Unlock the potential of zero-knowledge proofs

At a high level, a zero knowledge proof is a method that enables you to prove you have knowledge of, access to, or ownership of information without revealing anything about the information to a verifying party. The proof requires two key participants:

- The proving party (prover) that makes a claim without providing any information except the claim itself.
- The verifying party (verifier) that—by some means—determines the claim is valid without learning anything about the information that the proving party possesses.

What constitutes a zero-knowledge proof?

The conceptual underpinnings for zero knowledge proofs have been around for almost 40 years. In fact, the defining properties for a zero knowledge proof were first introduced in 1985, in the paper The Knowledge Complexity of Interactive Proof-Systems by Shafi Goldwasser, Silvio Micali, and Charles Rackof. These defining properties are:

- Completeness. If the claim is valid, an honest verifier can be convinced of it by an honest prover.
- Soundness. If the claim is false, no cheating prover can convince the honest verifier that it is true.
- Zero-Knowledge. If the claim is valid, the verifier learns nothing other than this fact; the verifier gains zero knowledge about why the statement is true.

Zero-knowledge proofs can take two forms: interactive or non-interactive.
With interactive zero-knowledge proofs, the prover and verifier engage in a series of challenges and responses to complete the proof. For non-interactive zero-knowledge proofs, the prover only needs to send a single message—a cryptographic hash—to the verifier. The verifier can determine the validity of the message without any further engagement from the prover.

Why do zero-knowledge proofs matter?

So, after almost 40 years and the ensuing advancements in modern cryptography since that time, why are zero-knowledge proofs a hot topic now?
Mainly, the growing interest in zero-knowledge technology—including the protocols that implement zero-knowledge proofs—is due to the vast expansion and importance of the internet.

Once almost exclusively the domain of scientists, government agencies, and academia with limited interactive capabilities, the internet's second wave of technical advancements and innovation—web2.0—brought the internet to the people, changing how we do just about everything. However, this ongoing digital transformation has exposed some costs and limitations across the technology stack that most of today's internet depends on.
For example, consolidation in social media and other industries has given a small number of organizations a great deal of control over user data and innovation while limiting the ways users can interact and control their own information.

Increasingly, the world accesses products and services online that require individuals to register by providing user names, passwords, email addresses, passport numbers, or bank and credit card information. All of this online activity puts our personal information at risk. Often, companies that require this information to establish accounts and authenticate user identities harvest this personal information and misuse it: directly through targeted marketing campaigns, indirectly by selling information to data brokers, or unintentionally through data breaches.

Developing applications with zero-knowledge technology

These threats to how personal information is handled and shared are why it's important for application developers to have tools that enable them to incorporate zero knowledge proofs into the programs and application workflows they build. By adopting zero-knowledge protocols, developers of all backgrounds can build more secure products that protect users' privacy.

Imagine for a moment that you could prove:

- You're over 21 without providing your age or date of birth.
- You qualify for a loan without disclosing financial records.
- You live in a certain jurisdiction without revealing where you live.
- You're a citizen of a given country without displaying a government-issued document like a driver's license or passport.

Zero-knowledge protocols—such as zk-SNARK and zk-STARK—provide the tools for developers to build the applications that support these types of real-world scenarios. The protocols themselves might take different approaches and have different trade-offs in efficiency and security, but they provide the foundation for building the applications of the future.