy-Centric Citizen Services. Citizens could one day prove eligibility, age, residency, income bracket, or other attributes for services without disclosing full personally identifiable information or sensitive documents.

Secure Cross-Domain Data Exchange. Operators handling classified data could prove policy compliance before releasing information across security boundaries.

Regulatory Compliance and Auditing. Organizations could prove compliance or audit outcomes without sharing logs, sensitive operational details, or proprietary information.

Together, these use cases illustrate how ZKPs can extend trust across domains where data sensitivity, operational security, and verification requirements would otherwise be in tension.

Market Dynamics to Watch

Startups Are Expanding Beyond Crypto Tooling

Startups are still the primary innovators of ZKPs. The market for pure-play ZKPs is relatively small, but the growing interest in privacy, scalability, and ZT architectures is attracting meaningful capital. While startups continue to focus on ZKP for privacy-preserving blockchain applications, many are also expanding into broader practical applications across multiple sectors.

What we're watching:

- Post-2020, average investment deal size and investor quality have surged, reflecting growing institutional confidence in horizontal ZKP infrastructure as a foundational technology beyond any single blockchain or application. For example, Matter Labs has attracted investment from top venture funds, including a16z, Lightspeed Ventures, and Coinbase Ventures.
- Emerging players are advancing ZKP infrastructure to improve scalability and efficiency—often through techniques that move computation off primary systems while preserving verifiability. For example, Succinct Labs has developed a prover network and an open-source zero-knowledge virtual machine (zkVM) to support scalable, developer-friendly proof generation.
- Leading startups are partnering with major cloud providers to accelerate adoption and integration. For example, Space and Time is integrating real-time, verifiable blockchain data into Microsoft Fabric and Azure, and working with Google Cloud BigQuery to enable verifiable queries within data warehousing environments.

Big Tech Is Actively Developing Scalable ZKP Frameworks

As the last decade of cryptocurrency development has shown, ZKPs are no longer theoretical. Now, an ecosystem beyond blockchain is beginning to take shape—driven initially by startups and increasingly scaled by Big Tech. While startups continue to lead in ZKP innovation, large technology companies are focusing on enterprise adoption and scalability, particularly by laying the groundwork for verifiable computation. Their involvement is helping shift trust from centralized systems to cryptographic guarantees, enabling proof-based access and computation. By accelerating adoption, standardization, and cross-platform interoperability, Big Tech is pushing ZKPs beyond niche use cases into mainstream cybersecurity and information security applications for both enterprises and consumers.

What we're watching:

- Google has integrated ZKP protocols into the Android operating system, enabling third-party developers to request proofs instead of raw data (e.g., a banking app verifying sufficient funds without accessing the exact account balance).
- Microsoft Research and Azure are exploring ZKPs within decentralized identity and confidential computing frameworks, aiming to enable proofs of data handling or compliance without exposing underlying content.
- Apple uses ZKP-style cryptography in Apple Wallet to verify that users are over 21 without revealing their name or date of birth to merchants.
- Meta has developed the Winterfell STARK prover and is actively researching ZK-based approaches for privacy-preserving and verifiable computation.

What's Next for the Technology

Projected Timeline of Development

The trajectory of ZKP adoption is likely to follow a familiar pattern: early technical maturation, followed by integration and standardization, and ultimately institutionalization across systems and policy frameworks. We expect ZKPs to reach a tipping point of commercial maturity outside the blockchain ecosystem within two to five years, at which point they will be more broadly evaluated and adopted as an adjunct to existing ZT frameworks.

TRAJECTORY OF ZKP ADOPTION

0–2 Years

PRACTICAL MATURATION

- Faster provers, smaller proofs, and better developer tooling
- Early production use in identity, access control, supply chain verification, and security auditing
- Federal pilots and early deployments
- ZK-native identity and credentials rep

2–5 Years

INTEGRATION + STANDARDIZATION

- ZK-native identity and credentials replace static identifiers and attribute sharing
- ZKP modules are integrated into identity and access management stacks
- Verifiable computation becomes standard for select cloud, edge, and AI workloads
- Early standards emerge for ZK-enabled ZT frameworks

