C'est très CHIC: A compact password-authenticated key exchange from lattice-based KEM

Afonso Arriaga[1], Manuel Barbosa[2,3,4], Stanislaw Jarecki[5], Marjan Škrobot[1]

(1) SnT - University of Luxembourg, Esch-sur-Alzette, Luxembourg
(2) FCUP, University of Porto, Porto, Portugal
(3) INESC TEC, Porto, Portugal
(4) Max Planck Institute for Security and Privacy, Bochum, Germany
(5) University of California, Irvine, USA

Asiacrypt' 24, Kolkata, India

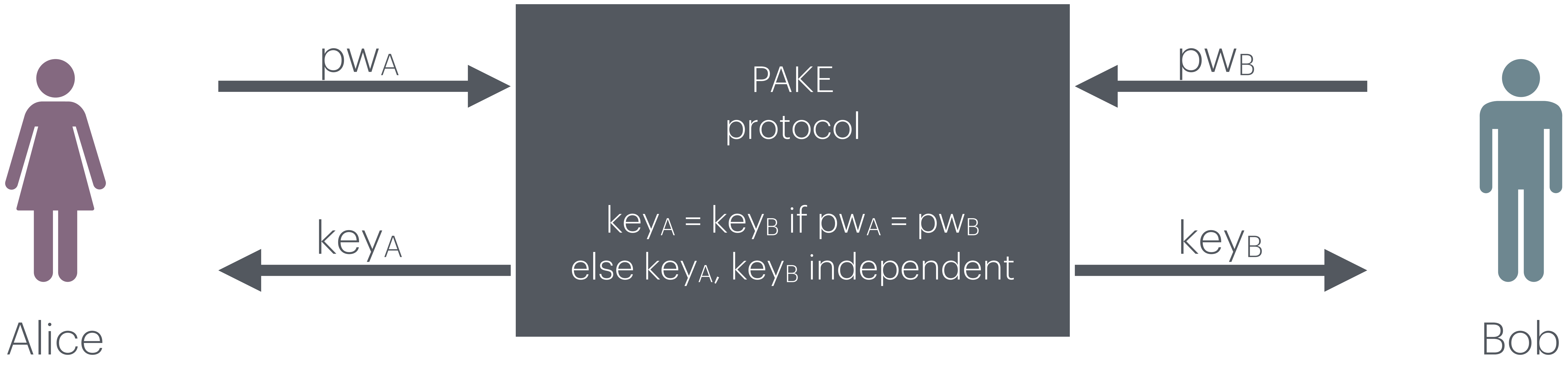
UNIVERSITÉ DU LUXEMBOURG

SNT

Luxembourg National Research Fund

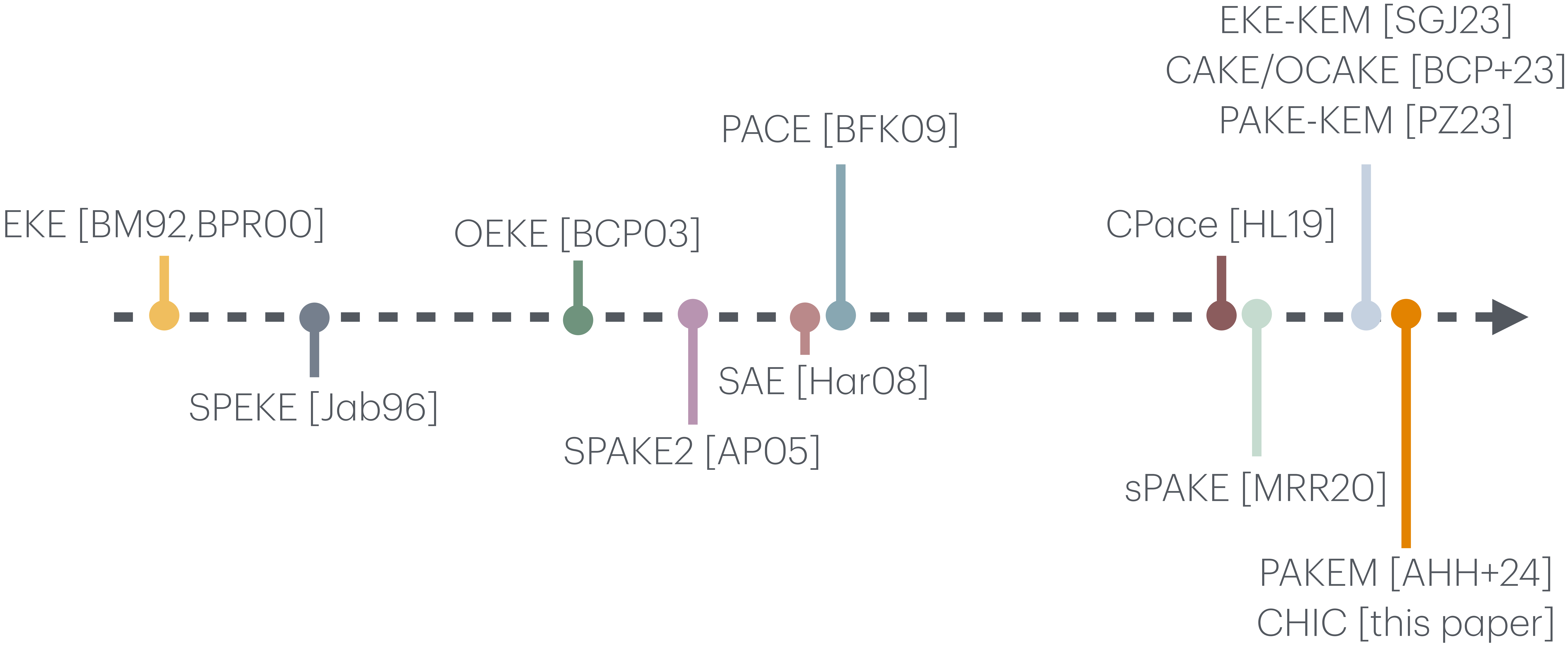
PAKE: Password Authenticated Key Exchange

Protocol



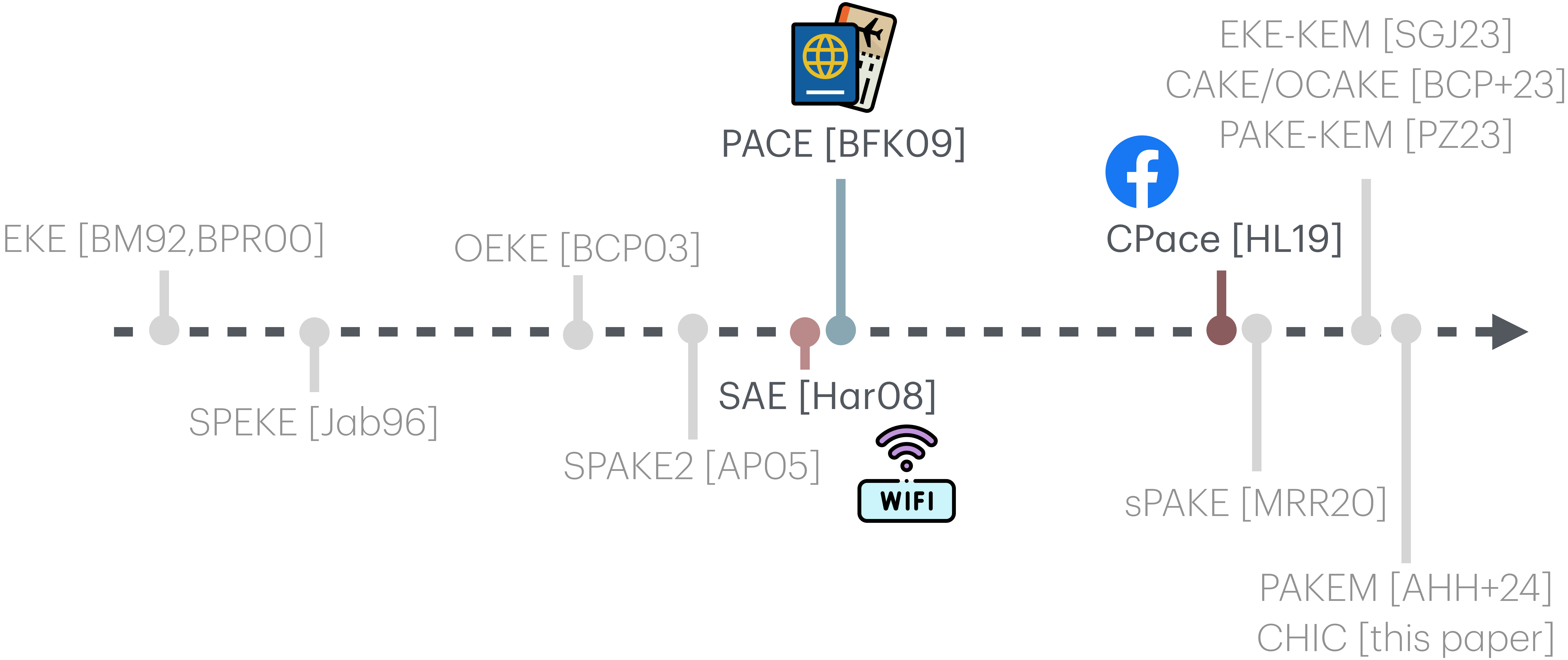
PAKE: Password Authenticated Key Exchange

