# Can we formally verify implementations of cryptographic libraries like the c-kzg library?

**Thanh-Hai Tran**
Independent Researcher

**Brett Decker**
Galois

**Marcella Hastings**
Galois

**Ryan McCleeary**
Galois

**Roberto Saltini**
Independent Researcher

# Implementing crytographic libraries is challenging



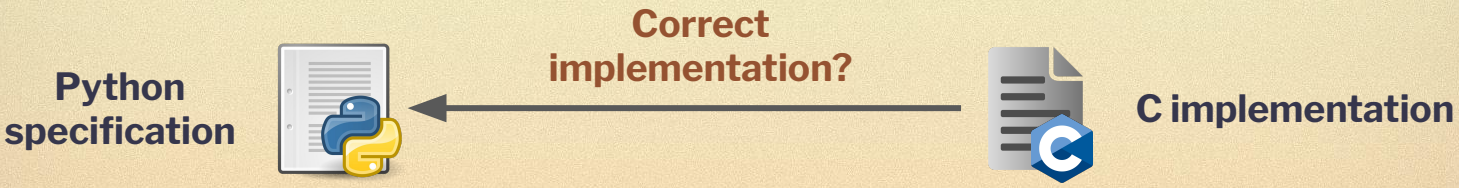

# Formal methods in cryptography

1. Correctness of cryptographic algorithms
2. Correctness of cryptographic libraries

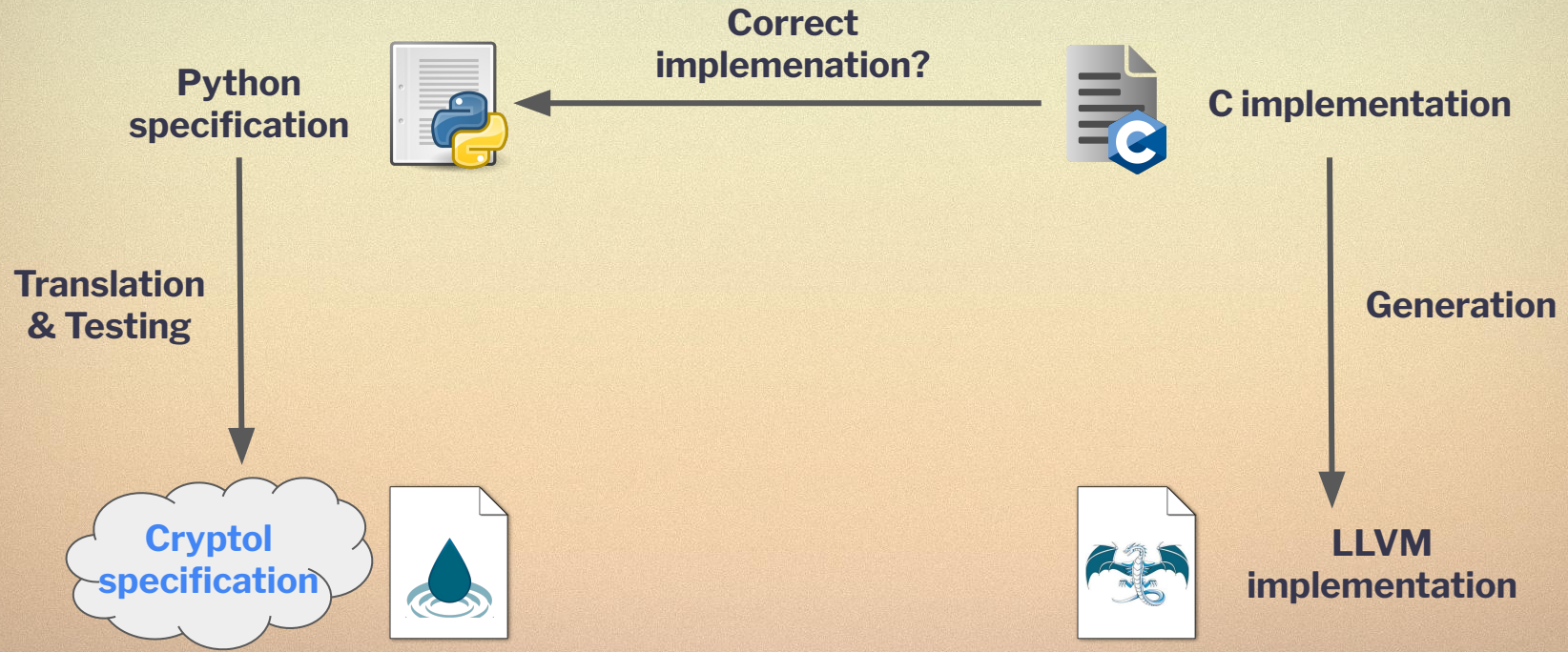


SSProve

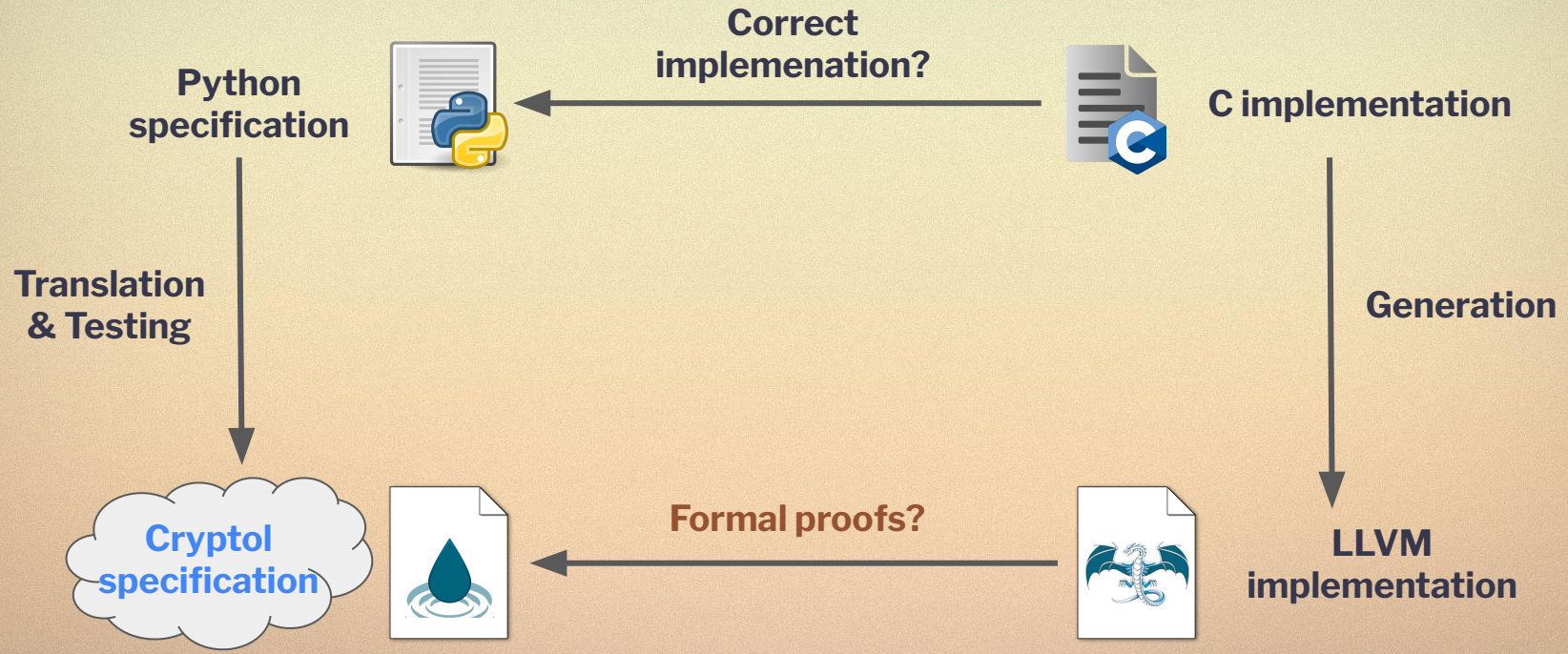
# Example: the c-kzg library in EIP 4844



# Verification of the c-kzg library



# Verification of the c-kzg library



# Verification of the c-kzg library (cont)

- Finished writing a Cryptol specification

```
compute_kzg_proof : Blob -> Bytes32 -> (KZGProof, Bytes32)
compute_kzg_proof blob z_bytes = (proof, y) where
    polynomial = blob_to_polynomial blob
    (proof, y') = compute_kzg_proof_impl polynomial (bytes_to_bls_field
                                                     z_bytes)
    y = split`{BYTES_PER_FIELD_ELEMENT} y'
```

- Tested the equivalence between Python functions and Cryptol functions
  - `compute_kzg_proof, evaluate_polynomial_in_evaluation_form, …`
- Formally proved correctness of some C functions
  - `bit_reversal_permutations, reverse_bits, …`

# Formal verification in cryptography

**Thanh-Hai Tran**

thanh_hai_tran

thanh_hai_tran