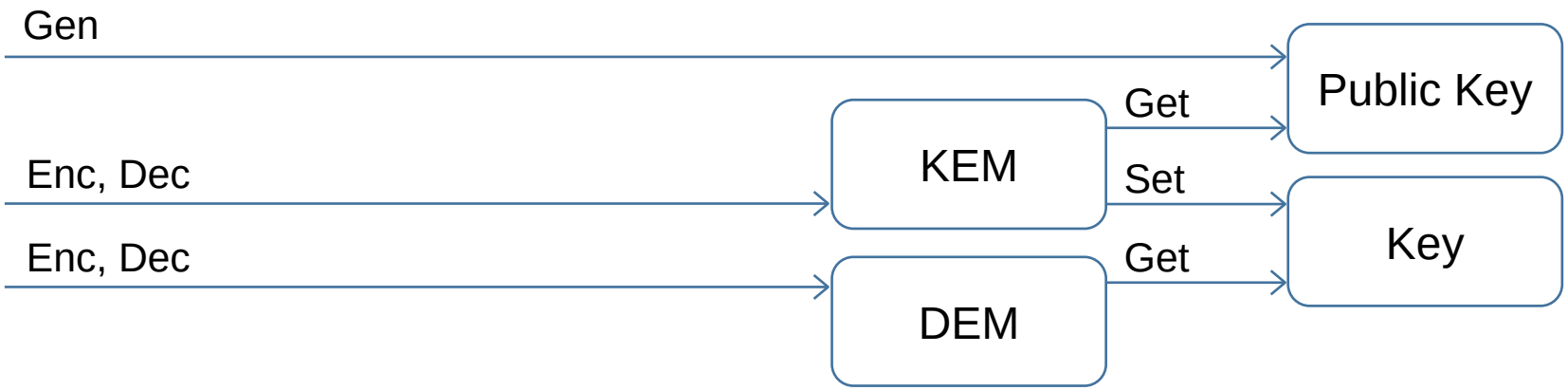
SSPVerif: Verifying State Separating Proofs

Chris Brzuska, Christoph Egger, **Jan Winkelmann**

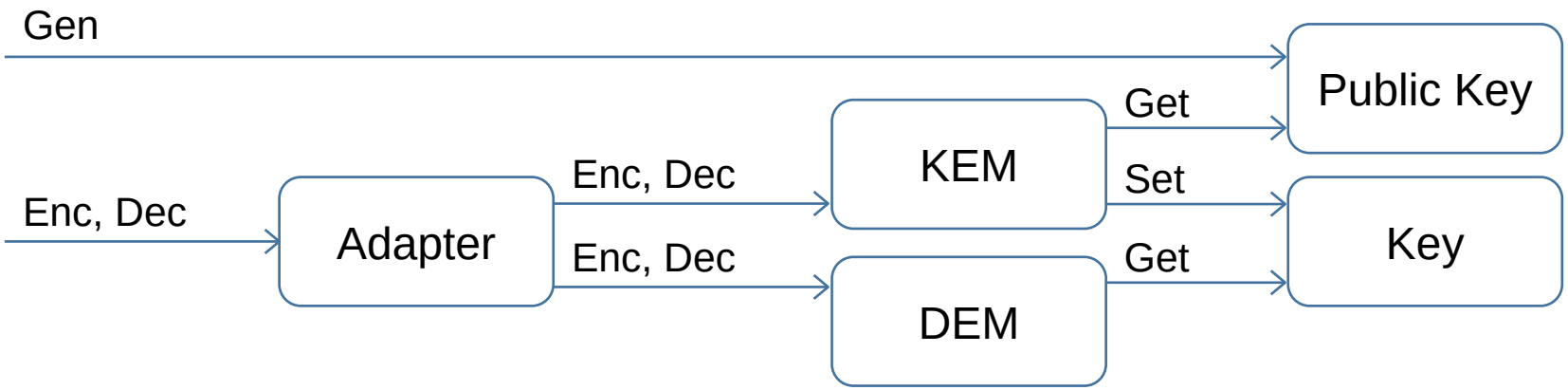
Context

- State Separating Proofs modularize computational cryptographic security proofs
  - Simple, obvious reductions
- Motivating Example: Hybrid Encryption from KEM/DEM
  - KEM: Key Encapsulation, asymmetric
  - DEM: Data Encapsulation, symmetric
  - HPKE: Hybrid Public Key Encryption