Timeline



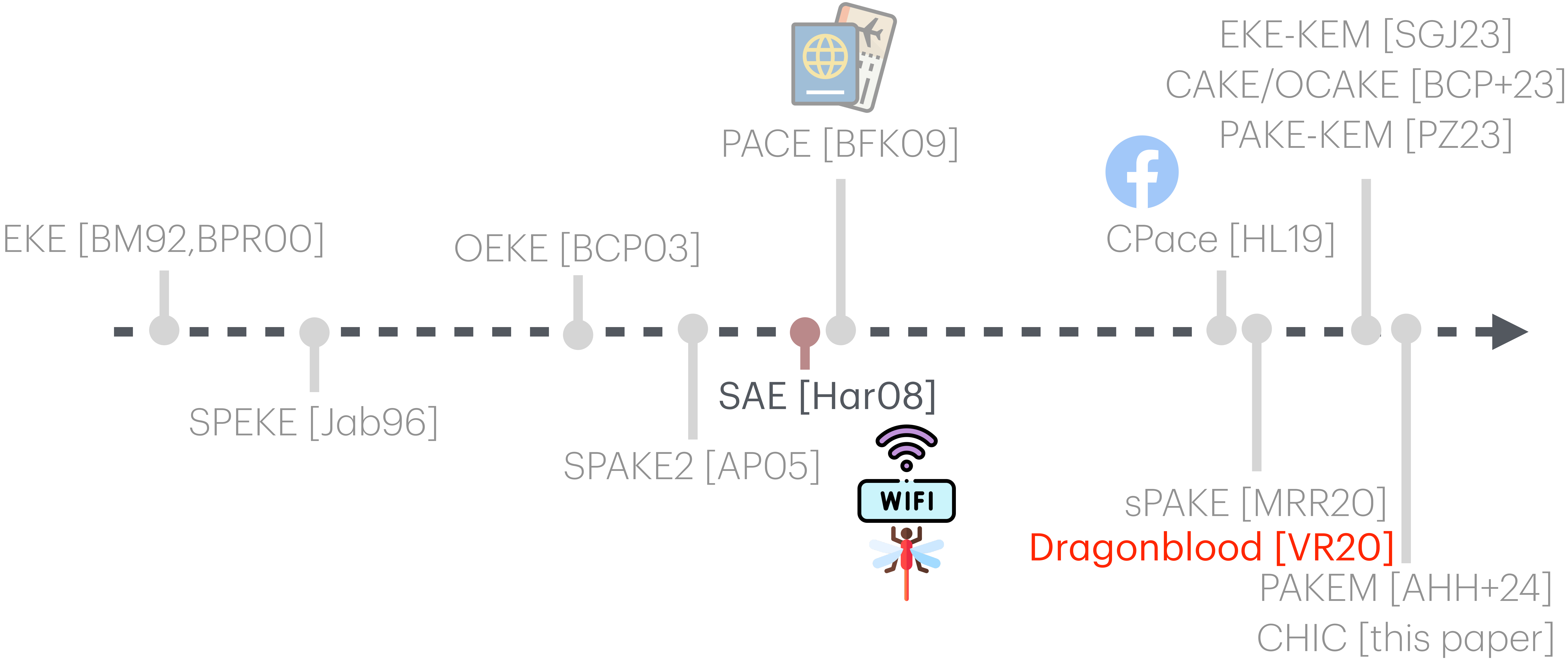
PAKE: Password Authenticated Key Exchange

Real-world adoption of symmetric PAKE protocols



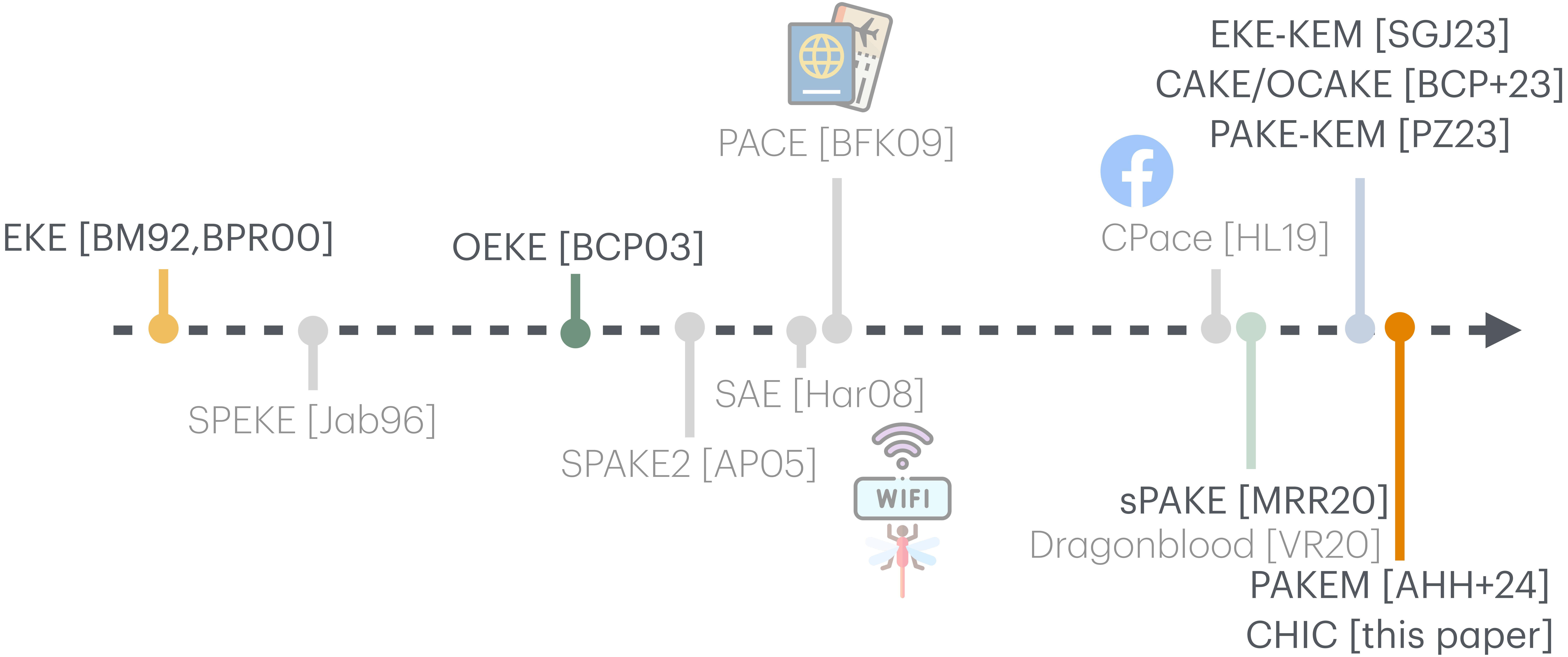
PAKE: Password Authenticated Key Exchange

