Relations Between Sigma Protocols, ZK Proofs, SNARKs, and Related Concepts

1. Sigma Protocols and Zero-Knowledge Proofs

Sigma Protocols

Sigma protocols are interactive proofs designed to prove knowledge of a witness (e.g., a secret key or discrete logarithm) without revealing it. They have the following properties:

- **Completeness**: An honest prover convinces an honest verifier.
- **Special Soundness**: Given two valid transcripts with the same commitment and different challenges, one can extract the witness.
- **Special Honest-Verifier Zero-Knowledge (SHVZK)**: A transcript can be simulated without the witness if the verifier's challenge is known.

Typical structure:

$$\begin{aligned} \text{P} \rightarrow \text{V} &: \text{Commitment } a \\ \text{V} \rightarrow \text{P} &: \text{Challenge } e \\ \text{P} \rightarrow \text{V} &: \text{Response } z \end{aligned}$$

Relation to Zero-Knowledge Proofs

Sigma protocols are a subclass of zero-knowledge proofs under the honest verifier assumption. By applying the Fiat-Shamir transform in the Random Oracle Model (ROM), they become non-interactive and fully zero-knowledge:

$$\text{Sigma Protocol} \xrightarrow{\text{Fiat-Shamir}} \text{NIZK (in ROM)}$$

2. Where PLONK and Groth16 Fit In

PLONK and Groth16 are **succinct non-interactive zero-knowledge proofs** (SNARKs):

- Prove arbitrary computations.
- Non-interactive.
- Zero-knowledge.
- Succinct (small proof sizes).

Scheme	Type	Interactive?	ZK?	General Purpose?	Trusted Se
Sigma	PoK, interactive	Yes	SHVZK	Limited	No
Fiat-Shamir(Sigma)	NIZK	No	Yes (ROM)	Limited	No
Groth16	zk-SNARK	No	Yes	Yes	Yes (per circ
PLONK	zk-SNARK	No	Yes	Yes	Yes (univer

3. Where PCPs, IPs, MIPs, IOPs, PCS, VRFs, and MPC Fit In

Foundational Proof Models

- **IP (Interactive Proofs)**: General model with full interaction.
- **MIP (Multi-Prover IPs)**: Multiple non-communicating provers. More powerful than IP.
- **PCP (Probabilistically Checkable Proofs)**: Verifier queries a few bits of a long proof.
- **IOP (Interactive Oracle Proofs)**: Interactive version where prover sends queryable strings (oracles).

Modern SNARKs (e.g., PLONK, STARKs) are built from IOPs combined with polynomial commitments and made non-interactive via Fiat-Shamir.

Polynomial Commitment Schemes (PCS)

A PCS allows one to:

- Commit to a polynomial $f(x)$.
- Later reveal $f(z)$ at a point z with a verifiable proof.

Examples:

- KZG (used in PLONK)
- FRI (used in STARKs)

Verifiable Random Functions (VRFs)

- Outputs random-looking values with a proof of correctness.
- Used in blockchains (e.g., Algorand), lotteries, and elections.
- Often constructed using group operations and sometimes zero-knowledge arguments.

Secure Multi-Party Computation (MPC)

- Enables computation over private inputs distributed across parties.
- Used to generate SNARK proofs securely or to simulate ZK proofs (e.g., MPC-in-the-head).

4. Unified Picture

Complexity Classes:

```
PCPs → IOPs → SNARKs
             ↓
      Polynomial Commitments
             ↓
       Fiat-Shamir → NIZK
```

Other Tools:

```
MPC |→ SNARK Setup or ZK Simulation
VRF |→ Uses ZK to prove correctness of randomness
```

5. Summary Table

Term	Type	Interactive?	ZK?	Application
Sigma	Protocol	Yes	Yes (HVZK)	PoK
PCP	Complexity Class	No	No	Theory
IOP	Protocol	Yes	Yes	SNARKs
PLONK	zk-SNARK	No	Yes	Circuits
Groth16	zk-SNARK	No	Yes	Circuits
PCS (KZG)	Commitment	No	No	Inside SNARKs
VRF	Primitive	No	No (may include ZK)	Verifiable randomness
MPC	Protocol	Yes	Yes	Private computation