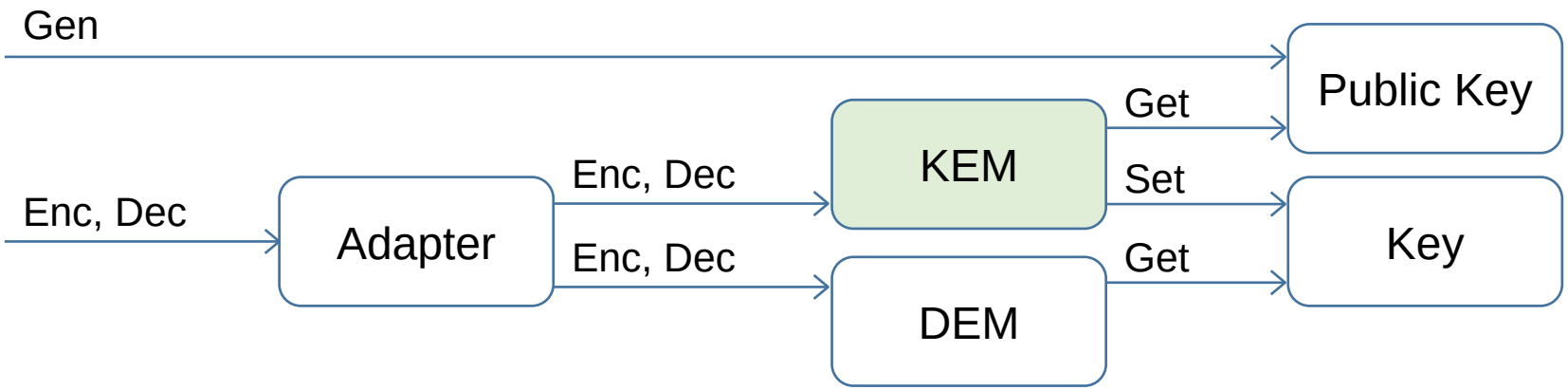
Context: KEM-DEM



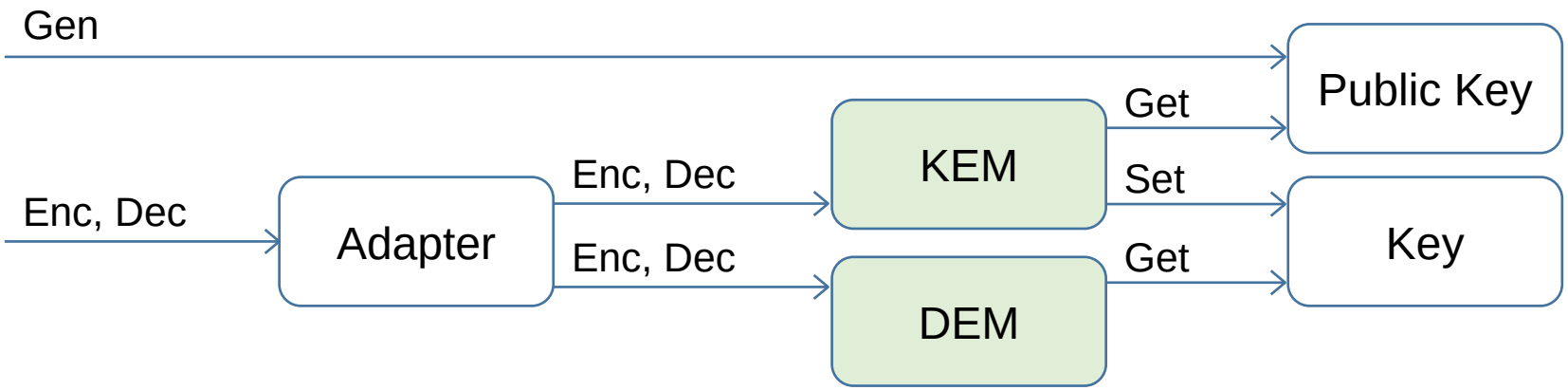
Context: KEM-DEM as HPKE



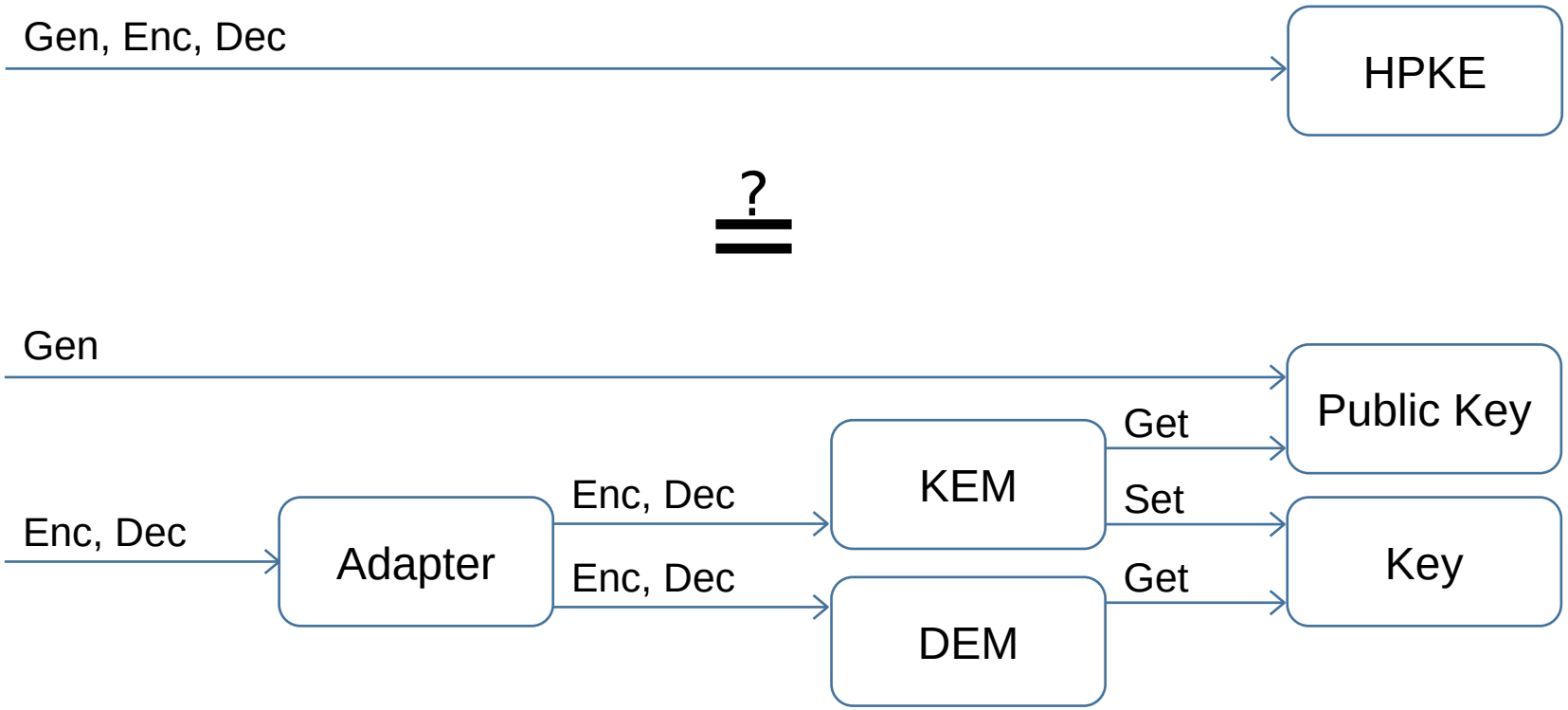
Context: KEM-DEM as HPKE



Context: KEM-DEM as HPKE



Problem

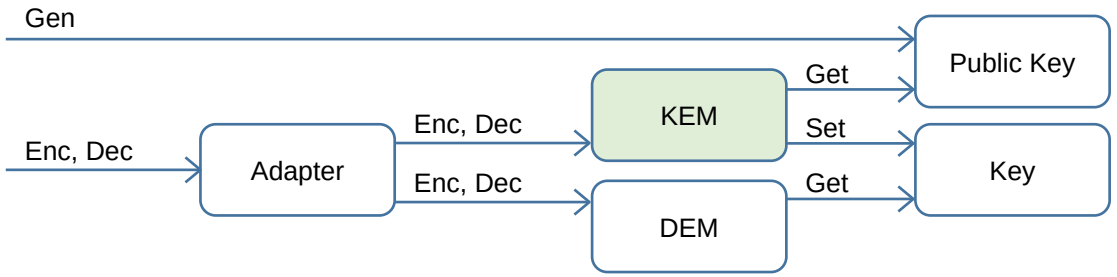

Gen, Enc, Dec
HPKE
?
Gen
Public Key
Get
KEM
Enc, Dec
Set
Enc, Dec
Adapter
Key
Enc, Dec
Get
DEM

SSPVerif

- Parses a language close to Pseudocode
- Generates SMT-LIB code
  - Packages
  - Helpers
- Consumes User-Provided Invariants (SMT-LIB)
- Invokes SMT solver (cvc5, z3) to determine equivalence
- Also: Typecheck code, compositions and reductions
- Planned: Better error reporting, tools that give insights into proofs

Challenges

- Users have to be relatively fluent in SMT-LIB to
  - Write invariants
  - Match the randomness between games
  - Debug their code
- Dealing with for loops



Case Study

- Formalizing Proof of Yao's Garbling Scheme by Brzuska and Oechsner
- Something with Key Exchange

Hopes

- Verify SSPs for very large protocols
- Complementary to EasyCrypt, captures orthogonal complexity