Real-world attacks

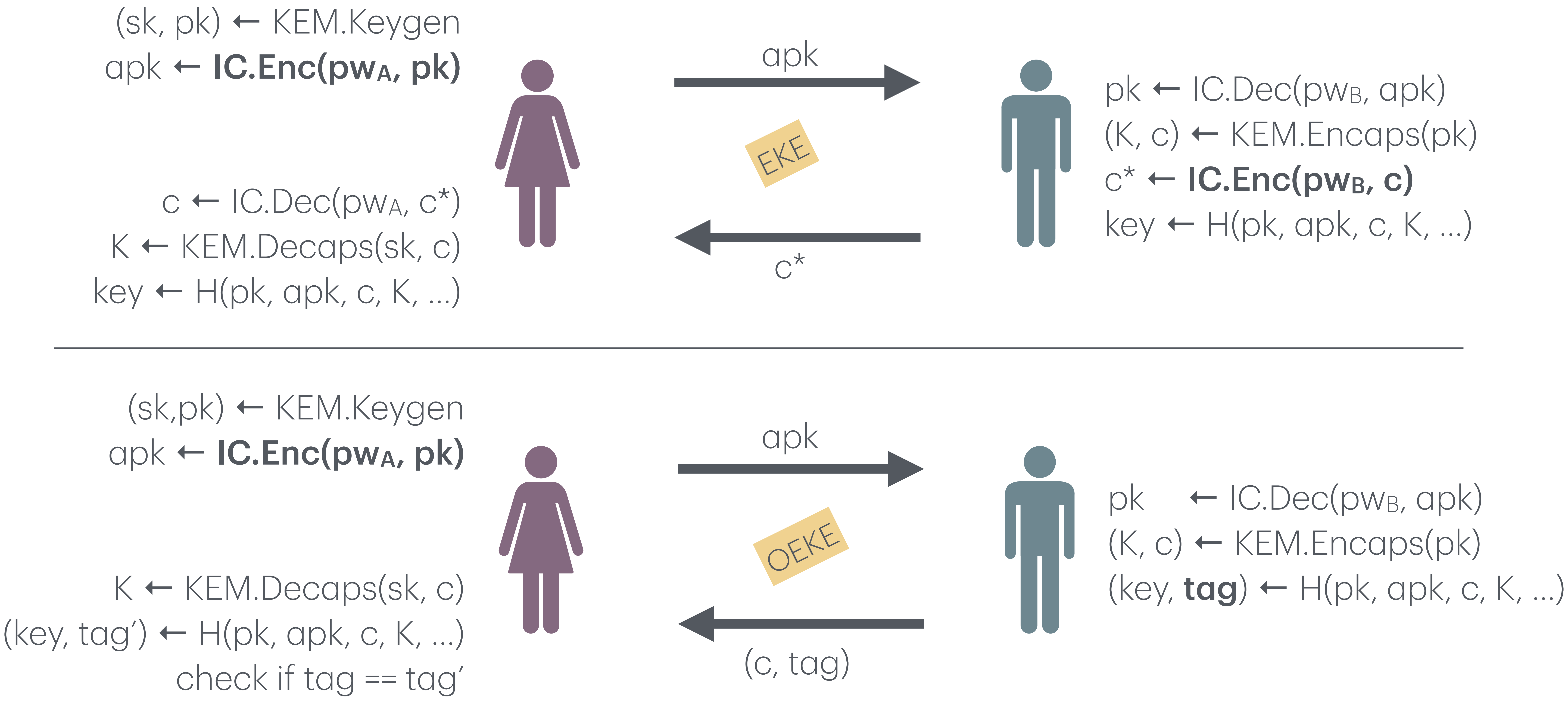


PAKE: Password Authenticated Key Exchange

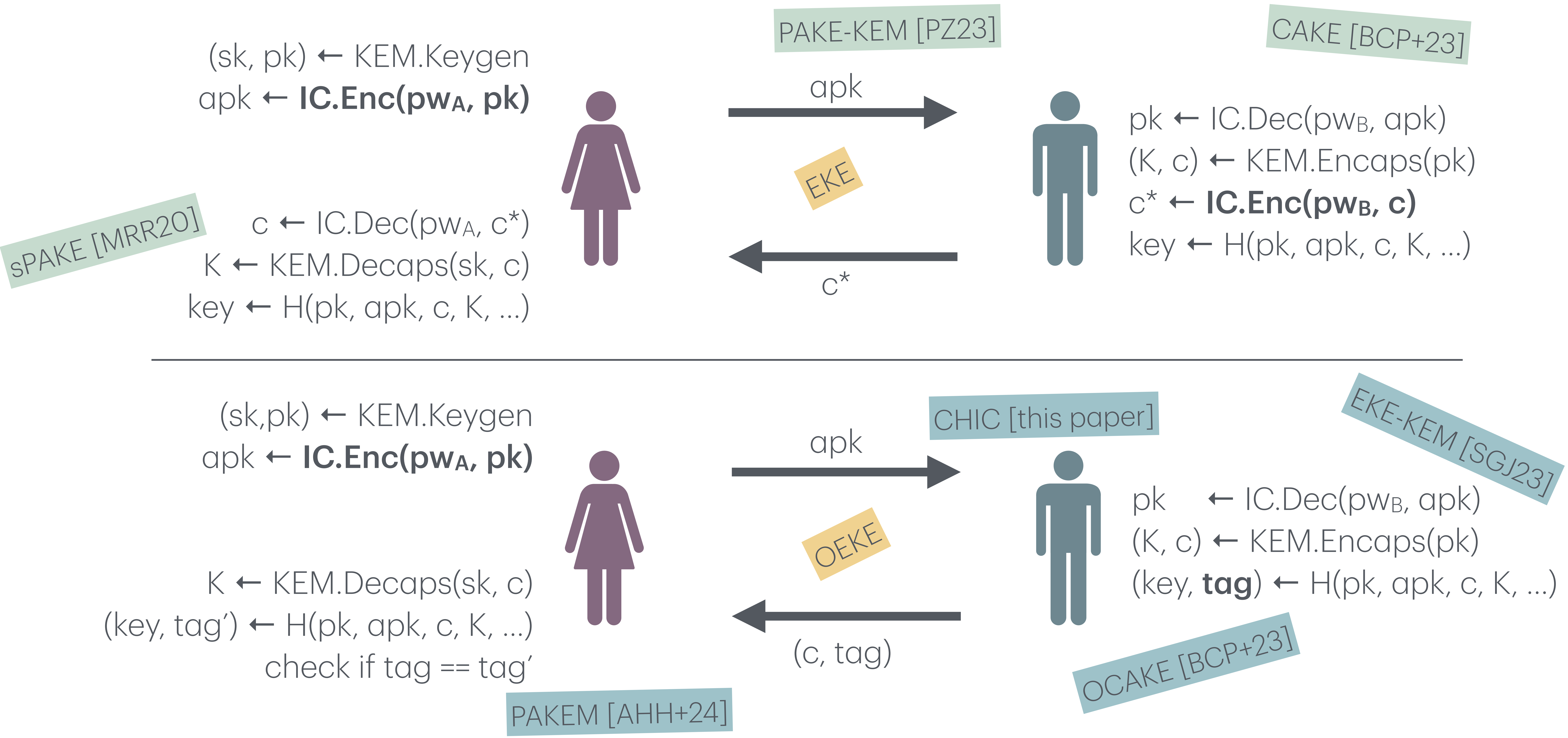
Towards post-quantum security...



EKE and OEKE design patterns



EKE and OEKE design patterns



Challenges in instantiating IC over groups

- Usually, ideal ciphers are instantiated with block ciphers.
- But what if we have an IC over a group?
- The domain of the IC and its instantiation must coincide; otherwise, it becomes trivial for an adversary to mount an offline dictionary attack. The attack proceeds as follows:

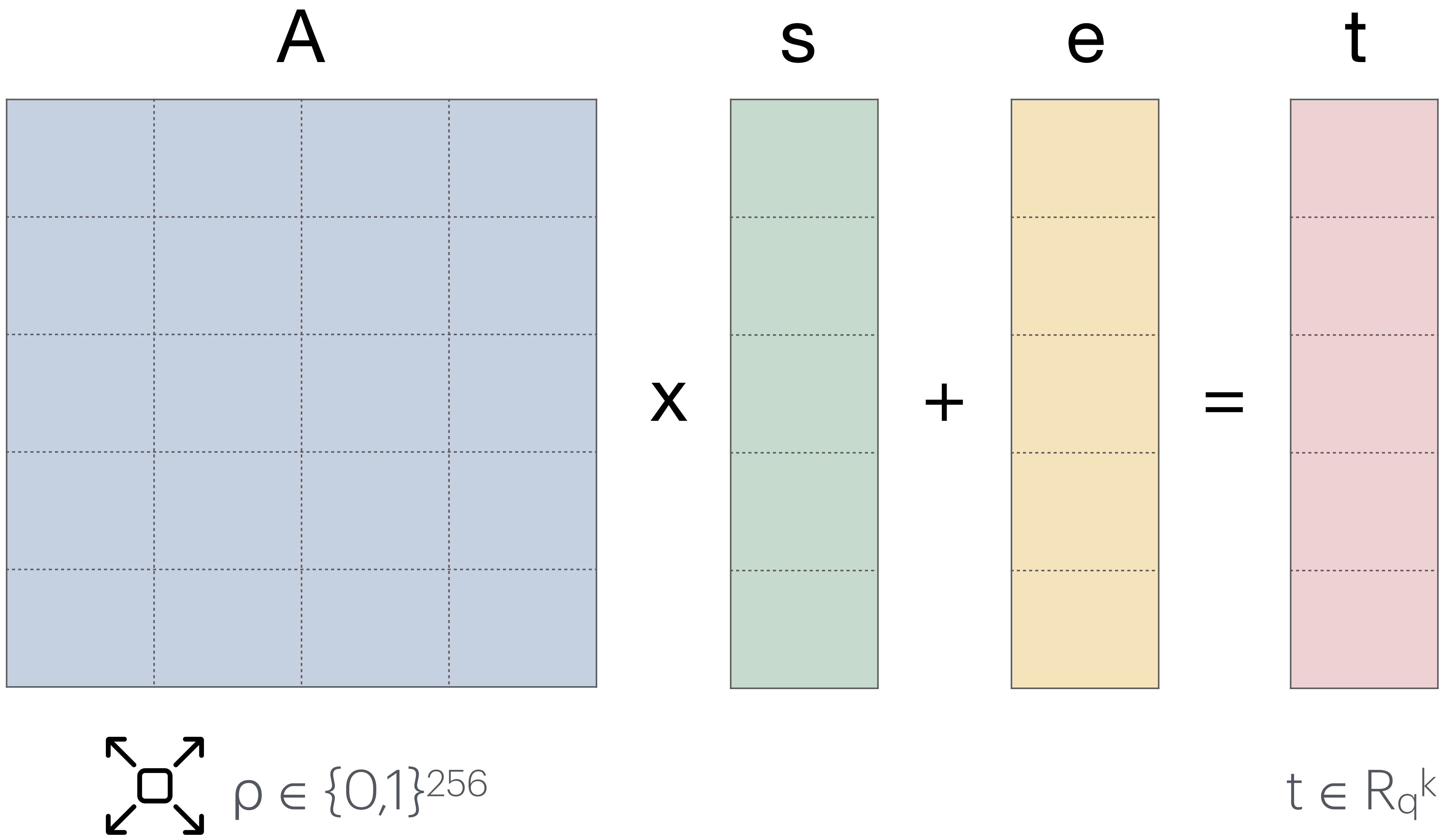
    1. Intercept apk = IC.Enc(pw, pk).