5–10 Years

INSTITUTIONALIZATION

- Standards are fully defined by NIST and adopted across U.S. government systems
- Cryptographic compliance becomes the default, with ZKPs embedded at protocol and hardware layers
- Privacy-preserving collaboration becomes feasible at scale across enterprises, agencies, and allies—even in adversarial environments

An Evolving Security Discipline

The trajectory for ZKPs is clear and very promising, but there are challenges and limitations to navigate before this security discipline becomes fully mainstream outside crypto-native environments:

- **Computational cost:** Generating proofs can be expensive, especially for complex logic or large datasets. While this is improving rapidly, prover-side costs remain a key bottleneck.
- **Latency:** Proof generation can introduce delays, which matters for real-time or tactical systems.
- **Engineering complexity:** Writing ZK-friendly programs is difficult and often requires redesigning systems around arithmetic circuits rather than normal software abstractions.
- **Integration with legacy systems:** Most enterprise and government infrastructure was not designed with cryptographic proofs in mind, so retrofitting ZKPs requires significant systems engineering.
- **Standards and assurance:** For government adoption, ZKPs still require more formal standards, certification pathways, and operational trust models.

ZKPs are early in the standards curve, similar to where post-quantum cryptography was a decade ago. We expect adoption will follow the maturation of tooling and assurance frameworks, and, fortunately, those wheels are already turning. Today, NIST's Privacy-Enhancing Cryptography (PEC) initiative—focused on developing a range of cryptographic tools—is beginning to shape the standardization of ZKPs. Its goal is to make these protocols interoperable, auditable, and enterprise-ready, while integrating them with other cryptographic approaches rather than treating ZKPs as a standalone technology.

The Emerging Assurance Stack

ZKPs do not replace existing identity or ZT tools. Rather, ZKPs are a natural addition to the layered security toolbox, reinforcing assurance at each layer of the stack:

They extend identity, access, and ZT tools by reducing the amount of information identities must reveal.

They complement secure enclaves by adding cryptographic verification to systems that otherwise rely mostly on trusted hardware.

LAYERS OF THE ASSURANCE STACK WITH ZKPS

LAYER 4 *ZKPs used here*	**Trust, Identity, & Interaction** Establishes trust between systems, users, or organizations *(includes 21 architectures)*
LAYER 3 *ZKPs defined here*	**Cryptographic Proof & Privacy** Lets one party convince another of correctness without revealing sensitive information
LAYER 2 *AR used here*	**Correctness & Assurance** Ensures systems behave as intended (includes formal verification, model checking, etc.)
LAYER 1 *AR defined here*	**Computation & Logic** Figures out what is true or what satisfies constraints (includes automated reasoning, formal logic engines, etc.)

They share mathematical DNA with automated reasoning and solve problems in different layers of the assurance stack.

ZKPs now sit at the intersection of cryptography, distributed systems, and national-scale security concerns, and the industry is approaching a critical inflection point in their development. What began as a curiosity has become a practical tool for verifying trust without disclosure—precisely what modern, adversarial, and data-sensitive environments demand.

The Booz Allen Tech Scouting and Intelligence team scans the market for novel technologies with the potential to disrupt U.S. government critical missions. Delivering research, trends spotting, and deep dive technical analysis, the team is hyper-focused on getting the best technology from the private sector into the hands of the warfighter and civilian federal workforce.

Livia Dworaczyk and Meghan Hauser are members of the Tech Scouting team, and Tom Hanson is part of Booz Allen's Post-Quantum Cryptography (PQC) practice.

SPEED READ

Zero-knowledge proofs (ZKPs) are emerging as a transformative way for government and industry to break the long-standing "proof vs. privacy" tradeoff—enabling verification of identity, compliance, and computation without exposing underlying sensitive data.

Once engineered largely for blockchain, modern ZKP systems are now maturing into mission-ready infrastructure that can support zero trust architectures, verifiable AI, secure inter-agency analytics, deepfake-resistant content authentication, and privacy-preserving citizen services.

As federal agencies explore pilots and as startups and Big Tech accelerate scalable frameworks, ZKPs are on track to become a pivotal layer of the government assurance stack—reducing data exposure, enabling cryptographic trust, and reshaping how sensitive systems operate in adversarial environments.

Booz Allen®