    2. Decrypt apk using a candidate password pw*.

    3. If pk* = IC.Dec(pw*, apk) does not belong to the public key space, then pw* is incorrect.

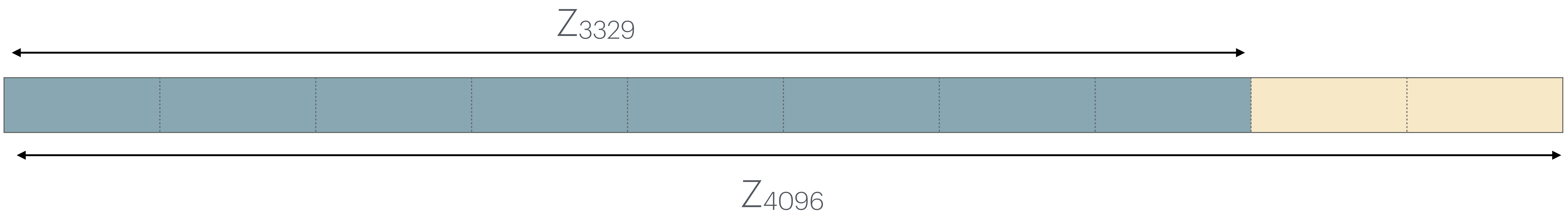
ML-KEM public keys

- Sample $\rho \in \{0,1\}^{256}$
- $A \leftarrow \text{Expand}(\rho) \in R_q^{k*k}$
- $t \leftarrow A * s + e$
- $pk \leftarrow (\rho, t)$

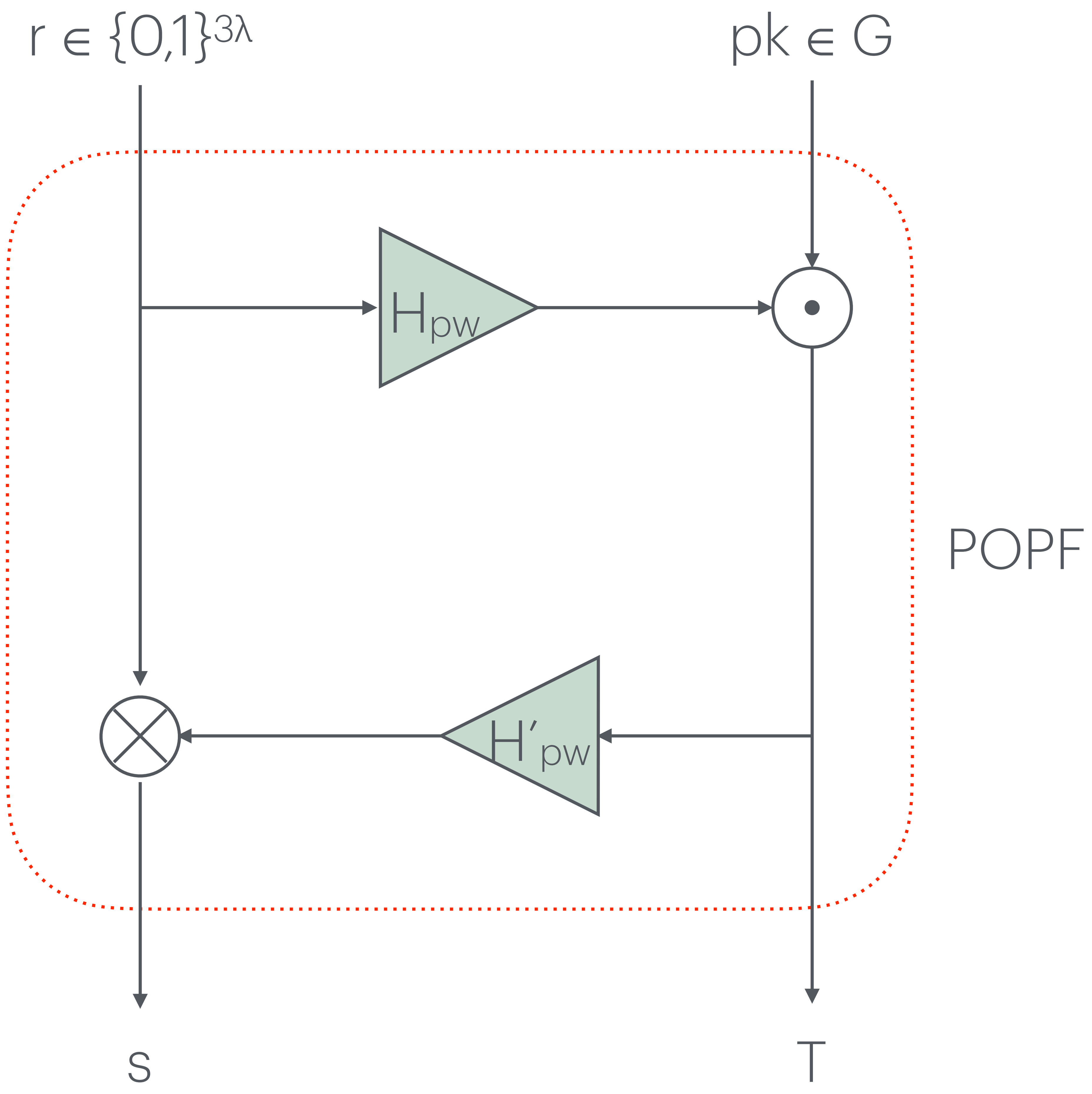


ML-KEM public keys

- R_q is a ring where the elements are polynomials with coefficients in Z_q
- FIPS 203 specifications set $q = 3329$
- $2^{11} < 3329 < 2^{12}$
- To encode a single element in Z_{3329}, we need 12 bits

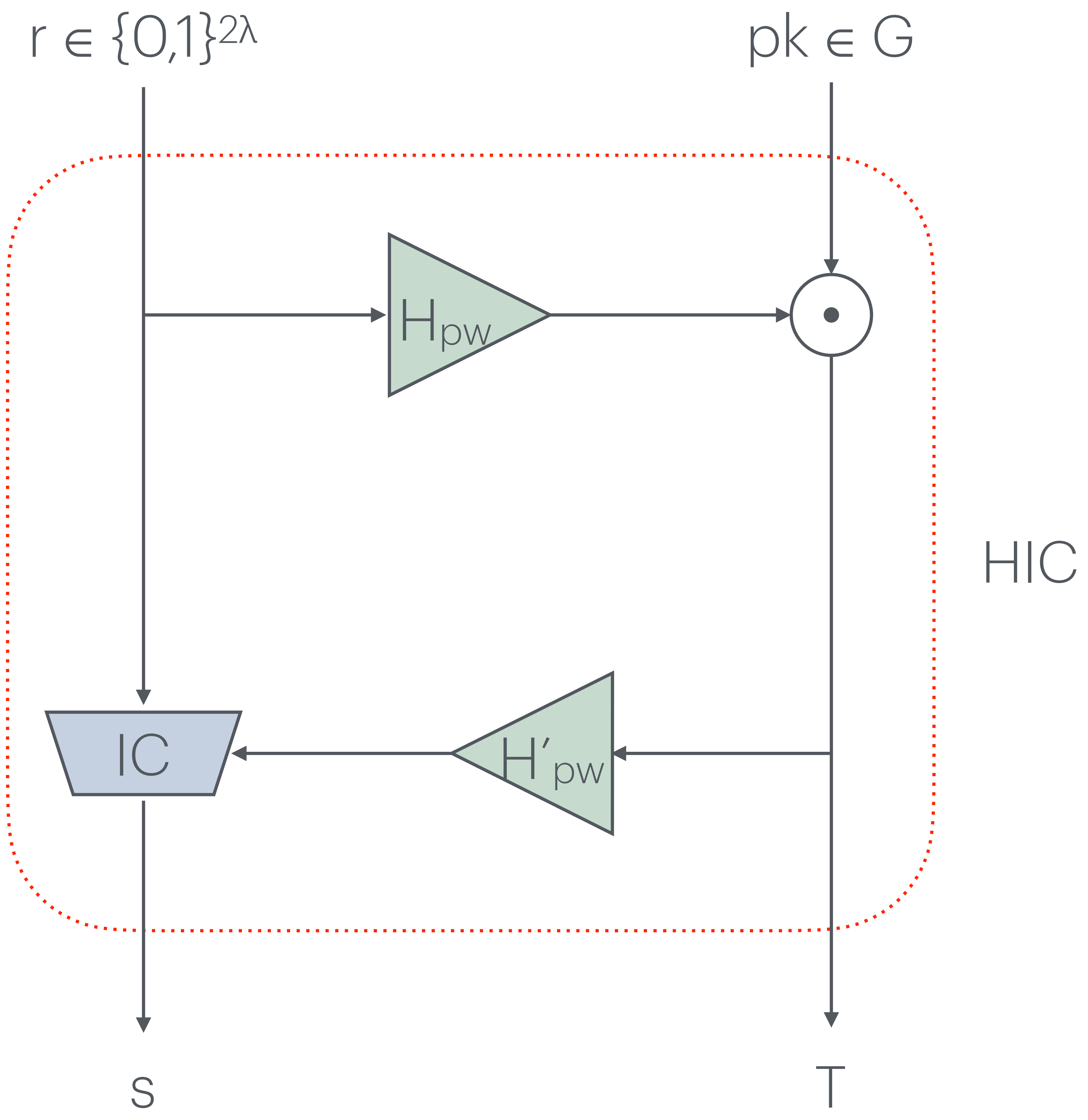


2-round Feistel [MRR20]



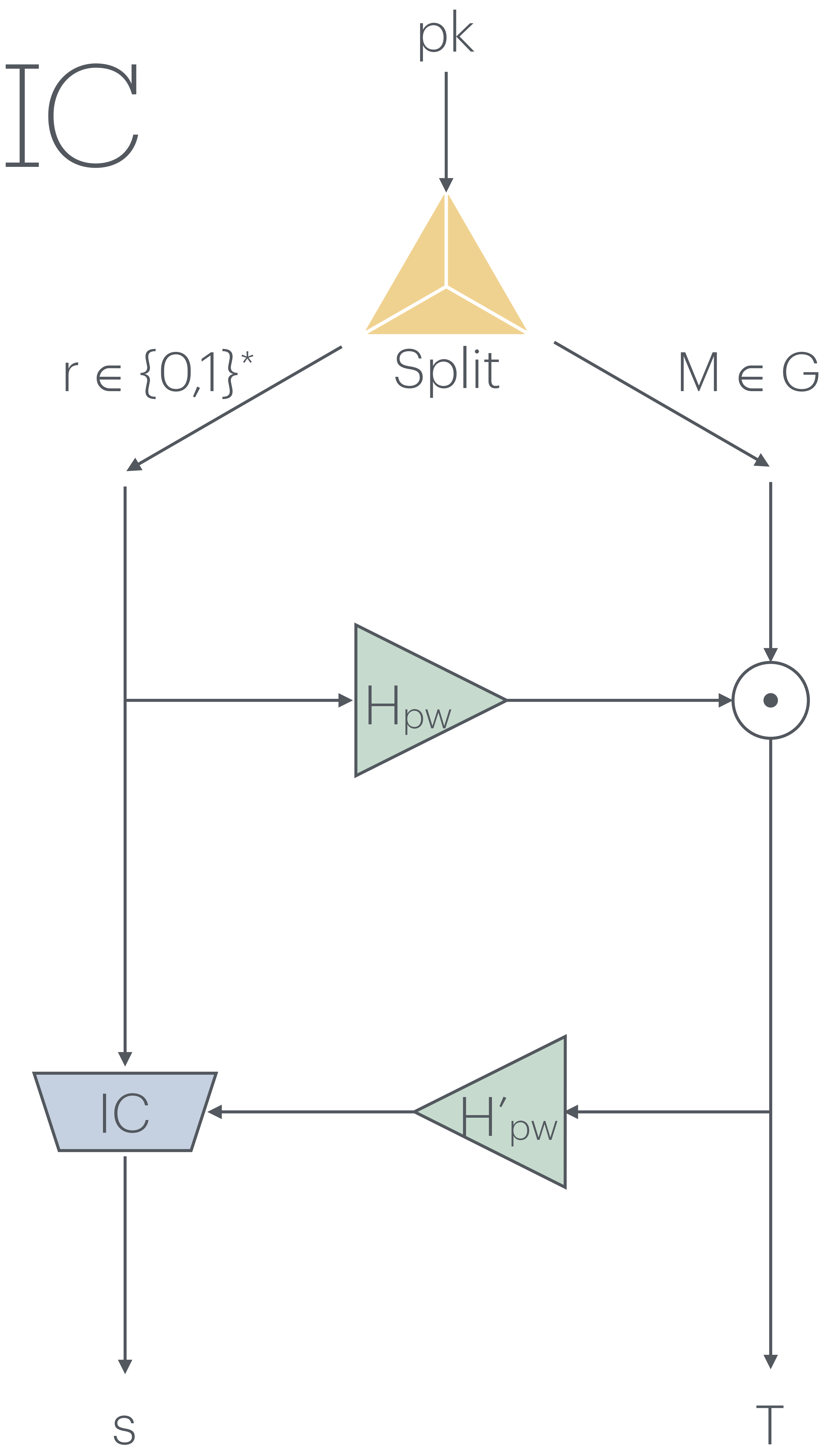
- Password-encrypt with a 2-round Feistel (2F).
- Public key on the right wire; random coins on the left wire.
- Ciphertext expands pk.
- Only one hash onto group needed instead of IC over the group.
- Modular approach with game-based definition POPF. However, 2F permits some malleability and POPF is insufficient for (O)EKE constructions.

A modified 2-round Feistel [SGJ23]



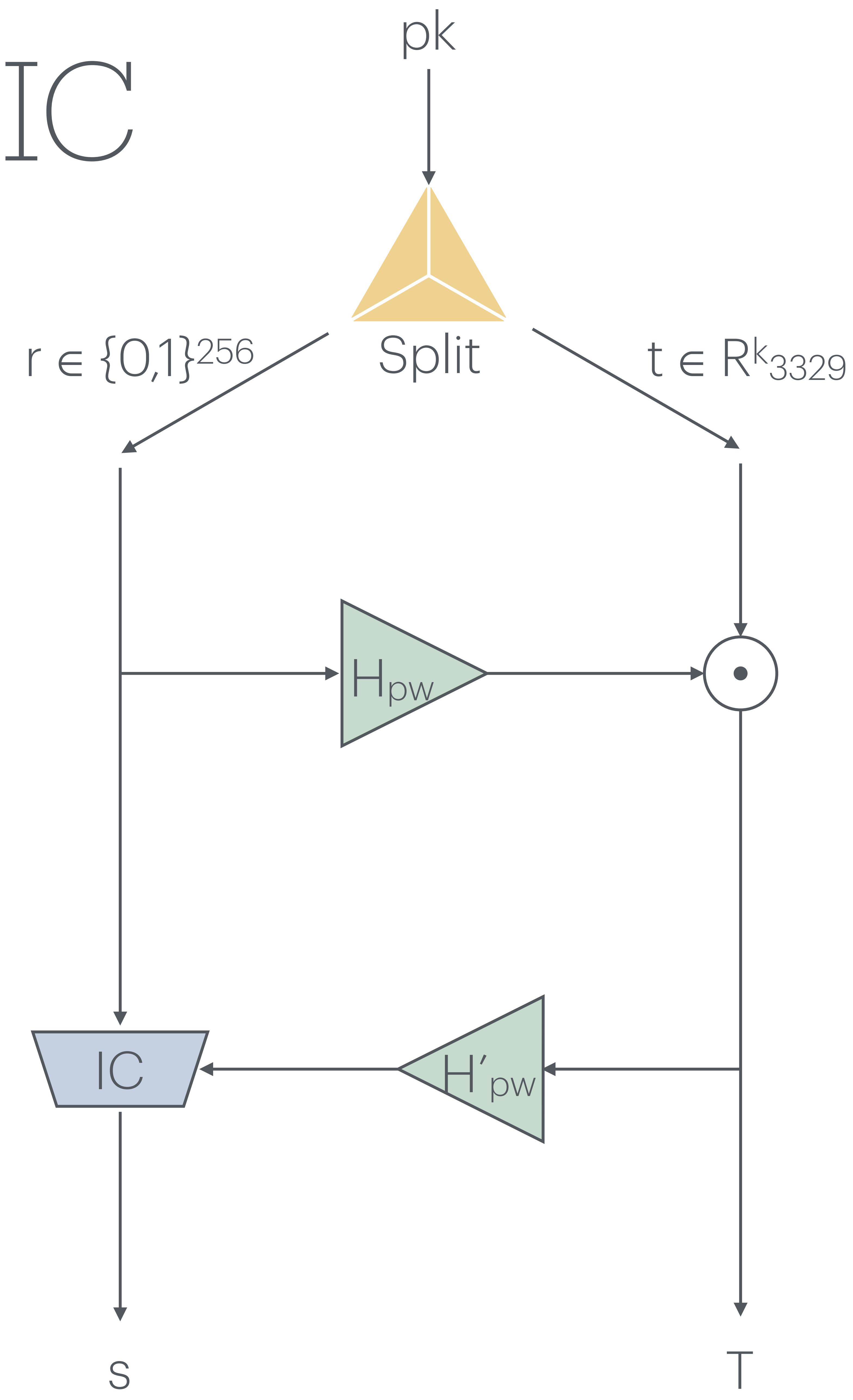
- Replace XOR operation with IC to avoid malleability
- IC over bit strings — easy to instantiate
- Modular approach with UC definition: Half-Ideal Cipher (HIC)
- Why "half-ideal"? The s-part is random.

Compact HIC



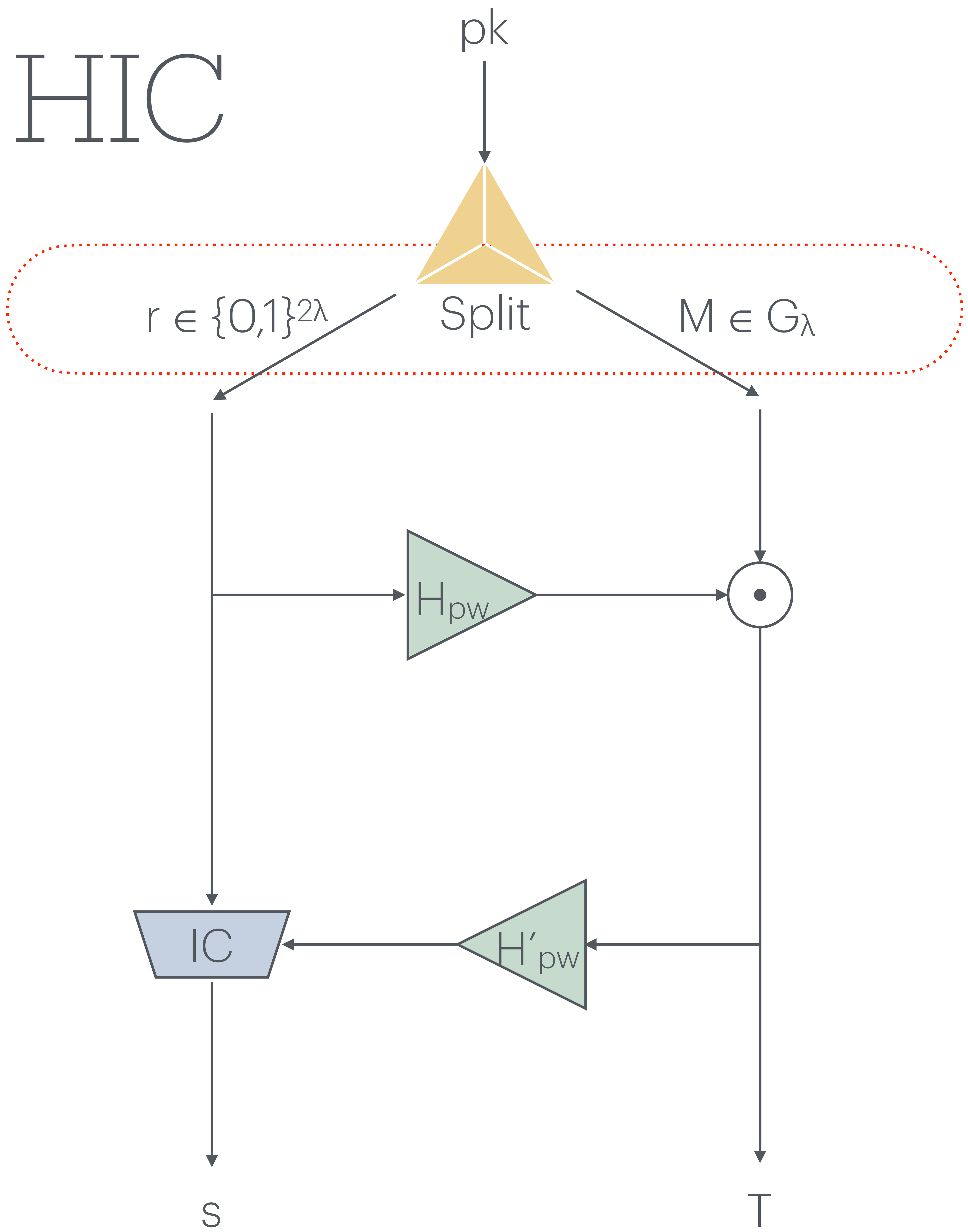
- Can we split the public and feed both wires of the 2F?

Compact HIC



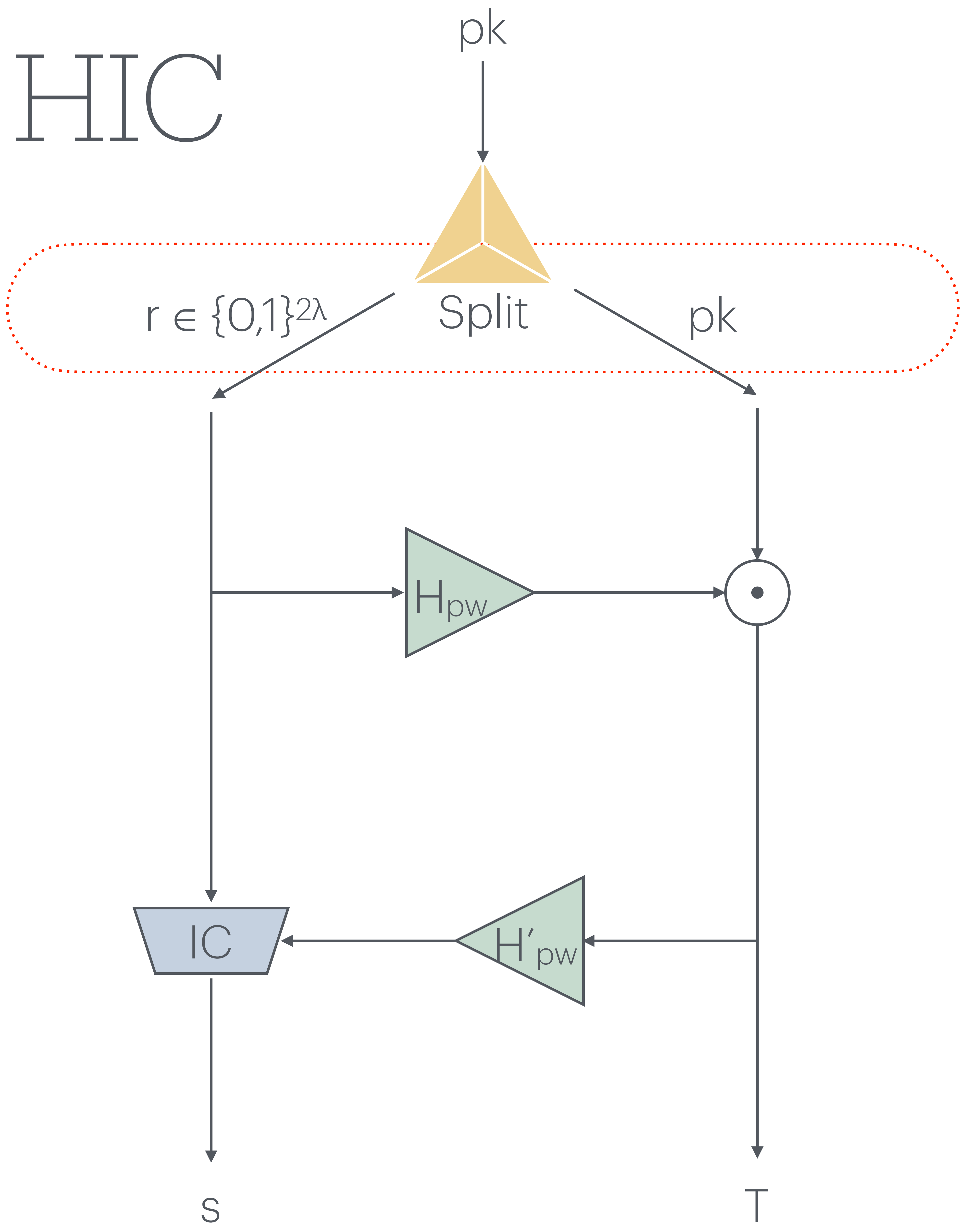
- In particular, we are interested in the trivial breakdown of ML-KEM public keys.
- Compact: avoids ciphertext expansion!
- Effectively, de-randomizes HIC.

Compact HIC



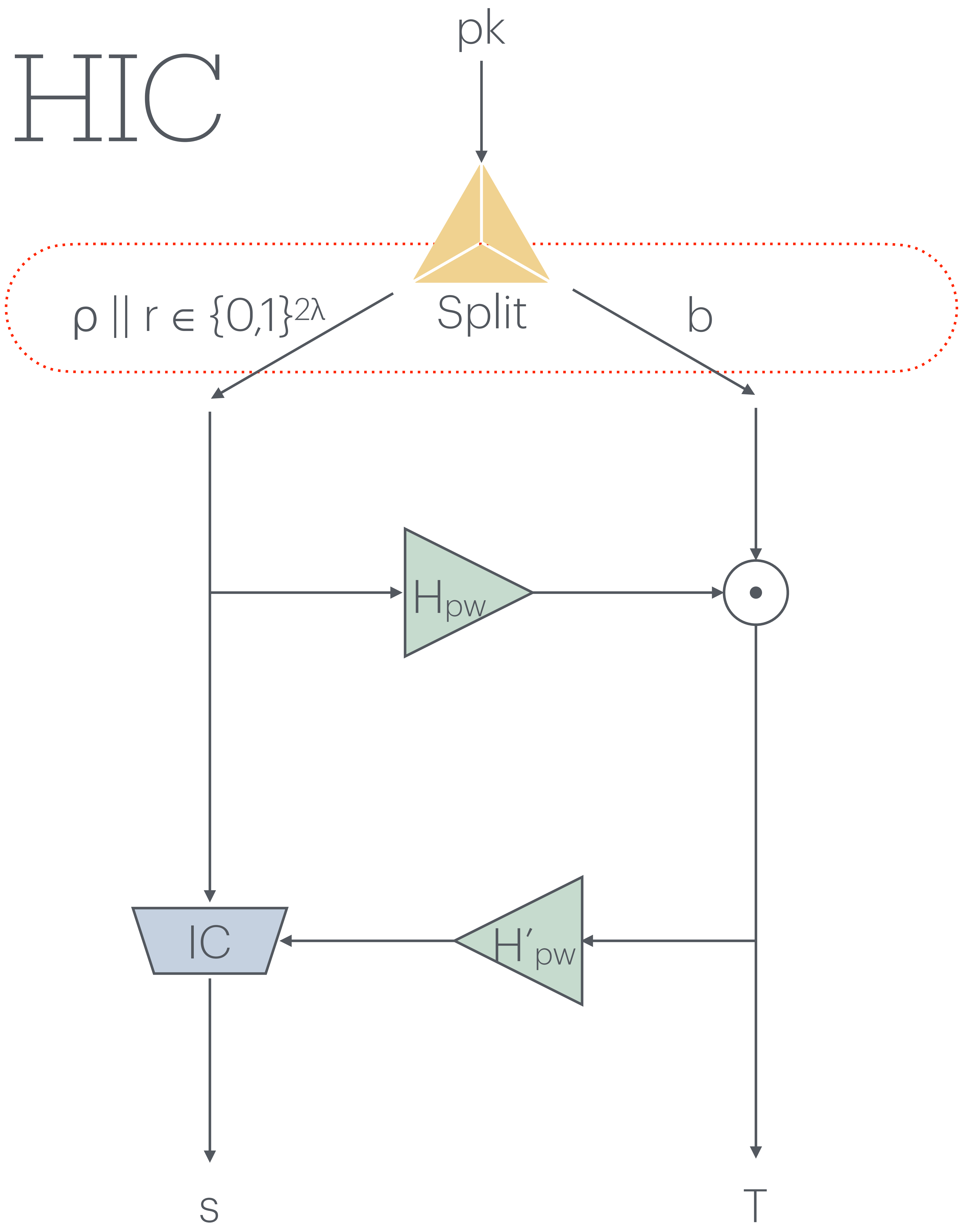
- In general, (O)EKE constructions require public keys to be uniformly distributed.
- HIC requires uniform r.
- We can combine both requirements into a single definition w.r.t. function Split.
- Experiment UNI-PK:
  $(\_, pk_0) \leftarrow$ Keygen
  $(r_0, M_0) \leftarrow Split(pk_0)$
  $(r_1, M_1) \leftarrow \{0,1\}^{2\lambda} \times G_\lambda$
  $b \leftarrow \{0,1\}$
  $b' \leftarrow A(r_b, M_b)$
  return $b == b'$

Compact HIC



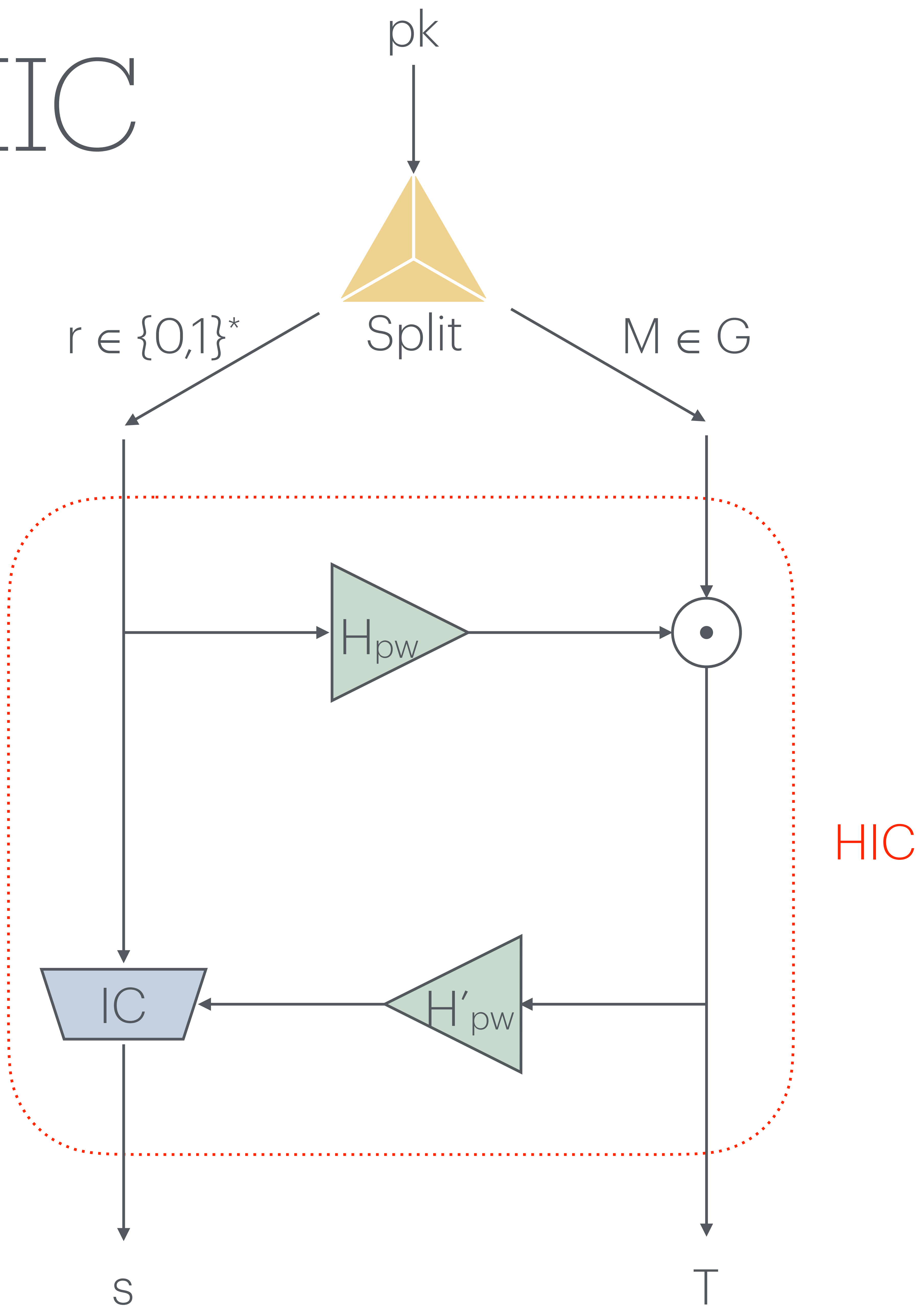
- The randomized identity Split recovers standard public key uniformity (also know as *fuzziness* [BCP+23]).
- Leads back to the HIC construction [SGJ23].

Compact HIC



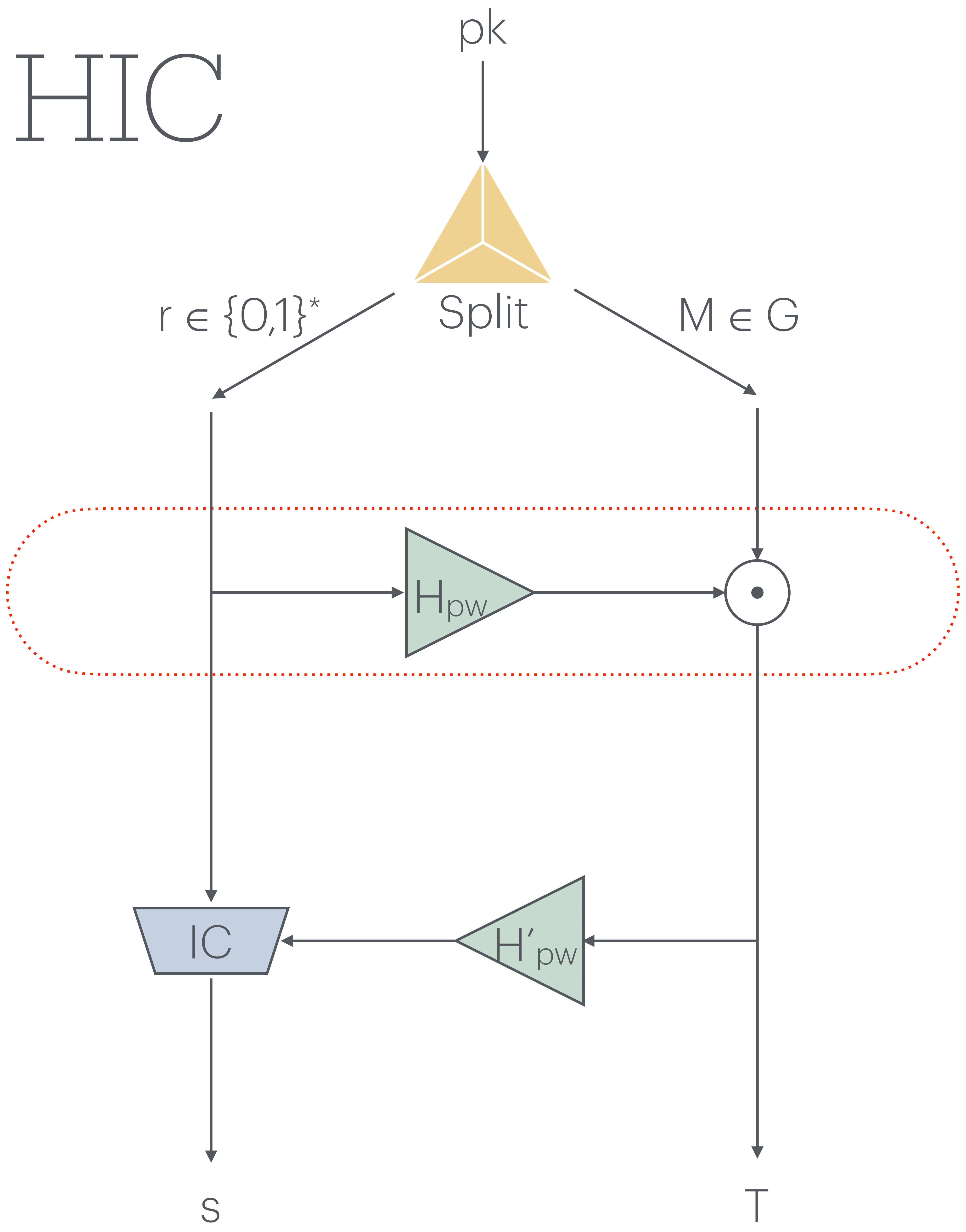
- FrodoKEM determines seed $\rho \in \{0,1\}^{128}$.
- A Randomized Split can easily extend FrodoKEM keys by appending random bits to ρ, ensuring it reaches the required length.
- This approach requires no modifications to FrodoKEM.

Compact HIC



- F_{HIC} features honest interfaces accessible by the environment Z, but no control over randomness r is provided.
- Unfortunately, we lose the modular HIC abstraction.
- Solution: Direct proof.

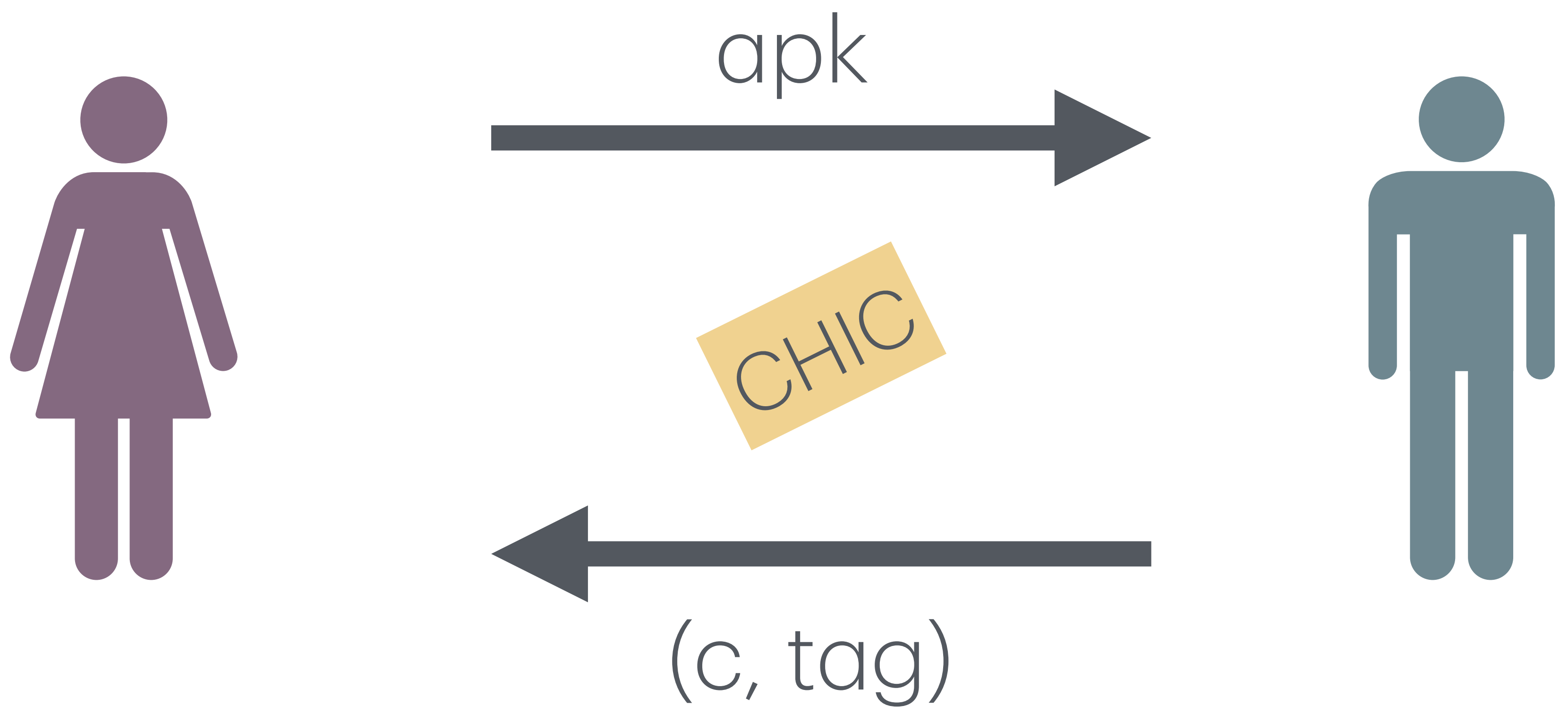
Compact HIC



- How to instantiate hash-onto-group H?
  - ★ FrodoKEM makes it easy because it uses power-of-two modulus. Simply use an eXtendable Output Function (XOF).
  - ★ For ML-KEM we reuse the rejection sampling procedure used to expand ρ into a matrix A of polynomial coefficient modulus 3329.

The CHIC protocol

(sk, pk) ← KEM.Keygen
(r, M) ← Split(pk)
(s, T) ← $m2F_{pw}$(r, M)
apk ← (s, T)



(s, T) ← apk
(r, M) ← $m2F^{-1}_{pw}$(s, T)
pk ← $Split^{-1}$(r, M)
(K, c) ← KEM.Encaps(pk)
(key, tag) ← H(pk, apk, c, K, ...)

K ← KEM.Decaps(sk, c)
(key, tag') ← H(pk, apk, c, K, ...)
check if tag == tag'

Requirements from KEM

CHIC UC-realizes F_{PAKE} [CHK05] in the RO and IC model provided that KEM has the following properties:

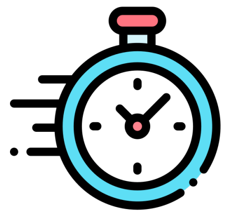
- **One-wayness of ciphertexts**: OW-CPA, but OW-PCA leads to tighter reduction.
- **Anonymity of ciphertexts**: ANO-PCA. Actively-secure KEM is necessary.
- **Uniformity of public keys**: UNI-PK.

Benchmarks

Experimental results in microseconds. Comparison of execution times of CHIC participants (two initiator stages and responder single stage) with respect to key exchange using only a CPA or CCA Kyber KEM.

	CPA KEM			CCA KEM			CHIC		
	Keygen	Enc	Dec	Keygen	Enc	Dec	Start	Resp	End
Kyber512	25	29	9	45	49	12	70	74	14
Kyber768	28	36	41	49	59	65	75	85	93
Kyber1024	36	56	53	61	87	83	89	123	117

Final remarks / future work

On timing-attacks 

- The rejection-sampling of ML-KEM is not constant-time, meaning Keygen is not constant-time.

- Timing attacks potentially affect any ML-KEM to PAKE compiler, regardless of how we instantiate the hash onto group H.

- FIPS 203 allows to limit the number of iterations of SampleNTT to 280, with a probability of failure of 2^{-261}.

- We are exploring how to implement a constant-time Keygen algorithm for ML-KEM and use it to instantiate the hash-to-group operation.

Final remarks / future work

On IC-256-256

- AES has only been standardized only with a 128-bit block size. However, our proofs requires avoiding collisions across different block cipher keys. Therefore, we need to instantiate the IC with a block cipher with 256-bit size blocks and 256-bit length keys.

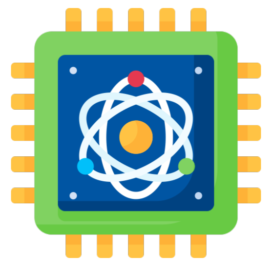
- To meet this requirement, we used Rijndael-256-256. If AES must be used, domain extenders can provide a solution [CDMS10].

- NIST is currently considering standardizing Rijndael-256-256 [edu.lu/pdmd3].

NIST

Final remarks / future work

On quantum adversaries



- The current proof is in the RO and IC model.
- However, a *harvest-now-decrypt-later* attack is ineffective against CHIC because the ciphertexts are post-quantum secure.

Thank you for your attention